Universidad Internacional de La Rioja
Escuela Superior de Ingeniería y Tecnología

Máster Universitario en Seguridad Informática

Aplicación de un ciclo de vida de desarrollo software seguro en un sistema de vigilancia militar

Trabajo fin de estudio presentado por:	Mónica Arribas Serrano
Tipo de trabajo:	Piloto Experimental
Director/a:	Dr. Javier Bermejo Higuera
Fecha:	02/03/2022

Resumen

Habitualmente los sistemas militares se ejecutan en redes aisladas, cuya securización típica es de nivel físico. Esto lleva a olvidar en muchos casos la securización del software depositando una confianza no justificada en la seguridad de estos sistemas.

En este piloto se pretende demostrar la afirmación anterior de la poca o nula securización de las aplicaciones de mando y control con el software militar Magec y la aplicación de un Ciclo de vida de desarrollo seguro de software (S-SDL). Para ello, se comenzará con un modelado de amenazas sobre la arquitectura de la aplicación con la herramienta Microsoft Threat Modeling Tool. Se continuará con el proyecto de una auditoría de código de la aplicación con la herramienta Fortify SCA. Y se finalizará con la realización de pruebas de penetración de la aplicación con las herramientas Nessus, Spike y Wireshark, siendo estas para escanear vulnerabilidades conocidas y para realizar pruebas de fuzzing respectivamente.

Palabras clave: Ciclo de vida de desarrollo seguro de software, aplicación de mando y control, modelado de amenazas, auditoría de código, pruebas de penetración.

Abstract

Military systems typically run on isolated networks, which are typically secured at the physical level. In many cases, this leads to neglecting software security and placing unwarranted confidence in the security of these systems.

This pilot is intended to demonstrate the above assertion that there is low or no security of command and control applications with Magec military software and the application of a Secure Software Development Lifecycle (S-SDL). This will start with a threat modelling of the application architecture with the Microsoft Threat Modeling Tool. It will continue with the project of a code audit of the application with the Fortify SCA tool. And it will end with pen testing of the application with the tools Nessus, Spike and Wireshark, these being for scanning known vulnerabilities and fuzzing tests respectively.

Keywords: Secure software development life cycle, Command and control app, Threat modelling, Code auditing, Pen-testing.

Índice de contenidos

Índice de figuras

Índice de tablas

1. Introducción

Para comprender la evolución actual de los sistemas de vigilancia militares, se debe comprender cómo han sido durante la historia. A lo largo de la historia militar se han pasado por diferentes tipos de sistemas de vigilancia. Pasando por el personal (personas que se dedican a cubrir el perímetro e ir informando de los diferentes eventos que iban ocurriendo), por cámaras de control de fronteras con vigilancia constante (personas que se dedican a controlar las cámaras y visualizar el contenido en directo) o hasta incluir diferentes sensores (radares, cámaras térmicas y de día, antenas, inhibidores de frecuencia...) en una única interfaz con backup de información.

La historia de la seguridad de estos sistemas de vigilancia ha evolucionado de seguridad personal a seguridad informática. Esto conlleva a que hay que realizar diferentes acciones sobre los sistemas de información involucrados en estos procesos de vigilancia.

La aplicación de esta seguridad informática en sistemas militares siempre se ha basado en ser una securización de tipo física, es decir, se securizan las redes y los equipos que van a ser usados en el entorno militar. Para llevar a cabo estas securizaciones se hace uso de las guías CCN-STIC. (Nacional, CCN CERT CNI, 2021) muestra todas las series que actualmente existen sobre procedimientos, normas y securización de equipos de diferentes sistemas operativos, controles de acceso y redes. Pero a pesar de tener securizada la parte física de los sistemas militares, actualmente no existe una norma o guía de cómo securizar software de uso militar.

El software militar, a pesar de estar en una red aislada y securizada, debería estar correctamente securizado con el fin de evitar vulnerar los tres pilares de la seguridad de la información, que son la confidencialidad, disponibilidad e integridad.

Para evitar cualquier ataque en este tipo de software, se pueden plantear preguntas del tipo: ¿Cómo se debe securizar este tipo de software?, ¿Es posible securizarlo como uno de uso civil?, o, ¿Está expuesto realmente el software a ataques vulnerables en estos tipos de sistemas?

Las respuestas de cada una de las tres preguntas anteriores se irán respondiendo a lo largo de este Piloto Experimental.

1.1. Motivación y justificación

"Cada vez más, la seguridad y la privacidad del software se están convirtiendo en problemas importantes para la sociedad. Casi todos los días nos enteramos de nuevos ataques y problemas de privacidad, y cada vez más afectan no solo a las grandes empresas, sino a todo el mundo" (Weir, Hermann, & Fahl, 2020) (p.2).

En concreto, las aplicaciones militares son más vulnerables frente a ataques que buscan obtener información táctica o cualquier otro tipo de información. Por tanto, es evidente que es necesario hacer un cambio en la forma de hacer y generar software militar.

En la mayoría de los casos, las aplicaciones software militares se encuentran en una red aislada securizada según las guías CCN-STIC (Nacional, CCN CERT CNI, 2021). Pero, como estamos comprobando, con securizar las redes y los equipos no es suficiente para liberarse de posibles ataques.

Actualmente, no existe ningún tipo de requisito relacionado con la securización del software que se ejecuta en esas máquinas ni con el cifrado de las diferentes comunicaciones con las mismas. Es por ello, que, si el atacante consiguiera franquear la seguridad a nivel de red, serían capaces de obtener cualquier tipo de información que manejen las diferentes aplicaciones que se ejecutan en cada una de las máquinas.

La principal causa de no existir ningún tipo de requisito relacionado con la securización del software se debe a que no han sucedido un incremento del número de ataques a los sistemas de la información militar hasta los últimos años. Según (Nacional, Ciberamenazas y tendencias Edición 2020, 2020): "En 2019, el CCN-CERT gestionó 42.997 ciber incidentes –más de un 11 % con respecto al año anterior–, de los cuales casi un 7,5 % fueron de peligrosidad muy alta o crítica" (p.17).

Debido al aumento de los ciber incidentes y que actualmente todo tiende a lo virtualizado, es para considerar la necesidad de realizar algún procedimiento que mejore la seguridad del software militar para que sea más robusto frente a los ataques.

Por estas razones, se cree conveniente aplicar un ciclo de vida del software seguro. Donde se realice un proceso de mejora de la seguridad durante todo el ciclo de vida del software, desde su arquitectura inicial pasando por su desarrollo y finalizando por su integración en cliente.

Según afirma (Bermejo Higuera, 2021): "El objetivo es producir un software más seguro y confiable" (p.7).

1.2. Planteamiento del problema

Partiendo de lo expuesto en el apartado **Motivación y justificación** del problema planteado, se propone la aplicación de un ciclo de vida del software seguro (S-SDLC) sobre el sistema de vigilancia militar optado con el fin de mejorar su seguridad mediante la aplicación de buenas prácticas de seguridad en todas y cada una de sus fases del ciclo de vida.

Partiendo de la idea anterior, (McGraw, 2005) identifica un modelo de ciclo de vida de desarrollo software basado en diferentes actividades o buenas prácticas de seguridad. De estas actividades identificadas se realizarán las siguientes: modelado de amenazas, auditoría de código y pruebas de penetración.

Para la primera actividad, el modelado de amenazas, se propone hacer uso de la herramienta Microsoft Threat Modeling Tool (Microsoft, Microsoft Threat Modeling Tool, 2017). Esta herramienta hace un cálculo de las posibles vulnerabilidades encontradas en una arquitectura y los diferentes riesgos a los que se enfrenta.

Para la siguiente actividad, auditoría de código, se propone hacer uso se la herramienta Fortity Static Code Analyzer de la empresa Fortify Inc (Focus, 2021). Esta herramienta es para realizar análisis de código estático (SAST).

La última actividad, pruebas de penetración, donde se propone, realizar un escaneo de vulnerabilidades conocidas con la herramienta Nessus (Chalvatzis, A. Karras, & C. Papademetiou, 2019) y las pruebas de fuzzing con la herramienta Spike (Bradshaw, 2010) y Wireshark (Wireshark, 2021).

Por medio del anterior planteamiento se pretende mejorar la seguridad de las aplicaciones de escritorio militares, con el fin de detectar en la fase más temprana las vulnerabilidades potenciales a las que se enfrenta el software y a los posibles ataques futuros.

1.3. Estructura del trabajo

En este apartado se describe brevemente el contenido de cada uno de los siguientes capítulos que componen el presente TFM.

- Contexto y estado del arte: Durante este capítulo, se busca determinar el contexto de las aplicaciones de mando y control militares con respecto a la seguridad de las aplicaciones softwares desplegadas en ese entorno y cómo aplicarlo.

  Para ello, se realiza una introducción a los sistemas de mando y control militares, se explica el concepto de Ciclo de Vida de Desarrollo Software Seguro, las herramientas a usar para aplicar unas actividades concretas del Ciclo de Vida de Desarrollo Software Seguro, y se aportan las conclusiones obtenidas durante este capítulo.
- Objetivos concretos y metodología de trabajo: Una vez obtenido un contexto y una idea del estado de la seguridad en las aplicaciones militares, se determinan el objetivo general y los objetivos específicos del Piloto Experimental, y se propone la metodología de trabajo del Piloto Experimental.
- Descripción detallada del Experimento: A lo largo de este capítulo se va a ir desarrollando el Piloto Experimental. Para ello se comienza con una descripción de la aplicación a analizar (Magec); se continúa con la actividad del S-SDLC llamada Modelado de amenazas con la herramienta Microsoft Threat Modeling Tool; seguidamente se realiza la Auditoría de Código fuente mediante el uso de la herramienta Fortify SCA; y se finaliza con el proceso de las pruebas de penetración, tanto con la herramienta Nessus para el escaneo de vulnerabilidades, como la herramienta Strike para las pruebas de fuzzing.
- Descripción de los resultados: Se estudian los resultados obtenidos a lo largo del capítulo anterior. Se obtienen diferentes gráficos de los resultados obtenidos de las actividades realizadas del S-SDLC y se evalúa el nivel de seguridad.
- Conclusiones y prospectiva: En este último capítulo, se realiza un análisis del Piloto Experimental y de los resultados obtenidos con el fin de llegar a obtener unas lecciones aprendidas, y se identifican las posibles futuras líneas de trabajo.

2. Contexto y Estado del Arte

Se ha realizado una búsqueda de la temática del presente piloto y no se ha podido encontrar nada relacionado con la aplicación de un S-SDLC en una aplicación de escritorio militar.

2.1. Introducción a los sistemas de mando y control militares

Los seres humanos desde que tienen conciencia social han tratado de ejercer sistemas rudimentarios de mando y control. Ya sea la comunicación del jefe de una tribu dirigiendo la caza de los animales como el de establecer las diferentes tareas a realizar cada persona de la tribu. (Cubeiro Cabello, 2001)

Con la evolución de la civilización, los sistemas de mando y control fueron evolucionando. Se comenzó con el uso de maquetas y mapas donde colocar las tropas y las formas de ataque y cifrado de las diferentes comunicaciones entre puestos de ataque. Seguidamente apareció el telégrafo como primer elemento electrónico usado en el entorno militar durante la Primera Guerra Mundial, a pesar de ser vulnerable y de tener una transmisión lenta de las comunicaciones. Con la aparición de los primeros ordenadores durante la Segunda Guerra Mundial todo cambió, ya que las comunicaciones y capacidades de transmisión aumentaron considerablemente. (Cubeiro Cabello, 2001)

A lo largo de los años, en concreto a finales de los años XX, aparecen unas aplicaciones de apoyo al mando y control que se instalan en ordenadores y que son capaces de comunicarse entre las diferentes unidades militares, llegando a ser capaces de comunicarse con otras unidades de mando militares de otros países. Estos sistemas son los conocidos por las siglas C4I (Command, Control, Communications, Computers and Intelligence). (Díaz del Río Durán, 2011) (Cubeiro Cabello, 2001), (V Congreso Nacional de i+d en Defensa y Seguridad, 2017) y (Arechaga Tarruell, 2017).

El fin de estos sistemas de mando y control es conseguir una integración y explotación de los diferentes sensores que puedan verse involucrados (radar, sonar, cámaras, inhibidores de frecuencia...). Estos sistemas se conocen como C4ISR (Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance). En este grupo de clasificación es en el que se encuentra el sistema de vigilancia militar que se va a analizar. (Díaz del Río Durán,

2011), (Cubeiro Cabello, 2001), (V Congreso Nacional de i+d en Defensa y Seguridad, 2017) y (Arechaga Tarruell, 2017)

(Clausewitz, 1999) dijo que en el campo de batalla se producía la niebla de guerra. Este término se refiere a la confusión que ocurre durante una batalla en la coordinación y planificación de las diferentes operaciones. Es decir, cuando un alto cargo está organizando las diferentes maniobras durante la batalla hay ciertas zonas que no se cubren, ya que es difícil controlar todo el campo de batalla. Los diferentes sistemas de mando y control, ya sea el C4I o C4ISR, intentan solventar la niebla de guerra, mediante comunicaciones más fluidas y con la mayor información del campo de batalla.

Debido a esta evolución en los sistemas de mando y control durante el siglo XXI, se comienzan a tener otros riesgos y amenazas no tradicionales como son los ataques a través del ciberespacio. Debido a esto, actualmente el escenario estratégico cambia hacia una gestión de las crisis y resolución de los conflictos de manera diferente a la que se aplicaba anteriormente. (Díaz del Río Durán, 2011)

Todas las comunicaciones dentro de un sistema de mando y control dependen de las redes y equipos informáticos, debido a que actualmente hay una mayor dependencia tecnológica y, con ello, una mayor probabilidad de sufrir ciberataques. Las consecuencias que conllevaría sufrir un ciberataque pueden ser políticas, económicas o intelectuales. Estos ataques pueden llevarse a cabo por hackers, organizaciones terroristas o, incluso, los sistemas de inteligencia de otro país. Debido a estas razones, el ciberespacio es un nuevo campo de batalla, en el que los heridos serían los sistemas de información y la diferente información robada pondría en riesgo las vidas humanas. (Díaz del Río Durán, 2011)

Esta ciberguerra se considera asimétrica, ya que la información de la que dependen los diferentes ejércitos y naciones puede hacerlos vulnerables. Cualquier nación, ya sea la más grande del mundo, puede ser atacada por un grupo pequeño de hackers informáticos. (Díaz del Río Durán, 2011)

Es por ello, que hay que proteger estos sistemas, y la mejor forma de realizarlo es por medio de una defensa por capas concéntricas. Para ello hay que tener un equilibrio entre la seguridad y la eficiencia del sistema, consiguiendo así una mejor operabilidad con los diferentes usuarios. (Díaz del Río Durán, 2011)

A pesar de que las redes donde está instalada la aplicación de mando y control no suelen tener conexión a Internet, es decir, es una red aislada, no se puede garantizar una seguridad al 100%. Para demostrar que es muy complicado garantizar la seguridad en las redes aisladas, se pone de ejemplo el ataque que se produjo en una central nuclear de Irán en enero de 2010. Donde un malware llamado STUXNET tomó el control de mil máquinas y las reprogramó para autodestruirse, este malware llego a la red aislada mediante la conexión a uno de los sistemas por USB (Mueller & Yadegari, 2012). Para aumentar la seguridad de estos sistemas es necesario realizar actualizaciones de los diferentes sistemas operativos y COTS (Commercial Off The Shelf) a través de internet en servidores aislados de la red securizada. (Díaz del Río Durán, 2011)

Por estas razones, las aplicaciones de escritorio militares deben ser securizadas durante todas las fases de su ciclo de vida de desarrollo. Por tanto, deberíamos realizar una securización de todo lo que está en la red aislada, es decir, comunicaciones, equipos, red y COTS, consecuentemente, el software que se usa en esta red también ha de ser securizado. De esta manera se conseguirá obtener una red más segura frente a posibles vulnerabilidades o ataques ya conocidos o no y evitar el robo de la información. Con ello se asegurarán los tres pilares de la seguridad de la información, que son, confidencialidad, disponibilidad e integridad.

2.2. ¿Cómo se aplica un S-SDLC?

Partiendo del contexto explicado en el apartado anterior (**Introducción a los sistemas de mando y control militares**), se obtiene la conclusión que es necesario aplicar seguridad durante el ciclo de vida del software. Todo lo descrito a continuación se basa en lo dicho en (Bermejo Higuera, 2021)

Figura 1. *Propiedades de un software seguro.*

Fuente: (Bermejo Higuera, 2021).

Para que un software sea seguro debería cumplir las propiedades esenciales siempre y las propiedades complementarias para conseguir una mejor securización. Las propiedades esenciales son confidencialidad, disponibilidad e

integridad; y, las complementarias son autenticación, fiabilidad, resiliencia, robustez, tolerancia y trazabilidad.

A continuación, se definen cada una de las propiedades, en primer lugar, las esenciales y posteriormente las complementarias:

- Confidencialidad: la propiedad que garantiza que el software mantiene las características y los activos ocultos a usuarios no autorizados.
- Disponibilidad: la propiedad que garantiza que el software es operativo y accesible en todo lugar por los diferentes usuarios autorizados, pudiendo acceder a la información y a los diferentes servicios.
- Integridad: la propiedad que garantiza que el código fuente no pueda ser modificado por diferentes usuarios no autorizados.
- Autenticación: la propiedad que garantiza que los usuarios son quienes dicen ser durante su uso en del software, es decir, se confirma que la identidad del usuario es la correcta.
- Fiabilidad: la propiedad que garantiza que el software funciona en la forma esperada en cualquier situación a la que se enfrente durante su ejecución.
- Resiliencia: la propiedad que garantiza que el software es capaz de aislar y limitar los daños producidos durante un ataque y de recuperarse al mínimo.
- Robustez: la propiedad que garantiza que el software tiene capacidad y resistencia hacia los posibles ataques.
- Tolerancia: la propiedad que garantiza que el software, a pesar de haber sufrido diferentes ataques, es capaz de seguir funcionando.
- Trazabilidad: la propiedad que garantiza quién hace cada proceso en el software, es decir, qué usuario realiza cada acción sobre el mismo.

Para que estas propiedades se cumplan haciendo que un software sea más seguro, hay que tener en cuenta los siguientes factores influyentes:

- Ambiente de operación: se deben configurar los entornos de producción de manera segura.
- Componentes adquiridos: se deben evaluar e integrar correctamente los componentes que se adquieren de otros correctamente.

- Configuraciones desplegadas: se debe tener configurado correctamente el software del entorno de producción.
- Conocimiento profesional: debe existir una concienciación de seguridad en todos los puestos que trabajan con el software.
- Herramientas de desarrollo: se deben considerar los lenguajes de programación, librerías y herramientas de desarrollo que usan los puestos dedicados al software.
- Principios de diseño y buenas prácticas de desarrollo: se deben hacer uso de buenas prácticas de desarrollo y diseño seguro.

A partir de las propiedades anteriores y los factores influyentes, es necesario conocer los principios de diseño de software seguro, qué es un ciclo de vida del software y qué es un S-SDLC. Estos conceptos se irán desarrollando en los siguientes apartados.

2.2.1. Principios de diseño de seguridad del software

Para que un software sea seguro, se deben aplicar los principios de diseño o buenas prácticas de desarrollo. Estos principios deben aplicarse tanto a aplicaciones de escritorio como a aplicaciones web, ya que todas ellas pueden verse atacadas en cualquier momento. Se parte de lo que se comenta en (Bermejo Higuera, 2021).

Los principios de diseño de seguridad a destacar son:

- Defensa en profundidad: Según (Nacional, CCN-STIC-400 Manual STIC, 2013) la Defensa en profundidad es la "Estrategia de protección consistente en introducir múltiples capas de seguridad, que permitan reducir la probabilidad de compromiso en caso de que una de las capas falle y en el peor de los casos minimizar el impacto". El *objetivo* de ese principio es la aplicación de varias capas de seguridad en los sistemas con el fin de reducir la probabilidad de que el sistema se vea comprometido.
- Diseño de software resistente: con este principio de diseño se pretende apoyar a las propiedades de resiliencia y robustez, consiguiendo el *objetivo* de reducir lo máximo posible el tiempo en el que un componente defectuoso de una aplicación no es capaz de protegerse de un ataque.
- Entorno de producción o ejecución inseguro: se deben considerar inseguras todas las aplicaciones instaladas en el equipo inseguras, debido a que pueden ser potenciales fallos en la seguridad y pueden penetrar en el perímetro de defensa de la aplicación

software. El *objetivo* es reducir las vulnerabilidades aplicando principios de validación de las entradas a la aplicación.

- Fallar de forma segura: el *objetivo* de este principio es minimizar la probabilidad de que un fallo en la aplicación pueda saltarse el mecanismo de seguridad implantado, dejando la aplicación vulnerable frente a futuros posibles ataques.
- Mínimos privilegios: para que una aplicación limite el daño al que puede estar expuesta, se aplica a cada uno de los usuarios unos privilegios restrictivos para evitar cualquier ataque de escalada de privilegios. El *objetivo* al que se quiere llegar es que aquellos usuarios que tengan necesidad de conocer tengan permisos y los que no o no se sepa no se les apliquen permisos.
- Registro de eventos de seguridad: para mejorar la auditoria de la aplicación, es necesario que sea capaz de crear eventos o logs de seguridad. El *objetivo* es poder visualizar en todo momento cómo está actuando la aplicación, ya que ayudaría a seguir el rastro de un posible atacante.
- Seguridad por defecto: el *objetivo* es minimizar la superficie de ataque de la aplicación mediante la habilitación de los servicios y elementos mínimos necesarios.
- Seguridad por oscuridad es un error: el mecanismo de seguridad por oscuridad oculta la información importante de la aplicación de manera difícil de obtener. Esto puede ser un error, ya que, al almacenar la información en una zona oscura a la vista del atacante puede llegar a obtener mayor cantidad de datos sobre la arquitectura y su código. El *objetivo* de este principio es concienciar a no usar la seguridad por oscuridad.
- Separación de código, ejecutables, datos de configuración y programa: consiste en separar los datos que genera el software en el equipo con el fin de minimizar el acceso a toda la aplicación, por ejemplo, separar en diferentes carpetas cada una de las características. El *objetivo* que se consigue es minimizar la posibilidad de que un atacante pueda obtener todos los datos de la aplicación.
- Separación de dominios: como su propio nombre indica, este principio de diseño especifica que se deben aislar los diferentes dominios que tenemos en la red, es decir, que el dominio A no pueda comunicarse con el B. El *objetivo* es conseguir minimizar la probabilidad de acceso a las diferentes ubicaciones en memoria por parte de los atacantes.

- Separación de privilegios: este principio de diseño está estrechamente relacionado con el de *Mínimos privilegios*. Para realizar esta separación, se crean diferentes roles de usuarios, es decir, en función del usuario y del rol al que pertenece puede realizar unas acciones u otras. El *objetivo* de este principio es, al igual que lo dicho antes, efectuar la asignación de roles a distintos usuarios pudiendo ejecutar diferentes tareas en función del rol.
- Simplicidad del diseño: para que un software obtenga un nivel de seguridad más alto, es importante realizar un diseño y una redacción de especificaciones sencillas. El objetivo al que se quiere llegar es, como se comenta, simplificar la complejidad del diseño con el fin de minimizar el número de ataques posibles contra el sistema.

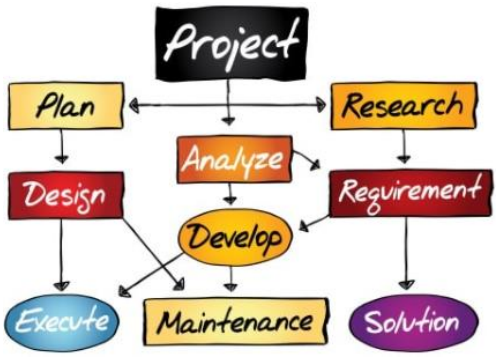
2.2.2. Ciclo de vida del software

Antes de continuar con la seguridad en el software, hay que tener clara la definición y actividades que contempla un ciclo de vida del software.

Según (Becerra & Sanjuan, 2017), el Ciclo de Vida del Software, también conocido como SDLC, es "un marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso".

A partir de esta definición, concluimos que el Ciclo de Vida de Desarrollo Software tiene como objetivo minimizar los costes en errores de implementación durante todo el proceso ya que se revisa durante todo el proceso. (unigoti, 2021)

Figura 2. *Fases del ciclo de vida del desarrollo software*.



Fuente: (unigoti, 2021).

Las actividades del SDLC según (tutorialspoint, 2021) son:

- Comunicación: en esta actividad se realiza la negociación de los requisitos del producto entre el proveedor y el cliente.
- Requisitos del Sistema: el equipo de desarrollo genera los requisitos para conseguir contemplar todo lo dispuesto por el cliente, tanto los requisitos funcionales como los del sistema.
- Estudio de factibilidad: el equipo de desarrollo realiza un estudio a partir de los requisitos creados en la actividad anterior en búsqueda de posibles funciones que se puedan o no realizar. Además, se realiza un análisis financiero y tecnológico.
- Análisis del sistema: el equipo de desarrollo realiza una planificación temporal y de los recursos necesarios para crear el producto.
- Diseño de Software: en esta actividad se realiza un diseño lógico y físico del sistema mediante el uso de diagramas y flujos de datos.
- Codificación: actividad en la que se desarrolla el código fuente del producto, se conoce también como fase de programación.
- Pruebas: se realiza un análisis de las diferentes funcionalidades desarrolladas en la actividad de codificación para detectar posibles errores y solucionarlos cuanto antes.
- Integración: se realiza una integración del producto con las diferentes aplicaciones externas necesarias para su uso (por ejemplo, una base de datos).
- Implementación: se realiza una instalación del producto desarrollado en el cliente.
- Mantenimiento y Funcionamiento: en esta actividad se corrobora si las funcionalidades requeridas funcionan correctamente y se entrega, si se ha solicitado, la diferente documentación de uso del producto.
- Disposición: esta actividad ocurre a lo largo del tiempo del producto estando en cliente, se comentan posibles errores tanto en ejecución como en tecnologías. Aquí es donde ocurriría el posible fin del producto.

Hay diferentes tipos de SDLC, los más conocidos son:

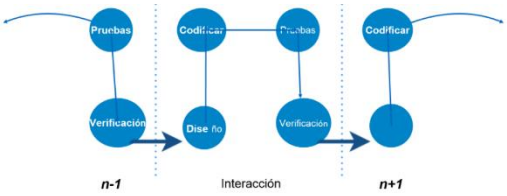
- Modelo en cascada: modelo simple, sigue las fases del SDLC.
- Modelo repetitivo: es cíclico, repite cada etapa después de cada desarrollo.
- Modelo en espiral: se basa en el riesgo durante todo el proceso.
- Modelo Big Bang: requiere mucha programación y financiación, pero, sin embargo, poca planificación.

- Modelo en V: en cada etapa se crean pruebas de validación y verificación del producto, consiguiendo que ambos vayan en paralelo.

Figura 3. *Modelo en cascada.*

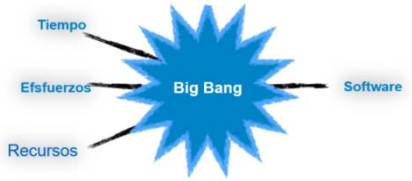
Fuente: (tutorialspoint, 2021).

Figura 4. *Modelo repetitivo.*



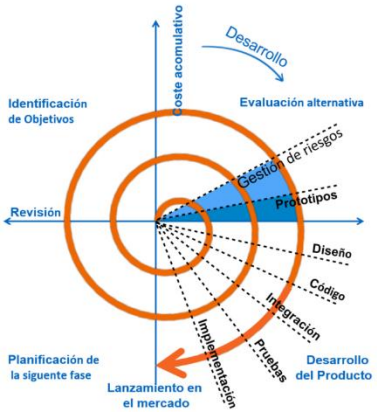
Fuente: (tutorialspoint, 2021).

Figura 5. *Modelo Big Bang.*



Fuente: (tutorialspoint, 2021).

Figura 6. *Modelo en espiral.*



Fuente: (tutorialspoint, 2021).

Figura 7. *Modelo en V.*

Fuente: (tutorialspoint, 2021).

Con el uso de un SDLC, los equipos de desarrollo son capaces de detectar y corregir errores de la codificación antes de llegar a la actividad de implementación. Lo que mejora la calidad y eficiencia del producto con creces. (Bermejo Higuera, 2021)

2.2.3. ¿Qué es un S-SDLC?

Una vez se ha conocido la definición de SDLC y sus diferentes actividades, las propiedades de un software seguro y los principios de diseño de seguridad de software, se puede proceder a focalizar la mirada en el término S-SDLC. Pero, como el propio título pregunta, ¿Qué significa este término?

(de Vicente Mohino, Bermejo Higuera, Bermejo Higuera, & Sicilia Montalvo, 2019) afirma:

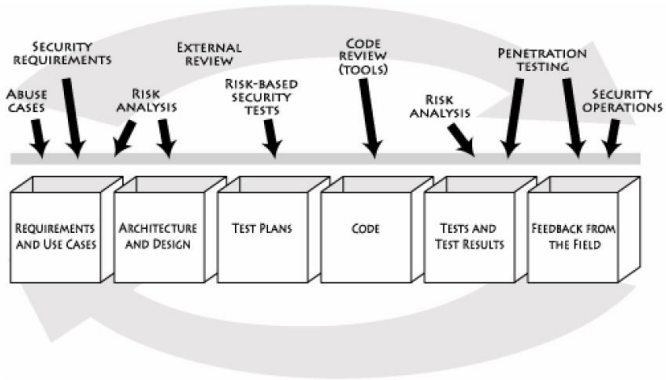
> "El desarrollo de software seguro y confiable requiere la adopción de un proceso sistemático o disciplina que aborde la seguridad en cada una de las fases de su ciclo de vida. Se debe integrar en el mismo dos tipos de actividades de seguridad la primera es el seguimiento de unos principios de diseño seguro y la segunda la inclusión de una serie de buenas prácticas de seguridad. A este nuevo ciclo de vida con prácticas de seguridad incluidas lo llamaremos S-SDLC."

Por tanto, partiendo de esta definición, el S-SDLC es un Ciclo de Vida del Software Seguro cuyas siglas vienen de su nombre en inglés (Secured Software Development Life Cycle). Este modelo estará basado en **Ciclo de vida del software** incorporando los diferentes **Principios de diseño de seguridad del software** con el fin de garantizar las propiedades de un software seguro en el programa en cuestión.

Existen diferentes tipos de S-SDLC, información obtenida de (Hernández Bejarano & Baquero Rey, 2020):

- McGraw's Seven Touchpoints: en esta metodología se plantean un total de siete puntos (auditoría de código, análisis de riesgos arquitectónicos, pruebas de penetración, pruebas de seguridad basadas en el riesgo, construcción de casos de abuso, requisitos de seguridad, operaciones de seguridad y revisiones externas). Este modelo combina aspectos técnicos y prácticos del desarrollo, consiguiendo ser considerado uno de los tres pilares de la seguridad en el software.

Figura 8. *Modelo McGraw.*



Fuente: (McGraw, 2005).

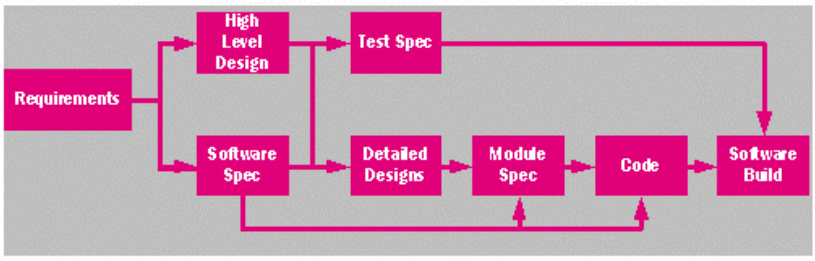
- Microsoft Security Development Lifecycle (SDL): metodología S-SDLC propuesta por Microsoft en 2004, está compuesta por un total de 16 actividades divididas en fase técnica y fase de mejora. Tiene dos versiones, una ágil y otra rígida.

Figura 9. *Modelo SDL.*

Fuente: (Hernández Bejarano & Baquero Rey, 2020).

- Correctness by Construction (CbyC): se basa en que hay que crear el software correctamente desde el inicio mediante la corrección y borrado de errores desde su inicio. Es recomendable su uso en software crítico.

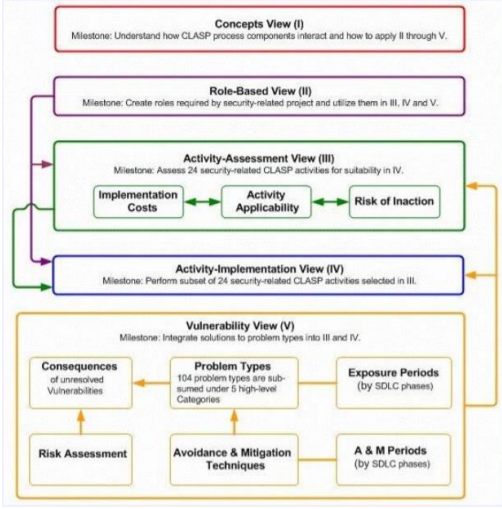
Figura 10. *Modelo CbyC.*



Fuente: (Amey, 2006).

- Comprensive, Lightweight Application Security Process (CLASP): se aplica un proceso de validación y auditoría de código desde el inicio. Se plantea un total de 104 fallos de seguridad y 5 niveles de seguridad, además, tiene 24 acciones de actuación.

Figura 11. *Modelo CLASP.*



Fuente: (Figueroa, 2016)Modelo CLASP.

- Software Assurance Maturity Model (SAMM): es un modelo adaptativo a cada empresa y proyecto. Está basado en 5 bloques: Gobierno, Construcción, Implementación, Verificación y Desarrollo.

Figura 12. *Modelo SAMM.*

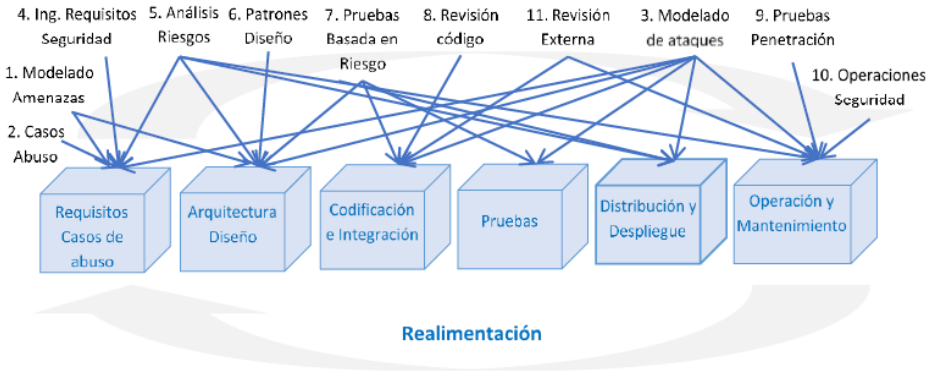
Fuente: (OWASP, 2021) Modelo SAMM.

- Nuevo Ciclo de Vida de Desarrollo Software con metodologías Agiles: es un modelo basado en 23 principios y en los pilares agilidad, seguridad y flexibilidad. Se aplicaría en equipos que trabajen con una metodología Ágil, como Scrum (de Vicente Mohino, Bermejo Higuera, Bermejo Higuera, & Sicilia Montalvo, 2019).

Además de los modelos anteriores, en (Bermejo Higuera, 2021) se define un S-SDLC basado en el descrito por (McGraw, 2005). El cual incluye mejoras en el ciclo de vida seguro del software. Se basa en 6 pilares (requisitos de casos de abuso, arquitectura de diseño, codificación e integración, pruebas, distribución y despliegue, y operación y mantenimiento) y en 11 actividades o mejores prácticas (Modelado de amenazas, Casos de abuso, Modelado de ataques, Ingeniería requisitos de seguridad, Análisis de riesgo arquitectónico, Patrones de

diseño, Pruebas de seguridad basadas en riesgo, auditoría de código, Pruebas de penetración, Configuraciones seguras, Operaciones de seguridad y Revisión externa).

Figura 13. *Modelo en cascada.*



Fuente: (Bermejo Higuera, 2021).

Realizando un análisis de todos, el más completo y el que más se acomoda al desarrollo de la aplicación de escritorio militar es el S-SDLC en cascada estipulado por (Bermejo Higuera, 2021).

Actividades del S-SDLC en cascada

Como se ha explicado anteriormente, el S-SDLC en cascada está basado en el descrito en (McGraw, 2005) y que incluye un total de seis pilares (requisitos de casos de abuso, arquitectura de diseño, codificación e integración, pruebas, distribución y despliegue, y operación y mantenimiento) y once actividades. A continuación, se describen cada una de las actividades (Bermejo Higuera, 2021):

- Modelado de amenazas: ayuda a identificar y planificar de forma correcta la mitigación de las amenazas, así como a evaluar los riesgos inherentes de un software (aplicación de escritorio o web) durante la fase de desarrollo. Es similar a un análisis de riesgos de una organización, aunque tiene un enfoque diferente. (Bermejo Higuera, 2021) y (P. F., 2006)

  Consiste en revisar mediante una metodología la arquitectura o el diseño del software para encontrar y corregir fácilmente los posibles problemas de seguridad. Esta técnica consiste en que los diferentes actores (desarrolladores, pruebas, gerencia, consultores…) participen en todo el proceso de identificación de las posibles amenazas que pueda tener el software, poniéndose en la piel del atacante y tomando una actitud defensiva. Con esto se consigue que se exploren las diferentes debilidades que están presentes o que surgen

durante su desarrollo y se definan las posibles contramedidas necesarias para evitar estas debilidades. (Bermejo Higuera, 2021) y (P. F., 2006)

El modelado de amenazas que se aplique debe ser capaz de (Bermejo Higuera, 2021) y (P. F., 2006):

- Identificar las amenazas potenciales y las condiciones necesarias para que un ataque se lleve a cabo con éxito.
- Facilitar la identificación de las vulnerabilidades ya eliminadas que afecten a alguna amenaza.
- Proporcionar la información sobre las contramedidas para minimizar los ataques o para mitigar los efectos de alguna vulnerabilidad en el software.
- Proveer la información sobre las diferentes medidas que previenen los posibles ataques.
- Transmitir la importancia de los riesgos tecnológicos en función del impacto del negocio a la gerencia.
- Proporcionar una estrategia sólida para evitar brechas de seguridad.
- Facilitar la comunicación y mejorar la concienciación de la importancia de la seguridad.
- Simplificar la actualización de los posibles componentes que se pueden reutilizar.

Existen un gran número de metodologías de modelado de amenazas como CORAS, CIGITAL, OCTAVE, PTA, Trike, PASTA… Nos centraremos en la metodología **Microsoft Threat Analysis and Modeling (TAM)** fue desarrollada por Microsoft y se basa en el uso de árboles de ataques para luego extrapolar las amenazas y realizar una clasificación y un ranking de éstas con el fin de priorizar las actuaciones necesarias para mitigar el riesgo. Es una técnica de ingeniería para ayudar a identificar amenazas, ataques, vulnerabilidades y contramedidas que podrían afectar la aplicación. Puede utilizar el modelado de amenazas para dar forma al diseño de la aplicación, cumplir los objetivos de seguridad de la empresa y reducir el riesgo. (P. F., 2006), (Bermejo Higuera, 2021) y (Microsoft, Threat Modeling, 2021)

TAM tiene un total de cinco fases (P. F., 2006), (Bermejo Higuera, 2021) y (Microsoft, Threat Modeling, 2021):

1) Definición: se definen los objetivos de seguridad, activos, actores y arquitectura de la aplicación.
2) Diagrama: se crea el diagrama a partir de la arquitectura de la aplicación planteada en el paso anterior. Esto se realiza mediante los **diagramas de flujo de datos (DFD)** con el fin de identificar las entidades externas, procesos, almacenamiento de datos, flujos de datos e interfaces.

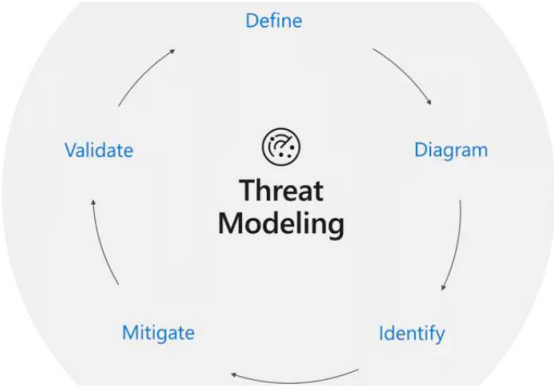


Figura 14. *Fases TAM.*
Fuente: (Microsoft, Threat Modeling, 2021).

3) Identificación: se identifican las amenazas de cada componente del diagrama en función de la arquitectura planteada, se documentan y se valoran las amenazas. La identificación se realiza con el **método STRIDE** (Spoofing [autenticación], Tampering [integridad], Repudiation [no repudio], Information disclosure [confidencialidad], Denial of Service [disponibilidad], Evaluation of privilege [autorización]). Para la valoración se usa el **método DREAD** (Damage, Reproducibility, Explotability, Affected, Disocverability) calculando el riesgo a partir de este método.
4) Mitigación: se da respuesta a las amenazas y se identifican técnicas para la mitigación de los riesgos. Se apoya en lo obtenido en el paso 3 con DREAD, calculando el **Riesgo** a partir de la probabilidad por el impacto potencial.
5) Validación: se rediseñan las amenazas, se usan salvaguardas estándares y personalizadas, y se aceptan los riesgos.

- Casos de abuso: son acciones que pueden suponer una pérdida para la organización o aquellas funciones que no se deben permitir usar en el sistema. Son el símil de lo que puede hacer un atacante en el sistema, por tanto, el mejor análisis de amenazas que se enfrenta el software. Para identificar las áreas de riesgo se deben identificar los objetivos y amenazas de seguridad y puntos del software que son vulnerables, y obtener los requisitos negativos y positivos de seguridad. Los diferentes casos de abuso describen qué no debe hacer el software cuando se hace un uso incorrecto o con mala intención por parte de un atacante. (Bermejo Higuera, 2021)

- Modelado de ataques: son las diferentes acciones que se definen para realizar ataques al software. Se hacen desde el punto de vista del atacante, al igual que los casos de abuso. Para realizar el modelado de ataque se realiza:
    - Patrones de ataque: se captura y representa el conocimiento y perspectiva de un atacante sobre cómo realizar el ataque, los métodos más comunes son los exploits. Se hace uso de la iniciativa **MITRE Common Attack Pattern Enumeration and Classification (CAPEC)** donde se almacenan definiciones comunes, taxonomía de clasificación, esquema de patrones de ataque y ataques reales.
    - Árboles de ataque: es un método sistemático que caracteriza la seguridad del software. Está basado en las dependencias y combinaciones de las vulnerabilidades del software que un atacante puede aprovechar. Analizando las diferentes ramas que puede tener el árbol se pueden identificar las posibles técnicas o métodos que pueden ser usados para comprometer el software. Los árboles de ataque tienen dos tipos de estructuras: Textual (estructura numérica) y Gráfica/Semántica (estructura en modo esquemático).

  Los ataques se pueden combinar y secuenciar mediante el uso de los patrones y árboles de ataque, consiguiendo diseñar respuestas a la mitigación de los ataques en el software. (Bermejo Higuera, 2021)
- Ingeniería de requisitos de seguridad: se basa en conseguir que se definan los aspectos funcionales y no funcionales del software, donde se deben considerar tanto los requisitos técnicos y funcionales como los de seguridad. Los requisitos de seguridad deben ser coherentes, completos, precisos, trazables y verificables. (Bermejo Higuera, 2021)
- Análisis de riesgo arquitectónico: se realiza un análisis de resistencia al ataque (se analiza la resistencia al ataque sobre el modelado de amenazas del diseño de la aplicación), de ambigüedad (se descubren nuevos riesgos en base a los principios de diseño mediante diferentes puntos de vista de diferentes arquitectos del software) y de debilidad (ayuda a comprender el impacto de las dependencias de los COTS). (Bermejo Higuera, 2021)

- Patrones de diseño: son una solución fácil de repetir a un problema recurrente de software. Por medio del uso de los patrones se contribuye a diseñar un software menos vulnerable y más tolerable y resistente a ataques. (Bermejo Higuera, 2021)
- Pruebas de seguridad basadas en riesgo: se identifican los riesgos del software y se diseñan pruebas basadas en los riesgos bajo la perspectiva del atacante. Existen dos aproximaciones de pruebas: Funcionales (se realizan sobre los mecanismos de seguridad para comprobar que las funcionalidades están desarrolladas correctamente) y de Perspectiva del atacante (se simula la actuación de un atacante en el software para comprobar el nivel de seguridad que se tiene). Existen tres tipos de pruebas basadas en riesgo (Bermejo Higuera, 2021):
    - Caja blanca: se debe conocer y entender el código fuente y el diseño del software. Es muy eficaz en busca de posibles errores de codificación. Se suelen apoyar en analizadores de código estático. Sirven para realizar revisiones de diseño, de código o análisis estático de código e inyección de fallos en código fuente.
    - Caja Negra: Se centra en estructuras de datos, APIs, componentes… Se realiza ejecutando el software y realizando sondeos en varias entradas. Se suelen realizar pruebas de penetración, análisis dinámico, análisis de código binario, inyección de fallos en binario, pruebas de fuzzing y escaneo de vulnerabilidades.
    - Caja Gris: es una mezcla de las de caja blanca y negra, donde se tiene el código fuente y la interfaz. Se realiza la prueba de análisis híbrido.
- Auditoría de código: es la actividad más importante entre las diferentes prácticas de seguridad dentro del S-SDLC. Hay dos tipos de errores: simples y carencias de conocimientos del programador. Las herramientas de análisis estático producen falsos positivos (un problema detectado que no lo es realmente) y falsos negativos (problemas que existen, pero la herramienta no lo detecta). Es importante haber realizado un análisis de riesgos arquitectónicos antes de realizar el análisis de código. (Bermejo Higuera, 2021)

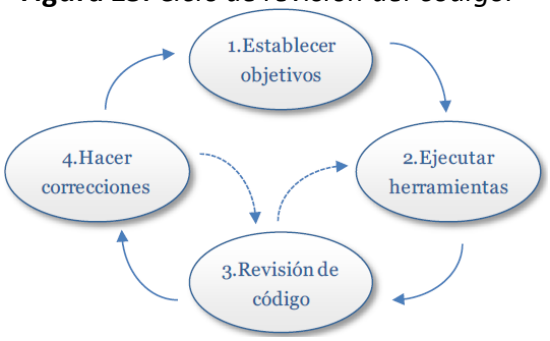
  Las herramientas de análisis de código tienen dos objetivos: descubrir bugs (problemas de menor gravedad) y defectos relevantes (problemas de mucha gravedad), este último es el más importante. Las técnicas a destacar de los analizadores son: analizador

léxico, sintáctico, semántico y estructuras de datos. Además, tienen varios tipos de análisis: estructural, de flujo de control, de flujo de datos, Taint propagation, poiter aliasing, local, global, interpretación abstracta, transformación de predicados, model checking y SAT solvers. (Bermejo Higuera, Apuntes seguridad en el software, 2021)

Para la revisión de código se recomienda seguir el ciclo de revisión, el cuál es:

1) Establecer objetivos
2) Ejecutar herramientas
3) Revisión de código
4) Hacer correcciones

Figura 15. *Ciclo de revisión del código.*



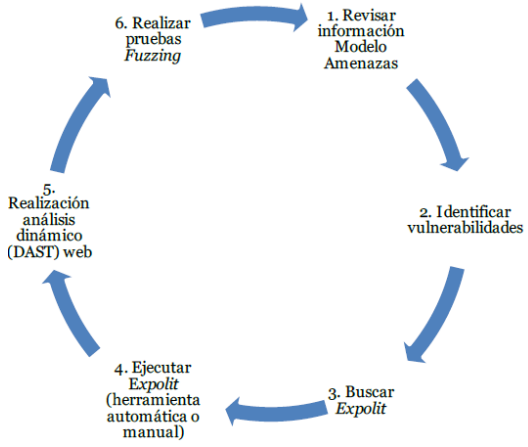
Fuente: (Bermejo Higuera, Apuntes seguridad en el software, 2021).

Las métricas que se obtienen del análisis estático de código son: medición de la densidad de vulnerabilidades, comparación de proyectos por severidad, clasificación de los resultados por categorias, monitorización de tendencias y métricas del proceso de revisión. (Bermejo Higuera, Apuntes seguridad en el software, 2021)

- Pruebas de penetración: se implementa y se comprueba la eficacia de las salvaguardas del software y hardware que se obtuvieron en análisis de riesgos arquitectónicos. Se centran en el comportamiento del software, sus iteraciones y vulnerabilidades que no se han detectado anteriormente. Se realiza un plan de pruebas de penetración donde se incluirán los peores escenarios para realizar intrusiones y ejecutar patrones de ataque. (Bermejo Higuera, Apuntes seguridad en el software, 2021) Las fases de las pruebas de penetración son seis fases:

1) Revisar la información de los casos de abuso, patrones de ataque y modelado de amenazas.
2) Identificar las vulnerabilidades del software mediante la herramienta de escaneo de vulnerabilidades.
3) Encontrar exploits en la web sobre la vulnerabilidad encontrada.

Figura 16. *Fases de las pruebas de penetración.*



Fuente: (Bermejo Higuera, Apuntes seguridad en el software, 2021).

4) Ejecutar el exploit contra la aplicación software (manual o automática).
5) Realizar pruebas DAST en caso de que la aplicación sea de tipo web.
6) Realizar pruebas de fuzzing.

Se prueba la aplicación de forma directa y de manera profunda la seguridad en base a los riesgos de seguridad. Es recomendable realizar las pruebas más de una vez, que haya realimentación entre las pruebas y que se evalúen los COTS. (Bermejo Higuera, Apuntes seguridad en el software, 2021)

El uso correcto de herramientas de pruebas de penetración proporciona la ventaja de resolver la mayoria de las casuisticas a analizar y que los resultados obtenidos al finalizar la prueba se pueden usar como métricas. (Bermejo Higuera, Apuntes seguridad en el software, 2021)

- Operaciones de seguridad: se centra en la distribución (reducir la probabilidad de acceso y manipulación del software durante la entrega al cliente), despliegue (configuración según lo establecido para evitar posibles entornos de despliegue inseguros) y operaciones (registrar eventos, monitorización y backup de los equipos). (Bermejo Higuera, Apuntes seguridad en el software, 2021)
- Revisión externa: es la revisión del ciclo de vida por personal ajeno para detectar posibles fallos no encontrados y tener otra perspectiva de seguridad y del riesgo del software. (Bermejo Higuera, Apuntes seguridad en el software, 2021)

2.3. Descripción de las diferentes herramientas

Una vez conocida la introducción sobre los sistemas de mando y control militares, el S-SDLC elegido a seguir y las actividades que lo conforman, se va a proceder a definir las herramientas que se van a hacer uso durante el Piloto Experimental.

2.3.1. Herramienta para la actividad de Modelado de Amenazas

Para la actividad de Modelado de Amenazas, como se ha explicado en **Actividades del S-SDLC en cascada**, existen varios tipos de modelados de amenazas. Esto conlleva a tener un gran número de herramientas para realizar el modelado en función de la metodología seleccionada.

En este caso, se ha decidido hacer uso de la metodología Microsoft Threat Analysis and Modeling (TAM) de Microsoft, debido a su sencillez y su facilidad de aprendizaje para usarse.

Por ello, como es lógico, se ha usado la herramienta que proporciona Microsoft para aplicar en su modelo, esta herramienta se llama Microsoft Threat Modeling Tool. (Microsoft, Microsoft Threat Modeling Tool, 2017)

(Microsoft, Microsoft Threat Modeling Tool, 2017) dice "La herramienta permite a cualquier persona comunicar el diseño de seguridad de sus sistemas, analizar los diseños de posibles problemas de seguridad con una metodología probada, y sugerir y administrar mitigaciones para problemas de seguridad". Con esta definición se prueba que la herramienta es muy sencilla.

Además, tiene unas funcionalidades destacadas:

- Automation: se proponen instrucciones y comentarios para dibujar el modelo.
- STRIDE: se realiza un análisis guiado en las amenazas y mitigaciones por cada elemento.
- Informes: se generan informes de las actividades de seguridad y de la prueba de la fase de comprobación.
- Metodología única: los usuarios visualizan y comprenden mejor las amenazas.
- Diseñada para desarrolladores y centrada en el software.
- Centrada en el análisis de diseño.

2.3.2. Herramienta para la actividad de Auditoría de Código

Para la actividad de Auditoría de Código, que se ha explicado en **Actividades del S-SDLC en cascada**, existen varios tipos de herramientas de análisis de código, entre ellas se encuentran Klocwork, Coverty, Checkmark, AppScan, SonarQube, Fortify, VisualCodeGreeper…

(Pérez García, 2019) realiza un análisis de diferentes herramientas de análisis de código estático (SAST), se estudian concretamente Fortify, SonarQube y VisualCodeGreeper. Los resultados los saca analizando las métricas Precisión y Recall, lenguajes C++ y Java, y resultados positivos de seguridad por CWE. A partir de estas métricas saca la conclusión que la mejor herramienta es Fortify Static Code Analysis (SCA). (Pérez García, 2019)

Por tanto, a partir de los datos obtenidos de (Pérez García, 2019), se concluye que se va a hacer uso de la herramienta Fortify para la actividad de Auditoría de código.

(Focus, 2021) dice "El análisis de código estático automatizado ayuda a los desarrolladores a eliminar las vulnerabilidades y crear un software seguro". Además, (Focus, 2021) da unas técnicas a realizar: programar de forma segura con SAST integrado, abarcar los lenguajes usados por todos los desarrolladores, iniciar escaneos automáticos rápidos, reparar el código fuente a velocidad de DevOps, seguridad automática en integración continua y escalar el programa a una aplicación segura.

2.3.3. Herramientas para la actividad de Pruebas de Penetración

La actividad de Pruebas de Penetración explicada en **Actividades del S-SDLC en cascada** propone aplicar las pruebas en seis fases. En el caso concreto que se encuentra este Piloto Experimental, se realizarán cinco de las seis fases propuestas, debido a que la aplicación no es de tipo web.

La fase de identificación de vulnerabilidades consiste, como su propio nombre indica, en realizar un escaneo de vulnerabilidades conocidas sobre el equipo donde se ejecuta la aplicación. Para esta fase, se necesita usar una herramienta. Existen muchas herramientas, entre las más conocidas están: Nessus, Nexpose, OpenVas, Nmap…

En base a las diferentes herramientas y sus capacidades, se ha decidido hacer uso de la herramienta Nessus. Ya que los resultados obtenidos en (Chalvatzis, A. Karras, & C. Papademetiou, 2019) concluyen que es la mejor herramienta, además de ser la herramienta más popular.

Las características de Nessus según (Chalvatzis, A. Karras, & C. Papademetiou, 2019) son:

- Tiene una extensa base de datos de vulnerabilidades que se actualiza diariamente.
- Está disponible para Windows, varias distribuciones de Linux y FreeBSD.
- Tiene arquitectura de cliente servidor.
- Es capaz de reconocer un servicio inteligente.
- Es compatible con Common Vulnerabilities and Exposures (CVE).
- Contiene su propio lenguaje de scripting llamado Nessus Attack Scripting Language (NASL).

La fase de pruebas de fuzzing consiste en introducir datos no válidos al software a través del propio entorno. Para ejecutar las pruebas se va a necesitar una herramienta. Existen un sinfín de herramientas como Spike, American Fuzzy Lop, Zzuf, Radamsa, Sulley…

En base a las descripciones de las diferentes herramientas disponibles, se propone hacer uso de la herramienta Spike, ya que es un kit de fuzzer. Este kit proporciona una API que permite crear un fuzzer propio basado en cualquier protocolo de red. Además, la programación de las pruebas es en C. (Bradshaw, 2010)

Según (Bradshaw, 2010), las características de Spike son:

- Tiene cadenas construidas para realizar pruebas de fuzzing que son muy efectivas para producir errores en la aplicación.
- Define el concepto "bloques" que se pueden usar durante la prueba de fuzzing en un protocolo de red con un tamaño exacto en los campos del mensaje.
- Soporta diferentes datos que se usan en protocolos de red.

Además, para las pruebas de fuzzing, como dice (Bradshaw, 2010), se hace uso de la herramienta Wireshark. Esta herramienta es un analizador de protocolos de red para visualizar el tráfico a nivel microscópico de la red (Wireshark, 2021).

2.4. Conclusiones

A lo largo del Contexto y Estado del Arte, se ha conseguido visualizar el estado actual de las aplicaciones de mando y control militar, más en concreto las orientadas a vigilancia.

Se ha puesto en contexto sobre qué es un S-SDLC, partiendo de que es un SDLC y qué es necesario aplicar sobre este para conseguir que sea seguro. Para ello se han explicado los principios de diseño de seguridad. Se han analizado diferentes modelos de S-SDLC y se ha llegado a la conclusión de cuál va a ser el modelo a aplicar en este Piloto Experimental, S-SDLC en cascada de (Bermejo Higuera, 2021). A continuación, se definen las once actividades del S-SDLC y cómo aplicarlas.

Se han decidido las herramientas que van a utilizarse durante este Piloto Experimental, más en concreto, se han seleccionado las herramientas de Microsoft Treath Modeling Tool para la actividad de Modelado de Amenazas; Fortify para la actividad de Auditoría de código; Nessus para la fase de escaneo de vulnerabilidades y Spike y Wireshark para la fase de pruebas de fuzzing en la actividad de pruebas de penetración.

A partir de lo identificado a lo largo del estado del arte, se ha llegado a la conclusión que se necesita aplicar un S-SDLC en el sistema de vigilancia militar. Para ello se realizarán las actividades Modelado de Amenazas, Auditoría de Código y Pruebas de Penetración.

Finalmente, con lo estudiado a lo largo del estado del arte, se llega a la conclusión que para que una aplicación, sea del tipo que sea (aplicación o web) debe ser securizada para mantener toda la información que maneja protegida. Por esta razón que se decide realizar este Piloto Experimental. La aportación que se pretende conseguir es que hay que aplicar buenas prácticas al SDLC.

3. Objetivos concretos y Metodología de Trabajo

3.1. Objetivo general

El objetivo general del presente Piloto Experimental consiste en aplicar buenas prácticas y principios de diseño de seguridad a un Ciclo de Vida de Desarrollo Software (SDLC) de un software dedicado a la vigilancia militar con el fin de mejorar la seguridad en la aplicación.

3.2. Objetivos específicos

Para cumplir el objetivo general planteado anteriormente, se propone cumplir los siguientes objetivos específicos:

- Realización de un Modelado de Amenazas del software con el uso de la herramienta Microsoft Treath Modeling Tool.
- Realización de una Auditoría de código mediante el uso de la herramienta Fortify SCA.
- Realización de Pruebas de penetración, aplicando la herramienta de fuzzing y de escaneo de vulnerabilidades.

3.3. Metodología de trabajo

Para conseguir cumplir el objetivo general y los objetivos específicos descritos en los apartados anteriores (**Objetivo general** y **Objetivos específicos**), se plantea la siguiente metodología de trabajo para la realización del Piloto Experimental. Las fases de la metodología se muestran en la *Figura 17*.

Figura 17. *Fases de la metodología del Piloto Experimental.*

Fuente: Elaboración propia.

1) Estudio del estado del arte: Se realiza una investigación sobre el estado de las aplicaciones de mando y control militares, cómo aplicar un Ciclo de Vida de Desarrollo de Software Seguro y las herramientas que se van a usar en las actividades propuestas.

Esta fase termina con las conclusiones de cuáles son las razones para realizar la contribución de este Piloto.

2) Establecimiento de los objetivos y metodología: En esta fase se establecen los objetivos específicos y general del Piloto y se finaliza con la metodología que se va a realizar durante el Piloto Experimental.

3) Realización de un Modelado de Amenazas: Tras finalizar el establecimiento de los objetivos y metodología, se comienza a realizar el Piloto Experimental con la actividad del S-SDLC llamada Modelado de amenazas con la herramienta Microsoft Threat Modeling Tool.

4) Realización de una Auditoria de Código: En esta fase, se realiza la segunda actividad planteada en el piloto del S-SDLC. Se realiza la Auditoría de Código fuente mediante el uso de la herramienta Fortify.

5) Realización de Pruebas de Penetración: Esta fase es la última de la ejecución de tareas de seguridad sobre el software de mando y control militar. Se realiza el proceso de las pruebas de penetración. En la fase que conforma estas pruebas se usan las herramientas Nessus para el escaneo de vulnerabilidades y Strike para las pruebas de fuzzing.

6) Estudio de los Resultados: Una vez realizadas todas las pruebas, se realiza un estudio de los resultados. Se obtienen los diferentes resultados de las actividades realizadas del S-SDLC y se evalúa el nivel de seguridad.

7) Conclusiones y retrospectiva: En esta última fase se realiza un análisis del Piloto Experimental y de los resultados obtenidos con el fin de llegar a obtener unas lecciones aprendidas y se desarrollan las futuras líneas de trabajo posibles a realizar.

4. Descripción Detallada del Piloto Experimental

A lo largo de este apartado se van a exponer las diferentes actividades realizadas sobre la aplicación a analizar y una descripción sobre la misma.

Se realizará en primer lugar una definición de la aplicación Magec, aplicación de mando y control militar a la que se le va a aplicar un S-SDLC.

A continuación, se realiza un modelado de amenazas de Magec, pasando por la definición de los objetivos de seguridad, activos, actores y arquitectura de la aplicación; realizando un diagrama DFD de la aplicación; identificando las amenazas y posibles patrones de ataque; y, identificando posibles salvaguardas a las amenazas detectadas.

Seguidamente, se realiza una auditoría del código de Magec identificando las diferentes vulnerabilidades en el código y la cantidad de los mismos.

Finalmente, se realizan pruebas de penetración sobre Magec, identificando posibles vulnerabilidades conocidas en el entorno de ejecución de la aplicación y realizando pruebas de fuzzing sobre la aplicación.

4.1. Definición de la Aplicación a Analizar: Magec

El producto que se va a analizar se llama **Magec**, es una aplicación de mando y control con integración y explotación de diferentes sensores, en concreto, este tipo de sistema está englobado dentro de los sistemas del tipo C4ISR en el nombrado militar.

Magec, en concreto, integra y explota cualquier tipo de cámara y cualquier tipo de radar que exista en el mercado.

El producto está dividido en dos ejecutables, uno se corresponde con una aplicación de escritorio que usará el cliente (**Magec Client**) y el otro es un servicio de ejecución en sistemas Windows autónoma (**Magec Service**). Además, existe una web de acceso llamada **Magec Web**.

La aplicación tiene los siguientes requisitos de negocio y técnicos:

- Magec se ejecuta en una red aislada, ya que se usa en un entorno militar.
- Existe dos tipos de usuario que se conectan a la aplicación: clientes y administrador.
- Los clientes deben poder interactuar con los sensores, ya sea para consumir información de ellos, ejecutar órdenes sobre ellos o consumir el estado su estado.

- Los clientes deben poder obtener información de las detecciones sobre posibles objetivos o posibles identificaciones en el mapa.
- Magec Service es el encargado de traducir los datos de los sensores y órdenes ejecutadas por las aplicaciones de Magec a un lenguaje común entre las diferentes aplicaciones.
- Magec Service genera la URL (Magec Web) accesible a través de navegadores web.
- Sólo debe de existir un único Magec Service.
- Hay dos tipos de conexiones por parte de los clientes a la aplicación: conexión mediante Magec Client y mediante Magec Web.
- Se almacenarán las acciones recogidas por los sensores en una base de datos no relacional.
- Se pueden ejecutar en el mismo equipo Magec Service, Magec Client y Magec Web.
- Magec Service envía la dirección multicast de cada stream de video a los clientes (Magec Web y Client).
- Magec Service debe ser capaz de publicar el sitio web.
- El usuario de Magec Client/Web es capaz de activar o desactivar la consumición de la información de los sensores.
- El administrador de Magec Service es el que configura el procesado de la información de los sensores.
- El administrador de Magec Service y la base de datos es el administrador de Windows del equipo.
- El administrador del sensor es quien activa, desactiva o configura el software específico cada sensor.
- Se realizarán copias de seguridad de la base de datos en un servidor ajeno al que esté instalado el Magec Service y la base de datos.
- La tecnología utilizada para las diferentes aplicaciones es .Net usando C#, la base de datos está implementada con MongoDB y la publicación de la web será con IIS.

El sistema está basado en una arquitectura de aplicaciones de escritorio, donde el cliente puede ser una aplicación de escritorio o un navegador que accede a los servicios proporcionados por el servicio de Windows, que contiene una base de datos con los datos almacenados por los sensores. El servicio de Windows es el traductor entre los sensores y los

clientes, traduciendo los datos al lenguaje común a las diferentes aplicaciones involucradas. También es el que publica la página web a la que se conectarán los diferentes clientes. Existirá un administrador que será quien administre los sensores y la base de datos.

Además, cabe destacar que los sensores necesitan ser manejados por unos usuarios físicamente en la ubicación de los mismos, con el fin de activar/desactivar el sensor y de configurarlo.

4.2. Realización De Un Modelado De Amenazas

Partiendo del apartado **Actividades del S-SDLC en cascada** en cascada donde se describen los pasos del Modelado de Amenazas según la metodología Microsoft Threat Analysis and Modeling (TAM), se realiza el modelado de amenazas. Como se observa en la Figura 14, la metodología TAM está dividida en cinco fases: definición, diagrama, identificación, mitigación y validación.

Además, en el apartado **Herramienta para la actividad de Modelado de Amenazas**, se describe que se va a hacer uso de la herramienta de Microsoft llamada Microsoft Threat Modeling Tool.

En los siguientes apartados se irá realizando el modelado de amenazas con la metodología TAM haciendo uso de la herramienta Microsoft Threat Modeling Tool.

Todos los resultados obtenidos con la herramienta Microsoft Threat Modeling Tool y descritos a lo largo de este capítulo están descritos en el **Anexo A**. **Informe de Microsoft Threat Modeling Tool**.

4.2.1. Definición

Esta fase está compuesta a su vez por cuatro fases: objetivos de seguridad, activos, actores y arquitectura de la aplicación.

A) Objetivos de Seguridad

Para que Magec sea más segura, se plantean los siguientes cinco objetivos de seguridad:

- **OB-1.** Mantener confidencialidad, integridad y disponibilidad de la información almacenada y trasmitida.
- **OB-2.** Proporcionar un servicio seguro a los clientes.
- **OB-3.** Proporcionar un servicio ininterrumpido a los clientes.

OB-4. Proporcionar una experiencia de usuario mejorada a los clientes.

OB-5. Se establecerán procesos de autenticación, autorización y auditoría.

OB-6. Cifrar la información transmitida entre las aplicaciones y sensores.

B) Actores

Los requisitos que se establecen para los actores humanos y no humanos son:

- Los clientes deben poder interaccionar con los sensores utilizando Magec Client o Web.
- El administrador debe ser capaz de configurar la MongoDB y configurar en Magec Service el procesado de la información de los sensores.
- El administrador de los sensores debe ser capaz de activar, desactivar y configurar con el software específico cada sensor.
- Periódicamente, la base de datos se debe copiar en un servidor ajeno a donde esté ejecutándose Magec Service y la MongoDB, con el fin de recuperarse en caso de un incidente de seguridad.

A partir de los requisitos anteriores se identifican dos actores humanos involucrados en el sistema: **clientes** y **administrador** en las aplicaciones de Magec y **administrador** de cada uno de los sensores**.** Como actores no humanos se incluye el proceso de almacenamiento del backup de la base de datos, la generación de la web, la detección y la traducción.

Se asigna un identificador único a cada actor. Se utiliza para poder realizar una referencia cruzada de los actores y su nivel de confianza con los puntos de entrada y los activos.

Tabla 1. ***Tabla de Actores.***

ID	Nombre	Descripción
1	Cliente en el sitio Web (Magec Web)	Usuario local que se conecta a Magec por medio de Magec Web.
2	Cliente en la aplicación de escritorio (Magec Client)	Usuario local que se conecta a Magec por medio de Magec Client.
3	Administrador en el servicio (Magec Service)	Usuario administrador local que configura el procesado de la información de los sensores.
4	Administrador del servidor de base de datos	Usuario administrador local que administra la base de datos de los sensores y detecciones.

5	Administrador de cada sensor	Usuario administrador de la aplicación del fabricante que administra la activación/desactivación de los sensores y configuración de los mismos.
6	Proceso de Back-up	Proceso que realiza una copia periódica de la base de datos en una ubicación de un tercero.
7	Proceso de generación de la web	Proceso que realiza la generación de una web para el funcionamiento de Magec Web mediante IIS.
8	Proceso de detección	Proceso que realiza Magec Service a partir de los datos de los sensores para descubrir posibles objetivos.
9	Proceso de traducción bidireccional	Proceso que realiza Magec Service a partir de los datos de los sensores y ordenes ejecutadas por las aplicaciones de Magec a un lenguaje común entre las diferentes aplicaciones.

Fuente: Elaboración propia.

C) Activos

Se identifican los activos del sistema y se describen en la *Tabla 2*:

Tabla 2. *Tabla de Activos.*

Tipo	ID	Nombre	Descripción	Actores
Activos Primarios de Información y Servicios	1	Servicio de comunicación con los sensores	Servicio a disposición de los clientes para la comunicación con los sensores.	3
	2	Datos de los sensores	Todos los datos asociados al uso de los sensores por un cliente.	1, 2, 3, 5
	3	Datos de las IPs	Todos los datos asociados a las IPs de cada uno de los clientes, sensores y el propio servicio para la comunicación entre ellos.	3
	4	Datos de las detecciones	Todos los datos asociados a las detecciones realizadas por los sensores.	8
Activos Secundarios Sistema y aplicación	5	Disponibilidad de los sensores y Magec Service	Los sensores y Magec Service deben estar disponibles en régimen de 24x7 para los clientes.	3, 4, 5
	6	Capacidad realizar peticiones a los sensores	Capacidad de realizar peticiones de a los sensores desde cualquier cliente (Magec Web o Client).	1, 2, 3
	7	Capacidad de recuperación de datos	Capacidad de realizar un back-up de la base de datos en un servidor diferente de la red.	4, 6
	8	Acceso al servidor de base de datos	Acceso al servidor de base de datos para administrarlo y demás usuarios en función de sus permisos.	4

	9	Accesos a los logs del sistema	Capacidad para auditar todos los eventos auditables que ocurrieron dentro del sistema Magec.	3, 4
	10	Publicación de la Web	Capacidad para publicar la web a la que se conectará Magec Service con IIS.	7
	11	Traducción de los datos	Capacidad de traducir los datos de los sensores, tanto los datos proporcionados como las ordenes sobre ellos.	9

Fuente: Elaboración propia.

D) Arquitectura de la aplicación

Para definir la arquitectura de la aplicación hay que saber cómo es la arquitectura y descomponer la aplicación en los componentes.

- **Definir la arquitectura**

Para definir la arquitectura es necesario realizar la matriz de control de acceso a los datos; determinar los componentes, servicios, protocolos y puertos de la aplicación; diseñar la autenticación de las entidades; identificar las tecnologías; y, determinar la topología lógica.

  - ***Matriz de control de acceso a los datos***

Los datos de la aplicación comprenden la información de los sensores (estado, ejecución de órdenes…) y de las IPs (correspondientes a cada uno de los equipos involucrados en el sistema Magec).

La matriz de control de acceso a los datos indica los derechos y privilegios (Create (C), Read (R), Update (U) o Delete (D)) que los actores tendrán sobre los diferentes tipos de datos del sistema.

Tabla 3. ***Tabla de control de acceso a los datos.***

		Usuarios (roles)		
		Administrador de MongoDB y Magec Service	Cliente (Magec Web y Client)	Administrador de los sensores
Datos	Datos de Sensores	R,U	R	C,R,U,D
	Datos de IPs	C,R,U,D	R,U	R
	Datos de las detecciones	C, R, U, D	R	-

Fuente: Elaboración propia.

  - ***Determinar los componentes, servicios, protocolos y puertos de la aplicación***

Los usuarios de Magec Web se conectarán a la web publicada por Magec Service a través del puerto 8080 (HTTP).

Los usuarios de Magec Client se conectarán a Magec Service mediante el rango de puertos 33000-43000 (TCP). Este rango se publica por Magec Service para exponer múltiples funcionalidades y Magec Client los auto-descubre. Esto ocurre solo en la red local, usando mensajes de broadcast.

Magec Service se conectará a la base de datos de MongoDB a través del puerto 27017, es el de por defecto, (TCP).

Dependiendo del sensor:

- El Radar se conectará a Magec Service a través del puerto 22010 (HTTP).
- La cámara se conectará a Magec Service a través del puerto 4002 (TCP).

Además, la cámara se comunicará con cada uno de los clientes (Magec Web y Client) a través del puerto 4321 (TCP Multicast) para el streaming de video.

- ***Diseño de la autenticación de las entidades***

Actualmente, no existe autenticación en el sistema.

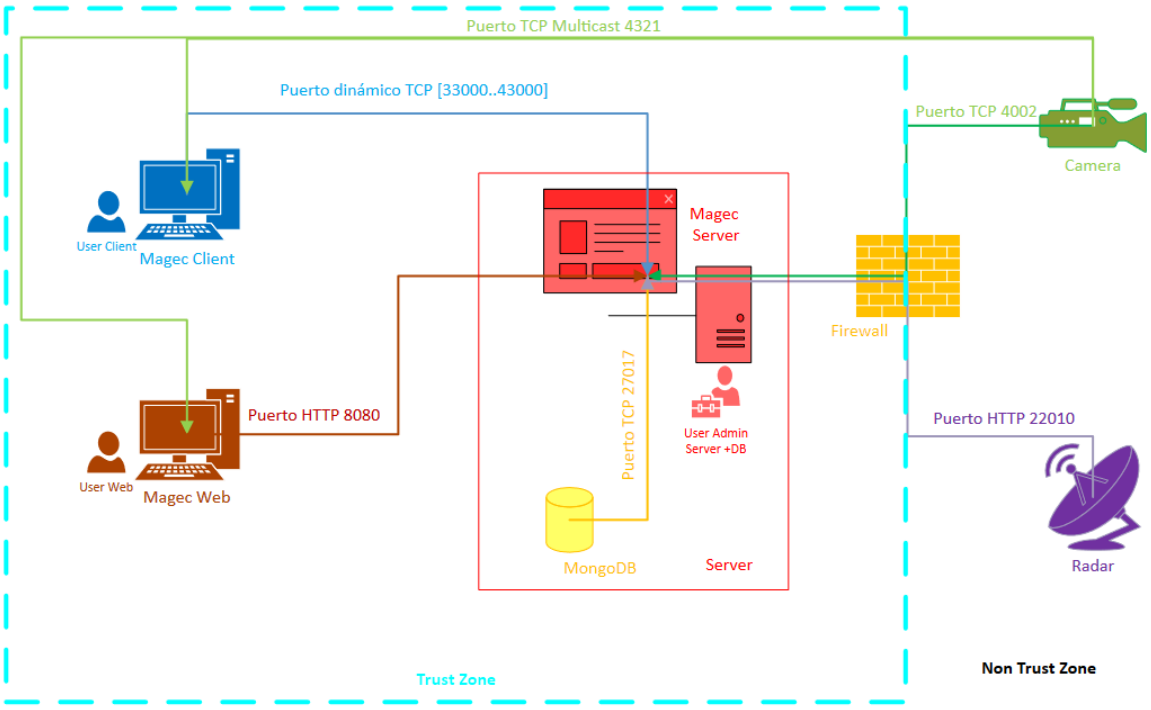
- ***Identificación de tecnologías***

Las diferentes aplicaciones están desarrolladas en .Net usando C # mientras que la base de datos se ha implementado en base al producto MongoDB en su última versión. Con .Net utilizará el framework .Net 4.8. Además, la tecnología de publicación de la web será IIS (Internet Information Services).

- ***Determinar la topología lógica***

La topología lógica de Magec es conforme a la ***Figura 18***.

Figura 18. *Topología lógica de Magec.*



Fuente: Elaboración propia.

- **Descomponer la aplicación**
  - ***Identificar las fronteras de confianza***

Un límite de confianza es el punto en el que cambia el nivel de seguridad. Para Magec existe la siguiente frontera de confianza: entre las aplicaciones de Magec y los sensores.

  - ***Definir los puntos de entrada***

Los puntos de entrada son los elementos que incorporan la entrada del usuario y definen las interfaces a través de las cuales los posibles agentes maliciosos pueden interactuar con la aplicación con el objetivo de introducir datos con carácter malicioso. En Magec, los puntos de entrada son:

Tabla 4. ***Tabla de puntos de entrada.***

Puntos de entrada			
ID	**Nombre**	**Descripción**	**Actores**
1	Magec Client	Cliente de Magec que interactúa por medio de una aplicación de escritorio	2
2	Magec Web	Cliente de Magec que interactúa por medio de un sitio web accesible a través de HTTP.	1
3	Magec Service	Administrador de Magec que configura el procesado de la información de los sensores.	3

4	MongoDB	Administrador de la base de datos que almacena los datos de los sensores y detecciones	4
5	Sensores	Administrador de los sensores de los que se alimenta Magec	5

Fuente: Elaboración propia.

- ***Identificar los puntos de salida***

Los puntos de salida son los elementos que muestran información desde el sistema y los procesos que extraen los datos. Los puntos de salida en Magec son:

Tabla 5. ***Tabla de los puntos de salida.***

Puntos de salida			
ID	**Nombre**	**Descripción**	**Actores**
1	Resultados en Magec Client	Aplicación utilizada para presentar los resultados de los sensores y de las detecciones.	2, 3
2	Resultados en Magec Web	Web utilizada para presentar los resultados de los sensores y de las detecciones.	1, 3
3	Proceso de generación del sitio web	Proceso que realiza la publicación del sitio web de Magec.	7
4	Procesos de copia de seguridad	Proceso que realiza una copia periódica de la base de datos a una ubicación de un tercero.	6
5	Proceso de detección	Proceso que realiza Magec Service a partir de los datos de los sensores para descubrir posibles objetivos.	3, 8

Fuente: Elaboración propia.

- ***Identificar dependencias externas***

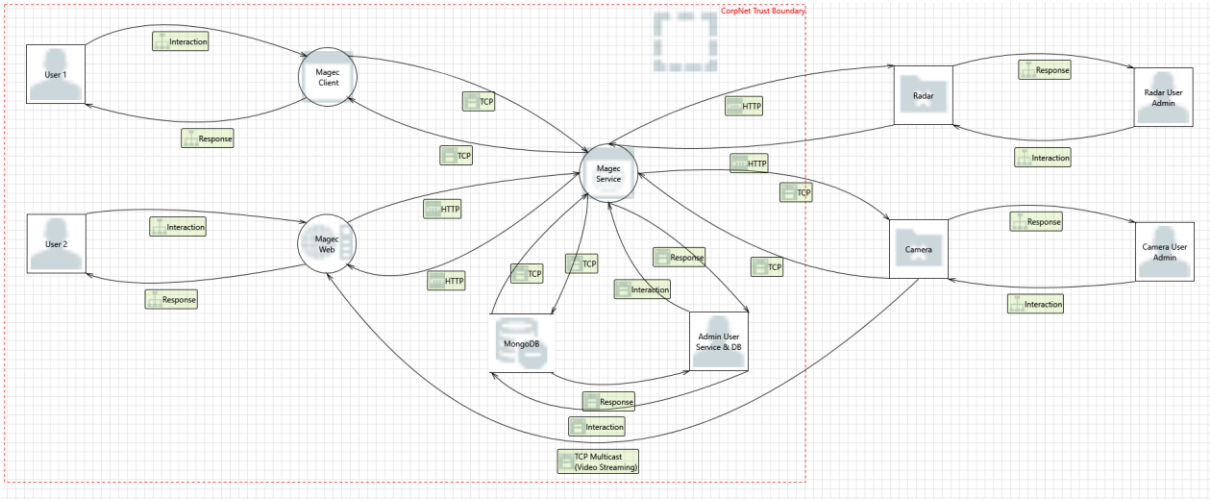
No existen dependencias externas, Magec se ejecuta en una red aislada del entorno militar.

4.2.2. Diagrama

Se crea el diagrama a partir de arquitectura de la aplicación planteada en el paso anterior con el uso de la herramienta Microsoft Threat Modeling Tool, en la vista de diseño.

El diagrama DFD obtenido es:

Figura 19. *Diagrama DFD de Magec.*



Fuente: Elaboración propia.

4.2.3. Identificación

A) Identificación de las Amenazas de cada componente de la Arquitectura

A partir del diagrama DFD creado en la herramienta Microsoft Threat Modeling Tool en la vista de diseño, se pasa a la vista de análisis. La herramienta identifica las amenazas mediante el método STRIDE: Spoofing (autenticación), Tampering (integridad), Repudiation (no repudio), Information disclosure (confidencialidad), Denial of Service (disponibilidad), y Evaluation of privilege (autorización).

Las amenazas obtenidas en la herramienta se mostrarán más documentadas en el apartado siguiente.

B) Documentación de las Amenazas

Gracias a la identificación anteriormente contada por la herramienta Microsoft Threat Modeling Tool, se han detectado un total de 67 amenazas en el sistema completo. Estas 67 amenazas a su vez están divididas en un total de 6 categorías de amenazas.

Se van a documentar las amenazas en grupos cuando sea posible:

Tabla 6. ***Documentación de las Amenazas.***

Descripción de la amenaza	ID 1: Posible consumo excesivo de recursos (ID Threat Tool 4)
Categoría	Denegación de servicio
Objetivo	Controlar el consumo de recursos

Técnicas de ataque	Un adversario hace que el objetivo asigne recursos excesivos para atender la solicitud de los atacantes, reduciendo así los recursos disponibles para servicios legítimos y degradando o negando servicios
Patrón de ataque CAPEC	CAPEC 130: Asignación excesiva
Código CWE (si aplica)	CWE 404: Cierre o liberación de recursos inapropiados
Descripción de la amenaza	ID 2: El flujo de datos se interrumpe potencialmente (IDs Threat Tool 13, 19, 27, 39, y 51)
Categoría	Denegación de servicio
Objetivo	Interrupción de los datos que fluyen a través del límite de confianza
Técnicas de ataque	El atacante subvierte el protocolo con el fin de hacerse pasar por otro, descubrir información confidencial, controlar el resultado de una sesión o realizar otros ataques
Patrón de ataque CAPEC	CAPEC 276: Manipulación de protocolos entre componentes
Código CWE (si aplica)	CWE 707: Neutralización inadecuada
Descripción de la amenaza	ID 3: Posible bloqueo o parada del proceso (IDs Threat Tool 12, 26 y 50)
Categoría	Denegación de servicio
Objetivo	Violación de la propiedad disponibilidad
Técnicas de ataque	El atacante modifica la información del estado del software de destino
Patrón de ataque CAPEC	CAPEC 74: Manipulación del estado
Código CWE (si aplica)	CWE 372: Distinción incompleta del estado interno
Descripción de la amenaza	ID 4: Elevación mediante suplantación de identidad (IDs Threat Tool 1, 14, 15, 16, 28, 29, 30, 32, 34, 36, 41, 44, 52, 53, 54 y 61)
Categoría	Elevación de privilegios
Objetivo	Suplantación, ejecución de código y cambio de flujo de la aplicación
Técnicas de ataque	El atacante explota una debilidad que le permite elevar su privilegio y realizar una acción para la que se supone que no está autorizado a realizar
Patrón de ataque CAPEC	CAPEC 233: Escalada de privilegios
Código CWE (si aplica)	CWE 269: Gestión inadecuada de privilegios
Descripción de la amenaza	ID 5: Control de acceso débil para un recurso (IDs Threat Tool 6 y 66)
Categoría	Divulgación de información
Objetivo	Protección inadecuada de los datos

Técnicas de ataque	El atacante explota la debilidad en la configuración de los controles de acceso y es capaz de eludir la protección prevista contra la que protegen estas medidas y obtiene acceso no autorizado al sistema
Patrón de ataque CAPEC	CAPEC 180: Explotación de niveles de seguridad de control de acceso configurados incorrectamente
Código CWE (si aplica)	CWE 732: Asignación incorrecta de permisos para recursos críticos
Descripción de la amenaza	ID 6: Escaneado del flujo de datos (IDs Threat Tool 11, 25 y 49)
Categoría	Divulgación de información
Objetivo	Flujo de datos sin cifrar
Técnicas de ataque	El atacante escucha pasivamente las comunicaciones de la red y captura el código de la aplicación destinado a un cliente autorizado
Patrón de ataque CAPEC	CAPEC 65: Código de aplicación escuchado
Código CWE (si aplica)	CWE 319: Transmisión de texto sin cifrar de información confidencial
Descripción de la amenaza	ID 7: Suplantación del almacén de datos de destino (IDs Threat Tool 3 y 62)
Categoría	Suplantación
Objetivo	Falta de mecanismo de autenticación para identificar al almacén de datos
Técnicas de ataque	El atacante aprovecha una autenticación incorrecta para proporcionar datos o servicios con una identidad falsificada
Patrón de ataque CAPEC	CAPEC 194: Falsificar la fuente de datos
Código CWE (si aplica)	CWE 287: Autenticación incorrecta
Descripción de la amenaza	ID 8: Suplantación del almacén de datos de origen y de los procesos y sensores (IDs Threat Tool 5, 7, 8, 20, 21, 45, 46 y 64)
Categoría	Suplantación
Objetivo	Falta de mecanismo de autenticación para identificar al almacén de datos y los procesos y sensores
Técnicas de ataque	El atacante engaña a una aplicación y la convence para que solicite un recurso desde una ubicación no deseada
Patrón de ataque CAPEC	CAPEC 154: Suplantación de la ubicación de recursos
Código CWE (si aplica)	--
Descripción de la amenaza	ID 9: Suplantación de entidad de un destino externo (IDs Threat Tool 17 y 37)
Categoría	Suplantación

Objetivo	Falta de mecanismo de autenticación para los sensores
Técnicas de ataque	El atacante modifica el contenido para que contenga algo diferente de lo que pretendía el productor de contenido original, manteniendo inalterada la fuente aparente del contenido
Patrón de ataque CAPEC	CAPEC 148: Suplantación de contenido
Código CWE (si aplica)	CWE 345: Verificación insuficiente de la autenticidad de los datos
Descripción de la amenaza	ID 10: Posible falta de validación de las entradas (IDs Threat Tool 9, 22 y 47)
Categoría	Manipulación
Objetivo	No se realiza verificación de la entrada de datos
Técnicas de ataque	El atacante aprovecha una debilidad en la validación de entrada al controlar el formato, la estructura y la composición de los datos en una interfaz de procesamiento de entrada
Patrón de ataque CAPEC	CAPEC 153: Manipulación de datos de entrada
Código CWE (si aplica)	CWE 20: Validación de entrada incorrecta
Descripción de la amenaza	ID 11: Procesamiento de la notación de objetos de JavaScript (ID Threat Tool 23)
Categoría	Manipulación
Objetivo	Secuestro de JSON
Técnicas de ataque	El atacante roba información posiblemente confidencial transmitida desde el servidor al cliente dentro del objeto JSON aprovechando la laguna en la Política del mismo origen del navegador que no prohíbe que JavaScript de un sitio web se incluya y ejecute en el contexto de otro sitio web
Patrón de ataque CAPEC	CAPEC 111: Secuestro de JSON o de JavaScript
Código CWE (si aplica)	CWE 352: Falsificación de solicitudes entre sitios (CSRF)
Descripción de la amenaza	ID 12: Manipulación de la memoria del proceso (IDs Threat Tool 31, 33 35 y 40)
Categoría	Manipulación
Objetivo	Exceso de acceso a memoria por parte de la aplicación
Técnicas de ataque	El atacante manipula un puntero dentro de una aplicación de destino, lo que hace que la aplicación acceda a una ubicación de memoria no deseada
Patrón de ataque CAPEC	CAPEC 129: Manipulación de puntero
Código CWE (si aplica)	CWE 682: Cálculo incorrecto

Descripción de la amenaza	ID 13: Flujo de datos autentificado comprometido (IDs Threat Tool 0, 2, 42, 43, 55, 56, 57, 58, 59, 60, 63 y 65)
Categoría	Manipulación
Objetivo	Modificación o lectura de los datos transmitidos a través de un flujo de datos autenticado
Técnicas de ataque	El atacante altera u obtiene datos de las transacciones entre dos componentes (normalmente cliente y servidor)
Patrón de ataque CAPEC	CAPEC 94: Adversario en el medio (AiTM)
Código CWE (si aplica)	CWE 290: Omisión de autenticación mediante suplantación
Descripción de la amenaza	ID 14: Repudio potencial de datos (IDs Threat Tool 10, 18, 24, 38 y 48)
Categoría	Repudio
Objetivo	Proporcionar datos de fuera de los límites de confianza
Técnicas de ataque	El atacante crea un enlace simbólico al usuario o a la aplicación objetivo que acceden al punto final del enlace, suponiendo que accede a un archivo concreto.
Patrón de ataque CAPEC	CAPEC 132: Ataque de enlace simbólico
Código CWE (si aplica)	CWE 59: Resolución de enlace incorrecto antes del acceso al archivo

Fuente: Elaboración propia.

C) Valoración de las Amenazas

En la valoración de las amenazas anteriores descritas según los patrones de ataque de CAPEC, se hace uso del **método DREAD** (Damage, Reproducibility, Explotability, Affected, Disocverability) calculando el riesgo a partir de este método.

Para ser más exactos en la valoración con el método DREAD, se usarán las consecuencias que plantean cada uno de los patrones de ataque de CAPEC.

Tabla 7. ***Valoración de las amenazas.***

		Prob. Ocurr. (P)			Impacto Pot. (I)		P	I	Riesgo
ID	*Amenaza*	*R*	*E*	*DI*	*D*	*A*	*(R+E+DI)*	*(D+A)*	*PxI*
1	Posible consumo excesivo de recursos	2	3	3	2	2	8	4	32
2	El flujo de datos se interrumpe potencialmente	1	1	1	3	3	3	6	18
3	Posible bloqueo o parada del proceso	1	2	2	3	3	5	6	30

4	Elevación mediante suplantación de identidad2	2	2	1	2	1	5	3	15
5	Control de acceso débil para un recurso	3	3	3	2	2	9	4	36
6	Escaneado del flujo de datos	1	1	1	2	2	3	4	12
7	Suplantación del almacén de datos de destino	2	2	3	3	3	7	6	42
8	Suplantación del almacén de datos de origen y de los procesos y sensores	3	3	2	3	3	8	6	48
9	Suplantación de entidad de un destino externo	1	1	1	3	3	3	6	18
10	Posible falta de validación de las entradas	2	2	1	3	2	5	5	25
11	Procesamiento de la notación de objetos de JavaScript	2	1	3	3	1	6	4	24
12	Manipulación de la memoria del proceso	2	2	1	2	2	5	4	20
13	Flujo de datos autentificado comprometido	1	1	1	3	3	3	6	18
14	Repudio potencial de los datos	3	2	1	2	3	6	5	30

Fuente: Elaboración propia.

4.2.4. Mitigación

Se apoya en lo obtenido en el apartado anterior, en concreto en la *Tabla 7* con DREAD, calculando el **Riesgo** a partir de la probabilidad por el impacto potencial. Para ello se da respuesta a las amenazas y se identifican técnicas para la mitigación de los riesgos. Para ello, observaremos por cada amenaza las salvaguardas o medidas de mitigación mostradas en la *Tabla 8*.

Tabla 8. ***Medidas de mitigación o salvaguardas a las amenazas.***

ID	*Amenaza*	*Salvaguardas o medidas de mitigación*
1	Posible consumo excesivo de recursos	• Limitar la cantidad de recursos a los que pueden acceder los usuarios sin privilegios • Suponer que todas las entradas son maliciosas y utilizar una configuración que limite los recursos • Limitar todas las solicitudes para que sea más difícil consumir recursos más rápidamente de lo que pueden volver a liberarse

2	El flujo de datos se interrumpe potencialmente	• Gestionar el estado del flujo de datos del software • Gestionar correctamente el estado de los usuarios para no sufrir ataques
3	Posible bloqueo o parada del proceso	• No confiar solamente en ubicaciones controlables por el usuario para mantener el estado del usuario • Evitar la información confidencial en ubicaciones controlables por el usuario • La información sensible que forma parte del estado del usuario debe protegerse adecuadamente para garantizar la confidencialidad e integridad en cada solicitud
4	Elevación mediante suplantación de identidad	• Ejecutar las aplicaciones con menos privilegios • Realizar una autenticación de los usuarios para acceder a la aplicación
5	Control de acceso débil para un recurso	• Configurar el control de acceso correctamente
6	Escaneado del flujo de datos	• Cifrar todas las comunicaciones entre cliente y servidor • Usar SSL, SSH o SCP • Utilizar ipconfig para detectar el sniffer que esté instalado en la red
7	Suplantación del almacén de datos de destino	• Supervisar que el almacenamiento de los datos sea el correcto
8	Suplantación del almacén de datos de origen y de los procesos y sensores	• Supervisar la actividad en la red para detectar cualquier cambio en la comunicación (anómalo o no autorizado)
9	Suplantación de entidad de un destino externo	• Supervisar que los datos suministrados y enviados a los destinos externos se hagan a la identidad correcta
10	Posible falta de validación de las entradas	• Validar las entradas controlando el formato, estructura y composición
11	Procesamiento de la notación de objetos de JavaScript	• Asegurar que el código del servidor sepa diferenciar entre solicitudes legítimas o falsificadas • Dificultar el acceso al contenido del JSON a través de un script en el lado del cliente y que los JSONs no transmitan datos confidenciales • Hacer que las URL del sistema de recuperación de JSON sean impredecibles y únicas por sesión
12	Manipulación de la memoria del proceso	• Evitar el uso de acceso a memoria excesivo • Copiar los datos que se proporcionan y validarlos
13	Flujo de datos autentificado comprometido	• Asegurar que las claves públicas estén firmadas por una autoridad de certificación • Encriptar con criptografía las comunicaciones • Utilizar autenticación mutua fuerte en todos los canales de comunicación y realizar un intercambio de claves públicas en un canal seguro
14	Repudio potencial de los datos	• Utilizar medios de registro de la fuente, hora y un resumen de los datos obtenidos

Fuente: Elaboración propia.

4.3. Realización de una Auditoria de Código

Al igual que en el subapartado anterior, se aplica la actividad de auditoría de código del apartado **Actividades del S-SDLC en cascada** del modelo en cascada. Esta actividad describe en la Figura 15 las cuatro fases del proceso de revisión de código: establecer objetivos, ejecutar la herramienta de análisis de código estático, revisar el código y hacer correcciones.

En el apartado **Herramienta para la actividad de Auditoría de Código** se describe la herramienta de Fortify Static Code Analyzer (SCA).

Todos los resultados obtenidos con la herramienta Fortify SCA y mencionados a lo largo de este capítulo están descritos en el **Anexo B. Informes de Fortify SCA**.

4.3.1. Proceso de análisis de código estático

Durante el proceso de análisis de código estático de código fuente se han establecido los siguientes objetivos:

- Encontrar posibles vulnerabilidades y fallos en seguridad de Magec.
- Mejorar la seguridad de Magec, identificando los verdaderos positivos que provee la herramienta.
- Analizar los errores de programación categorizados como Critical y High, según la prioridad de Fortify.

Seguidamente se ejecutó la herramienta de análisis de código estático Fortify SCA sobre las soluciones involucradas en la aplicación software de C#. Para ello se lanzó una compilación de cada una de las soluciones con Fortify SCA, donde se obtuvieron diferentes errores de programación (bugs) cometidos en fase de desarrollo.

A continuación, se realizó la revisión de código a partir de los bugs mostrados en Fortify SCA, reconociendo los falsos positivos (not an issue), los verdaderos positivos (explotable), los posibles verdaderos positivos (suspicious) y las malas prácticas (bad practice).

Finalmente, se han realizado las correcciones de los falsos positivos, las malas prácticas y los posibles verdaderos positivos sobre las soluciones de código fuente involucradas en Magec.

4.3.2. Resultados de la herramienta

En Magec se ven involucradas 13 soluciones, donde se identifican las diferentes categorías de fortify frente a la prioridad de los errores y al tipo de error:

A) Framework

En esta solución se han detectado un total de 3 errores de programación prioridad Critical y 148 de prioridad High en 41201 líneas de código.

Tabla 9. *Resultados Framework.*

	Critical			High		
	Falso positivo	Verdadero positivo	Mala práctica	Falso positivo	Verdadero positivo	Mala práctica
Dynamic Code Evaluation: Serializable Delegate	1					
Insecure Randomness				27	6	
Missing XML Validation					3	
Null Dereference				47		
Often Misused: Authentication					1	
Password Management: Hardcoded Password					1	
Path Manipulation				6	11	
Path Manipulation: Base Path Overwriting				3		
Portability Flaw: File Separator				1		
Privacy Violation: Heap Inspection					7	
Process Control				1	4	
Unreleased Resource: LDAP				2		
Unreleased Resource: Streams				4		
Unsafe Native Invoke				9		
Weak Encryption: Insecure Mode of Operation		2				
XML External Entity Injection				15		

Fuente: Elaboración propia.

B) HMI

En esta solución se han detectado un total de 8 errores de programación de prioridad High en 3406 líneas de código.

Tabla 10. *Resultados HMI.*

	High		
	Falso positivo	Verdadero positivo	Mala práctica
Null Dereference	5	3	

Fuente: Elaboración propia.

C) MagecService

En esta solución se han detectado un total de 80 errores de programación de prioridad High en 16290 líneas de código.

Tabla 11. ***Resultados MagecService.***

	High		
	Falso positivo	Verdadero positivo	Mala práctica
Insecure Randomness	39	1	
Missing XML Validation		1	
Null Dereference	23		
Path Manipulation	2		
Path Manipulation: Base Path Overwriting	1		
Process Control	2		
Unreleased Resource: Streams		3	
Weak XML Schema: Lax Processing			2
Weak XML Schema: Type Any			1
Weak XML Schema: Unbounded Occurrences			5

Fuente: Elaboración propia.

D) MagecClient

En esta solución se han detectado un total de 2 errores de programación de prioridad Critical y 433 de prioridad High en 120404 líneas de código.

Tabla 12. ***Resultados MagecClient.***

	Critical			High		
	Falso positivo	Verdadero positivo	Mala práctica	Falso positivo	Verdadero positivo	Mala práctica
Command Injection				1		
Dynamic Code Evaluation: Unsafe Deserialization				21		
Insecure Randomness				12		
Missing XML Validation					5	
Null Dereference				110	13	
Password Management: Hardcoded Password		1		2		
Path Manipulation				30		
Path Manipulation: Base Path Overwriting				3		
Portability Flaw: File Separator				30		
Privacy Violation: Heap Inspection				2	1	
Process Control				2		
Unreleased Resource: Streams				3	5	1

Unreleased Resource: Unmanaged Object				8		
Unsafe Native Invoke				1		
Weak Encryption		1				
Weak XML Schema: Lax Processing						5
Weak XML Schema: Type Any						4
Weak XML Schema: Unbounded Occurrences						164
XML External Entity Injection				9		
XML Injection					1	

Fuente: Elaboración propia.

E) Core

En esta solución se han detectado un total de 5 errores de programación de prioridad High en 1128 líneas de código.

Tabla 13. ***Resultados Core.***

	High		
	Falso positivo	Verdadero positivo	Mala práctica
Insecure Randomness		1	
Null Dereference	4		

Fuente: Elaboración propia.

F) InterfazBase

En esta solución se han detectado un total de 7 errores de programación de prioridad High en 2249 líneas de código.

Tabla 14. ***Resultados InterfazBase.***

	High		
	Falso positivo	Verdadero positivo	Mala práctica
Weak XML Schema: Lax Processing			2
Weak XML Schema: Type Any			1
Weak XML Schema: Unbounded Occurrences			4

Fuente: Elaboración propia.

G) InterfazServer

En esta solución se han detectado un total de 6 errores de programación de prioridad High en 2164 líneas de código.

Tabla 15. ***Resultados InterfazServer.***

	High		
	Falso positivo	Verdadero positivo	Mala práctica
Null Dereference	6		

Fuente: Elaboración propia.

H) Sensores

En esta solución se han detectado un total de 84 errores de programación de prioridad High en 21800 líneas de código.

Tabla 16. *Resultados Sensores.*

	High		
	Falso positivo	Verdadero positivo	Mala práctica
Insecure Randomness	19	5	
Null Dereference	38		2
Password Management: Hardcoded Password		2	
Portability Flaw: File Separator	1		
Unreleased Resource		1	
Unreleased Resource: Streams			9
Weak XML Schema: Unbounded Occurrences			6
XML External Entity Injection	1		

Fuente: Elaboración propia.

I) Video

No se ha encontrado ningún error de programación de prioridad Critical o High en las 1762 líneas de código de esta solución.

J) DataBaseServices

En esta solución se han detectado un total de 21 errores de programación de prioridad High en 9658 líneas de código.

Tabla 17. *Resultados DataBaseServices.*

	High		
	Falso positivo	Verdadero positivo	Mala práctica
Weak XML Schema: Unbounded Occurrences			21

Fuente: Elaboración propia.

K) Camaras

En esta solución se han detectado un total de 18 errores de programación de prioridad High en 8925 líneas de código.

Tabla 18. ***Resultados Camaras.***

	High		
	Falso positivo	Verdadero positivo	Mala práctica
Insecure Randomness	14		
Null Dereference			2
Unreleased Resource: Streams			2

Fuente: Elaboración propia.

L) SensoresSeguimiento

En esta solución se han detectado un total de 19 errores de programación de prioridad High en 8647 líneas de código.

Tabla 19. ***Resultados SensoresSeguimiento.***

	High		
	Falso positivo	Verdadero positivo	Mala práctica
Insecure Randomness	3	9	
Null Dereference	6		1

Fuente: Elaboración propia.

M) VideoRecording

En esta solución se han detectado un total de 5 errores de programación de prioridad High en 2218 líneas de código.

Tabla 20. ***Resultados VideoRecording.***

	High		
	Falso positivo	Verdadero positivo	Mala práctica
Insecure Randomness	1	2	
Unreleased Resource: Streams		2	

Fuente: Elaboración propia.

En la *Tabla 21* se resumen las 13 tablas anteriores correspondientes a las soluciones de Magec representadas iguales que las anteriores:

Tabla 21. ***Resultados totales de Magec.***

	Critical			High		
	Falsos Positivos	Verdaderos positivos	Mala práctica	Falsos Positivos	Verdaderos positivos	Mala práctica
Command Injection				1		

Dynamic Code Evaluation: Serializable Delegate	1					
Dynamic Code Evaluation: Unsafe Deserialization				21		
Insecure Randomness				115	24	
Missing XML Validation					9	
Null Dereference				239	16	5
Often Misused: Authentication					1	
Password Management: Hardcoded Password		1		2	3	
Path Manipulation				38	11	
Path Manipulation: Base Path Overwriting				7		
Portability Flaw: File Separator				32		
Privacy Violation: Heap Inspection				2	8	
Process Control				5	4	
Unreleased Resource					1	
Unreleased Resource: LDAP				2		
Unreleased Resource: Streams				7	10	12
Unreleased Resource: Unmanaged Object				8		
Unsafe Native Invoke				10		
Weak Encryption		1				
Weak Encryption: Insecure Mode of Operation		2				
Weak XML Schema: Lax Processing						9
Weak XML Schema: Type Any						6
Weak XML Schema: Unbounded Occurrences						200
XML External Entity Injection				25		
XML Injection					1	

Fuente: Elaboración propia.

En Magec se han detectado un total de 839 errores de programación, de los cuales 5 son críticos.

4.4. Realización De Pruebas De Penetración

En este subapartado, como en los subapartados anteriores, se aplica la actividad de pruebas de penetración del apartado **Actividades del S-SDLC en cascada** del modelo en cascada. Esta actividad está compuesta por seis fases (Figura 16), pero al estar en un entorno sin web sólo aplican las siguientes cinco: revisar información del Modelo de Amenazas, identificar vulnerabilidades, buscar exploits, ejecutar exploits y realizar pruebas de fuzzing.

En el apartado **Herramientas para la actividad de Pruebas de Penetración** se describen las herramientas que se van a usar: Nessus para el escaneo de vulnerabilidades y Spike y Wireshark para las pruebas de fuzzing.

En ambas fases se tiene una distribución formada por tres máquinas. Dos máquinas víctimas con Windows 10 como sistema operativo donde se ejecutará en una Magec Service y en la otra Magec Client. Además, se tiene una máquina Kali Linux, la cual actuará como atacante, que tendrá instaladas las dos herramientas necesarias.

Todos los resultados obtenidos en subapartado **Identificar vulnerabilidades** con la herramienta Nessus están descritos en el informe del **Anexo C. Informe de Nessus** y los del subapartado **Realizar pruebas de fuzzing** en el Anexo **D. Informe de Spike**.

4.4.1. Identificar vulnerabilidades

Durante esta fase de las pruebas de penetración, se busca o se tiene como objetivo analizar los equipos donde se ejecuta Magec en búsqueda de vulnerabilidades conocidas.

Para ello, se ha puesto en ejecución la aplicación Magec y se ha analizado con Nessus las dos máquinas donde se está ejecutando la misma.

Gracias a este análisis con Nessus, se han obtenido los siguientes resultados por cada máquina:

A) Máquina donde se ejecuta Magec Service

En la máquina donde se ejecuta Magec Service, Nessus ha detectado un total de 2 vulnerabilidades conocidas de severidad Medium y 102 de severidad Info.

Tabla 22. ***Vulnerabilidades conocidas en Magec Service.***

	Medium	Info
SSL Certificate Cannot Be Trusted	1	
SMB Signing not required	1	
ICMP Timestamp Request Remote Date Disclosure		1
Common Platform Enumeration (CPE)		1
DCE Services Enumeration		9
Debugging Log Report		1
Device Type		1
Errors in nessusd.dump		1
Ethernet Card Manufacturer Detection		1
Ethernet MAC Addresses		1

HTTP Methods Allowed (per directory)		2
HTTP Server Type and Version		2
HyperText Transfer Protocol (HTTP) Information		2
IP Protocols Scan		1
Link-Local Multicast Name Resolution (LLMNR) Detection		1
Microsoft Windows NTLMSSP Authentication Request Remote Network Name Disclosure		1
Microsoft Windows SMB NativeLanManager Remote System Information Disclosure		1
Microsoft Windows SMB Service Detection		2
Microsoft Windows SMB Versions Supported (remote check)		1
Microsoft Windows SMB2 and SMB3 Dialects Supported (remote check)		1
Nessus Launched Plugin List		1
Nessus Scan Information		1
Nessus Server Detection		12
Nessus TCP scanner		24
Nessus UDP Scanner		13
OS Identification		1
OS Security Patch Assessment Failed		1
Patch Report		1
Ping the remote host		1
SSL / TLS Versions Supported		1
SSL Certificate Information		1
SSL Cipher Suites Supported		1
SSL Perfect Forward Secrecy Cipher Suites Supported		1
Service Detection		3
Strict Transport Security (STS) Detection		1
TLS Version 1.2 Protocol Detection		1
TLS Version 1.3 Protocol Detection		1
Target Credential Status by Authentication Protocol - Failure for Provided Credentials		1
Traceroute Information		1
VMware Virtual Machine Detection		1
WMI Not Available		1
INFO Web Application Sitemap		1
Web Server Directory Enumeration		1
Windows NetBIOS / SMB Remote Host Information Disclosure		1

Fuente: Elaboración propia.

B) Máquina donde se ejecuta Magec Client

En la máquina donde se ejecuta Magec client, Nessus ha detectado un total de 2 vulnerabilidades conocidas de severidad Medium y 74 de severidad Info.

Tabla 23. ***Vulnerabilidades conocidas en Magec Client.***

	Medium	Info
SSL Certificate Cannot Be Trusted	1	

SMB Signing not required	1	
ICMP Timestamp Request Remote Date Disclosure		1
Common Platform Enumeration (CPE)		1
DCE Services Enumeration		9
Debugging Log Report		1
Device Type		1
Errors in nessusd.dump		1
Ethernet Card Manufacturer Detection		1
Ethernet MAC Addresses		1
HTTP Methods Allowed (per directory)		1
HTTP Server Type and Version		1
HyperText Transfer Protocol (HTTP) Information		1
IP Protocols Scan		1
Link-Local Multicast Name Resolution (LLMNR) Detection		1
Microsoft Windows SMB NativeLanManager Remote System Information Disclosure		1
Microsoft Windows SMB Service Detection		2
Microsoft Windows SMB Versions Supported (remote check)		1
Microsoft Windows SMB2 and SMB3 Dialects Supported (remote check)		1
Nessus Launched Plugin List		1
Nessus Scan Information		1
Nessus Server Detection		2
Nessus TCP scanner		13
Nessus UDP Scanner		11
OS Identification		1
OS Security Patch Assessment Failed		1
Patch Report		1
Ping the remote host		1
SSL / TLS Versions Supported		1
SSL Certificate Information		1
SSL Cipher Suites Supported		1
SSL Perfect Forward Secrecy Cipher Suites Supported		1
Service Detection		2
Strict Transport Security (STS) Detection		1
TLS Version 1.2 Protocol Detection		1
TLS Version 1.3 Protocol Detection		1
Target Credential Status by Authentication Protocol - Failure for Provided Credentials		1
Traceroute Information		1
VMware Virtual Machine Detection		1
WMI Not Available		1
Web Application Sitemap		1
Web Server Directory Enumeration		1
Windows NetBIOS / SMB Remote Host Information Disclosure		1

Fuente: Elaboración propia.

En la *Tabla 24* se resumen las dos tablas anteriores correspondientes a las vulnerabilidades encontradas por Nessus en las máquinas donde se ejecuta Magec:

Tabla 24. ***Vulnerabilidades conocidas en las máquinas donde se ejecuta Magec.***

	Medium	Info
SSL Certificate Cannot Be Trusted	2	
SMB Signing not required	2	
ICMP Timestamp Request Remote Date Disclosure		2
Common Platform Enumeration (CPE)		2
DCE Services Enumeration		18
Debugging Log Report		2
Device Type		2
Errors in nessusd.dump		2
Ethernet Card Manufacturer Detection		2
Ethernet MAC Addresses		2
HTTP Methods Allowed (per directory)		3
HTTP Server Type and Version		3
HyperText Transfer Protocol (HTTP) Information		3
IP Protocols Scan		2
Link-Local Multicast Name Resolution (LLMNR) Detection		2
Microsoft Windows NTLMSSP Authentication Request Remote Network Name Disclosure		1
Microsoft Windows SMB NativeLanManager Remote System Information Disclosure		2
Microsoft Windows SMB Service Detection		4
Microsoft Windows SMB Versions Supported (remote check)		2
Microsoft Windows SMB2 and SMB3 Dialects Supported (remote check)		2
Nessus Launched Plugin List		2
Nessus Scan Information		2
Nessus Server Detection		14
Nessus TCP scanner		37
Nessus UDP Scanner		24
OS Identification		2
OS Security Patch Assessment Failed		2
Patch Report		2
Ping the remote host		2
SSL / TLS Versions Supported		2
SSL Certificate Information		2
SSL Cipher Suites Supported		2
SSL Perfect Forward Secrecy Cipher Suites Supported		2
Service Detection		5
Strict Transport Security (STS) Detection		2
TLS Version 1.2 Protocol Detection		2
TLS Version 1.3 Protocol Detection		2

Target Credential Status by Authentication Protocol - Failure for Provided Credentials		2
Traceroute Information		2
VMware Virtual Machine Detection		2
WMI Not Available		2
INFO Web Application Sitemap		2
Web Server Directory Enumeration		2
Windows NetBIOS / SMB Remote Host Information Disclosure		2

Fuente: Elaboración propia.

En las máquinas donde se ejecuta Magec se han detectado un total de 180 vulnerabilidades, de los cuales 4 son high.

4.4.2. Realizar pruebas de fuzzing

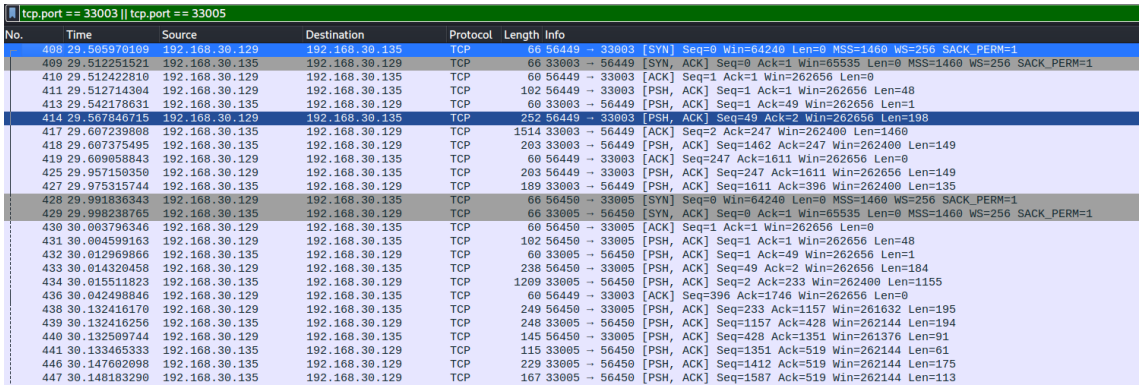
A lo largo de la fase de realizar pruebas de fuzzing se pretende descubrir defectos o vulnerabilidades en el ejecutable de la aplicación que puedan llegar a ser explotadas y convertirse en un día cero (0 Day).

Para entrar en el contexto, se parte de una situación inicial donde la máquina que ejecuta Magec Client tiene la IP 192.168.30.129, la que ejecuta Magec Service 192.168.30.135 y la máquina atacante tiene la 192.168.30.133.

Se comienza ejecutando la aplicación Magec y analizado el tráfico de red con Wireshark. Este tráfico entre Magec Service y Client detectado por la Kali, como es obvio, son comunicaciones entre Service y Client con el protocolo TCP, donde se observan diferentes puertos de comunicación 33003, 33005 y 33009. Analizando las comunicaciones con Wireshark se observan que 33003 y 33005 son puertos de comunicación de la cámara y el 33009 el del radar.

A continuación, se muestra en las Figura 20 y Figura 21 donde se ve el tráfico de los sensores Cámara y Radar respectivamente.

Figura 20. *Tráfico entre Magec Client y Service del Sensor Cámara.*

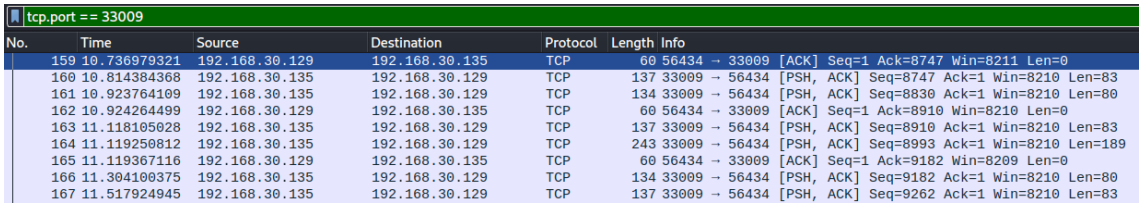


tcp.port == 33003 || tcp.port == 33005

No.	Time	Source	Destination	Protocol	Length	Info
408	29.505970109	192.168.30.129	192.168.30.135	TCP	66	56449 → 33003 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
409	29.512251521	192.168.30.135	192.168.30.129	TCP	66	33003 → 56449 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
410	29.512422810	192.168.30.129	192.168.30.135	TCP	60	56449 → 33003 [ACK] Seq=1 Ack=1 Win=262656 Len=0
411	29.512714304	192.168.30.129	192.168.30.135	TCP	102	56449 → 33003 [PSH, ACK] Seq=1 Ack=1 Win=262656 Len=48
413	29.542178631	192.168.30.135	192.168.30.129	TCP	60	33003 → 56449 [PSH, ACK] Seq=1 Ack=49 Win=262656 Len=1
414	29.567846715	192.168.30.129	192.168.30.135	TCP	252	56449 → 33003 [PSH, ACK] Seq=49 Ack=2 Win=262656 Len=198
417	29.607239808	192.168.30.135	192.168.30.129	TCP	1514	33003 → 56449 [ACK] Seq=2 Ack=247 Win=262400 Len=1460
418	29.607375495	192.168.30.135	192.168.30.129	TCP	203	33003 → 56449 [PSH, ACK] Seq=1462 Ack=247 Win=262400 Len=149
419	29.609058843	192.168.30.129	192.168.30.135	TCP	60	56449 → 33003 [ACK] Seq=247 Ack=1611 Win=262656 Len=0
425	29.957150350	192.168.30.129	192.168.30.135	TCP	203	56449 → 33003 [PSH, ACK] Seq=247 Ack=1611 Win=262656 Len=149
427	29.975315744	192.168.30.135	192.168.30.129	TCP	189	33003 → 56449 [PSH, ACK] Seq=1611 Ack=396 Win=262400 Len=135
428	29.991836343	192.168.30.129	192.168.30.135	TCP	66	56450 → 33005 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
429	29.998238765	192.168.30.135	192.168.30.129	TCP	66	33005 → 56450 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
430	30.003796346	192.168.30.129	192.168.30.135	TCP	60	56450 → 33005 [ACK] Seq=1 Ack=1 Win=262656 Len=0
431	30.004599163	192.168.30.129	192.168.30.135	TCP	102	56450 → 33005 [PSH, ACK] Seq=1 Ack=1 Win=262656 Len=48
432	30.012969866	192.168.30.135	192.168.30.129	TCP	60	33005 → 56450 [PSH, ACK] Seq=1 Ack=49 Win=262656 Len=1
433	30.014320458	192.168.30.129	192.168.30.135	TCP	238	56450 → 33005 [PSH, ACK] Seq=49 Ack=2 Win=262656 Len=184
434	30.015511823	192.168.30.135	192.168.30.129	TCP	1209	33005 → 56450 [PSH, ACK] Seq=2 Ack=233 Win=262400 Len=1155
436	30.042498846	192.168.30.129	192.168.30.135	TCP	60	56449 → 33003 [ACK] Seq=396 Ack=1746 Win=262656 Len=0
438	30.132416170	192.168.30.135	192.168.30.129	TCP	249	56450 → 33005 [PSH, ACK] Seq=233 Ack=1157 Win=261632 Len=195
439	30.132416256	192.168.30.135	192.168.30.129	TCP	248	33005 → 56450 [PSH, ACK] Seq=1157 Ack=428 Win=262144 Len=194
440	30.132509744	192.168.30.129	192.168.30.135	TCP	145	56450 → 33005 [PSH, ACK] Seq=428 Ack=1351 Win=261376 Len=91
441	30.133465333	192.168.30.135	192.168.30.129	TCP	115	33005 → 56450 [PSH, ACK] Seq=1351 Ack=519 Win=262144 Len=61
446	30.147602098	192.168.30.135	192.168.30.129	TCP	229	33005 → 56450 [PSH, ACK] Seq=1412 Ack=519 Win=262144 Len=175
447	30.148183290	192.168.30.135	192.168.30.129	TCP	167	33005 → 56450 [PSH, ACK] Seq=1587 Ack=519 Win=262144 Len=113

Fuente: Elaboración propia.

Figura 21. *Tráfico entre Magec Client y Service del Sensor Radar.*



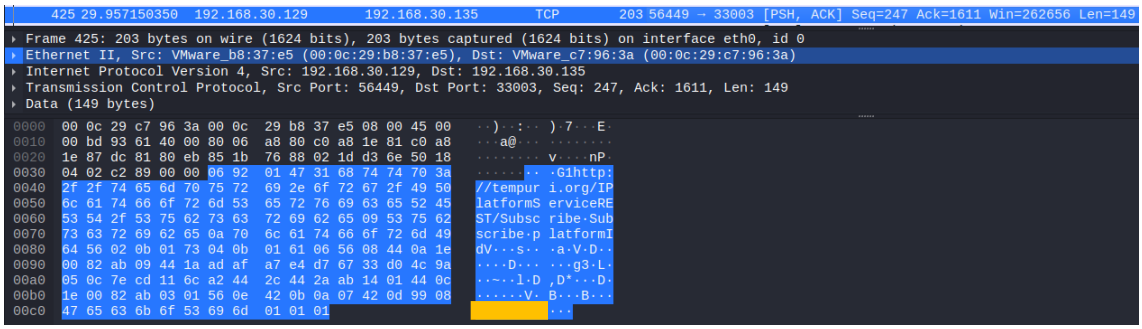
tcp.port == 33009

No.	Time	Source	Destination	Protocol	Length	Info
159	10.736979321	192.168.30.129	192.168.30.135	TCP	60	56434 → 33009 [ACK] Seq=1 Ack=8747 Win=8211 Len=0
160	10.814384368	192.168.30.135	192.168.30.129	TCP	137	33009 → 56434 [PSH, ACK] Seq=8747 Ack=1 Win=8210 Len=83
161	10.923764109	192.168.30.135	192.168.30.129	TCP	134	33009 → 56434 [PSH, ACK] Seq=8830 Ack=1 Win=8210 Len=80
162	10.924264499	192.168.30.129	192.168.30.135	TCP	60	56434 → 33009 [ACK] Seq=1 Ack=8910 Win=8210 Len=0
163	11.118105028	192.168.30.135	192.168.30.129	TCP	137	33009 → 56434 [PSH, ACK] Seq=8910 Ack=1 Win=8210 Len=83
164	11.119250812	192.168.30.135	192.168.30.129	TCP	243	33009 → 56434 [PSH, ACK] Seq=8993 Ack=1 Win=8210 Len=189
165	11.119367116	192.168.30.129	192.168.30.135	TCP	60	56434 → 33009 [ACK] Seq=1 Ack=9182 Win=8209 Len=0
166	11.304100375	192.168.30.135	192.168.30.129	TCP	134	33009 → 56434 [PSH, ACK] Seq=9182 Ack=1 Win=8210 Len=80
167	11.517924945	192.168.30.135	192.168.30.129	TCP	137	33009 → 56434 [PSH, ACK] Seq=9262 Ack=1 Win=8210 Len=83

Fuente: Elaboración propia.

Se pretende aplicar la técnica de fuzzing a los datos de Magec Service, por tanto, se analizan las peticiones de Client a Service para mal formar este tipo de peticiones, nos focalizamos en cámara que es donde se ha detectado más información de las peticiones:

Figura 22. *Suscripción al Sensor Cámara en el puerto 33003.*



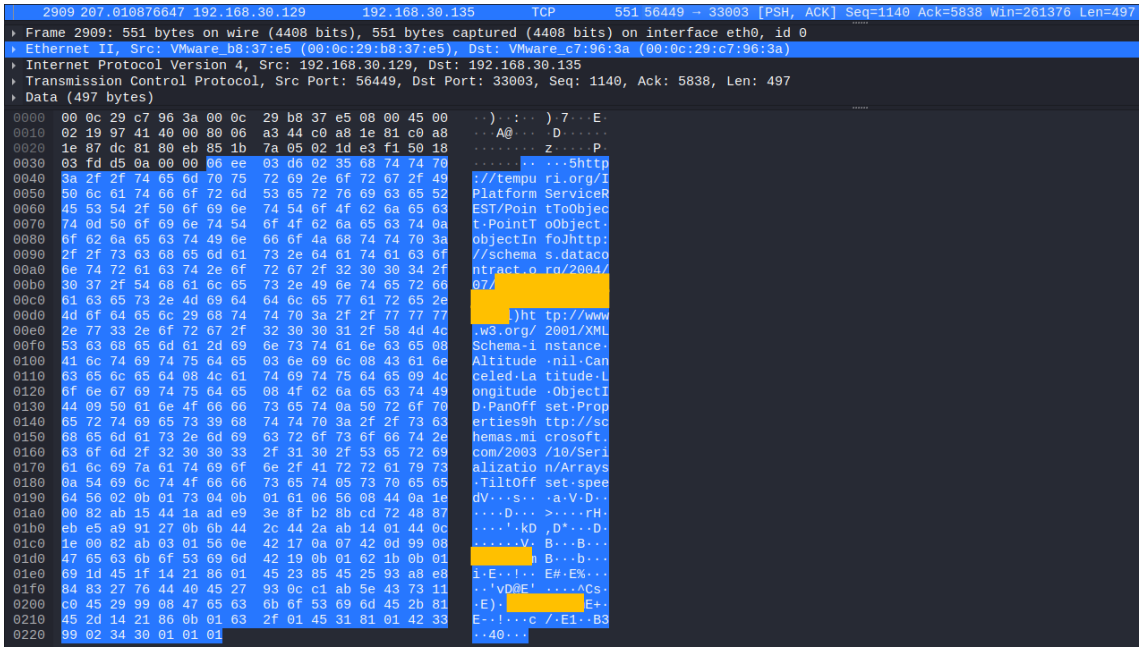
425 29.957150350 192.168.30.129 192.168.30.135 TCP 203 56449 → 33003 [PSH, ACK] Seq=247 Ack=1611 Win=262656 Len=149

▸ Frame 425: 203 bytes on wire (1624 bits), 203 bytes captured (1624 bits) on interface eth0, id 0
▸ Ethernet II, Src: VMware_b8:37:e5 (00:0c:29:b8:37:e5), Dst: VMware_c7:96:3a (00:0c:29:c7:96:3a)
▸ Internet Protocol Version 4, Src: 192.168.30.129, Dst: 192.168.30.135
▸ Transmission Control Protocol, Src Port: 56449, Dst Port: 33003, Seq: 247, Ack: 1611, Len: 149
▸ Data (149 bytes)

```
0000  00 0c 29 c7 96 3a 00 0c  29 b8 37 e5 08 00 45 00   ··)··:·· )·7···E·
0010  00 bd 93 61 40 00 80 06  a8 80 c0 a8 1e 81 c0 a8   ···a@··· ········
0020  1e 87 dc 81 80 eb 85 1b  76 88 02 1d d3 6e 50 18   ········ v····nP·
0030  04 02 c2 89 00 00 06 92  01 47 31 68 74 74 70 3a   ········ ·G1http:
0040  2f 2f 74 65 6d 70 75 72  69 2e 6f 72 67 2f 49 50   //tempur i.org/IP
0050  6c 61 74 66 6f 72 6d 53  65 72 76 69 63 65 52 45   latformS erviceRE
0060  53 54 2f 53 75 62 73 63  72 69 62 65 09 53 75 62   ST/Subsc ribe·Sub
0070  73 63 72 69 62 65 0a 70  6c 61 74 66 6f 72 6d 49   scribe·p latformI
0080  64 56 02 0b 01 73 04 0b  01 61 06 56 08 44 0a 1e   dV···s·· ·a·V·D··
0090  00 82 ab 09 44 1a ad af  a7 e4 d7 67 33 d0 4c 9a   ····D··· ···g3·L·
00a0  05 0c 7e cd 11 6c a2 44  2c 44 2a ab 14 01 44 0c   ··~··l·D ,D*···D·
00b0  1e 00 82 ab 03 01 56 0e  42 0b 0a 07 42 0d 99 08   ·····V· B···B···
00c0  47 65 63 6b 6f 53 69 6d  01 01 01                 Geckoim ···
```

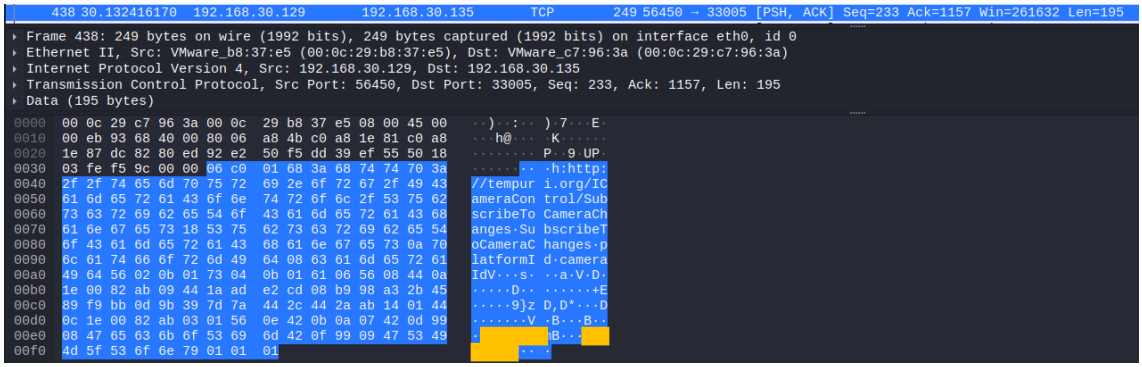
Fuente: Elaboración propia.

Figura 23. *Movimiento del Sensor Cámara en el puerto 33003.*



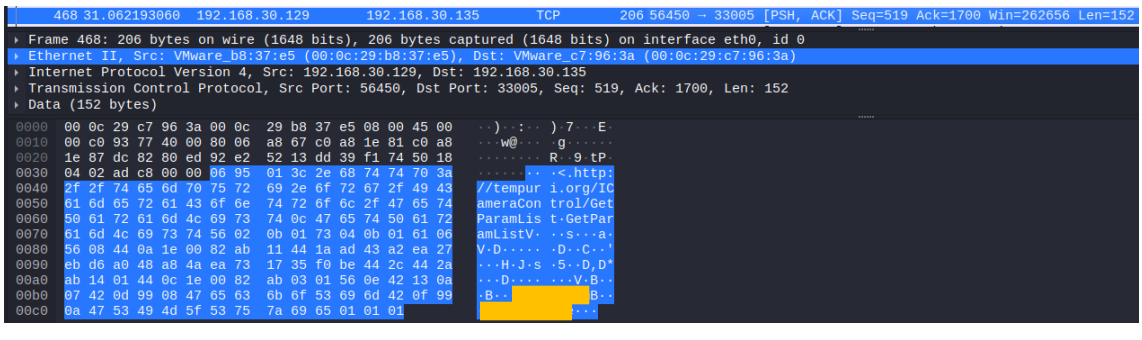
Fuente: Elaboración propia.

Figura 24. *Suscripción al Sensor Cámara en el puerto 33005.*



Fuente: Elaboración propia.

Figura 25. *Obtención de parámetros al Sensor Cámara en el puerto 33005.*



Fuente: Elaboración propia.

Partiendo de las peticiones detectadas con Wireshark mostradas anteriormente en las Figura 22, Figura 23, Figura 24 y Figura 25; se crean los scripts necesarios para mal formar la información:

Figura 26. *Script para la Suscripción al Sensor Cámara en el puerto 33003.*

```
//Corromper una suscripcion a la camara 33003
s_string_variable(".");
s_string_variable(".g");
s_string_variable(".g1");
s_string("http://tempuri.org/IPlatformServiceREST/Subscribe");
s_string_variable(".");
s_string("Subscribe");
s_string_variable(".");
s_string("platformId");
s_string_variable(".");
s_string_variable(".a");
s_string_variable(".ab");
s_string_variable(".abe");
s_string_variable(".abdg");
s_string_variable(".abdgl");
s_string_variable(".abdgls");
s_string_variable(".abdglsv");
s_string_variable(".abdglsv*");
s_string_variable(".abdglsv*,");
s_string("[redacted]");
s_string_variable(".");
```

Fuente: Elaboración propia.

Figura 27. *Script para la Suscripción al Sensor Cámara en el puerto 33005.*

```
//Corromper una suscripcion a una camara 33005
s_string_variable(".");
s_string_variable(".h");
s_string_variable(".h:");
s_string("http://tempuri.otg/ICameraControl/SubscribeToCameraChanges");
s_string_variable(".");
s_string("SubscribeToCametaChanges");
s_string_variable(".");
s_string("platformIdcameraId");
s_string_variable(".");
s_string_variable(".a");
s_string_variable(".ab");
s_string_variable(".abc");
s_string_variable(".abcd");
s_string_variable(".abcdg");
s_string_variable(".abcdgs");
s_string_variable(".abcdgsv");
s_string_variable(".abcdgsv9");
s_string_variable(".abcdgsv9$");
s_string([redacted]);
s_string_variable(".");
s_string_variable(".b");
s_string([redacted]");
```

Fuente: Elaboración propia.

Figura 28. *Script para el Movimiento del Sensor Cámara en el puerto 33003.*

```
//Corromper un movimiento de la camara 33003
s_string_variable(".");
s_string_variable(".5");
s_string("http://tempuri.org/IPlatformServiceREST/PointToObject");
s_string_variable(".");
s_string("PointToObject");
s_string_variable(".");
s_string("objectInfoJ");
s_string("http://schemas.datacontract.org/2004/07/[redacted])");
s_string("http://www.w3.org/2001/XMLSchema-instace");
s_string_variable(".");
s_string("Altitude");
s_string_variable(".");
s_string("nil");
s_string_variable(".");
s_string("Canceled");
s_string_variable(".");
s_string("Latitude");
s_string_variable(".");
s_string("Longitude");
s_string_variable(".");
s_string("ObjectID");
s_string_variable(".");
s_string("ObjectID");
s_string_variable(".");
s_string("PanOffset");
s_string_variable(".");
s_string("Properties9h");
s_string("http://schemas.microsoft.com/2003/10/Serialization/Arrays");
s_string_variable(".");
s_string("TiltOffset");
s_string_variable(".");
s_string("speed");
s_string_variable(".");
s_string_variable(".a");
s_string_variable(".ad");
s_string_variable(".adh");
s_string_variable(".adhk");
s_string_variable(".adhkr");
s_string_variable(".adhkrs");
s_string_variable(".adhkrsv");
s_string_variable(".adhkrsv>");
s_string_variable(".adhkrsv>,");
s_string_variable(".adhkrsv>,*");
s_string([redacted]);
s_string_variable(".");
s_string_variable(".b");
s_string_variable(".bc");
s_string_variable(".bcd");
s_string_variable(".bcde");
s_string_variable(".bcdei");
s_string_variable(".bcdeis");
s_string_variable(".bcdeisv");
s_string_variable(".bcdeisv@");
```

```
s_string_variable(".bcdeisv@#");
s_string_variable(".bcdeisv@#%");
s_string([redacted]);
s_string_variable(".");
s_string_variable(".b");
s_string_variable(".bc");
s_string_variable(".bce");
s_string_variable(".bce0");
s_string_variable(".bce03");
s_string_variable(".bce034");
s_string_variable(".bce034/");
s_string_variable(".bce034/!");
s_string_variable(".bce034/!-");
s_string_variable(".bce034/!-+");
```

Fuente: Elaboración propia.

Figura 29. *Script para la Obtención de parámetros al Sensor Cámara en el puerto 33005.*

```
//Corromper una obtencion de parametros
s_string_variable(".");
s_string_variable(".<");
s_string("http://tempuri.org/ICameraControl/GetParamList");
s_string_variable(".");
s_string_variable(".a");
s_string_variable(".ad");
s_string_variable(".adh");
s_string_variable(".adhk");
s_string_variable(".adhkr");
s_string_variable(".adhkrs");
s_string_variable(".adhkrsv");
s_string_variable(".acdhjsv,");
s_string_variable(".acdhjsv,*");
s_string("[redacted]");
s_string_variable(".");
s_string("[redacted]");
s_string_variable(".");
```

Fuente: Elaboración propia.

Para realizar el ataque se han lanzado en serie cada uno de los scripts como se describe en las siguientes *Figura 30*, Figura 31, Figura 32 y Figura 33:

Figura 30. *Comando para realizar una prueba de fuzzing de la Suscripción al Sensor Cámara en el puerto 33003.*

```
osboxes% /usr/bin/generic_send_tcp 192.168.30.135 33003 /usr/share/spike/camerasuscriptionattack3.spk 0 0
```

Fuente: Elaboración propia.

Figura 31. *Comando para realizar una prueba de fuzzing del Movimiento del Sensor Cámara en el puerto 33003.*

```
osboxes% /usr/bin/generic_send_tcp 192.168.30.135 33003 /usr/share/spike/cameramovementattack3.spk 0 0
```

Fuente: Elaboración propia.

Figura 32. *Comando para realizar una prueba de fuzzing de la Suscripción al Sensor Cámara en el puerto 33005.*

```
osboxes% /usr/bin/generic_send_tcp 192.168.30.135 33005 /usr/share/spike/cameraattack.spk 0 0
```

Fuente: Elaboración propia.

Figura 33. *Comando para realizar una prueba de fuzzing de la Obtención de parámetros al Sensor Cámara en el puerto 33003.*

```
osboxes% /usr/bin/generic_send_tcp 192.168.30.135 33005 /usr/share/spike/cameraobtainparametersattack.spk 0 0
```

Fuente: Elaboración propia.

Aparte de lanzar los comandos anteriores en serie, se lanzaron los cuatro en paralelo.

Durante todas las ejecuciones en serie y paralelo de los comandos, se ha realizado un análisis de los recursos del sistema en el equipo donde se ejecuta Magec Service en busca de detectar algún consumo excesivo de recursos.

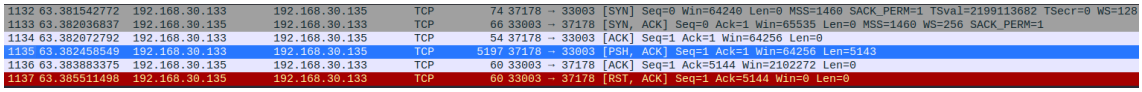
Tabla 25. ***Consumo de recursos en la máquina que ejecuta Magec Service durante las pruebas de fuzzing.***

	CPU	Memoria	Red
Sin clientes	0%	2,3 MB	0 Mbps
Con un Cliente real conectado	0%	7 MB	0,1 Mbps
Con un Cliente real conectado y fuzzeando Figura *30*	28%	13,1 MB	2,5 Mbps
Con un Cliente real conectado y fuzzeando con *Figura 31*	31,2%	15,5 MB	4,1 Mbps
Con un Cliente real conectado y fuzzeando con Figura 32	30,5%	9,8 MB	4,7 Mbps
Con un Cliente real conectado y fuzzeando con Figura 33	30,6%	11,5 MB	4,7 Mbps
Con un Cliente real conectado y fuzzeando con todos en paralelo	51,6%	18 MB	6,6 Mbps

Fuente: Elaboración propia.

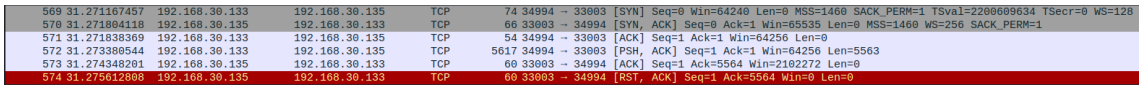
Posteriormente, si se analizan los paquetes recibidos en Wireshark en las pruebas de fuzzing en serie, en concreto se realiza un seguimiento de una trama en cada uno de ellos se observa lo siguiente:

Figura 34. *Seguimiento de una trama de la prueba de fuzzing de la Suscripción al Sensor Cámara en el puerto 33003.*



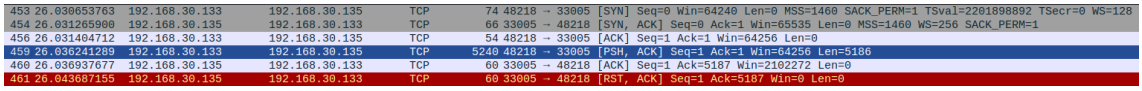
Fuente: Elaboración propia.

Figura 35. *Seguimiento de una trama de la prueba de fuzzing del Movimiento del Sensor Cámara en el puerto 33003.*



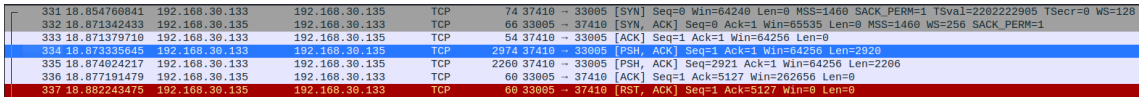
Fuente: Elaboración propia.

Figura 36. *Seguimiento de una trama de la prueba de fuzzing de la Suscripción al Sensor Cámara en el puerto 33005.*



Fuente: Elaboración propia.

Figura 37. *Seguimiento de una trama de la prueba de fuzzing de la Obtención de parámetros al Sensor Cámara en el puerto 33003.*



Fuente: Elaboración propia.

Para finalizar las pruebas de fuzzing, se concluye con que en el cliente real (Magec Client) se ha visto alterada la recepción de los datos por parte de Magec Service del Sensor Cámara y Radar.

5. Estudio de los resultados

En este apartado se estudiarán los resultados obtenidos a lo largo del apartado **Descripción Detallada del Piloto Experimental**

Para ello, se extraen diferentes gráficos con los distintos resultados obtenidos en cada una de las fases del S-SDLC aplicadas sobre Magec.

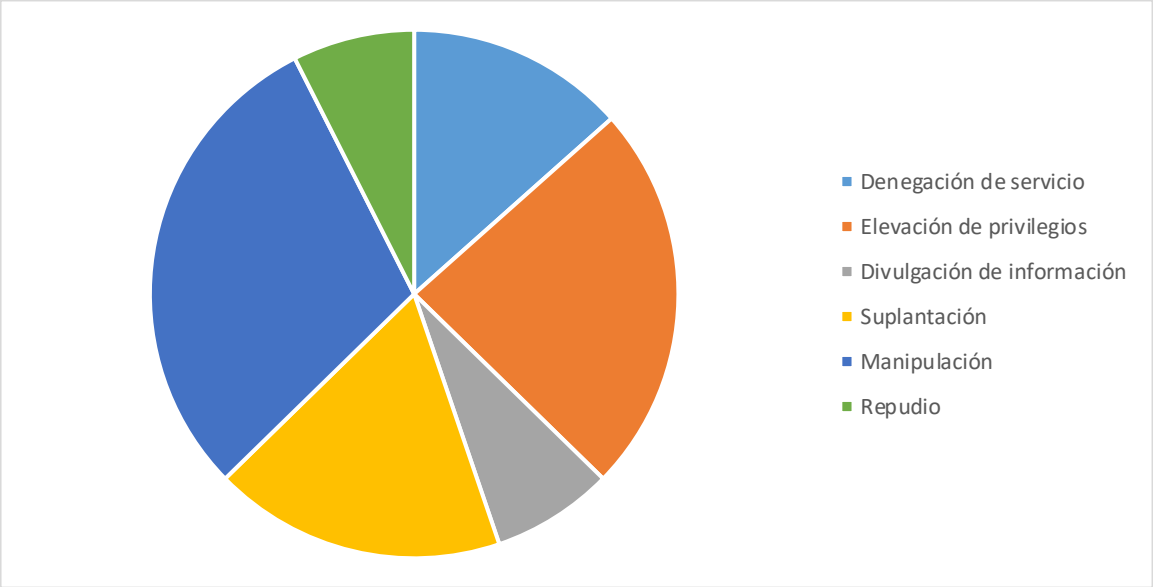
Además, se obtienen ciertas conclusiones sobre los resultados para comprender el estado de seguridad de la aplicación.

5.1. Modelado de Amenazas

A partir de los resultados obtenidos en **Realización De Un Modelado De Amenazas** se obtienen tres tipos de gráficos: amenazas por tipo, por valoración DREAD y por riesgo.

En la Figura 38 se muestra el primer gráfico, donde se observan todas las amenazas detectadas por la herramienta Microsoft Threat Modeling Tool sobre la arquitectura de la aplicación Magec.

Figura 38. *Gráfico Amenazas organizadas por tipo.*



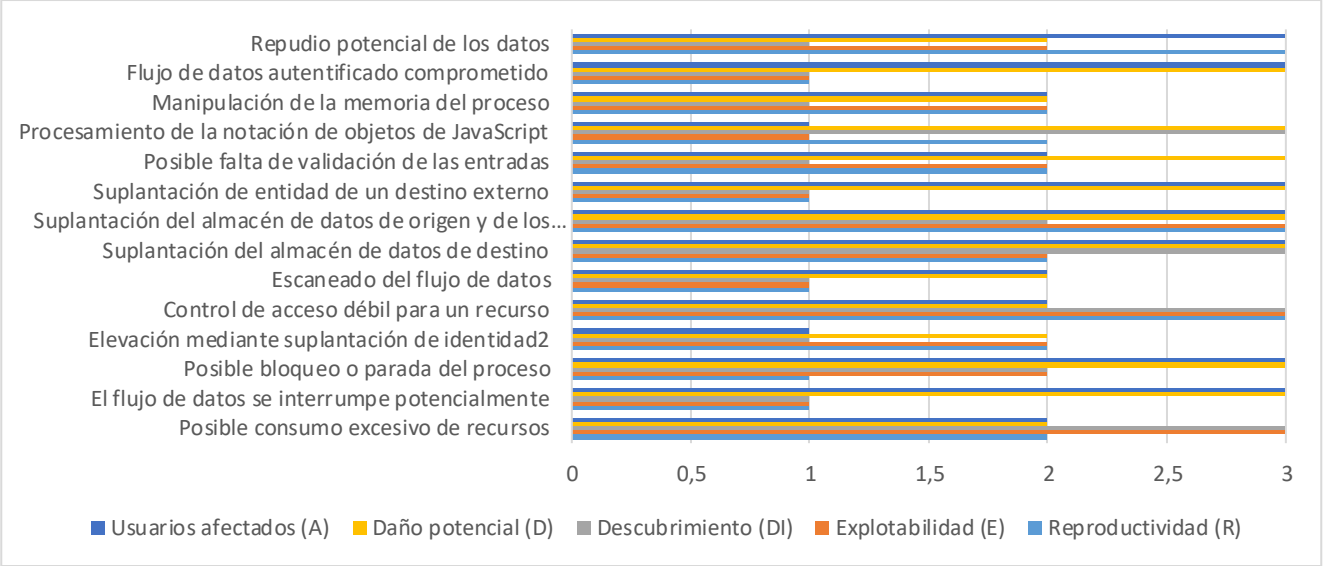
Fuente: Elaboración propia.

En este grafico se puede observar que, entre las 67 amenazas identificadas durante el proceso del modelado de amenazas, la mayor amenaza a la que se expone Magec son las de tipo Manipulación. Las segundas amenazas son las de tipo Elevación de Privilegios y las terceras amenazas son las de tipo Suplantación o Spoofing.

Esto nos hace ver que la arquitectura de Magec debería reforzar la seguridad para evitar principalmente los posibles ataques de manipulación, elevación de privilegios y suplantación.

En la Figura 39 se muestra un gráfico de la valoración asignada a cada una de las amenazas identificadas en durante el modelado de amenazas con DREAD.

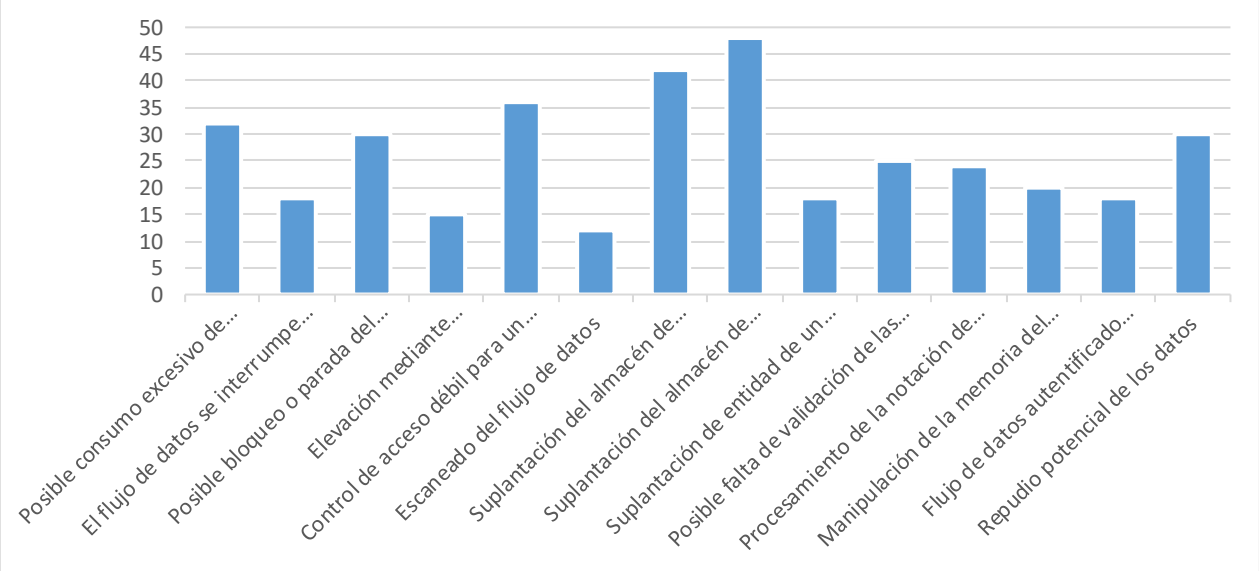
Figura 39. *Gráfico Amenazas organizadas por la valoración DREAD.*



Fuente: Elaboración propia.

En la Figura 40 se visualizan las diferentes amenazas en función del riesgo calculado gracias a la valoración DREAD.

Figura 40. *Gráfico Amenazas en función del riesgo.*



Fuente: Elaboración propia.

Si se analizan las Figura 39 y Figura 40, más focalizando la vista en la segunda, gracias a la valoración DREAD establecida a cada una de las amenazas, se puede observar que las amenazas con mayor riesgo son la de Suplantación del almacén de datos de destino y la de Suplantación del almacén de datos de origen y de los procesos y sensores.

Estas dos amenazas, gracias al modelado de amenazas definido se podrían solventar en Magec con la implantación de las siguientes salvaguardas, por un lado, la realización de una correcta supervisión del almacenamiento de los datos y, por otro, una supervisión de la actividad en la red para detectar cualquier cambio en la comunicación, ya sea anómalo o no autorizado.

Partiendo de las conclusiones obtenidas en las diferentes figuras, se puede concluir que a pesar de que el mayor conjunto de amenazas a las que se expone Magec son los de Manipulación, hay que focalizar la vista en los de tipo Suplantación. Esto se debe a que los mayores riesgos identificados se encuentran organizados en este tipo de amenazas.

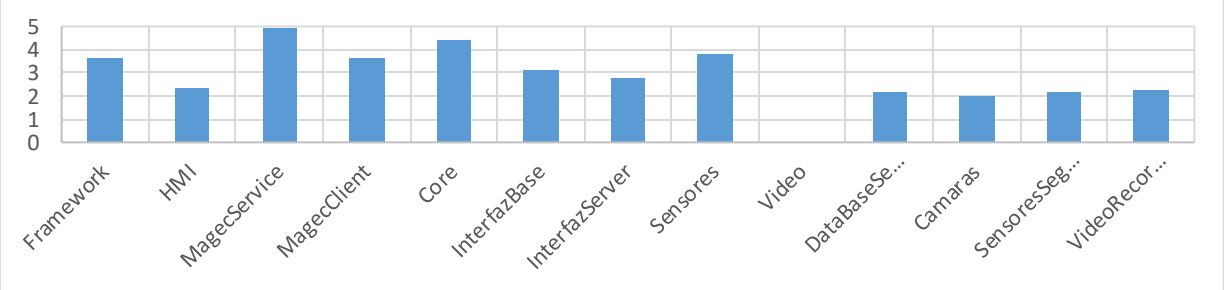
5.2. Auditoría de Código

Gracias a los resultados obtenidos en el apartado de **Resultados de la herramienta**, se pueden obtener diferentes métricas sobre la auditoría de código ejecutada sobre Magec.

Para ver más claro los resultados obtenidos se muestran diferentes gráficos: densidad de vulnerabilidades, comparación de soluciones por severidad de las vulnerabilidades, clasificación de las vulnerabilidades por categorías, soluciones por tipo de error, número de falsos positivos por número total de hallazgos de la herramienta y número de verdaderos positivos por número total de hallazgos de la herramienta.

Se comienza analizando la densidad de vulnerabilidades en porcentaje por cada solución involucrada en Magec, esto se visualiza en la Figura 41. Para ello se ha realizado el cálculo de Vulnerabilidades entre Líneas de código de cada solución.

Figura 41. *Gráfico Densidad de vulnerabilidades.*

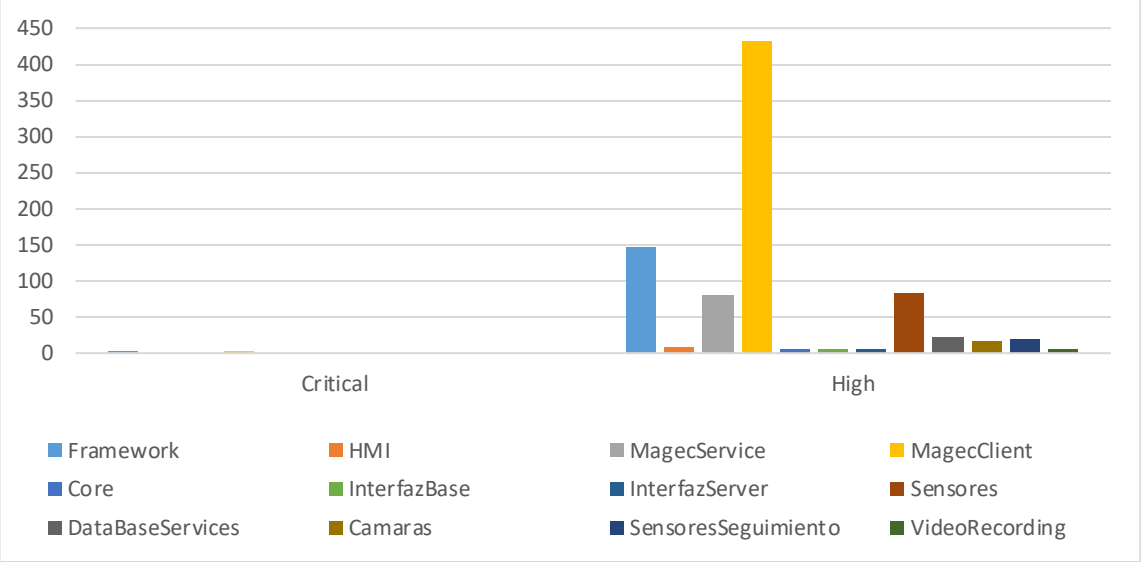


Fuente: Elaboración propia.

Como se puede observar, la densidad de vulnerabilidades en las soluciones MagecService y Core son altas. Lo que lleva a decir que son las soluciones con mayor probabilidad de ser atacadas. Hay que tener en cuenta que la solución de MagecService se corresponde con el servicio de Windows de la arquitectura Magec, la cual es accesible si se accede a ese equipo donde esté instalado.

En la Figura 42 se describen cada una de las soluciones por el nivel de severidad de las vulnerabilidades detectadas por Fortify SCA.

Figura 42. *Gráfico Comparación de proyectos por severidad*

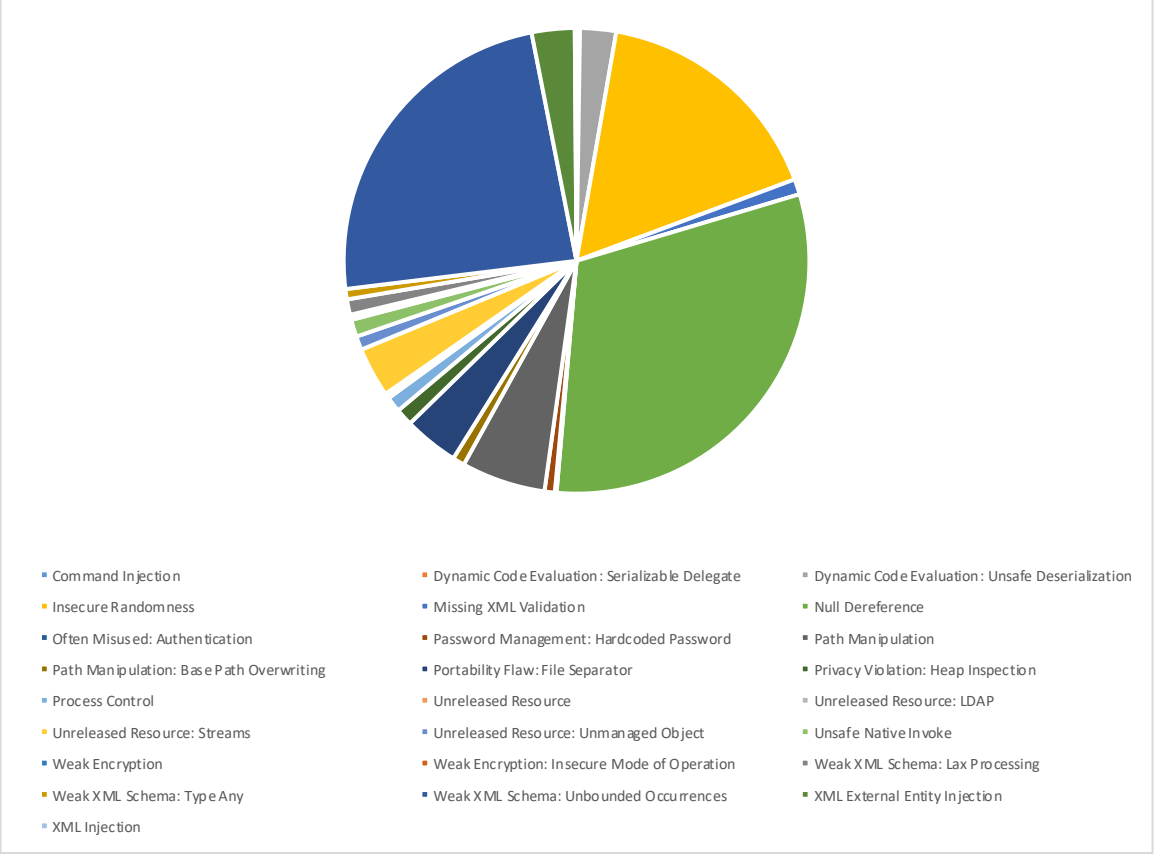


Fuente: Elaboración propia.

Se observa que el nivel de seguridad con mayor cantidad de vulnerabilidades es el High. A pesar de ello, las soluciones Framework y MagecClient tienen vulnerabilidades de nivel de seguridad Critical. Las cuales hay que prestar mucha atención por si fuesen verdaderos positivos, ya que podrían suponer un gran problema para Magec.

A continuación, en la Figura 43 se representan las vulnerabilidades detectadas por Fortify SCA en función de las categorías de cada vulnerabilidad.

Figura 43. *Gráfico Clasificación de las vulnerabilidades por categorías.*



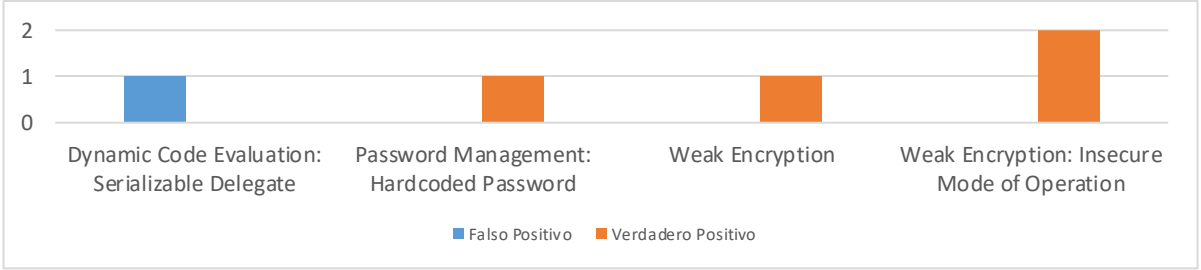
Fuente: Elaboración propia.

A partir de este gráfico, se observa que hay tres categorías predominantes de vulnerabilidades del código de Magec. Estas categorías son Null Dereference, Weak XML Schema: Unbounded Occurrences y Insecure Randomness.

Para asegurar que estas categorías son tan críticas, hay que analizar si realmente son verdaderos positivos o no y si están estas categorías en vulnerabilidades críticas.

Para afinar más los resultados de las Figura 42 y Figura 43, se introduce el siguiente gráfico, Figura 44, que está formado por las vulnerabilidades de nivel de seguridad Critical, además se muestran los tipos de error que son y la categoría a la que pertenecen dichas vulnerabilidades.

Figura 44. *Vulnerabilidades de severidad críticas organizadas por categoría y tipo de error.*



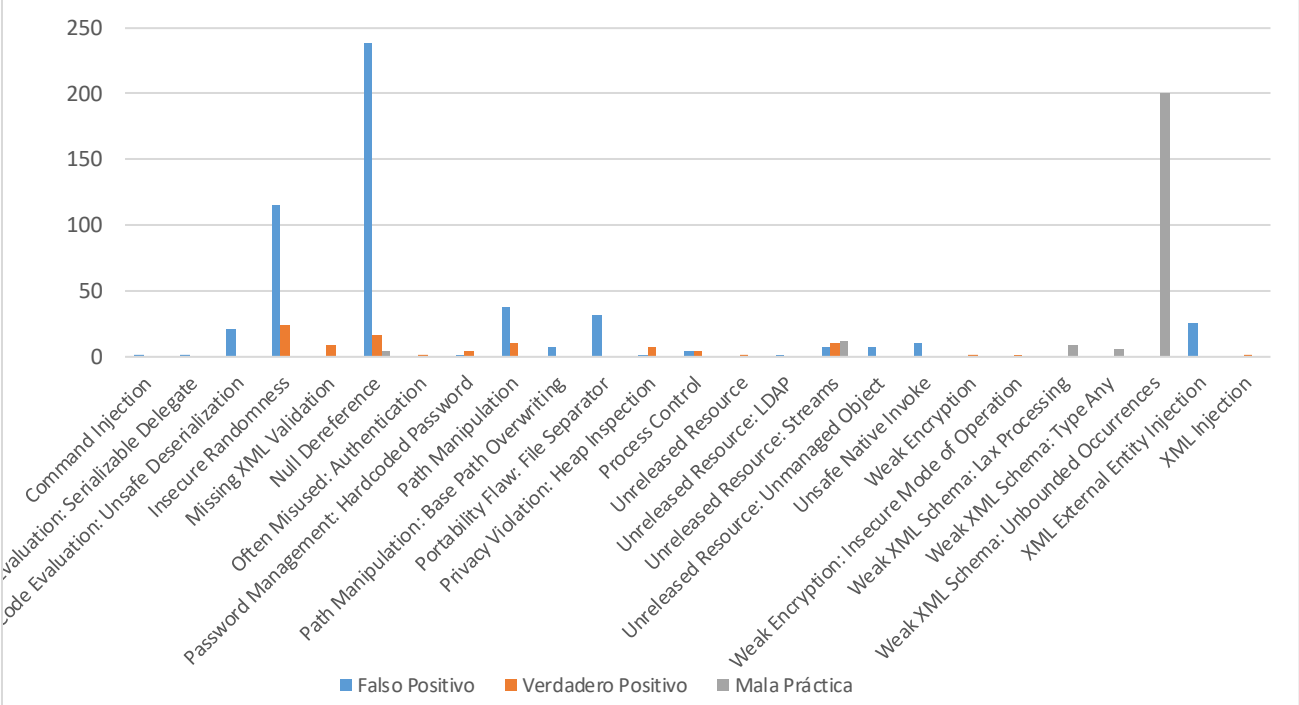
Fuente: Elaboración propia.

Gracias a este gráfico, se puede afirmar que cuatro de las cinco vulnerabilidades de nivel de seguridad Critical son verdaderos positivos. Esto conlleva a que se deben solucionar lo antes posible para evitar posibles catástrofes en la aplicación Magec.

A su vez, las categorías en las que se conforman estas vulnerabilidades de nivel de seguridad Critical no están dentro de las más predominantes descritas en la Figura 43.

Para fortalecer este último argumento, se ha realizado la Figura 45 en la que se han representado los tipos de error detectados por cada categoría.

Figura 45. *Gráfico Tipo de error por las categorías de las vulnerabilidades.*



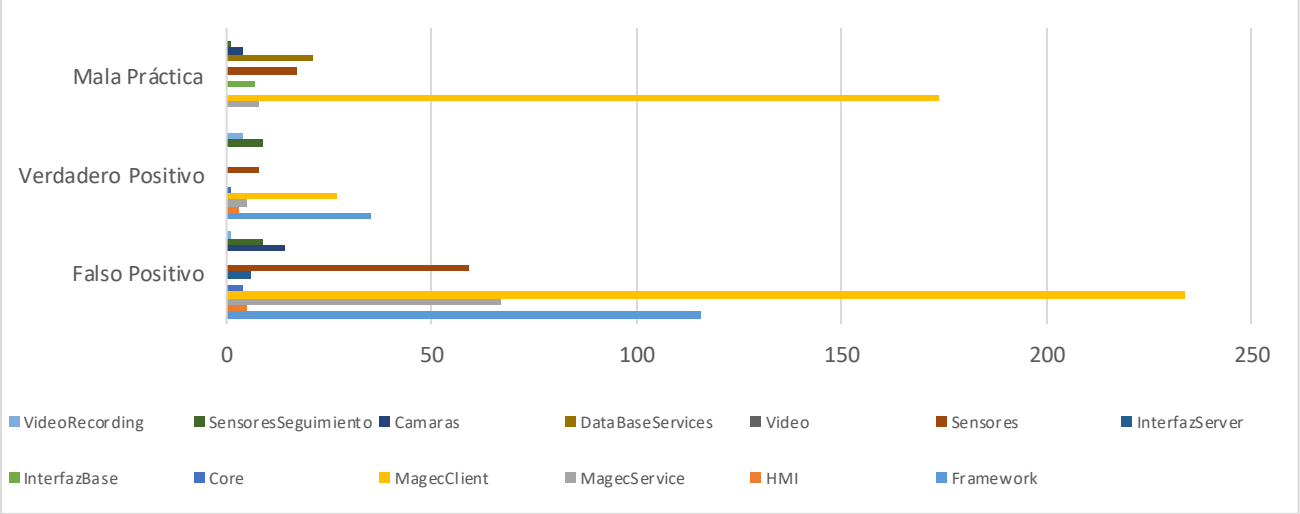
Fuente: Elaboración propia.

En este gráfico se observa claramente que 2 de las 3 peores categorías de vulnerabilidades son Null Dereference e Insecure Randomness, la mayoría de las vulnerabilidades que les conforman son Falsos Positivos, lo cual no son realmente errores.

A la que hay que prestar atención es a la segunda categoría con mayor número de vulnerabilidades, Weak XML Schema: Unboundded Occurrences, que todas ellas son de tipo Mala Práctica, dato revelador a tener en cuenta para corregirlas y evitar posibles ataques en Magec.

Seguidamente, se muestra el gráfico donde se plasman los tipos de error detectados por cada solución de Magec (Figura 46).

Figura 46. *Gráfico Tipo de error por las soluciones de Magec.*

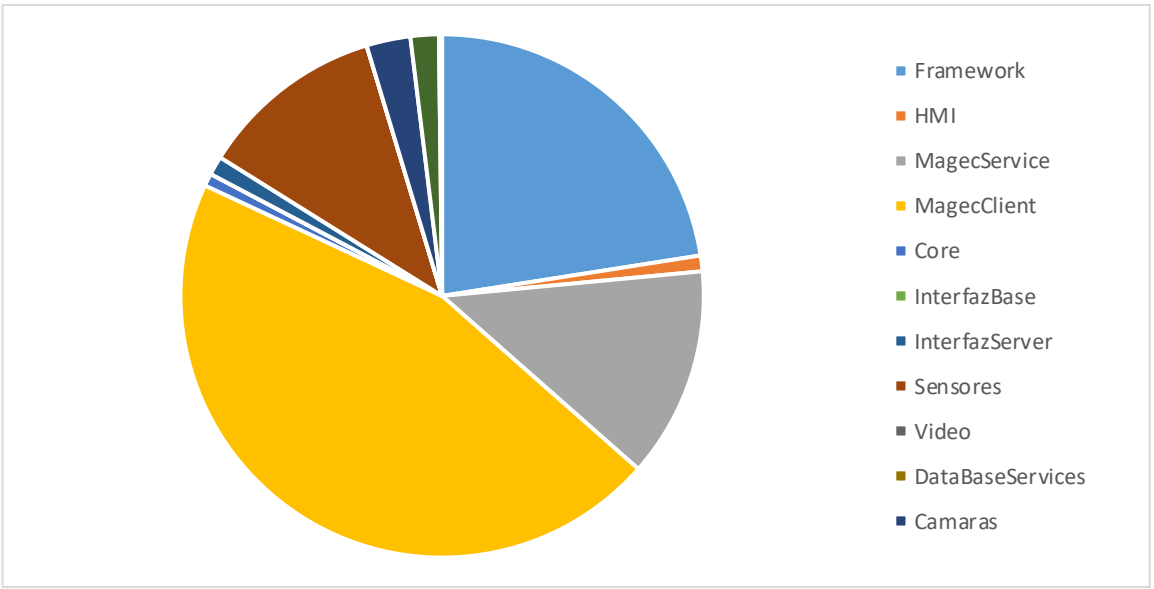


Fuente: Elaboración propia.

Se observa que los verdaderos positivos no son muy altos en los niveles de seguridad Critical y High con valores inferiores a cincuenta vulnerabilidades en las diferentes soluciones de Magec. Esto nos lleva a pensar que el código tiene un nivel de seguridad medio, ya que la comparación con falsos positivos obtenidos es muy alta con respecto a los verdaderos positivos.

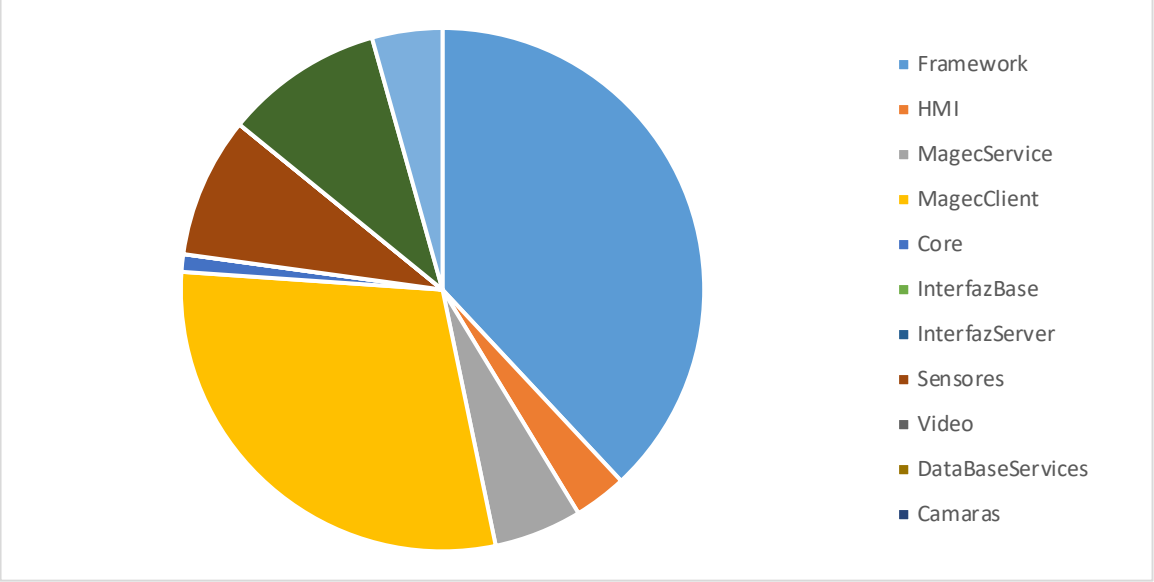
Finalmente, se muestran las Figura 47 y Figura 48 con el número total de vulnerabilidades identificadas como Falso Positivo y Verdadero Positivo en todas las soluciones de Magec, respectivamente.

Figura 47. *Gráfico Número total de Falsos Positivos.*



Fuente: Elaboración propia.

Figura 48. *Gráfico Número total de Verdaderos Positivos*



Fuente: Elaboración propia.

De estos dos últimos gráficos, se observa que tanto los falsos positivos como los verdaderos positivos más altos están en MagecClient y Framework.

Hay que resaltar de forma notable, fortaleciendo el comentario de la Figura 46, el número de falsos positivos es seis veces mayor que el número de verdaderos positivos.

Resumiendo, las conclusiones obtenidas de los resultados analizados es que las soluciones en peor estado son MagecService y Core, a pesar de ser las peores en cuanto a densidad de vulnerabilidades, no contienen vulnerabilidades de nivel de seguridad Critical.

También, las categorías en peores condiciones son Null Dereference, Weak XML Schema: Unbounded Occurrences y Insecure Randomness, aunque en la primera y última la mayoría de las vulnerabilidades son falsos positivos.

Las vulnerabilidades con nivel de seguridad Critical se encuentran en MagecClient y Framework, donde cuatro de cinco son verdaderos positivos.

Finalmente, se ha descubierto que el número de falsos positivos es seis veces mayor que el de verdaderos positivos.

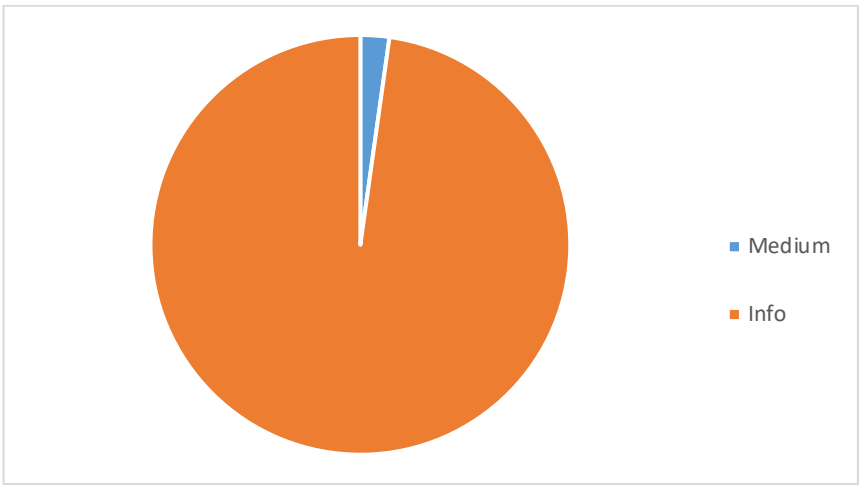
5.3. Pruebas de Penetración

Partiendo de los datos obtenidos en **Identificar vulnerabilidades** sobre las vulnerabilidades conocidas obtenidas con Nessus, se obtienen tres gráficos: número total de vulnerabilidades

en función de la severidad, número total de vulnerabilidades por severidad Medium en función del tipo y número total de vulnerabilidades en función del tipo.

En la Figura 49 se muestra el primer gráfico, donde se observan todas las vulnerabilidades conocidas detectadas por la herramienta Nessus sobre las máquinas donde se ejecuta la aplicación Magec.

Figura 49. *Gráfico total de vulnerabilidades por severidad.*

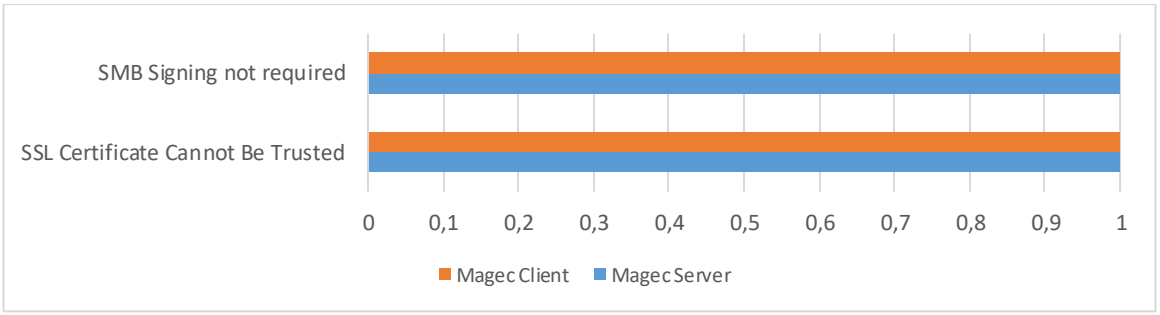


Fuente: Elaboración propia.

En este gráfico se puede observar que sólo 4 de las 180 vulnerabilidades detectadas por Nessus son las que hay que prestar atención. Las demás vulnerabilidades son de severidad informativa, por lo que no son de criticidad alta de revisión.

Para focalizar la atención en las vulnerabilidades detectadas por Nessus de severidad medium, se representan en la Figura 50.

Figura 50. *Gráfico total de vulnerabilidades por severidad Medium y categoría.*



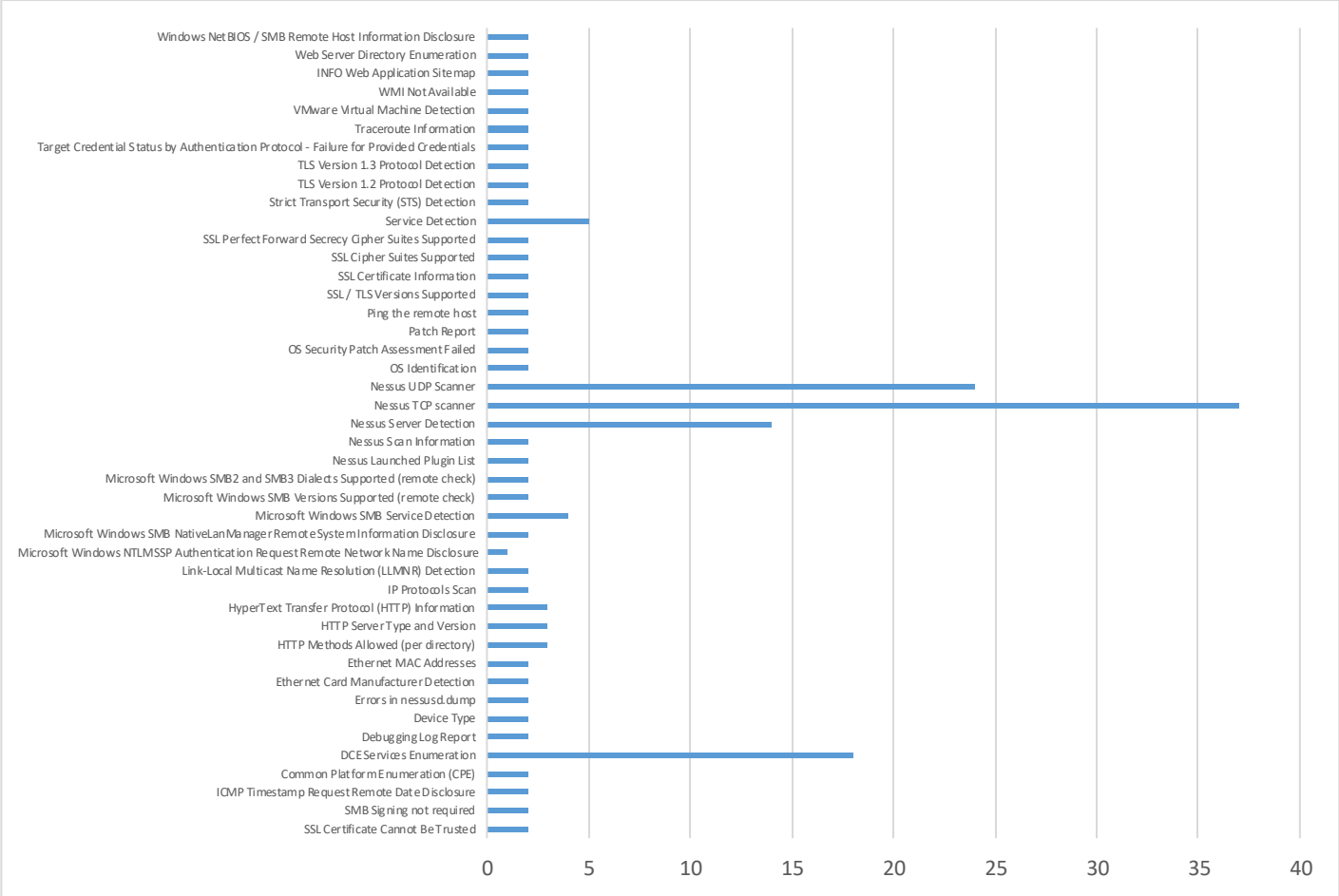
Fuente: Elaboración propia.

Se observa en el gráfico que ambas máquinas tienen las mismas vulnerabilidades conocidas, que son que no se requiere identificación del usuario para entrar al sistema y que los

certificados pueden no ser fiables. Ambas vulnerabilidades se identificaban en el modelado de amenazas como problemáticas por suplantación o spoofing.

A continuación, se muestra el último gráfico identificado, *Figura 51*, con todas las vulnerabilidades por tipo.

Figura 51. *Gráfico total de vulnerabilidades por tipo.*

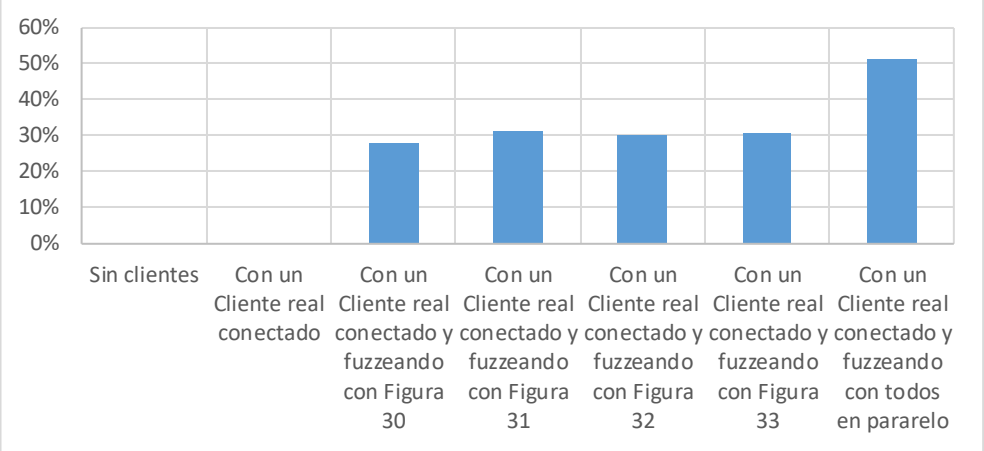


Fuente: Elaboración propia.

En este gráfico se puede observar dónde se focaliza el mayor número de vulnerabilidades, que como se ha indicado en la Figura 50 las vulnerabilidades son de severidad Info, y están en Nessus TCP y UDP scanner. En estos tipos lo relevante está en los diferentes puertos TCP y UDP abiertos que pueden conllevar a un ataque.

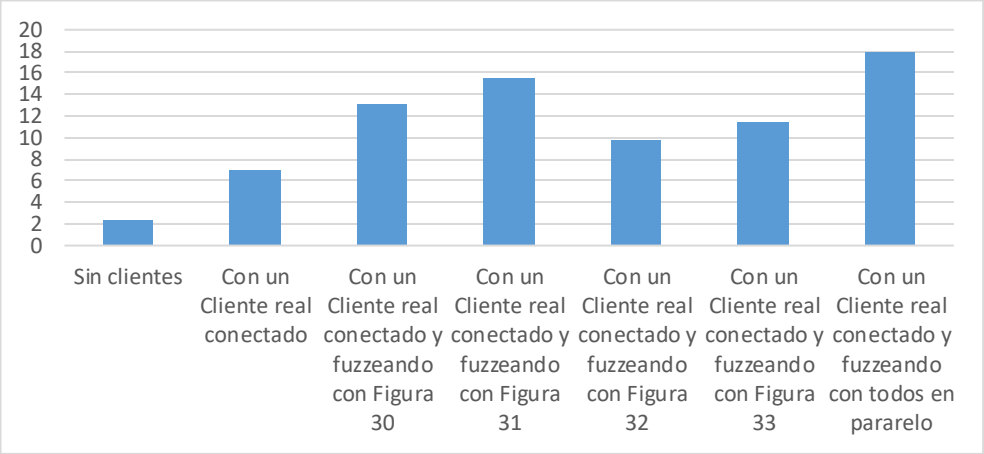
Partiendo de los datos obtenidos en **Realizar pruebas de fuzzing** sobre Magec Service, se obtienen tres gráficos: consumo de CPU, de Memoria y de Red; representados respectivamente en las Figura 52, Figura 53 y Figura 54.

Figura 52. *Gráfico de consumo de CPU en la máquina que ejecuta Magec Service.*



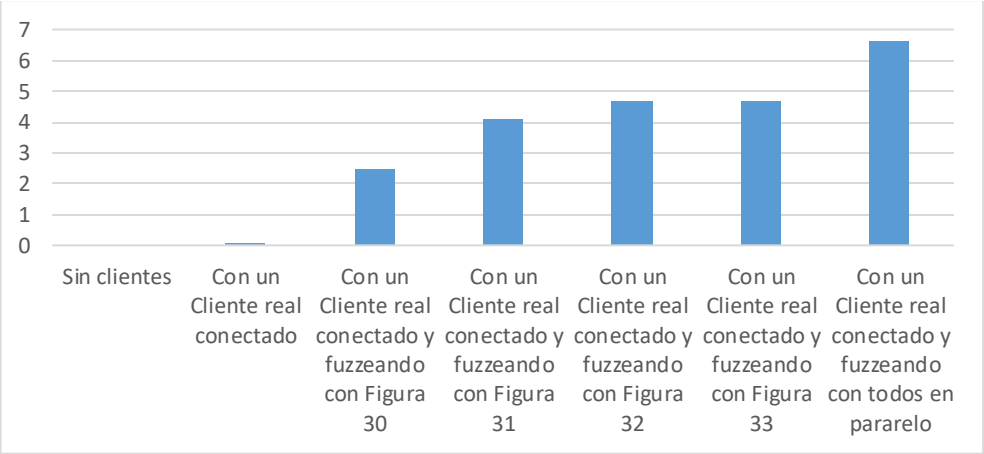
Fuente: Elaboración propia.

Figura 53. *Gráfico de consumo de Memoria en la máquina que ejecuta Magec Service.*



Fuente: Elaboración propia.

Figura 54. *Gráfico de consumo de Red en la máquina que* ejecuta Magec Service.



Fuente: Elaboración propia.

En estos tres gráficos se observa claramente que los recursos que usa la aplicación Magec Service en el equipo se multiplican por más de 30 veces el uso de CPU, por más de 2 el uso de Memoria y por más de 4 el uso de Red.

Analizando estos datos con los resultados sobre el consumo de los Sensores en la máquina cliente (Magec Client), se puede llegar a la conclusión que se puede producir un ataque de Denegación de Servicio a Magec Service. Esto se debe a que en el propio cliente durante el proceso de consumo de los sensores se produce un retraso leve cuando se realizan las pruebas de fuzzing en serie. Además, cuando se realizan las pruebas de fuzzing en paralelo el retraso en el cliente puede llegar a ser muy alto.

Estos retrasos comentados en la obtención de los datos de los sensores pueden llevar a producir un incumplimiento de la propiedad Disponibilidad. El no tener disponibilidad del sistema Magec Service en el entorno en el que se ejecuta, es decir, en el entorno militar; puede conllevar a muchos riesgos. Estos riesgos pueden ser de pérdida de información sensible, degradación de los datos, etc.

En el caso de este software la propiedad disponibilidad prevalece al mismo nivel que integridad y confidencialidad. Por tanto, hay que cuidar que el sistema nunca pierda disponibilidad para que siempre se cumplan los tres pilares.

También, sobre los datos obtenidos, se ha observado que las peticiones TCP transmitidas en la red van en claro, por lo que, se puede llegar a robar información sensible. Esto debería solucionarse con algún tipo cifrado de las comunicaciones.

Otra conclusión obtenida de los resultados es que, a pesar de que las comunicaciones van en claro, son difíciles de falsear con pruebas de fuzzing. Esto se debe a que Magec Service rechaza peticiones que no entran en su filtro de recepción de datos entre cliente-servidor.

Finalmente, se puede concluir que se han encontrado bugs de seguridad en la transmisión de los datos en las comunicaciones y en un posible ataque de Denegación de servicio al servidor.

5.4. Resumen de los Resultados

Partiendo de los estudios de los resultados obtenidos en los subapartados **Modelado de Amenazas**, **Auditoría de Código** y **Pruebas de Penetración**, se va obtiene un resumen de la seguridad de la aplicación de mando y control de vigilancia militar Magec.

Se comienza por el *Modelado de Amenazas*, donde se observa que las mayores amenazas a las que se expone Magec son Manipulación, Elevación de Privilegios y Suplantación o Spoofing. Además, gracias a la valoración DREAD se observa que las amenazas con más riesgo son la de Suplantación de los almacenes de datos de destino y origen y la Suplantación de los procesos y sensores. Estas posibles amenazas pueden llevar a una falta de las propiedades integridad, confidencialidad y autenticación.

Como conclusiones finales en esta actividad, se obtiene que la mayoría de las amenazas a las que se expone Magec son las de Manipulación, pero hay que prestar también atención a los de tipo Suplantación al ser los que peor valoración DREAD tienen.

Se continúa con la *Auditoría de Código*, y la primera conclusión que se tiene es que la mayor densidad de vulnerabilidades se encuentra en las soluciones MagecService y Core, por lo que tienen mayor probabilidad de ser atacadas. A pesar que las categorías con mayor número de vulnerabilidades son Null Dereference, Weak XML Schema: Unbounded Occurrences y Insecure Randomness, sólo hay que prestar atención a la que está en segundo lugar ya que las vulnerabilidades son de tipo mala práctica, en las otras dos categorías son en su gran mayoría falsos positivos y no hay que prestarles atención.

También, se han analizado un total de cinco vulnerabilidades de nivel de seguridad Critical en MagecClient y Framework, de las cuales cuatro de las cinco vulnerabilidades de nivel de seguridad Critical son verdaderos positivos. Además, se ha obtenido que el número total de falsos positivos es seis veces mayor que el de verdaderos positivos.

Se concluye, que en esta actividad se ha detectado que el nivel de seguridad del código de la aplicación Magec es medio, ya que la comparación con falsos positivos obtenidos es muy alta con respecto a los verdaderos positivos.

Observando los fallos en seguridad se pueden llegar a vulnerar las comunicaciones, ya que podrían no estar cifrándose de la mejor manera, llegando a vulnerar la propiedad de confidencialidad.

Finalizando con la actividad de *Pruebas de Penetración*, se ha detectado en la fase de identificación de vulnerabilidades que sólo 4 de las 180 son de severidad media y las demás de tipo informativa, a las que no hay que prestar atención. Las vulnerabilidades conocidas de severidad media son que no se requiere identificación del usuario para entrar al sistema y que

los certificados pueden no ser fiables. El mayor número de vulnerabilidades se encuentran en Nessus TCP y UDP scanner que son los puertos que se descubren abiertos y por donde puede concurrir un ataque.

Siguiendo con la fase de pruebas de fuzzing, se ha detectado durante las pruebas de fuzzing a Magec Service que los recursos que usa la aplicación en el equipo se multiplican por más de 30 veces el uso de CPU, por más de 2 el uso de Memoria y por más de 4 el uso de Red. Gracias a estos de datos se observa que se puede producir un ataque de Denegación de Servicio a la aplicación Service. Esto conlleva a un incumplimiento de la propiedad Disponibilidad. Cuando la disponibilidad de Magec Service se ve afectada, se pueden tener asociados muchos riesgos en seguridad.

Igualmente, se han detectado peticiones TCP en claro entre servidor y cliente, llegando a causar robo de información sensible. Un punto a favor, es que Magec Service rechaza peticiones que no entran en su filtro de recepción de datos entre cliente-servidor.

Como resultado de esta actividad en nivel de seguridad, hay que resaltar que las vulnerabilidades conocidas son de tipo Suplantación de identidad y las encontradas por medio de las pruebas de fuzzing son de tipo vulneración de la propiedad Disponibilidad y de robo de información sensible.

Finalmente, sobre la seguridad de Magec se pueden concluir que cuando se vulnera la propiedad disponibilidad el sistema podría quedar no operativo y no accesible; cuando se produce una falta de confidencialidad se llegaría a robar información sensible en las peticiones TCP; cuando se manipula información sensible se produciría la vulneración de la propiedad integridad; cuando se produzca una suplantación de los almacenes de datos y de los sensores se produciría una vulneración de la propiedad autenticación; y, cuando se vulneran las propiedades de resiliencia y robustez no se podría garantizar que aísle los daños y los limite, ya que se pueden producir denegaciones de servicio.

6. Conclusiones y prospectiva

El objetivo general planteado para este Piloto Experimental era aplicar buenas prácticas y principios de diseño de seguridad a un Ciclo de Vida de Desarrollo Software (SDLC) de un software dedicado a la vigilancia militar con el fin de mejorar la seguridad en la aplicación. Para llegar a cumplir este objetivo se plantearon tres más específicos que eran aplicar tres actividades del Ciclo de Vida de Desarrollo Seguro de Software (S-SDLC) en cascada seleccionado durante el estado del arte, siendo éstas el Modelado de Amenazas, la Auditoría de Código y las Pruebas de Penetración. Todos los objetivos específicos se han cumplido, ya que se han realizado las tres actividades sobre el software de vigilancia militar Magec. Implícitamente, al lograrse los objetivos específicos, el general se ha cumplido, ya que se han integrado las actividades del S-SDLC en cascada dichas anteriormente, reconociendo las posibles amenazas a las que se enfrenta Magec.

En la introducción al Piloto Experimental se realizaron las siguientes tres preguntas: ¿Cómo se debe securizar este tipo de software?, ¿Es posible securizarlo como uno de uso civil?, y, ¿Está expuesto realmente el software a ataques vulnerables en estos tipos de sistemas?

Como respuesta a las dos primeras preguntas, se resolvieron a lo largo del estado del arte. La securización del software militar es igual a la de un software de uso común. Esta securización se basa en la implantación de un S-SDLC que aplica diferentes actividades relacionadas con la seguridad a lo largo del ciclo de vida del software.

A lo largo de este Piloto Experimental se han realizado diferentes análisis de seguridad en Magec, desde el análisis de la arquitectura en búsqueda de posibles amenazas, pasando por un análisis de código estático, escaneando las vulnerabilidades conocidas de las máquinas donde se ejecuta y realizando las pruebas de fuzzing a los puertos de Magec. En cada una de las actividades del S-SDLC se han ido detectando diferentes vulnerabilidades de Magec.

Comenzando con el modelado de amenazas, se ha detectado que las mayores amenazas son de tipo Manipulación, Elevación de Privilegios y Suplantación, esta última es en la que se encuentran las amenazas con mayor riesgo según DREAD. A partir de estas conclusiones se puede observar que estas posibles amenazas pueden llevar a una falta de las propiedades integridad, confidencialidad y autenticación.

Gracias a la auditoria de código se han detectado un número bastante significativo de verdaderos positivos y malas prácticas de desarrollo en el código fuente de Magec, donde las categorías predominantes son las de Null Dereference e Insecure Randomness. A pesar de esto, predominan los falsos positivos, que son seis veces más que los verdaderos positivos. De esta actividad se puede extraer los puntos por donde podrían atacar el sistema, más en concreto, las comunicaciones podrían no estar cifrándose de la mejor manera llegando a vulnerar la propiedad de confidencialidad.

Los resultados de las pruebas de penetración se dividen en escaneo de vulnerabilidades y pruebas de fuzzing. En el escaneo de vulnerabilidades conocidas de los equipos en los que se ejecuta Magec se ha observado que sólo hay que prestar atención a las de severidad media, que son las más altas, que son no requerir la identificación del usuario para entrar al sistema y los certificados no fiables. Focalizando en la del inicio de sesión se puede unificar con los resultados del modelado de amenazas, esto se debe a que esta vulnerabilidad puede causar un ataque de Suplantación y de Elevación de privilegios. Esto puede llevar a que ocurra una falta de las propiedades confidencialidad y autenticación.

Los resultados de las pruebas de fuzzing nos interesan por los recursos consumidos por la aplicación Magec Service, que excede en recursos del sistema con cada prueba de fuzzing, llegando a producir una denegación de servicio al cliente Magec Client. Además, se ha detectado que las peticiones entre cliente-servidor por medio de TCP van en claro, no van cifradas. Estas vulnerabilidades de Magec pueden conllevar a una pérdida de las propiedades disponibilidad, confidencialidad, resiliencia y robustez. Un punto a destacar de las pruebas de fuzzing es que las peticiones no son aceptadas por Magec Service, lo que se deduce que el sistema está filtrando las peticiones que recibe, detectando y descartando las que no son correctas. Esto conlleva a que la suplantación de un cliente sea complicada de conseguir.

Por tanto, las propiedades que se ven afectadas por fallos de seguridad son disponibilidad en las pruebas de fuzzing; confidencialidad en las tres actividades realizadas; integridad en el modelado de amenazas; autenticación identificada en las tres actividades; y, resiliencia y robustez identificadas en las pruebas de penetración.

Pero, aún falta por responder a la tercera pregunta de la introducción reseñada en párrafos anteriores. Como se ha observado en los resultados obtenidos del análisis de Magec, los sistemas militares tienen las mismas vulnerabilidades que uno común. A pesar de que el

software se está ejecutando en una red aislada hay muchos factores que pueden incurrir en un ataque, ya sea por el propio personal que lo usa o porque "alguien" consigue entrar en la red por una puerta trasera. Por tanto, los softwares militares deben securizarse igual o más que los softwares de uso civil, ya que las consecuencias de un ataque pueden ser terribles.

Para finalizar, se concluye que los softwares militares deben comenzar a ser securizados con el fin de ganar seguridad global en los sistemas. Gracias a este Piloto Experimental se ha aprendido que la aplicación de un S-SDLC en un desarrollo software es de gran ayuda. Ya que con aplicar las diferentes actividades que plantea, en este caso el S-SDLC en cascada, se pueden conseguir muchas mejoras en seguridad, tanto desde el punto de vista de los requisitos, como desde la parte de desarrollo o desde el punto de vista de las pruebas de seguridad. Con la aplicación de este ciclo de vida se puede llegar a entregar un software más resistente frente a posibles ataques y mejorar la seguridad de los sistemas en redes aisladas asegurando las propiedades esenciales y complementarias de la seguridad del software.

Finalmente, se van a comentar posibles líneas de trabajo futuro que no se han podido profundizar en el Piloto Experimental:

- Realizar casos de abuso de la aplicación para fortalecer las pruebas de penetración.
- Realizar un modelado de ataques de la arquitectura Magec.
- Identificar posibles requisitos de seguridad de la arquitectura Magec.
- Realizar un análisis de riesgo arquitectónico.
- Realizar pruebas de fuzzing de las peticiones TCP a Magec Client simulando ser Magec Service.
- Realizar un análisis de código binario sobre Magec.
- Inyectar fallos en los binarios de Magec Client y Service.

7. Referencias Bibliográficas

1. Amey, P. (5 de Diciembre de 2006). *CISA*. Obtenido de Correctness by Construction: https://us-cert.cisa.gov/bsi/articles/knowledge/sdlc-process/correctness-by-construction
2. Arechaga Tarruell, G. (2017). Un Ejército Altamente Tecnológico. Toda una apuesta por el I+D+i gracias a los nuevos sistemas de mando y control. *Revista Ejército Nº 915*.
3. Becerra, P., & Sanjuan, M. (2017). Revisión de estado del arte del ciclo de vida de desarrollo de software seguro con la metodología SCRUM. *Investigación y desarrollo en TIC*. Obtenido de http://revistas.unisimon.edu.co/index.php/identic/article/view/2474
4. Bermejo Higuera, J. (2021). *Apuntes seguridad en el software.* Madrid: UNIR.
5. Bradshaw, S. (11 de Diciembre de 2010). *An introduction to fuzzing: using fuzzers (SPIKE) to find vulnerabilities*. Obtenido de INFOSEC: https://resources.infosecinstitute.com/topic/intro-to-fuzzing/
6. Chalvatzis, I., A. Karras, D., & C. Papademetiou, R. (2019). *Evaluation of Security Vulnerability Scanners for Small and Medium Enterprises Business Networks Resilience towards Risk Assessment.* Dalian, China: International Conference on Artificial Intelligence and Computer Applications (ICAICA).
7. Clausewitz, C. v. (1999). *De la Guerra (2 vol.).* Ministerio de Defensa.
8. Cubeiro Cabello, E. (2001). *dialnet.* Obtenido de Los sistemas de mando y control militar: https://dialnet.unirioja.es/descarga/articulo/4602258.pdf
9. de Vicente Mohino, J., Bermejo Higuera, J., Bermejo Higuera, J., & Sicilia Montalvo, J. (2019). The Application of a New Secure Software. *Electronics*, 1218.
10. Díaz del Río Durán, J. J. (2011). *dialnet.* Obtenido de La ciberseguridad en el ámbito militar: https://dialnet.unirioja.es/descarga/articulo/3837348.pdf
11. Figueroa, V. (2016). *OWASP.* Obtenido de Secure Software Development Life Cycle: https://owasp.org/www-pdf-archive/OWASP-LATAMTour-Patagonia-2016-rvfigueroa.pdf

12. Focus, M. (2021). *Fortify Static Code Analzer*. Obtenido de https://www.microfocus.com/es-es/products/static-code-analysis-sast/overview

13. Hernández Bejarano, M., & Baquero Rey, L. E. (2020). *Ciclo de vida de desarrollo ágil de software seguro.* Bogotá: Los Libertadores.

14. McGraw, G. (2005). *Software Security - Buildinf Security In.* Addison-Wesley Software Security Series.

15. Microsoft. (16 de Febrero de 2017). *Microsoft Threat Modeling Tool*. Obtenido de https://docs.microsoft.com/es-es/azure/security/develop/threat-modeling-tool

16. Microsoft. (2021). *Threat Modeling*. Obtenido de Microsoft: https://www.microsoft.com/en-us/securityengineering/sdl/threatmodeling

17. Mueller, P., & Yadegari, B. (2012). *The Stuxnet Worm.* Obtenido de Département des sciences de Informatique, Université del Arizona: https://www2.cs.arizona.edu/~collberg/Teaching/466-566/2012/Resources/presentations/topic9-final/report.pdf

18. Nacional, C. C. (Mayo de 2013). *CCN-STIC-400 Manual STIC.* Obtenido de https://www.ccn-cert.cni.es/series-ccn-stic/guias-de-acceso-publico-ccn-stic/4-ccn-stic-400-manual-stic/file.html

19. Nacional, C. C. (Septiembre de 2020). *Ciberamenazas y tendencias Edición 2020.* Obtenido de https://cuadernosdeseguridad.com/wp-content/uploads/2020/10/Informe-Ciberamenazas-Tendencias-2020.pdf

20. Nacional, C. C. (Septiembre de 2021). *CCN CERT CNI*. Obtenido de https://www.ccn-cert.cni.es/guias/guias-series-ccn-stic.html

21. OWASP. (Octubre de 2021). *OWASP*. Obtenido de OWASP SAMM: https://owasp.org/www-project-samm/

22. P. F., D. (18 de Diciembre de 2006). *Análisis y Modelado de Amenazas.* Obtenido de Academia: https://www.academia.edu/36649705/Analisis_y_Modelado_de_Amenazas

23. Pérez García, J. (19 de Septiembre de 2019). *Análisis de fiabilidad de las herramientas SAST. Privativo vs Libre.* Obtenido de Reunir:

https://reunir.unir.net/bitstream/handle/123456789/9554/P%c3%a9rez%20Garc%c3%ada%2c%20Javier.pdf?sequence=1&isAllowed=y

24. tutorialspoint. (Octubre de 2021). *Software - Ciclo de Vida de Desarrollo*. Obtenido de https://www.tutorialspoint.com/es/software_engineering/software_development_life_cycle.htm

25. unigoti. (2021). *Ciclo de vida del desarrollo de software*. Obtenido de https://ungoti.com/es/soluciones/desarrollo-de-software/sdlc/

26. V Congreso Nacional de i+d en Defensa y Seguridad. (2017). Toledo: Catálogo General de Publicaciones Oficiales. Obtenido de https://publicaciones.defensa.gob.es/media/downloadable/files/links/a/c/actas_v_congreso_id_2017.pdf

27. Weir, C., Hermann, B., & Fahl, S. (2020). From Needs to Actions to Secure Apps? The Effect of Requirements and Developer Practices on App Security. *29th USENIC Security Symposium*, (pág. 18).

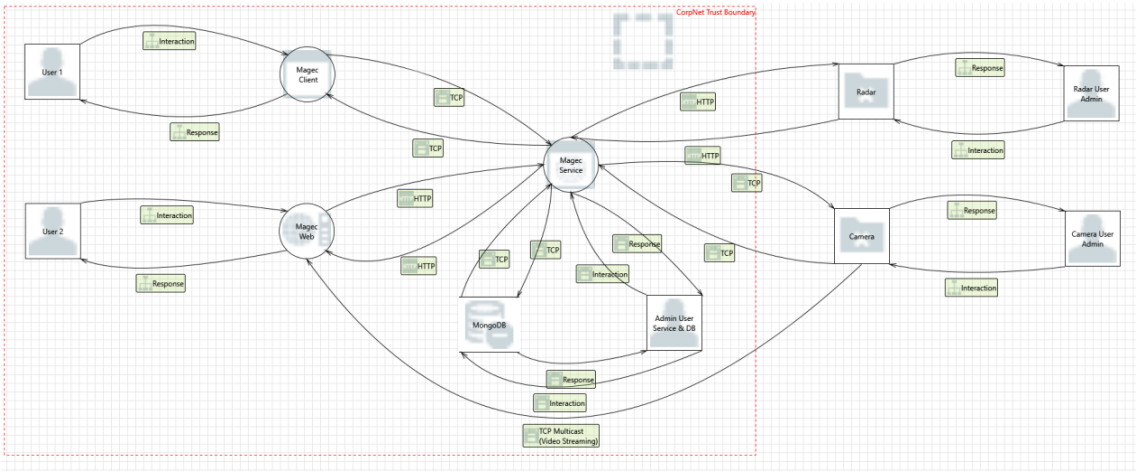
28. Wireshark. (2021). *Wireshark*. Obtenido de https://www.wireshark.org/

Anexo A. Informe de Microsoft Threat Modeling Tool

En este anexo se incluye el informe obtenido con la herramienta Microsoft Threat Modeling Tool.

A) Diagram: Magec Architecture

Figura 55. *Diagrama de la arquitectura de Magec.*



Fuente: Elaboración propia.

B) Magec Architecture Diagram Summary:

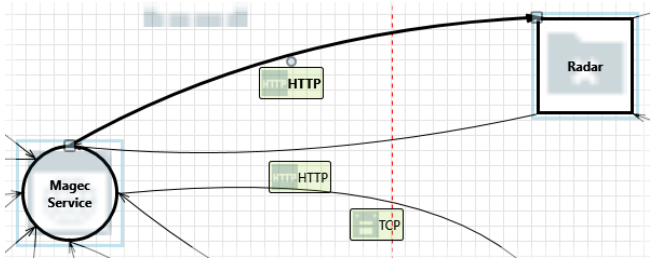
Tabla 26. ***Resumen del diagrama de la arquitectura Magec.***

Not Started	67
Not Applicable	0
Needs Investigation	0
Mitigation Implemented	0
Total	67
Total Migrated	0

Fuente: Elaboración propia.

C) Interaction: HTTP

Figura 56. *Iteración HTTP Magec Service – Radar.*



Fuente: Elaboración propia.

1. Spoofing the Magec Service Process [State: Not Started] [Priority: High]

Tabla 27. ***Spoofing del proceso de Magec Service.***

Category:	Spoofing
Description:	Magec Service may be spoofed by an attacker and this may lead to information disclosure by Radar. Consider using a standard authentication mechanism to identify the destination process.
Justification:	CAPEC 154
Dread-Damage (D):	Alto (3)
Dread-Reproducibility (R):	Alto (3)
Dread-Exploitability (E):	Alto (3)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Medio (2)
Riesgo = (R+E+DI) x (D+A) = PxI=:	48
Safeguard 1:	Supervisar la actividad en la red para detectar cualquier cambio en la comunicación (anómalo o no autorizado)
Safeguard 2:	
Safeguard 3:	

Fuente: Elaboración propia.

2. Spoofing the Radar External Entity [State: Not Started] [Priority: High]

Tabla 28. ***Spoofing de la entidad externa Radar.***

Category:	Spoofing

Description:	Radar may be spoofed by an attacker and this may lead to unauthorized access to Magec Service. Consider using a standard authentication mechanism to identify the external entity.
Justification:	CAPEC 154
Dread-Damage (D):	Alto (3)
Dread-Reproducibility (R):	Alto (3)
Dread-Exploitability (E):	Alto (3)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Medio (2)
Riesgo = (R+E+DI) x (D+A) = PxI=:	48
Safeguard 1:	Supervisar la actividad en la red para detectar cualquier cambio en la comunicación (anómalo o no autorizado)
Safeguard 2:	
Safeguard 3:	

Fuente: Elaboración propia.

3. Potential Lack of Input Validation for Magec Service [State: Not Started] [Priority: Medium]

Tabla 29. ***Falta potencial de validación de entrada para el servicio Magec.***

Category:	Tampering
Description:	Data flowing across HTTP may be tampered with by an attacker. This may lead to a denial of service attack against Magec Service or an elevation of privilege attack against Magec Service or an information disclosure by Magec Service. Failure to verify that input is as expected is a root cause of a very large number of exploitable issues. Consider all paths and the way they handle data. Verify that all input is verified for correctness using an approved list input validation approach.
Justification:	CAPEC 153
Dread-Damage (D):	Alto (3)

Dread-Reproducibility (R):	Medio (2)
Dread-Exploitability (E):	Medio (2)
Dread-Affected users (A):	Medio (2)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	25
Safeguard 1:	Validar las entradas controlando el formato, estructura y composición
Safeguard 2:	
Safeguard 3:	

Fuente: Elaboración propia.

4. JavaScript Object Notation Processing [State: Not Started] [Priority: Medium]

Tabla 30. ***Procesamiento de notación de cobjetos de JavaScript.***

Category:	Tampering
Description:	If a dataflow contains JSON, JSON processing and hijacking threats may be exploited.
Justification:	CAPEC 111
Dread-Damage (D):	Alto (3)
Dread-Reproducibility (R):	Medio (2)
Dread-Exploitability (E):	Bajo (1)
Dread-Affected users (A):	Bajo (1)
Dread-Discoverablity (DI):	Alto (3)
Riesgo = (R+E+DI) x (D+A) = PxI=:	24
Safeguard 1:	Asegurar que el código del servidor sepa diferenciar entre solicitudes legitimas o falsificadas
Safeguard 2:	Dificultar el acceso al contenido del JSON a través de un script en el lado del cliente y que los JSONs no transmiten datos confidenciales

Safeguard 3:	Hacer que las URL del sistema de recuperación de JSON sea impredecible y únicas por sesión

Fuente: Elaboración propia.

5. Potential Data Repudiation by Magec Service [State: Not Started] [Priority: Medium]

Tabla 31. ***Posible repudio de datos por Magec Service.***

Category:	Repudiation
Description:	Magec Service claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.
Justification:	CAPEC 132
Dread-Damage (D):	Medio (2)
Dread-Reproducibility (R):	Alto (3)
Dread-Exploitability (E):	Medio (2)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	30
Safeguard 1:	Utilizar medios de registro de la fuente, hora y un resumen de los datos obtenidos
Safeguard 2:	
Safeguard 3:	

Fuente: Elaboración propia.

6. Data Flow Sniffing [State: Not Started] [Priority: Low]

Tabla 32. ***Rastreo de flujo de datos.***

Category:	Information Disclosure
Description:	Data flowing across HTTP may be sniffed by an attacker. Depending on what type of data an attacker can read, it may be used to attack other parts of the system or simply be a disclosure of information leading to compliance violations. Consider encrypting the data flow.

Justification:	CAPEC 65
Dread-Damage (D):	Medio (2)
Dread-Reproducibility (R):	Bajo (1)
Dread-Exploitability (E):	Bajo (1)
Dread-Affected users (A):	Medio (2)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	12
Safeguard 1:	Cifrar todas las comunicaciones entre cliente y servidor
Safeguard 2:	Usar SSL, SSH o SCP
Safeguard 3:	Utilizar ipconfig para detectar el sniffer que esté instalado en la red

Fuente: Elaboración propia.

7. Potential Process Crash or Stop for Magec Service [State: Not Started] [Priority: Medium]

Tabla 33. ***Bloqueo potencial del proceso o parada por Magec Service.***

Category:	Denial Of Service
Description:	Magec Service crashes, halts, stops or runs slowly; in all cases violating an availability metric.
Justification:	CAPEC 74
Dread-Damage (D):	Alto (3)
Dread-Reproducibility (R):	Bajo (1)
Dread-Exploitability (E):	Medio (2)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Medio (2)
Riesgo = (R+E+DI) x (D+A) = PxI=:	30
Safeguard 1:	No confiar solamente en ubicaciones controlables por el usuario para mantener el estado del usuario
Safeguard 2:	Evitar la información confidencial en ubicaciones controlables por el usuario

Safeguard 3:	La información sensible que forma parte del estado del usuario debe protegerse adecuadamente para garantizar la confidencialidad e integridad en cada solicitud

Fuente: Elaboración propia.

8. Data Flow HTTP Is Potentially Interrupted [State: Not Started] [Priority: Low]

Tabla 34. ***El flujo de datos HTTP está potencialmente interrumpido.***

Category:	Denial Of Service
Description:	An external agent interrupts data flowing across a trust boundary in either direction.
Justification:	CAPEC 276
Dread-Damage (D):	Alto (3)
Dread-Reproducibility (R):	Bajo (1)
Dread-Exploitability (E):	Bajo (1)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	18
Safeguard 1:	Gestionar el estado del flujo de datos del software
Safeguard 2:	Gestionar correctamente el estado de los usuarios para no sufrir ataques
Safeguard 3:	

Fuente: Elaboración propia.

9. Elevation Using Impersonation [State: Not Started] [Priority: Low]

Tabla 35. ***Elevación mediante suplantación.***

Category:	Elevation Of Privilege
Description:	Magec Service may be able to impersonate the context of Radar in order to gain additional privilege.
Justification:	CAPEC 233

Dread-Damage (D):	Medio (2)
Dread-Reproducibility (R):	Medio (2)
Dread-Exploitability (E):	Medio (2)
Dread-Affected users (A):	Bajo (1)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	15
Safeguard 1:	Ejecutar las aplicaciones con menos privilegios
Safeguard 2:	Realizar una autenticación de los usuarios para acceder a la aplicación
Safeguard 3:	

Fuente: Elaboración propia.

10. Magec Service May be Subject to Elevation of Privilege Using Remote Code Execution [State: Not Started] [Priority: Low]

Tabla 36. ***Magec Service puede estar sujeto a la elevación de privilegios mediante la ejecución remota de código.***

Category:	Elevation Of Privilege
Description:	Radar may be able to remotely execute code for Magec Service.
Justification:	CAPEC 233
Dread-Damage (D):	Medio (2)
Dread-Reproducibility (R):	Medio (2)
Dread-Exploitability (E):	Medio (2)
Dread-Affected users (A):	Bajo (1)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	15
Safeguard 1:	Ejecutar las aplicaciones con menos privilegios
Safeguard 2:	Realizar una autenticación de los usuarios para acceder a la aplicación

Safeguard 3:	

Fuente: Elaboración propia.

11. Elevation by Changing the Execution Flow in Magec Service [State: Not Started] [Priority: Low]

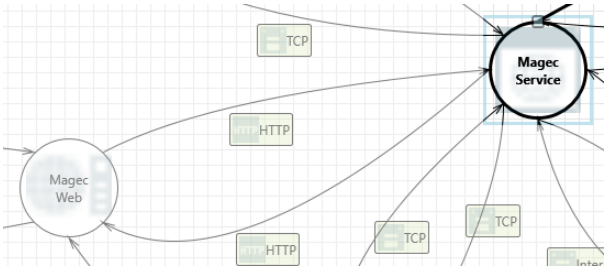
Tabla 37. ***Elevación cambiando el flujo de ejecución en Magec Service.***

Category:	Elevation Of Privilege
Description:	An attacker may pass data into Magec Service in order to change the flow of program execution within Magec Service to the attacker's choosing.
Justification:	CAPEC 233
Dread-Damage (D):	Medio (2)
Dread-Reproducibility (R):	Medio (2)
Dread-Exploitability (E):	Medio (2)
Dread-Affected users (A):	Bajo (1)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	15
Safeguard 1:	Ejecutar las aplicaciones con menos privilegios
Safeguard 2:	Realizar una autenticación de los usuarios para acceder a la aplicación
Safeguard 3:	

Fuente: Elaboración propia.

D) Interaction: HTTP

Figura 57. *Iteración HTTP Magec Service – Magec Web.*



Fuente: Elaboración propia.

12. Magec Service Process Memory Tampered [State: Not Started] [Priority: Medium]

Tabla 38. ***Memoria del proceso Magec Service manipulada.***

Category:	Tampering
Description:	If Magec Service is given access to memory, such as shared memory or pointers, or is given the ability to control what Magec Web executes (for example, passing back a function pointer.), then Magec Service can tamper with Magec Web. Consider if the function could work with less access to memory, such as passing data rather than pointers. Copy in data provided, and then validate it.
Justification:	CAPEC 129
Dread-Damage (D):	Medio (2)
Dread-Reproducibility (R):	Medio (2)
Dread-Exploitability (E):	Medio (2)
Dread-Affected users (A):	Medio (2)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	20
Safeguard 1:	Evitar el uso de acceso a memoria excesivo
Safeguard 2:	Copiar los datos que se proporcionan y validarlos
Safeguard 3:	

Fuente: Elaboración propia.

13. Elevation Using Impersonation [State: Not Started] [Priority: Low]

Tabla 39. ***Elevación usando suplantación.***

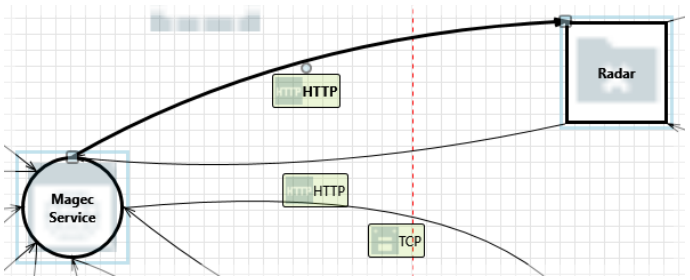
Category:	Elevation Of Privilege
Description:	Magec Web may be able to impersonate the context of Magec Service in order to gain additional privilege.
Justification:	CAPEC 233
Dread-Damage (D):	Medio (2)

Dread-Reproducibility (R):	Medio (2)
Dread-Exploitability (E):	Medio (2)
Dread-Affected users (A):	Bajo (1)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	15
Safeguard 1:	Ejecutar las aplicaciones con menos privilegios
Safeguard 2:	Realizar una autenticación de los usuarios para acceder a la aplicación
Safeguard 3:	

Fuente: Elaboración propia.

E) Interaction: HTTP

Figura 58. *Iteración HTTP Magec Service – Radar.*



Fuente: Elaboración propia.

14. Spoofing of the Radar External Destination Entity [State: Not Started] [Priority: Low]

Tabla 40. ***Suplantación de identidad de la entidad de destino externa radar.***

Category:	Spoofing
Description:	Radar may be spoofed by an attacker and this may lead to data being sent to the attacker's target instead of Radar. Consider using a standard authentication mechanism to identify the external entity.
Justification:	CAPEC 148
Dread-Damage (D):	Alto (3)
Dread-Reproducibility (R):	Bajo (1)

Dread-Exploitability (E):	Bajo (1)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	18
Safeguard 1:	Supervisar que los datos suministrados y enviados a los destinos externos se hagan a la identidad correcta
Safeguard 2:	
Safeguard 3:	

Fuente: Elaboración propia.

15. External Entity Radar Potentially Denies Receiving Data [State: Not Started] [Priority: Medium]

Tabla 41. ***La entidad externa radar niega potencialmente la recepción de datos.***

Category:	Repudiation
Description:	Radar claims that it did not receive data from a process on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.
Justification:	CAPEC 132
Dread-Damage (D):	Medio (2)
Dread-Reproducibility (R):	Alto (3)
Dread-Exploitability (E):	Medio (2)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	30
Safeguard 1:	Utilizar medios de registro de la fuente, hora y un resumen de los datos obtenidos
Safeguard 2:	
Safeguard 3:	

Fuente: Elaboración propia.

16. Data Flow HTTP Is Potentially Interrupted [State: Not Started] [Priority: Low]

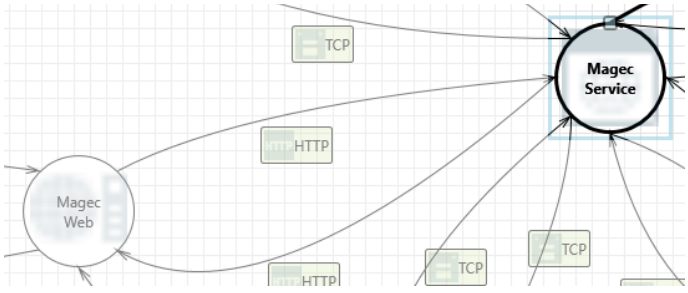
Tabla 42. ***El flujo de datos HTTP está potencialmente interrumpido.***

Category:	Denial Of Service
Description:	An external agent interrupts data flowing across a trust boundary in either direction.
Justification:	CAPEC 276
Dread-Damage (D):	Alto (3)
Dread-Reproducibility (R):	Bajo (1)
Dread-Exploitability (E):	Bajo (1)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	18
Safeguard 1:	Gestionar el estado del flujo de datos del software
Safeguard 2:	Gestionar correctamente el estado de los usuarios para no sufrir ataques
Safeguard 3:	

Fuente: Elaboración propia.

F) Interaction: HTTP

Figura 59. *Iteración HTTP Magec Service – Magec Web.*



Fuente: Elaboración propia.

17. Magec Web Process Memory Tampered [State: Not Started] [Priority: Medium]

Tabla 43. ***Memoria de proceso Magec Web manipulada.***

Category:	Tampering
Description:	If Magec Web is given access to memory, such as shared memory or pointers, or is given the ability to control what Magec Service executes (for example, passing back a function pointer.), then Magec Web can tamper with Magec Service. Consider if the function could work with less access to memory, such as passing data rather than pointers. Copy in data provided, and then validate it.
Justification:	CAPEC 129
Dread-Damage (D):	Medio (2)
Dread-Reproducibility (R):	Medio (2)
Dread-Exploitability (E):	Medio (2)
Dread-Affected users (A):	Medio (2)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	20
Safeguard 1:	Evitar el uso de acceso a memoria excesivo
Safeguard 2:	Copiar los datos que se proporcionan y validarlos
Safeguard 3:	

Fuente: Elaboración propia.

18. Elevation Using Impersonation [State: Not Started] [Priority: Low]

Tabla 44. ***Elevación usando suplantación.***

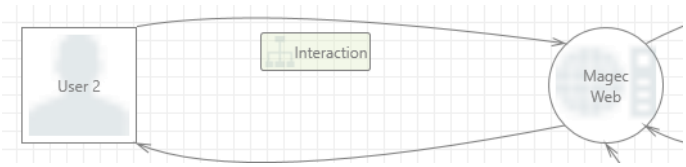
Category:	Elevation Of Privilege
Description:	Magec Service may be able to impersonate the context of Magec Web in order to gain additional privilege.
Justification:	CAPEC 233
Dread-Damage (D):	Medio (2)
Dread-Reproducibility (R):	Medio (2)
Dread-Exploitability (E):	Medio (2)

Dread-Affected users (A):	Bajo (1)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	15
Safeguard 1:	Ejecutar las aplicaciones con menos privilegios
Safeguard 2:	Realizar una autenticación de los usuarios para acceder a la aplicación
Safeguard 3:	

Fuente: Elaboración propia.

G) Interaction: Interaction

Figura 60. *Iteración Magec Web – User 2.*



Fuente: Elaboración propia.

19. Authenticated Data Flow Compromised [State: Not Started] [Priority: Low]

Tabla 45. ***Flujo de datos autenticado comprometido.***

Category:	Tampering
Description:	An attacker can read or modify data transmitted over an authenticated dataflow.
Justification:	CAPEC 94
Dread-Damage (D):	Alto (3)
Dread-Reproducibility (R):	Bajo (1)
Dread-Exploitability (E):	Bajo (1)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	18

Safeguard 1:	Asegurar que las claves públicas estén firmadas por una autoridad de certificación
Safeguard 2:	Encriptar con criptografía las comunicaciones
Safeguard 3:	Utilizar autenticación mutua fuerte en todos los canales de comunicación y realizar un intercambio de claves públicas en un canal seguro

Fuente: Elaboración propia.

20. Elevation Using Impersonation [State: Not Started] [Priority: Low]

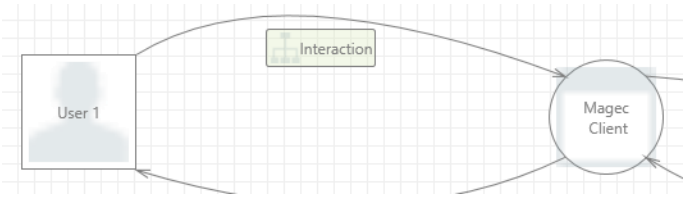
Tabla 46. ***Elevación usando suplantación.***

Category:	Elevation Of Privilege
Description:	Magec Web may be able to impersonate the context of User 2 in order to gain additional privilege.
Justification:	CAPEC 233
Dread-Damage (D):	Medio (2)
Dread-Reproducibility (R):	Medio (2)
Dread-Exploitability (E):	Medio (2)
Dread-Affected users (A):	Bajo (1)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	15
Safeguard 1:	Ejecutar las aplicaciones con menos privilegios
Safeguard 2:	Realizar una autenticación de los usuarios para acceder a la aplicación
Safeguard 3:	

Fuente: Elaboración propia.

H) Interaction: Interaction

Figura 61. *Iteración Magec Client – User 1.*



Fuente: Elaboración propia.

21. Authenticated Data Flow Compromised [State: Not Started] [Priority: Low]

Tabla 47. ***Flujo de datos autenticado comprometido.***

Category:	Tampering
Description:	An attacker can read or modify data transmitted over an authenticated dataflow.
Justification:	CAPEC 94
Dread-Damage (D):	Alto (3)
Dread-Reproducibility (R):	Bajo (1)
Dread-Exploitability (E):	Bajo (1)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	18
Safeguard 1:	Asegurar que las claves públicas estén firmadas por una autoridad de certificación
Safeguard 2:	Encriptar con criptografía las comunicaciones
Safeguard 3:	Utilizar autenticación mutua fuerte en todos los canales de comunicación y realizar un intercambio de claves públicas en un canal seguro

Fuente: Elaboración propia.

22. Elevation Using Impersonation [State: Not Started] [Priority: Low]

Tabla 48. ***Elevación usando suplantación.***

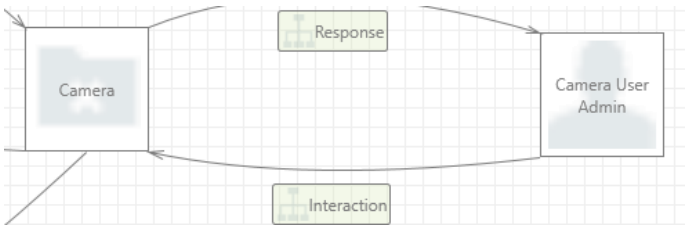
Category:	Elevation Of Privilege

Description:	Magec Client may be able to impersonate the context of User 1 in order to gain additional privilege.
Justification:	CAPEC 233
Dread-Damage (D):	Medio (2)
Dread-Reproducibility (R):	Medio (2)
Dread-Exploitability (E):	Medio (2)
Dread-Affected users (A):	Bajo (1)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	15
Safeguard 1:	Ejecutar las aplicaciones con menos privilegios
Safeguard 2:	Realizar una autenticación de los usuarios para acceder a la aplicación
Safeguard 3:	

Fuente: Elaboración propia.

I) Interaction: Interaction

Figura 62. *Iteración Camera – Camera User Admin.*



Fuente: Elaboración propia.

23. Authenticated Data Flow Compromised [State: Not Started] [Priority: Low]

Tabla 49. ***Flujo de datos autenticado comprometido.***

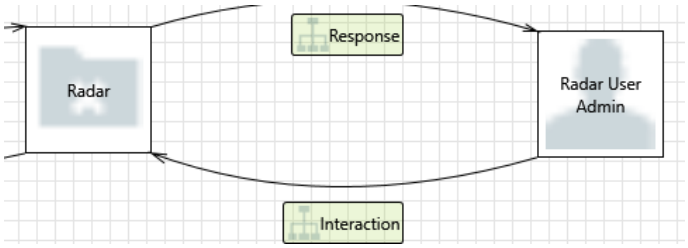
Category:	Tampering
Description:	An attacker can read or modify data transmitted over an authenticated dataflow.
Justification:	CAPEC 94

Dread-Damage (D):	Alto (3)
Dread-Reproducibility (R):	Bajo (1)
Dread-Exploitability (E):	Bajo (1)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	18
Safeguard 1:	Asegurar que las claves públicas estén firmadas por una autoridad de certificación
Safeguard 2:	Encriptar con criptografía las comunicaciones
Safeguard 3:	Utilizar autenticación mutua fuerte en todos los canales de comunicación y realizar un intercambio de claves públicas en un canal seguro

Fuente: Elaboración propia.

J) Interaction: Interaction

Figura 63. *Iteración Radar – Camera User Admin.*



Fuente: Elaboración propia.

24. Authenticated Data Flow Compromised [State: Not Started] [Priority: Low]

Tabla 50. ***Flujo de datos autenticado comprometido.***

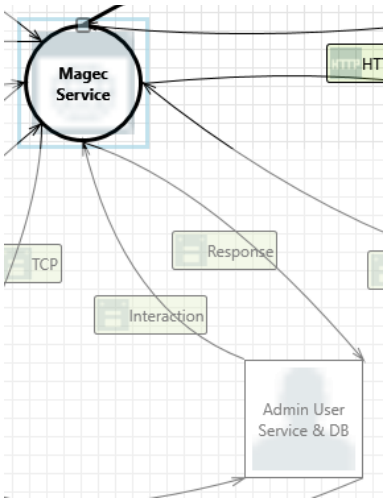
Category:	Tampering
Description:	An attacker can read or modify data transmitted over an authenticated dataflow.
Justification:	CAPEC 94
Dread-Damage (D):	Alto (3)

Dread-Reproducibility (R):	Bajo (1)
Dread-Exploitability (E):	Bajo (1)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	18
Safeguard 1:	Asegurar que las claves públicas estén firmadas por una autoridad de certificación
Safeguard 2:	Encriptar con criptografía las comunicaciones
Safeguard 3:	Utilizar autenticación mutua fuerte en todos los canales de comunicación y realizar un intercambio de claves públicas en un canal seguro

Fuente: Elaboración propia.

K) Interaction: Interaction

Figura 64. *Iteración Magec Service – Admin User Service & DB.*



Fuente: Elaboración propia.

25. Authenticated Data Flow Compromised [State: Not Started] [Priority: Low]

Tabla 51. ***Flujo de datos autenticado comprometido.***

Category:	Tampering

Description:	An attacker can read or modify data transmitted over an authenticated dataflow.
Justification:	CAPEC 94
Dread-Damage (D):	Alto (3)
Dread-Reproducibility (R):	Bajo (1)
Dread-Exploitability (E):	Bajo (1)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	18
Safeguard 1:	Asegurar que las claves públicas estén firmadas por una autoridad de certificación
Safeguard 2:	Encriptar con criptografía las comunicaciones
Safeguard 3:	Utilizar autenticación mutua fuerte en todos los canales de comunicación y realizar un intercambio de claves públicas en un canal seguro

Fuente: Elaboración propia.

26. Elevation Using Impersonation [State: Not Started] [Priority: Low]

Tabla 52. ***Elevación usando suplantación.***

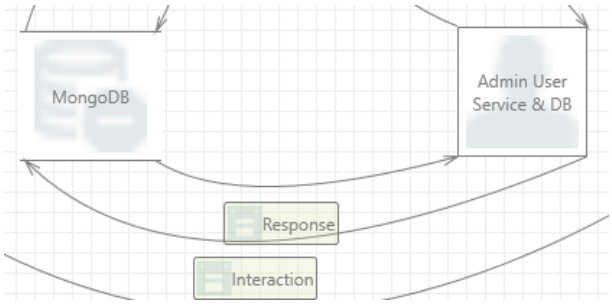
Category:	Elevation Of Privilege
Description:	Magec Service may be able to impersonate the context of Admin User Service & DB in order to gain additional privilege.
Justification:	CAPEC 233
Dread-Damage (D):	Medio (2)
Dread-Reproducibility (R):	Medio (2)
Dread-Exploitability (E):	Medio (2)
Dread-Affected users (A):	Bajo (1)
Dread-Discoverablity (DI):	Bajo (1)

Riesgo = (R+E+DI) x (D+A) = PxI=:	15
Safeguard 1:	Ejecutar las aplicaciones con menos privilegios
Safeguard 2:	Realizar una autenticación de los usuarios para acceder a la aplicación
Safeguard 3:	

Fuente: Elaboración propia.

L) Interaction: Interaction

Figura 65. *Iteración MongoDB – Admin User Service & DB.*



Fuente: Elaboración propia.

27. Spoofing of Destination Data Store MongoDB [State: Not Started] [Priority: High]

Tabla 53. ***Falsificación del almacén de datos de destino MongoDB.***

Category:	Spoofing
Description:	MongoDB may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of MongoDB. Consider using a standard authentication mechanism to identify the destination data store.
Justification:	CAPEC 194
Dread-Damage (D):	Alto (3)
Dread-Reproducibility (R):	Medio (2)
Dread-Exploitability (E):	Medio (2)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Alto (3)

Riesgo = (R+E+DI) x (D+A) = PxI=:	42
Safeguard 1:	Supervisar que el almacenamiento de los datos sea el correcto
Safeguard 2:	
Safeguard 3:	

Fuente: Elaboración propia.

28. Authenticated Data Flow Compromised [State: Not Started] [Priority: Low]

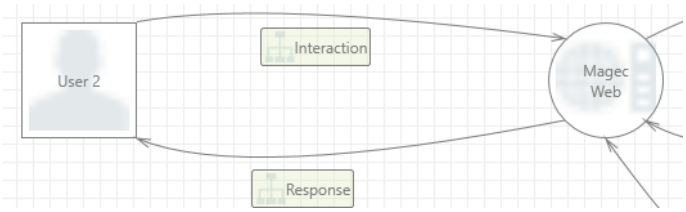
Tabla 54. ***Flujo de datos autenticado comprometido.***

Category:	Tampering
Description:	An attacker can read or modify data transmitted over an authenticated dataflow.
Justification:	CAPEC 94
Dread-Damage (D):	Alto (3)
Dread-Reproducibility (R):	Bajo (1)
Dread-Exploitability (E):	Bajo (1)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	18
Safeguard 1:	Asegurar que las claves públicas estén firmadas por una autoridad de certificación
Safeguard 2:	Encriptar con criptografía las comunicaciones
Safeguard 3:	Utilizar autenticación mutua fuerte en todos los canales de comunicación y realizar un intercambio de claves públicas en un canal seguro

Fuente: Elaboración propia.

M) Interaction: Response

Figura 66. *Iteración de respuesta Magec Web – User 2.*



Fuente: Elaboración propia.

29. Authenticated Data Flow Compromised [State: Not Started] [Priority: Low]

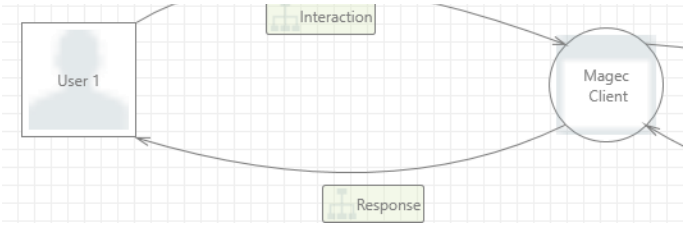
Tabla 55. ***Flujo de datos autenticado comprometido.***

Category:	Tampering
Description:	An attacker can read or modify data transmitted over an authenticated dataflow.
Justification:	CAPEC 94
Dread-Damage (D):	Alto (3)
Dread-Reproducibility (R):	Bajo (1)
Dread-Exploitability (E):	Bajo (1)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	18
Safeguard 1:	Asegurar que las claves públicas estén firmadas por una autoridad de certificación
Safeguard 2:	Encriptar con criptografía las comunicaciones
Safeguard 3:	Utilizar autenticación mutua fuerte en todos los canales de comunicación y realizar un intercambio de claves públicas en un canal seguro

Fuente: Elaboración propia.

N) Interaction: Response

Figura 67. *Iteración de respuesta Magec Client – User 1.*



Fuente: Elaboración propia.

30. Authenticated Data Flow Compromised [State: Not Started] [Priority: Low]

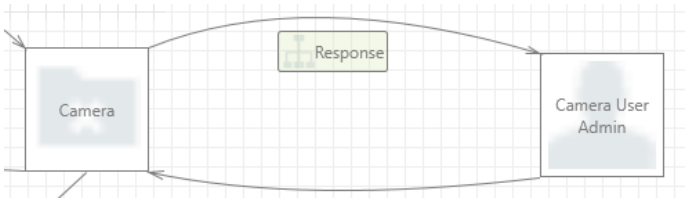
Tabla 56. ***Flujo de datos autenticado comprometido.***

Category:	Tampering
Description:	An attacker can read or modify data transmitted over an authenticated dataflow.
Justification:	CAPEC 94
Dread-Damage (D):	Alto (3)
Dread-Reproducibility (R):	Bajo (1)
Dread-Exploitability (E):	Bajo (1)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	18
Safeguard 1:	Asegurar que las claves públicas estén firmadas por una autoridad de certificación
Safeguard 2:	Encriptar con criptografía las comunicaciones
Safeguard 3:	Utilizar autenticación mutua fuerte en todos los canales de comunicación y realizar un intercambio de claves públicas en un canal seguro

Fuente: Elaboración propia.

O) Interaction: Response

Figura 68. *Iteración de respuesta Camera – Camera User Admin.*



Fuente: Elaboración propia.

31. Authenticated Data Flow Compromised [State: Not Started] [Priority: Low]

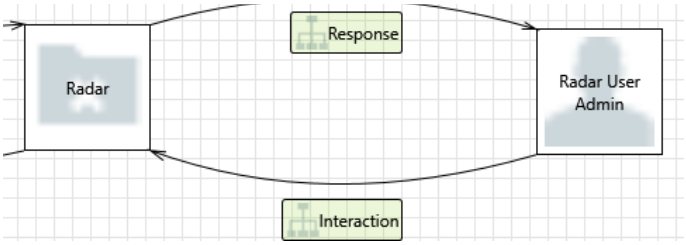
Tabla 57. ***Flujo de datos autenticado comprometido.***

Category:	Tampering
Description:	An attacker can read or modify data transmitted over an authenticated dataflow.
Justification:	CAPEC 94
Dread-Damage (D):	Alto (3)
Dread-Reproducibility (R):	Bajo (1)
Dread-Exploitability (E):	Bajo (1)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	18
Safeguard 1:	Asegurar que las claves públicas estén firmadas por una autoridad de certificación
Safeguard 2:	Encriptar con criptografía las comunicaciones
Safeguard 3:	Utilizar autenticación mutua fuerte en todos los canales de comunicación y realizar un intercambio de claves públicas en un canal seguro

Fuente: Elaboración propia.

P) Interaction: Response

Figura 69. *Iteración de respuesta Radar – Camera User Admin.*



Fuente: Elaboración propia.

32. Authenticated Data Flow Compromised [State: Not Started] [Priority: Low]

Tabla 58. ***Flujo de datos autenticado comprometido.***

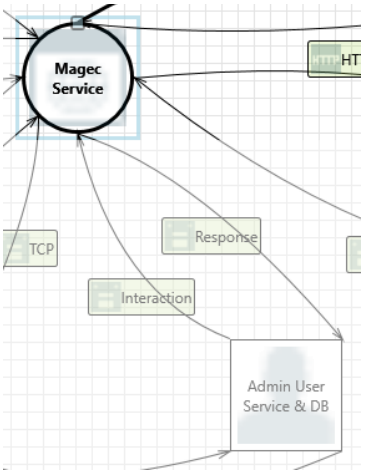
Category:	Tampering
Description:	An attacker can read or modify data transmitted over an authenticated dataflow.
Justification:	CAPEC 94
Dread-Damage (D):	Alto (3)
Dread-Reproducibility (R):	Bajo (1)
Dread-Exploitability (E):	Bajo (1)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	18
Safeguard 1:	Asegurar que las claves públicas estén firmadas por una autoridad de certificación
Safeguard 2:	Encriptar con criptografía las comunicaciones
Safeguard 3:	Utilizar autenticación mutua fuerte en todos los canales de comunicación y realizar un intercambio de claves públicas en un canal seguro

Fuente: Elaboración propia.

Q) Interaction: Response

Figura 70. *Iteración de respuesta Magec Service – Admin User Service & DB.*

Fuente: Elaboración propia.



33. Authenticated Data Flow Compromised [State: Not Started] [Priority: Low]

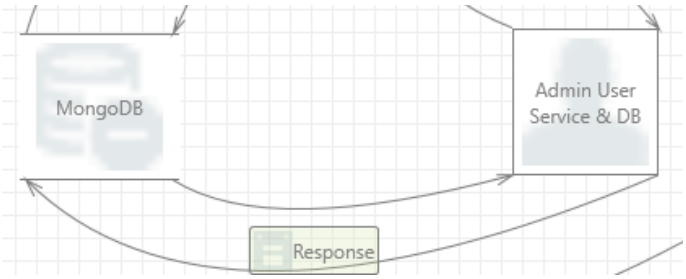
Tabla 59. ***Flujo de datos autenticado comprometido.***

Category:	Tampering
Description:	An attacker can read or modify data transmitted over an authenticated dataflow.
Justification:	CAPEC 94
Dread-Damage (D):	Alto (3)
Dread-Reproducibility (R):	Bajo (1)
Dread-Exploitability (E):	Bajo (1)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	18
Safeguard 1:	Asegurar que las claves públicas estén firmadas por una autoridad de certificación
Safeguard 2:	Encriptar con criptografía las comunicaciones
Safeguard 3:	Utilizar autenticación mutua fuerte en todos los canales de comunicación y realizar un intercambio de claves públicas en un canal seguro

Fuente: Elaboración propia.

R) Interaction: Response

Figura 71. *Iteración de respuesta MongoDB – Admin User Service & DB.*



Fuente: Elaboración propia.

34. Spoofing of Source Data Store MongoDB [State: Not Started] [Priority: High]

Tabla 60. ***Falsificación del almacén de datos de origen MongoDB.***

Category:	Spoofing
Description:	MongoDB may be spoofed by an attacker and this may lead to incorrect data delivered to Admin User Service & DB. Consider using a standard authentication mechanism to identify the source data store.
Justification:	CAPEC 154
Dread-Damage (D):	Alto (3)
Dread-Reproducibility (R):	Alto (3)
Dread-Exploitability (E):	Alto (3)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Medio (2)
Riesgo = (R+E+DI) x (D+A) = PxI=:	48
Safeguard 1:	Supervisar la actividad en la red para detectar cualquier cambio en la comunicación (anómalo o no autorizado)
Safeguard 2:	
Safeguard 3:	

Fuente: Elaboración propia.

35. Authenticated Data Flow Compromised [State: Not Started] [Priority: High]

Tabla 61. ***Flujo de datos autenticado comprometido.***

Category:	Tampering
Description:	An attacker can read or modify data transmitted over an authenticated dataflow.
Justification:	CAPEC 94
Dread-Damage (D):	Alto (3)
Dread-Reproducibility (R):	Bajo (1)
Dread-Exploitability (E):	Bajo (1)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	18
Safeguard 1:	Asegurar que las claves públicas estén firmadas por una autoridad de certificación
Safeguard 2:	Encriptar con criptografía las comunicaciones
Safeguard 3:	Utilizar autenticación mutua fuerte en todos los canales de comunicación y realizar un intercambio de claves públicas en un canal seguro

Fuente: Elaboración propia.

36. Weak Access Control for a Resource [State: Not Started] [Priority: High]

Tabla 62. ***Control de acceso débil para un recurso.***

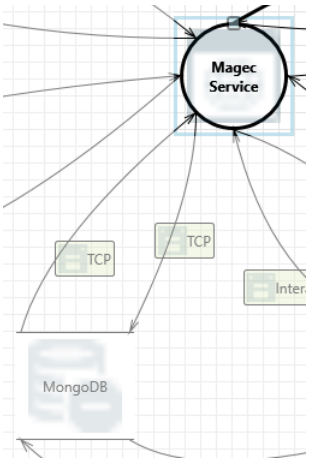
Category:	Information Disclosure
Description:	Improper data protection of MongoDB can allow an attacker to read information not intended for disclosure. Review authorization settings.
Justification:	CAPEC 180
Dread-Damage (D):	Medio (2)
Dread-Reproducibility (R):	Alto (3)
Dread-Exploitability (E):	Alto (3)

Dread-Affected users (A):	Medio (2)
Dread-Discoverablity (DI):	Alto (3)
Riesgo = (R+E+DI) x (D+A) = PxI=:	36
Safeguard 1:	Configurar el control de acceso correctamente
Safeguard 2:	
Safeguard 3:	

Fuente: Elaboración propia.

S) Interaction: TCP

Figura 72. *Iteración TCP Magec Service – MongoDB.*



Fuente: Elaboración propia.

37. Spoofing of Destination Data Store MongoDB [State: Not Started] [Priority: High]

Tabla 63. ***Falsificación del almacén de datos de destino MongoDB.***

Category:	Spoofing
Description:	MongoDB may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of MongoDB. Consider using a standard authentication mechanism to identify the destination data store.
Justification:	CAPEC 194
Dread-Damage (D):	Alto (3)
Dread-Reproducibility (R):	Medio (2)

Dread-Exploitability (E):	Medio (2)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Alto (3)
Riesgo = (R+E+DI) x (D+A) = PxI=:	42
Safeguard 1:	Supervisar que el almacenamiento de los datos sea el correcto
Safeguard 2:	
Safeguard 3:	

Fuente: Elaboración propia.

38. Potential Excessive Resource Consumption for Magec Service or MongoDB [State: Not Started] [Priority: Medium]

Tabla 64. ***Potencial consumo excesivo de recursos para Magec Service o MongoDB.***

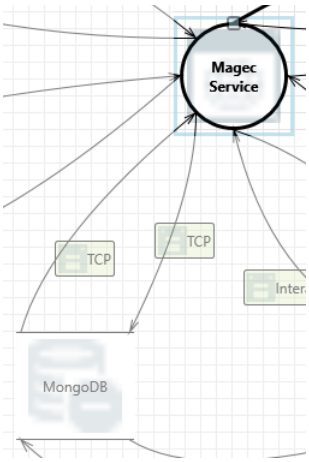
Category:	Denial Of Service
Description:	Does Magec Service or MongoDB take explicit steps to control resource consumption? Resource consumption attacks can be hard to deal with, and there are times that it makes sense to let the OS do the job. Be careful that your resource requests don't deadlock, and that they do timeout.
Justification:	CAPEC 130
Dread-Damage (D):	Medio (2)
Dread-Reproducibility (R):	Medio (2)
Dread-Exploitability (E):	Alto (3)
Dread-Affected users (A):	Medio (2)
Dread-Discoverablity (DI):	Alto (3)
Riesgo = (R+E+DI) x (D+A) = PxI=:	32
Safeguard 1:	Limitar la cantidad de recursos a los que pueden acceder los usuarios sin privilegios

Safeguard 2:	Suponer que todas las entradas son maliciosas y utilizar una configuración que limite los recursos
Safeguard 3:	Limitar todas las solicitudes para que sea más difícil consumir recursos más rápidamente de lo que pueden volver a liberarse

Fuente: Elaboración propia.

T) Interaction: TCP

Figura 73. *Iteración TCP Magec Service – MongoDB.*



Fuente: Elaboración propia.

39. Spoofing of Source Data Store MongoDB [State: Not Started] [Priority: High]

Tabla 65. ***Falsificación del almacén de datos de origen MongoDB.***

Category:	Spoofing
Description:	MongoDB may be spoofed by an attacker and this may lead to incorrect data delivered to Magec Service. Consider using a standard authentication mechanism to identify the source data store.
Justification:	CAPEC 154
Dread-Damage (D):	Alto (3)
Dread-Reproducibility (R):	Alto (3)
Dread-Exploitability (E):	Alto (3)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Medio (2)

Riesgo = (R+E+DI) x (D+A) = PxI=:	48
Safeguard 1:	Supervisar la actividad en la red para detectar cualquier cambio en la comunicación (anómalo o no autorizado)
Safeguard 2:	
Safeguard 3:	

Fuente: Elaboración propia.

40. Weak Access Control for a Resource [State: Not Started] [Priority: High]

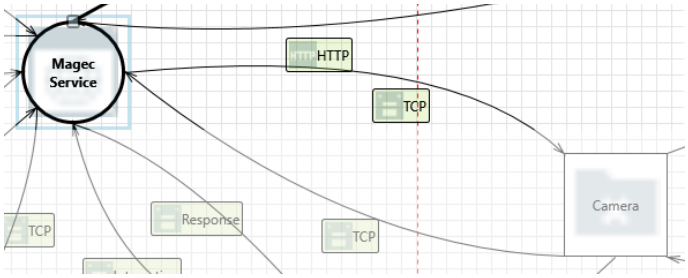
Tabla 66. ***Control de acceso débil para un recurso.***

Category:	Information Disclosure
Description:	Improper data protection of MongoDB can allow an attacker to read information not intended for disclosure. Review authorization settings.
Justification:	CAPEC 180
Dread-Damage (D):	Medio (2)
Dread-Reproducibility (R):	Alto (3)
Dread-Exploitability (E):	Alto (3)
Dread-Affected users (A):	Medio (2)
Dread-Discoverablity (DI):	Alto (3)
Riesgo = (R+E+DI) x (D+A) = PxI=:	36
Safeguard 1:	Configurar el control de acceso correctamente
Safeguard 2:	
Safeguard 3:	

Fuente: Elaboración propia.

U) Interaction: TCP

Figura 74. *Iteración TCP Magec Server – Camera.*



Fuente: Elaboración propia.

41. Spoofing the Magec Service Process [State: Not Started] [Priority: High]

Tabla 67. ***Falsificación del proceso de Magec Service.***

Category:	Spoofing
Description:	Magec Service may be spoofed by an attacker and this may lead to information disclosure by Camera. Consider using a standard authentication mechanism to identify the destination process.
Justification:	CAPEC 154
Dread-Damage (D):	Alto (3)
Dread-Reproducibility (R):	Alto (3)
Dread-Exploitability (E):	Alto (3)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Medio (2)
Riesgo = (R+E+DI) x (D+A) = PxI=:	48
Safeguard 1:	Supervisar la actividad en la red para detectar cualquier cambio en la comunicación (anómalo o no autorizado)
Safeguard 2:	
Safeguard 3:	

Fuente: Elaboración propia.

42. Spoofing the Camera External Entity [State: Not Started] [Priority: High]

Tabla 68. ***Falsificación de la entidad externa cámara.***

Category:	Spoofing

Description:	Camera may be spoofed by an attacker and this may lead to unauthorized access to Magec Service. Consider using a standard authentication mechanism to identify the external entity.
Justification:	CAPEC 154
Dread-Damage (D):	Alto (3)
Dread-Reproducibility (R):	Alto (3)
Dread-Exploitability (E):	Alto (3)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Medio (2)
Riesgo = (R+E+DI) x (D+A) = PxI=:	48
Safeguard 1:	Supervisar la actividad en la red para detectar cualquier cambio en la comunicación (anómalo o no autorizado)
Safeguard 2:	
Safeguard 3:	

Fuente: Elaboración propia.

43. Potential Lack of Input Validation for Magec Service [State: Not Started] [Priority: Medium]

Tabla 69. ***Falta potencial de validación de entrada para Magec Service.***

Category:	Tampering
Description:	Data flowing across TCP may be tampered with by an attacker. This may lead to a denial of service attack against Magec Service or an elevation of privilege attack against Magec Service or an information disclosure by Magec Service. Failure to verify that input is as expected is a root cause of a very large number of exploitable issues. Consider all paths and the way they handle data. Verify that all input is verified for correctness using an approved list input validation approach.
Justification:	CAPEC 153
Dread-Damage (D):	Alto (3)

Dread-Reproducibility (R):	Medio (2)
Dread-Exploitability (E):	Medio (2)
Dread-Affected users (A):	Medio (2)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	25
Safeguard 1:	Validar las entradas controlando el formato, estructura y composición
Safeguard 2:	
Safeguard 3:	

Fuente: Elaboración propia.

44. Potential Data Repudiation by Magec Service [State: Not Started] [Priority: Medium]

Tabla 70. ***Posible repudio de datos por parte de Magec Service.***

Category:	Repudiation
Description:	Magec Service claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.
Justification:	CAPEC 132
Dread-Damage (D):	Medio (2)
Dread-Reproducibility (R):	Alto (3)
Dread-Exploitability (E):	Medio (2)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	30
Safeguard 1:	Utilizar medios de registro de la fuente, hora y un resumen de los datos obtenidos
Safeguard 2:	
Safeguard 3:	

Fuente: Elaboración propia.

45. Data Flow Sniffing [State: Not Started] [Priority: Low]

Tabla 71. ***Rastreo de flujo de datos.***

Category:	Information Disclosure
Description:	Data flowing across TCP may be sniffed by an attacker. Depending on what type of data an attacker can read, it may be used to attack other parts of the system or simply be a disclosure of information leading to compliance violations. Consider encrypting the data flow.
Justification:	CAPEC 65
Dread-Damage (D):	Medio (2)
Dread-Reproducibility (R):	Bajo (1)
Dread-Exploitability (E):	Bajo (1)
Dread-Affected users (A):	Bajo (1)
Dread-Discoverablity (DI):	Medio (2)
Riesgo = (R+E+DI) x (D+A) = PxI=:	12
Safeguard 1:	Cifrar todas las comunicaciones entre cliente y servidor
Safeguard 2:	Usar SSL, SSH o SCP
Safeguard 3:	Utilizar ipconfig para detectar el sniffer que esté instalado en la red

Fuente: Elaboración propia.

46. Potential Process Crash or Stop for Magec Service [State: Not Started] [Priority: Medium]

Tabla 72. ***Posible bloqueo o detención del proceso para Magec Service.***

Category:	Denial Of Service
Description:	Magec Service crashes, halts, stops or runs slowly; in all cases violating an availability metric.
Justification:	CAPEC 74
Dread-Damage (D):	Alto (3)
Dread-Reproducibility (R):	Bajo (1)

Dread-Exploitability (E):	Medio (2)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Medio (2)
Riesgo = (R+E+DI) x (D+A) = PxI=:	30
Safeguard 1:	No confiar solamente en ubicaciones controlables por el usuario para mantener el estado del usuario y evitar la información confidencial en ubicaciones controlables por el usuario
Safeguard 2:	Evitar la información confidencial en ubicaciones controlables por el usuario
Safeguard 3:	La información sensible que forma parte del estado del usuario debe protegerse adecuadamente para garantizar la confidencialidad e integridad en cada solicitud

Fuente: Elaboración propia.

47. Data Flow TCP Is Potentially Interrupted [State: Not Started] [Priority: Low]

Tabla 73. ***El flujo de datos TCP está potencialmente interrumpido.***

Category:	Denial Of Service
Description:	An external agent interrupts data flowing across a trust boundary in either direction.
Justification:	CAPEC 276
Dread-Damage (D):	Alto (3)
Dread-Reproducibility (R):	Bajo (1)
Dread-Exploitability (E):	Bajo (1)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	18
Safeguard 1:	Gestionar el estado del flujo de datos del software
Safeguard 2:	Gestionar correctamente el estado de los usuarios para no sufrir ataques
Safeguard 3:	

Fuente: Elaboración propia.

48. Elevation Using Impersonation [State: Not Started] [Priority: Medium]

Tabla 74. ***Elevación usando suplantación.***

Category:	Elevation Of Privilege
Description:	Magec Service may be able to impersonate the context of Camera in order to gain additional privilege.
Justification:	CAPEC 233
Dread-Damage (D):	Medio (2)
Dread-Reproducibility (R):	Medio (2)
Dread-Exploitability (E):	Medio (2)
Dread-Affected users (A):	Bajo (1)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	15
Safeguard 1:	Ejecutar las aplicaciones con menos privilegios
Safeguard 2:	Realizar una autenticación de los usuarios para acceder a la aplicación
Safeguard 3:	

Fuente: Elaboración propia.

49. Magec Service May be Subject to Elevation of Privilege Using Remote Code Execution [State: Not Started] [Priority: Low]

Tabla 75. ***Magec Service puede estar sujeto a la elevación de privilegios mediante la ejecución remota de código.***

Category:	Elevation Of Privilege
Description:	Camera may be able to remotely execute code for Magec Service.
Justification:	CAPEC 233
Dread-Damage (D):	Medio (2)
Dread-Reproducibility (R):	Medio (2)
Dread-Exploitability (E):	Medio (2)

Dread-Affected users (A):	Bajo (1)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	15
Safeguard 1:	Ejecutar las aplicaciones con menos privilegios
Safeguard 2:	Realizar una autenticación de los usuarios para acceder a la aplicación
Safeguard 3:	

Fuente: Elaboración propia.

50. Elevation by Changing the Execution Flow in Magec Service [State: Not Started] [Priority: Low]

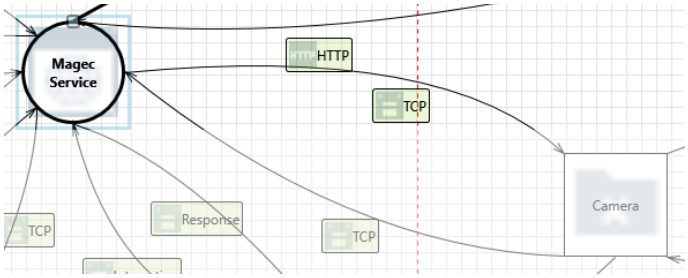
Tabla 76. ***Elevación cambiando el flujo de ejecución en Magec Service.***

Category:	Elevation Of Privilege
Description:	An attacker may pass data into Magec Service in order to change the flow of program execution within Magec Service to the attacker's choosing.
Justification:	CAPEC 233
Dread-Damage (D):	Medio (2)
Dread-Reproducibility (R):	Medio (2)
Dread-Exploitability (E):	Medio (2)
Dread-Affected users (A):	Bajo (1)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	15
Safeguard 1:	Ejecutar las aplicaciones con menos privilegios
Safeguard 2:	Realizar una autenticación de los usuarios para acceder a la aplicación
Safeguard 3:	

Fuente: Elaboración propia.

V) Interaction: TCP

Figura 75. *Iteración TCP Magec Service – Camera.*



Fuente: Elaboración propia.

51. Spoofing of the Camera External Destination Entity [State: Not Started] [Priority: Low]

Tabla 77. ***Suplantación de identidad de la entidad de destino externa cámara.***

Category:	Spoofing
Description:	Camera may be spoofed by an attacker and this may lead to data being sent to the attacker's target instead of Camera. Consider using a standard authentication mechanism to identify the external entity.
Justification:	CAPEC 148
Dread-Damage (D):	Alto (3)
Dread-Reproducibility (R):	Bajo (1)
Dread-Exploitability (E):	Bajo (1)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	18
Safeguard 1:	Supervisar que los datos suministrados y enviados a los destinos externos se hagan a la identidad correcta
Safeguard 2:	
Safeguard 3:	

Fuente: Elaboración propia.

52. External Entity Camera Potentially Denies Receiving Data [State: Not Started] [Priority: Medium]

Tabla 78. ***La entidad externa cámara niega potencialmente la recepción de datos.***

Category:	Repudiation

Description:	Camera claims that it did not receive data from a process on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.
Justification:	CAPEC 132
Dread-Damage (D):	Medio (2)
Dread-Reproducibility (R):	Alto (3)
Dread-Exploitability (E):	Medio (2)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	30
Safeguard 1:	Utilizar medios de registro de la fuente, hora y un resumen de los datos obtenidos
Safeguard 2:	
Safeguard 3:	

Fuente: Elaboración propia.

53. Data Flow TCP Is Potentially Interrupted [State: Not Started] [Priority: Low]

Tabla 79. ***El flujo de datos TCP está potencialmente interrumpido.***

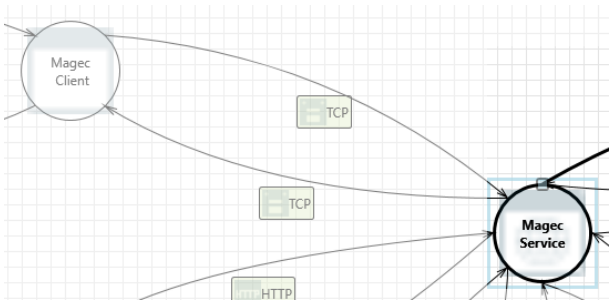
Category:	Denial Of Service
Description:	An external agent interrupts data flowing across a trust boundary in either direction.
Justification:	CAPEC 276
Dread-Damage (D):	Alto (3)
Dread-Reproducibility (R):	Bajo (1)
Dread-Exploitability (E):	Bajo (1)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	18
Safeguard 1:	Gestionar el estado del flujo de datos del software

Safeguard 2:	Gestionar correctamente el estado de los usuarios para no sufrir ataques
Safeguard 3:	

Fuente: Elaboración propia.

W) Interaction: TCP

Figura 76. *Iteración TCP Magec Client – Magec Service.*



Fuente: Elaboración propia.

54. Magec Service Process Memory Tampered [State: Not Started] [Priority: Medium]

Tabla 80. ***Memoria del proceso Magec Service alterada.***

Category:	Tampering
Description:	If Magec Service is given access to memory, such as shared memory or pointers, or is given the ability to control what Magec Client executes (for example, passing back a function pointer.), then Magec Service can tamper with Magec Client. Consider if the function could work with less access to memory, such as passing data rather than pointers. Copy in data provided, and then validate it.
Justification:	CAPEC 129
Dread-Damage (D):	Medio (2)
Dread-Reproducibility (R):	Medio (2)
Dread-Exploitability (E):	Medio (2)
Dread-Affected users (A):	Medio (2)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	20

Safeguard 1:	Evitar el uso de acceso a memoria excesivo
Safeguard 2:	Copiar los datos que se proporcionan y validarlos
Safeguard 3:	

Fuente: Elaboración propia.

55. Elevation Using Impersonation [State: Not Started] [Priority: Low]

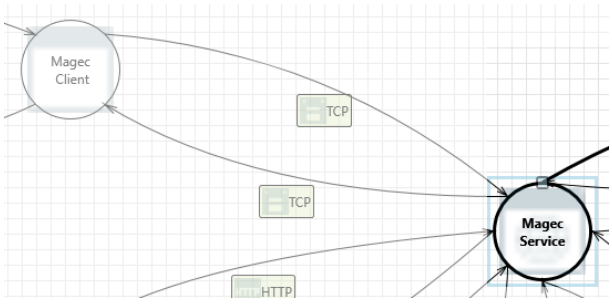
Tabla 81. ***Elevación usando suplantación.***

Category:	Elevation Of Privilege
Description:	Magec Client may be able to impersonate the context of Magec Service in order to gain additional privilege.
Justification:	CAPEC 233
Dread-Damage (D):	Medio (2)
Dread-Reproducibility (R):	Medio (2)
Dread-Exploitability (E):	Medio (2)
Dread-Affected users (A):	Bajo (1)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	15
Safeguard 1:	Ejecutar las aplicaciones con menos privilegios
Safeguard 2:	Realizar una autenticación de los usuarios para acceder a la aplicación
Safeguard 3:	

Fuente: Elaboración propia.

X) Interaction: TCP

Figura 77. *Iteración TCP Magec Client – Magec Service.*



Fuente: Elaboración propia.

56. Magec Client Process Memory Tampered [State: Not Started] [Priority: Medium]

Tabla 82. ***Memoria de proceso Magec Client alterada.***

Category:	Tampering
Description:	If Magec Client is given access to memory, such as shared memory or pointers, or is given the ability to control what Magec Service executes (for example, passing back a function pointer.), then Magec Client can tamper with Magec Service. Consider if the function could work with less access to memory, such as passing data rather than pointers. Copy in data provided, and then validate it.
Justification:	CAPEC 129
Dread-Damage (D):	Medio (2)
Dread-Reproducibility (R):	Medio (2)
Dread-Exploitability (E):	Medio (2)
Dread-Affected users (A):	Medio (2)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	20
Safeguard 1:	Evitar el uso de acceso a memoria excesivo
Safeguard 2:	Copiar los datos que se proporcionan y validarlos
Safeguard 3:	

Fuente: Elaboración propia.

57. Elevation Using Impersonation [State: Not Started] [Priority: Low]

Tabla 83. ***Elevación usando suplantación.***

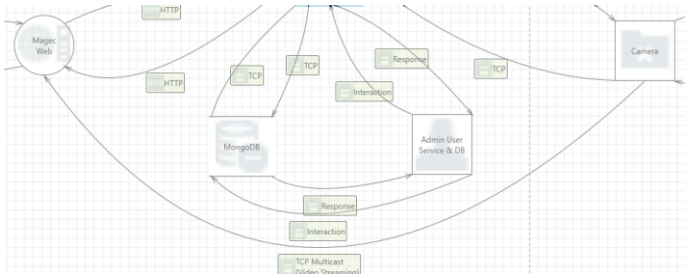
Category:	Elevation Of Privilege
Description:	Magec Service may be able to impersonate the context of Magec Client in order to gain additional privilege.
Justification:	CAPEC 233
Dread-Damage (D):	Medio (2)

Dread-Reproducibility (R):	Medio (2)
Dread-Exploitability (E):	Medio (2)
Dread-Affected users (A):	Bajo (1)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	15
Safeguard 1:	Ejecutar las aplicaciones con menos privilegios
Safeguard 2:	Realizar una autenticación de los usuarios para acceder a la aplicación
Safeguard 3:	

Fuente: Elaboración propia.

Y) Interaction: TCP Multicast (Video Streaming)

Figura 78. *Iteración TCP Multicast Magec Web – Camera.*



Fuente: Elaboración propia.

58. Spoofing the Magec Web Process [State: Not Started] [Priority: High]

Tabla 84. ***Falsificación del proceso web de Magec.***

Category:	Spoofing
Description:	Magec Web may be spoofed by an attacker and this may lead to information disclosure by Camera. Consider using a standard authentication mechanism to identify the destination process.
Justification:	CAPEC 154
Dread-Damage (D):	Alto (3)
Dread-Reproducibility (R):	Alto (3)
Dread-Exploitability (E):	Alto (3)

Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Medio (2)
Riesgo = (R+E+DI) x (D+A) = PxI=:	48
Safeguard 1:	Supervisar la actividad en la red para detectar cualquier cambio en la comunicación (anómalo o no autorizado)
Safeguard 2:	
Safeguard 3:	

Fuente: Elaboración propia.

59. Spoofing the Camera External Entity [State: Not Started] [Priority: High]

Tabla 85. ***Falsificación de la entidad externa cámara.***

Category:	Spoofing
Description:	Camera may be spoofed by an attacker and this may lead to unauthorized access to Magec Web. Consider using a standard authentication mechanism to identify the external entity.
Justification:	CAPEC 154
Dread-Damage (D):	Alto (3)
Dread-Reproducibility (R):	Alto (3)
Dread-Exploitability (E):	Alto (3)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Medio (2)
Riesgo = (R+E+DI) x (D+A) = PxI=:	48
Safeguard 1:	Supervisar la actividad en la red para detectar cualquier cambio en la comunicación (anómalo o no autorizado)
Safeguard 2:	
Safeguard 3:	

Fuente: Elaboración propia.

60. Potential Lack of Input Validation for Magec Web [State: Not Started] [Priority: Medium]

Tabla 86. ***Falta potencial de validación de entrada para Magec Web.***

Category:	Tampering
Description:	Data flowing across TCP Multicast (Video Streaming) may be tampered with by an attacker. This may lead to a denial of service attack against Magec Web or an elevation of privilege attack against Magec Web or an information disclosure by Magec Web. Failure to verify that input is as expected is a root cause of a very large number of exploitable issues. Consider all paths and the way they handle data. Verify that all input is verified for correctness using an approved list input validation approach.
Justification:	CAPEC 153
Dread-Damage (D):	Alto (3)
Dread-Reproducibility (R):	Medio (2)
Dread-Exploitability (E):	Medio (2)
Dread-Affected users (A):	Medio (2)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	25
Safeguard 1:	Validar las entradas controlando el formato, estructura y composición
Safeguard 2:	
Safeguard 3:	

Fuente: Elaboración propia.

61. Potential Data Repudiation by Magec Web [State: Not Started] [Priority: Medium]

Tabla 87. ***Posible repudio de datos por parte de Magec Web.***

Category:	Repudiation
Description:	Magec Web claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.
Justification:	CAPEC 132
Dread-Damage (D):	Medio (2)
Dread-Reproducibility (R):	Alto (3)

Dread-Exploitability (E):	Medio (2)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	30
Safeguard 1:	Utilizar medios de registro de la fuente, hora y un resumen de los datos obtenidos
Safeguard 2:	
Safeguard 3:	

Fuente: Elaboración propia.

62. Data Flow Sniffing [State: Not Started] [Priority: Low]

Tabla 88. ***Rastreo de flujo de datos.***

Category:	Information Disclosure
Description:	Data flowing across TCP Multicast (Video Streaming) may be sniffed by an attacker. Depending on what type of data an attacker can read, it may be used to attack other parts of the system or simply be a disclosure of information leading to compliance violations. Consider encrypting the data flow.
Justification:	CAPEC 65
Dread-Damage (D):	Medio (2)
Dread-Reproducibility (R):	Bajo (1)
Dread-Exploitability (E):	Bajo (1)
Dread-Affected users (A):	Medio (2)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	12
Safeguard 1:	Cifrar todas las comunicaciones entre cliente y servidor
Safeguard 2:	Usar SSL, SSH o SCP
Safeguard 3:	Utilizar ipconfig para detectar el sniffer que esté instalado en la red

Fuente: Elaboración propia.

63. Potential Process Crash or Stop for Magec Web [State: Not Started] [Priority: Medium]

Tabla 89. ***Posible bloqueo o detención del proceso para Magec Web.***

Category:	Denial Of Service
Description:	Magec Web crashes, halts, stops or runs slowly; in all cases violating an availability metric.
Justification:	CAPEC 74
Dread-Damage (D):	Alto (3)
Dread-Reproducibility (R):	Bajo (1)
Dread-Exploitability (E):	Medio (2)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Medio (2)
Riesgo = (R+E+DI) x (D+A) = PxI=:	30
Safeguard 1:	No confiar solamente en ubicaciones controlables por el usuario para mantener el estado del usuario
Safeguard 2:	Evitar la información confidencial en ubicaciones controlables por el usuario
Safeguard 3:	La información sensible que forma parte del estado del usuario debe protegerse adecuadamente para garantizar la confidencialidad e integridad en cada solicitud

Fuente: Elaboración propia.

64. Data Flow TCP Multicast (Video Streaming) Is Potentially Interrupted [State: Not Started] [Priority: Low]

Tabla 90. ***El flujo de datos de multicast TCP (transmisión de video) está potencialmente interrumpido.***

Category:	Denial Of Service
Description:	An external agent interrupts data flowing across a trust boundary in either direction.
Justification:	CAPEC 276
Dread-Damage (D):	Alto (3)

Dread-Reproducibility (R):	Bajo (1)
Dread-Exploitability (E):	Bajo (1)
Dread-Affected users (A):	Alto (3)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	18
Safeguard 1:	Gestionar el estado del flujo de datos del software
Safeguard 2:	Gestionar correctamente el estado de los usuarios para no sufrir ataques
Safeguard 3:	

Fuente: Elaboración propia.

65. Elevation Using Impersonation [State: Not Started] [Priority: Low]

Tabla 91. ***Elevación usando suplantación.***

Category:	Elevation Of Privilege
Description:	Magec Web may be able to impersonate the context of Camera in order to gain additional privilege.
Justification:	CAPEC 233
Dread-Damage (D):	Medio (2)
Dread-Reproducibility (R):	Medio (2)
Dread-Exploitability (E):	Medio (2)
Dread-Affected users (A):	Bajo (1)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	15
Safeguard 1:	Ejecutar las aplicaciones con menos privilegios
Safeguard 2:	Realizar una autenticación de los usuarios para acceder a la aplicación
Safeguard 3:	

Fuente: Elaboración propia.

66. Magec Web May be Subject to Elevation of Privilege Using Remote Code Execution [State: Not Started] [Priority: Low]

Tabla 92. ***Magec Web puede estar sujeto a la elevación de privilegios mediante la ejecución remota de código.***

Category:	Elevation Of Privilege
Description:	Camera may be able to remotely execute code for Magec Web.
Justification:	CAPEC 233
Dread-Damage (D):	Medio (2)
Dread-Reproducibility (R):	Medio (2)
Dread-Exploitability (E):	Medio (2)
Dread-Affected users (A):	Bajo (1)
Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	15
Safeguard 1:	Ejecutar las aplicaciones con menos privilegios
Safeguard 2:	Realizar una autenticación de los usuarios para acceder a la aplicación
Safeguard 3:	

Fuente: Elaboración propia.

67. Elevation by Changing the Execution Flow in Magec Web [State: Not Started] [Priority: Low]

Tabla 93. ***Elevación cambiando el flujo de ejecución en Magec Web.***

Category:	Elevation Of Privilege
Description:	An attacker may pass data into Magec Web in order to change the flow of program execution within Magec Web to the attacker's choosing.
Justification:	CAPEC 233
Dread-Damage (D):	Medio (2)
Dread-Reproducibility (R):	Medio (2)
Dread-Exploitability (E):	Medio (2)
Dread-Affected users (A):	Bajo (1)

Dread-Discoverablity (DI):	Bajo (1)
Riesgo = (R+E+DI) x (D+A) = PxI=:	15
Safeguard 1:	Ejecutar las aplicaciones con menos privilegios
Safeguard 2:	Realizar una autenticación de los usuarios para acceder a la aplicación
Safeguard 3:	

Fuente: Elaboración propia.

Anexo B. Informes de Fortify SCA

Análisis detallados de cada Hallazgos de las soluciones Core y Framework de la herramienta Fortify SCA, una de las peores de densidad de vulnerabilidades y una de las que contienen vulnerabilidades de severidad Critical. Al ser tan extensos los informes, solo se adjuntarán estas soluciones.

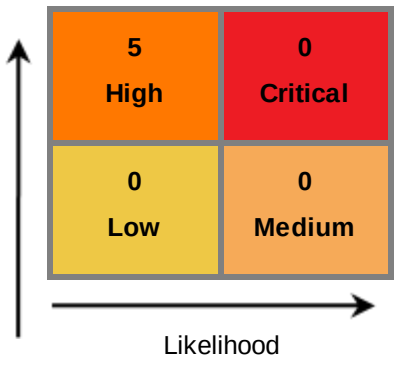
Executive Summary

This workbook is intended to provide all necessary details and information for a developer to understand and remediate the different issues discovered during the Core project audit. The information contained in this workbook is targeted at project managers and developers.

This section provides an overview of the issues uncovered during analysis.

Project Name:	Core
Project Version:	
SCA:	Results Present
WebInspect:	Results Not Present
WebInspect Agent:	Results Not Present
Other:	Results Not Present

Issues by Priority



Top Ten Critical Categories

This project does not contain any critical issues

Project Description

This section provides an overview of the Fortify scan engines used for this project, as well as the project meta-information.

SCA

Date of Last Analysis:	Nov 29, 2021, 2:35 PM	**Engine Version:**	20.2.0.0139
Host Name:	DESKTOP	**Certification:**	VALID
Number of Files:	94	**Lines of Code:**	1,128

Rulepack Name	Rulepack Version
Reglas de codificación segura de Fortify, Núcleo, .NET	2021.2.1.0001
Reglas de codificación segura de Fortify, Extendido, Configuración	2021.2.1.0001
Reglas de codificación segura de Fortify, Extendido, Contenido	2021.2.1.0001

Reglas de codificación segura de Fortify, Extendido, .NET 2021.2.1.0001

Issue Breakdown by Fortify Categories

The following table depicts a summary of all issues grouped vertically by Fortify Category. For each category, the total number of issues is shown by Fortify Priority Order, including information about the number of audited issues.

Tabla 94. *Vulnerabilidades de la solución Core.*

Category	Fortify Priority (audited/total)				Total Issues
	Critical	High	Medium	Low	
Insecure Randomness	0	1 / 1	0	0	1 / 1
Null Dereference	0	4 / 4	0	0	4 / 4

Fuente: Elaboración propia.

Results Outline

Insecure Randomness (1 issue)

Abstract

Los generadores de números pseudoaleatorios estándar no pueden soportar ataques criptográficos.

Explanation

Cuando se utiliza una función que puede producir valores predecibles como un origen de aleatoriedad en un contexto de seguridad, se producen errores de aleatoriedad no segura. Los equipos son máquinas deterministas y como tales no pueden producir una aleatoriedad auténtica. Los generadores de números pseudoaleatorios (PRNG, Pseudorandom Number Generator) aproximan la aleatoriedad de forma algorítmica, a partir de un valor de inicialización desde el que se calculan los valores siguientes. Existen dos tipos de PRNG: estadísticos y criptográficos. Los PRNG estadísticos proporcionan propiedades estadísticas útiles, pero su resultado es altamente predecible y crea una secuencia numérica fácil de reproducir que no es adecuada para aquellos casos en los que la seguridad dependa de que los valores generados sean impredecibles. Los PRNG criptográficos resuelven este problema mediante la generación de resultados que sean más difíciles de predecir. Para que un valor sea criptográficamente seguro, debe ser imposible o altamente improbable para un usuario malintencionado distinguir entre el valor aleatorio generado y un valor verdaderamente aleatorio. Por lo general, si un algoritmo PRNG no se anuncia como criptográficamente seguro, es probable que se trate de un PRNG estadístico y no debería usarse en contextos relativos a la seguridad en los que su uso podría provocar vulnerabilidades graves, como contraseñas temporales fáciles de adivinar, claves criptográficas predecibles, secuestro de sesiones y suplantación de identidad DNS. **Ejemplo:** el siguiente código utiliza un PRNG estadístico para crear una dirección URL de un recibo que permanece activo durante un período de tiempo después de una compra.

```
string GenerateReceiptURL(string baseUrl) {
    Random Gen = new Random();
    return (baseUrl + Gen.Next().toString() + ".html");
}
```

Este código utiliza la función `Random.Next()` para generar identificadores "únicos" para las páginas de recibos que crea. Como `Random.Next()` es un PRNG estadístico, es fácil que un usuario malintencionado adivine las cadenas que genera. Aunque el diseño subyacente del sistema de recepción también es defectuoso, sería más seguro si utilizara un generador de números aleatorios que no generase identificadores de recibos predecibles, como un PRNG criptográfico.

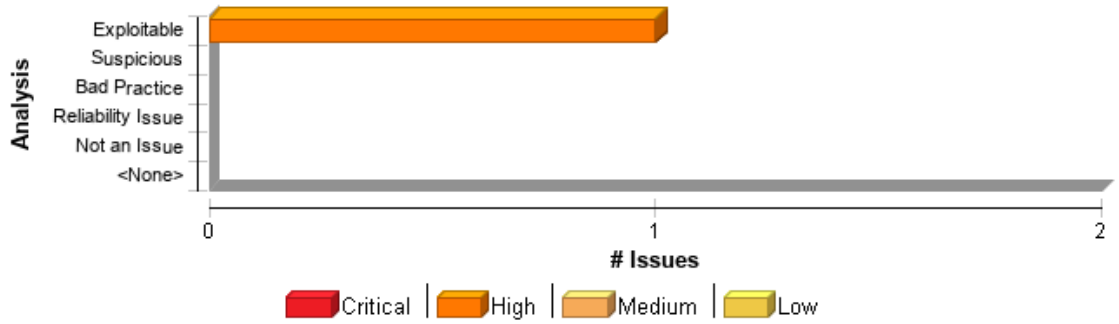
Recommendation

Cuando la imprevisibilidad sea fundamental, como es el caso de la mayoría de los usos de seguridad de aleatoriedad, utilice un PRNG criptográfico. Independientemente del PRNG que elija, utilice siempre un valor con suficiente entropía para inicializar el algoritmo. (No utilice valores tales como la hora actual, dado que ofrecen únicamente una entropía insignificante). .NET Framework proporciona un PRNG criptográfico en `System.Security.Cryptography.RandomNumberGenerator`. Como sucede con otras clases basadas en algoritmo en `System.Security`, `RandomNumberGenerator` proporciona un contenedor de implementación independiente alrededor de un determinado conjunto de algoritmos. Al solicitar una instancia de un objeto `RandomNumberGenerator` mediante `RandomNumberGenerator.Create()`, puede solicitar una implementación específica del algoritmo. Si el algoritmo está disponible, entonces se ofrece como un objeto `RandomNumberGenerator`. Si no está disponible o no se especifica una implementación concreta, entonces se le ofrecerá una implementación `RandomNumberGenerator` que haya seleccionado el sistema. Microsoft proporciona una única implementación de `RandomNumberGenerator` con la instancia de .NET Framework denominada `RNGCryptoServiceProvider`, que Microsoft describe de la siguiente forma: "Para formar la inicialización del generador de números aleatorios, una aplicación de llamada proporciona bits que pueda tener (por ejemplo, entrada de temporización de ratón o teclado) y que, a continuación, se añaden tanto a la inicialización almacenada como a diversos datos del usuario y del sistema como, por ejemplo, los ID de proceso y subproceso, la hora y el contador del sistema, el estado de la memoria, los clústeres de disco libres y el bloque de entorno de usuario con el algoritmo hash. A este resultado se le aplica el algoritmo hash SHA-1 y la salida se utiliza para inicializar una secuencia RC4, que a su vez se usa como secuencia aleatoria y para actualizar la inicialización almacenada". Sin embargo, los aspectos concretos de la implementación de Microsoft del algoritmo `RNGCryptoServiceProvider` están documentados de forma vaga, y no queda claro qué orígenes de entropía emplea la implementación y en qué circunstancias, y, por lo tanto, qué cantidad de auténtica aleatoriedad hay en su salida. Aunque hay cierta especulación en Internet acerca de la implementación de Microsoft, no hay pruebas que contradigan la reclamación de

que el algoritmo es seguro desde la perspectiva criptográfica y puede utilizarse sin problemas en contextos basados en la seguridad.

Issue Summary

Figura 79. *Gráfico de vulnerabilidades de categoría Insecure Randomness.*



Fuente: Elaboración propia.

Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Insecure Randomness	1	0	0	1
Total	**1**	**0**	**0**	**1**

Insecure Randomness	**High**
Package: Core	
ConfigItem.cs, line 328 (Insecure Randomness)	**High**
Issue Details	
Kingdom: Security Features	
Scan Engine: SCA (Semantic)	
Audit Details	
Analysis	Exploitable
Audit Comments	
MAS: Sat Dec 11 2021 14:15:02 GMT+0100 (CET)	
Usar la libreria RandomNumberGenerator que es segura en vez de Random.Next	
Sink Details	
Sink: Next()	
Enclosing Method: SaveMark()	
File: ConfigItem.cs:328	
Taint Flags:	

```
325
326  private void SaveMark ( BinaryWriter writer )
327  {
328  Int32 data1 = random.Next ();
329  Int32 data2 = data1 ^ CheckValue;
330  writer.Write ( data1 );
331  writer.Write ( data2 );
```

Null Dereference (4 issues)

Abstract

El programa puede eliminar potencialmente la referencia de un puntero nulo, generando una `NullException`.

Explanation

Los punteros nulos suelen producirse debido a que se ha infringido alguna presuposición del programador. La mayoría de los problemas con el puntero nulo provocan problemas generales de confiabilidad de software. Sin embargo, si un atacante puede desencadenar intencionadamente la eliminación de la referencia del puntero nulo, es posible que también pueda usar la excepción resultante para omitir la lógica de seguridad o para hacer que la aplicación revele información de depuración, la cual será valiosa para la planificación de ataques posteriores. **Ejemplo 1:** en el siguiente código, el programador presupone que el sistema tiene siempre definida una propiedad denominada "`cmd`". Si un usuario malintencionado puede controlar el entorno del programa para que no se defina "`cmd`", el programa genera una excepción de puntero nulo al intentar llamar al método `Trim()`.

```
string cmd = null;
...
cmd = Environment.GetEnvironmentVariable("cmd");
cmd = cmd.Trim();
```

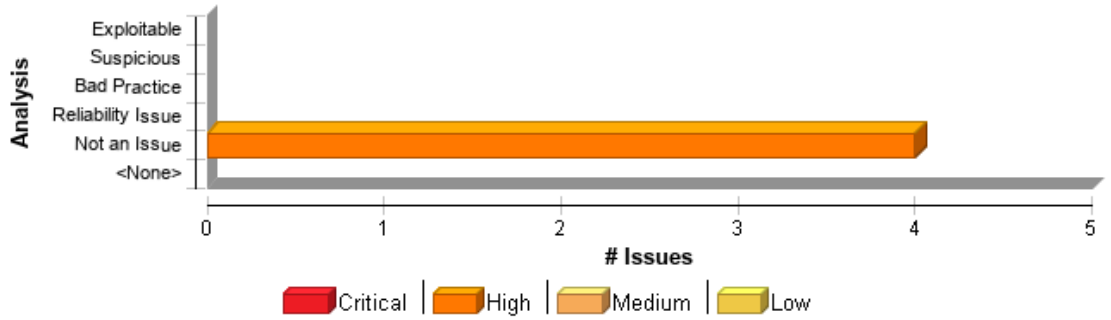
Recommendation

Los problemas de seguridad provocados por la eliminación de las referencias de punteros nulos están casi siempre relacionados con la forma en que el programa administra las excepciones de tiempo de ejecución. Si el software cuenta con un enfoque sólido y bien ejecutado en cuanto a la administración

de las excepciones de tiempo de ejecución, se reducirá considerablemente la posibilidad de que se produzcan daños de seguridad.

Issue Summary

Figura 80. *Gráfico de vulnerabilidades de categoría Null Dereference.*



Fuente: Elaboración propia.

Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Null Dereference	4	0	0	4
Total	**4**	**0**	**0**	**4**

Null Dereference — **High**

Package: Core.ObservableList@1

ObservableList.cs, line 161 (Null Dereference) — **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Sat Dec 11 2021 14:20:39 GMT+0100 (CET)

No aplica porque se comprueba que subscriber nunca es null

Sink Details

Sink: .ListCleared(...) : is not checked for null value before being dereferenced

Enclosing Method: Invoke()

File: ObservableList.cs:161

Taint Flags:

```
158  {
159  try
160  {
161  (s as IObservableListSubscriber<T>).ListCleared (this);
162  }
163  catch (Exception e)
164  {
```

ObservableList.cs, line 137 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Sat Dec 11 2021 14:20:34 GMT+0100 (CET)

No aplica porque se comprueba que subscriber nunca es null

Sink Details

Sink: .ListItemChanged(...) : is not checked for null value before being dereferenced

Enclosing Method: Invoke()

File: ObservableList.cs:137

Taint Flags:

```
134  {
135  try
136  {
137  (s as IObservableListSubscriber<T>).ListItemChanged (this, index);
138  }
139  catch (Exception e)
140  {
```

ObservableList.cs, line 113 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Sat Dec 11 2021 14:20:29 GMT+0100 (CET)

No aplica porque se comprueba que subscriber nunca es null

Sink Details

Sink: .ListItemInserted(...) : is not checked for null value before being dereferenced

Enclosing Method: Invoke()

File: ObservableList.cs:113

Taint Flags:

```
110  {
111  try
112  {
113  (s as IObservableListSubscriber<T>).ListItemInserted (this, index);
114  }
115  catch (Exception e)
116  {
```

ObservableList.cs, line 89 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Sat Dec 11 2021 14:20:22 GMT+0100 (CET)

No aplica porque se comprueba que subscriber nunca es null

Sink Details

Sink: .ListItemDeleted(...) : is not checked for null value before being dereferenced

Enclosing Method: Invoke()

File: ObservableList.cs:89

Taint Flags:

```
86  {
87  try
88  {
```

```
89   (s as IObservableListSubscriber<T>).ListItemDeleted (this, index,
item);
90   }
91   catch ( Exception e )
92   {
```

Executive Summary

This workbook is intended to provide all necessary details and information for a developer to understand and remediate the different issues discovered during the Framework project audit. The information contained in this workbook is targeted at project managers and developers.

This section provides an overview of the issues uncovered during analysis.

Project Name:	Framework
Project Version:	
SCA:	Results Present
WebInspect:	Results Not Present
WebInspect Agent:	Results Not Present
Other:	Results Not Present

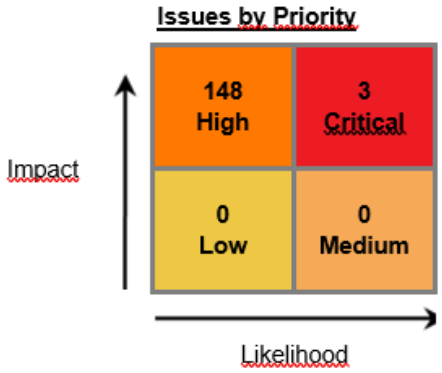
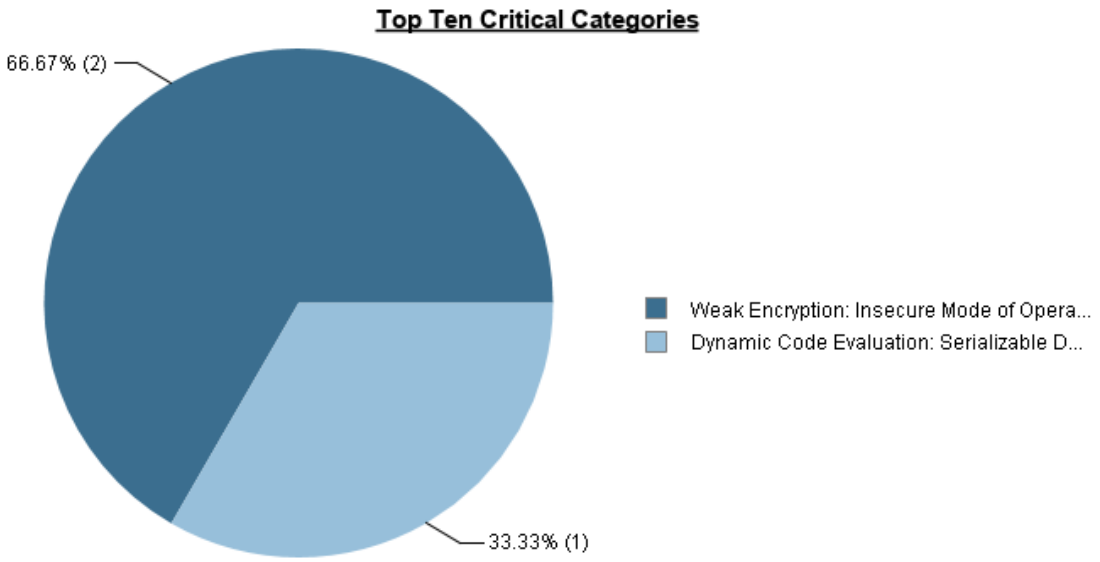


Figura 81. *Top 10 Categorías Criticas Framework.*



Fuente: Elaboración propia.

Project Description

This section provides an overview of the Fortify scan engines used for this project, as well as the project meta-information.

SCA

Date of Last Analysis:	Nov 29, 2021, 2:02 PM	**Engine Version:**	20.2.0.0139
Host Name:	DESKTOP	**Certification:**	VALID
Number of Files:	554	**Lines of Code:**	41,201

Rulepack Name	**Rulepack Version**
Reglas de codificación segura de Fortify, Núcleo, .NET	2021.2.1.0001
Reglas de codificación segura de Fortify, Extendido, Configuración	2021.2.1.0001
Reglas de codificación segura de Fortify, Extendido, Contenido	2021.2.1.0001
Reglas de codificación segura de Fortify, Extendido, .NET	2021.2.1.0001

Issue Breakdown by Fortify Categories

The following table depicts a summary of all issues grouped vertically by Fortify Category. For each category, the total number of issues is shown by Fortify Priority Order, including information about the number of audited issues.

Tabla 95. ***Vulnerabilidades de la solución Framework.***

Category	Fortify Priority (audited/total)				Total Issues
	Critical	High	Medium	Low	
Dynamic Code Evaluation: Serializable Delegate	1 / 1	0			1 / 1
Insecure Randomness	0	33 / 33	0	0	33 / 33
Missing XML Validation	0	3 / 3			3 / 3
Null Dereference	0	47 / 47	0	0	47 / 47
Often Misused: Authentication	0	1 / 1			1 / 1
Password Management: Hardcoded Password	0	1 / 1	0	0	1 / 1
Path Manipulation	0	17 / 17			17 / 17
Path Manipulation: Base Path Overwriting	0	3 / 3	0	0	3 / 3
Portability Flaw: File Separator	0	1 / 1			1 / 1
Privacy Violation: Heap Inspection	0	7 / 7	0	0	7 / 7
Process Control	0	5 / 5			5 / 5
Unreleased Resource: LDAP	0	2 / 2	0	0	2 / 2
Unreleased Resource: Streams	0	4 / 4			4 / 4
Unsafe Native Invoke	0	9 / 9	0	0	9 / 9
Weak Encryption: Insecure Mode of Operation	2 / 2	0			2 / 2
XML External Entity Injection	0	15 / 15	0	0	15 / 15

Fuente: Elaboración propia.

Results Outline

Dynamic Code Evaluation: Serializable Delegate (1 issue)

Abstract

El campo serializable `Delegate` en una clase determinada puede introducir una vulnerabilidad de ejecución de código arbitrario durante la serialización de dicha clase, o después de ella.

Explanation

Delegate se usa para hacer referencia a una llamada de método que se puede invocar más tarde en el código de usuario. .NET usa la serialización personalizada mientras serializa los tipos `Delegate` y utiliza la clase `System.DelegateSerializationHolder` para almacenar la información del método que se adjunta o se suscribe a `Delegate`. La secuencia serializada del objeto `Delegate` no es adecuada para el almacenamiento persistente ni para transferirla a la aplicación remota debido a que un atacante podría sustituir la información del método con información que apuntase a gráficos de objeto maliciosos, lo que podría causar una vulnerabilidad de ejecución de código arbitrario cuando se invocase durante la serialización, o después de ella. **Ejemplo 1:** La siguiente clase contiene un campo serializable `Delegate` y se invoca en el método `Executor`:

```
...
[Serializable]
class DynamicRunnner
{
    Delegate _del;
    string[] _arg;
    public DynamicRunnner(Delegate dval, params string[] arg)
    {
        _del = dval;
        _arg = arg;
    }
    public bool Executor()
    {
        return (bool)_del.DynamicInvoke(_arg);
    }
}
...
```

Si el desarrollador deserializa una secuencia que no es de confianza de la clase mencionada en el `Example 1`, un atacante podría sustituir la información del método con información que apuntase a `Process.Start`, con lo que se crearía un proceso arbitrario cuando se llamase al método `Executor`.

Recommendation

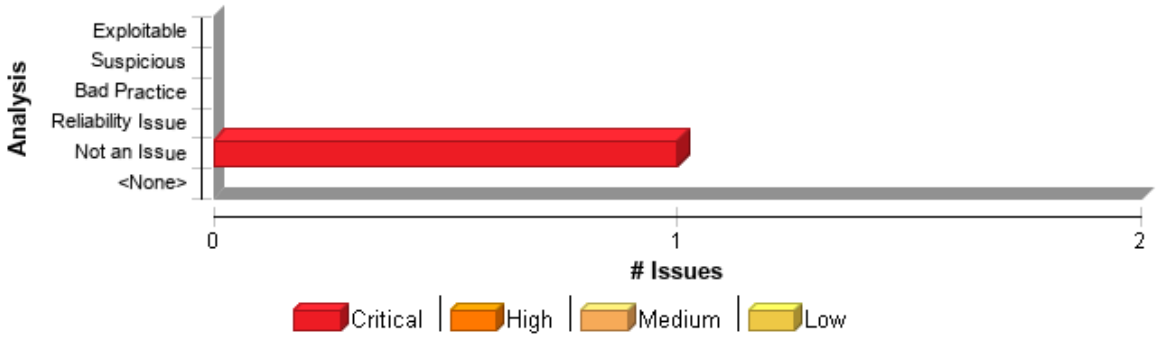
Esta vulnerabilidad se puede solucionar marcando el campo `Delegate` mediante el atributo `[NonSerialized]` tal como se muestra en el `Example 2`, lo que evitará la ejecución de código arbitrario tras la deserialización. **Ejemplo 2:**

```
...
[Serializable]
class DynamicRunnner
{
    [field: NonSerialized]
    Delegate _del;
    string[] _arg;
    public DynamicRunnner(Delegate dval, params string[] arg)
    {
        _del = dval;
        _arg = arg;
    }
    public bool Run()
    {
        return (bool)_del.DynamicInvoke(_arg);
    }
}
...
```

Además, no deserialice datos que no son de confianza sin validar los contenidos de la secuencia de objetos.

Issue Summary

Figura 82. *Gráfico de vulnerabilidades de categoría Dynamic Code Evaluation: Serializable Delegate.*



Fuente: Elaboración propia.

Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Dynamic Code Evaluation: Serializable Delegate	1	0	0	1
Total	**1**	**0**	**0**	**1**

Dynamic Code Evaluation: Serializable Delegate	**Critical**
Package: Framework.Localization	
guageTextType.cs, line 413 (Dynamic Code Evaluation: Serializable)	**Critical**
Issue Details	

Kingdom: Input Validation and Representation

Scan Engine: SCA (Structural)

Audit Details	
Analysis	Not an Issue

Audit Comments

MAS: Fri Dec 31 2021 13:00:43 GMT+0100 (CET)

Solo mira cambios en el objeto, no serializa nada

Sink Details

Sink: Field: PropertyChanged

File: MultiLanguageTextType.cs:413

Taint Flags:

```
No snippet available
```

Insecure Randomness (33 issues)

Abstract

Los generadores de números pseudoaleatorios estándar no pueden soportar ataques criptográficos.

Explanation

Cuando se utiliza una función que puede producir valores predecibles como un origen de aleatoriedad en un contexto de seguridad, se producen errores de aleatoriedad no segura. Los equipos son máquinas deterministas y como tales no pueden producir una aleatoriedad auténtica. Los generadores de números pseudoaleatorios (PRNG, Pseudorandom Number Generator) aproximan la aleatoriedad de forma algorítmica, a partir de un valor de inicialización desde el que se calculan los valores siguientes. Existen dos tipos de PRNG: estadísticos y criptográficos. Los PRNG estadísticos proporcionan propiedades estadísticas útiles, pero su resultado es altamente predecible y crea una secuencia numérica fácil de reproducir que no es adecuada para aquellos casos en los que la seguridad dependa de que los valores generados sean impredecibles. Los PRNG criptográficos resuelven este problema mediante la generación de resultados que sean más difíciles de predecir. Para que un valor sea criptográficamente seguro, debe ser imposible o altamente improbable para un usuario

malintencionado distinguir entre el valor aleatorio generado y un valor verdaderamente aleatorio. Por lo general, si un algoritmo PRNG no se anuncia como criptográficamente seguro, es probable que se trate de un PRNG estadístico y no debería usarse en contextos relativos a la seguridad en los que su uso podría provocar vulnerabilidades graves, como contraseñas temporales fáciles de adivinar, claves criptográficas predecibles, secuestro de sesiones y suplantación de identidad DNS. **Ejemplo:** el siguiente código utiliza un PRNG estadístico para crear una dirección URL de un recibo que permanece activo durante un período de tiempo después de una compra.

```
string GenerateReceiptURL(string baseUrl) {
    Random Gen = new Random();
    return (baseUrl + Gen.Next().toString() + ".html");
}
```

Este código utiliza la función `Random.Next()` para generar identificadores "únicos" para las páginas de recibos que crea. Como `Random.Next()` es un PRNG estadístico, es fácil que un usuario malintencionado adivine las cadenas que genera. Aunque el diseño subyacente del sistema de recepción también es defectuoso, sería más seguro si utilizara un generador de números aleatorios que no generase identificadores de recibos predecibles, como un PRNG criptográfico.

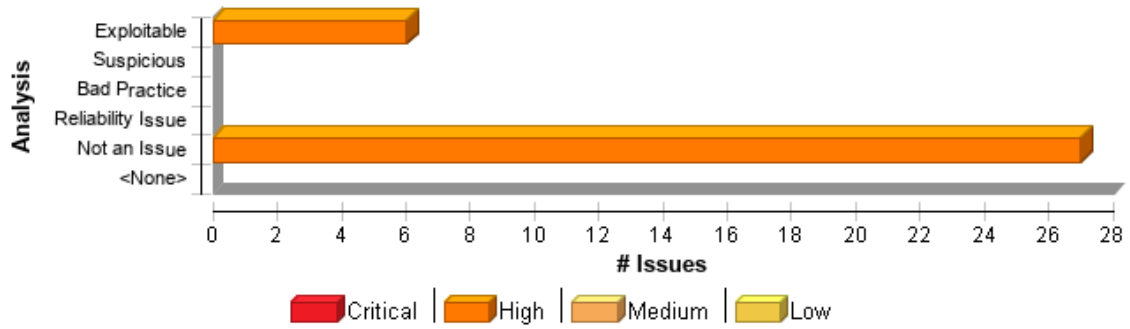
Recommendation

Cuando la imprevisibilidad sea fundamental, como es el caso de la mayoría de los usos de seguridad de aleatoriedad, utilice un PRNG criptográfico. Independientemente del PRNG que elija, utilice siempre un valor con suficiente entropía para inicializar el algoritmo. (No utilice valores tales como la hora actual, dado que ofrecen únicamente una entropía insignificante). .NET Framework proporciona un PRNG criptográfico en `System.Security.Cryptography.RandomNumberGenerator`. Como sucede con otras clases basadas en algoritmo en `System.Security`, `RandomNumberGenerator` proporciona un contenedor de implementación independiente alrededor de un determinado conjunto de algoritmos. Al solicitar una instancia de un objeto `RandomNumberGenerator` mediante `RandomNumberGenerator.Create()`, puede solicitar una implementación específica del algoritmo. Si el algoritmo está disponible, entonces se ofrece como un objeto `RandomNumberGenerator`. Si no está disponible o no se especifica una implementación concreta, entonces se le ofrecerá una implementación `RandomNumberGenerator` que haya seleccionado el sistema. Microsoft proporciona una única implementación de `RandomNumberGenerator` con la instancia de .NET Framework denominada `RNGCryptoServiceProvider`, que Microsoft describe de la siguiente forma: "Para formar la inicialización del generador de números aleatorios, una aplicación de llamada proporciona bits que pueda tener (por ejemplo, entrada de temporización de ratón o teclado) y que, a continuación, se añaden tanto a la inicialización almacenada como a diversos datos del usuario y del sistema como, por ejemplo, los ID de proceso y subproceso, la hora y el contador del sistema, el estado de la memoria, los clústeres de disco libres y el bloque de entorno de usuario con el algoritmo hash. A este resultado se le aplica el algoritmo hash SHA-1 y la salida se utiliza para inicializar una secuencia RC4, que a su vez se usa como secuencia aleatoria y para actualizar la inicialización almacenada". Sin embargo, los

aspectos concretos de la implementación de Microsoft del algoritmo `RNGCryptoServiceProvider` están documentados de forma vaga, y no queda claro qué orígenes de entropía emplea la implementación y en qué circunstancias, y, por lo tanto, qué cantidad de auténtica aleatoriedad hay en su salida. Aunque hay cierta especulación en Internet acerca de la implementación de Microsoft, no hay pruebas que contradigan la reclamación de que el algoritmo es seguro desde la perspectiva criptográfica y puede utilizarse sin problemas en contextos basados en la seguridad.

Issue Summary

Figura 83. *Gráfico de vulnerabilidades de categoría Insecure Randomness.*



Fuente: Elaboración propia.

Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Insecure Randomness	33	0	0	33
Total	**33**	**0**	**0**	**33**

Insecure Randomness	**High**
Package: CommsLinkTest	
TestPipeCommsLink.cs, line 246 (Insecure Randomness)	**High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis	Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 21:01:34 GMT+0100 (CET)

Es un test, no es explotable

Sink Details

Sink: Next()

Enclosing Method: get_PipeRandomName()

File: TestPipeCommsLink.cs:246

Taint Flags:

```
No snippet available
```

TestBridge.cs, line 149 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 21:01:25 GMT+0100 (CET)

Es un test, no es explotable

Sink Details

Sink: Next()

Enclosing Method: GetRandomMessage()

File: TestBridge.cs:149

Taint Flags:

```
No snippet available
```

TestBridge.cs, line 83 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 21:01:18 GMT+0100 (CET)

Es un test, no es explotable

Sink Details

Sink: Next()

Enclosing Method: TestSendReceive()

File: TestBridge.cs:83

Taint Flags:

No snippet available

TestBridge.cs, line 87 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 21:01:21 GMT+0100 (CET)

Es un test, no es explotable

Sink Details

Sink: Next()

Enclosing Method: TestSendReceive()

File: TestBridge.cs:87

Taint Flags:

No snippet available

TestBridge.cs, line 152 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 21:01:29 GMT+0100 (CET)

Es un test, no es explotable

Sink Details

Sink: Next()

Enclosing Method: GetRandomMessage()

File: TestBridge.cs:152

Taint Flags:

No snippet available

TestTCPCommsLink.cs, line 344 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 21:01:39 GMT+0100 (CET)

Es un test, no es explotable

Sink Details

Sink: Next()

Enclosing Method: GetClientServerStrings()

File: TestTCPCommsLink.cs:344

Taint Flags:

```
No snippet available
```

Package: TestConfiguration

ConfigurationItem_ReaderWriterLock.cs, line 50 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 20:29:10 GMT+0100 (CET)

Son test, no aplica

Sink Details

Sink: Next()

Enclosing Method: BuildRandomTree()

File: ConfigurationItem_ReaderWriterLock.cs:50

Taint Flags:

```
No snippet available
```

ConfigurationItem_basic_test.cs, line 440 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 20:28:39 GMT+0100 (CET)

Son test, no se pueden atacar

Sink Details

Sink: NextDouble()

Enclosing Method: ConfigurationItem_Get_single_float_Test()

File: ConfigurationItem_basic_test.cs:440

Taint Flags:

```
No snippet available
```

ConfigurationItem_ReaderWriterLock.cs, line 178 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 20:30:10 GMT+0100 (CET)

Son test, no aplica

Sink Details

Sink: Next()

Enclosing Method: DoWrite()

File: ConfigurationItem_ReaderWriterLock.cs:178

Taint Flags:

```
No snippet available
```

ConfigurationItem_ReaderWriterLock.cs, line 159 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 20:30:02 GMT+0100 (CET)

Son test, no aplica

Sink Details

Sink: Next()

Enclosing Method: DoWrite()

File: ConfigurationItem_ReaderWriterLock.cs:159

Taint Flags:

```
No snippet available
```

ConfigurationItem_ReaderWriterLock.cs, line 94 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 20:29:42 GMT+0100 (CET)

Son test, no aplica

Sink Details

Sink: Next()

Enclosing Method: DoRead()

File: ConfigurationItem_ReaderWriterLock.cs:94

Taint Flags:

```
No snippet available
```

ConfigurationItem_ReaderWriterLock.cs, line 78 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 20:29:32 GMT+0100 (CET)

Son test, no aplica

Sink Details

Sink: Next()

Enclosing Method: DoRead()

File: ConfigurationItem_ReaderWriterLock.cs:78

Taint Flags:

```
No snippet available
```

ConfigurationItem_ReaderWriterLock.cs, line 181 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 20:30:17 GMT+0100 (CET)

Son test, no aplica

Sink Details

Sink: Next()

Enclosing Method: DoWrite()

File: ConfigurationItem_ReaderWriterLock.cs:181

Taint Flags:

```
No snippet available
```

ConfigurationItem_ReaderWriterLock.cs, line 163 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 20:30:06 GMT+0100 (CET)

Son test, no aplica

Sink Details

Sink: Next()

Enclosing Method: DoWrite()

File: ConfigurationItem_ReaderWriterLock.cs:163

Taint Flags:

```
No snippet available
```

ConfigurationItem_ReaderWriterLock.cs, line 62 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 20:29:25 GMT+0100 (CET)

Son test, no aplica

Sink Details

Sink: Next()

Enclosing Method: BuildRandomTree()

File: ConfigurationItem_ReaderWriterLock.cs:62

Taint Flags:

```
No snippet available
```

ConfigurationItem_ReaderWriterLock.cs, line 58 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 20:29:21 GMT+0100 (CET)

Son test, no aplica

Sink Details

Sink: Next()

Enclosing Method: BuildRandomTree()

File: ConfigurationItem_ReaderWriterLock.cs:58

Taint Flags:

No snippet available

ConfigurationItem_ReaderWriterLock.cs, line 97 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 20:29:47 GMT+0100 (CET)

Son test, no aplica

Sink Details

Sink: Next()

Enclosing Method: DoRead()

File: ConfigurationItem_ReaderWriterLock.cs:97

Taint Flags:

No snippet available

ConfigurationItem_ReaderWriterLock.cs, line 81 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 20:29:37 GMT+0100 (CET)

Son test, no aplica

Sink Details

Sink: Next()

Enclosing Method: DoRead()

File: ConfigurationItem_ReaderWriterLock.cs:81

Taint Flags:

No snippet available

Package: TestConfiguration.ConfigurationItem_ReaderWriterLock

ConfigurationItem_ReaderWriterLock.cs, line 122 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 20:29:53 GMT+0100 (CET)

Son test, no aplica

Sink Details

Sink: Next()

Enclosing Method: Invoke()

File: ConfigurationItem_ReaderWriterLock.cs:122

Taint Flags:

```
No snippet available
```

ConfigurationItem_ReaderWriterLock.cs, line 202 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 20:30:23 GMT+0100 (CET)

Son test, no aplica

Sink Details

Sink: Next()

Enclosing Method: Invoke()

File: ConfigurationItem_ReaderWriterLock.cs:202

Taint Flags:

```
No snippet available
```

Package: TestProgram

Program.cs, line 128 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis Exploitable

Audit Comments

MAS: Fri Dec 10 2021 21:00:37 GMT+0100 (CET)

Usar la libreria RandomNumberGenerator que es segura

Sink Details

Sink: Next()

Enclosing Method: Main()

File: Program.cs:128

Taint Flags:

```
No snippet available
```

Program.cs, line 109 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis Exploitable

Audit Comments

MAS: Fri Dec 10 2021 21:00:29 GMT+0100 (CET)

Usar la libreria RandomNumberGenerator que es segura

Sink Details

Sink: Next()

Enclosing Method: Main()

File: Program.cs:109

Taint Flags:

```
No snippet available
```

Package: Framework.Test

Program.cs, line 42 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis Exploitable

Audit Comments

MAS: Fri Dec 10 2021 20:59:58 GMT+0100 (CET)

Usar la libreria RandomNumberGenerator que es segura

Sink Details

Sink: Next()

Enclosing Method: Main()

File: Program.cs:42

Taint Flags:

```
No snippet available
```

Program.cs, line 52 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis Exploitable

Audit Comments

MAS: Fri Dec 10 2021 21:00:22 GMT+0100 (CET)

Usar la libreria RandomNumberGenerator que es segura

Sink Details

Sink: Next()

Enclosing Method: Main()

File: Program.cs:52

Taint Flags:

```
No snippet available
```

Program.cs, line 49 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis Exploitable

Audit Comments

MAS: Fri Dec 10 2021 21:00:15 GMT+0100 (CET)

Usar la libreria RandomNumberGenerator que es segura

Sink Details

Sink: Next()

Enclosing Method: Main()

File: Program.cs:49

Taint Flags:

```
No snippet available
```

Program.cs, line 45 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis Exploitable

Audit Comments

MAS: Fri Dec 10 2021 21:00:06 GMT+0100 (CET)

Usar la libreria RandomNumberGenerator que es segura

Sink Details

Sink: Next()

Enclosing Method: Main()

File: Program.cs:45

Taint Flags:

```
No snippet available
```

Package: Framework.Tools

Matrix.cs, line 252 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 20:59:32 GMT+0100 (CET)

Usar la libreria RandomNumberGenerator que es segura

MAS: Fri Dec 31 2021 13:03:55 GMT+0100 (CET)

No hace operaciones criptograficas

Sink Details

Sink: Next()

Enclosing Method: SelectDevice()

File: Matrix.cs:252

Taint Flags:

```
No snippet available
```

Matrix.cs, line 251 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 20:59:19 GMT+0100 (CET)

Usar la libreria RandomNumberGenerator que es segura

MAS: Fri Dec 31 2021 13:03:51 GMT+0100 (CET)

No hace operaciones criptograficas

Sink Details

Sink: Next()

Enclosing Method: SelectDevice()

File: Matrix.cs:251

Taint Flags:

```
No snippet available
```

Package: Tools.Test

AutoDiscoveryService_test.cs, line 27 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 20:18:51 GMT+0100 (CET)

Es un test, no aplica

Sink Details

Sink: Next()

Enclosing Method: AutoDiscovery_BasicTest()

File: AutoDiscoveryService_test.cs:27

Taint Flags:

```
No snippet available
```

Package: Tools.Test.TestSubscriberList

SubscriberListTest.cs, line 38 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 21:00:51 GMT+0100 (CET)

Es un test, no es explotable

Sink Details

Sink: Next()

Enclosing Method: InterfaceClass()

File: SubscriberListTest.cs:38

Taint Flags:

```
No snippet available
```

SubscriberListTest.cs, line 156 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 21:01:02 GMT+0100 (CET)

Es un test, no es explotable

Sink Details

Sink: Next()

Enclosing Method: DelegateClass()

File: SubscriberListTest.cs:156

Taint Flags:

```
No snippet available
```

Package: Tools.Test.TestWeakReferenceList

WeakReferenceListTest.cs, line 39 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 21:01:45 GMT+0100 (CET)

Es un test, no es explotable

Sink Details

Sink: Next()

Enclosing Method: InterfaceClass()

File: WeakReferenceListTest.cs:39

Taint Flags:

```
No snippet available
```

Package: nConfiguration.Test

nConfigTest.cs, line 105 (Insecure Randomness) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Semantic)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 20:59:47 GMT+0100 (CET)

Es un test, no aplica

Sink Details

Sink: Next()

Enclosing Method: SystemConfig_SetParam()

File: nConfigTest.cs:105

Taint Flags:

No snippet available

Missing XML Validation (3 issues)

Abstract

Si no se consigue habilitar la validación a la hora de analizar XML, el atacante tendrá la oportunidad de proporcionar entradas maliciosas.

Explanation

Los ataques con más éxito comienzan con una violación de los supuestos del programador. Si se acepta un documento XML sin validarlo con respecto a un esquema XML o DTD, el programador deja una puerta abierta para que los atacantes proporcionen entradas inesperadas, inadmisibles o maliciosas. Un analizador XML no puede validar todos los aspectos del contenido de un documento; no puede comprender la semántica completa de los datos. No obstante, puede realizar un trabajo completo y exhaustivo de comprobación de la estructura del documento y, así, garantizar que el código que procesa el documento que contiene cuenta con el formato correcto.

Recommendation

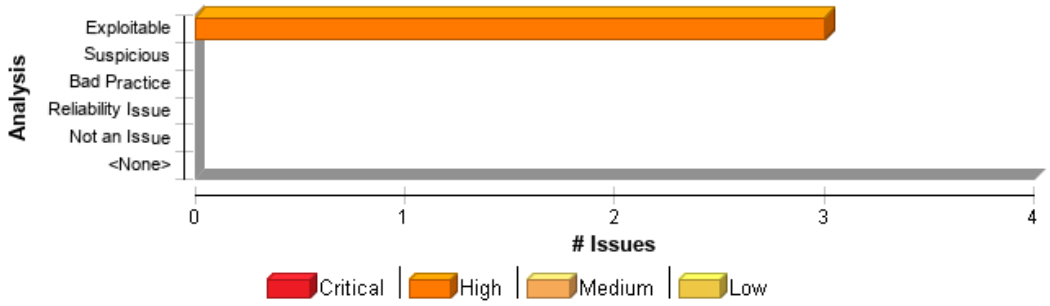
Habilite siempre la validación a la hora de analizar XML. Si surgen problemas a la hora de habilitar la validación porque las reglas de definición de un documento con formato correcto sean complejas o se desconozcan, existen muchas posibilidades de que existan errores de seguridad relacionados. **Ejemplo:** El código siguiente muestra cómo habilitar la validación cuando se utiliza `XmlReader`.

```
XmlReaderSettings settings = new XmlReaderSettings();
settings.Schemas.Add(schema);
settings.ValidationType = ValidationType.Schema;
StringReader sr = new StringReader(xmlDoc);
```

```
XmlReader reader = XmlReader.Create(sr, settings);
```

Issue Summary

Figura 84. *Gráfico de vulnerabilidades de categoría Missing XML Validation.*



Fuente: Elaboración propia.

Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Missing XML Validation	3	0	0	3
Total	**3**	**0**	**0**	**3**

Missing XML Validation	**High**
Package: Framework.Tools	
Xml.cs, line 134 (Missing XML Validation)	**High**
Issue Details	

Kingdom: Input Validation and Representation

Scan Engine: SCA (Semantic)

Audit Details

Analysis Exploitable

Audit Comments

MAS: Tue Dec 07 2021 19:46:36 GMT+0100 (CET)

No se realiza una validación correcta de la estructura del xml

Sink Details

Sink: Create()

Enclosing Method: ValidateSchema()

File: Xml.cs:134

Taint Flags:

No snippet available

Package: Framework.Tools.Serialization

WCFSerializationHelper.cs, line 107 (Missing XML Validation) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Semantic)

Audit Details

Analysis Exploitable

Audit Comments

MAS: Tue Dec 07 2021 19:42:42 GMT+0100 (CET)

No se valida si el xml tiene una estructura correcta

Sink Details

Sink: Create()

Enclosing Method: WcfDeserializeObjectFromString()

File: WCFSerializationHelper.cs:107

Taint Flags:

```
No snippet available
```

WCFSerializationHelper.cs, line 86 (Missing XML Validation) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Semantic)

Audit Details

Analysis Exploitable

Audit Comments

MAS: Tue Dec 07 2021 19:42:17 GMT+0100 (CET)

No se valida si el xml tiene una estructura correcta

Sink Details

Sink: Create()

Enclosing Method: WcfDeserializeFromString()

File: WCFSerializationHelper.cs:86

Taint Flags:

```
No snippet available
```

Null Dereference (47 issues)

Abstract

El programa puede eliminar potencialmente la referencia de un puntero nulo, generando una `NullException`.

Explanation

Los punteros nulos suelen producirse debido a que se ha infringido alguna presuposición del programador. La mayoría de los problemas con el puntero nulo provocan problemas generales de confiabilidad de software. Sin embargo, si un atacante puede desencadenar intencionadamente la eliminación de la referencia del puntero nulo, es posible que también pueda usar la excepción resultante para omitir la lógica de seguridad o para hacer que la aplicación revele información de depuración, la cual será valiosa para la planificación de ataques posteriores. **Ejemplo 1:** en el siguiente código, el programador presupone que el sistema tiene siempre definida una propiedad denominada "`cmd`". Si un usuario malintencionado puede controlar el entorno del programa para que no se defina "`cmd`", el programa genera una excepción de puntero nulo al intentar llamar al método `Trim()`.

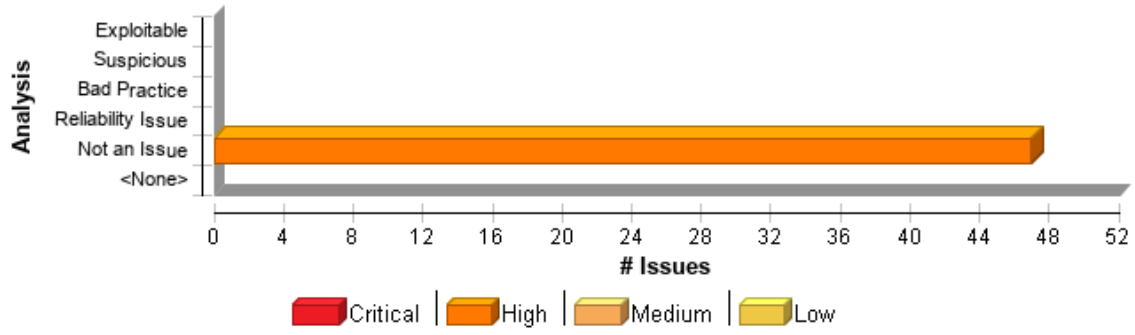
```
string cmd = null;
...
cmd = Environment.GetEnvironmentVariable("cmd");
cmd = cmd.Trim();
```

Recommendation

Los problemas de seguridad provocados por la eliminación de las referencias de punteros nulos están casi siempre relacionados con la forma en que el programa administra las excepciones de tiempo de ejecución. Si el software cuenta con un enfoque sólido y bien ejecutado en cuanto a la administración de las excepciones de tiempo de ejecución, se reducirá considerablemente la posibilidad de que se produzcan daños de seguridad.

Issue Summary

Figura 85. *Gráfico de vulnerabilidades de categoría Null Dereference.*



Fuente: Elaboración propia.

Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Null Dereference	47	0	0	47
Total	**47**	**0**	**0**	**47**

Null Dereference — **High**

Package: Framework.Configuration

ConfigurationItem.cs, line 1137 (Null Dereference) — **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 12:47:39 GMT+0100 (CET)

Se comprueba que no es nulo en la línea 1133

Sink Details

Sink: *(objItem) : objItem is not checked for null value before being dereferenced

Enclosing Method: Mount()

File: ConfigurationItem.cs:1137

Taint Flags:

```
No snippet available
```

ConfigurationItem.cs, line 2215 (Null Dereference) — **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 13:03:07 GMT+0100 (CET)

No se comprueba que enumeration es nulo

Sink Details

Sink: enumeration.GetEnumerator() : enumeration is not checked for null value before being dereferenced

Enclosing Method: StoreEnumeration()

File: ConfigurationItem.cs:2215

Taint Flags:

```
No snippet available
```

ProviderConfigNode.cs, line 233 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 14:46:20 GMT+0100 (CET)

Nunca other será null porque se comprueba en la línea 230 con result

Sink Details

Sink: other.get_DefaultValue() : other is not checked for null value before being dereferenced

Enclosing Method: Equals()

File: ProviderConfigNode.cs:233

Taint Flags:

```
No snippet available
```

ProviderConfigNode.cs, line 178 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 14:44:32 GMT+0100 (CET)

No se comprueba que result no sea null

Sink Details

Sink: *(result) : result is not checked for null value before being dereferenced

Enclosing Method: CloneNode()

File: ProviderConfigNode.cs:178

Taint Flags:

No snippet available

Package: Framework.Localization

Tools.cs, line 562 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 14:50:07 GMT+0100 (CET)

Nunca es null, porque nunca xDoc es null

Sink Details

Sink: xRoot.Add(...) : xRoot is not checked for null value before being dereferenced

Enclosing Method: BuildMultilanguageXml()

File: Tools.cs:562

Taint Flags:

No snippet available

Tools.cs, line 591 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 14:50:44 GMT+0100 (CET)

Nunca es null, porque xDoc nunca lo es

Sink Details

Sink: xRoot.Add(...) : xRoot is not checked for null value before being dereferenced

Enclosing Method: BuildMultilanguageXml()

File: Tools.cs:591

Taint Flags:

No snippet available

Tools.cs, line 558 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 14:49:42 GMT+0100 (CET)

xDoc nunca será null, se comprueba previamente

Sink Details

Sink: xDoc.get_FirstNode() : xDoc is not checked for null value before being dereferenced

Enclosing Method: BuildMultilanguageXml()

File: Tools.cs:558

Taint Flags:

```
No snippet available
```

Tools.cs, line 587 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 14:50:32 GMT+0100 (CET)

Nunca es null, porque xDoc nunca lo es

Sink Details

Sink: xDoc.get_FirstNode() : xDoc is not checked for null value before being dereferenced

Enclosing Method: BuildMultilanguageXml()

File: Tools.cs:587

Taint Flags:

```
No snippet available
```

Package: Framework.Tools

IPUtils.cs, line 245 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Mon Dec 13 2021 13:01:36 GMT+0100 (CET)

Nunca será null, se comprueba en esa misma línea

Sink Details

Sink: network.get_SupportsMulticast() : network is not checked for null value before being dereferenced

Enclosing Method: PacketsCanBeTransmitted()

File: IPUtils.cs:245

Taint Flags:

```
No snippet available
```

DoEvents.cs, line 64 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 13:12:44 GMT+0100 (CET)

Se fuerza que frame sea falso

Sink Details

Sink: frame.set_Continue(...) : frame is not checked for null value before being dereferenced

Enclosing Method: ExitFrame()

File: DoEvents.cs:64

Taint Flags:

```
No snippet available
```

Package: Framework.Tools.Collections

ObservableCollectionEx.cs, line 522 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 14:24:02 GMT+0100 (CET)

No llegará nunca vacío

Sink Details

Sink: .Refresh() : is not checked for null value before being dereferenced

Enclosing Method: Dispose()

File: ObservableCollectionEx.cs:522

Taint Flags:

```
No snippet available
```

Package: Framework.Tools.Collections.ObservableCollectionEx@1.NotificationInfo

ObservableCollectionEx.cs, line 600 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 14:24:40 GMT+0100 (CET)

Se comprueba que wrapper no es nulo

Sink Details

Sink: *(wrapper) : wrapper is not checked for null value before being dereferenced

Enclosing Method: Invoke()

File: ObservableCollectionEx.cs:600

Taint Flags:

```
No snippet available
```

Package: Framework.Tools.Message

SystemMessages.cs, line 81 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 14:48:37 GMT+0100 (CET)

No hace falta el chequeo, aquí se realiza si es null se pone return null si no el valor

Sink Details

Sink: attribs.get_Length() : attribs is not checked for null value before being dereferenced

Enclosing Method: GetStringValue()

File: SystemMessages.cs:81

Taint Flags:

```
No snippet available
```

Package: Framework.Tools.WcfExtension

MessageContractAnnotation.cs, line 93 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 14:08:44 GMT+0100 (CET)

No se comprueba que complexType es null ni la expresión que está aplicandose

Sink Details

Sink: .get_Items() : is not checked for null value before being dereferenced

Enclosing Method: ExportAnnotationForMessageBody()

File: MessageContractAnnotation.cs:93

Taint Flags:

```
No snippet available
```

XmlDocumentation.cs, line 265 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 14:52:28 GMT+0100 (CET)

No se comprueba si mi es null, lo que conlleva a que methodInfo pueda ser null

Sink Details

Sink: methodInfo.GetParameters() : methodInfo is not checked for null value before being dereferenced

Enclosing Method: CreateMemberName()

File: XmlDocumentation.cs:265

Taint Flags:

```
No snippet available
```

DocumentationAttribute.cs, line 303 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 13:11:40 GMT+0100 (CET)

Se comprueba que dataContractExporter no es nulo

Sink Details

Sink: dataContractExporter.get_Options() : dataContractExporter is not checked for null value before being dereferenced

Enclosing Method: ExportContract()

File: DocumentationAttribute.cs:303

Taint Flags:

```
No snippet available
```

Package: Framework.Tools.Win32

DriveMountDetector.cs, line 181 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 13:17:04 GMT+0100 (CET)

Se comprueba que la variable wpf_windows no es null, lo que conlleva a que lo que llega a source no es null

Sink Details

Sink: source.AddHook(...) : source is not checked for null value before being dereferenced

Enclosing Method: wpf_window_SourceInitialized()

File: DriveMountDetector.cs:181

Taint Flags:

```
No snippet available
```

Package: Plugins

AppController.cs, line 560 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Thu Dec 09 2021 12:17:29 GMT+0100 (CET)

No se comprueba si el objeto member es null

Sink Details

Sink: member.get_Member() : member is not checked for null value before being dereferenced

Enclosing Method: GetMemberInfo()

File: AppController.cs:560

Taint Flags:

```
No snippet available
```

AppController.cs, line 577 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Thu Dec 09 2021 12:17:42 GMT+0100 (CET)

No se comprueba si el objeto member es null

Sink Details

Sink: method.get_Method() : method is not checked for null value before being dereferenced

Enclosing Method: GetMethodInfo()

File: AppController.cs:577

Taint Flags:

```
No snippet available
```

Package: Framework.CommsLink.PipeCommsLink

PipeCommsLink.cs, line 317 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 14:44:03 GMT+0100 (CET)

No es nunca null

Sink Details

Sink: .CommsLinkStatusChanged(...) : is not checked for null value before being dereferenced

Enclosing Method: Invoke()

File: PipeCommsLink.cs:317

Taint Flags:

```
No snippet available
```

Package: Framework.CommsLink.SerialCommsLink

SerialCommsLink.cs, line 437 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 14:46:59 GMT+0100 (CET)

No es nunca null

Sink Details

Sink: .CommsLinkStatusChanged(...) : is not checked for null value before being dereferenced

Enclosing Method: Invoke()

File: SerialCommsLink.cs:437

Taint Flags:

```
No snippet available
```

Package: Framework.CommsLink.TCPCommsLink

TCPCommsLink.cs, line 655 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 14:48:56 GMT+0100 (CET)

No es nunca null

Sink Details

Sink: .CommsLinkStatusChanged(...) : is not checked for null value before being dereferenced

Enclosing Method: Invoke()

File: TCPCommsLink.cs:655

Taint Flags:

```
No snippet available
```

Package: Framework.CommsLink.UDPCommsLink

UDPCommsLink.cs, line 459 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 14:51:00 GMT+0100 (CET)

No es nunca null

Sink Details

Sink: .CommsLinkStatusChanged(...) : is not checked for null value before being dereferenced

Enclosing Method: Invoke()

File: UDPCommsLink.cs:459

Taint Flags:

```
No snippet available
```

Package: Framework

CFStorage.cs, line 359 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 09:26:04 GMT+0100 (CET)

No se comprueba que childrenRoot es nulo

Sink Details

Sink: childrenRoot.get_SID() : childrenRoot is not checked for null value before being dereferenced

Enclosing Method: AddStorage()

File: CFStorage.cs:359

Taint Flags:

```
No snippet available
```

CFStorage.cs, line 84 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Thu Dec 09 2021 12:21:45 GMT+0100 (CET)

Se comprueba que childrenTree no es nulo

Sink Details

Sink: .get_SID() : is not checked for null value before being dereferenced

Enclosing Method: LoadChildren()

File: CFStorage.cs:84

Taint Flags:

```
No snippet available
```

DirectoryEntry.cs, line 420 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 13:05:19 GMT+0100 (CET)

No se comprueba que d es nulo

Sink Details

Sink: d.SetEntryName(...) : d is not checked for null value before being dereferenced

Enclosing Method: AssignValueTo()

File: DirectoryEntry.cs:420

Taint Flags:

```
No snippet available
```

CompoundFile.cs, line 2240 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 11:40:59 GMT+0100 (CET)

No se comprueba que oldChain es nulo

Sink Details

Sink: oldChain.get_Count() : oldChain is not checked for null value before being dereferenced

Enclosing Method: SetStreamLength()

File: CompoundFile.cs:2240

Taint Flags:

```
No snippet available
```

CFStorage.cs, line 473 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 11:25:01 GMT+0100 (CET)

Se comprueba que Children es nulo en la línea 472

Sink Details

Sink: .get_SID() : is not checked for null value before being dereferenced

Enclosing Method: Delete()

File: CFStorage.cs:473

Taint Flags:

```
No snippet available
```

CFStorage.cs, line 494 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 11:26:23 GMT+0100 (CET)

Se comprueba que childrenRoot es nulo en la línea 447

MAS: Fri Dec 10 2021 11:37:36 GMT+0100 (CET)

**foundObj

Sink Details

Sink: .get_SID() : is not checked for null value before being dereferenced

Enclosing Method: Delete()

File: CFStorage.cs:494

Taint Flags:

```
No snippet available
```

CFStorage.cs, line 501 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 11:26:46 GMT+0100 (CET)

Se comprueba que Children es nulo en la línea 500

Sink Details

Sink: .get_SID() : is not checked for null value before being dereferenced

Enclosing Method: Delete()

File: CFStorage.cs:501

Taint Flags:

```
No snippet available
```

CFStorage.cs, line 135 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 09:08:55 GMT+0100 (CET)

No se comprueba si Children es nulo

Sink Details

Sink: .get_SID() : is not checked for null value before being dereferenced

Enclosing Method: AddStream()

File: CFStorage.cs:135

Taint Flags:

```
No snippet available
```

CFStorage.cs, line 468 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 11:24:34 GMT+0100 (CET)

No se comprueba que childrenRoot es nulo

MAS: Fri Dec 10 2021 11:37:12 GMT+0100 (CET)

**ded

Sink Details

Sink: ded.get_Name() : ded is not checked for null value before being dereferenced

Enclosing Method: Delete()

File: CFStorage.cs:468

Taint Flags:

```
No snippet available
```

Package: Framework.CFStorage

CFStorage.cs, line 399 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 11:36:43 GMT+0100 (CET)

No se comprueba que targetNode es nulo

Sink Details

Sink: d.get_StgType() : d is not checked for null value before being dereferenced

Enclosing Method: Invoke()

File: CFStorage.cs:399

Taint Flags:

```
No snippet available
```

Package: Framework.CompoundFile

CompoundFile.cs, line 2820 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 11:42:14 GMT+0100 (CET)

No se comprueba que itemAsStream es nulo

Sink Details

Sink: itemAsStream.get_Name() : itemAsStream is not checked for null value before being dereferenced

Enclosing Method: Invoke()

File: CompoundFile.cs:2820

Taint Flags:

```
No snippet available
```

CompoundFile.cs, line 2826 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 12:00:14 GMT+0100 (CET)

Se comprueba previamente con item.IsStorage en la línea 2823

Sink Details

Sink: itemAsStorage.get_Name() : itemAsStorage is not checked for null value before being dereferenced

Enclosing Method: Invoke()

File: CompoundFile.cs:2826

Taint Flags:

```
No snippet available
```

Package: Framework.FileReader

FileReader.cs, line 100 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 13:34:13 GMT+0100 (CET)

No se comprueba que node as IDirectoryEntry es nulo

Sink Details

Sink: .get_Name() : is not checked for null value before being dereferenced

Enclosing Method: Invoke()

File: FileReader.cs:100

Taint Flags:

```
No snippet available
```

Package: Framework.Services.Discovery

ServiceInfo.cs, line 109 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 14:47:21 GMT+0100 (CET)

Nunca other será null porque se comprueba en la línea 106 con result

Sink Details

Sink: other.get_Name() : other is not checked for null value before being dereferenced

Enclosing Method: Equals()

File: ServiceInfo.cs:109

Taint Flags:

```
No snippet available
```

Package: Framework.Tools.GUI

PieMenu.cs, line 674 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 14:25:41 GMT+0100 (CET)

menu_item nunca será nulo

Sink Details

Sink: menu_item.get_Background() : menu_item is not checked for null value before being dereferenced

Enclosing Method: DrawMenu()

File: PieMenu.cs:674

Taint Flags:

```
No snippet available
```

PieMenu.cs, line 733 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 14:26:41 GMT+0100 (CET)

header no es nulo, se comprueba que headerlist no sea null

Sink Details

Sink: header.get_Source() : header is not checked for null value before being dereferenced

Enclosing Method: DrawMenu()

File: PieMenu.cs:733

Taint Flags:

```
No snippet available
```

ThreadedWindowWrapper.cs, line 52 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Mon Dec 13 2021 13:04:04 GMT+0100 (CET)

No aplica, es un form

Sink Details

Sink: this.windowInstance.add_Loaded(...) : this.windowInstance is not checked for null value before being dereferenced

Enclosing Method: RunThread()

File: ThreadedWindowWrapper.cs:52

Taint Flags:

```
No snippet available
```

PieMenu.cs, line 178 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 14:25:16 GMT+0100 (CET)

i nunca es null

Sink Details

Sink: .CalculateSize(...) : is not checked for null value before being dereferenced

Enclosing Method: MeasureOverride()

File: PieMenu.cs:178

Taint Flags:

```
No snippet available
```

PieMenuItem.cs, line 72 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 14:38:55 GMT+0100 (CET)

i nunca será null porque forma parte de un foreach

Sink Details

Sink: .CalculateSize(...) : is not checked for null value before being dereferenced

Enclosing Method: CalculateSize()

File: PieMenuItem.cs:72

Taint Flags:

```
No snippet available
```

GridViewSort.cs, line 217 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 13:44:58 GMT+0100 (CET)

No se comprueba que binding es nulo

Sink Details

Sink: binding.get_Path() : binding is not checked for null value before being dereferenced

Enclosing Method: GetPropertyName()

File: GridViewSort.cs:217

Taint Flags:

```
No snippet available
```

WindowsManager.cs, line 588 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 14:53:00 GMT+0100 (CET)

Se realiza la comprobación de que mw no es null en la misma sentencia

Sink Details

Sink: mw.get_Name() : mw is not checked for null value before being dereferenced

Enclosing Method: GetMagneticWindows()

File: WindowsManager.cs:588

Taint Flags:

```
No snippet available
```

Package: fastJSON

JSON.cs, line 632 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 14:02:03 GMT+0100 (CET)

Se comprueba que t2 no es nulo ya que se comprueba previamente gtype que no sea null

Sink Details

Sink: t2.GetElementType() : t2 is not checked for null value before being dereferenced

Enclosing Method: RootDictionary()

File: JSON.cs:632

Taint Flags:

No snippet available

JSON.cs, line 1009 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 14:03:05 GMT+0100 (CET)

Se comprueba que types no es null, lo que conlleva a que t2 no es null y por tanto ga tampoco

Sink Details

Sink: t2.GetGenericArguments() : t2 is not checked for null value before being dereferenced

Enclosing Method: CreateStringKeyDictionary()

File: JSON.cs:1009

Taint Flags:

No snippet available

JSON.cs, line 366 (Null Dereference) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 14:01:17 GMT+0100 (CET)

Se comprueba que el valor de o no es null

Sink Details

Sink: .get_Count() : is not checked for null value before being dereferenced

Enclosing Method: ToObject()

File: JSON.cs:366

Taint Flags:

No snippet available

Often Misused: Authentication (1 issue)

Abstract

Los atacantes pueden reemplazar las entradas DNS. Por motivos de seguridad, no confíe en nombres DNS.

Explanation

Muchos servidores DNS son susceptibles de sufrir ataques, por eso debe suponer que su software se ejecutará alguna vez en un entorno con un servidor DNS afectado. Si los atacantes están autorizados a hacer actualizaciones de DNS (a veces denominado envenenamiento de la caché del DNS), pueden redirigir su tráfico de red a través de sus equipos o hacer que parezca que sus direcciones IP forman parte de su dominio. No base la seguridad de su sistema en nombres DNS. **Ejemplo:** el siguiente ejemplo de código utiliza una búsqueda DNS para determinar si una solicitud entrante es de un host de confianza o no. Si un atacante es capaz de dañar la caché DNS, puede obtener un estatus de confianza.

```
IPAddress hostIPAddress = IPAddress.Parse(RemoteIpAddress);
IPHostEntry hostInfo = Dns.GetHostByAddress(hostIPAddress);
if (hostInfo.HostName.EndsWith("trustme.com")) {
  trusted = true;
}
```

Las direcciones IP son más confiables que los nombres DNS, pero también se pueden reemplazar. Los atacantes pueden falsificar fácilmente las direcciones IP de origen de los paquetes que envían, aunque los paquetes de respuesta volverán a la dirección IP falsificada. Para ver los paquetes de respuesta, el atacante tiene que examinar el tráfico entre el equipo víctima y la dirección IP falsificada. A fin de completar el examen necesario, los atacantes normalmente intentan ubicarse en la misma subred que el equipo víctima. Los atacantes podrían ser capaces de evitar este requisito empleando enrutamiento de origen, aunque el enrutamiento de origen esté deshabilitado actualmente en gran parte de Internet. En resumen, la verificación de las direcciones IP puede ser una parte útil de un esquema de autenticación, pero no debe ser el único factor necesario para la autenticación.

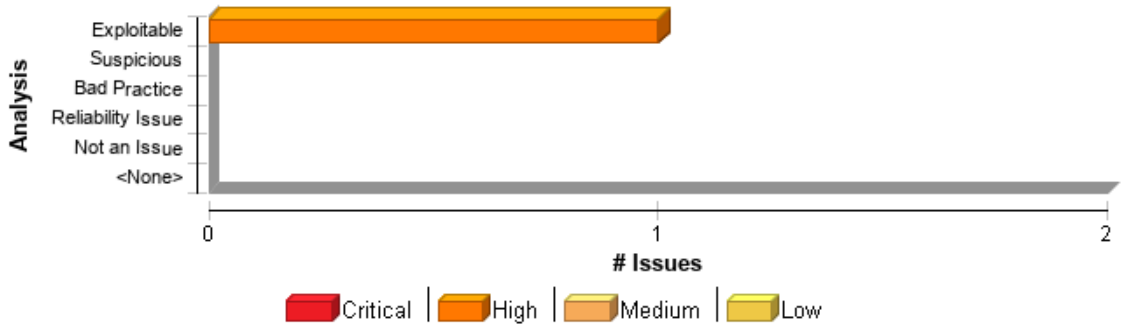
Recommendation

Puede aumentar la confianza en una búsqueda del nombre de dominio si comprueba que las entradas DNS enviadas y devueltas del host coinciden. Los atacantes no podrán reemplazar las entradas DNS enviadas y devueltas, ni las entradas DNS inversas, sin controlar los servidores de nombres del dominio de destino. Sin embargo, este no es un método infalible: los atacantes podrían ser capaces de convencer al registrador del dominio para que convierta el dominio en un servidor de nombres malintencionado. Basar la autenticación en entradas DNS es simplemente una propuesta arriesgada. A pesar de que ningún mecanismo de autenticación es confiable, hay alternativas mejores que la autenticación basada en host. Los sistemas de contraseñas ofrecen una seguridad aceptable, pero se pueden elegir malas contraseñas, transmitir contraseñas inseguras y realizar una mala administración

de las mismas. Merece la pena considerar un esquema criptográfico como SSL, aunque dichos esquemas son normalmente tan complejos que conllevan el peligro de significativos errores de implementación. Además, siempre se puede robar la información de la clave. En muchas ocasiones, la autenticación multifactor que incluye un token físico ofrece la mayor seguridad disponible a un precio razonable.

Issue Summary

Figura 86. *Gráfico de vulnerabilidades de categoría Often Misused: Authentication.*



Fuente: Elaboración propia.

Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Often Misused: Authentication	1	0	0	1
Total	**1**	**0**	**0**	**1**

Often Misused: Authentication	**High**
Package: Framework.Tools	
IPUtils.cs, line 152 (Often Misused: Authentication)	**High**
Issue Details	
Kingdom: API Abuse	
Scan Engine: SCA (Semantic)	

Audit Details	
Analysis	Exploitable
Audit Comments	
MAS: Tue Dec 07 2021 20:15:20 GMT+0100 (CET)	
Comprobar que las entradas DNS enviadas y devueltas del host coinciden	
Sink Details	
Sink: GetHostName()	
Enclosing Method: GetMachineNetworks()	
File: IPUtils.cs:152	

Taint Flags:

No snippet available

Password Management: Hardcoded Password (1 issue)

Abstract

Las contraseñas codificadas pueden poner en riesgo la seguridad del sistema de una forma que no es fácil resolver.

Explanation

Nunca es una buena idea integrar una contraseña en el código. Integrar una contraseña en el código no solo permite a todos los desarrolladores del proyecto ver la contraseña, sino que también hace que sea extremadamente difícil solucionar el problema. Una vez que el código está en producción, la contraseña no se puede cambiar sin aplicar revisiones al software. Si la cuenta protegida por la contraseña se ve comprometida, los propietarios del sistema deberán elegir entre la seguridad y la disponibilidad. **Ejemplo:** El siguiente código utiliza una contraseña codificada para crear una credencial de red:

```
...
NetworkCredential netCred =
            new NetworkCredential("user", "password", domain);
...
```

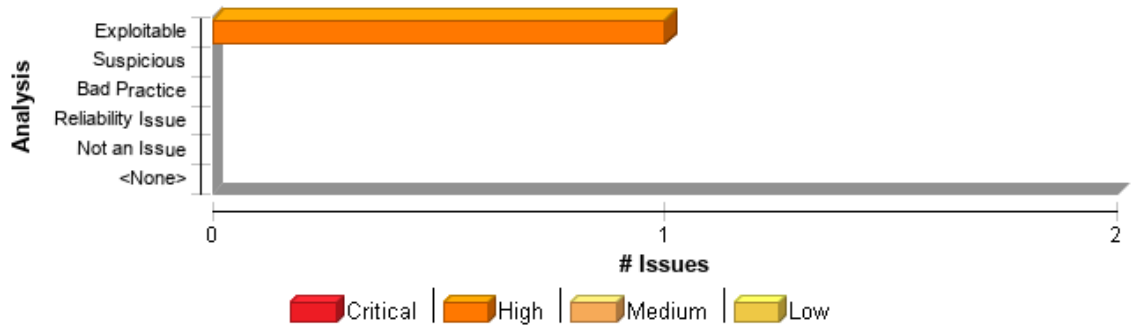
Este código se ejecutará correctamente, pero cualquiera que tenga acceso a este tendrá acceso a la contraseña. Una vez lanzado el programa, no se podrá cambiar el usuario de las credenciales de red "scott" con la contraseña "tiger", a menos que el programa tenga instaladas las revisiones. Un empleado con acceso a esta información podría utilizarla para irrumpir en el sistema. Si los usuarios malintencionados tienen acceso al ejecutable de la aplicación, podrían desensamblar el código, donde se encuentran los valores de las contraseñas utilizadas.

Recommendation

Las contraseñas nunca deben estar integrada en el código y, por lo general, deben ser confusas y deben administrarse en un origen externo. Si se almacenan contraseñas de texto sin formato en cualquier lugar del sistema, se permite que un usuario con permisos suficientes pueda leerlas y utilizarlas incorrectamente. Microsoft(R) proporciona una herramienta que se puede utilizar junto con la Interfaz de programación de aplicaciones de protección de datos (DPAPI) de Windows para proteger entradas de aplicaciones confidenciales en los archivos de configuración [1].

Issue Summary

Figura 87. *Gráfico de vulnerabilidades de categoría Password Management: Hardcoded Password.*



Fuente: Elaboración propia.

Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Password Management: Hardcoded Password	1	0	0	1
Total	**1**	**0**	**0**	**1**

Password Management: Hardcoded Password	**High**
Package: Framework.Localization	
Tools.cs, line 31 (Password Management: Hardcoded Password)	**High**
Issue Details	

Kingdom: Security Features

Scan Engine: SCA (Structural)

Audit Details	
Analysis	Exploitable

Audit Comments

MAS: Tue Dec 07 2021 19:47:59 GMT+0100 (CET)

Password hadcodeada en el cs

Sink Details

Sink: Field: FILE_PASSWORD

File: Tools.cs:31

Taint Flags:

```
No snippet available
```

Path Manipulation (17 issues)

Abstract

Si se permite que una entrada de usuario controle las rutas de acceso que se usan en operaciones del sistema de archivos, un atacante podría acceder a recursos del sistema protegidos o modificarlos de algún modo.

Explanation

Se producen errores de manipulación de la ruta de acceso cuando se cumplen las dos condiciones siguientes: 1. Un atacante puede especificar una ruta de acceso que se utiliza en una operación en el sistema de archivos. 2. Al especificar el recurso, el usuario malintencionado consigue una capacidad que de otro modo no estaría permitida. Por ejemplo, el programa puede otorgar al atacante la capacidad de sobrescribir el archivo especificado o ejecutar una configuración controlada por el atacante. **Ejemplo 1:** El código siguiente utiliza la entrada de una solicitud HTTP para crear un nombre de archivo. El programador no ha considerado la posibilidad de que un usuario malintencionado pueda proporcionar un nombre de archivo como "..\\..\\Windows\\System32\\krnl386.exe", lo que hará que la aplicación elimine un archivo importante del sistema de Windows.

```
String rName = Request.Item("reportName");
...
File.delete("C:\\users\\reports\\" + rName);
```

Ejemplo 2: el siguiente código utiliza una entrada de un archivo de configuración para determinar el archivo que se debe abrir y reenviar al usuario. Si el programa se ejecuta con privilegios adecuados y los usuarios malintencionados pueden modificar el archivo de configuración, estos podrán utilizar el programa para leer cualquier archivo del sistema que termine con la extensión ".txt".

```
sr = new StreamReader(resmngr.GetString("sub")+".txt");
while ((line = sr.ReadLine()) != null) {
Console.WriteLine(line);
}
```

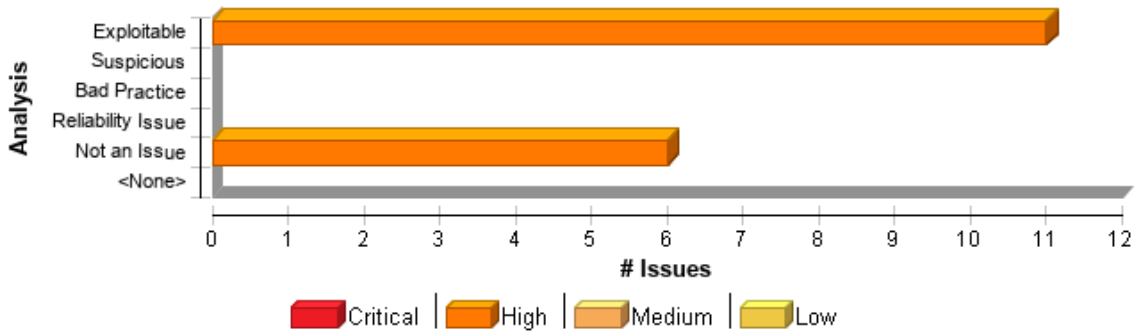
Recommendation

La mejor forma de evitar la manipulación de rutas de acceso es con un nivel de direccionamiento indirecto: cree una lista de valores autorizados que los usuarios puedan seleccionar. Con este método, la entrada proporcionada por el usuario nunca se utiliza directamente para especificar el nombre de recurso. En algunas situaciones, este método no resulta práctico, ya que el conjunto de nombres de recursos autorizados es demasiado grande o demasiado difícil de mantener. Los programadores a menudo recurren a implementar una lista de rechazados en estas situaciones. Este tipo de lista se utiliza para rechazar o eludir de forma selectiva caracteres potencialmente peligrosos antes de utilizar la entrada. Sin embargo, cualquier lista de caracteres no seguros de estas características tenderá a estar incompleta y a quedar obsoleta. Un enfoque más práctico consiste en crear una lista de caracteres

permitidos que pueden aparecer en el nombre del recurso y aceptar la entrada formada exclusivamente por caracteres del conjunto aprobado.

Issue Summary

Figura 88. *Gráfico de vulnerabilidades de categoría Path Manipulation.*



Fuente: Elaboración propia.

Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Path Manipulation	17	0	0	17
Total	**17**	**0**	**0**	**17**

Path Manipulation	**High**
Package: Framework.Tools	
FileTraceListener.cs, line 35 (Path Manipulation)	**High**
Issue Details	

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 21:24:43 GMT+0100 (CET)

Es un acceso al path del log de la aplicación, no aplica seguridad del path

Source Details

Source: System.Environment.GetCommandLineArgs()

From: Framework.Tools.Tracer.GetExePath

File: Tracer.cs:959

```
956
957  private static string GetExePath ()
```

```
958  {
959   string dir = Path.GetDirectoryName (Environment.GetCommandLineArgs
()[0]);
960  if (string.IsNullOrWhiteSpace (dir))
961  {
962  dir = Directory.GetCurrentDirectory ();
```

Sink Details

Sink: System.IO.Directory.CreateDirectory()

Enclosing Method: FileTraceListener()

File: FileTraceListener.cs:35

Taint Flags: ARGS, VALIDATED_PATH_MANIPULATION_BASE_PATH_OVERWRITING

No snippet available

FileTraceListener.cs, line 24 (Path Manipulation) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 21:22:30 GMT+0100 (CET)

Es un acceso al path del log de la aplicación, no aplica seguridad del path

Source Details

Source: System.Environment.ExpandEnvironmentVariables()

From: Framework.Tools.Tracer.GetTempPath

File: Tracer.cs:947

```
944
945  private static string GetTempPath ()
946  {
947  return Environment.ExpandEnvironmentVariables ("%temp%");
948  }
949
950  private static string GetUserPath ()
```

Sink Details

Sink: System.IO.File.Open()

Enclosing Method: FileTraceListener()

File: FileTraceListener.cs:24

Taint Flags: ENVIRONMENT, VALIDATED_PATH_MANIPULATION_BASE_PATH_OVERWRITING

```
No snippet available
```

Tracer.cs, line 327 (Path Manipulation) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Exploitable

Audit Comments

MAS: Sat Dec 11 2021 13:14:35 GMT+0100 (CET)

Se deben controlar los argumentos introducidas en la consola, para evitar acceso a ficheros protegidos

Source Details

Source: System.Environment.GetCommandLineArgs()

From: Framework.Tools.Tracer.GetExePath

File: Tracer.cs:959

```
956
957  private static string GetExePath ()
958  {
959   string dir = Path.GetDirectoryName (Environment.GetCommandLineArgs
()[0]);
960  if (string.IsNullOrWhiteSpace (dir))
961  {
962  dir = Directory.GetCurrentDirectory ();
```

Sink Details

Sink: System.IO.DirectoryInfo.DirectoryInfo()

Enclosing Method: GetValidLogFileLocation()

File: Tracer.cs:327

Taint Flags: ARGS

No snippet available

Paths.cs, line 142 (Path Manipulation) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Exploitable

Audit Comments

MAS: Sat Dec 11 2021 13:12:54 GMT+0100 (CET)

Se debe controlar el path de entrada al método GetAbsolutePath, para evitar acceso a ficheros protegidos por parte de un atacante

Source Details

Source: System.Environment.ExpandEnvironmentVariables()

From: Framework.Tools.Paths.GetAbsolutePath

File: Paths.cs:65

```
62
63  if (!string.IsNullOrEmpty (pPath))
64  {
65  string sPath = Environment.ExpandEnvironmentVariables (pPath);
66
67  if (!String.IsNullOrEmpty (sPath))
68  {
```

Sink Details

Sink: System.IO.DirectoryInfo.DirectoryInfo()

Enclosing Method: GetAbsolutePath()

File: Paths.cs:142

Taint Flags: ENVIRONMENT

No snippet available

FileTraceListener.cs, line 24 (Path Manipulation) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 21:22:25 GMT+0100 (CET)

Es un acceso al path del log de la aplicación, no aplica seguridad del path

Source Details

Source: System.Environment.GetCommandLineArgs()

From: Framework.Tools.Tracer.GetExePath

File: Tracer.cs:959

```
956
957  private static string GetExePath ()
958  {
959   string dir = Path.GetDirectoryName (Environment.GetCommandLineArgs
()[0]);
960  if (string.IsNullOrWhiteSpace (dir))
961  {
962  dir = Directory.GetCurrentDirectory ();
```

Sink Details

Sink: System.IO.File.Open()

Enclosing Method: FileTraceListener()

File: FileTraceListener.cs:24

Taint Flags: ARGS, VALIDATED_PATH_MANIPULATION_BASE_PATH_OVERWRITING

No snippet available

Tracer.cs, line 339 (Path Manipulation) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Sat Dec 11 2021 13:17:12 GMT+0100 (CET)

No aplica, temp folder siempre es la misma

Source Details

Source: System.Environment.ExpandEnvironmentVariables()

From: Framework.Tools.Tracer.GetTempPath

File: Tracer.cs:947

```
944
945  private static string GetTempPath ()
946  {
947  return Environment.ExpandEnvironmentVariables ("%temp%");
948  }
949
950  private static string GetUserPath ()
```

Sink Details

Sink: System.IO.DirectoryInfo.DirectoryInfo()

Enclosing Method: GetValidLogFileLocation()

File: Tracer.cs:339

Taint Flags: ENVIRONMENT

No snippet available

Tracer.cs, line 1173 (Path Manipulation) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Exploitable

Audit Comments

MAS: Sat Dec 11 2021 13:17:32 GMT+0100 (CET)

Se deben controlar los argumentos introducidas en la consola, para evitar acceso a ficheros protegidos

Source Details

Source: System.Environment.GetCommandLineArgs()

From: Framework.Tools.Tracer.GetExePath

File: Tracer.cs:959

```
956
```

```
957  private static string GetExePath ()
958  {
959   string dir = Path.GetDirectoryName (Environment.GetCommandLineArgs
()[0]);
960  if (string.IsNullOrWhiteSpace (dir))
961  {
962  dir = Directory.GetCurrentDirectory ();
```

Sink Details

Sink: System.IO.File.ReadAllLines()

Enclosing Method: LoadConfigFile()

File: Tracer.cs:1173

Taint Flags: ARGS

No snippet available

Tracer.cs, line 1068 (Path Manipulation) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Sat Dec 11 2021 13:17:26 GMT+0100 (CET)

Se deben controlar los argumentos introducidas en la consola, para evitar acceso a ficheros protegidos

Source Details

Source: System.Environment.GetCommandLineArgs()

From: Framework.Tools.Tracer.GetExePath

File: Tracer.cs:959

```
956
957  private static string GetExePath ()
958  {
959   string dir = Path.GetDirectoryName (Environment.GetCommandLineArgs
()[0]);
960  if (string.IsNullOrWhiteSpace (dir))
961  {
```

```
962   dir = Directory.GetCurrentDirectory ();
```

Sink Details

Sink: System.IO.File.WriteAllText()

Enclosing Method: SaveSwitchesToConfigFile()

File: Tracer.cs:1068

Taint Flags: ARGS

No snippet available

FileTraceListener.cs, line 35 (Path Manipulation) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Fri Dec 10 2021 21:24:41 GMT+0100 (CET)

Es un acceso al path del log de la aplicación, no aplica seguridad del path

Source Details

Source: System.Environment.ExpandEnvironmentVariables()

From: Framework.Tools.Tracer.GetTempPath

File: Tracer.cs:947

```
944
945   private static string GetTempPath ()
946   {
947   return Environment.ExpandEnvironmentVariables ("%temp%");
948   }
949
950   private static string GetUserPath ()
```

Sink Details

Sink: System.IO.Directory.CreateDirectory()

Enclosing Method: FileTraceListener()

File: FileTraceListener.cs:35

Taint Flags: ENVIRONMENT, VALIDATED_PATH_MANIPULATION_BASE_PATH_OVERWRITING

No snippet available

Package: BuildPacket

Program.cs, line 81 (Path Manipulation) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Exploitable

Audit Comments

MAS: Sat Dec 11 2021 13:13:22 GMT+0100 (CET)

Se deben controlar los argumentos de entrada al método main, para evitar acceso a ficheros protegidos

Source Details

Source: Main(0)

From: BuildPacket.Program.Main

File: Program.cs:21

```
18
19  partial class Program
20  {
21  static void Main (string[] args)
22  {
23  try
24  {
```

Sink Details

Sink: System.IO.File.Delete()

Enclosing Method: BuildCommand()

File: Program.cs:81

Taint Flags: ARGS

No snippet available

Program.cs, line 82 (Path Manipulation) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Exploitable

Audit Comments

MAS: Sat Dec 11 2021 13:13:33 GMT+0100 (CET)

Se deben controlar los argumentos de entrada al método main, para evitar acceso a ficheros protegidos

Source Details

Source: Main(0)

From: BuildPacket.Program.Main

File: Program.cs:21

```
18
19  partial class Program
20  {
21  static void Main (string[] args)
22  {
23  try
24  {
```

Sink Details

Sink: System.IO.File.Move()

Enclosing Method: BuildCommand()

File: Program.cs:82

Taint Flags: ARGS

No snippet available

Program.cs, line 82 (Path Manipulation) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Exploitable

Audit Comments

MAS: Sat Dec 11 2021 13:13:38 GMT+0100 (CET)

Se deben controlar los argumentos de entrada al método main, para evitar acceso a ficheros protegidos

Source Details

Source: Main(0)

From: BuildPacket.Program.Main

File: Program.cs:21

```
18
19  partial class Program
20  {
21  static void Main (string[] args)
22  {
23  try
24  {
```

Sink Details

Sink: System.IO.File.Move()

Enclosing Method: BuildCommand()

File: Program.cs:82

Taint Flags: ARGS

No snippet available

CommandOptions.cs, line 182 (Path Manipulation) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Exploitable

Audit Comments

MAS: Fri Dec 10 2021 21:08:09 GMT+0100 (CET)

Se deben controlar los argumentos de entrada al método main

Source Details

Source: Main(0)

From: BuildPacket.Program.Main

File: Program.cs:21

```
18
19  partial class Program
20  {
21  static void Main (string[] args)
22  {
23  try
24  {
```

Sink Details

Sink: System.IO.File.ReadAllLines()

Enclosing Method: ReadMultipleFileSpecs()

File: CommandOptions.cs:182

Taint Flags: ARGS, START_CHECKED_STRING

No snippet available

Package: Framework

FileWriter.cs, line 313 (Path Manipulation) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Exploitable

Audit Comments

MAS: Fri Dec 10 2021 21:45:50 GMT+0100 (CET)

Se deben controlar los argumentos de entrada al método main, para evitar acceso a ficheros protegido

Source Details

Source: Main(0)

From: BuildPacket.Program.Main

File: Program.cs:21

```
18
19  partial class Program
20  {
21  static void Main (string[] args)
22  {
23  try
```

```
24  {
```

Sink Details

Sink: System.IO.FileStream.FileStream()

Enclosing Method: ReadFileBytes()

File: FileWriter.cs:313

Taint Flags: ARGS, CANONICAL_PATH, START_CHECKED_STRING

No snippet available

CompoundFile.cs, line 731 (Path Manipulation) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Exploitable

Audit Comments

MAS: Fri Dec 10 2021 21:14:44 GMT+0100 (CET)

Se deben controlar los argumentos de entrada al método main, para evitar acceso a ficheros protegidos

Source Details

Source: Main(0)

From: BuildPacket.Program.Main

File: Program.cs:21

```
18
19  partial class Program
20  {
21  static void Main (string[] args)
22  {
23  try
24  {
```

Sink Details

Sink: System.IO.FileStream.FileStream()

Enclosing Method: LoadFile()

File: CompoundFile.cs:731

Taint Flags: ARGS

No snippet available

CompoundFile.cs, line 736 (Path Manipulation) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Exploitable

Audit Comments

MAS: Fri Dec 10 2021 21:14:57 GMT+0100 (CET)

Se deben controlar los argumentos de entrada al método main, para evitar acceso a ficheros protegidos

Source Details

Source: Main(0)

From: BuildPacket.Program.Main

File: Program.cs:21

```
18
19  partial class Program
20  {
21  static void Main (string[] args)
22  {
23  try
24  {
```

Sink Details

Sink: System.IO.FileStream.FileStream()

Enclosing Method: LoadFile()

File: CompoundFile.cs:736

Taint Flags: ARGS

No snippet available

FileWriter.cs, line 71 (Path Manipulation) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Exploitable

Audit Comments

MAS: Fri Dec 10 2021 21:45:40 GMT+0100 (CET)

Se deben controlar los argumentos de entrada al método main, para evitar acceso a ficheros protegidos

Source Details

Source: Main(0)

From: BuildPacket.Program.Main

File: Program.cs:21

```
18
19  partial class Program
20  {
21  static void Main (string[] args)
22  {
23  try
24  {
```

Sink Details

Sink: System.IO.FileStream.FileStream()

Enclosing Method: Dispose()

File: FileWriter.cs:71

Taint Flags: ARGS

No snippet available

Path Manipulation: Base Path Overwriting (3 issues)

Abstract

Permitir que una entrada de usuario controle las rutas de acceso que se usan en operaciones del sistema de archivos, permitiría a un atacante acceder a recursos del sistema protegidos o modificarlos de otro modo.

Explanation

Path.Combine utiliza varias rutas de archivos como argumentos. Las concatena para obtener la ruta completa, lo cual habitualmente viene seguido de una llamada a `read()` o `write()` para ese archivo. La documentación describe varios escenarios distintos según si el primer parámetro o los restantes son

rutas absolutas. Si se proporciona una ruta absoluta para el segundo parámetro o los restantes, `Path.Combine()` devolverá esa ruta absoluta. Los parámetros anteriores se ignorarán. Las implicaciones en este caso son significativas para aplicaciones que tienen código similar al ejemplo siguiente. **Ejemplo 1:**El código siguiente carga un archivo de forma insegura con elementos de ruta controlados por el usuario:

```
// Called with user-controlled data
public static bytes[] getFile(String filename)
{
  String imageDir = "\\FILESHARE\images\";
  filepath = Path.Combine(imageDir, filename);
  return File.ReadAllBytes(filepath);
}
```

Al proporcionar una ruta absoluta (p. ej., `C:\\inetpub\wwwroot\web.config`), un atacante puede controlar qué archivo devuelve la aplicación.

Recommendation

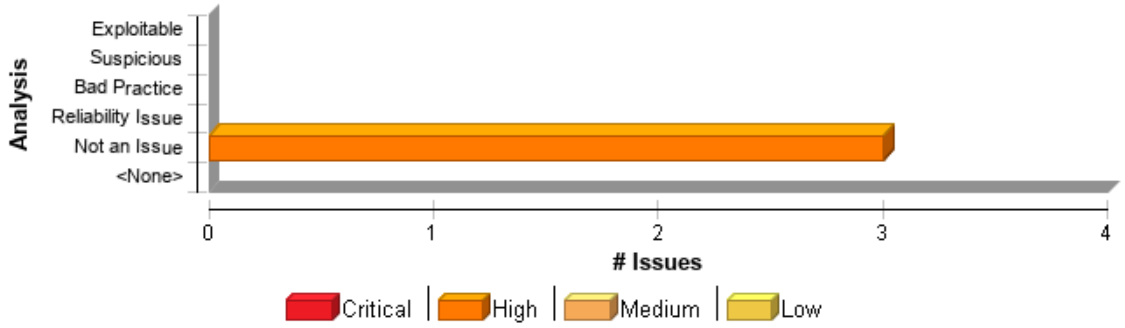
Asegúrese de que se compruebe que las rutas controladas por el usuario son absolutas. Puede utilizar `System.IO.Path.IsPathRooted()`, puesto que devolverá `true` para las rutas absolutas del sistema de archivos y UNC. **Ejemplo 2:**El código siguiente comprueba que el elemento de ruta controlado por el usuario no represente una ruta absoluta:

```
// Called with user-controlled data
public static bytes[] GetFile(String filename) {
  // Ensure that all path elements are safe path elements.
  if (Path.IsPathRooted(filename))
  {
      throw new ArgumentNullException("error");
  }
  String filepath = Path.Combine("\\FILESHARE\images", filename);
   // Additional checks to prevent other path traversal attacks

  return File.ReadAllBytes(filepath);
}
```

Issue Summary

Figura 89. *Gráfico de vulnerabilidades de categoríaPath Manipulation: Base Path Overwriting.*



Fuente: Elaboración propia.

Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Path Manipulation: Base Path Overwriting	3	0	0	3
Total	**3**	**0**	**0**	**3**

Path Manipulation: Base Path Overwriting	**High**
Package: Framework.Tools	
FileTraceListener.cs, line 37 (Path Manipulation: Base Path Overwriting)	**High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Tue Dec 07 2021 20:46:39 GMT+0100 (CET)

Se realiza una comprobación de la path previamente

Source Details

Source: System.Environment.ExpandEnvironmentVariables()

From: Framework.Tools.Tracer.GetTempPath

File: Tracer.cs:947

```
944
945  private static string GetTempPath ()
946  {
947  return Environment.ExpandEnvironmentVariables ("%temp%");
948  }
949
950  private static string GetUserPath ()
```

Sink Details

Sink: System.IO.Path.Combine()

Enclosing Method: FileTraceListener()

File: FileTraceListener.cs:37

Taint Flags: ENVIRONMENT, VALIDATED_PATH_MANIPULATION_ZIP_ENTRY_OVERWRITE

```
No snippet available
```

Paths.cs, line 81 (Path Manipulation: Base Path Overwriting) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Tue Dec 07 2021 20:40:53 GMT+0100 (CET)

Se realiza una comprobación de la path previamente

Source Details

Source: System.Environment.ExpandEnvironmentVariables()

From: Framework.Tools.Paths.GetAbsolutePath

File: Paths.cs:65

```
62
63  if (!string.IsNullOrEmpty (pPath))
64  {
65  string sPath = Environment.ExpandEnvironmentVariables (pPath);
66
67  if (!String.IsNullOrEmpty (sPath))
68  {
```

Sink Details

Sink: System.IO.Path.Combine()

Enclosing Method: GetAbsolutePath()

File: Paths.cs:81

Taint Flags: ENVIRONMENT

No snippet available

FileTraceListener.cs, line 37 (Path Manipulation: Base Path Overwriting) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Tue Dec 07 2021 20:46:42 GMT+0100 (CET)

Se realiza una comprobación de la path previamente

Source Details

Source: System.Environment.GetCommandLineArgs()

From: Framework.Tools.Tracer.GetExePath

File: Tracer.cs:959

```
956
957  private static string GetExePath ()
958  {
959   string dir = Path.GetDirectoryName (Environment.GetCommandLineArgs
()[0]);
960  if (string.IsNullOrWhiteSpace (dir))
961  {
962  dir = Directory.GetCurrentDirectory ();
```

Sink Details

Sink: System.IO.Path.Combine()

Enclosing Method: FileTraceListener()

File: FileTraceListener.cs:37

Taint Flags: ARGS, VALIDATED_PATH_MANIPULATION_ZIP_ENTRY_OVERWRITE

No snippet available

Portability Flaw: File Separator (1 issue)

Abstract

El uso de separadores de archivos codificados provoca problemas de portabilidad.

Explanation

Cada sistema operativo utiliza caracteres diferentes como separadores de archivos. Por ejemplo, los sistemas Microsoft Windows utilizan "\", mientras que los sistemas UNIX utilizan "/". Cuando las aplicaciones tienen que ejecutarse en distintas plataformas, el uso de separadores de archivos codificados puede provocar la ejecución incorrecta de la lógica de la aplicación y, posiblemente, una denegación de servicio. **Ejemplo 1:** el siguiente código utiliza un separador de archivos codificados para abrir un archivo:

```
...
FileStream f = File.Create(directoryName + "\\" + fileName);
...
```

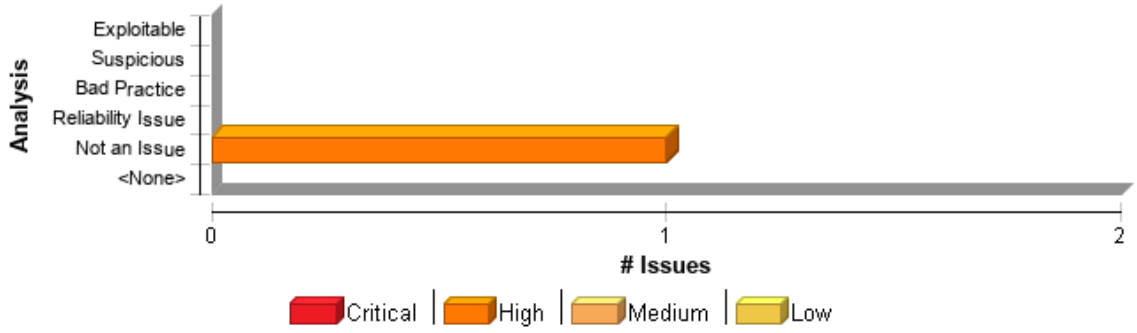
Recommendation

Con el fin de escribir código portable, evite el uso de separadores de archivos codificados. En su lugar, utilice las interfaces de programación de aplicaciones (API, Application Program Interface) independientes de las plataformas que proporciona la biblioteca de lenguajes. **Ejemplo 2:** El siguiente código implementa la misma funcionalidad que el `Example 1`, pero utiliza la API independiente de la plataforma para especificar un separador de archivos:

```
...
FileStream f = File.Create(directoryName +
Path.DirectorySeparatorChar.ToString() + fileName);
...
```

Issue Summary

Figura 90. *Gráfico de vulnerabilidades de categoría Portability Flaw: File Separator.*



Fuente: Elaboración propia.

Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Portability Flaw: File Separator	1	0	0	1
Total	**1**	**0**	**0**	**1**

Portability Flaw: File Separator	**High**

Package: Framework.Localization

Tools.cs, line 195 (Portability Flaw: File Separator) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Tue Dec 07 2021 20:49:19 GMT+0100 (CET)

Esta app se usa únicamente en Windows, lo que no implica a error en portabilidad

Source Details

Source: Read

From: Framework.Localization.Tools.ExistAnyMultilanguageFile

File: Tools.cs:192

```
189  try
190  {
191
192  string path = AppDomain.CurrentDomain.BaseDirectory + @"\Literals\";
193  if (Directory.Exists (path))
194  {
195      string[]    files    =    Directory.GetFiles    (path,    "*.xml",
SearchOption.AllDirectories);
```

Sink Details

Sink: System.IO.Directory.GetFiles()

Enclosing Method: ExistAnyMultilanguageFile()

File: Tools.cs:195

Taint Flags: FILESEPARATOR

No snippet available

Privacy Violation: Heap Inspection (7 issues)

Abstract

Mediante la inspección del montón es posible extraer datos confidenciales que se almacenen de forma no segura.

Explanation

Se pueden producir fugas de datos confidenciales (por ejemplo, contraseñas, números de la seguridad social, números de tarjetas de crédito, etc.) almacenados en la memoria si estos se han guardado en un objeto `String` administrado. Los objetos `String` no están fijos, por lo que el recopilador de elementos no utilizados puede cambiar la ubicación de los mismos como desee y dejar varias copias en la memoria. Estos objetos no se cifran de forma predeterminada, por lo que cualquier usuario pueda leer la memoria del proceso será capaz de ver el contenido. Además, si la memoria del proceso se puede intercambiar en el disco, el contenido no cifrado de la cadena se escribirá en un archivo de intercambio. Por último, como los objetos `String` son inmutables, la eliminación del valor de un objeto `String` de la memoria solo la puede realizar el recopilador de elementos no utilizados CLR. No es necesario ejecutar el recopilador de elementos no utilizados a menos que CLR disponga de poca memoria, por lo que no hay ninguna garantía en cuanto al momento en que se realizará la recopilación de elementos no utilizados. Si se bloquea la aplicación, el volcado de memoria de la misma puede mostrar información confidencial. **Ejemplo 1:** el siguiente método devuelve una contraseña desde la consola y la almacena en un objeto `String` poco seguro.

```
public static String getPassword() {
    String inputPassword = "";
    ConsoleKeyInfo nextKey = Console.ReadKey(true);
    while (nextKey.Key != Console.ReadKey(true)) {
        inputPassword.AppendChar(nextKey.KeyChar);
        Console.Write("*");
        nextKey = Console.ReadKey(true);
    }
    return inputPassword;
}
```

Recommendation

En lugar de almacenar datos sensibles en objetos como `String`s, almacénelos en una matriz que puede anularse o encriptarse en la memoria. **Ejemplo 2:** El siguiente método toma una `Action` que transfiere un parámetro como una matriz.

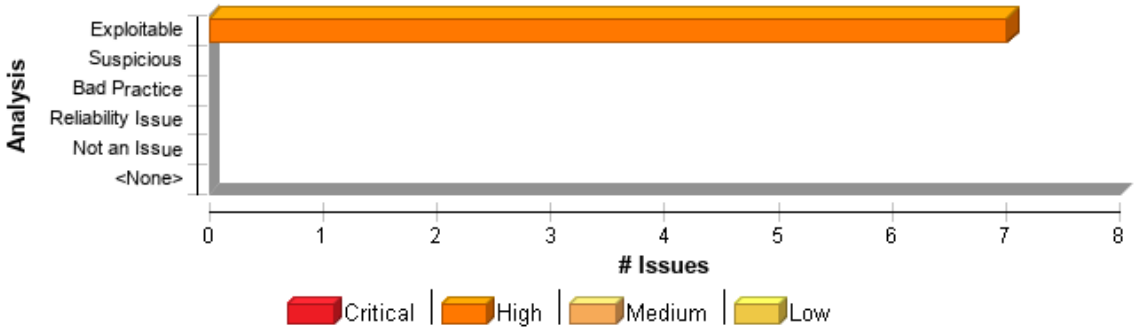
```
static void GetPasswordAndDo(Action<Char?[]> ActionWithPassword)
{
    var MAX_PASSWORD_LENGTH = 60;
    var currentIndex = 0;
    var password = new Char?[MAX_PASSWORD_LENGTH];
     Console.WriteLine("Please enter your password:");
    ConsoleKeyInfo nextKey = Console.ReadKey(true);
    while (nextKey.Key != ConsoleKey.Escape && nextKey.Key !=
ConsoleKey.Enter && currentIndex < MAX_PASSWORD_LENGTH)
    {
```

```
            password[currentIndex] = nextKey.KeyChar;
            currentIndex++;
            Console.Write("*");
            nextKey = Console.ReadKey(true);
        }
        ActionWithPassword(password);
        Array.Fill(password, null);
}
```

El **ejemplo 2** permite a un desarrollador transferir una lambda que recibe una contraseña como una matriz, que se anula al retornar la lambda. Esta solución solo mitiga parcialmente el problema, ya que el montón todavía podría inspeccionarse durante la ejecución de `ActionWithPassword`, pero reduce el tiempo de aprovechamiento. Si se espera que la ejecución de `ActionWithPassword` llevará mucho tiempo (o resultará en una excepción no detectada), se aumenta la posibilidad de comprobar la memoria y, por lo tanto, encriptar la matriz en tiempo de ejecución podría ser una solución más conveniente.

Issue Summary

Figura 91. *Gráfico de vulnerabilidades de categoría Privacy Violation: Heap Inspection.*



Fuente: Elaboración propia.

Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Privacy Violation: Heap Inspection	7	0	0	7
Total	**7**	**0**	**0**	**7**

Privacy Violation: Heap Inspection	**High**
Package: Framework.Configuration	
ConfigDatabase.cs, line 268 (Privacy Violation: Heap Inspection)	**High**
Issue Details	

Kingdom: Security Features

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Exploitable

Audit Comments

MAS: Wed Dec 08 2021 11:53:04 GMT+0100 (CET)

El objeto myConnString almacena datos confidenciales (password) como Strings

MAS: Wed Dec 08 2021 11:55:15 GMT+0100 (CET)

La password está hardcodeada en código fuente en la línea 385

MAS: Wed Dec 08 2021 11:55:25 GMT+0100 (CET)

Viene de la variable pName como parámetro

Source Details

Source: Read myUser.Password

From: Framework.Configuration.ConfigDatabase.GetFhrDatConnectionString

File: ConfigDatabase.cs:265

```
262  myConnString.Append(";user=");
263  myConnString.Append(myUser.User);
264  myConnString.Append(";password=");
265  myConnString.Append(myUser.Password);
266  myConnString.Append(";");
267
268  strResult = myConnString.ToString();
```

Sink Details

Sink: Assignment to strResult

Enclosing Method: GetFhrDatConnectionString()

File: ConfigDatabase.cs:268

Taint Flags: PRIVATE

No snippet available

ConfigDatabase.cs, line 268 (Privacy Violation: Heap Inspection) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Exploitable

Audit Comments

MAS: Wed Dec 08 2021 11:54:07 GMT+0100 (CET)

La password está hardcodeada en código fuente en la línea 385

MAS: Wed Dec 08 2021 11:54:24 GMT+0100 (CET)

Viene de la variable pName como parámetro

Source Details

Source: Read xNodePassword

From: Framework.Configuration.ConfigDatabase.GetUser

File: ConfigDatabase.cs:390

```
387
388  if (xNodeUser != null && xNodePassword != null)
389  {
390      myUser = new UserCredentials(xNodeUser.InnerText,
xNodePassword.InnerText);
391  }
392
393  return myUser;
```

Sink Details

Sink: Assignment to strResult

Enclosing Method: GetFhrDatConnectionString()

File: ConfigDatabase.cs:268

Taint Flags: PRIVATE

No snippet available

ConfigDatabase.cs, line 268 (Privacy Violation: Heap Inspection) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Exploitable

Audit Comments

MAS: Wed Dec 08 2021 11:54:56 GMT+0100 (CET)

La password está hardcodeada en código fuente en la línea 385

MAS: Wed Dec 08 2021 11:55:05 GMT+0100 (CET)

Viene de la variable pName como parámetro

Source Details

Source: Read pPassword

From: Framework.Configuration.ConfigDatabase.UserCredentials.UserCredentials

File: ConfigDatabase.cs:410

```
407  public UserCredentials(string pUser, string pPassword)
408  {
409  this.User = pUser;
410  this.Password = pPassword;
411  }
412  }
413  }
```

Sink Details

Sink: Assignment to strResult

Enclosing Method: GetFhrDatConnectionString()

File: ConfigDatabase.cs:268

Taint Flags: PRIVATE

No snippet available

Package: Framework.Configuration.nConfig

ConfigDatabase.cs, line 211 (Privacy Violation: Heap Inspection) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Exploitable

Audit Comments

MAS: Wed Dec 08 2021 11:33:12 GMT+0100 (CET)

El objeto myConnString almacena datos confidenciales (password) como Strings

MAS: Wed Dec 08 2021 11:40:08 GMT+0100 (CET)

La password viene de la línea 328

MAS: Wed Dec 08 2021 11:40:11 GMT+0100 (CET)

Viene de la variable pName como parámetro

Source Details

Source: Read myUser.Password

From: Framework.Configuration.nConfig.ConfigDatabase.GetFhrDatConnectionString

File: ConfigDatabase.cs:208

```
205  myConnString.Append(";user=");
206  myConnString.Append(myUser.User);
207  myConnString.Append(";password=");
208  myConnString.Append(myUser.Password);
209  myConnString.Append(";");
210
211  strResult = myConnString.ToString();
```

Sink Details

Sink: Assignment to strResult

Enclosing Method: GetFhrDatConnectionString()

File: ConfigDatabase.cs:211

Taint Flags: PRIVATE

No snippet available

ConfigDatabase.cs, line 211 (Privacy Violation: Heap Inspection) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Exploitable

Audit Comments

MAS: Wed Dec 08 2021 11:36:05 GMT+0100 (CET)

La password está hardcodeada en código fuente en la línea 328

MAS: Wed Dec 08 2021 11:39:42 GMT+0100 (CET)

Viene de la variable pName como parámetro

Source Details

Source: Read xNodePassword

From: Framework.Configuration.nConfig.ConfigDatabase.GetUser

File: ConfigDatabase.cs:332

```
329  UserCredentials myUser = null;
```

```
330
331  if (xNodeUser != null && xNodePassword != null)
332        myUser       =       new       UserCredentials(xNodeUser.InnerText,
xNodePassword.InnerText);
333
334  return myUser;
335  }
```

Sink Details

Sink: Assignment to strResult

Enclosing Method: GetFhrDatConnectionString()

File: ConfigDatabase.cs:211

Taint Flags: PRIVATE

No snippet available

ConfigDatabase.cs, line 211 (Privacy Violation: Heap Inspection) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Exploitable

Audit Comments

MAS: Wed Dec 08 2021 11:36:52 GMT+0100 (CET)

La password está hardcodeada en la línea 328

MAS: Wed Dec 08 2021 11:39:47 GMT+0100 (CET)

Viene de la variable pName como parámetro

Source Details

Source: Read pPassword

From: Framework.Configuration.nConfig.ConfigDatabase.UserCredentials.UserCredentials

File: ConfigDatabase.cs:351

```
348  public UserCredentials(string pUser, string pPassword)
349  {
350  this.User = pUser;
351  this.Password = pPassword;
352  }
```

```
353   }
354   }
```

Sink Details

Sink: Assignment to strResult

Enclosing Method: GetFhrDatConnectionString()

File: ConfigDatabase.cs:211

Taint Flags: PRIVATE

No snippet available

Package: Framework.Tools

Cryptographer.cs, line 211 (Privacy Violation: Heap Inspection) **High**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Exploitable

Audit Comments

MAS: Wed Dec 08 2021 11:30:53 GMT+0100 (CET)

La clave sha no se limpia del buffer después de su uso

Sink Details

Sink: sha = new SHA1CryptoServiceProvider() : Key algorithm initialized

Enclosing Method: getHash()

File: Cryptographer.cs:211

Taint Flags:

No snippet available

Process Control (5 issues)

Abstract

La carga de bibliotecas o de ejecutables desde un origen o un entorno que no son de confianza puede ocasionar que una aplicación ejecute comandos malintencionados en nombre de un atacante.

Explanation

Las vulnerabilidades de control del proceso presentan dos formas: - Un atacante podría cambiar el nombre de la biblioteca o del ejecutable que carga el programa: el atacante controla de manera explícita

cuál es el nombre de la biblioteca o del ejecutable. - Un atacante podría cambiar el entorno en el que se carga la biblioteca o el ejecutable: el atacante controla de manera implícita lo que significa el nombre de la biblioteca o el ejecutable. En este caso, nos preocupa principalmente el primer escenario, la posibilidad de que un usuario malintencionado pueda controlar el nombre de la biblioteca que se ha cargado. Las vulnerabilidades de control del proceso de este tipo se producen cuando: 1. Los datos entran en la aplicación desde una fuente no confiable. 2. Los datos se utilizan como una cadena o como parte de una cadena, la cual representa una biblioteca o un ejecutable que la aplicación carga. 3. Al ejecutar el código desde la biblioteca o el ejecutable, la aplicación concede al atacante un privilegio o una capacidad que no tendría de otro modo. **Ejemplo 1:** el siguiente código de una utilidad de sistema con privilegios utiliza la propiedad de configuración de aplicación `APPHOME` y, a continuación, carga una biblioteca nativa en función de una ruta de acceso relativa desde el directorio especificado.

```
...
string lib = ConfigurationManager.AppSettings["APPHOME"];
Environment.ExitCode = AppDomain.CurrentDomain.ExecuteAssembly(lib);
...
```

Este código permite a un atacante cargar una biblioteca o un ejecutable y posiblemente ejecutar código arbitrario con el privilegio elevado de la aplicación mediante la modificación de la propiedad `APPHOME` de la aplicación, con el objetivo de que apunte a una ruta diferente que contenga una versión malintencionada de `LIBNAME`. Como el programa no valida el valor leído en el entorno, si el atacante puede controlar el valor de la propiedad del sistema `APPHOME`, puede engañar a la aplicación para que ejecute código malintencionado y asumir el control del sistema.

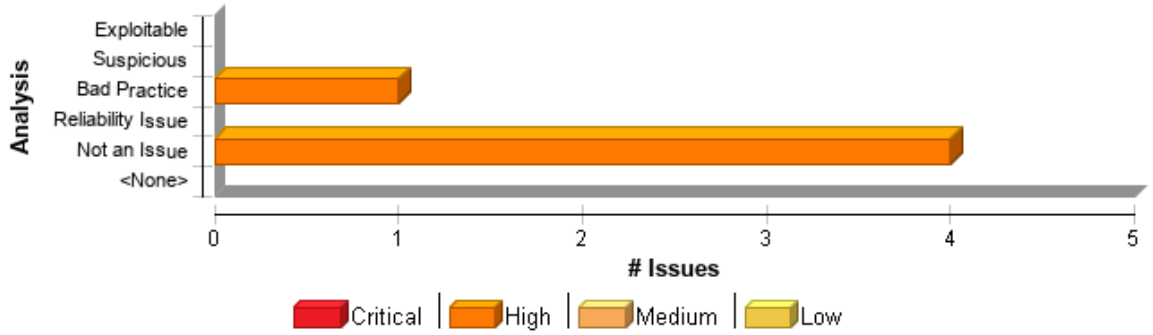
Recommendation

No permita que los usuarios controlen las bibliotecas o los ejecutables cargados por el programa. En los casos en los que la entrada de usuario debe influir en la selección de las bibliotecas o los ejecutables que se van a cargar, usualmente la aplicación espera una entrada determinada para trabajar con solo un conjunto muy reducido de valores. En lugar de usar la entrada para garantizar la seguridad y evitar ataques malintencionados, la aplicación solo debe usar la entrada para realizar una selección a partir de un conjunto predeterminado de bibliotecas o ejecutables seguros. Si la entrada parece ser malintencionada, la biblioteca o el ejecutable que se cargarán deben utilizar una selección segura desde este conjunto de forma predeterminada o el programa debe impedir de forma estable que la operación continúe. Puede que un atacante controle de forma indirecta las bibliotecas o los ejecutables cargados por un programa mediante la modificación del entorno. No se debe confiar en el entorno y deberán tomarse precauciones para evitar que un usuario malintencionado manipule el entorno para realizar un ataque. En casos en los que no se conozca la ruta en el momento de la compilación, debe crearse una ruta absoluta a partir de los valores de confianza durante la ejecución. Se debe comprobar la integridad de los nombres y las rutas de acceso de las bibliotecas o los ejecutables leídos en el entorno en relación con un conjunto de elementos invariables que definen a los valores válidos. En ocasiones, pueden realizarse otras comprobaciones para detectar si el entorno puede haber sido manipulado. Por ejemplo,

si un archivo de configuración tiene permisos de escritura para todos los usuarios, el programa puede negarse a realizar la ejecución. En aquellos casos en los que se conozca por adelantado la información de la biblioteca que se va a cargar, el programa puede realizar comprobaciones para verificar la identidad del archivo. Si una biblioteca debe pertenecer siempre a un determinado usuario o tener un determinado conjunto de permisos asignado a ella, estas propiedades pueden comprobarse mediante programación antes de que se cargue esta. En último término, puede que resulte imposible proteger un programa por completo de atacantes creativos que estén decididos a controlar las bibliotecas o los ejecutables que carga. Debe intentar identificar cada manipulación que se pueda imaginar de los valores de entrada y el entorno, y protegerse de ella. El objetivo debe ser parar tantos tipos de ataque como sea posible.

Issue Summary

Figura 92. *Gráfico de vulnerabilidades de categoría Process Control.*



Fuente: Elaboración propia.

Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Process Control	5	0	0	5
Total	**5**	**0**	**0**	**5**

Process Control	**High**
Package: Framework	
Interceptor.cs, line 110 (Process Control)	**High**
Issue Details	

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details	
Analysis	Not an Issue
Audit Comments	

MAS: Wed Dec 08 2021 13:18:19 GMT+0100 (CET)

Se extrae el nombre del argumento de entrada

Source Details

Source: System.IO.FileStream.FileStream()

From: Framework.FileSystemStorage.GetData

File: FileSystemStorage.cs:26

```
23  string fullName = ResolveName (fname);
24  if (string.IsNullOrWhiteSpace (fullName) == false)
25  {
26    using (var strm = new FileStream (fullName, FileMode.Open,
FileAccess.Read))
27  {
28  result = new byte[strm.Length];
29  var res = strm.Read (result, 0, (int)strm.Length);
```

Sink Details

Sink: System.Reflection.Assembly.Load()

Enclosing Method: CurrentDomain_AssemblyResolve()

File: Interceptor.cs:110

Taint Flags: FILE_SYSTEM, NUMBER

No snippet available

Interceptor.cs, line 110 (Process Control) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Wed Dec 08 2021 13:18:24 GMT+0100 (CET)

Se extrae el nombre del argumento de entrada

Source Details

Source: System.IO.BinaryReader.ReadBytes()

From: Framework.CompoundFile.GetData

File: CompoundFile.cs:

2422

```
2419
2420  BinaryReader br = new BinaryReader(miniView);
2421
2422  result = br.ReadBytes((int)de.Size);
2423  br.Close();
2424
2425  }
```

Sink Details

Sink: System.Reflection.Assembly.Load()

Enclosing Method: CurrentDomain_AssemblyResolve()

File: Interceptor.cs:110

Taint Flags: NUMBER, STREAM

No snippet available

Interceptor.cs, line 110 (Process Control) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Wed Dec 08 2021 13:18:16 GMT+0100 (CET)

Se extrae el nombre del argumento de entrada

Source Details

Source: System.IO.FileStream.Read()

From: Framework.FileSystemStorage.GetData

File: FileSystemStorage.cs:29

```
26  using (var strm = new FileStream (fullName, FileMode.Open,
FileAccess.Read))
27  {
28  result = new byte[strm.Length];
29  var res = strm.Read (result, 0, (int)strm.Length);
30  if (res != strm.Length)
```

```
31  {
32  logger.Error ($"Error loading file [{fullName}]");
```

Sink Details

Sink: System.Reflection.Assembly.Load()

Enclosing Method: CurrentDomain_AssemblyResolve()

File: Interceptor.cs:110

Taint Flags: NUMBER, STREAM

No snippet available

Interceptor.cs, line 110 (Process Control) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Wed Dec 08 2021 13:18:22 GMT+0100 (CET)

Se extrae el nombre del argumento de entrada

Source Details

Source: Framework.StreamView.Read()

From: Framework.CompoundFile.GetData

File: CompoundFile.cs:2433

```
2430
2431  result = new byte[(int)de.Size];
2432
2433  sView.Read(result, 0, result.Length);
2434
2435  }
2436
```

Sink Details

Sink: System.Reflection.Assembly.Load()

Enclosing Method: CurrentDomain_AssemblyResolve()

File: Interceptor.cs:110

Taint Flags: NUMBER, STREAM

No snippet available

Package: Framework.Plugins.Editor

EditPluginViewModel.cs, line 194 (Process Control) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Bad Practice

Audit Comments

MAS: Wed Dec 08 2021 13:15:10 GMT+0100 (CET)

De busca un fichero de configuración concreto que puede ser sustituido por otro y ser atacado

Source Details

Source: System.Runtime.Serialization.Formatters.Binary.BinaryFormatter.Deserialize()

From: Framework.Tools.CloneUtils.Clone

File: CloneUtils.cs:50

```
47  {
48  bf.Serialize (memStream, item);
49  memStream.Seek (0, SeekOrigin.Begin);
50  result = (T)bf.Deserialize (memStream);
51  }
52  return result;
53  }
```

Sink Details

Sink: System.Reflection.Assembly.LoadFrom()

Enclosing Method: GetAssemblyTypes()

File: EditPluginViewModel.cs:194

Taint Flags: SERIALIZED

No snippet available

Unreleased Resource: LDAP (2 issues)

Abstract

Es posible que el programa no pueda liberar un recurso de LDAP.

Explanation

Es posible que el programa no pueda liberar un recurso de LDAP. Las pérdidas de recursos presentan dos causas habituales: - Condiciones de error y otras circunstancias excepcionales. - Confusión en cuanto a la parte del programa responsable de liberar el recurso. La mayoría de los problemas de recursos no liberados provocan problemas generales de confiabilidad del software. Sin embargo, si un usuario malintencionado puede activar de forma intencionada una pérdida de recursos, es posible que este pueda iniciar un ataque de denegación de servicio agotando el conjunto de recursos. **Ejemplo:** en condiciones normales, el siguiente código ejecuta una consulta LDAP, procesa los resultados devueltos por Active Directory y cierra el objeto `DirectoryEntry` asignado. Sin embargo, si se produce una excepción al ejecutar la consulta LDAP o al procesar los resultados, el objeto `DirectoryEntry` no se cerrará. Esto introducirá una pérdida de memoria en la aplicación, ya que `DirectoryEntry` usa las API de COM internamente para realizar consultas en el servidor de Active Directory.

```
  ...
  DirectoryEntry entry = new
DirectoryEntry("LDAP://CN=users,DC=fabrikam,DC=com");
  DirectorySearcher mySearcher = new DirectorySearcher(entry);
  SearchResultCollection result = mySearcher.FindAll();
  CheckUsers(result);
  mySearcher.Dispose();
  entry.Close();
  ...
```

Recommendation

No utilice nunca `Finalize()` para reclamar recursos. Para que pueda llamarse a un método `Finalize()` de un objeto, el recopilador de elementos no utilizados determina si el objeto reúne los requisitos para la recopilación de elementos no utilizados. Como no es necesario ejecutar el recopilador de elementos no utilizados a menos que la VM tenga poca memoria, no hay garantía de que se llame al método `Finalize()` del objeto de forma oportuna, si es que alguna vez se lo invoca (el lenguaje no garantiza que se realice esta acción). Cuando se ejecute finalmente el recopilador de elementos no utilizados, esto puede provocar que se reclame una gran cantidad de recursos en un corto periodo de tiempo, lo que puede dar lugar a un repentino aumento del rendimiento y a una reducción de la capacidad del sistema. El efecto es cada vez más pronunciado a medida que aumenta la carga en el sistema. En lugar de cerrar de forma explícita los objetos que administran recursos, utilice la palabra de C# "using", que implementa la interfaz `IDisposable` para realizar una limpieza. Los siguientes dos bloques de código obtienen los mismos resultados: El siguiente código utiliza la palabra clave `finally`:

```
DirectoryEntry entry = null;
 try
```

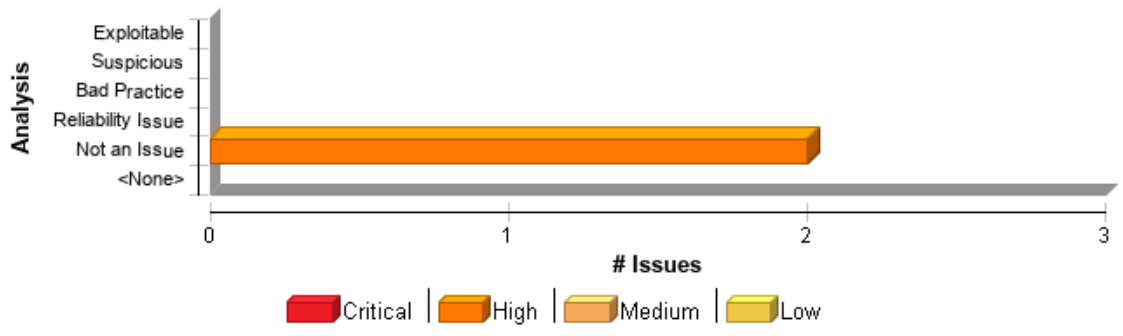
```
{
  entry = new DirectoryEntry("LDAP://CN=users,DC=fabrikam,DC=com");
  DirectorySearcher mySearcher = null;
  try
  {
    mySearcher = new DirectorySearcher(entry);
    SearchResultCollection result = mySearcher.FindAll();
    CheckUsers(result);
  }finally
  {
    if (mySearcher != null) { mySearcher.Dispose(); }
  }
}finally
{
  if (entry != null) { entry.Close(); }
}
```

El siguiente código utiliza la palabra clave `using`:

```
  using (DirectoryEntry entry = new
DirectoryEntry("LDAP://CN=users,DC=fabrikam,DC=com"))
  {
    using (DirectorySearcher mySearcher = new DirectorySearcher(entry))
    {
        SearchResultCollection result = mySearcher.FindAll();
        CheckUsers(result);
    }
  }
```

Issue Summary

Figura 93. *Gráfico de vulnerabilidades de categoría Unreleased Resource: LDAP.*



Fuente: Elaboración propia.

Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Unreleased Resource: LDAP	2	0	0	2
Total	**2**	**0**	**0**	**2**

Unreleased Resource: LDAP — **High**

Package: Framework.Tools

LDAPManager.cs, line 689 (Unreleased Resource: LDAP) — **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Tue Dec 07 2021 21:05:24 GMT+0100 (CET)

No se libera la variable search

Sink Details

Sink: search = new DirectorySearcher(...)

Enclosing Method: GetGruposLDAP()

File: LDAPManager.cs:689

Taint Flags:

```
No snippet available
```

LDAPManager.cs, line 326 (Unreleased Resource: LDAP) — **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Tue Dec 07 2021 21:05:28 GMT+0100 (CET)

No se libera la variable search

Sink Details

Sink: search = new DirectorySearcher(...)

Enclosing Method: IsAuthenticated()

File: LDAPManager.cs:326

Taint Flags:

No snippet available

Unreleased Resource: Streams (4 issues)

Abstract

Es posible que el programa no pueda liberar un recurso del sistema.

Explanation

Es posible que el programa no pueda liberar un recurso del sistema. Las pérdidas de recursos presentan dos causas habituales: - Condiciones de error y otras circunstancias excepcionales. - Confusión en cuanto a la parte del programa responsable de liberar el recurso. La mayoría de los problemas de recursos no liberados provocan problemas generales de confiabilidad del software. Sin embargo, si un usuario malintencionado puede activar de forma intencionada una pérdida de recursos, es posible que este pueda iniciar un ataque de denegación de servicio agotando el conjunto de recursos. **Ejemplo:** el siguiente método nunca cierra el identificador de archivo que abre. El método `Finalize()` de `StreamReader` con el tiempo llama a `Close()`, pero no hay ninguna garantía en cuanto el tiempo que pasará antes de que se llame al método `Finalize()`. De hecho, no hay ninguna garantía de que se llame en algún momento al método `Finalize()`. En un entorno muy activo, esto puede provocar que la VM utilice todos los identificadores de archivo disponibles.

```
private void processFile(string fName) {
        StreamWriter sw = new StreamWriter(fName);
        string line;
        while ((line = sr.ReadLine()) != null)
                processLine(line);
}
```

Recommendation

No utilice nunca `Finalize()` para reclamar recursos. Para que pueda llamarse a un método `Finalize()` de un objeto, el recopilador de elementos no utilizados determina si el objeto reúne los requisitos para la recopilación de elementos no utilizados. Como no es necesario ejecutar el recopilador de elementos no utilizados a menos que la VM tenga poca memoria, no hay garantía de que se llame al método `Finalize()` del objeto de forma oportuna, si es que alguna vez se lo invoca (el lenguaje no garantiza que se realice esta acción). Cuando se ejecute finalmente el recopilador de elementos no utilizados, esto puede provocar que se reclame una gran cantidad de recursos en un corto periodo de tiempo, lo que puede dar lugar a un repentino aumento del rendimiento y a una reducción de la capacidad del sistema. El efecto es cada vez más pronunciado a medida que aumenta la carga en el sistema. En lugar de cerrar de forma explícita los objetos que administran recursos, utilice la palabra

de C# "using", que implementa la interfaz `IDisposable` para realizar una limpieza. Los siguientes dos bloques de código obtienen los mismos resultados: El siguiente código utiliza la palabra clave `finally`:

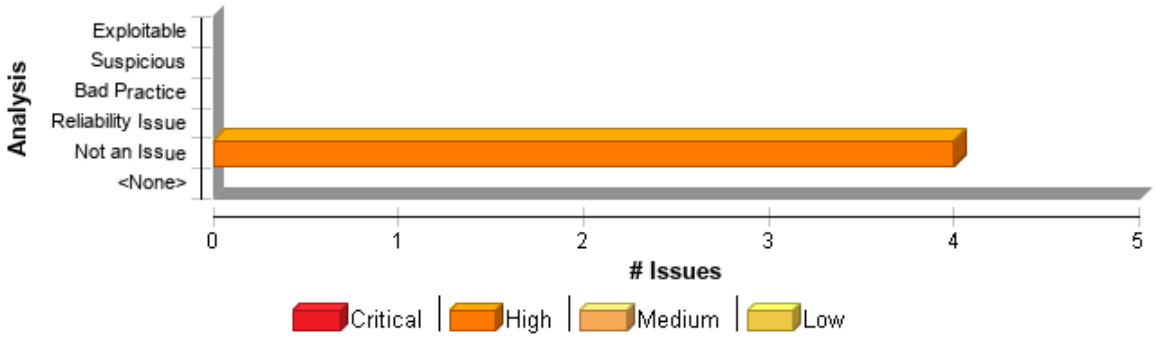
```
StreamReader sr;
try {
  sr = new StreamReader(myFileStream);
  doWork(sr);
} finally {
  if (sr != null) {
    sr.Close();
  }
}
```

El siguiente código utiliza la palabra clave `using`:

```
  using (StreamReader sr = new StreamReader(myFileStream)) {
    doWork(sr);
  }
```

Issue Summary

Figura 94. *Gráfico de vulnerabilidades de categoría Unreleased Resource: Streams.*



Fuente: Elaboración propia.

Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Unreleased Resource: Streams	4	0	0	4
Total	**4**	**0**	**0**	**4**

Unreleased Resource: Streams	**High**
Package: Framework.Tools	
Cryptographer.cs, line 171 (Unreleased Resource: Streams)	**High**
Issue Details	

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Tue Dec 07 2021 21:14:23 GMT+0100 (CET)

No se libera la variable ms

MAS: Tue Dec 07 2021 21:14:42 GMT+0100 (CET)

No se liberan sr y cs

Sink Details

Sink: sr = new StreamReader(cs, ...)

Enclosing Method: Decrypt()

File: Cryptographer.cs:171

Taint Flags:

```
No snippet available
```

Cryptographer.cs, line 90 (Unreleased Resource: Streams) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Tue Dec 07 2021 21:13:24 GMT+0100 (CET)

No se libera la variable ms

MAS: Tue Dec 07 2021 21:14:55 GMT+0100 (CET)

No se libera cs

Sink Details

Sink: cs = new CryptoStream(...)

Enclosing Method: Crypt()

File: Cryptographer.cs:90

Taint Flags:

```
No snippet available
```

Package: Framework.CommsLink

TCPCommsLink.cs, line 729 (Unreleased Resource: Streams) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Tue Dec 07 2021 21:17:56 GMT+0100 (CET)

No se libera la varibale client

Sink Details

Sink: ?.InitializeClientInfo(client)

Enclosing Method: InitializeTCP()

File: TCPCommsLink.cs:729

Taint Flags:

```
No snippet available
```

Package: Framework.Services.Discovery

DiscoveryClientTask.cs, line 106 (Unreleased Resource: Streams) **High**

Issue Details

Kingdom: Code Quality

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Tue Dec 07 2021 21:16:54 GMT+0100 (CET)

No se libera la variable receiver

Sink Details

Sink: receiver = MakeUdpReceiver(...)

Enclosing Method: Setup()

File: DiscoveryClientTask.cs:106

Taint Flags:

```
No snippet available
```

Unsafe Native Invoke (9 issues)

Abstract

El uso inadecuado de los servicios de invocación de plataforma puede provocar que las aplicaciones administradas sean vulnerables a los fallos de seguridad en otros lenguajes.

Explanation

Se producen errores de invocación nativa poco segura cuando una aplicación administrada utiliza P/Invoke para llamar al código (no administrado) nativo en otro lenguaje de programación. **Ejemplo 1:** el siguiente código C# define una clase denominada `Echo`. La clase declara un método nativo que utiliza C para devolver comandos introducidos en la consola al usuario.

```
class Echo
{
  [DllImport("mylib.dll")]
  internal static extern void RunEcho();
   static void main(String[] args)
  {
    RunEcho();
  }
}
```

El siguiente código C define el método nativo implementado en la clase `Echo`:

```
#include <stdio.h>
 void __stdcall RunEcho()
{
  char* buf = (char*) malloc(64 * sizeof(char));
  gets(buf);
  printf(buf);
}
```

Como el eco se ha implementado en el código administrado, podría parecer que este es inmune a problemas de memoria, como las vulnerabilidades de buffer overflow. Aunque el entorno administrado protege de forma eficaz las operaciones de memoria, esta protección no se extiende a las vulnerabilidades que se producen en el código nativo al que se accede mediante P/Invoke. A pesar de las protecciones de la memoria ofrecidas por el entorno de tiempo de ejecución administrado, el código nativo de este ejemplo es vulnerable al buffer overflow debido a que utiliza `gets()`, que no lleva a cabo ninguna comprobación de límites en su entrada. Asimismo, `buf` se asigna, pero no se libera, por lo que produce una fuga de memoria. La vulnerabilidad presente en el `Example 1` podría detectarse fácilmente a través de una auditoría de código fuente de la implementación del método nativo. Es posible que esta operación no sea factible o posible en función de la disponibilidad del código fuente y la forma en que se ha creado el proyecto, aunque, en la mayoría de los casos, este método es suficiente. Sin embargo, la capacidad para compartir objetos entre los entornos administrado y nativo amplía el riesgo potencial de que se produzcan casos más insidiosos cuando una administración inadecuada de los datos en el código

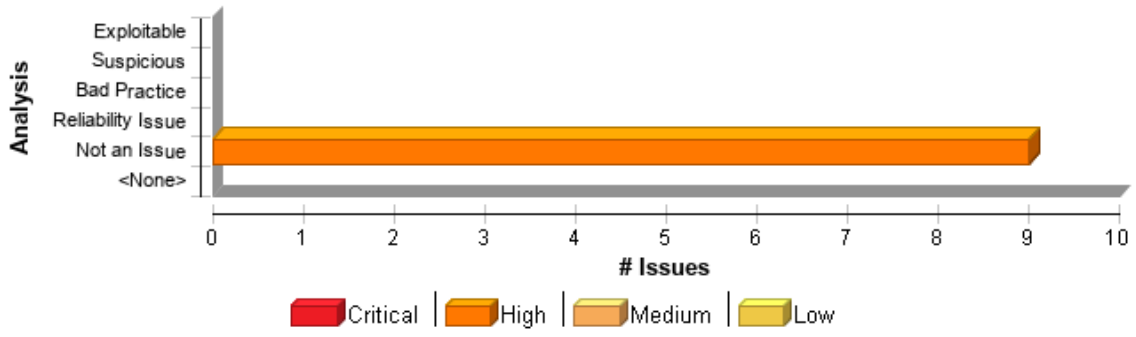
administrado puede provocar vulnerabilidades inesperadas u operaciones que no son seguras en el código nativo, lo que daña las estructuras de datos del código administrado. Las vulnerabilidades del código nativo al que se accede a través de una aplicación administrada se explotan normalmente del mismo modo que en las aplicaciones escrita en el lenguaje nativo. El único reto que se le plantea al usuario malintencionado en relación con un ataque de este tipo consiste en identificar que la aplicación administrada utiliza código nativo para realizar determinadas operaciones. Esto puede lograrse de diversas formas, incluida la identificación de los comportamientos específicos que a menudo se implementan con el código nativo o explorando una fuga de información del sistema en la aplicación administrada que deje al descubierto su uso de P/Invoke.

Recommendation

Realice una auditoría de todo el código fuente que compone una determinada aplicación, incluidos los métodos nativos implementados en código nativo. Durante las auditorías, asegúrese de que las diferencias en las comprobaciones de límites y otro comportamiento entre el código administrado y nativo se tienen en cuenta y se gestionan correctamente. En concreto, compruebe que los objetos compartidos se administran correctamente en todas las etapas: antes de transferirlos al código nativo, mientras este los manipula, y después de que se devuelvan a la aplicación administrada.

Issue Summary

Figura 95. *Gráfico de vulnerabilidades de categoría Unsafe Native Invoke.*



Fuente: Elaboración propia.

Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Unsafe Native Invoke	9	0	0	9
Total	**9**	**0**	**0**	**9**

Unsafe Native Invoke	**High**
Package: Framework.Tools	
CommonConsole.cs, line 640 (Unsafe Native Invoke)	**High**
Issue Details	

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Wed Dec 08 2021 12:42:21 GMT+0100 (CET)

El método OutputDebugString se usa correctamente

Source Details

Source: System.Environment.GetCommandLineArgs()

From: Framework.Tools.Tracer.GetExePath

File: Tracer.cs:959

```
956
957  private static string GetExePath ()
958  {
959   string dir = Path.GetDirectoryName (Environment.GetCommandLineArgs
()[0]);
960  if (string.IsNullOrWhiteSpace (dir))
961  {
962  dir = Directory.GetCurrentDirectory ();
```

Sink Details

Sink: Framework.Tools.CommonConsole.OutputDebugString()

Enclosing Method: OutputToDebugger()

File: CommonConsole.cs:640

Taint Flags: ARGS, VALIDATED_PATH_MANIPULATION_BASE_PATH_OVERWRITING

No snippet available

CommonConsole.cs, line 640 (Unsafe Native Invoke) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Wed Dec 08 2021 12:42:12 GMT+0100 (CET)

El método OutputDebugString se usa correctamente

Source Details

Source: System.Environment.ExpandEnvironmentVariables()

From: Framework.Tools.Tracer.GetTempPath

File: Tracer.cs:947

```
944
945  private static string GetTempPath ()
946  {
947  return Environment.ExpandEnvironmentVariables ("%temp%");
948  }
949
950  private static string GetUserPath ()
```

Sink Details

Sink: Framework.Tools.CommonConsole.OutputDebugString()

Enclosing Method: OutputToDebugger()

File: CommonConsole.cs:640

Taint Flags: ENVIRONMENT, VALIDATED_PATH_MANIPULATION_BASE_PATH_OVERWRITING

No snippet available

Package: Framework.Tools.Cryptographer

Cryptographer.cs, line 326 (Unsafe Native Invoke) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Wed Dec 08 2021 12:42:40 GMT+0100 (CET)

El método CryptEncrypt se usa correctamente

Source Details

Source: System.IO.FileStream.FileStream()

From: Framework.Tools.Cryptographer.CryptoDllFileFromToBuffer

File: Cryptographer.cs:501

```
498  {
499  try
500  {
501      using (FileStream file = new FileStream(filename,
FileMode.OpenOrCreate, FileAccess.Read))
502  {
503  byte[] fileBuffer = new byte[file.Length];
504  file.Read(fileBuffer, 0, (int)file.Length);
```

Sink Details

Sink: Framework.Tools.Cryptographer.AdvapiCrypt.CryptEncrypt()

Enclosing Method: EncryptBytes()

File: Cryptographer.cs:326

Taint Flags: FILE_SYSTEM

No snippet available

Cryptographer.cs, line 340 (Unsafe Native Invoke) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Wed Dec 08 2021 12:43:01 GMT+0100 (CET)

El método CryptDecrypt se usa correctamente

Source Details

Source: System.IO.FileStream.Read()

From: Framework.Tools.Cryptographer.DecryptoDllFromFileToBuffer

File: Cryptographer.cs:587

```
584  }
585
586  byte[] fileBuffer = new byte[file.Length];
587  file.Read(fileBuffer, 0, (int)file.Length);
588  file.Close();
```

```
589  return DecryptoDll(fileBuffer);
590
```

Sink Details

Sink: Framework.Tools.Cryptographer.AdvapiCrypt.CryptDecrypt()

Enclosing Method: DecryptBytes()

File: Cryptographer.cs:340

Taint Flags: STREAM

No snippet available

Cryptographer.cs, line 340 (Unsafe Native Invoke) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Wed Dec 08 2021 12:43:08 GMT+0100 (CET)

El método CryptDecrypt se usa correctamente

Source Details

Source: System.IO.FileStream.Read()

From: Framework.Configuration.SystemCfgConfigurationProvider.ReadEncryptedFile

File: SystemCfgConfigurationProvider.cs:185

```
182  try
183  {
184  crypted_data = new byte[fs.Length];
185  fs.Read(crypted_data, 0, (int) fs.Length);
186  }
187  finally
188  {
```

Sink Details

Sink: Framework.Tools.Cryptographer.AdvapiCrypt.CryptDecrypt()

Enclosing Method: DecryptBytes()

File: Cryptographer.cs:340

Taint Flags: STREAM

No snippet available

Cryptographer.cs, line 340 (Unsafe Native Invoke) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Wed Dec 08 2021 12:42:57 GMT+0100 (CET)

El método CryptDecrypt se usa correctamente

Source Details

Source: System.IO.FileStream.FileStream()

From: Framework.Tools.Cryptographer.DecryptoDllFromFileToBuffer

File: Cryptographer.cs:583

```
580  }
581  else
582  {
583      file    =    new    FileStream(pFilename,    FileMode.OpenOrCreate,
FileAccess.Read);
584  }
585
586  byte[] fileBuffer = new byte[file.Length];
```

Sink Details

Sink: Framework.Tools.Cryptographer.AdvapiCrypt.CryptDecrypt()

Enclosing Method: DecryptBytes()

File: Cryptographer.cs:340

Taint Flags: FILE_SYSTEM

No snippet available

Cryptographer.cs, line 326 (Unsafe Native Invoke) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Wed Dec 08 2021 12:42:46 GMT+0100 (CET)

El método CryptEncrypt se usa correctamente

Source Details

Source: System.IO.FileStream.Read()

From: Framework.Tools.Cryptographer.CryptoDllFileFromToBuffer

File: Cryptographer.cs:504

```
501     using (FileStream file = new FileStream(filename,
FileMode.OpenOrCreate, FileAccess.Read))
502   {
503   byte[] fileBuffer = new byte[file.Length];
504   file.Read(fileBuffer, 0, (int)file.Length);
505   file.Close();
506   return CryptoDll(fileBuffer);
507   }
```

Sink Details

Sink: Framework.Tools.Cryptographer.AdvapiCrypt.CryptEncrypt()

Enclosing Method: EncryptBytes()

File: Cryptographer.cs:326

Taint Flags: STREAM

No snippet available

Cryptographer.cs, line 369 (Unsafe Native Invoke) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Wed Dec 08 2021 12:43:20 GMT+0100 (CET)

El método CryptHashData se usa correctamente

Source Details

Source: System.IO.StreamReader.ReadToEnd()

From: Framework.Tools.Cryptographer.Decrypt

File: Cryptographer.cs:175

```
172
173   try
174   {
175   resul = sr.ReadToEnd();
176   }
177   catch (Exception)
178   {
```

Sink Details

Sink: Framework.Tools.Cryptographer.AdvapiCrypt.CryptHashData()

Enclosing Method: Initialize()

File: Cryptographer.cs:369

Taint Flags: STREAM

No snippet available

Cryptographer.cs, line 340 (Unsafe Native Invoke) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Wed Dec 08 2021 12:43:05 GMT+0100 (CET)

El método CryptDecrypt se usa correctamente

Source Details

Source: System.IO.FileStream.FileStream()

From: Framework.Tools.Cryptographer.OpenExclusive

File: Cryptographer.cs:679

```
676  {
677  try
678  {
679      stream = new FileStream(fullPath, FileMode.OpenOrCreate,
FileAccess.Read, FileShare.None);
680  break;
681  }
682  catch (IOException)
```

Sink Details

Sink: Framework.Tools.Cryptographer.AdvapiCrypt.CryptDecrypt()

Enclosing Method: DecryptBytes()

File: Cryptographer.cs:340

Taint Flags: FILE_SYSTEM

No snippet available

Weak Encryption: Insecure Mode of Operation (2 issues)

Abstract

No utilice algoritmos de cifrado criptográfico con un modo de operación no seguro.

Explanation

El modo de operación de un cifrado de bloque es un algoritmo que describe cómo aplicar repetidamente una operación de un solo bloque de cifrado para transformar de forma segura cantidades de datos mayores que un bloque. Algunos modos de operación son Electronic Codebook (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB) y Counter (CTR). El modo ECB es inherentemente poco seguro, ya que produce el mismo texto cifrado para bloques idénticos de texto sin formato. El modo CBC es vulnerable a los ataques de oráculo de relleno. El modo CTR es la mejor elección, ya que no presenta estas debilidades. **Ejemplo 1:** el siguiente código utiliza un cifrado AES con el modo ECB:

```
...
var objAesCryptoService = new AesCryptoServiceProvider();
objAesCryptoService.Mode = CipherMode.ECB;
objAesCryptoService.Padding = PaddingMode.PKCS7;
objAesCryptoService.Key = securityKeyArray;
var objCrytpoTransform = objAesCryptoService.CreateEncryptor();
...
```

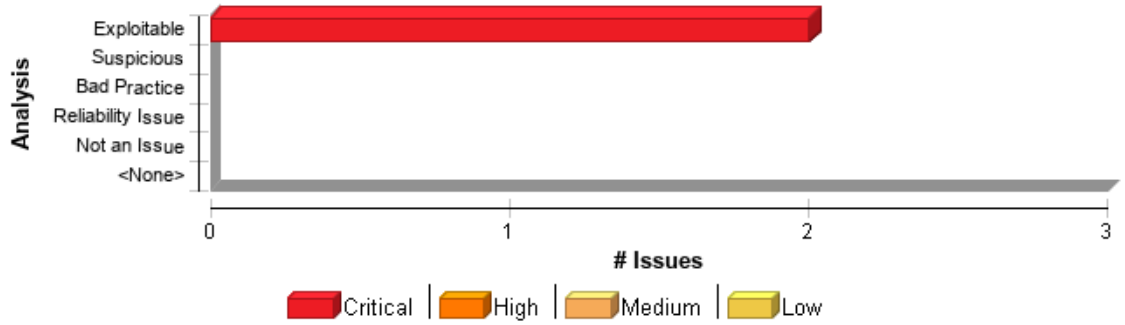
Recommendation

Evite usar los modos de operación ECB y CBC al cifrar datos de tamaño superior a un bloque. El modo CBC es algo ineficaz y supone un riesgo grave si se usa con SSL [1]. En su lugar, utilice el modo CCM (Counter with CBC-MAC) o, si el rendimiento es un problema, el modo GCM (Galois/Counter Mode), si están disponibles. **Ejemplo 2:** El siguiente código utiliza el cifrado AES con el modo GCM:

```
...
var cipher = new AesGcm(securityKeyArray)
...
```

Issue Summary

Figura 96. *Gráfico de vulnerabilidades de categoría Weak Encryption: Insecure Mode of Operation.*



Fuente: Elaboración propia.

Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
Weak Encryption: Insecure Mode of Operation	2	0	0	2
Total	**2**	**0**	**0**	**2**

Weak Encryption: Insecure Mode of Operation	**Critical**
Package: Framework.Tools	
Cryptographer.cs, line 63 (Weak Encryption: Insecure Mode of Operation)	**Critical**
Issue Details	
Kingdom: Security Features	
Scan Engine: SCA (Structural)	

Audit Details	
Analysis	Exploitable

Audit Comments

MAS: Tue Dec 07 2021 19:36:53 GMT+0100 (CET)

Hay que cambiar el modo de operación del cifrado

Sink Details

Sink: FunctionCall: set_Mode

Enclosing Method: Crypt()

File: Cryptographer.cs:63

Taint Flags:

```
No snippet available
```

Cryptographer.cs, line 140 (Weak Encryption: Insecure Mode of Operation) **Critical**

Issue Details

Kingdom: Security Features

Scan Engine: SCA (Structural)

Audit Details

Analysis Exploitable

Audit Comments

MAS: Tue Dec 07 2021 19:36:37 GMT+0100 (CET)

Hay que cambiar el modo de operación del descifrado

Sink Details

Sink: FunctionCall: set_Mode

Enclosing Method: Decrypt()

File: Cryptographer.cs:140

Taint Flags:

```
No snippet available
```

XML External Entity Injection (15 issues)

Abstract

El uso de analizadores de XML configurados para no evitar ni limitar la resolución de entidades externas puede exponer al analizador a un ataque de entidades externas de XML.

Explanation

Los ataques de entidades externas de XML se benefician de una función de XML para crear documentos de forma dinámica en el momento de procesamiento. Una entidad XML permite incluir datos de forma dinámica desde un recurso dado. Las entidades externas permiten a un documento XML incluir datos desde un URI externo. A menos que se configure de otra forma, las entidades

externas fuerzan al analizador de XML a acceder al recurso que especifica el URI, como por ejemplo un archivo del equipo local o de un sistema remoto. Este comportamiento expone la aplicación a ataques de entidades externas (XXE) de XML, los cuales se pueden utilizar para llevar a cabo una denegación de servicio del sistema local, obtener acceso no autorizado a archivos del equipo local, explorar equipos remotos y denegar el servicio de sistemas remotos. En el siguiente documento XML se muestra un ejemplo de un ataque XXE.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
 <!DOCTYPE foo [
  <!ELEMENT foo ANY >
  <!ENTITY xxe SYSTEM "file:///c:/winnt/win.ini" >]><foo>&xxe;</foo>
```

Este ejemplo podría revelar el contenido del archivo de sistema C:\winnt\win.ini, en caso de que el analizador de XML intente sustituir la entidad con el contenido del archivo.

Recommendation

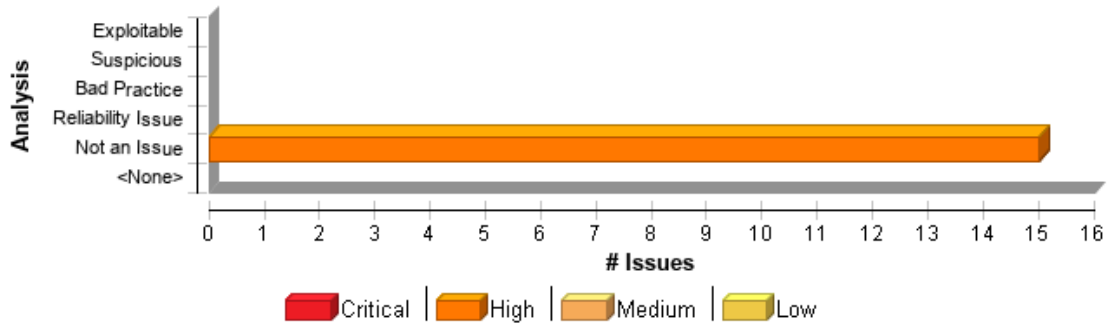
Un analizador XML debe configurarse de forma segura para que así no permita que las entidades externas formen parte de un documento XML entrante. La mejor forma de evitar ataques XXE es desactivar la resolución de la entidad XML. Para ello, se puede deshabilitar el parámetro DTD de la línea `DtdProcessing` estableciéndolo como `DtdProcessing.Prohibit`, o bien deshabilitar la resolución de la entidad XML estableciendo la propiedad `XmlReaderSettings.XmlResolver` como `null`:

```
XmlReaderSettings settings = new XmlReaderSettings();
settings.DtdProcessing = DtdProcessing.Prohibit;
settings.XmlResolver = null;
XmlReader reader = XmlReader.Create(stream, settings);
```

Si en su aplicación deben procesarse entidades externas, debería crear un XmlResolver personalizado con las siguientes funciones: - Establecer un tiempo de expiración de la solicitud para evitar ataques con retraso infinito - Limitar la cantidad de datos que recuperará - Restringir el XmlResolver para que no recupere recursos del host local En el artículo de Microsoft que aparece en la sección Referencias se describe el proceso completo de personalización de un XmlResolver.

Issue Summary

Figura 97. *Gráfico de vulnerabilidades de categoría XML External Entity Injection.*



Fuente: Elaboración propia.

Engine Breakdown

	SCA	WebInspect	SecurityScope	Total
XML External Entity Injection	15	0	0	15
Total	**15**	**0**	**0**	**15**

XML External Entity Injection	High

Package: Framework.Configuration

XMLConfigurationProvider.cs, line 177 (XML External Entity Injection) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Wed Dec 08 2021 13:26:44 GMT+0100 (CET)

El xml se crea vacío y luego ya se cargan los datos a este

Sink Details

Sink: Load(...) : XML reader instantiated allowing DTD processing

Enclosing Method: ReadDocument()

File: XMLConfigurationProvider.cs:177

Taint Flags:

```
No snippet available
```

ConfigDatabase.cs, line 223 (XML External Entity Injection) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Wed Dec 08 2021 13:22:58 GMT+0100 (CET)

El xml se crea vacío y luego ya se cargan los datos a este

Sink Details

Sink: LoadXml(...) : XML reader instantiated allowing DTD processing

Enclosing Method: LoadFromFile()

File: ConfigDatabase.cs:223

Taint Flags:

No snippet available

SystemCfgConfigurationProvider.cs, line 156 (XML External Entity Injection) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Data Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Wed Dec 08 2021 13:25:30 GMT+0100 (CET)

El xml se crea vacío y luego ya se cargan los datos a este

Source Details

Source: System.IO.FileStream.Read()

From: Framework.Configuration.SystemCfgConfigurationProvider.ReadEncrypt

edFile**File:** SystemCfgConfigurationProvider.cs:185

```
182  try
183  {
184  crypted_data = new byte[fs.Length];
185  fs.Read(crypted_data, 0, (int) fs.Length);
186  }
187  finally
188  {
```

Sink Details

Sink: System.Xml.XmlDocument.LoadXml()

Enclosing Method: LoadEncryptedFile()

File: SystemCfgConfigurationProvider.cs:156

Taint Flags: STREAM

```
No snippet available
```

Package: Framework.Configuration

ConfigDatabase.cs, line 170 (XML External Entity Injection) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Wed Dec 08 2021 13:22:36 GMT+0100 (CET)

El xml se crea vacío y luego ya se cargan los datos a este

Sink Details

Sink: LoadXml(...) : XML reader instantiated allowing DTD processing

Enclosing Method: LoadFromFile()

File: ConfigDatabase.cs:170

Taint Flags:

```
No snippet available
```

nConfig.cs, line 112 (XML External Entity Injection) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Wed Dec 08 2021 13:25:03 GMT+0100 (CET)

El xml se crea vacío y luego ya se cargan los datos a este

Sink Details

Sink: Load(...) : XML reader instantiated allowing DTD processing

Enclosing Method: XmlGetSet()

File: nConfig.cs:112

Taint Flags:

```
No snippet available
```

Package: Framework.Tools

Xml.cs, line 266 (XML External Entity Injection) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Wed Dec 08 2021 13:26:29 GMT+0100 (CET)

El xml se crea vacío y luego ya se cargan los datos a este

Sink Details

Sink: LoadXml(...) : XML reader instantiated allowing DTD processing

Enclosing Method: XMLTransform()

File: Xml.cs:266

Taint Flags:

```
No snippet available
```

ResourceFiles.cs, line 28 (XML External Entity Injection) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Wed Dec 08 2021 13:25:14 GMT+0100 (CET)

El xml se crea vacío y luego ya se cargan los datos a este

Sink Details

Sink: Load(...) : XML reader instantiated allowing DTD processing

Enclosing Method: GetCriteriaViewFilter()

File: ResourceFiles.cs:28

Taint Flags:

```
No snippet available
```

Package: Framework.Tools.WcfExtension

MessageContractAnnotation.cs, line 223 (XML External Entity Injection) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Wed Dec 08 2021 13:24:13 GMT+0100 (CET)

El xml se crea vacío y luego ya se cargan los datos a este

Sink Details

Sink: LoadXml(...) : XML reader instantiated allowing DTD processing

Enclosing Method: CreateXmlSchemaAnnotation()

File: MessageContractAnnotation.cs:223

Taint Flags:

```
No snippet available
```

XmlDocumentation.cs, line 116 (XML External Entity Injection) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Wed Dec 08 2021 13:26:55 GMT+0100 (CET)

El xml se crea vacío y luego ya se cargan los datos a este

Sink Details

Sink: LoadXml(...) : XML reader instantiated allowing DTD processing

Enclosing Method: Load()

File: XmlDocumentation.cs:116

Taint Flags:

```
No snippet available
```

XmlDocumentation.cs, line 152 (XML External Entity Injection) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Wed Dec 08 2021 13:26:59 GMT+0100 (CET)

El xml se crea vacío y luego ya se cargan los datos a este

Sink Details

Sink: LoadXml(...) : XML reader instantiated allowing DTD processing

Enclosing Method: Load()

File: XmlDocumentation.cs:152

Taint Flags:

```
No snippet available
```

Package: TestConfiguration

ConfigurationItem_providers_test.cs, line 298 (XML External Entity Injection) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Wed Dec 08 2021 13:23:30 GMT+0100 (CET)

Son pruebas de test, no es un error

MAS: Wed Dec 08 2021 13:23:49 GMT+0100 (CET)

El xml se crea vacío y luego ya se cargan los datos a este

Sink Details

Sink: LoadXml(...) : XML reader instantiated allowing DTD processing

Enclosing Method: ConfigurationXMLProvider_TestDefineTagsToIgnore()

File: ConfigurationItem_providers_test.cs:298

Taint Flags:

```
No snippet available
```

XmlConfigurationProvider_test.cs, line 19 (XML External Entity Injection) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Wed Dec 08 2021 13:26:49 GMT+0100 (CET)

El xml se crea vacío y luego ya se cargan los datos a este

Sink Details

Sink: LoadXml(...) : XML reader instantiated allowing DTD processing

Enclosing Method: GetXmlDoc()

File: XmlConfigurationProvider_test.cs:19

Taint Flags:

```
No snippet available
```

Package: Framework.Tools.GUI

WindowsManager.cs, line 964 (XML External Entity Injection) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Wed Dec 08 2021 13:25:58 GMT+0100 (CET)

El xml se crea vacío y luego ya se cargan los datos a este

Sink Details

Sink: Load(...) : XML reader instantiated allowing DTD processing

Enclosing Method: GetUserPreferencesFile()

File: WindowsManager.cs:964

Taint Flags:

No snippet available

WindowsManager.cs, line 1241 (XML External Entity Injection) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Wed Dec 08 2021 13:26:10 GMT+0100 (CET)

El xml se crea vacío y luego ya se cargan los datos a este

Sink Details

Sink: LoadXml(...) : XML reader instantiated allowing DTD processing

Enclosing Method: SaveFormPersistenceInfo()

File: WindowsManager.cs:1241

Taint Flags:

No snippet available

WindowsManager.cs, line 1222 (XML External Entity Injection) **High**

Issue Details

Kingdom: Input Validation and Representation

Scan Engine: SCA (Control Flow)

Audit Details

Analysis Not an Issue

Audit Comments

MAS: Wed Dec 08 2021 13:26:03 GMT+0100 (CET)

El xml se crea vacío y luego ya se cargan los datos a este

Sink Details

Sink: LoadXml(...) : XML reader instantiated allowing DTD processing

Enclosing Method: SaveWindowsPersistenceInfo()

File: WindowsManager.cs:1222

Taint Flags:

No snippet available

Anexo C. Informe de Nessus

En este anexo se incluye el informe obtenido con la herramienta Nessus para el escaneo de vulnerabilidades por el host.

A) 192.168.30.129

0	0	2	0	74
CRITICAL	HIGH	MEDIUM	LOW	INFO

Scan Information
Start time: Sat Feb 12 16:44:57 2022
End time: Sat Feb 12 18:22:24 2022

Host Information
IP: 192.168.30.129
OS: Microsoft Windows 10 Enterprise

Vulnerabilities

57608 - SMB Signing not required
Synopsis: Signing is not required on the remote SMB server.
Description: Signing is not required on the remote SMB server. An unauthenticated, remote attacker can exploit this to conduct man-in-the-middle attacks against the SMB server.
Solution: Enforce message signing in the host's configuration. On Windows, this is found in the policy setting 'Microsoft network server: Digitally sign communications (always)'. On Samba, the setting is called 'server signing'. See the 'see also' links for further details.
Risk Factor: Medium
Plugin Output: tcp/445/cifs

51192 - SSL Certificate Cannot Be Trusted
Synopsis: The SSL certificate for this service cannot be trusted.
Description: The server's X.509 certificate cannot be trusted. This situation can occur in three different ways, in which the chain of trust can be broken, as stated below:
- First, the top of the certificate chain sent by the server might not be descended from a known public certificate authority. This can occur either when the top of the chain is an unrecognized, self-signed certificate, or when intermediate certificates are missing that would connect the top of the certificate chain to a known public certificate authority.
- Second, the certificate chain may contain a certificate that is not valid at the time of the scan. This can occur either when the scan occurs before one of the certificate's 'notBefore' dates, or after one of the certificate's 'notAfter' dates.
- Third, the certificate chain may contain a signature that either didn't match the certificate's information or could not be verified. Bad signatures can be fixed by getting the certificate with the bad signature to be re-signed by its issuer. Signatures that could not be verified are the result of the certificate's issuer using a signing algorithm that Nessus either does not support or does not recognize.
If the remote host is a public host in production, any break in the chain makes it more difficult for users to verify the authenticity and identity of the web server. This could make it easier to carry out man-in-the-middle attacks against the remote host.
Solution: Purchase or generate a proper SSL certificate for this service.
Risk Factor: Medium
Plugin Output: tcp/8834/www

45590 - Common Platform Enumeration (CPE)

Synopsis: It was possible to enumerate CPE names that matched on the remote system.
Description: By using information obtained from a Nessus scan, this plugin reports CPE (Common Platform Enumeration) matches for various hardware and software products found on a host. Note that if an official CPE is not available for the product, this plugin computes the best possible CPE based on the information available from the scan.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/0

10736 - DCE Services Enumeration

Synopsis: A DCE/RPC service is running on the remote host.
Description: By sending a Lookup request to the portmapper (TCP 135 or epmapper PIPE) it was possible to enumerate the Distributed Computing Environment (DCE) services running on the remote port. Using this information it is possible to connect and bind to each service by sending an RPC request to the remote port/pipe.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/135/epmap

10736 - DCE Services Enumeration

Synopsis: A DCE/RPC service is running on the remote host.
Description: By sending a Lookup request to the portmapper (TCP 135 or epmapper PIPE) it was possible to enumerate the Distributed Computing Environment (DCE) services running on the remote port. Using this information it is possible to connect and bind to each service by sending an RPC request to the remote port/pipe.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/445/cifs

10736 - DCE Services Enumeration

Synopsis: A DCE/RPC service is running on the remote host.
Description: By sending a Lookup request to the portmapper (TCP 135 or epmapper PIPE) it was possible to enumerate the Distributed Computing Environment (DCE) services running on the remote port. Using this information it is possible to connect and bind to each service by sending an RPC request to the remote port/pipe.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/49664/dce-rpc

10736 - DCE Services Enumeration

Synopsis: A DCE/RPC service is running on the remote host.
Description: By sending a Lookup request to the portmapper (TCP 135 or epmapper PIPE) it was possible to enumerate the Distributed Computing Environment (DCE) services running on the remote port. Using this information it is possible to connect and bind to each service by sending an RPC request to the remote port/pipe.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/49665/dce-rcp

10736 - DCE Services Enumeration

Synopsis: A DCE/RPC service is running on the remote host.
Description: By sending a Lookup request to the portmapper (TCP 135 or epmapper PIPE) it was possible to enumerate the Distributed Computing Environment (DCE) services running on the remote port. Using this information it is possible to connect and bind to each service by sending an RPC request to the remote port/pipe.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/493666/dce-rpc

10736 - DCE Services Enumeration

Synopsis: A DCE/RPC service is running on the remote host.
Description: By sending a Lookup request to the portmapper (TCP 135 or epmapper PIPE) it was possible to enumerate the Distributed Computing Environment (DCE) services running on the remote port. Using this information it is possible to connect and bind to each service by sending an RPC request to the remote port/pipe.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/49667/dce-rpc

10736 - DCE Services Enumeration

Synopsis: A DCE/RPC service is running on the remote host.
Description: By sending a Lookup request to the portmapper (TCP 135 or epmapper PIPE) it was possible to enumerate the Distributed Computing Environment (DCE) services running on the remote port. Using this information it is possible to connect and bind to each service by sending an RPC request to the remote port/pipe.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/49668/dce-rpc

10736 - DCE Services Enumeration

Synopsis: A DCE/RPC service is running on the remote host.
Description: By sending a Lookup request to the portmapper (TCP 135 or epmapper PIPE) it was possible to enumerate the Distributed Computing Environment (DCE) services running on the remote port. Using this information it is possible to connect and bind to each service by sending an RPC request to the remote port/pipe.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/49669/dce-rpc

10736 - DCE Services Enumeration

Synopsis: A DCE/RPC service is running on the remote host.
Description: By sending a Lookup request to the portmapper (TCP 135 or epmapper PIPE) it was possible to enumerate the Distributed Computing Environment (DCE) services running on the remote port. Using this information it is possible to connect and bind to each service by sending an RPC request to the remote port/pipe.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/49672/dce-rpc

84239 - Debugging Log Report

Synopsis: This plugin gathers the logs written by other plugins and reports them.
Description: Logs generated by other plugins are reported by this plugin. Plugin debugging must be enabled in the policy in order for this plugin to run.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/0

54615 - Device Type

Synopsis: It is possible to guess the remote device type.
Description: Based on the remote operating system, it is possible to determine what the remote system type is (eg: a printer, router, general-purpose computer, etc).
Solution: n/a
Risk Factor: None
Plugin Output: tcp/0

117530 - Errors in nessusd.dump

Synopsis: This plugin parses information from the nessusd.dump log file and reports on errors.
Description: This plugin parses information from the nessusd.dump log file and reports on errors.
Solution: n/a

Risk Factor: None
Plugin Output: tcp/0

35716 - Ethernet Card Manufacturer Detection

Synopsis: The manufacturer can be identified from the Ethernet OUI.
Description: Each ethernet MAC address starts with a 24-bit Organizationally Unique Identifier (OUI). These OUIs are registered by IEEE.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/0

86420 - Ethernet MAC Addresses

Synopsis: This plugin gathers MAC addresses from various sources and consolidates them into a list.
Description: This plugin gathers MAC addresses discovered from both remote probing of the host (e.g. SNMP and Netbios) and from running local checks (e.g. ifconfig). It then consolidates the MAC addresses into a single, unique, and uniform list.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/0

43111 - HTTP Methods Allowed (per directory)

Synopsis: This plugin determines which HTTP methods are allowed on various CGI directories.
Description: By calling the OPTIONS method, it is possible to determine which HTTP methods are allowed on each directory. The following HTTP methods are considered insecure: PUT, DELETE, CONNECT, TRACE, HEAD. Many frameworks and languages treat 'HEAD' as a 'GET' request, albeit one without any body in the response. If a security constraint was set on 'GET' requests such that only 'authenticatedUsers' could access GET requests for a particular servlet or resource, it would be bypassed for the 'HEAD' version. This allowed unauthorized blind submission of any privileged GET request. As this list may be incomplete, the plugin also tests - if 'Thorough tests' are enabled or 'Enable web applications tests' is set to 'yes' in the scan policy - various known HTTP methods on each directory and considers them as unsupported if it receives a response code of 400, 403, 405, or 501. Note that the plugin output is only informational and does not necessarily indicate the presence of any security vulnerabilities.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/8834/www

10107 - HTTP Server Type and Version

Synopsis: A web server is running on the remote host.
Description: This plugin attempts to determine the type and the version of the remote web server.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/8834/www

24260 - HyperText Transfer Protocol (HTTP) Information

Synopsis: Some information about the remote HTTP configuration can be extracted.
Description: This test gives some information about the remote HTTP protocol - the version used, whether HTTP Keep-Alive and HTTP pipelining are enabled, etc... This test is informational only and does not denote any security problem.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/8834/www

10114 - ICMP Timestamp Request Remote Date Disclosure

Synopsis: It is possible to determine the exact time set on the remote host.
Description: The remote host answers to an ICMP timestamp request. This allows an attacker to know the date that is set on the targeted machine, which may assist an unauthenticated, remote attacker in defeating time-based authentication protocols. Timestamps returned from machines running Windows Vista / 7 / 2008 / 2008 R2 are deliberately incorrect, but usually within 1000 seconds of the actual system time.

Solution: Filter out the ICMP timestamp requests (13), and the outgoing ICMP timestamp replies (14).
Risk Factor: None
Plugin Output: icmp/0

14788 - IP Protocols Scan

Synopsis: This plugin detects the protocols understood by the remote IP stack.
Description: This plugin detects the protocols understood by the remote IP stack.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/0

53513 - Link-Local Multicast Name Resolution (LLMNR) Detection

Synopsis: The remote device supports LLMNR.
Description: The remote device answered to a Link-local Multicast Name Resolution (LLMNR) request. This protocol provides a name lookup service similar to NetBIOS or DNS. It is enabled by default on modern Windows versions.
Solution: Make sure that use of this software conforms to your organization's acceptable use and security policies.
Risk Factor: None
Plugin Output: udp/5355/llmnr

10785 - Microsoft Windows SMB NativeLanManager Remote System Information Disclosure

Synopsis: It was possible to obtain information about the remote operating system.
Description: Nessus was able to obtain the remote operating system name and version (Windows and/or Samba) by sending an authentication request to port 139 or 445. Note that this plugin requires SMB to be enabled on the host.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/445/cifs

11011 - Microsoft Windows SMB Service Detection

Synopsis: A file / print sharing service is listening on the remote host.
Description: The remote service understands the CIFS (Common Internet File System) or Server Message Block (SMB) protocol, used to provide shared access to files, printers, etc between nodes on a network.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/139/smb

11011 - Microsoft Windows SMB Service Detection

Synopsis: A file / print sharing service is listening on the remote host.
Description: The remote service understands the CIFS (Common Internet File System) or Server Message Block (SMB) protocol, used to provide shared access to files, printers, etc between nodes on a network.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/445/cifs

100871 - Microsoft Windows SMB Versions Supported (remote check)

Synopsis: It was possible to obtain information about the version of SMB running on the remote host.
Description: Nessus was able to obtain the version of SMB running on the remote host by sending an authentication request to port 139 or 445.Note that this plugin is a remote check and does not work on agents.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/445/cifs

106716 - Microsoft Windows SMB2 and SMB3 Dialects Supported (remote check)

Synopsis: It was possible to obtain information about the dialects of SMB2 and SMB3 available on the remote host.

Description: Nessus was able to obtain the set of SMB2 and SMB3 dialects running on the remote host by sending an authentication request to port 139 or 445.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/445/cifs

112154 - Nessus Launched Plugin List

Synopsis: This plugin displays information about the launched plugins.
Description: This plugin displays the list of launched plugins in a semicolon delimited list.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/0

19506 - Nessus Scan Information

Synopsis: This plugin displays information about the Nessus scan.
Description: This plugin displays, for each tested host, information about the scan itself:
- The version of the plugin set.
- The type of scanner (Nessus or Nessus Home).
- The version of the Nessus Engine.
- The port scanner(s) used.
- The port range scanned.
- The ping round trip time
- Whether credentialed or third-party patch management checks are possible.
- Whether the display of superseded patches is enabled
- The date of the scan.
- The duration of the scan.
- The number of hosts scanned in parallel.
- The number of checks done in parallel.

Solution: n/a
Risk Factor: None
Plugin Output: tcp/0

10147 - Nessus Server Detection

Synopsis: A Nessus daemon is listening on the remote port.
Description: A Nessus daemon is listening on the remote port.
Solution: Ensure that the remote Nessus installation has been authorized.
Risk Factor: None
Plugin Output: tcp/8834/www

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/135/epmap

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/139/smb

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.

Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/445/cifs

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/5040

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/7680

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/8834/www

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/49664/dce-rpc

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/49665/dce-rpc

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/49666/dce-rpc

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/49667/dce-rpc

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/49668/dce-rpc

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/49669/dce-rpc

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/49672/dce-rpc

34277 - Nessus UDP Scanner

Synopsis: It is possible to determine which UDP ports are open.
Description: This plugin runs a UDP port scan against the target. It is possible to determine which UDP ports are open by sending UDP packets on every port. If the port is open, the application will most often keep quiet. If the port is closed, the TCP/IP stack may send back an ICMP Host unreachable / bad port packet. However, this is assuming there are no intermediate devices between the scanner and the target. Firewalls often block ICMP, which will prevent responses that identify closed ports. The scanning primarily relies on the absence of a response to identify open ports and in complex environments with many intermediate devices, the detection can often be unreliable. UDP scanning takes a long time to complete. The scanner must limit the number of concurrent probes because ICMP is often rate limited. Also, since open ports do not respond, the scanner must wait for a timeout period to be reasonably sure that no response will be received. Given the typical environments being scanned today, the results of this plugin should be thoroughly vetted and be used as weak signals for further investigation. It is likely that a large number of assets will be detected if there are intermediate devices between the scanner and the targets. Consider using the netstat or SNMP port enumeration options instead if possible.
Solution: Protect your target with an IP filter or implement ICMP rate limitation.
Risk Factor: None
Plugin Output: udp/137/netbios-ns

34277 - Nessus UDP Scanner

Synopsis: It is possible to determine which UDP ports are open.
Description: This plugin runs a UDP port scan against the target. It is possible to determine which UDP ports are open by sending UDP packets on every port. If the port is open, the application will most often

keep quiet. If the port is closed, the TCP/IP stack may send back an ICMP Host unreachable / bad port packet. However, this is assuming there are no intermediate devices between the scanner and the target. Firewalls often block ICMP, which will prevent responses that identify closed ports. The scanning primarily relies on the absence of a response to identify open ports and in complex environments with many intermediate devices, the detection can often be unreliable. UDP scanning takes a long time to complete. The scanner must limit the number of concurrent probes because ICMP is often rate limited. Also, since open ports do not respond, the scanner must wait for a timeout period to be reasonably sure that no response will be received. Given the typical environments being scanned today, the results of this plugin should be thoroughly vetted and be used as weak signals for further investigation. It is likely that a large number of assets will be detected if there are intermediate devices between the scanner and the targets. Consider using the netstat or SNMP port enumeration options instead if possible.
Solution: Protect your target with an IP filter or implement ICMP rate limitation.
Risk Factor: None
Plugin Output: udp/138

34277 - Nessus UDP Scanner

Synopsis: It is possible to determine which UDP ports are open.
Description: This plugin runs a UDP port scan against the target. It is possible to determine which UDP ports are open by sending UDP packets on every port. If the port is open, the application will most often keep quiet.: If the port is closed, the TCP/IP stack may send back an ICMP Host unreachable / bad port packet. However, this is assuming there are no intermediate devices between the scanner and the target. Firewalls often block ICMP, which will prevent responses that identify closed ports. The scanning primarily relies on the absence of a response to identify open ports and in complex environments with many intermediate devices, the detection can often be unreliable. UDP scanning takes a long time to complete. The scanner must limit the number of concurrent probes because ICMP is often rate limited. Also, since open ports do not respond, the scanner must wait for a timeout period to be reasonably sure that no response will be received. Given the typical environments being scanned today, the results of this plugin should be thoroughly vetted and be used as weak signals for further investigation. It is likely that a large number of assets will be detected if there are intermediate devices between the scanner and the targets. Consider using the netstat or SNMP port enumeration options instead if possible.
Solution: Protect your target with an IP filter or implement ICMP rate limitation.
Risk Factor: None
Plugin Output: udp/500

34277 - Nessus UDP Scanner:

Synopsis: It is possible to determine which UDP ports are open.
Description: This plugin runs a UDP port scan against the target. It is possible to determine which UDP ports are open by sending UDP packets on every port. If the port is open, the application will most often keep quiet. If the port is closed, the TCP/IP stack may send back an ICMP Host unreachable / bad port packet. However, this is assuming there are no intermediate devices between the scanner and the target. Firewalls often block ICMP, which will prevent responses that identify closed ports. The scanning primarily relies on the absence of a response to identify open ports and in complex environments with many intermediate devices, the detection can often be unreliable. UDP scanning takes a long time to complete. The scanner must limit the number of concurrent probes because ICMP is often rate limited. Also, since open ports do not respond, the scanner must wait for a timeout period to be reasonably sure that no response will be received. Given the typical environments being scanned today, the results of this plugin should be thoroughly vetted and be used as weak signals for further investigation. It is likely that a large number of assets will be detected if there are intermediate devices between the scanner and the targets. Consider using the netstat or SNMP port enumeration options instead if possible.
Solution: Protect your target with an IP filter or implement ICMP rate limitation.
Risk Factor: None
Plugin Output: udp/1900

34277 - Nessus UDP Scanner

Synopsis: It is possible to determine which UDP ports are open.
Description: This plugin runs a UDP port scan against the target. It is possible to determine which UDP ports are open by sending UDP packets on every port. If the port is open, the application will most often

keep quiet. If the port is closed, the TCP/IP stack may send back an ICMP Host unreachable / bad port packet. However, this is assuming there are no intermediate devices between the scanner and the target. Firewalls often block ICMP, which will prevent responses that identify closed ports. The scanning primarily relies on the absence of a response to identify open ports and in complex environments with many intermediate devices, the detection can often be unreliable. UDP scanning takes a long time to complete. The scanner must limit the number of concurrent probes because ICMP is often rate limited. Also, since open ports do not respond, the scanner must wait for a timeout period to be reasonably sure that no response will be received. Given the typical environments being scanned today, the results of this plugin should be thoroughly vetted and be used as weak signals for further investigation. It is likely that a large number of assets will be detected if there are intermediate devices between the scanner and the targets. Consider using the netstat or SNMP port enumeration options instead if possible.
Solution: Protect your target with an IP filter or implement ICMP rate limitation.
Risk Factor: None
Plugin Output: udp/3702

34277 - Nessus UDP Scanner

Synopsis: It is possible to determine which UDP ports are open.
Description: This plugin runs a UDP port scan against the target. It is possible to determine which UDP ports are open by sending UDP packets on every port. If the port is open, the application will most often keep quiet. If the port is closed, the TCP/IP stack may send back an ICMP Host unreachable / bad port packet. However, this is assuming there are no intermediate devices between the scanner and the target. Firewalls often block ICMP, which will prevent responses that identify closed ports. The scanning primarily relies on the absence of a response to identify open ports and in complex environments with many intermediate devices, the detection can often be unreliable. UDP scanning takes a long time to complete. The scanner must limit the number of concurrent probes because ICMP is often rate limited. Also, since open ports do not respond, the scanner must wait for a timeout period to be reasonably sure that no response will be received. Given the typical environments being scanned today, the results of this plugin should be thoroughly vetted and be used as weak signals for further investigation. It is likely that a large number of assets will be detected if there are intermediate devices between the scanner and the targets. Consider using the netstat or SNMP port enumeration options instead if possible.
Solution: Protect your target with an IP filter or implement ICMP rate limitation.
Risk Factor: None
Plugin Output: udp/4500

34277 - Nessus UDP Scanner

Synopsis: It is possible to determine which UDP ports are open.
Description: This plugin runs a UDP port scan against the target. It is possible to determine which UDP ports are open by sending UDP packets on every port. If the port is open, the application will most often keep quiet. If the port is closed, the TCP/IP stack may send back an ICMP Host unreachable / bad port packet. However, this is assuming there are no intermediate devices between the scanner and the target. Firewalls often block ICMP, which will prevent responses that identify closed ports. The scanning primarily relies on the absence of a response to identify open ports and in complex environments with many intermediate devices, the detection can often be unreliable. UDP scanning takes a long time to complete. The scanner must limit the number of concurrent probes because ICMP is often rate limited. Also, since open ports do not respond, the scanner must wait for a timeout period to be reasonably sure that no response will be received. Given the typical environments being scanned today, the results of this plugin should be thoroughly vetted and be used as weak signals for further investigation. It is likely that a large number of assets will be detected if there are intermediate devices between the scanner and the targets. Consider using the netstat or SNMP port enumeration options instead if possible.
Solution: Protect your target with an IP filter or implement ICMP rate limitation.
Risk Factor: None
Plugin Output: udp/5050

34277 - Nessus UDP Scanner

Synopsis: It is possible to determine which UDP ports are open.
Description: This plugin runs a UDP port scan against the target. It is possible to determine which UDP ports are open by sending UDP packets on every port. If the port is open, the application will most often

keep quiet. If the port is closed, the TCP/IP stack may send back an ICMP Host unreachable / bad port packet. However, this is assuming there are no intermediate devices between the scanner and the target. Firewalls often block ICMP, which will prevent responses that identify closed ports. The scanning primarily relies on the absence of a response to identify open ports and in complex environments with many intermediate devices, the detection can often be unreliable. UDP scanning takes a long time to complete. The scanner must limit the number of concurrent probes because ICMP is often rate limited. Also, since open ports do not respond, the scanner must wait for a timeout period to be reasonably sure that no response will be received.Given the typical environments being scanned today, the results of this plugin should be thoroughly vetted and be used as weak signals for further investigation. It is likely that a large number of assets will be detected if there are intermediate devices between the scanner and the targets. Consider using the netstat or SNMP port enumeration options instead if possible.
Solution: Protect your target with an IP filter or implement ICMP rate limitation.
Risk Factor: None
Plugin Output: udp/5353

34277 - Nessus UDP Scanner

Synopsis: It is possible to determine which UDP ports are open.
Description: This plugin runs a UDP port scan against the target. It is possible to determine which UDP ports are open by sending UDP packets on every port. If the port is open, the application will most often keep quiet. If the port is closed, the TCP/IP stack may send back an ICMP Host unreachable / bad port packet. However, this is assuming there are no intermediate devices between the scanner and the target. Firewalls often block ICMP, which will prevent responses that identify closed ports. The scanning primarily relies on the absence of a response to identify open ports and in complex environments with many intermediate devices, the detection can often be unreliable. UDP scanning takes a long time to complete. The scanner must limit the number of concurrent probes because ICMP is often rate limited. Also, since open ports do not respond, the scanner must wait for a timeout period to be reasonably sure that no response will be received. Given the typical environments being scanned today, the results of this plugin should be thoroughly vetted and be used as weak signals for further investigation. It is likely that a large number of assets will be detected if there are intermediate devices between the scanner and the targets. Consider using the netstat or SNMP port enumeration options instead if possible.
Solution: Protect your target with an IP filter or implement ICMP rate limitation.
Risk Factor: None
Plugin Output: udp/5355/llmnr

34277 - Nessus UDP Scanner

Synopsis: It is possible to determine which UDP ports are open.
Description: This plugin runs a UDP port scan against the target. It is possible to determine which UDP ports are open by sending UDP packets on every port. If the port is open, the application will most often keep quiet. If the port is closed, the TCP/IP stack may send back an ICMP Host unreachable / bad port packet. However, this is assuming there are no intermediate devices between the scanner and the target. Firewalls often block ICMP, which will prevent responses that identify closed ports. The scanning primarily relies on the absence of a response to identify open ports and in complex environments with many intermediate devices, the detection can often be unreliable. UDP scanning takes a long time to complete. The scanner must limit the number of concurrent probes because ICMP is often rate limited. Also, since open ports do not respond, the scanner must wait for a timeout period to be reasonably sure that no response will be received. Given the typical environments being scanned today, the results of this plugin should be thoroughly vetted and be used as weak signals for further investigation. It is likely that a large number of assets will be detected if there are intermediate devices between the scanner and the targets. Consider using the netstat or SNMP port enumeration options instead if possible.
Solution: Protect your target with an IP filter or implement ICMP rate limitation.
Risk Factor: None
Plugin Output: udp/50974

34277 - Nessus UDP Scanner

Synopsis: It is possible to determine which UDP ports are open.
Description: This plugin runs a UDP port scan against the target. It is possible to determine which UDP ports are open by sending UDP packets on every port. If the port is open, the application will most often

keep quiet. If the port is closed, the TCP/IP stack may send back an ICMP Host unreachable / bad port packet. However, this is assuming there are no intermediate devices between the scanner and the target. Firewalls often block ICMP, which will prevent responses that identify closed ports. The scanning primarily relies on the absence of a response to identify open ports and in complex environments with many intermediate devices, the detection can often be unreliable. UDP scanning takes a long time to complete. The scanner must limit the number of concurrent probes because ICMP is often rate limited. Also, since open ports do not respond, the scanner must wait for a timeout period to be reasonably sure that no response will be received. Given the typical environments being scanned today, the results of this plugin should be thoroughly vetted and be used as weak signals for further investigation. It is likely that a large number of assets will be detected if there are intermediate devices between the scanner and the targets. Consider using the netstat or SNMP port enumeration options instead if possible.
Solution: Protect your target with an IP filter or implement ICMP rate limitation.
Risk Factor: None
Plugin Output: udp/64127

11936 - OS Identification

Synopsis: It is possible to guess the remote operating system.
Description: Using a combination of remote probes (e.g., TCP/IP, SMB, HTTP, NTP, SNMP, etc.), it is possible to guess the name of the remote operating system in use. It is also possible sometimes to guess the version of the operating system.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/0

21745 - OS Security Patch Assessment Failed

Synopsis: Errors prevented OS Security Patch Assessment.
Description: OS Security Patch Assessment is not available for this host because either the credentials supplied in the scan policy did not allow Nessus to log into it or some other problem occurred.
Solution: Fix the problem(s) so that OS Security Patch Assessment is possible.
Risk Factor: None
Plugin Output: tcp/0

66334 - Patch Report

Synopsis: The remote host is missing several patches.
Description: The remote host is missing one or more security patches. This plugin lists the newest version of each patch to install to make sure the remote host is up-to-date.
Solution: Install the patches listed below.
Risk Factor: None
Plugin Output: tcp/0

10180 - Ping the remote host

Synopsis: It was possible to identify the status of the remote host (alive or dead).
Description: Nessus was able to determine if the remote host is alive using one or more of the following ping types:
- An ARP ping, provided the host is on the local subnet and Nessus is running over Ethernet.
- An ICMP ping.
- A TCP ping, in which the plugin sends to the remote host a packet with the flag SYN, and the host will reply with a RST or a SYN/ACK.
- A UDP ping (e.g., DNS, RPC, and NTP).
Solution: n/a
Risk Factor: None
Plugin Output: tcp/0

56984 - SSL / TLS Versions Supported

Synopsis: The remote service encrypts communications.
Description: This plugin detects which SSL and TLS versions are supported by the remote service for encrypting communications.
Solution: n/a
Risk Factor: None

Plugin Output: tcp/8834/www

10863 - SSL Certificate Information

Synopsis: This plugin displays the SSL certificate.
Description: This plugin connects to every SSL-related port and attempts to extract and dump the X.509 certificate.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/8834/www

21643 - SSL Cipher Suites Supported

Synopsis: The remote service encrypts communications using SSL.
Description: This plugin detects which SSL ciphers are supported by the remote service for encrypting communications.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/8834/www

57041 - SSL Perfect Forward Secrecy Cipher Suites Supported

Synopsis: The remote service supports the use of SSL Perfect Forward Secrecy ciphers, which maintain confidentiality even if the key is stolen.
Description: The remote host supports the use of SSL ciphers that offer Perfect Forward Secrecy (PFS) encryption. These cipher suites ensure that recorded SSL traffic cannot be broken at a future date if the server's private key is compromised.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/8834/www

22964 - Service Detection

Synopsis: The remote service could be identified.
Description: Nessus was able to identify the remote service by its banner or by looking at the error message it sends when it receives an HTTP request.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/8834/www y tcp/8834/www

42822 - Strict Transport Security (STS) Detection

Synopsis: The remote web server implements Strict Transport Security.
Description: The remote web server implements Strict Transport Security (STS). The goal of STS is to make sure that a user does not accidentally downgrade the security of his or her browser. All unencrypted HTTP connections are redirected to HTTPS. The browser is expected to treat all cookies as 'secure' and to close the connection in the event of potentially insecure situations.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/8834/www

136318 - TLS Version 1.2 Protocol Detection

Synopsis: The remote service encrypts traffic using a version of TLS.
Description: The remote service accepts connections encrypted using TLS 1.2.
Solution: N/A
Risk Factor: None
Plugin Output: tcp/8834/www

138330 - TLS Version 1.3 Protocol Detection

Synopsis: The remote service encrypts traffic using a version of TLS.
Description: The remote service accepts connections encrypted using TLS 1.3.
Solution: N/A
Risk Factor: None
Plugin Output: tcp/8834/www

104410 - Target Credential Status by Authentication Protocol - Failure for Provided Credentials

Synopsis: Nessus was unable to log into the detected authentication protocol, using the provided credentials, in order to perform credentialed checks.
Description: Nessus failed to successfully authenticate directly to the remote target on an available authentication protocol. Nessus was able to connect to the remote port and identify that the service running on the port supports an authentication protocol, but Nessus failed to authenticate to the remote service using the provided credentials. There may have been a failure in protocol negotiation or communication that prevented authentication from being attempted or all of the provided credentials for the authentication protocol may have been invalid. A protocol failure may indicate a compatibility issue with the protocol configuration. A protocol failure due to an environmental issue such as resource or congestion issues may also prevent valid credentials from being identified. See plugin output for error details.Please note the following:
- This plugin reports per protocol, so it is possible for valid credentials to be provided for one protocol and not another. For example, authentication may succeed via SSH but fail via SMB, while no credentials were provided for an available SNMP service.
- Providing valid credentials for all available authentication protocols may improve scan coverage, but the value of successful authentication for a given protocol may vary from target to target depending upon what data (if any) is gathered from the target via that protocol. For example, successful authentication via SSH is more valuable for Linux targets than for Windows targets, and likewise successful authentication via SMB is more valuable for Windows targets than for Linux targets.
Solution: Address the reported problem(s) so that credentialed checks can be executed.
Risk Factor: None
Plugin Output: tcp/445/cifs

10287 - Traceroute Information

Synopsis: It was possible to obtain traceroute information.
Description: Makes a traceroute to the remote host.
Solution: n/a
Risk Factor: None
Plugin Output: udp/0

20094 - VMware Virtual Machine Detection

Synopsis: The remote host is a VMware virtual machine.
Description: According to the MAC address of its network adapter, the remote host is a VMware virtual machine.
Solution: Since it is physically accessible through the network, ensure that its configuration matches your organization's security policy.
Risk Factor: None
Plugin Output: tcp/0

135860 - WMI Not Available

Synopsis: WMI queries could not be made against the remote host.
Description: WMI (Windows Management Instrumentation) is not available on the remote host over DCOM. WMI queries are used to gather information about the remote host, such as its current state, network interface configuration, etc. Without this information Nessus may not be able to identify installed software or security vunerabilities that exist on the remote host.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/445/cifs

91815 - Web Application Sitemap

Synopsis: The remote web server hosts linkable content that can be crawled by Nessus.
Description: The remote web server contains linkable content that can be used to gather information about a target.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/8834/www

11032 - Web Server Directory Enumeration

Synopsis: It is possible to enumerate directories on the web server.

Description: This plugin attempts to determine the presence of various common directories on the remote web server. By sending a request for a directory, the web server response code indicates if it is a valid directory or not.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/8834/www

10150 - Windows NetBIOS / SMB Remote Host Information Disclosure

Synopsis: It was possible to obtain the network name of the remote host.
Description: The remote host is listening on UDP port 137 or TCP port 445, and replies to NetBIOS nbtscan or SMB requests. Note that this plugin gathers information to be used in other plugins, but does not itself generate a report.
Solution: n/a
Risk Factor: None
Plugin Output: udp/137/netbios-ns

B) 192.168.30.135

Scan Information
Start time: Sat Feb 12 16:45:07 2022
End time: Sat Feb 12 18:21:13 2022

Host Information
IP: 192.168.30.135
OS: Microsoft Windows

Vulnerabilities

57608 - SMB Signing not required

Synopsis: Signing is not required on the remote SMB server.
Description: Signing is not required on the remote SMB server. An unauthenticated, remote attacker can exploit this to conduct man-in-the-middle attacks against the SMB server.
Solution: Enforce message signing in the host's configuration. On Windows, this is found in the policy setting 'Microsoft network server: Digitally sign communications (always)'. On Samba, the setting is called 'server signing'. See the 'see also' links for further details.
Risk Factor: Medium
Plugin Output: tcp/445/cifs

51192 - SSL Certificate Cannot Be Trusted

Synopsis: The SSL certificate for this service cannot be trusted.
Description: The server's X.509 certificate cannot be trusted. This situation can occur in three different ways, in which the chain of trust can be broken, as stated below:
- First, the top of the certificate chain sent by the server might not be descended from a known public certificate authority. This can occur either when the top of the chain is an unrecognized, self-signed certificate, or when intermediate certificates are missing that would connect the top of the certificate chain to a known public certificate authority.
- Second, the certificate chain may contain a certificate that is not valid at the time of the scan. This can occur either when the scan occurs before one of the certificate's 'notBefore' dates, or after one of the certificate's 'notAfter' dates.
- Third, the certificate chain may contain a signature that either didn't match the certificate's information

or could not be verified. Bad signatures can be fixed by getting the certificate with the bad signature to be re-signed by its issuer. Signatures that could not be verified are the result of the certificate's issuer using a signing algorithm that Nessus either does not support or does not recognize. If the remote host is a public host in production, any break in the chain makes it more difficult for users to verify the authenticity and identity of the web server. This could make it easier to carry out man-in-the-middle attacks against the remote host.
Solution: Purchase or generate a proper SSL certificate for this service.
Risk Factor: Medium
Plugin Output: tcp/8834/www

45590 - Common Platform Enumeration (CPE)

Synopsis: It was possible to enumerate CPE names that matched on the remote system.
Description: By using information obtained from a Nessus scan, this plugin reports CPE (Common Platform Enumeration) matches for various hardware and software products found on a host. Note that if an official CPE is not available for the product, this plugin computes the best possible CPE based on the information available from the scan.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/0

10736 - DCE Services Enumeration

Synopsis: A DCE/RPC service is running on the remote host.
Description: By sending a Lookup request to the portmapper (TCP 135 or epmapper PIPE) it was possible to enumerate the Distributed Computing Environment (DCE) services running on the remote port. Using this information it is possible to connect and bind to each service by sending an RPC request to the remote port/pipe.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/135/epmap

10736 - DCE Services Enumeration

Synopsis: A DCE/RPC service is running on the remote host.
Description: By sending a Lookup request to the portmapper (TCP 135 or epmapper PIPE) it was possible to enumerate the Distributed Computing Environment (DCE) services running on the remote port. Using this information it is possible to connect and bind to each service by sending an RPC request to the remote port/pipe.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/445/cifs

10736 - DCE Services Enumeration

Synopsis: A DCE/RPC service is running on the remote host.
Description: By sending a Lookup request to the portmapper (TCP 135 or epmapper PIPE) it was possible to enumerate the Distributed Computing Environment (DCE) services running on the remote port. Using this information it is possible to connect and bind to each service by sending an RPC request to the remote port/pipe.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/49664/dce-rpc

10736 - DCE Services Enumeration

Synopsis: A DCE/RPC service is running on the remote host.
Description: By sending a Lookup request to the portmapper (TCP 135 or epmapper PIPE) it was possible to enumerate the Distributed Computing Environment (DCE) services running on the remote port. Using this information it is possible to connect and bind to each service by sending an RPC request to the remote port/pipe.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/49665/dce-rpc

10736 - DCE Services Enumeration
Synopsis: A DCE/RPC service is running on the remote host.
Description: By sending a Lookup request to the portmapper (TCP 135 or epmapper PIPE) it was possible to enumerate the Distributed Computing Environment (DCE) services running on the remote port. Using this information it is possible to connect and bind to each service by sending an RPC request to the remote port/pipe.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/49666/dce-rpc

10736 - DCE Services Enumeration
Synopsis: A DCE/RPC service is running on the remote host.
Description: By sending a Lookup request to the portmapper (TCP 135 or epmapper PIPE) it was possible to enumerate the Distributed Computing Environment (DCE) services running on the remote port. Using this information it is possible to connect and bind to each service by sending an RPC request to the remote port/pipe.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/49667/dce-rpc

10736 - DCE Services Enumeration
Synopsis: A DCE/RPC service is running on the remote host.
Description: By sending a Lookup request to the portmapper (TCP 135 or epmapper PIPE) it was possible to enumerate the Distributed Computing Environment (DCE) services running on the remote port. Using this information it is possible to connect and bind to each service by sending an RPC request to the remote port/pipe.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/49668/dce-rpc

10736 - DCE Services Enumeration
Synopsis: A DCE/RPC service is running on the remote host.
Description: By sending a Lookup request to the portmapper (TCP 135 or epmapper PIPE) it was possible to enumerate the Distributed Computing Environment (DCE) services running on the remote port. Using this information it is possible to connect and bind to each service by sending an RPC request to the remote port/pipe.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/49669/dce-rpc

10736 - DCE Services Enumeration
Synopsis: A DCE/RPC service is running on the remote host.
Description: By sending a Lookup request to the portmapper (TCP 135 or epmapper PIPE) it was possible to enumerate the Distributed Computing Environment (DCE) services running on the remote port. Using this information it is possible to connect and bind to each service by sending an RPC request to the remote port/pipe.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/49672/dce-rpc

84239 - Debugging Log Report
Synopsis: This plugin gathers the logs written by other plugins and reports them.
Description: Logs generated by other plugins are reported by this plugin. Plugin debugging must be enabled in the policy in order for this plugin to run.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/0

54615 - Device Type
Synopsis: It is possible to guess the remote device type.

Description: Based on the remote operating system, it is possible to determine what the remote system type is (eg: a printer, router, general-purpose computer, etc).
Solution: n/a
Risk Factor: None
Plugin Output: tcp/0

117530 - Errors in nessusd.dump

Synopsis: This plugin parses information from the nessusd.dump log file and reports on errors.
Description: This plugin parses information from the nessusd.dump log file and reports on errors.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/0

35716 - Ethernet Card Manufacturer Detection

Synopsis: The manufacturer can be identified from the Ethernet OUI.
Description: Each ethernet MAC address starts with a 24-bit Organizationally Unique Identifier (OUI). These OUIs are registered by IEEE.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/0

86420 - Ethernet MAC Addresses

Synopsis: This plugin gathers MAC addresses from various sources and consolidates them into a list.
Description: This plugin gathers MAC addresses discovered from both remote probing of the host (e.g. SNMP and Netbios) and from running local checks (e.g. ifconfig). It then consolidates the MAC addresses into a single, unique, and uniform list.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/0

43111 - HTTP Methods Allowed (per directory)

Synopsis: This plugin determines which HTTP methods are allowed on various CGI directories.
Description: By calling the OPTIONS method, it is possible to determine which HTTP methods are allowed on each directory. The following HTTP methods are considered insecure: PUT, DELETE, CONNECT, TRACE, HEAD. Many frameworks and languages treat 'HEAD' as a 'GET' request, albeit one without any body in the response. If a security constraint was set on 'GET' requests such that only 'authenticatedUsers' could access GET requests for a particular servlet or resource, it would be bypassed for the 'HEAD' version. This allowed unauthorized blind submission of any privileged GET request. As this list may be incomplete, the plugin also tests - if 'Thorough tests' are enabled or 'Enable web applications tests' is set to 'yes' in the scan policy - various known HTTP methods on each directory and considers them as unsupported if it receives a response code of 400, 403, 405, or 501. Note that the plugin output is only informational and does not necessarily indicate the presence of any security vulnerabilities.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/80/www

43111 - HTTP Methods Allowed (per directory)

Synopsis: This plugin determines which HTTP methods are allowed on various CGI directories.
Description: By calling the OPTIONS method, it is possible to determine which HTTP methods are allowed on each directory. The following HTTP methods are considered insecure: PUT, DELETE, CONNECT, TRACE, HEAD. Many frameworks and languages treat 'HEAD' as a 'GET' request, albeit one without any body in the response. If a security constraint was set on 'GET' requests such that only 'authenticatedUsers' could access GET requests for a particular servlet or resource, it would be bypassed for the 'HEAD' version. This allowed unauthorized blind submission of any privileged GET request. As this list may be incomplete, the plugin also tests - if 'Thorough tests' are enabled or 'Enable web applications tests' is set to 'yes' in the scan policy - various known HTTP methods on each directory and considers them as unsupported if it receives a response code of 400, 403, 405, or 501. Note that the

plugin output is only informational and does not necessarily indicate the presence of any security vulnerabilities.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/8834/www

10107 - HTTP Server Type and Version

Synopsis: A web server is running on the remote host.
Description: This plugin attempts to determine the type and the version of the remote web server.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/80/www

10107 - HTTP Server Type and Version

Synopsis: A web server is running on the remote host.
Description: This plugin attempts to determine the type and the version of the remote web server.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/8834/www

24260 - HyperText Transfer Protocol (HTTP) Information

Synopsis: Some information about the remote HTTP configuration can be extracted.
Description: This test gives some information about the remote HTTP protocol - the version used, whether HTTP Keep-Alive and HTTP pipelining are enabled, etc... This test is informational only and does not denote any security problem.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/80/www

24260 - HyperText Transfer Protocol (HTTP) Information

Synopsis: Some information about the remote HTTP configuration can be extracted.
Description: This test gives some information about the remote HTTP protocol - the version used, whether HTTP Keep-Alive and HTTP pipelining are enabled, etc... This test is informational only and does not denote any security problem.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/8834/www

10114 - ICMP Timestamp Request Remote Date Disclosure

Synopsis: It is possible to determine the exact time set on the remote host.
Description: The remote host answers to an ICMP timestamp request. This allows an attacker to know the date that is set on the targeted machine, which may assist an unauthenticated, remote attacker in defeating time-based authentication protocols. Timestamps returned from machines running Windows Vista / 7 / 2008 / 2008 R2 are deliberately incorrect, but usually within 1000 seconds of the actual system time.
Solution: Filter out the ICMP timestamp requests (13), and the outgoing ICMP timestamp replies (14).
Risk Factor: None
Plugin Output: icmp/0

14788 - IP Protocols Scan

Synopsis: This plugin detects the protocols understood by the remote IP stack.
Description: This plugin detects the protocols understood by the remote IP stack.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/0

53513 - Link-Local Multicast Name Resolution (LLMNR) Detection

Synopsis: The remote device supports LLMNR.
Description: The remote device answered to a Link-local Multicast Name Resolution (LLMNR) request. This protocol provides a name lookup service similar to NetBIOS or DNS. It is enabled by default on modern Windows versions.

Solution: Make sure that use of this software conforms to your organization's acceptable use and security policies.
Risk Factor: None
Plugin Output: udp/5355/llmnr

42410 - Microsoft Windows NTLMSSP Authentication Request Remote Network Name Disclosure

Synopsis: It is possible to obtain the network name of the remote host.
Description: The remote host listens on tcp port 445 and replies to SMB requests. By sending an NTLMSSP authentication request it is possible to obtain the name of the remote system and the name of its domain.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/445/cifs

10785 - Microsoft Windows SMB NativeLanManager Remote System Information Disclosure

Synopsis: It was possible to obtain information about the remote operating system.
Description: Nessus was able to obtain the remote operating system name and version (Windows and/or Samba) by sending an authentication request to port 139 or 445. Note that this plugin requires SMB to be enabled on the host.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/445/cifs

11011 - Microsoft Windows SMB Service Detection

Synopsis: A file / print sharing service is listening on the remote host.
Description: The remote service understands the CIFS (Common Internet File System) or Server Message Block (SMB) protocol, used to provide shared access to files, printers, etc between nodes on a network.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/139/smb

11011 - Microsoft Windows SMB Service Detection

Synopsis: A file / print sharing service is listening on the remote host.
Description: The remote service understands the CIFS (Common Internet File System) or Server Message Block (SMB) protocol, used to provide shared access to files, printers, etc between nodes on a network.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/445/cifs

100871 - Microsoft Windows SMB Versions Supported (remote check)

Synopsis: It was possible to obtain information about the version of SMB running on the remote host.
Description: Nessus was able to obtain the version of SMB running on the remote host by sending an authentication request to port 139 or 445.Note that this plugin is a remote check and does not work on agents.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/445/cifs

106716 - Microsoft Windows SMB2 and SMB3 Dialects Supported (remote check)

Synopsis: It was possible to obtain information about the dialects of SMB2 and SMB3 available on the remote host.
Description: Nessus was able to obtain the set of SMB2 and SMB3 dialects running on the remote host by sending an authentication request to port 139 or 445.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/445/cifs

112154 - Nessus Launched Plugin List

Synopsis: This plugin displays information about the launched plugins.
Description: This plugin displays the list of launched plugins in a semicolon delimited list.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/0

19506 - Nessus Scan Information

Synopsis: This plugin displays information about the Nessus scan.
Description: This plugin displays, for each tested host, information about the scan itself:
- The version of the plugin set.
- The type of scanner (Nessus or Nessus Home).
- The version of the Nessus Engine.
- The port scanner(s) used.
- The port range scanned.
- The ping round trip time
- Whether credentialed or third-party patch management checks are possible.
- Whether the display of superseded patches is enabled
- The date of the scan.
- The duration of the scan.
- The number of hosts scanned in parallel.
- The number of checks done in parallel.

Solution: n/a
Risk Factor: None
Plugin Output: tcp/0

10147 - Nessus Server Detection

Synopsis: A Nessus daemon is listening on the remote port.
Description: A Nessus daemon is listening on the remote port.
Solution: Ensure that the remote Nessus installation has been authorized.
Risk Factor: None
Plugin Output: tcp/8834/www

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/80/www

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/135/epmap

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/139/smb

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/445/cifs

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/5040

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/7680

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/8834/www

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/33003

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/33005

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.

Risk Factor: None
Plugin Output: tcp/33009

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/33013

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/33221

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/33338

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/33339

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/43015

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/43017

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.

Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/43019

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/49664/dce-rpc

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/49665/dce-rpc

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/49666/dce-rpc

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/49667/dce-rpc

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/49668/dce-rpc

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/49669/dce-rpc

10335 - Nessus TCP scanner

Synopsis: It is possible to determine which TCP ports are open.
Description: This plugin is a classical TCP port scanner. It shall be reasonably quick even against a firewalled target. Once a TCP connection is open, it grabs any available banner for the service identification plugins. Note that TCP scanners are more intrusive than SYN (half open) scanners.
Solution: Protect your target with an IP filter.
Risk Factor: None
Plugin Output: tcp/49672/dce-rpc

34277 - Nessus UDP Scanner

Synopsis: It is possible to determine which UDP ports are open.
Description: This plugin runs a UDP port scan against the target. It is possible to determine which UDP ports are open by sending UDP packets on every port. If the port is open, the application will most often keep quiet. If the port is closed, the TCP/IP stack may send back an ICMP Host unreachable / bad port packet. However, this is assuming there are no intermediate devices between the scanner and the target. Firewalls often block ICMP, which will prevent responses that identify closed ports. The scanning primarily relies on the absence of a response to identify open ports and in complex environments with many intermediate devices, the detection can often be unreliable. UDP scanning takes a long time to complete. The scanner must limit the number of concurrent probes because ICMP is often rate limited. Also, since open ports do not respond, the scanner must wait for a timeout period to be reasonably sure that no response will be received. Given the typical environments being scanned today, the results of this plugin should be thoroughly vetted and be used as weak signals for further investigation. It is likely that a large number of assets will be detected if there are intermediate devices between the scanner and the targets. Consider using the netstat or SNMP port enumeration options instead if possible.
Solution: Protect your target with an IP filter or implement ICMP rate limitation.
Risk Factor: None
Plugin Output: udp/137

34277 - Nessus UDP Scanner

Synopsis: It is possible to determine which UDP ports are open.
Description: This plugin runs a UDP port scan against the target. It is possible to determine which UDP ports are open by sending UDP packets on every port. If the port is open, the application will most often keep quiet. If the port is closed, the TCP/IP stack may send back an ICMP Host unreachable / bad port packet. However, this is assuming there are no intermediate devices between the scanner and the target. Firewalls often block ICMP, which will prevent responses that identify closed ports. The scanning primarily relies on the absence of a response to identify open ports and in complex environments with many intermediate devices, the detection can often be unreliable. UDP scanning takes a long time to complete. The scanner must limit the number of concurrent probes because ICMP is often rate limited. Also, since open ports do not respond, the scanner must wait for a timeout period to be reasonably sure that no response will be received. Given the typical environments being scanned today, the results of this plugin should be thoroughly vetted and be used as weak signals for further investigation. It is likely that a large number of assets will be detected if there are intermediate devices between the scanner and the targets. Consider using the netstat or SNMP port enumeration options instead if possible.
Solution: Protect your target with an IP filter or implement ICMP rate limitation.
Risk Factor: None
Plugin Output: udp/138

34277 - Nessus UDP Scanner

Synopsis: It is possible to determine which UDP ports are open.
Description: This plugin runs a UDP port scan against the target. It is possible to determine which UDP ports are open by sending UDP packets on every port. If the port is open, the application will most often keep quiet. If the port is closed, the TCP/IP stack may send back an ICMP Host unreachable / bad port packet. However, this is assuming there are no intermediate devices between the scanner and the target. Firewalls often block ICMP, which will prevent responses that identify closed ports. The scanning primarily relies on the absence of a response to identify open ports and in complex environments with many intermediate devices, the detection can often be unreliable. UDP scanning takes a long time to complete. The scanner must limit the number of concurrent probes because ICMP is often rate limited.

Also, since open ports do not respond, the scanner must wait for a timeout period to be reasonably sure that no response will be received. Given the typical environments being scanned today, the results of this plugin should be thoroughly vetted and be used as weak signals for further investigation. It is likely that a large number of assets will be detected if there are intermediate devices between the scanner and the targets. Consider using the netstat or SNMP port enumeration options instead if possible.
Solution: Protect your target with an IP filter or implement ICMP rate limitation.
Risk Factor: None
Plugin Output: udp/500

34277 - Nessus UDP Scanner

Synopsis: It is possible to determine which UDP ports are open.
Description: This plugin runs a UDP port scan against the target. It is possible to determine which UDP ports are open by sending UDP packets on every port. If the port is open, the application will most often keep quiet. If the port is closed, the TCP/IP stack may send back an ICMP Host unreachable / bad port packet. However, this is assuming there are no intermediate devices between the scanner and the target. Firewalls often block ICMP, which will prevent responses that identify closed ports. The scanning primarily relies on the absence of a response to identify open ports and in complex environments with many intermediate devices, the detection can often be unreliable. UDP scanning takes a long time to complete. The scanner must limit the number of concurrent probes because ICMP is often rate limited. Also, since open ports do not respond, the scanner must wait for a timeout period to be reasonably sure that no response will be received. Given the typical environments being scanned today, the results of this plugin should be thoroughly vetted and be used as weak signals for further investigation. It is likely that a large number of assets will be detected if there are intermediate devices between the scanner and the targets. Consider using the netstat or SNMP port enumeration options instead if possible.
Solution: Protect your target with an IP filter or implement ICMP rate limitation.
Risk Factor: None
Plugin Output: udp/1900

34277 - Nessus UDP Scanner

Synopsis: It is possible to determine which UDP ports are open.
Description: This plugin runs a UDP port scan against the target. It is possible to determine which UDP ports are open by sending UDP packets on every port. If the port is open, the application will most often keep quiet. If the port is closed, the TCP/IP stack may send back an ICMP Host unreachable / bad port packet. However, this is assuming there are no intermediate devices between the scanner and the target. Firewalls often block ICMP, which will prevent responses that identify closed ports. The scanning primarily relies on the absence of a response to identify open ports and in complex environments with many intermediate devices, the detection can often be unreliable. UDP scanning takes a long time to complete. The scanner must limit the number of concurrent probes because ICMP is often rate limited. Also, since open ports do not respond, the scanner must wait for a timeout period to be reasonably sure that no response will be received. Given the typical environments being scanned today, the results of this plugin should be thoroughly vetted and be used as weak signals for further investigation. It is likely that a large number of assets will be detected if there are intermediate devices between the scanner and the targets. Consider using the netstat or SNMP port enumeration options instead if possible.
Solution: Protect your target with an IP filter or implement ICMP rate limitation.
Risk Factor: None
Plugin Output: udp/3702

34277 - Nessus UDP Scanner

Synopsis: It is possible to determine which UDP ports are open.
Description: This plugin runs a UDP port scan against the target. It is possible to determine which UDP ports are open by sending UDP packets on every port. If the port is open, the application will most often keep quiet. If the port is closed, the TCP/IP stack may send back an ICMP Host unreachable / bad port packet. However, this is assuming there are no intermediate devices between the scanner and the target. Firewalls often block ICMP, which will prevent responses that identify closed ports. The scanning primarily relies on the absence of a response to identify open ports and in complex environments with many intermediate devices, the detection can often be unreliable. UDP scanning takes a long time to complete. The scanner must limit the number of concurrent probes because ICMP is often rate limited.

Also, since open ports do not respond, the scanner must wait for a timeout period to be reasonably sure that no response will be received. Given the typical environments being scanned today, the results of this plugin should be thoroughly vetted and be used as weak signals for further investigation. It is likely that a large number of assets will be detected if there are intermediate devices between the scanner and the targets. Consider using the netstat or SNMP port enumeration options instead if possible.
Solution: Protect your target with an IP filter or implement ICMP rate limitation.
Risk Factor: None
Plugin Output: udp/4500

34277 - Nessus UDP Scanner

Synopsis: It is possible to determine which UDP ports are open.
Description: This plugin runs a UDP port scan against the target. It is possible to determine which UDP ports are open by sending UDP packets on every port. If the port is open, the application will most often keep quiet. If the port is closed, the TCP/IP stack may send back an ICMP Host unreachable / bad port packet. However, this is assuming there are no intermediate devices between the scanner and the target. Firewalls often block ICMP, which will prevent responses that identify closed ports. The scanning primarily relies on the absence of a response to identify open ports and in complex environments with many intermediate devices, the detection can often be unreliable. UDP scanning takes a long time to complete. The scanner must limit the number of concurrent probes because ICMP is often rate limited. Also, since open ports do not respond, the scanner must wait for a timeout period to be reasonably sure that no response will be received. Given the typical environments being scanned today, the results of this plugin should be thoroughly vetted and be used as weak signals for further investigation. It is likely that a large number of assets will be detected if there are intermediate devices between the scanner and the targets. Consider using the netstat or SNMP port enumeration options instead if possible.
Solution: Protect your target with an IP filter or implement ICMP rate limitation.
Risk Factor: None
Plugin Output: udp/5050

34277 - Nessus UDP Scanner

Synopsis: It is possible to determine which UDP ports are open.
Description: This plugin runs a UDP port scan against the target. It is possible to determine which UDP ports are open by sending UDP packets on every port. If the port is open, the application will most often keep quiet. If the port is closed, the TCP/IP stack may send back an ICMP Host unreachable / bad port packet. However, this is assuming there are no intermediate devices between the scanner and the target. Firewalls often block ICMP, which will prevent responses that identify closed ports. The scanning primarily relies on the absence of a response to identify open ports and in complex environments with many intermediate devices, the detection can often be unreliable. UDP scanning takes a long time to complete. The scanner must limit the number of concurrent probes because ICMP is often rate limited. Also, since open ports do not respond, the scanner must wait for a timeout period to be reasonably sure that no response will be received. Given the typical environments being scanned today, the results of this plugin should be thoroughly vetted and be used as weak signals for further investigation. It is likely that a large number of assets will be detected if there are intermediate devices between the scanner and the targets. Consider using the netstat or SNMP port enumeration options instead if possible.
Solution: Protect your target with an IP filter or implement ICMP rate limitation.
Risk Factor: None
Plugin Output: udp/5353

34277 - Nessus UDP Scanner

Synopsis: It is possible to determine which UDP ports are open.
Description: This plugin runs a UDP port scan against the target. It is possible to determine which UDP ports are open by sending UDP packets on every port. If the port is open, the application will most often keep quiet. If the port is closed, the TCP/IP stack may send back an ICMP Host unreachable / bad port packet. However, this is assuming there are no intermediate devices between the scanner and the target. Firewalls often block ICMP, which will prevent responses that identify closed ports. The scanning primarily relies on the absence of a response to identify open ports and in complex environments with many intermediate devices, the detection can often be unreliable. UDP scanning takes a long time to complete. The scanner must limit the number of concurrent probes because ICMP is often rate limited.

Also, since open ports do not respond, the scanner must wait for a timeout period to be reasonably sure that no response will be received. Given the typical environments being scanned today, the results of this plugin should be thoroughly vetted and be used as weak signals for further investigation. It is likely that a large number of assets will be detected if there are intermediate devices between the scanner and the targets. Consider using the netstat or SNMP port enumeration options instead if possible.
Solution: Protect your target with an IP filter or implement ICMP rate limitation.
Risk Factor: None
Plugin Output: udp/5355/llmnr

34277 - Nessus UDP Scanner

Synopsis: It is possible to determine which UDP ports are open.
Description: This plugin runs a UDP port scan against the target. It is possible to determine which UDP ports are open by sending UDP packets on every port. If the port is open, the application will most often keep quiet. If the port is closed, the TCP/IP stack may send back an ICMP Host unreachable / bad port packet. However, this is assuming there are no intermediate devices between the scanner and the target. Firewalls often block ICMP, which will prevent responses that identify closed ports. The scanning primarily relies on the absence of a response to identify open ports and in complex environments with many intermediate devices, the detection can often be unreliable. UDP scanning takes a long time to complete. The scanner must limit the number of concurrent probes because ICMP is often rate limited. Also, since open ports do not respond, the scanner must wait for a timeout period to be reasonably sure that no response will be received. Given the typical environments being scanned today, the results of this plugin should be thoroughly vetted and be used as weak signals for further investigation. It is likely that a large number of assets will be detected if there are intermediate devices between the scanner and the targets. Consider using the netstat or SNMP port enumeration options instead if possible.
Solution: Protect your target with an IP filter or implement ICMP rate limitation.
Risk Factor: None
Plugin Output: udp/22412

34277 - Nessus UDP Scanner

Synopsis: It is possible to determine which UDP ports are open.
Description: This plugin runs a UDP port scan against the target. It is possible to determine which UDP ports are open by sending UDP packets on every port. If the port is open, the application will most often keep quiet. If the port is closed, the TCP/IP stack may send back an ICMP Host unreachable / bad port packet. However, this is assuming there are no intermediate devices between the scanner and the target. Firewalls often block ICMP, which will prevent responses that identify closed ports. The scanning primarily relies on the absence of a response to identify open ports and in complex environments with many intermediate devices, the detection can often be unreliable. UDP scanning takes a long time to complete. The scanner must limit the number of concurrent probes because ICMP is often rate limited. Also, since open ports do not respond, the scanner must wait for a timeout period to be reasonably sure that no response will be received. Given the typical environments being scanned today, the results of this plugin should be thoroughly vetted and be used as weak signals for further investigation. It is likely that a large number of assets will be detected if there are intermediate devices between the scanner and the targets. Consider using the netstat or SNMP port enumeration options instead if possible.
Solution: Protect your target with an IP filter or implement ICMP rate limitation.
Risk Factor: None
Plugin Output: udp/45678

34277 - Nessus UDP Scanner

Synopsis: It is possible to determine which UDP ports are open.
Description: This plugin runs a UDP port scan against the target. It is possible to determine which UDP ports are open by sending UDP packets on every port. If the port is open, the application will most often keep quiet. If the port is closed, the TCP/IP stack may send back an ICMP Host unreachable / bad port packet. However, this is assuming there are no intermediate devices between the scanner and the target. Firewalls often block ICMP, which will prevent responses that identify closed ports. The scanning primarily relies on the absence of a response to identify open ports and in complex environments with many intermediate devices, the detection can often be unreliable. UDP scanning takes a long time to complete. The scanner must limit the number of concurrent probes because ICMP is often rate limited.

Also, since open ports do not respond, the scanner must wait for a timeout period to be reasonably sure that no response will be received. Given the typical environments being scanned today, the results of this plugin should be thoroughly vetted and be used as weak signals for further investigation. It is likely that a large number of assets will be detected if there are intermediate devices between the scanner and the targets. Consider using the netstat or SNMP port enumeration options instead if possible.
Solution: Protect your target with an IP filter or implement ICMP rate limitation.
Risk Factor: None
Plugin Output: udp/52796

34277 - Nessus UDP Scanner

Synopsis: It is possible to determine which UDP ports are open.
Description: This plugin runs a UDP port scan against the target. It is possible to determine which UDP ports are open by sending UDP packets on every port. If the port is open, the application will most often keep quiet. If the port is closed, the TCP/IP stack may send back an ICMP Host unreachable / bad port packet. However, this is assuming there are no intermediate devices between the scanner and the target. Firewalls often block ICMP, which will prevent responses that identify closed ports. The scanning primarily relies on the absence of a response to identify open ports and in complex environments with many intermediate devices, the detection can often be unreliable. UDP scanning takes a long time to complete. The scanner must limit the number of concurrent probes because ICMP is often rate limited. Also, since open ports do not respond, the scanner must wait for a timeout period to be reasonably sure that no response will be received. Given the typical environments being scanned today, the results of this plugin should be thoroughly vetted and be used as weak signals for further investigation. It is likely that a large number of assets will be detected if there are intermediate devices between the scanner and the targets. Consider using the netstat or SNMP port enumeration options instead if possible.
Solution: Protect your target with an IP filter or implement ICMP rate limitation.
Risk Factor: None
Plugin Output: udp/54754

11936 - OS Identification

Synopsis: It is possible to guess the remote operating system.
Description: Using a combination of remote probes (e.g., TCP/IP, SMB, HTTP, NTP, SNMP, etc.), it is possible to guess the name of the remote operating system in use. It is also possible sometimes to guess the version of the operating system.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/0

21745 - OS Security Patch Assessment Failed

Synopsis: Errors prevented OS Security Patch Assessment.
Description: OS Security Patch Assessment is not available for this host because either the credentials supplied in the scan policy did not allow Nessus to log into it or some other problem occurred.
Solution: Fix the problem(s) so that OS Security Patch Assessment is possible.
Risk Factor: None
Plugin Output: tcp/0

66334 - Patch Report

Synopsis: The remote host is missing several patches.
Description: The remote host is missing one or more security patches. This plugin lists the newest version of each patch to install to make sure the remote host is up-to-date.
Solution: Install the patches listed below.
Risk Factor: None
Plugin Output: tcp/0

10180 - Ping the remote host

Synopsis: It was possible to identify the status of the remote host (alive or dead).
Description: Nessus was able to determine if the remote host is alive using one or more of the following ping types:
- An ARP ping, provided the host is on the local subnet and Nessus is running over Ethernet.
- An ICMP ping.

- A TCP ping, in which the plugin sends to the remote host a packet with the flag SYN, and the host will reply with a RST or a SYN/ACK.
- A UDP ping (e.g., DNS, RPC, and NTP).

Solution: n/a
Risk Factor: None
Plugin Output: tcp/0

56984 - SSL / TLS Versions Supported

Synopsis: The remote service encrypts communications.
Description: This plugin detects which SSL and TLS versions are supported by the remote service for encrypting communications.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/8834/www

10863 - SSL Certificate Information

Synopsis: This plugin displays the SSL certificate.
Description: This plugin connects to every SSL-related port and attempts to extract and dump the X.509 certificate.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/8834/www

21643 - SSL Cipher Suites Supported

Synopsis: The remote service encrypts communications using SSL.
Description: This plugin detects which SSL ciphers are supported by the remote service for encrypting communications.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/8834/www

57041 - SSL Perfect Forward Secrecy Cipher Suites Supported

Synopsis: The remote service supports the use of SSL Perfect Forward Secrecy ciphers, which maintain confidentiality even if the key is stolen.
Description: The remote host supports the use of SSL ciphers that offer Perfect Forward Secrecy (PFS) encryption. These cipher suites ensure that recorded SSL traffic cannot be broken at a future date if the server's private key is compromised.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/8834/www

22964 - Service Detection

Synopsis: The remote service could be identified.
Description: Nessus was able to identify the remote service by its banner or by looking at the error message it sends when it receives an HTTP request.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/80/www

22964 - Service Detection

Synopsis: The remote service could be identified.
Description: Nessus was able to identify the remote service by its banner or by looking at the error message it sends when it receives an HTTP request.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/8834/www y tcp/8834/www

42822 - Strict Transport Security (STS) Detection

Synopsis: The remote web server implements Strict Transport Security.
Description: The remote web server implements Strict Transport Security (STS). The goal of STS is to make sure that a user does not accidentally downgrade the security of his or her

browser. All unencrypted HTTP connections are redirected to HTTPS. The browser is expected to treat all cookies as 'secure' and to close the connection in the event of potentially insecure situations.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/8834/www

136318 - TLS Version 1.2 Protocol Detection

Synopsis: The remote service encrypts traffic using a version of TLS.
Description: The remote service accepts connections encrypted using TLS 1.2.
Solution: N/A
Risk Factor: None
Plugin Output: tcp/8834/www

138330 - TLS Version 1.3 Protocol Detection

Synopsis: The remote service encrypts traffic using a version of TLS.
Description: The remote service accepts connections encrypted using TLS 1.3.
Solution: N/A
Risk Factor: None
Plugin Output: tcp/8834/www

104410 - Target Credential Status by Authentication Protocol - Failure for Provided Credentials

Synopsis: Nessus was unable to log into the detected authentication protocol, using the provided credentials, in order to perform credentialed checks.
Description: Nessus failed to successfully authenticate directly to the remote target on an available authentication protocol. Nessus was able to connect to the remote port and identify that the service running on the port supports an authentication protocol, but Nessus failed to authenticate to the remote service using the provided credentials. There may have been a failure in protocol negotiation or communication that prevented authentication from being attempted or all of the provided credentials for the authentication protocol may have been invalid. A protocol failure may indicate a compatibility issue with the protocol configuration. A protocol failure due to an environmental issue such as resource or congestion issues may also prevent valid credentials from being identified. See plugin output for error details. Please note the following:
- This plugin reports per protocol, so it is possible for valid credentials to be provided for one protocol and not another. For example, authentication may succeed via SSH but fail via SMB, while no credentials were provided for an available SNMP service.
- Providing valid credentials for all available authentication protocols may improve scan coverage, but the value of successful authentication for a given protocol may vary from target to target depending upon what data (if any) is gathered from the target via that protocol. For example, successful authentication via SSH is more valuable for Linux targets than for Windows targets, and likewise successful authentication via SMB is more valuable for Windows targets than for Linux targets.
Solution: Address the reported problem(s) so that credentialed checks can be executed.
Risk Factor: None
Plugin Output: tcp/445/cifs

10287 - Traceroute Information

Synopsis: It was possible to obtain traceroute information.
Description: Makes a traceroute to the remote host.
Solution: n/a
Risk Factor: None
Plugin Output: udp/0

20094 - VMware Virtual Machine Detection

Synopsis: The remote host is a VMware virtual machine.
Description: According to the MAC address of its network adapter, the remote host is a VMware virtual machine.
Solution: Since it is physically accessible through the network, ensure that its configuration matches your organization's security policy.
Risk Factor: None
Plugin Output: tcp/0

135860 - WMI Not Available

Synopsis: WMI queries could not be made against the remote host.
Description: WMI (Windows Management Instrumentation) is not available on the remote host over DCOM. WMI queries are used to gather information about the remote host, such as its current state, network interface configuration, etc. Without this information Nessus may not be able to identify installed software or security vunerabilities that exist on the remote host.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/445/cifs

91815 - Web Application Sitemap

Synopsis: The remote web server hosts linkable content that can be crawled by Nessus.
Description: The remote web server contains linkable content that can be used to gather information about a target.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/8834/www

11032 - Web Server Directory Enumeration

Synopsis: It is possible to enumerate directories on the web server.
Description: This plugin attempts to determine the presence of various common directories on the remote web server. By sending a request for a directory, the web server response code indicates if it is a valid directory or not.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/8834/www

10150 - Windows NetBIOS / SMB Remote Host Information Disclosure

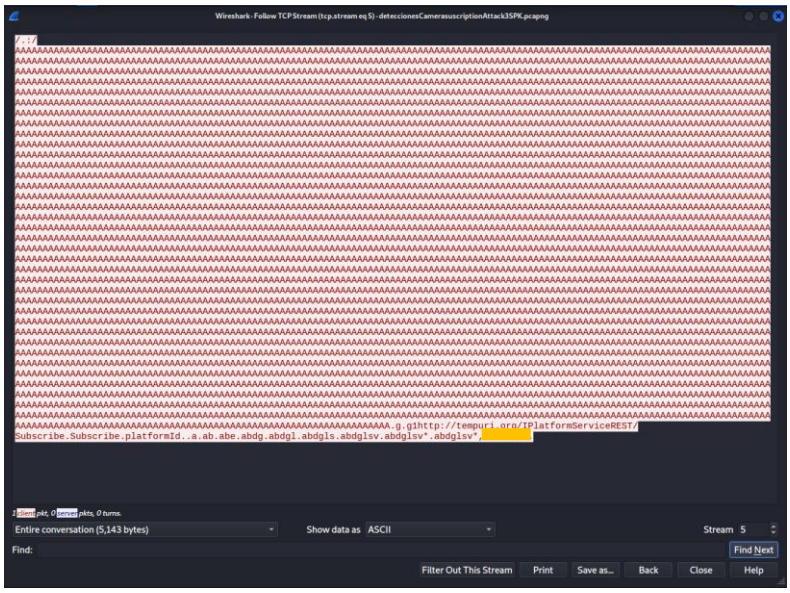
Synopsis: It was possible to obtain the network name of the remote host.
Description: The remote host is listening on UDP port 137 or TCP port 445, and replies to NetBIOS nbtscan or SMB requests. Note that this plugin gathers information to be used in other plugins, but does not itself generate a report.
Solution: n/a
Risk Factor: None
Plugin Output: tcp/445/cifs

Anexo D. Informe de Spike y Wireshark

En este anexo se incluyen las conversaciones de las diferentes pruebas de fuzzing con el servidor obtenidos con Wireshark y una parte de la salida de una prueba de fuzzing con Spike.

Las conversaciones de las diferentes pruebas de fuzzing se caracterizan por no tener respuesta del servidor. A continuación, se muestran las conversaciones:

Figura 98. *Conversación petición de la prueba de fuzzing-Servidor para Suscribirse al Sensor Cámara en el puerto 33003*

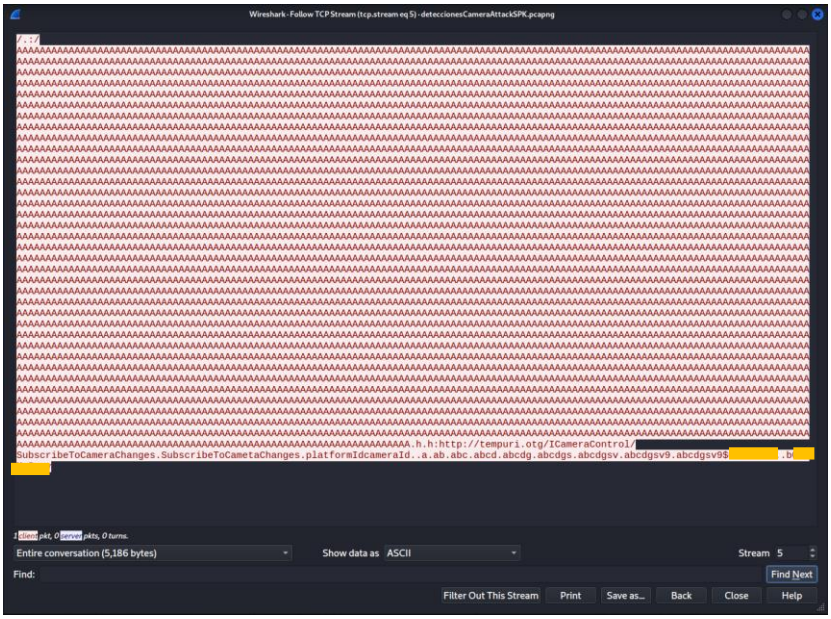


Fuente: Elaboración propia.

Figura 99. *Conversación petición de la prueba de fuzzing-Servidor para Movimientos del Sensor Cámara en el puerto 33003*

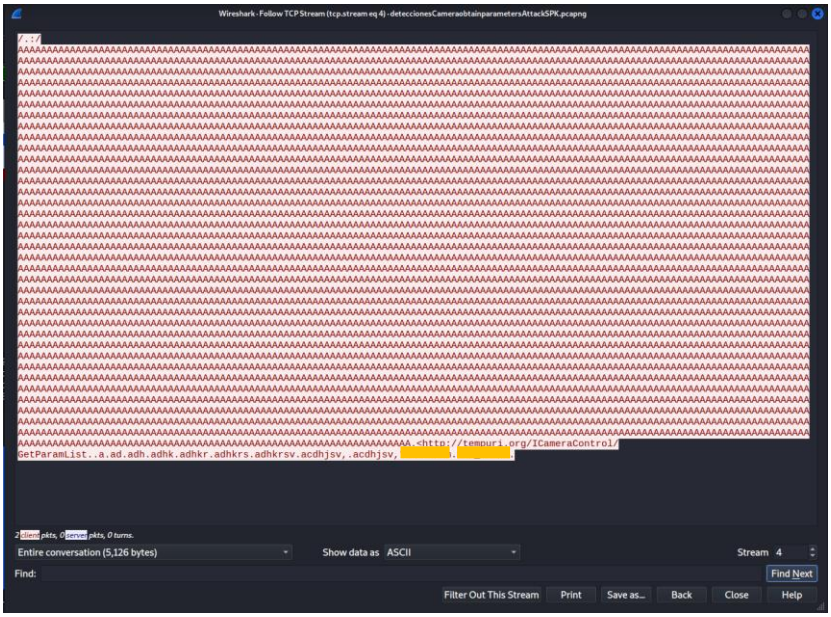
Fuente: Elaboración propia.

Figura 100. *Conversación petición de la prueba de fuzzing-Servidor para Suscribirse al Sensor Cámara en el puerto 33005*



Fuente: Elaboración propia.

Figura 101. *Conversación petición de la prueba de fuzzing-Servidor para obtener Parámetros al Sensor Cámara en el puerto 33005*



Fuente: Elaboración propia.

Finalmente, se muestran unas trazas que deja Spike durante las pruebas de fuzzing de cada variable programada en el script. En la *Tabla 96* se pueden observar las peticiones que realiza una prueba de fuzzing con una única variable y el tamaño de la variable de la prueba de fuzzing.

En concreto, estas variables de la prueba de fuzzing se corresponden con la de Suscripción a un Sensor Cámara en el puerto 33005.

Tabla 96. ***Datos del fuzzeo de la variable 0 del script de la Suscripción de un Sensor Cámara en el puerto 33005***

Total Number of Strings is 681	Fuzzing Variable 0:227	Variablesize= 5000	Fuzzing Variable 0:682	Fuzzing Variable 0:1137	Fuzzing Variable 0:1592
Fuzzing	Variablesize= 4095	Fuzzing Variable 0:455	Fuzzing Variable 0:683	Fuzzing Variable 0:1138	Fuzzing Variable 0:1593
Fuzzing Variable 0:0	Fuzzing Variable 0:228	Variablesize= 4097	Fuzzing Variable 0:684	Fuzzing Variable 0:1139	Fuzzing Variable 0:1594
Fuzzing Variable 0:1	Variablesize= 2048	Fuzzing Variable 0:456	Fuzzing Variable 0:685	Fuzzing Variable 0:1140	Fuzzing Variable 0:1595
Variablesize= 5004	Fuzzing Variable 0:229	Variablesize= 4096	Fuzzing Variable 0:686	Fuzzing Variable 0:1141	Fuzzing Variable 0:1596
Fuzzing Variable 0:2	Variablesize= 1024	Fuzzing Variable 0:457	Fuzzing Variable 0:687	Fuzzing Variable 0:1142	Fuzzing Variable 0:1597
Variablesize= 5005	Fuzzing Variable 0:230	Variablesize= 4095	Fuzzing Variable 0:688	Fuzzing Variable 0:1143	Fuzzing Variable 0:1598
Fuzzing Variable 0:3	Variablesize= 1023	Fuzzing Variable 0:458	Fuzzing Variable 0:689	Fuzzing Variable 0:1144	Fuzzing Variable 0:1599
Variablesize= 21	Fuzzing Variable 0:231	Variablesize= 2048	Fuzzing Variable 0:690	Fuzzing Variable 0:1145	Fuzzing Variable 0:1600
Fuzzing Variable 0:4	Variablesize= 512	Fuzzing Variable 0:459	Fuzzing Variable 0:691	Fuzzing Variable 0:1146	Fuzzing Variable 0:1601
Variablesize= 3	Fuzzing Variable 0:232	Variablesize= 1024	Fuzzing Variable 0:692	Fuzzing Variable 0:1147	Fuzzing Variable 0:1602
Fuzzing Variable 0:5	Variablesize= 420	Fuzzing Variable 0:460	Fuzzing Variable 0:693	Fuzzing Variable 0:1148	Fuzzing Variable 0:1603
Variablesize= 2	Fuzzing Variable 0:233	Variablesize= 1023	Fuzzing Variable 0:694	Fuzzing Variable 0:1149	Fuzzing Variable 0:1604
Fuzzing Variable 0:6	Variablesize= 257	Fuzzing Variable 0:461	Fuzzing Variable 0:695	Fuzzing Variable 0:1150	Fuzzing Variable 0:1605
Variablesize= 7	Fuzzing Variable 0:234	Variablesize= 512	Fuzzing Variable 0:696	Fuzzing Variable 0:1151	Fuzzing Variable 0:1606
Fuzzing Variable 0:7	Variablesize= 256	Fuzzing Variable 0:462	Fuzzing Variable 0:697	Fuzzing Variable 0:1152	Fuzzing Variable 0:1607
Variablesize= 48	Fuzzing Variable 0:235	Variablesize= 420	Fuzzing Variable 0:698	Fuzzing Variable 0:1153	Fuzzing Variable 0:1608
Fuzzing Variable 0:8	Variablesize= 240	Fuzzing Variable 0:463	Fuzzing Variable 0:699	Fuzzing Variable 0:1154	Fuzzing Variable 0:1609
Variablesize= 45	Fuzzing Variable 0:236	Variablesize= 257	Fuzzing Variable 0:700	Fuzzing Variable 0:1155	Fuzzing Variable 0:1610
Fuzzing Variable 0:9	Variablesize= 128	Fuzzing Variable 0:464	Fuzzing Variable 0:701	Fuzzing Variable 0:1156	Fuzzing Variable 0:1611
Variablesize= 49	Fuzzing Variable 0:237	Variablesize= 256	Fuzzing Variable 0:702	Fuzzing Variable 0:1157	Fuzzing Variable 0:1612
Fuzzing Variable 0:10	Variablesize= 65534	Fuzzing Variable 0:465	Fuzzing Variable 0:703	Fuzzing Variable 0:1158	Fuzzing Variable 0:1613
Variablesize= 46	Fuzzing Variable 0:238	Variablesize= 240	Fuzzing Variable 0:704	Fuzzing Variable 0:1159	Fuzzing Variable 0:1614
Fuzzing Variable 0:11	Variablesize= 32768	Fuzzing Variable 0:466	Fuzzing Variable 0:705	Fuzzing Variable 0:1160	Fuzzing Variable 0:1615
Variablesize= 49	Fuzzing Variable 0:239	Variablesize= 128	Fuzzing Variable 0:706	Fuzzing Variable 0:1161	Fuzzing Variable 0:1616
Fuzzing Variable 0:12	Variablesize= 32767	Fuzzing Variable 0:467	Fuzzing Variable 0:707	Fuzzing Variable 0:1162	Fuzzing Variable 0:1617
Variablesize= 46	Fuzzing Variable 0:240	Variablesize= 65534	Fuzzing Variable 0:708	Fuzzing Variable 0:1163	Fuzzing Variable 0:1618
Fuzzing Variable 0:13	Variablesize= 32766	Fuzzing Variable 0:468	Fuzzing Variable 0:709	Fuzzing Variable 0:1164	Fuzzing Variable 0:1619
Variablesize= 47	Fuzzing Variable 0:241	Variablesize= 32768	Fuzzing Variable 0:710	Fuzzing Variable 0:1165	Fuzzing Variable 0:1620
Fuzzing Variable 0:14	Variablesize= 32765	Fuzzing Variable 0:469	Fuzzing Variable 0:711	Fuzzing Variable 0:1166	Fuzzing Variable 0:1621
Variablesize= 44	Fuzzing Variable 0:242	Variablesize= 32767	Fuzzing Variable 0:712	Fuzzing Variable 0:1167	Fuzzing Variable 0:1622
Fuzzing Variable 0:15	Variablesize= 32764	Fuzzing Variable 0:470	Fuzzing Variable 0:713	Fuzzing Variable 0:1168	Fuzzing Variable 0:1623
Variablesize= 53	Fuzzing Variable 0:243	Variablesize= 32766	Fuzzing Variable 0:714	Fuzzing Variable 0:1169	Fuzzing Variable 0:1624
Fuzzing Variable 0:16	Variablesize= 32763	Fuzzing Variable 0:471	Fuzzing Variable 0:715	Fuzzing Variable 0:1170	Fuzzing Variable 0:1625

Variablesize= 50	Fuzzing Variable 0:244	Variablesize= 32765	Fuzzing Variable 0:716	Fuzzing Variable 0:1171	Fuzzing Variable 0:1626
Fuzzing Variable 0:17	Variablesize= 32762	Fuzzing Variable 0:472	Fuzzing Variable 0:717	Fuzzing Variable 0:1172	Fuzzing Variable 0:1627
Variablesize= 30	Fuzzing Variable 0:245	Variablesize= 32764	Fuzzing Variable 0:718	Fuzzing Variable 0:1173	Fuzzing Variable 0:1628
Fuzzing Variable 0:18	Variablesize= 20000	Fuzzing Variable 0:473	Fuzzing Variable 0:719	Fuzzing Variable 0:1174	Fuzzing Variable 0:1629
Variablesize= 23	Fuzzing Variable 0:246	Variablesize= 32763	Fuzzing Variable 0:720	Fuzzing Variable 0:1175	Fuzzing Variable 0:1630
Fuzzing Variable 0:19	Variablesize= 10000	Fuzzing Variable 0:474	Fuzzing Variable 0:721	Fuzzing Variable 0:1176	Fuzzing Variable 0:1631
Variablesize= 48	Fuzzing Variable 0:247	Variablesize= 32762	Fuzzing Variable 0:722	Fuzzing Variable 0:1177	Fuzzing Variable 0:1632
Fuzzing Variable 0:20	Variablesize= 5000	Fuzzing Variable 0:475	Fuzzing Variable 0:723	Fuzzing Variable 0:1178	Fuzzing Variable 0:1633
Variablesize= 36	Fuzzing Variable 0:248	Variablesize= 20000	Fuzzing Variable 0:724	Fuzzing Variable 0:1179	Fuzzing Variable 0:1634
Fuzzing Variable 0:21	Variablesize= 4097	Fuzzing Variable 0:476	Fuzzing Variable 0:725	Fuzzing Variable 0:1180	Fuzzing Variable 0:1635
Variablesize= 18	Fuzzing Variable 0:249	Variablesize= 10000	Fuzzing Variable 0:726	Fuzzing Variable 0:1181	Fuzzing Variable 0:1636
Fuzzing Variable 0:22	Variablesize= 4096	Fuzzing Variable 0:477	Fuzzing Variable 0:727	Fuzzing Variable 0:1182	Fuzzing Variable 0:1637
Variablesize= 15	Fuzzing Variable 0:250	Variablesize= 5000	Fuzzing Variable 0:728	Fuzzing Variable 0:1183	Fuzzing Variable 0:1638
Fuzzing Variable 0:23	Variablesize= 4095	Fuzzing Variable 0:478	Fuzzing Variable 0:729	Fuzzing Variable 0:1184	Fuzzing Variable 0:1639
Variablesize= 16	Fuzzing Variable 0:251	Variablesize= 4097	Fuzzing Variable 0:730	Fuzzing Variable 0:1185	Fuzzing Variable 0:1640
Fuzzing Variable 0:24	Variablesize= 2048	Fuzzing Variable 0:479	Fuzzing Variable 0:731	Fuzzing Variable 0:1186	Fuzzing Variable 0:1641
Variablesize= 3	Fuzzing Variable 0:252	Variablesize= 4096	Fuzzing Variable 0:732	Fuzzing Variable 0:1187	Fuzzing Variable 0:1642
Fuzzing Variable 0:25	Variablesize= 1024	Fuzzing Variable 0:480	Fuzzing Variable 0:733	Fuzzing Variable 0:1188	Fuzzing Variable 0:1643
Variablesize= 4	Fuzzing Variable 0:253	Variablesize= 4095	Fuzzing Variable 0:734	Fuzzing Variable 0:1189	Fuzzing Variable 0:1644
Fuzzing Variable 0:26	Variablesize= 1023	Fuzzing Variable 0:481	Fuzzing Variable 0:735	Fuzzing Variable 0:1190	Fuzzing Variable 0:1645
Variablesize= 5000	Fuzzing Variable 0:254	Variablesize= 2048	Fuzzing Variable 0:736	Fuzzing Variable 0:1191	Fuzzing Variable 0:1646
Fuzzing Variable 0:27	Variablesize= 512	Fuzzing Variable 0:482	Fuzzing Variable 0:737	Fuzzing Variable 0:1192	Fuzzing Variable 0:1647
Variablesize= 5000	Fuzzing Variable 0:255	Variablesize= 1024	Fuzzing Variable 0:738	Fuzzing Variable 0:1193	Fuzzing Variable 0:1648
Fuzzing Variable 0:28	Variablesize= 420	Fuzzing Variable 0:483	Fuzzing Variable 0:739	Fuzzing Variable 0:1194	Fuzzing Variable 0:1649
Variablesize= 5000	Fuzzing Variable 0:256	Variablesize= 1023	Fuzzing Variable 0:740	Fuzzing Variable 0:1195	Fuzzing Variable 0:1650
Fuzzing Variable 0:29	Variablesize= 257	Fuzzing Variable 0:484	Fuzzing Variable 0:741	Fuzzing Variable 0:1196	Fuzzing Variable 0:1651
Variablesize= 5000	Fuzzing Variable 0:257	Variablesize= 512	Fuzzing Variable 0:742	Fuzzing Variable 0:1197	Fuzzing Variable 0:1652
Fuzzing Variable 0:30	Variablesize= 256	Fuzzing Variable 0:485	Fuzzing Variable 0:743	Fuzzing Variable 0:1198	Fuzzing Variable 0:1653
Variablesize= 2050	Fuzzing Variable 0:258	Variablesize= 420	Fuzzing Variable 0:744	Fuzzing Variable 0:1199	Fuzzing Variable 0:1654
Fuzzing Variable 0:31	Variablesize= 240	Fuzzing Variable 0:486	Fuzzing Variable 0:745	Fuzzing Variable 0:1200	Fuzzing Variable 0:1655
Variablesize= 115	Fuzzing Variable 0:259	Variablesize= 257	Fuzzing Variable 0:746	Fuzzing Variable 0:1201	Fuzzing Variable 0:1656
Fuzzing Variable 0:32	Variablesize= 128	Fuzzing Variable 0:487	Fuzzing Variable 0:747	Fuzzing Variable 0:1202	Fuzzing Variable 0:1657
Variablesize= 130	Fuzzing Variable 0:260	Variablesize= 256	Fuzzing Variable 0:748	Fuzzing Variable 0:1203	Fuzzing Variable 0:1658
Fuzzing Variable 0:33	Variablesize= 65534	Fuzzing Variable 0:488	Fuzzing Variable 0:749	Fuzzing Variable 0:1204	Fuzzing Variable 0:1659
Variablesize= 125	Fuzzing Variable 0:261	Variablesize= 240	Fuzzing Variable 0:750	Fuzzing Variable 0:1205	Fuzzing Variable 0:1660
Fuzzing Variable 0:34	Variablesize= 32768	Fuzzing Variable 0:489	Fuzzing Variable 0:751	Fuzzing Variable 0:1206	Fuzzing Variable 0:1661
Variablesize= 116	Fuzzing Variable 0:262	Variablesize= 128	Fuzzing Variable 0:752	Fuzzing Variable 0:1207	Fuzzing Variable 0:1662
Fuzzing Variable 0:35	Variablesize= 32767	Fuzzing Variable 0:490	Fuzzing Variable 0:753	Fuzzing Variable 0:1208	Fuzzing Variable 0:1663
Variablesize= 130	Fuzzing Variable 0:263	Variablesize= 65534	Fuzzing Variable 0:754	Fuzzing Variable 0:1209	Fuzzing Variable 0:1664

Fuzzing Variable 0:36	Variablesize= 32766	Fuzzing Variable 0:491	Fuzzing Variable 0:755	Fuzzing Variable 0:1210	Fuzzing Variable 0:1665
Variablesize= 5	Fuzzing Variable 0:264	Variablesize= 32768	Fuzzing Variable 0:756	Fuzzing Variable 0:1211	Fuzzing Variable 0:1666
Fuzzing Variable 0:37	Variablesize= 32765	Fuzzing Variable 0:492	Fuzzing Variable 0:757	Fuzzing Variable 0:1212	Fuzzing Variable 0:1667
Variablesize= 9	Fuzzing Variable 0:265	Variablesize= 32767	Fuzzing Variable 0:758	Fuzzing Variable 0:1213	Fuzzing Variable 0:1668
Fuzzing Variable 0:38	Variablesize= 32764	Fuzzing Variable 0:493	Fuzzing Variable 0:759	Fuzzing Variable 0:1214	Fuzzing Variable 0:1669
Variablesize= 7	Fuzzing Variable 0:266	Variablesize= 32766	Fuzzing Variable 0:760	Fuzzing Variable 0:1215	Fuzzing Variable 0:1670
Fuzzing Variable 0:39	Variablesize= 32763	Fuzzing Variable 0:494	Fuzzing Variable 0:761	Fuzzing Variable 0:1216	Fuzzing Variable 0:1671
Variablesize= 9	Fuzzing Variable 0:267	Variablesize= 32765	Fuzzing Variable 0:762	Fuzzing Variable 0:1217	Fuzzing Variable 0:1672
Fuzzing Variable 0:40	Variablesize= 32762	Fuzzing Variable 0:495	Fuzzing Variable 0:763	Fuzzing Variable 0:1218	Fuzzing Variable 0:1673
Variablesize= 1	Fuzzing Variable 0:268	Variablesize= 32764	Fuzzing Variable 0:764	Fuzzing Variable 0:1219	Fuzzing Variable 0:1674
Fuzzing Variable 0:41	Variablesize= 20000	Fuzzing Variable 0:496	Fuzzing Variable 0:765	Fuzzing Variable 0:1220	Fuzzing Variable 0:1675
Variablesize= 1	Fuzzing Variable 0:269	Variablesize= 32763	Fuzzing Variable 0:766	Fuzzing Variable 0:1221	Fuzzing Variable 0:1676
Fuzzing Variable 0:42	Variablesize= 10000	Fuzzing Variable 0:497	Fuzzing Variable 0:767	Fuzzing Variable 0:1222	Fuzzing Variable 0:1677
Variablesize= 2	Fuzzing Variable 0:270	Variablesize= 32762	Fuzzing Variable 0:768	Fuzzing Variable 0:1223	Fuzzing Variable 0:1678
Fuzzing Variable 0:43	Variablesize= 5000	Fuzzing Variable 0:498	Fuzzing Variable 0:769	Fuzzing Variable 0:1224	Fuzzing Variable 0:1679
Variablesize= 10	Fuzzing Variable 0:271	Variablesize= 20000	Fuzzing Variable 0:770	Fuzzing Variable 0:1225	Fuzzing Variable 0:1680
Fuzzing Variable 0:44	Variablesize= 4097	Fuzzing Variable 0:499	Fuzzing Variable 0:771	Fuzzing Variable 0:1226	Fuzzing Variable 0:1681
Variablesize= 10	Fuzzing Variable 0:272	Variablesize= 10000	Fuzzing Variable 0:772	Fuzzing Variable 0:1227	Fuzzing Variable 0:1682
Fuzzing Variable 0:45	Variablesize= 4096	Fuzzing Variable 0:500	Fuzzing Variable 0:773	Fuzzing Variable 0:1228	Fuzzing Variable 0:1683
Variablesize= 11	Fuzzing Variable 0:273	Variablesize= 5000	Fuzzing Variable 0:774	Fuzzing Variable 0:1229	Fuzzing Variable 0:1684
Fuzzing Variable 0:46	Variablesize= 4095	Fuzzing Variable 0:501	Fuzzing Variable 0:775	Fuzzing Variable 0:1230	Fuzzing Variable 0:1685
Variablesize= 10	Fuzzing Variable 0:274	Variablesize= 4097	Fuzzing Variable 0:776	Fuzzing Variable 0:1231	Fuzzing Variable 0:1686
Fuzzing Variable 0:47	Variablesize= 2048	Fuzzing Variable 0:502	Fuzzing Variable 0:777	Fuzzing Variable 0:1232	Fuzzing Variable 0:1687
Variablesize= 3	Fuzzing Variable 0:275	Variablesize= 4096	Fuzzing Variable 0:778	Fuzzing Variable 0:1233	Fuzzing Variable 0:1688
Fuzzing Variable 0:48	Variablesize= 1024	Fuzzing Variable 0:503	Fuzzing Variable 0:779	Fuzzing Variable 0:1234	Fuzzing Variable 0:1689
Variablesize= 9	Fuzzing Variable 0:276	Variablesize= 4095	Fuzzing Variable 0:780	Fuzzing Variable 0:1235	Fuzzing Variable 0:1690
Fuzzing Variable 0:49	Variablesize= 1023	Fuzzing Variable 0:504	Fuzzing Variable 0:781	Fuzzing Variable 0:1236	Fuzzing Variable 0:1691
Variablesize= 3600	Fuzzing Variable 0:277	Variablesize= 2048	Fuzzing Variable 0:782	Fuzzing Variable 0:1237	Fuzzing Variable 0:1692
Fuzzing Variable 0:50	Variablesize= 512	Fuzzing Variable 0:505	Fuzzing Variable 0:783	Fuzzing Variable 0:1238	Fuzzing Variable 0:1693
Variablesize= 2400	Fuzzing Variable 0:278	Variablesize= 1024	Fuzzing Variable 0:784	Fuzzing Variable 0:1239	Fuzzing Variable 0:1694
Fuzzing Variable 0:51	Variablesize= 420	Fuzzing Variable 0:506	Fuzzing Variable 0:785	Fuzzing Variable 0:1240	Fuzzing Variable 0:1695
Variablesize= 4800	Fuzzing Variable 0:279	Variablesize= 1023	Fuzzing Variable 0:786	Fuzzing Variable 0:1241	Fuzzing Variable 0:1696
Fuzzing Variable 0:52	Variablesize= 257	Fuzzing Variable 0:507	Fuzzing Variable 0:787	Fuzzing Variable 0:1242	Fuzzing Variable 0:1697
Variablesize= 9	Fuzzing Variable 0:280	Variablesize= 512	Fuzzing Variable 0:788	Fuzzing Variable 0:1243	Fuzzing Variable 0:1698
Fuzzing Variable 0:53	Variablesize= 256	Fuzzing Variable 0:508	Fuzzing Variable 0:789	Fuzzing Variable 0:1244	Fuzzing Variable 0:1699
Variablesize= 65534	Fuzzing Variable 0:281	Variablesize= 420	Fuzzing Variable 0:790	Fuzzing Variable 0:1245	Fuzzing Variable 0:1700
Fuzzing Variable 0:54	Variablesize= 240	Fuzzing Variable 0:509	Fuzzing Variable 0:791	Fuzzing Variable 0:1246	Fuzzing Variable 0:1701
Variablesize= 32768	Fuzzing Variable 0:282	Variablesize= 257	Fuzzing Variable 0:792	Fuzzing Variable 0:1247	Fuzzing Variable 0:1702
Fuzzing Variable 0:55	Variablesize= 128	Fuzzing Variable 0:510	Fuzzing Variable 0:793	Fuzzing Variable 0:1248	Fuzzing Variable 0:1703

Variablesize= 32767	Fuzzing Variable 0:283	Variablesize= 256	Fuzzing Variable 0:794	Fuzzing Variable 0:1249	Fuzzing Variable 0:1704
Fuzzing Variable 0:56	Variablesize= 65534	Fuzzing Variable 0:511	Fuzzing Variable 0:795	Fuzzing Variable 0:1250	Fuzzing Variable 0:1705
Variablesize= 32766	Fuzzing Variable 0:284	Variablesize= 240	Fuzzing Variable 0:796	Fuzzing Variable 0:1251	Fuzzing Variable 0:1706
Fuzzing Variable 0:57	Variablesize= 32768	Fuzzing Variable 0:512	Fuzzing Variable 0:797	Fuzzing Variable 0:1252	Fuzzing Variable 0:1707
Variablesize= 32765	Fuzzing Variable 0:285	Variablesize= 128	Fuzzing Variable 0:798	Fuzzing Variable 0:1253	Fuzzing Variable 0:1708
Fuzzing Variable 0:58	Variablesize= 32767	Fuzzing Variable 0:513	Fuzzing Variable 0:799	Fuzzing Variable 0:1254	Fuzzing Variable 0:1709
Variablesize= 32764	Fuzzing Variable 0:286	Variablesize= 65534	Fuzzing Variable 0:800	Fuzzing Variable 0:1255	Fuzzing Variable 0:1710
Fuzzing Variable 0:59	Variablesize= 32766	Fuzzing Variable 0:514	Fuzzing Variable 0:801	Fuzzing Variable 0:1256	Fuzzing Variable 0:1711
Variablesize= 32763	Fuzzing Variable 0:287	Variablesize= 32768	Fuzzing Variable 0:802	Fuzzing Variable 0:1257	Fuzzing Variable 0:1712
Fuzzing Variable 0:60	Variablesize= 32765	Fuzzing Variable 0:515	Fuzzing Variable 0:803	Fuzzing Variable 0:1258	Fuzzing Variable 0:1713
Variablesize= 32762	Fuzzing Variable 0:288	Variablesize= 32767	Fuzzing Variable 0:804	Fuzzing Variable 0:1259	Fuzzing Variable 0:1714
Fuzzing Variable 0:61	Variablesize= 32764	Fuzzing Variable 0:516	Fuzzing Variable 0:805	Fuzzing Variable 0:1260	Fuzzing Variable 0:1715
Variablesize= 20000	Fuzzing Variable 0:289	Variablesize= 32766	Fuzzing Variable 0:806	Fuzzing Variable 0:1261	Fuzzing Variable 0:1716
Fuzzing Variable 0:62	Variablesize= 32763	Fuzzing Variable 0:517	Fuzzing Variable 0:807	Fuzzing Variable 0:1262	Fuzzing Variable 0:1717
Variablesize= 10000	Fuzzing Variable 0:290	Variablesize= 32765	Fuzzing Variable 0:808	Fuzzing Variable 0:1263	Fuzzing Variable 0:1718
Fuzzing Variable 0:63	Variablesize= 32762	Fuzzing Variable 0:518	Fuzzing Variable 0:809	Fuzzing Variable 0:1264	Fuzzing Variable 0:1719
Variablesize= 5000	Fuzzing Variable 0:291	Variablesize= 32764	Fuzzing Variable 0:810	Fuzzing Variable 0:1265	Fuzzing Variable 0:1720
Fuzzing Variable 0:64	Variablesize= 20000	Fuzzing Variable 0:519	Fuzzing Variable 0:811	Fuzzing Variable 0:1266	Fuzzing Variable 0:1721
Variablesize= 4097	Fuzzing Variable 0:292	Variablesize= 32763	Fuzzing Variable 0:812	Fuzzing Variable 0:1267	Fuzzing Variable 0:1722
Fuzzing Variable 0:65	Variablesize= 10000	Fuzzing Variable 0:520	Fuzzing Variable 0:813	Fuzzing Variable 0:1268	Fuzzing Variable 0:1723
Variablesize= 4096	Fuzzing Variable 0:293	Variablesize= 32762	Fuzzing Variable 0:814	Fuzzing Variable 0:1269	Fuzzing Variable 0:1724
Fuzzing Variable 0:66	Variablesize= 5000	Fuzzing Variable 0:521	Fuzzing Variable 0:815	Fuzzing Variable 0:1270	Fuzzing Variable 0:1725
Variablesize= 4095	Fuzzing Variable 0:294	Variablesize= 20000	Fuzzing Variable 0:816	Fuzzing Variable 0:1271	Fuzzing Variable 0:1726
Fuzzing Variable 0:67	Variablesize= 4097	Fuzzing Variable 0:522	Fuzzing Variable 0:817	Fuzzing Variable 0:1272	Fuzzing Variable 0:1727
Variablesize= 2048	Fuzzing Variable 0:295	Variablesize= 10000	Fuzzing Variable 0:818	Fuzzing Variable 0:1273	Fuzzing Variable 0:1728
Fuzzing Variable 0:68	Variablesize= 4096	Fuzzing Variable 0:523	Fuzzing Variable 0:819	Fuzzing Variable 0:1274	Fuzzing Variable 0:1729
Variablesize= 1024	Fuzzing Variable 0:296	Variablesize= 5000	Fuzzing Variable 0:820	Fuzzing Variable 0:1275	Fuzzing Variable 0:1730
Fuzzing Variable 0:69	Variablesize= 4095	Fuzzing Variable 0:524	Fuzzing Variable 0:821	Fuzzing Variable 0:1276	Fuzzing Variable 0:1731
Variablesize= 1023	Fuzzing Variable 0:297	Variablesize= 4097	Fuzzing Variable 0:822	Fuzzing Variable 0:1277	Fuzzing Variable 0:1732
Fuzzing Variable 0:70	Variablesize= 2048	Fuzzing Variable 0:525	Fuzzing Variable 0:823	Fuzzing Variable 0:1278	Fuzzing Variable 0:1733
Variablesize= 512	Fuzzing Variable 0:298	Variablesize= 4096	Fuzzing Variable 0:824	Fuzzing Variable 0:1279	Fuzzing Variable 0:1734
Fuzzing Variable 0:71	Variablesize= 1024	Fuzzing Variable 0:526	Fuzzing Variable 0:825	Fuzzing Variable 0:1280	Fuzzing Variable 0:1735
Variablesize= 420	Fuzzing Variable 0:299	Variablesize= 4095	Fuzzing Variable 0:826	Fuzzing Variable 0:1281	Fuzzing Variable 0:1736
Fuzzing Variable 0:72	Variablesize= 1023	Fuzzing Variable 0:527	Fuzzing Variable 0:827	Fuzzing Variable 0:1282	Fuzzing Variable 0:1737
Variablesize= 257	Fuzzing Variable 0:300	Variablesize= 2048	Fuzzing Variable 0:828	Fuzzing Variable 0:1283	Fuzzing Variable 0:1738
Fuzzing Variable 0:73	Variablesize= 512	Fuzzing Variable 0:528	Fuzzing Variable 0:829	Fuzzing Variable 0:1284	Fuzzing Variable 0:1739
Variablesize= 256	Fuzzing Variable 0:301	Variablesize= 1024	Fuzzing Variable 0:830	Fuzzing Variable 0:1285	Fuzzing Variable 0:1740
Fuzzing Variable 0:74	Variablesize= 420	Fuzzing Variable 0:529	Fuzzing Variable 0:831	Fuzzing Variable 0:1286	Fuzzing Variable 0:1741
Variablesize= 240	Fuzzing Variable 0:302	Variablesize= 1023	Fuzzing Variable 0:832	Fuzzing Variable 0:1287	Fuzzing Variable 0:1742

Fuzzing Variable 0:75	Variablesize= 257	Fuzzing Variable 0:530	Fuzzing Variable 0:833	Fuzzing Variable 0:1288	Fuzzing Variable 0:1743
Variablesize= 128	Fuzzing Variable 0:303	Variablesize= 512	Fuzzing Variable 0:834	Fuzzing Variable 0:1289	Fuzzing Variable 0:1744
Fuzzing Variable 0:76	Variablesize= 256	Fuzzing Variable 0:531	Fuzzing Variable 0:835	Fuzzing Variable 0:1290	Fuzzing Variable 0:1745
Variablesize= 65534	Fuzzing Variable 0:304	Variablesize= 420	Fuzzing Variable 0:836	Fuzzing Variable 0:1291	Fuzzing Variable 0:1746
Fuzzing Variable 0:77	Variablesize= 240	Fuzzing Variable 0:532	Fuzzing Variable 0:837	Fuzzing Variable 0:1292	Fuzzing Variable 0:1747
Variablesize= 32768	Fuzzing Variable 0:305	Variablesize= 257	Fuzzing Variable 0:838	Fuzzing Variable 0:1293	Fuzzing Variable 0:1748
Fuzzing Variable 0:78	Variablesize= 128	Fuzzing Variable 0:533	Fuzzing Variable 0:839	Fuzzing Variable 0:1294	Fuzzing Variable 0:1749
Variablesize= 32767	Fuzzing Variable 0:306	Variablesize= 256	Fuzzing Variable 0:840	Fuzzing Variable 0:1295	Fuzzing Variable 0:1750
Fuzzing Variable 0:79	Variablesize= 65534	Fuzzing Variable 0:534	Fuzzing Variable 0:841	Fuzzing Variable 0:1296	Fuzzing Variable 0:1751
Variablesize= 32766	Fuzzing Variable 0:307	Variablesize= 240	Fuzzing Variable 0:842	Fuzzing Variable 0:1297	Fuzzing Variable 0:1752
Fuzzing Variable 0:80	Variablesize= 32768	Fuzzing Variable 0:535	Fuzzing Variable 0:843	Fuzzing Variable 0:1298	Fuzzing Variable 0:1753
Variablesize= 32765	Fuzzing Variable 0:308	Variablesize= 128	Fuzzing Variable 0:844	Fuzzing Variable 0:1299	Fuzzing Variable 0:1754
Fuzzing Variable 0:81	Variablesize= 32767	Fuzzing Variable 0:536	Fuzzing Variable 0:845	Fuzzing Variable 0:1300	Fuzzing Variable 0:1755
Variablesize= 32764	Fuzzing Variable 0:309	Variablesize= 65534	Fuzzing Variable 0:846	Fuzzing Variable 0:1301	Fuzzing Variable 0:1756
Fuzzing Variable 0:82	Variablesize= 32766	Fuzzing Variable 0:537	Fuzzing Variable 0:847	Fuzzing Variable 0:1302	Fuzzing Variable 0:1757
Variablesize= 32763	Fuzzing Variable 0:310	Variablesize= 32768	Fuzzing Variable 0:848	Fuzzing Variable 0:1303	Fuzzing Variable 0:1758
Fuzzing Variable 0:83	Variablesize= 32765	Fuzzing Variable 0:538	Fuzzing Variable 0:849	Fuzzing Variable 0:1304	Fuzzing Variable 0:1759
Variablesize= 32762	Fuzzing Variable 0:311	Variablesize= 32767	Fuzzing Variable 0:850	Fuzzing Variable 0:1305	Fuzzing Variable 0:1760
Fuzzing Variable 0:84	Variablesize= 32764	Fuzzing Variable 0:539	Fuzzing Variable 0:851	Fuzzing Variable 0:1306	Fuzzing Variable 0:1761
Variablesize= 20000	Fuzzing Variable 0:312	Variablesize= 32766	Fuzzing Variable 0:852	Fuzzing Variable 0:1307	Fuzzing Variable 0:1762
Fuzzing Variable 0:85	Variablesize= 32763	Fuzzing Variable 0:540	Fuzzing Variable 0:853	Fuzzing Variable 0:1308	Fuzzing Variable 0:1763
Variablesize= 10000	Fuzzing Variable 0:313	Variablesize= 32765	Fuzzing Variable 0:854	Fuzzing Variable 0:1309	Fuzzing Variable 0:1764
Fuzzing Variable 0:86	Variablesize= 32762	Fuzzing Variable 0:541	Fuzzing Variable 0:855	Fuzzing Variable 0:1310	Fuzzing Variable 0:1765
Variablesize= 5000	Fuzzing Variable 0:314	Variablesize= 32764	Fuzzing Variable 0:856	Fuzzing Variable 0:1311	Fuzzing Variable 0:1766
Fuzzing Variable 0:87	Variablesize= 20000	Fuzzing Variable 0:542	Fuzzing Variable 0:857	Fuzzing Variable 0:1312	Fuzzing Variable 0:1767
Variablesize= 4097	Fuzzing Variable 0:315	Variablesize= 32763	Fuzzing Variable 0:858	Fuzzing Variable 0:1313	Fuzzing Variable 0:1768
Fuzzing Variable 0:88	Variablesize= 10000	Fuzzing Variable 0:543	Fuzzing Variable 0:859	Fuzzing Variable 0:1314	Fuzzing Variable 0:1769
Variablesize= 4096	Fuzzing Variable 0:316	Variablesize= 32762	Fuzzing Variable 0:860	Fuzzing Variable 0:1315	Fuzzing Variable 0:1770
Fuzzing Variable 0:89	Variablesize= 5000	Fuzzing Variable 0:544	Fuzzing Variable 0:861	Fuzzing Variable 0:1316	Fuzzing Variable 0:1771
Variablesize= 4095	Fuzzing Variable 0:317	Variablesize= 20000	Fuzzing Variable 0:862	Fuzzing Variable 0:1317	Fuzzing Variable 0:1772
Fuzzing Variable 0:90	Variablesize= 4097	Fuzzing Variable 0:545	Fuzzing Variable 0:863	Fuzzing Variable 0:1318	Fuzzing Variable 0:1773
Variablesize= 2048	Fuzzing Variable 0:318	Variablesize= 10000	Fuzzing Variable 0:864	Fuzzing Variable 0:1319	Fuzzing Variable 0:1774
Fuzzing Variable 0:91	Variablesize= 4096	Fuzzing Variable 0:546	Fuzzing Variable 0:865	Fuzzing Variable 0:1320	Fuzzing Variable 0:1775
Variablesize= 1024	Fuzzing Variable 0:319	Variablesize= 5000	Fuzzing Variable 0:866	Fuzzing Variable 0:1321	Fuzzing Variable 0:1776
Fuzzing Variable 0:92	Variablesize= 4095	Fuzzing Variable 0:547	Fuzzing Variable 0:867	Fuzzing Variable 0:1322	Fuzzing Variable 0:1777
Variablesize= 1023	Fuzzing Variable 0:320	Variablesize= 4097	Fuzzing Variable 0:868	Fuzzing Variable 0:1323	Fuzzing Variable 0:1778
Fuzzing Variable 0:93	Variablesize= 2048	Fuzzing Variable 0:548	Fuzzing Variable 0:869	Fuzzing Variable 0:1324	Fuzzing Variable 0:1779
Variablesize= 512	Fuzzing Variable 0:321	Variablesize= 4096	Fuzzing Variable 0:870	Fuzzing Variable 0:1325	Fuzzing Variable 0:1780
Fuzzing Variable 0:94	Variablesize= 1024	Fuzzing Variable 0:549	Fuzzing Variable 0:871	Fuzzing Variable 0:1326	Fuzzing Variable 0:1781

Variablesize= 420	Fuzzing Variable 0:322	Variablesize= 4095	Fuzzing Variable 0:872	Fuzzing Variable 0:1327	Fuzzing Variable 0:1782
Fuzzing Variable 0:95	Variablesize= 1023	Fuzzing Variable 0:550	Fuzzing Variable 0:873	Fuzzing Variable 0:1328	Fuzzing Variable 0:1783
Variablesize= 257	Fuzzing Variable 0:323	Variablesize= 2048	Fuzzing Variable 0:874	Fuzzing Variable 0:1329	Fuzzing Variable 0:1784
Fuzzing Variable 0:96	Variablesize= 512	Fuzzing Variable 0:551	Fuzzing Variable 0:875	Fuzzing Variable 0:1330	Fuzzing Variable 0:1785
Variablesize= 256	Fuzzing Variable 0:324	Variablesize= 1024	Fuzzing Variable 0:876	Fuzzing Variable 0:1331	Fuzzing Variable 0:1786
Fuzzing Variable 0:97	Variablesize= 420	Fuzzing Variable 0:552	Fuzzing Variable 0:877	Fuzzing Variable 0:1332	Fuzzing Variable 0:1787
Variablesize= 240	Fuzzing Variable 0:325	Variablesize= 1023	Fuzzing Variable 0:878	Fuzzing Variable 0:1333	Fuzzing Variable 0:1788
Fuzzing Variable 0:98	Variablesize= 257	Fuzzing Variable 0:553	Fuzzing Variable 0:879	Fuzzing Variable 0:1334	Fuzzing Variable 0:1789
Variablesize= 128	Fuzzing Variable 0:326	Variablesize= 512	Fuzzing Variable 0:880	Fuzzing Variable 0:1335	Fuzzing Variable 0:1790
Fuzzing Variable 0:99	Variablesize= 256	Fuzzing Variable 0:554	Fuzzing Variable 0:881	Fuzzing Variable 0:1336	Fuzzing Variable 0:1791
Variablesize= 65534	Fuzzing Variable 0:327	Variablesize= 420	Fuzzing Variable 0:882	Fuzzing Variable 0:1337	Fuzzing Variable 0:1792
Fuzzing Variable 0:100	Variablesize= 240	Fuzzing Variable 0:555	Fuzzing Variable 0:883	Fuzzing Variable 0:1338	Fuzzing Variable 0:1793
Variablesize= 32768	Fuzzing Variable 0:328	Variablesize= 257	Fuzzing Variable 0:884	Fuzzing Variable 0:1339	Fuzzing Variable 0:1794
Fuzzing Variable 0:101	Variablesize= 128	Fuzzing Variable 0:556	Fuzzing Variable 0:885	Fuzzing Variable 0:1340	Fuzzing Variable 0:1795
Variablesize= 32767	Fuzzing Variable 0:329	Variablesize= 256	Fuzzing Variable 0:886	Fuzzing Variable 0:1341	Fuzzing Variable 0:1796
Fuzzing Variable 0:102	Variablesize= 65534	Fuzzing Variable 0:557	Fuzzing Variable 0:887	Fuzzing Variable 0:1342	Fuzzing Variable 0:1797
Variablesize= 32766	Fuzzing Variable 0:330	Variablesize= 240	Fuzzing Variable 0:888	Fuzzing Variable 0:1343	Fuzzing Variable 0:1798
Fuzzing Variable 0:103	Variablesize= 32768	Fuzzing Variable 0:558	Fuzzing Variable 0:889	Fuzzing Variable 0:1344	Fuzzing Variable 0:1799
Variablesize= 32765	Fuzzing Variable 0:331	Variablesize= 128	Fuzzing Variable 0:890	Fuzzing Variable 0:1345	Fuzzing Variable 0:1800
Fuzzing Variable 0:104	Variablesize= 32767	Fuzzing Variable 0:559	Fuzzing Variable 0:891	Fuzzing Variable 0:1346	Fuzzing Variable 0:1801
Variablesize= 32764	Fuzzing Variable 0:332	Variablesize= 65534	Fuzzing Variable 0:892	Fuzzing Variable 0:1347	Fuzzing Variable 0:1802
Fuzzing Variable 0:105	Variablesize= 32766	Fuzzing Variable 0:560	Fuzzing Variable 0:893	Fuzzing Variable 0:1348	Fuzzing Variable 0:1803
Variablesize= 32763	Fuzzing Variable 0:333	Variablesize= 32768	Fuzzing Variable 0:894	Fuzzing Variable 0:1349	Fuzzing Variable 0:1804
Fuzzing Variable 0:106	Variablesize= 32765	Fuzzing Variable 0:561	Fuzzing Variable 0:895	Fuzzing Variable 0:1350	Fuzzing Variable 0:1805
Variablesize= 32762	Fuzzing Variable 0:334	Variablesize= 32767	Fuzzing Variable 0:896	Fuzzing Variable 0:1351	Fuzzing Variable 0:1806
Fuzzing Variable 0:107	Variablesize= 32764	Fuzzing Variable 0:562	Fuzzing Variable 0:897	Fuzzing Variable 0:1352	Fuzzing Variable 0:1807
Variablesize= 20000	Fuzzing Variable 0:335	Variablesize= 32766	Fuzzing Variable 0:898	Fuzzing Variable 0:1353	Fuzzing Variable 0:1808
Fuzzing Variable 0:108	Variablesize= 32763	Fuzzing Variable 0:563	Fuzzing Variable 0:899	Fuzzing Variable 0:1354	Fuzzing Variable 0:1809
Variablesize= 10000	Fuzzing Variable 0:336	Variablesize= 32765	Fuzzing Variable 0:900	Fuzzing Variable 0:1355	Fuzzing Variable 0:1810
Fuzzing Variable 0:109	Variablesize= 32762	Fuzzing Variable 0:564	Fuzzing Variable 0:901	Fuzzing Variable 0:1356	Fuzzing Variable 0:1811
Variablesize= 5000	Fuzzing Variable 0:337	Variablesize= 32764	Fuzzing Variable 0:902	Fuzzing Variable 0:1357	Fuzzing Variable 0:1812
Fuzzing Variable 0:110	Variablesize= 20000	Fuzzing Variable 0:565	Fuzzing Variable 0:903	Fuzzing Variable 0:1358	Fuzzing Variable 0:1813
Variablesize= 4097	Fuzzing Variable 0:338	Variablesize= 32763	Fuzzing Variable 0:904	Fuzzing Variable 0:1359	Fuzzing Variable 0:1814
Fuzzing Variable 0:111	Variablesize= 10000	Fuzzing Variable 0:566	Fuzzing Variable 0:905	Fuzzing Variable 0:1360	Fuzzing Variable 0:1815
Variablesize= 4096	Fuzzing Variable 0:339	Variablesize= 32762	Fuzzing Variable 0:906	Fuzzing Variable 0:1361	Fuzzing Variable 0:1816
Fuzzing Variable 0:112	Variablesize= 5000	Fuzzing Variable 0:567	Fuzzing Variable 0:907	Fuzzing Variable 0:1362	Fuzzing Variable 0:1817
Variablesize= 4095	Fuzzing Variable 0:340	Variablesize= 20000	Fuzzing Variable 0:908	Fuzzing Variable 0:1363	Fuzzing Variable 0:1818
Fuzzing Variable 0:113	Variablesize= 4097	Fuzzing Variable 0:568	Fuzzing Variable 0:909	Fuzzing Variable 0:1364	Fuzzing Variable 0:1819
Variablesize= 2048	Fuzzing Variable 0:341	Variablesize= 10000	Fuzzing Variable 0:910	Fuzzing Variable 0:1365	Fuzzing Variable 0:1820

Fuzzing Variable 0:114	Variablesize= 4096	Fuzzing Variable 0:569	Fuzzing Variable 0:911	Fuzzing Variable 0:1366	Fuzzing Variable 0:1821
Variablesize= 1024	Fuzzing Variable 0:342	Variablesize= 5000	Fuzzing Variable 0:912	Fuzzing Variable 0:1367	Fuzzing Variable 0:1822
Fuzzing Variable 0:115	Variablesize= 4095	Fuzzing Variable 0:570	Fuzzing Variable 0:913	Fuzzing Variable 0:1368	Fuzzing Variable 0:1823
Variablesize= 1023	Fuzzing Variable 0:343	Variablesize= 4097	Fuzzing Variable 0:914	Fuzzing Variable 0:1369	Fuzzing Variable 0:1824
Fuzzing Variable 0:116	Variablesize= 2048	Fuzzing Variable 0:571	Fuzzing Variable 0:915	Fuzzing Variable 0:1370	Fuzzing Variable 0:1825
Variablesize= 512	Fuzzing Variable 0:344	Variablesize= 4096	Fuzzing Variable 0:916	Fuzzing Variable 0:1371	Fuzzing Variable 0:1826
Fuzzing Variable 0:117	Variablesize= 1024	Fuzzing Variable 0:572	Fuzzing Variable 0:917	Fuzzing Variable 0:1372	Fuzzing Variable 0:1827
Variablesize= 420	Fuzzing Variable 0:345	Variablesize= 4095	Fuzzing Variable 0:918	Fuzzing Variable 0:1373	Fuzzing Variable 0:1828
Fuzzing Variable 0:118	Variablesize= 1023	Fuzzing Variable 0:573	Fuzzing Variable 0:919	Fuzzing Variable 0:1374	Fuzzing Variable 0:1829
Variablesize= 257	Fuzzing Variable 0:346	Variablesize= 2048	Fuzzing Variable 0:920	Fuzzing Variable 0:1375	Fuzzing Variable 0:1830
Fuzzing Variable 0:119	Variablesize= 512	Fuzzing Variable 0:574	Fuzzing Variable 0:921	Fuzzing Variable 0:1376	Fuzzing Variable 0:1831
Variablesize= 256	Fuzzing Variable 0:347	Variablesize= 1024	Fuzzing Variable 0:922	Fuzzing Variable 0:1377	Fuzzing Variable 0:1832
Fuzzing Variable 0:120	Variablesize= 420	Fuzzing Variable 0:575	Fuzzing Variable 0:923	Fuzzing Variable 0:1378	Fuzzing Variable 0:1833
Variablesize= 240	Fuzzing Variable 0:348	Variablesize= 1023	Fuzzing Variable 0:924	Fuzzing Variable 0:1379	Fuzzing Variable 0:1834
Fuzzing Variable 0:121	Variablesize= 257	Fuzzing Variable 0:576	Fuzzing Variable 0:925	Fuzzing Variable 0:1380	Fuzzing Variable 0:1835
Variablesize= 128	Fuzzing Variable 0:349	Variablesize= 512	Fuzzing Variable 0:926	Fuzzing Variable 0:1381	Fuzzing Variable 0:1836
Fuzzing Variable 0:122	Variablesize= 256	Fuzzing Variable 0:577	Fuzzing Variable 0:927	Fuzzing Variable 0:1382	Fuzzing Variable 0:1837
Variablesize= 65534	Fuzzing Variable 0:350	Variablesize= 420	Fuzzing Variable 0:928	Fuzzing Variable 0:1383	Fuzzing Variable 0:1838
Fuzzing Variable 0:123	Variablesize= 240	Fuzzing Variable 0:578	Fuzzing Variable 0:929	Fuzzing Variable 0:1384	Fuzzing Variable 0:1839
Variablesize= 32768	Fuzzing Variable 0:351	Variablesize= 257	Fuzzing Variable 0:930	Fuzzing Variable 0:1385	Fuzzing Variable 0:1840
Fuzzing Variable 0:124	Variablesize= 128	Fuzzing Variable 0:579	Fuzzing Variable 0:931	Fuzzing Variable 0:1386	Fuzzing Variable 0:1841
Variablesize= 32767	Fuzzing Variable 0:352	Variablesize= 256	Fuzzing Variable 0:932	Fuzzing Variable 0:1387	Fuzzing Variable 0:1842
Fuzzing Variable 0:125	Variablesize= 65534	Fuzzing Variable 0:580	Fuzzing Variable 0:933	Fuzzing Variable 0:1388	Fuzzing Variable 0:1843
Variablesize= 32766	Fuzzing Variable 0:353	Variablesize= 240	Fuzzing Variable 0:934	Fuzzing Variable 0:1389	Fuzzing Variable 0:1844
Fuzzing Variable 0:126	Variablesize= 32768	Fuzzing Variable 0:581	Fuzzing Variable 0:935	Fuzzing Variable 0:1390	Fuzzing Variable 0:1845
Variablesize= 32765	Fuzzing Variable 0:354	Variablesize= 128	Fuzzing Variable 0:936	Fuzzing Variable 0:1391	Fuzzing Variable 0:1846
Fuzzing Variable 0:127	Variablesize= 32767	Fuzzing Variable 0:582	Fuzzing Variable 0:937	Fuzzing Variable 0:1392	Fuzzing Variable 0:1847
Variablesize= 32764	Fuzzing Variable 0:355	Variablesize= 65534	Fuzzing Variable 0:938	Fuzzing Variable 0:1393	Fuzzing Variable 0:1848
Fuzzing Variable 0:128	Variablesize= 32766	Fuzzing Variable 0:583	Fuzzing Variable 0:939	Fuzzing Variable 0:1394	Fuzzing Variable 0:1849
Variablesize= 32763	Fuzzing Variable 0:356	Variablesize= 32768	Fuzzing Variable 0:940	Fuzzing Variable 0:1395	Fuzzing Variable 0:1850
Fuzzing Variable 0:129	Variablesize= 32765	Fuzzing Variable 0:584	Fuzzing Variable 0:941	Fuzzing Variable 0:1396	Fuzzing Variable 0:1851
Variablesize= 32762	Fuzzing Variable 0:357	Variablesize= 32767	Fuzzing Variable 0:942	Fuzzing Variable 0:1397	Fuzzing Variable 0:1852
Fuzzing Variable 0:130	Variablesize= 32764	Fuzzing Variable 0:585	Fuzzing Variable 0:943	Fuzzing Variable 0:1398	Fuzzing Variable 0:1853
Variablesize= 20000	Fuzzing Variable 0:358	Variablesize= 32766	Fuzzing Variable 0:944	Fuzzing Variable 0:1399	Fuzzing Variable 0:1854
Fuzzing Variable 0:131	Variablesize= 32763	Fuzzing Variable 0:586	Fuzzing Variable 0:945	Fuzzing Variable 0:1400	Fuzzing Variable 0:1855
Variablesize= 10000	Fuzzing Variable 0:359	Variablesize= 32765	Fuzzing Variable 0:946	Fuzzing Variable 0:1401	Fuzzing Variable 0:1856
Fuzzing Variable 0:132	Variablesize= 32762	Fuzzing Variable 0:587	Fuzzing Variable 0:947	Fuzzing Variable 0:1402	Fuzzing Variable 0:1857
Variablesize= 5000	Fuzzing Variable 0:360	Variablesize= 32764	Fuzzing Variable 0:948	Fuzzing Variable 0:1403	Fuzzing Variable 0:1858
Fuzzing Variable 0:133	Variablesize= 20000	Fuzzing Variable 0:588	Fuzzing Variable 0:949	Fuzzing Variable 0:1404	Fuzzing Variable 0:1859

Variablesize= 4097	Fuzzing Variable 0:361	Variablesize= 32763	Fuzzing Variable 0:950	Fuzzing Variable 0:1405	Fuzzing Variable 0:1860
Fuzzing Variable 0:134	Variablesize= 10000	Fuzzing Variable 0:589	Fuzzing Variable 0:951	Fuzzing Variable 0:1406	Fuzzing Variable 0:1861
Variablesize= 4096	Fuzzing Variable 0:362	Variablesize= 32762	Fuzzing Variable 0:952	Fuzzing Variable 0:1407	Fuzzing Variable 0:1862
Fuzzing Variable 0:135	Variablesize= 5000	Fuzzing Variable 0:590	Fuzzing Variable 0:953	Fuzzing Variable 0:1408	Fuzzing Variable 0:1863
Variablesize= 4095	Fuzzing Variable 0:363	Variablesize= 20000	Fuzzing Variable 0:954	Fuzzing Variable 0:1409	Fuzzing Variable 0:1864
Fuzzing Variable 0:136	Variablesize= 4097	Fuzzing Variable 0:591	Fuzzing Variable 0:955	Fuzzing Variable 0:1410	Fuzzing Variable 0:1865
Variablesize= 2048	Fuzzing Variable 0:364	Variablesize= 10000	Fuzzing Variable 0:956	Fuzzing Variable 0:1411	Fuzzing Variable 0:1866
Fuzzing Variable 0:137	Variablesize= 4096	Fuzzing Variable 0:592	Fuzzing Variable 0:957	Fuzzing Variable 0:1412	Fuzzing Variable 0:1867
Variablesize= 1024	Fuzzing Variable 0:365	Variablesize= 5000	Fuzzing Variable 0:958	Fuzzing Variable 0:1413	Fuzzing Variable 0:1868
Fuzzing Variable 0:138	Variablesize= 4095	Fuzzing Variable 0:593	Fuzzing Variable 0:959	Fuzzing Variable 0:1414	Fuzzing Variable 0:1869
Variablesize= 1023	Fuzzing Variable 0:366	Variablesize= 4097	Fuzzing Variable 0:960	Fuzzing Variable 0:1415	Fuzzing Variable 0:1870
Fuzzing Variable 0:139	Variablesize= 2048	Fuzzing Variable 0:594	Fuzzing Variable 0:961	Fuzzing Variable 0:1416	Fuzzing Variable 0:1871
Variablesize= 512	Fuzzing Variable 0:367	Variablesize= 4096	Fuzzing Variable 0:962	Fuzzing Variable 0:1417	Fuzzing Variable 0:1872
Fuzzing Variable 0:140	Variablesize= 1024	Fuzzing Variable 0:595	Fuzzing Variable 0:963	Fuzzing Variable 0:1418	Fuzzing Variable 0:1873
Variablesize= 420	Fuzzing Variable 0:368	Variablesize= 4095	Fuzzing Variable 0:964	Fuzzing Variable 0:1419	Fuzzing Variable 0:1874
Fuzzing Variable 0:141	Variablesize= 1023	Fuzzing Variable 0:596	Fuzzing Variable 0:965	Fuzzing Variable 0:1420	Fuzzing Variable 0:1875
Variablesize= 257	Fuzzing Variable 0:369	Variablesize= 2048	Fuzzing Variable 0:966	Fuzzing Variable 0:1421	Fuzzing Variable 0:1876
Fuzzing Variable 0:142	Variablesize= 512	Fuzzing Variable 0:597	Fuzzing Variable 0:967	Fuzzing Variable 0:1422	Fuzzing Variable 0:1877
Variablesize= 256	Fuzzing Variable 0:370	Variablesize= 1024	Fuzzing Variable 0:968	Fuzzing Variable 0:1423	Fuzzing Variable 0:1878
Fuzzing Variable 0:143	Variablesize= 420	Fuzzing Variable 0:598	Fuzzing Variable 0:969	Fuzzing Variable 0:1424	Fuzzing Variable 0:1879
Variablesize= 240	Fuzzing Variable 0:371	Variablesize= 1023	Fuzzing Variable 0:970	Fuzzing Variable 0:1425	Fuzzing Variable 0:1880
Fuzzing Variable 0:144	Variablesize= 257	Fuzzing Variable 0:599	Fuzzing Variable 0:971	Fuzzing Variable 0:1426	Fuzzing Variable 0:1881
Variablesize= 128	Fuzzing Variable 0:372	Variablesize= 512	Fuzzing Variable 0:972	Fuzzing Variable 0:1427	Fuzzing Variable 0:1882
Fuzzing Variable 0:145	Variablesize= 256	Fuzzing Variable 0:600	Fuzzing Variable 0:973	Fuzzing Variable 0:1428	Fuzzing Variable 0:1883
Variablesize= 65534	Fuzzing Variable 0:373	Variablesize= 420	Fuzzing Variable 0:974	Fuzzing Variable 0:1429	Fuzzing Variable 0:1884
Fuzzing Variable 0:146	Variablesize= 240	Fuzzing Variable 0:601	Fuzzing Variable 0:975	Fuzzing Variable 0:1430	Fuzzing Variable 0:1885
Variablesize= 32768	Fuzzing Variable 0:374	Variablesize= 257	Fuzzing Variable 0:976	Fuzzing Variable 0:1431	Fuzzing Variable 0:1886
Fuzzing Variable 0:147	Variablesize= 128	Fuzzing Variable 0:602	Fuzzing Variable 0:977	Fuzzing Variable 0:1432	Fuzzing Variable 0:1887
Variablesize= 32767	Fuzzing Variable 0:375	Variablesize= 256	Fuzzing Variable 0:978	Fuzzing Variable 0:1433	Fuzzing Variable 0:1888
Fuzzing Variable 0:148	Variablesize= 65534	Fuzzing Variable 0:603	Fuzzing Variable 0:979	Fuzzing Variable 0:1434	Fuzzing Variable 0:1889
Variablesize= 32766	Fuzzing Variable 0:376	Variablesize= 240	Fuzzing Variable 0:980	Fuzzing Variable 0:1435	Fuzzing Variable 0:1890
Fuzzing Variable 0:149	Variablesize= 32768	Fuzzing Variable 0:604	Fuzzing Variable 0:981	Fuzzing Variable 0:1436	Fuzzing Variable 0:1891
Variablesize= 32765	Fuzzing Variable 0:377	Variablesize= 128	Fuzzing Variable 0:982	Fuzzing Variable 0:1437	Fuzzing Variable 0:1892
Fuzzing Variable 0:150	Variablesize= 32767	Fuzzing Variable 0:605	Fuzzing Variable 0:983	Fuzzing Variable 0:1438	Fuzzing Variable 0:1893
Variablesize= 32764	Fuzzing Variable 0:378	Variablesize= 65534	Fuzzing Variable 0:984	Fuzzing Variable 0:1439	Fuzzing Variable 0:1894
Fuzzing Variable 0:151	Variablesize= 32766	Fuzzing Variable 0:606	Fuzzing Variable 0:985	Fuzzing Variable 0:1440	Fuzzing Variable 0:1895
Variablesize= 32763	Fuzzing Variable 0:379	Variablesize= 32768	Fuzzing Variable 0:986	Fuzzing Variable 0:1441	Fuzzing Variable 0:1896
Fuzzing Variable 0:152	Variablesize= 32765	Fuzzing Variable 0:607	Fuzzing Variable 0:987	Fuzzing Variable 0:1442	Fuzzing Variable 0:1897
Variablesize= 32762	Fuzzing Variable 0:380	Variablesize= 32767	Fuzzing Variable 0:988	Fuzzing Variable 0:1443	Fuzzing Variable 0:1898

Fuzzing Variable 0:153	Variablesize= 32764	Fuzzing Variable 0:608	Fuzzing Variable 0:989	Fuzzing Variable 0:1444	Fuzzing Variable 0:1899
Variablesize= 20000	Fuzzing Variable 0:381	Variablesize= 32766	Fuzzing Variable 0:990	Fuzzing Variable 0:1445	Fuzzing Variable 0:1900
Fuzzing Variable 0:154	Variablesize= 32763	Fuzzing Variable 0:609	Fuzzing Variable 0:991	Fuzzing Variable 0:1446	Fuzzing Variable 0:1901
Variablesize= 10000	Fuzzing Variable 0:382	Variablesize= 32765	Fuzzing Variable 0:992	Fuzzing Variable 0:1447	Fuzzing Variable 0:1902
Fuzzing Variable 0:155	Variablesize= 32762	Fuzzing Variable 0:610	Fuzzing Variable 0:993	Fuzzing Variable 0:1448	Fuzzing Variable 0:1903
Variablesize= 5000	Fuzzing Variable 0:383	Variablesize= 32764	Fuzzing Variable 0:994	Fuzzing Variable 0:1449	Fuzzing Variable 0:1904
Fuzzing Variable 0:156	Variablesize= 20000	Fuzzing Variable 0:611	Fuzzing Variable 0:995	Fuzzing Variable 0:1450	Fuzzing Variable 0:1905
Variablesize= 4097	Fuzzing Variable 0:384	Variablesize= 32763	Fuzzing Variable 0:996	Fuzzing Variable 0:1451	Fuzzing Variable 0:1906
Fuzzing Variable 0:157	Variablesize= 10000	Fuzzing Variable 0:612	Fuzzing Variable 0:997	Fuzzing Variable 0:1452	Fuzzing Variable 0:1907
Variablesize= 4096	Fuzzing Variable 0:385	Variablesize= 32762	Fuzzing Variable 0:998	Fuzzing Variable 0:1453	Fuzzing Variable 0:1908
Fuzzing Variable 0:158	Variablesize= 5000	Fuzzing Variable 0:613	Fuzzing Variable 0:999	Fuzzing Variable 0:1454	Fuzzing Variable 0:1909
Variablesize= 4095	Fuzzing Variable 0:386	Variablesize= 20000	Fuzzing Variable 0:1000	Fuzzing Variable 0:1455	Fuzzing Variable 0:1910
Fuzzing Variable 0:159	Variablesize= 4097	Fuzzing Variable 0:614	Fuzzing Variable 0:1001	Fuzzing Variable 0:1456	Fuzzing Variable 0:1911
Variablesize= 2048	Fuzzing Variable 0:387	Variablesize= 10000	Fuzzing Variable 0:1002	Fuzzing Variable 0:1457	Fuzzing Variable 0:1912
Fuzzing Variable 0:160	Variablesize= 4096	Fuzzing Variable 0:615	Fuzzing Variable 0:1003	Fuzzing Variable 0:1458	Fuzzing Variable 0:1913
Variablesize= 1024	Fuzzing Variable 0:388	Variablesize= 5000	Fuzzing Variable 0:1004	Fuzzing Variable 0:1459	Fuzzing Variable 0:1914
Fuzzing Variable 0:161	Variablesize= 4095	Fuzzing Variable 0:616	Fuzzing Variable 0:1005	Fuzzing Variable 0:1460	Fuzzing Variable 0:1915
Variablesize= 1023	Fuzzing Variable 0:389	Variablesize= 4097	Fuzzing Variable 0:1006	Fuzzing Variable 0:1461	Fuzzing Variable 0:1916
Fuzzing Variable 0:162	Variablesize= 2048	Fuzzing Variable 0:617	Fuzzing Variable 0:1007	Fuzzing Variable 0:1462	Fuzzing Variable 0:1917
Variablesize= 512	Fuzzing Variable 0:390	Variablesize= 4096	Fuzzing Variable 0:1008	Fuzzing Variable 0:1463	Fuzzing Variable 0:1918
Fuzzing Variable 0:163	Variablesize= 1024	Fuzzing Variable 0:618	Fuzzing Variable 0:1009	Fuzzing Variable 0:1464	Fuzzing Variable 0:1919
Variablesize= 420	Fuzzing Variable 0:391	Variablesize= 4095	Fuzzing Variable 0:1010	Fuzzing Variable 0:1465	Fuzzing Variable 0:1920
Fuzzing Variable 0:164	Variablesize= 1023	Fuzzing Variable 0:619	Fuzzing Variable 0:1011	Fuzzing Variable 0:1466	Fuzzing Variable 0:1921
Variablesize= 257	Fuzzing Variable 0:392	Variablesize= 2048	Fuzzing Variable 0:1012	Fuzzing Variable 0:1467	Fuzzing Variable 0:1922
Fuzzing Variable 0:165	Variablesize= 512	Fuzzing Variable 0:620	Fuzzing Variable 0:1013	Fuzzing Variable 0:1468	Fuzzing Variable 0:1923
Variablesize= 256	Fuzzing Variable 0:393	Variablesize= 1024	Fuzzing Variable 0:1014	Fuzzing Variable 0:1469	Fuzzing Variable 0:1924
Fuzzing Variable 0:166	Variablesize= 420	Fuzzing Variable 0:621	Fuzzing Variable 0:1015	Fuzzing Variable 0:1470	Fuzzing Variable 0:1925
Variablesize= 240	Fuzzing Variable 0:394	Variablesize= 1023	Fuzzing Variable 0:1016	Fuzzing Variable 0:1471	Fuzzing Variable 0:1926
Fuzzing Variable 0:167	Variablesize= 257	Fuzzing Variable 0:622	Fuzzing Variable 0:1017	Fuzzing Variable 0:1472	Fuzzing Variable 0:1927
Variablesize= 128	Fuzzing Variable 0:395	Variablesize= 512	Fuzzing Variable 0:1018	Fuzzing Variable 0:1473	Fuzzing Variable 0:1928
Fuzzing Variable 0:168	Variablesize= 256	Fuzzing Variable 0:623	Fuzzing Variable 0:1019	Fuzzing Variable 0:1474	Fuzzing Variable 0:1929
Variablesize= 65534	Fuzzing Variable 0:396	Variablesize= 420	Fuzzing Variable 0:1020	Fuzzing Variable 0:1475	Fuzzing Variable 0:1930
Fuzzing Variable 0:169	Variablesize= 240	Fuzzing Variable 0:624	Fuzzing Variable 0:1021	Fuzzing Variable 0:1476	Fuzzing Variable 0:1931
Variablesize= 32768	Fuzzing Variable 0:397	Variablesize= 257	Fuzzing Variable 0:1022	Fuzzing Variable 0:1477	Fuzzing Variable 0:1932
Fuzzing Variable 0:170	Variablesize= 128	Fuzzing Variable 0:625	Fuzzing Variable 0:1023	Fuzzing Variable 0:1478	Fuzzing Variable 0:1933
Variablesize= 32767	Fuzzing Variable 0:398	Variablesize= 256	Fuzzing Variable 0:1024	Fuzzing Variable 0:1479	Fuzzing Variable 0:1934
Fuzzing Variable 0:171	Variablesize= 65534	Fuzzing Variable 0:626	Fuzzing Variable 0:1025	Fuzzing Variable 0:1480	Fuzzing Variable 0:1935
Variablesize= 32766	Fuzzing Variable 0:399	Variablesize= 240	Fuzzing Variable 0:1026	Fuzzing Variable 0:1481	Fuzzing Variable 0:1936
Fuzzing Variable 0:172	Variablesize= 32768	Fuzzing Variable 0:627	Fuzzing Variable 0:1027	Fuzzing Variable 0:1482	Fuzzing Variable 0:1937

Variablesize= 32765	Fuzzing Variable 0:400	Variablesize= 128	Fuzzing Variable 0:1028	Fuzzing Variable 0:1483	Fuzzing Variable 0:1938
Fuzzing Variable 0:173	Variablesize= 32767	Fuzzing Variable 0:628	Fuzzing Variable 0:1029	Fuzzing Variable 0:1484	Fuzzing Variable 0:1939
Variablesize= 32764	Fuzzing Variable 0:401	Variablesize= 65534	Fuzzing Variable 0:1030	Fuzzing Variable 0:1485	Fuzzing Variable 0:1940
Fuzzing Variable 0:174	Variablesize= 32766	Fuzzing Variable 0:629	Fuzzing Variable 0:1031	Fuzzing Variable 0:1486	Fuzzing Variable 0:1941
Variablesize= 32763	Fuzzing Variable 0:402	Variablesize= 32768	Fuzzing Variable 0:1032	Fuzzing Variable 0:1487	Fuzzing Variable 0:1942
Fuzzing Variable 0:175	Variablesize= 32765	Fuzzing Variable 0:630	Fuzzing Variable 0:1033	Fuzzing Variable 0:1488	Fuzzing Variable 0:1943
Variablesize= 32762	Fuzzing Variable 0:403	Variablesize= 32767	Fuzzing Variable 0:1034	Fuzzing Variable 0:1489	Fuzzing Variable 0:1944
Fuzzing Variable 0:176	Variablesize= 32764	Fuzzing Variable 0:631	Fuzzing Variable 0:1035	Fuzzing Variable 0:1490	Fuzzing Variable 0:1945
Variablesize= 20000	Fuzzing Variable 0:404	Variablesize= 32766	Fuzzing Variable 0:1036	Fuzzing Variable 0:1491	Fuzzing Variable 0:1946
Fuzzing Variable 0:177	Variablesize= 32763	Fuzzing Variable 0:632	Fuzzing Variable 0:1037	Fuzzing Variable 0:1492	Fuzzing Variable 0:1947
Variablesize= 10000	Fuzzing Variable 0:405	Variablesize= 32765	Fuzzing Variable 0:1038	Fuzzing Variable 0:1493	Fuzzing Variable 0:1948
Fuzzing Variable 0:178	Variablesize= 32762	Fuzzing Variable 0:633	Fuzzing Variable 0:1039	Fuzzing Variable 0:1494	Fuzzing Variable 0:1949
Variablesize= 5000	Fuzzing Variable 0:406	Variablesize= 32764	Fuzzing Variable 0:1040	Fuzzing Variable 0:1495	Fuzzing Variable 0:1950
Fuzzing Variable 0:179	Variablesize= 20000	Fuzzing Variable 0:634	Fuzzing Variable 0:1041	Fuzzing Variable 0:1496	Fuzzing Variable 0:1951
Variablesize= 4097	Fuzzing Variable 0:407	Variablesize= 32763	Fuzzing Variable 0:1042	Fuzzing Variable 0:1497	Fuzzing Variable 0:1952
Fuzzing Variable 0:180	Variablesize= 10000	Fuzzing Variable 0:635	Fuzzing Variable 0:1043	Fuzzing Variable 0:1498	Fuzzing Variable 0:1953
Variablesize= 4096	Fuzzing Variable 0:408	Variablesize= 32762	Fuzzing Variable 0:1044	Fuzzing Variable 0:1499	Fuzzing Variable 0:1954
Fuzzing Variable 0:181	Variablesize= 5000	Fuzzing Variable 0:636	Fuzzing Variable 0:1045	Fuzzing Variable 0:1500	Fuzzing Variable 0:1955
Variablesize= 4095	Fuzzing Variable 0:409	Variablesize= 20000	Fuzzing Variable 0:1046	Fuzzing Variable 0:1501	Fuzzing Variable 0:1956
Fuzzing Variable 0:182	Variablesize= 4097	Fuzzing Variable 0:637	Fuzzing Variable 0:1047	Fuzzing Variable 0:1502	Fuzzing Variable 0:1957
Variablesize= 2048	Fuzzing Variable 0:410	Variablesize= 10000	Fuzzing Variable 0:1048	Fuzzing Variable 0:1503	Fuzzing Variable 0:1958
Fuzzing Variable 0:183	Variablesize= 4096	Fuzzing Variable 0:638	Fuzzing Variable 0:1049	Fuzzing Variable 0:1504	Fuzzing Variable 0:1959
Variablesize= 1024	Fuzzing Variable 0:411	Variablesize= 5000	Fuzzing Variable 0:1050	Fuzzing Variable 0:1505	Fuzzing Variable 0:1960
Fuzzing Variable 0:184	Variablesize= 4095	Fuzzing Variable 0:639	Fuzzing Variable 0:1051	Fuzzing Variable 0:1506	Fuzzing Variable 0:1961
Variablesize= 1023	Fuzzing Variable 0:412	Variablesize= 4097	Fuzzing Variable 0:1052	Fuzzing Variable 0:1507	Fuzzing Variable 0:1962
Fuzzing Variable 0:185	Variablesize= 2048	Fuzzing Variable 0:640	Fuzzing Variable 0:1053	Fuzzing Variable 0:1508	Fuzzing Variable 0:1963
Variablesize= 512	Fuzzing Variable 0:413	Variablesize= 4096	Fuzzing Variable 0:1054	Fuzzing Variable 0:1509	Fuzzing Variable 0:1964
Fuzzing Variable 0:186	Variablesize= 1024	Fuzzing Variable 0:641	Fuzzing Variable 0:1055	Fuzzing Variable 0:1510	Fuzzing Variable 0:1965
Variablesize= 420	Fuzzing Variable 0:414	Variablesize= 4095	Fuzzing Variable 0:1056	Fuzzing Variable 0:1511	Fuzzing Variable 0:1966
Fuzzing Variable 0:187	Variablesize= 1023	Fuzzing Variable 0:642	Fuzzing Variable 0:1057	Fuzzing Variable 0:1512	Fuzzing Variable 0:1967
Variablesize= 257	Fuzzing Variable 0:415	Variablesize= 2048	Fuzzing Variable 0:1058	Fuzzing Variable 0:1513	Fuzzing Variable 0:1968
Fuzzing Variable 0:188	Variablesize= 512	Fuzzing Variable 0:643	Fuzzing Variable 0:1059	Fuzzing Variable 0:1514	Fuzzing Variable 0:1969
Variablesize= 256	Fuzzing Variable 0:416	Variablesize= 1024	Fuzzing Variable 0:1060	Fuzzing Variable 0:1515	Fuzzing Variable 0:1970
Fuzzing Variable 0:189	Variablesize= 420	Fuzzing Variable 0:644	Fuzzing Variable 0:1061	Fuzzing Variable 0:1516	Fuzzing Variable 0:1971
Variablesize= 240	Fuzzing Variable 0:417	Variablesize= 1023	Fuzzing Variable 0:1062	Fuzzing Variable 0:1517	Fuzzing Variable 0:1972
Fuzzing Variable 0:190	Variablesize= 257	Fuzzing Variable 0:645	Fuzzing Variable 0:1063	Fuzzing Variable 0:1518	Fuzzing Variable 0:1973
Variablesize= 128	Fuzzing Variable 0:418	Variablesize= 512	Fuzzing Variable 0:1064	Fuzzing Variable 0:1519	Fuzzing Variable 0:1974
Fuzzing Variable 0:191	Variablesize= 256	Fuzzing Variable 0:646	Fuzzing Variable 0:1065	Fuzzing Variable 0:1520	Fuzzing Variable 0:1975
Variablesize= 65534	Fuzzing Variable 0:419	Variablesize= 420	Fuzzing Variable 0:1066	Fuzzing Variable 0:1521	Fuzzing Variable 0:1976

Fuzzing Variable 0:192	Variablesize= 240	Fuzzing Variable 0:647	Fuzzing Variable 0:1067	Fuzzing Variable 0:1522	Fuzzing Variable 0:1977
Variablesize= 32768	Fuzzing Variable 0:420	Variablesize= 257	Fuzzing Variable 0:1068	Fuzzing Variable 0:1523	Fuzzing Variable 0:1978
Fuzzing Variable 0:193	Variablesize= 128	Fuzzing Variable 0:648	Fuzzing Variable 0:1069	Fuzzing Variable 0:1524	Fuzzing Variable 0:1979
Variablesize= 32767	Fuzzing Variable 0:421	Variablesize= 256	Fuzzing Variable 0:1070	Fuzzing Variable 0:1525	Fuzzing Variable 0:1980
Fuzzing Variable 0:194	Variablesize= 65534	Fuzzing Variable 0:649	Fuzzing Variable 0:1071	Fuzzing Variable 0:1526	Fuzzing Variable 0:1981
Variablesize= 32766	Fuzzing Variable 0:422	Variablesize= 240	Fuzzing Variable 0:1072	Fuzzing Variable 0:1527	Fuzzing Variable 0:1982
Fuzzing Variable 0:195	Variablesize= 32768	Fuzzing Variable 0:650	Fuzzing Variable 0:1073	Fuzzing Variable 0:1528	Fuzzing Variable 0:1983
Variablesize= 32765	Fuzzing Variable 0:423	Variablesize= 128	Fuzzing Variable 0:1074	Fuzzing Variable 0:1529	Fuzzing Variable 0:1984
Fuzzing Variable 0:196	Variablesize= 32767	Fuzzing Variable 0:651	Fuzzing Variable 0:1075	Fuzzing Variable 0:1530	Fuzzing Variable 0:1985
Variablesize= 32764	Fuzzing Variable 0:424	Variablesize= 42	Fuzzing Variable 0:1076	Fuzzing Variable 0:1531	Fuzzing Variable 0:1986
Fuzzing Variable 0:197	Variablesize= 32766	Fuzzing Variable 0:652	Fuzzing Variable 0:1077	Fuzzing Variable 0:1532	Fuzzing Variable 0:1987
Variablesize= 32763	Fuzzing Variable 0:425	Variablesize= 8	Fuzzing Variable 0:1078	Fuzzing Variable 0:1533	Fuzzing Variable 0:1988
Fuzzing Variable 0:198	Variablesize= 32765	Fuzzing Variable 0:653	Fuzzing Variable 0:1079	Fuzzing Variable 0:1534	Fuzzing Variable 0:1989
Variablesize= 32762	Fuzzing Variable 0:426	Variablesize= 3042	Fuzzing Variable 0:1080	Fuzzing Variable 0:1535	Fuzzing Variable 0:1990
Fuzzing Variable 0:199	Variablesize= 32764	Fuzzing Variable 0:654	Fuzzing Variable 0:1081	Fuzzing Variable 0:1536	Fuzzing Variable 0:1991
Variablesize= 20000	Fuzzing Variable 0:427	Variablesize= 512	Fuzzing Variable 0:1082	Fuzzing Variable 0:1537	Fuzzing Variable 0:1992
Fuzzing Variable 0:200	Variablesize= 32763	Fuzzing Variable 0:655	Fuzzing Variable 0:1083	Fuzzing Variable 0:1538	Fuzzing Variable 0:1993
Variablesize= 10000	Fuzzing Variable 0:428	Variablesize= 1	Fuzzing Variable 0:1084	Fuzzing Variable 0:1539	Fuzzing Variable 0:1994
Fuzzing Variable 0:201	Variablesize= 32762	Fuzzing Variable 0:656	Fuzzing Variable 0:1085	Fuzzing Variable 0:1540	Fuzzing Variable 0:1995
Variablesize= 5000	Fuzzing Variable 0:429	Variablesize= 1	Fuzzing Variable 0:1086	Fuzzing Variable 0:1541	Fuzzing Variable 0:1996
Fuzzing Variable 0:202	Variablesize= 20000	Fuzzing Variable 0:657	Fuzzing Variable 0:1087	Fuzzing Variable 0:1542	Fuzzing Variable 0:1997
Variablesize= 4097	Fuzzing Variable 0:430	Variablesize= 1	Fuzzing Variable 0:1088	Fuzzing Variable 0:1543	Fuzzing Variable 0:1998
Fuzzing Variable 0:203	Variablesize= 10000	Fuzzing Variable 0:658	Fuzzing Variable 0:1089	Fuzzing Variable 0:1544	Fuzzing Variable 0:1999
Variablesize= 4096	Fuzzing Variable 0:431	Variablesize= 1	Fuzzing Variable 0:1090	Fuzzing Variable 0:1545	Fuzzing Variable 0:2000
Fuzzing Variable 0:204	Variablesize= 5000	Fuzzing Variable 0:659	Fuzzing Variable 0:1091	Fuzzing Variable 0:1546	Fuzzing Variable 0:2001
Variablesize= 4095	Fuzzing Variable 0:432	Variablesize= 1	Fuzzing Variable 0:1092	Fuzzing Variable 0:1547	Fuzzing Variable 0:2002
Fuzzing Variable 0:205	Variablesize= 4097	Fuzzing Variable 0:660	Fuzzing Variable 0:1093	Fuzzing Variable 0:1548	Fuzzing Variable 0:2003
Variablesize= 2048	Fuzzing Variable 0:433	Variablesize= 1	Fuzzing Variable 0:1094	Fuzzing Variable 0:1549	Fuzzing Variable 0:2004
Fuzzing Variable 0:206	Variablesize= 4096	Fuzzing Variable 0:661	Fuzzing Variable 0:1095	Fuzzing Variable 0:1550	Fuzzing Variable 0:2005
Variablesize= 1024	Fuzzing Variable 0:434	Variablesize= 0	Fuzzing Variable 0:1096	Fuzzing Variable 0:1551	Fuzzing Variable 0:2006
Fuzzing Variable 0:207	Variablesize= 4095	Fuzzing Variable 0:662	Fuzzing Variable 0:1097	Fuzzing Variable 0:1552	Fuzzing Variable 0:2007
Variablesize= 1023	Fuzzing Variable 0:435	Variablesize= 5	Fuzzing Variable 0:1098	Fuzzing Variable 0:1553	Fuzzing Variable 0:2008
Fuzzing Variable 0:208	Variablesize= 2048	Fuzzing Variable 0:663	Fuzzing Variable 0:1099	Fuzzing Variable 0:1554	Fuzzing Variable 0:2009
Variablesize= 512	Fuzzing Variable 0:436	Variablesize= 4	Fuzzing Variable 0:1100	Fuzzing Variable 0:1555	Fuzzing Variable 0:2010
Fuzzing Variable 0:209	Variablesize= 1024	Fuzzing Variable 0:664	Fuzzing Variable 0:1101	Fuzzing Variable 0:1556	Fuzzing Variable 0:2011
Variablesize= 420	Fuzzing Variable 0:437	Variablesize= 3	Fuzzing Variable 0:1102	Fuzzing Variable 0:1557	Fuzzing Variable 0:2012
Fuzzing Variable 0:210	Variablesize= 1023	Fuzzing Variable 0:665	Fuzzing Variable 0:1103	Fuzzing Variable 0:1558	Fuzzing Variable 0:2013
Variablesize= 257	Fuzzing Variable 0:438	Variablesize= 6	Fuzzing Variable 0:1104	Fuzzing Variable 0:1559	Fuzzing Variable 0:2014
Fuzzing Variable 0:211	Variablesize= 512	Fuzzing Variable 0:666	Fuzzing Variable 0:1105	Fuzzing Variable 0:1560	Fuzzing Variable 0:2015

Variablesize= 256	Fuzzing Variable 0:439	Variablesize= 37	Fuzzing Variable 0:1106	Fuzzing Variable 0:1561	Fuzzing Variable 0:2016
Fuzzing Variable 0:212	Variablesize= 420	Fuzzing Variable 0:667	Fuzzing Variable 0:1107	Fuzzing Variable 0:1562	Fuzzing Variable 0:2017
Variablesize= 240	Fuzzing Variable 0:440	Variablesize= 21	Fuzzing Variable 0:1108	Fuzzing Variable 0:1563	Fuzzing Variable 0:2018
Fuzzing Variable 0:213	Variablesize= 257	Fuzzing Variable 0:668	Fuzzing Variable 0:1109	Fuzzing Variable 0:1564	Fuzzing Variable 0:2019
Variablesize= 128	Fuzzing Variable 0:441	Variablesize= 12	Fuzzing Variable 0:1110	Fuzzing Variable 0:1565	Fuzzing Variable 0:2020
Fuzzing Variable 0:214	Variablesize= 256	Fuzzing Variable 0:669	Fuzzing Variable 0:1111	Fuzzing Variable 0:1566	Fuzzing Variable 0:2021
Variablesize= 65534	Fuzzing Variable 0:442	Variablesize= 12	Fuzzing Variable 0:1112	Fuzzing Variable 0:1567	Fuzzing Variable 0:2022
Fuzzing Variable 0:215	Variablesize= 240	Fuzzing Variable 0:670	Fuzzing Variable 0:1113	Fuzzing Variable 0:1568	Fuzzing Variable 0:2023
Variablesize= 32768	Fuzzing Variable 0:443	Variablesize= 13	Fuzzing Variable 0:1114	Fuzzing Variable 0:1569	Fuzzing Variable 0:2024
Fuzzing Variable 0:216	Variablesize= 128	Fuzzing Variable 0:671	Fuzzing Variable 0:1115	Fuzzing Variable 0:1570	Fuzzing Variable 0:2025
Variablesize= 32767	Fuzzing Variable 0:444	Variablesize= 8	Fuzzing Variable 0:1116	Fuzzing Variable 0:1571	Fuzzing Variable 0:2026
Fuzzing Variable 0:217	Variablesize= 65534	Fuzzing Variable 0:672	Fuzzing Variable 0:1117	Fuzzing Variable 0:1572	Fuzzing Variable 0:2027
Variablesize= 32766	Fuzzing Variable 0:445	Variablesize= 6	Fuzzing Variable 0:1118	Fuzzing Variable 0:1573	Fuzzing Variable 0:2028
Fuzzing Variable 0:218	Variablesize= 32768	Fuzzing Variable 0:673	Fuzzing Variable 0:1119	Fuzzing Variable 0:1574	Fuzzing Variable 0:2029
Variablesize= 32765	Fuzzing Variable 0:446	Variablesize= 12	Fuzzing Variable 0:1120	Fuzzing Variable 0:1575	Fuzzing Variable 0:2030
Fuzzing Variable 0:219	Variablesize= 32767	Fuzzing Variable 0:674	Fuzzing Variable 0:1121	Fuzzing Variable 0:1576	Fuzzing Variable 0:2031
Variablesize= 32764	Fuzzing Variable 0:447	Variablesize= 12	Fuzzing Variable 0:1122	Fuzzing Variable 0:1577	Fuzzing Variable 0:2032
Fuzzing Variable 0:220	Variablesize= 32766	Fuzzing Variable 0:675	Fuzzing Variable 0:1123	Fuzzing Variable 0:1578	Fuzzing Variable 0:2033
Variablesize= 32763	Fuzzing Variable 0:448	Variablesize= 7	Fuzzing Variable 0:1124	Fuzzing Variable 0:1579	Fuzzing Variable 0:2034
Fuzzing Variable 0:221	Variablesize= 32765	Fuzzing Variable 0:676	Fuzzing Variable 0:1125	Fuzzing Variable 0:1580	Fuzzing Variable 0:2035
Variablesize= 32762	Fuzzing Variable 0:449	Variablesize= 4	Fuzzing Variable 0:1126	Fuzzing Variable 0:1581	Fuzzing Variable 0:2036
Fuzzing Variable 0:222	Variablesize= 32764	Fuzzing Variable 0:677	Fuzzing Variable 0:1127	Fuzzing Variable 0:1582	Fuzzing Variable 0:2037
Variablesize= 20000	Fuzzing Variable 0:450	Variablesize= 3	Fuzzing Variable 0:1128	Fuzzing Variable 0:1583	Fuzzing Variable 0:2038
Fuzzing Variable 0:223	Variablesize= 32763	Fuzzing Variable 0:678	Fuzzing Variable 0:1129	Fuzzing Variable 0:1584	Fuzzing Variable 0:2039
Variablesize= 10000	Fuzzing Variable 0:451	Variablesize= 16	Fuzzing Variable 0:1130	Fuzzing Variable 0:1585	Fuzzing Variable 0:2040
Fuzzing Variable 0:224	Variablesize= 32762	Fuzzing Variable 0:679	Fuzzing Variable 0:1131	Fuzzing Variable 0:1586	Fuzzing Variable 0:2041
Variablesize= 5000	Fuzzing Variable 0:452	Variablesize= 15	Fuzzing Variable 0:1132	Fuzzing Variable 0:1587	Fuzzing Variable 0:2042
Fuzzing Variable 0:225	Variablesize= 20000	Fuzzing Variable 0:680	Fuzzing Variable 0:1133	Fuzzing Variable 0:1588	Fuzzing Variable 0:2043
Variablesize= 4097	Fuzzing Variable 0:453	Variablesize= 29	Fuzzing Variable 0:1134	Fuzzing Variable 0:1589	
Fuzzing Variable 0:226	Variablesize= 10000	Fuzzing Variable 0:681	Fuzzing Variable 0:1135	Fuzzing Variable 0:1590	
Variablesize= 4096	Fuzzing Variable 0:454	Variablesize= 25	Fuzzing Variable 0:1136	Fuzzing Variable 0:1591	

Fuente: Elaboración propia.