CAPACIDADES TÉCNICAS, LEGALES Y DE GESTIÓN PARA EQUIPOS BLUE TEAM Y RED TEAM

EDUARDO ISAAC BALLESTEROS MUÑOZ

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA UNAD
ESCUELA DE CIENCIAS BÁSICAS E INGENIERÍA
ESPECIALIZACIÓN EN SEGURIDAD INFORMÁTICA
BOGOTÁ D.C.
2021

CAPACIDADES TÉCNICAS, LEGALES Y DE GESTIÓN PARA EQUIPOS BLUE TEAM Y RED TEAM

EDUARDO ISAAC BALLESTEROS MUÑOZ

Trabajo de Grado para optar por el título
Especialista en Seguridad Informática

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA UNAD
ESCUELA DE CIENCIAS BÁSICAS E INGENIERÍA
ESPECIALIZACIÓN EN SEGURIDAD INFORMÁTICA
BOGOTÁ D.C.
2021

CONTENIDO

pág.

GLOSARIO 9

RESUMEN 12

INTRODUCCIÓN 13

1. PLANTEAMIENTO DEL PROBLEMA 15

2. JUSTIFICACIÓN 16

3. OBJETIVOS 17

3.1. OBJETIVO GENERAL 17

3.2. OBJETIVOS ESPECIFICOS 17

4. MARCO TEÓRICO 18

4.1. PRUEBAS DE PENETRACIÓN 18

4.1.1. Fase de Reconocimiento 21

4.1.2. Fase de Escaneo 21

4.1.3. Fase de Enumeración 22

4.1.4. Fase de Acceso 22

4.1.5. Fase de Mantenimiento de Acceso 22

4.1.6. Metasploit 22

4.2. ENTORNOS VULNERABLES 29

4.2.1. Damn Vulnerable Web App (DVWA) 29

4.2.2. Bricks 32

4.3. ESCANEO DE PUERTOS 32

4.3.1. Nmap 36

4.4. SNIFFING 37

4.4.1. Sniffer 38

4.4.2. Wireshark 43
4.4.2.1. Funcionamiento de Wireshark 45
4.5. SEGURIDAD EN APLICACIONES WEB 45
4.6. SEGURIDAD EN BASES DE DATOS 51
4.6.1. Inyección SQL 53
4.6.2. SQLMap 58
4.6.3. Havij 59
5. DESARROLLO 60
5.1. PRÁCTICAS NMAP 60
5.2. PRÁCTICAS WIRESHARK 78
5.3. PRÁCTICAS METASPLOIT 90
6. CONCLUSIONES 94
7. RECOMENDACIONES 96
8. BIBLIOGRAFÍA 98

LISTA DE TABLAS

pág.

Tabla 1. Componentes y módulos de Metasploit 23
Tabla 2. Puertos más usados 35
Tabla 3. Clasificación del riesgo 55
Tabla 4. Sqlmap, herramienta de inyección SQL. 59
Tabla 5. CIDR 66
Tabla 6. Operadores para filtros 87

LISTA DE FIGURAS

pág.

Figura 1. Metasploit estructura 24
Figura 2. Módulos de Metasploit 24
Figura 3. Módulos auxiliares presentes en Metasploit 25
Figura 4. Damn Vulnerable Web Application “VWA”, aplicación web vulnerable. 30
Figura 5. DVWA, configuración de niveles de seguridad. 31
Figura 6. Bricks, aplicación web vulnerable 32
Figura 7. 3 way handshake. 34
Figura 8. Usos de los analizadores de paquetes. 40
Figura 9. Ventajas Wireshark. 43
Figura 10. Vulnerabilidades en Aplicaciones Web Con Severidad Alta. 47
Figura 11. Havij. Herramienta de inyección SQL. 59
Figura 12. Nmap escaneo por defecto. nmap 192.168.1.10 60
Figura 13. Escaneo de direcciones IP por rango. nmap 192.168.1.1-4 61
Figura 14. Escaneo de múltiples direcciones IP separadas por comas. nmap 192.168.1.1 192.168.1.10 192.168.1.11 61
Figura 15. Escaneo de múltiples direcciones IP, excluyendo un rango. nmap 192.168.1.1-20 –exclude 192.168.1.9-11 62
Figura 16. Escaneo de múltiples direcciones IP, excluyendo una lista de direcciones IP separadas por coma. 63
Figura 17.Escaneo de direcciones IP por rango con notación "Octate Range". Nmap 192.168.1.1.*. 65
Figura 18. Escaneo de direcciones IP con notación CIDR. Escaneo de direcciones IP por rango con notación CIDR. nmap 192.168.1.0/24 66
Figura 19. Escaneo por rango de puertos. nmap 192.168.1.11 –p1-1024 68
Figura 20. Escaneo por rango de puertos. nmap 192.168.1.11 –p- 68
Figura 21. Escaneo de servicios. Nmap –sV 192.168.1.11 69

Figura 22. Registro de escaneos. Nmap 192.168.1.11 –oA logEduardoBallesteros 70
Figura 23. Registro de escaneos. Archivos creados por Nmap 71
Figura 24. Registro de escaneos. Visualización de archivos creados por Nmap. 71
Figura 25. Registro de escaneos. Visualización de archivos creados por Nmap. 71
Figura 26. Detección de hosts en línea. nmap 192.168.1.11 -sn 72
Figura 27. Detección de host en línea. Validación de ping a host 192.168.1.11 73
Figura 28. Detección de host en línea. Deshabilitar ping en máquina destino Windows 7 73
Figura 29. Detección de host en línea. Deshabilitar ping en máquina destino Windows 7 74
Figura 30. Detección de host en línea. Deshabilitar ping en máquina destino Windows 7 74
Figura 31. Detección de host en línea. Validación de ping a host 192.168.1.11 75
Figura 32. Detección de host en línea con flag –sn. nmap 192.168.1.11 -Pn 75
Figura 33. Detección de host en línea con flag –Pn. nmap 192.168.1.11 –PS 135 76
Figura 34. Sistema operativo del host destino. Windows 7 Pro 76
Figura 35. Detección del Sistema Operativo. nmap 192.168.1.11 -O 77
Figura 36. Wireshark, captura de tráfico 79
Figura 37. Wireshark, captura de tráfico 79
Figura 38. Wireshark, detener captura de tráfico en tiempo real. 80
Figura 39. Wireshark, detener captura de tráfico en tiempo real. Capture Options. 80
Figura 40. Wireshark, detener captura de tráfico en tiempo real. Capture Options. 80
Figura 41. Wireshark, creación de anotaciones. Capture File Properties. 81
Figura 42. Wireshark, creación de anotaciones. Capture File Properties. 81
Figura 43. Wireshark, creación de anotaciones. Packet Comment. 82
Figura 44. Wireshark, creación de anotaciones. Packet Comment. 82

Figura 45. Wireshark, creación de anotaciones. Packet Comment. 83
Figura 46. Wireshark, creación de filtros de anotaciones. Packet Comments. 83
Figura 47. Wireshark, creación de filtros de anotaciones, nueva columna Packet comments 84
Figura 48. Wireshark, creación de filtros de anotaciones.Apply as Filter. 84
Figura 49. Wireshark, creación de filtros de anotaciones.Apply as filter. 85
Figura 50. Wireshark, manejo de filtros. Using this filter. 85
Figura 51. Wireshark, manejo de filtros.Enter a capture filter. 86
Figura 52. Wireshark, manejo de filtros en interfaz principal. 86
Figura 53. Wireshark, manejo de filtros. Paquetes con IP origen y operador eq. 87
Figura 54. Wireshark, manejo de filtros. Paquetes con IP origen y operador ==. 87
Figura 55. Wireshark, manejo de filtros. Paquetes con IP destino y operador eq. 88
Figura 56. Wireshark, manejo de filtros. Paquetes con IP destino y operador ==. 88
Figura 57. Wireshark, manejo de filtros. Paquetes con IP origen o IP destino. 88
Figura 58. Wireshark, manejo de filtros. Paquetes con longitud mayor a 17000 bytes. 89
Figura 59. Wireshark, manejo de filtros. Paquetes con IP origen o IP destino. 89
Figura 60. Metasploit, comando banner 90
Figura 61. Metasploit, comando version 90
Figura 62. Metasploit, comando search 91
Figura 63. Metasploit, comando set y setg 92
Figura 64. Metasploit, comando get y getg 92
Figura 65. Metasploit, comando unset y unsetg 93
Figura 66. Metasploit, comando show options 93

GLOSARIO

Black box: Método de prueba donde el tester no tiene información sobre la estructura interna o el funcionamiento de un sistema.

Bug bounty: Trato ofrecido por muchas organizaciones que permite a las personas a ser recompensadas por informar errores, en particular aquellos asociados a exploits y vulnerabilidades.

Enrutador: Hardware que contribuye a que las redes LAN y WAN dispongan de las capacidades de interacción y conexión. Los enrutadores relacionan los encabezados de cada paquete a un segmento específico de la red y seleccionan la mejor ruta, optimizando el rendimiento de la red.

Exploit: Fragmento de código, programa o secuencia de comandos que se aprovecha de una vulnerabilidad para obtener un acceso no autorizado a un sistema o aplicación.

Firewall: Son dispositivos hardware o software diseñados para defenderse del ataque no autorizado hacia o desde una red. Su objetivo es asegurar que todas las comunicaciones con redes externas se realicen conforme a las políticas de seguridad establecidas por la organización.

Framework: Marco de trabajo, es un conjunto de conceptos, prácticas y herramientas para resolver algún problema.

Fuzzing: Técnica de prueba automatizada en la que se proporcionan datos no válidos, inesperados o aleatorios como entrada a una aplicación.

Host: Es cualquier máquina que tienen una dirección IP y que puede recibir como enviar datos en una red.

ICMP: Protocolo de Control de Mensajes de Internet, proporciona funciones de diagnóstico y de reporte de errores referentes a la entrega de paquetes IP.

IP: Protocolo de capa de red que ofrece un servicio de interconexión no orientada a la conexión, este protocolo ofrece funciones de direccionamiento, especificación de tipo de servicio, fragmentación y ensamblado de paquetes y de seguridad.

NIC Network Interface Card: Conocidas como las tarjetas de red, permiten realizar la interconexión entre diferentes máquinas.

Paquete: Unidad de transmisión del nivel de red del estándar OSI, se compone de un encabezado y datos, el encabezado contiene un número de identificación, las direcciones de origen y de destino, y datos de control de errores.

Payload: Código que es ejecutado en el Sistema durante o después de la explotación para realizar la tarea deseada.

Ping: Señal enviada desde un host con los propósitos de verificar si el otro host se encuentra disponible y poder medir el tiempo de respuesta de este.

PDU: Unidad de datos de protocolo, es un bloque específico de datos transferido en una red.

Riesgo: Todo lo que pueda afectar la confidencialidad, integridad y disponibilidad de los datos. Software sin parchear, servidores mal configurados, hábitos inseguros de navegación en Internet, son ejemplos que contribuyen al riesgo.

Sniffer: Software encargado de interceptar datos que circulan por una red, almacenan el tráfico para su posterior análisis, con el objetivo de conseguir información.

TCP: Protocolo de control de transmisión, este protocolo es de la capa de transporte y proporciona un circuito virtual denominado conexión, tiene chequeo de errores, control de flujo, capacidad de interrupción y es full dúplex.

Threat: Todo lo que tenga potencial de causar daños a un sistema, red o aplicación.

Vulnerabilidad: Debilidad en un Sistema que puede permitir que un atacante obtenga acceso no autorizado a él.

White Box: Método de prueba durante el cual el tester tienen un conocimiento completo de la estructura interna y el funcionamiento del sistema.

RESUMEN

Actualmente, la seguridad informática se ha convertido en un tema de gran relevancia para la sociedad, se producen a diario ciberdelitos que atentan contra el derecho a la intimidad de las personas y causan pérdidas económicas a particulares y empresas.

La seguridad informática se caracteriza por una evolución continua en la cual se producen cambios rápidamente. Por ello, se hace necesario que los profesionales puedan estar en continua evolución no sólo a nivel teórico sino también a nivel práctico. El entrenamiento práctico en seguridad informática está adquiriendo mayor importancia en la formación de personal especializado, el amplio número de incidentes que ocurren diariamente nos recuerdan constantemente la necesidad de una formación completa utilizando un enfoque creativo y eficaz ya que es necesaria no sólo una formación teórica sino también una formación práctica.

INTRODUCCIÓN

La información, el conocimiento y la tecnología son herramientas vitales para el desarrollo y el crecimiento económico, social y cultural de todos los países. Las sociedades en que vivimos, llamadas por los sociólogos, sociedades de la información[1], están caracterizadas, entre otras cualidades, por el manejo de un volumen colosal y aun así aceleradamente creciente, de información de todo tipo, lo que conlleva a métodos y técnicas de almacenamiento de los datos y acceso a los mismos radicalmente diferentes de las usadas anteriormente por el hombre[2].

Nos encontramos en la era de la conectividad universal, de virus, de hackers, de fraudes electrónicos, no hay un momento en el que no nos importe la seguridad y el enorme crecimiento de uso de los computadores y sus respectivas interconexiones mediante redes como internet, ha hecho que organizaciones e individuos dependan cada vez más de la información que se almacena y se transmite a través de estos sistemas, esto ha llevado a un aumento de la conciencia de la necesidad de proteger la información y garantizar la autenticidad de la misma.

Las aplicaciones web debido a su naturaleza de estar en línea las expone a la visita de usuarios que llegan con objetivos distintos, uno de ellos es el hacerse fama violando la seguridad de estos sitios, valiéndose de diferentes estrategias y herramientas.

---

[1] La Sociedad de la Información, Recuperado el 3 de Octubre de 2020 del sitio web de McGraw-Hill Interamericana de España, SL: http://www.mcgraw-hill.es/bcv/guide/capitulo/8448146905.pdf.

[2] GÓMEZ, Alberto y DE ABAJO, Nicolás. Los Sistemas De Información En La Empresa. 1ª Edición. Oviedo Principado De Asturias: Servicio De Publicaciones De La Universidad De Oviedo, 1997. 3 p.

Las amenazas que pesan sobre los sitios web son de naturaleza bastante diversa y están en constante evolución. Para el caso de específico de este proyecto se trabajarán las amenazas de tipo inyección de código SQL a sitios web.

1. PLANTEAMIENTO DEL PROBLEMA

La rápida evolución de los sistemas de información en estos tiempos requiere que todas las organizaciones adopten un conjunto mínimo de controles de seguridad para proteger sus sistemas de información. Preservar la confidencialidad, disponibilidad e integridad de la información, es una prioridad para cualquier entidad u organización, la falta de personal calificado causa que las organizaciones en esta área se encuentren expuestas a un nivel de amenaza muy alto que puede conllevar a la pérdida de información crítica y en muchos casos originar la detención de servicios y procesos fundamentales.

Un conocimiento adecuado de las herramientas de seguridad tanto a nivel ofensiva como defensiva es fundamental en cualquier organización y puede marcar la diferencia al momento de hacer frente a algún evento o incidente informático en una organización.

Con este fin, se quiere realizar un análisis y evaluación de las diferentes herramientas para pruebas de penetración, contención y detección desde un punto de vista práctico con el fin de mejorar los esquemas de ciberseguridad de las organizaciones.

¿Cómo la implementación y el buen uso de las herramientas adecuadas para pruebas de penetración, contención y detección ayudan a detectar y minimizar el impacto de incidentes informáticos en las organizaciones?

2. JUSTIFICACIÓN

Los incidentes informáticos son cada vez más frecuentes y las estadísticas nos muestran que Colombia es un país vulnerable, la probabilidad de ser víctima es bastante alta si no se tienen unos protocolos y buenas prácticas de seguridad bien establecidos o si no se tiene el personal idóneo en las organizaciones para poder hacer frente a este tipo de problemas.

Con este análisis de herramientas de seguridad ofensiva y defensiva tanto a nivel teórico como práctico, se les permitirá a las personas entender de una mejor forma y a poder establecer las herramientas que más se adecuen según el caso para así poder generar de manera oportuna estrategias para fortalecer la seguridad de la información y una rápida y potencial remediación de un incidente de seguridad que se pueda presentar en un momento determinado.

3. OBJETIVOS

3.1. OBJETIVO GENERAL

Analizar diferentes herramientas para pruebas de penetración, contención y detección desde un punto de vista práctico.

3.2. OBJETIVOS ESPECIFICOS

- Adquirir la mayor cantidad de conocimiento práctico sobre el uso de herramientas para pruebas de penetración, contención y detección.

- Realizar un análisis de diferentes herramientas para pruebas de penetración, contención y detección.

- Generar recomendaciones de herramientas para seguridad ofensiva y defensiva.

4. MARCO TEÓRICO

4.1. PRUEBAS DE PENETRACIÓN

El uso de la tecnología ha aumentado de forma exponencial, la mayoría de las empresas dependen parcial o completamente del uso de la tecnología, si bien estos adelantos cambian completamente la forma en que se hacen las cosas facilitando todo tipo de trabajo, también traen consigo nuevas amenazas.

Los atacantes descubren formas nuevas e innovadoras de manipular estas tecnologías para obtener ganancias, siendo este un motivo de preocupación para miles de organizaciones y empresas de todo el mundo, debido al gran desafío por mantener seguros sus datos. La protección de los datos es ciertamente importante, siendo igualmente importante comprobar si se han puesto en funcionamiento los mecanismos de protección adecuados, estos mecanismos de protección pueden fallar, por lo que probarlos antes de que alguien los pueda explotar es un gran desafío en la actualidad.

La evaluación de vulnerabilidades y las pruebas de penetración han ganado una gran importancia y ahora se incluyen trivialmente en todos los programas de cumplimiento. Con la evaluación de vulnerabilidades y las pruebas de penetración realizadas de manera correcta, las organizaciones pueden asegurarse de que han implementado los controles de seguridad correctos y que están funcionando como se esperaba.

Se debe diferenciar entre evaluación de vulnerabilidades y pruebas de penetración, si bien son conceptos que están relacionados, primero se puede realizar una evaluación de la vulnerabilidad del objetivo para evaluar las debilidades del sistema y posteriormente realizar una prueba de penetración planificada para verificar si el objetivo es vulnerable o no, sin realizar una

evaluación de una vulnerabilidad no será posible planificar y ejecutar una prueba de penetración. La mayoría de las evaluaciones de vulnerabilidad son de naturaleza no invasiva a diferencia de las pruebas de penetración que posiblemente pueden causar daños al objetivo si no se realiza de manera controlada, dependiendo de las necesidades específicas de cumplimiento, algunas organizaciones eligen solo realizar evaluaciones de vulnerabilidades, mientras que otras continua y realizar también pruebas de penetración.

Las pruebas de penetración son una práctica para poner a prueba un sistema informático, red o aplicación para encontrar vulnerabilidades que un atacante podría explotar. Son consideradas ataques autorizados a un sistema informático que se realiza para evaluar la seguridad del sistema o red, esta prueba se realiza para identificar vulnerabilidades e identificar riesgos. Consiste en una serie de pruebas ofensivas contra los mecanismos de defensa existentes en el entorno que se está analizando, estas pruebas comprenden desde el análisis de dispositivos físicos y digitales, hasta el análisis del factor humano utilizando Ingeniería Social.

Estas pruebas, buscan verificar bajo ciertas situaciones cuál es el comportamiento de los mecanismos de defensa, específicamente, se busca detectar vulnerabilidades en los mismos. Además, se identifican aquellas faltas de controles y brechas de seguridad que pueden existir entre la información crítica y los controles existentes. El principal objetivo de las pruebas de penetración consiste en determinar las debilidades de seguridad, una prueba de penetración también puede ser utilizada para probar el cumplimiento de la política de seguridad de una organización, la conciencia de seguridad de sus empleados y la capacidad de la organización para identificar y responder a los incidentes de seguridad.

Se debe tener en cuenta que una buena prueba de penetración no se trata solo de ejecutar un conjunto de algunas herramientas automatizadas contra un objetivo

específico. Las pruebas de penetración se deben entender como un proceso completo que involucra múltiples etapas y cada etapa es igualmente importante para el éxito, para realizar todas las tareas planteadas a lo largo de todas las etapas de las pruebas de penetración, se necesitan usar varias herramientas diferentes y en muchos casos se hace necesario realizar algunas tareas manualmente. Luego, al final, se deben combinar los resultados de todas las herramientas utilizadas para generar un informe único y significativo. [3]

Las pruebas de penetración se pueden clasificar en tres tipos: caja blanca, caja negra y caja gris.

Una prueba de penetración de caja blanca es un tipo de prueba en la que el cliente comparte completamente todos los datos y los detalles relacionados con el sistema, la red o la aplicación de destino, como las credenciales de inicio de sesión y el código fuente de la aplicación que debe probarse. Dado que la información proporcionada por el cliente con respecto a su sistema es altamente confidencial, se recomienda que se realice un cifrado de toda esta información.

Una prueba de penetración de caja negra es una prueba simulada por un atacante en la que el probador de penetración actuará como un actor de amenazas sin información interna sobre los sistemas, redes o aplicaciones objetivo. Este tipo de prueba realmente se centra en una fase denominada de reconocimiento. Cuanta más información pueda obtener un pentester sobre una organización objetivo, mejores serán los resultados. En este tipo de prueba, el tester no cuenta con ningún diagrama de arquitectura, diseño de la red ni ningún archivo de código fuente.

---

[3] SAGAR, Rahalkar. Metasploit 5.0 for Beginners. Perform Penetration Testing to Secure Your IT Environment Against Threats and Vulnerabilities. 2ª Edición. Packt Publishing Ltd, 2020.

Y una prueba de penetración de caja gris es el punto medio entre la prueba de caja blanca y la de caja negra. En una prueba típica de caja gris, el tester recibe algunos conocimientos de las aplicaciones, sistemas o redes. Por su naturaleza, este tipo de prueba es bastante eficiente, usando la información proporcionada por el cliente, el pentester puede enfocarse en los sistemas con mayores riesgos y ahorrar mucho tiempo realizando su propio reconocimiento.

Una prueba de penetración comprende múltiples etapas con diferentes tipos de actividades en distintos ámbitos y entornos, la profundidad con que se lleven a cabo las actividades dependerá de ciertos factores, entre los que se destaca el riesgo que puede generar hacia el cliente alguno de los métodos que se apliquen durante la evaluación.

A continuación, se describen las diferentes etapas llevadas a cabo al momento de hacer un pentesting:

4.1.1. Fase de Reconocimiento. Posiblemente, esta sea una de las etapas que más tiempo demande. Asimismo, se definen objetivos y se recopila toda la información posible que luego será utilizada a lo largo de las siguientes fases. La información que se busca abarca desde nombres y direcciones de correo de los empleados de la organización, hasta la topología de la red, direcciones IP, entre otros. Cabe destacar que el tipo de información o la profundidad de la prueba dependerán de los objetivos que se hayan fijado en la auditoría.

4.1.2. Fase de Escaneo. Utilizando la información obtenida previamente se buscan posibles vectores de ataque. Esta etapa involucra el escaneo de puertos y servicios. Posteriormente se realiza el escaneo de vulnerabilidades que permitirá definir los vectores de ataque.

4.1.3. Fase de Enumeración. El objetivo de esta etapa es la obtención de los datos referente a los usuarios, nombres de equipos, servicios de red, entre otros. A esta altura de la auditoría, se realizan conexiones activas con el sistema y se ejecutan consultas dentro del mismo.

4.1.4. Fase de Acceso. En esta etapa finalmente se realiza el acceso al sistema. Esta tarea se logra a partir de la explotación de aquellas vulnerabilidades detectadas que fueron aprovechadas por el auditor para comprometer el sistema.

4.1.5. Fase de Mantenimiento de Acceso. Luego de haberse obtenido el acceso al sistema, se busca la manera de preservar el sistema comprometido a disposición de quien lo ha atacado. El objetivo es mantener el acceso al mencionado sistema perdurable en el tiempo.

4.1.6. Metasploit. Se considera una de las herramientas de auditoría más efectivas para realizar pruebas de penetración en la actualidad. Metasploit ofrece una amplia variedad de exploits, un excelente entorno de desarrollo de estos, capacidades de recopilación de información y pruebas web, entre muchas otras cosas.

Nace en 2003 cuando H. D. Moore escribió una herramienta de red utilizando el lenguaje de programación Perl, en 2007, se reescribió en Ruby. El proyecto Metasploit recibió un gran impulso comercial cuando la empresa Rapid7 adquirió el proyecto en 2009. Metasploit es esencialmente un marco de pruebas de penetración robusto y versátil donde se pueden realizar todas las tareas que están involucradas en un ciclo de vida de pruebas de penetración. Con la ayuda de Metasploit solo se necesita enfocarse en los objetivos centrales, las acciones de apoyo se realizarán a través de varios componentes y módulos del marco que posee Metasploit, dado que es un marco completo y no solo una aplicación, se

puede personalizar y ampliar según nuestros requisitos. Metasploit es, sin duda, una herramienta muy poderosa para las pruebas de penetración, sin embargo, ciertamente no es una vara mágica que pueda ayudarte a hackear cualquier sistema, es importante comprender las capacidades de Metasploit para poder aprovecharlo de manera óptima durante las pruebas de penetración. [4]

La siguiente tabla lista varios de los componentes y módulo de Metasploit que pueden ser usados en cada uno de las fases de una prueba de penetración:

Tabla 1. Componentes y módulos de Metasploit

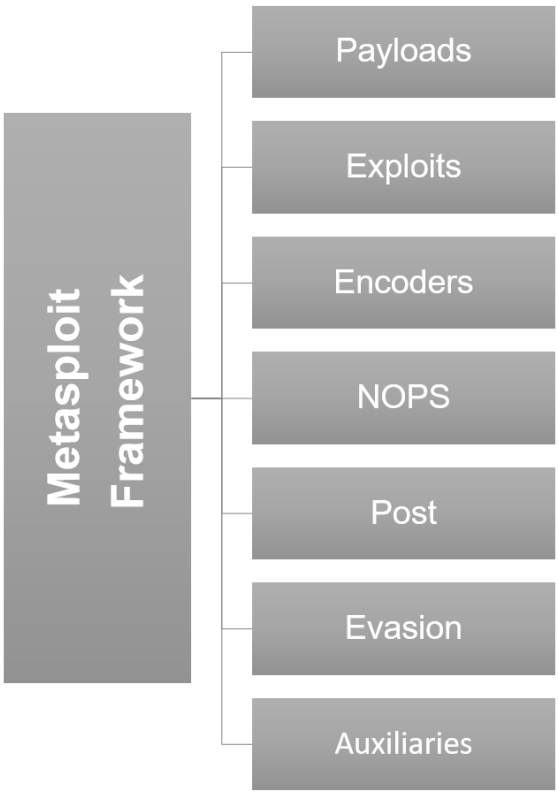
Fase	Componente o módulo de Metasploit
Recopilación de información	Módulos auxiliares: portscan/syn, portscan/tcp, smb_version, db_nmap, scanner/ftp/ftp_version.
Enumeración	smb/smb_enumshares, smb/smb_enumusers, smb/smb_login.
Obtener acceso	Todos los exploits y payloads.
Escalada de privilegios	meterpreter-use priv y meterpreter-getsystem
Mantener acceso	meterpreter –run persistence
Cubriendo pistas	Metasploit Anti-Forensic Project

Fuente: Autor

Metasploit tiene varias categorías de componentes según su función en las fases de prueba de penetración. Cada una de las categorías de componentes tiene varios módulos y complementos para usar en el proceso de explotación.

[4] SAGAR, Rahalkar. Metasploit 5.0 for Beginners. Perform Penetration Testing to Secure Your IT Environment Against Threats and Vulnerabilities. 2ª Edición. Packt Publishing Ltd, 2020.

Figura 1. Metasploit estructura



Fuente: Autor

Figura 2. Módulos de Metasploit

```
estudiante@seminario:/usr/share/metasploit-framework/modules$ ls /usr/share/metasploit-framework/modules/
auxiliary  encoders  evasion  exploits  nops  payloads  post
```

Fuente: Autor

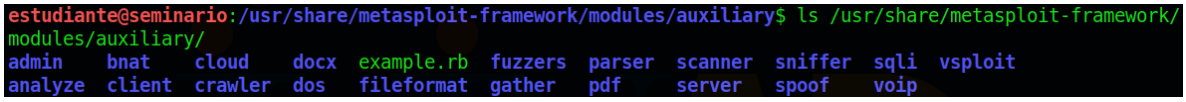
A continuación, se detallan cada uno de los componentes:

Módulos Auxiliares: Metasploit es un framework de prueba de penetración completo y no solo una herramienta, debido a que no es simplemente un software sino un framework, significa que consta de muchas herramientas y utilidades. Los módulos auxiliares en Metasploit Framework no son más que pequeñas piezas de código que están destinadas a realizar tareas específicas dentro del ciclo de vida de pruebas de penetración, como, por ejemplo, la tarea simple de verificar si un

certificado de un servidor en particular ha caducado o no, o verificar si alguno de los servidores FTP permite el acceso anónimo. Estas tareas se pueden realizar utilizando los módulos auxiliares presentes en Metasploit. Hay más de 1,000 módulos auxiliares distribuidos en 19 categorías en Metasploit Framework.

La siguiente figura muestra las categorías de módulos auxiliares presentes en Metasploit.

Figura 3. Módulos auxiliares presentes en Metasploit



Fuente: Autor

No es necesario conocer todos y cada uno de los módulos individualmente, pero es importante saber dónde encontrar estos módulos auxiliares y buscar el módulo correcto en el contexto requerido.

Payloads: Los payloads en Metasploit nos permiten decidir qué acciones se deben realizar en el sistema objetivo una vez que el exploit sea exitoso. A continuación, se enumeran los tipos de payloads.

- Singles: a veces también se denominan inline payloads. Los payloads en esta categoría son una unidad completamente autónoma del exploit y requieren shellcode, lo que significa que tienen todo lo que se requiere para explotar la vulnerabilidad en el objetivo. La desventaja de estos payloads son su tamaño, dado que contienen el código de shell y el exploit completo, a veces pueden ser bastante pesados, lo que los vuelve inútiles en ciertos escenarios.
- Stagers: Existen ciertos escenarios en los que el tamaño del payload es muy importante, este tipo de payload simplemente establece una conexión entre el sistema atacante y el sistema objetivo, no tiene el shellcode necesario para

explotar la vulnerabilidad en el sistema de destino. Al ser de tamaño muy pequeño, encaja bien en muchos escenarios.

- Stages: Una vez que el payload ha establecido una conexión entre el sistema atacante y el sistema de destino, los stages payloads se descargan en el sistema de destino. Contienen el código de shell necesario para aprovechar la vulnerabilidad en el sistema de destino.

Exploits: Un exploit es un fragmento de código que da el acceso requerido al sistema de destino. Existen más de 2500 exploits repartidos en más de 19 categorías según el sistema operativo o plataforma compatible. La decisión de utilizar un exploit en particular contra un objetivo solo se puede tomar después de una extensa enumeración y evaluación de las vulnerabilidades de nuestro objetivo.

Una enumeración adecuada y una evaluación de vulnerabilidades del objetivo nos proporcionará la siguiente información en función de la cual podemos elegir el exploit correcto:

- Sistema operativo del sistema de destino (incluida la versión exacta y la arquitectura).
- Puertos abiertos en el sistema de destino (Protocolo de control de transmisión (TCP) y Protocolo de datagramas de usuario (UDP).
- Servicios junto con versiones que se ejecutan en el sistema de destino.
- Probabilidad de que un servicio en particular sea vulnerable.

A continuación, se enumeran algunas de las categorías de exploits disponibles en Metasploit:

- Linux
- Windows
- Unix

- Apple iOS
- Android
- Php
- Mssql
- Wifi
- Solaris

Encoders: En cualquier escenario de prueba de penetración del mundo real, es muy posible que el intento de atacar el sistema objetivo sea detectado por algún tipo de software de seguridad presente en el sistema objetivo, esto puede poner en peligro el acceso al sistema remoto. El trabajo de los encoders es ofuscar nuestro exploit y payload de tal manera que en el sistema de destino pase desapercibido para todos los sistemas de seguridad. A continuación, se listan algunas de las categorías de encoders disponibles en Metasploit:

- cmd
- mipsle
- ruby
- sparc
- ppc
- x86
- x64
- php
- generic

NOPS: En el contexto del lenguaje ensamblador NOP significa instrucción sin operación. Los NOP pueden ser útiles en ocasiones al escribir exploits o shellcodes. Agregar NOP puede ayudar significativamente a modificar las firmas del payload y por lo tanto evitar la detección. Metasploit viene con NOP para varias plataformas, como se listan a continuación:

- aarch64
- aarmle
- mipsbe
- php
- ppc
- sparc
- tty
- x64
- x86

POST: Los módulos de POST contienen varios scripts y utilidades que nos ayudan a penetrar más en nuestro sistema destino después de una explotación exitosa, una vez que se explota con éxito una vulnerabilidad y se accede a un sistema destino, los módulos POST ayudan a:

- Escalar privilegios de usuario
- Volcar credenciales del SO
- Robar cookies y contraseñas guardadas
- Obtener registros de claves del sistema de destino
- Ejecutar scripts de PowerShell
- Hacer el acceso persistente

Metasploit tiene más de 250 programas de utilidad y scripts posteriores a la explotación.

A continuación, se listan algunas categorías de módulos POST disponibles en Metasploit:

- Linux
- Windows
- Cisco

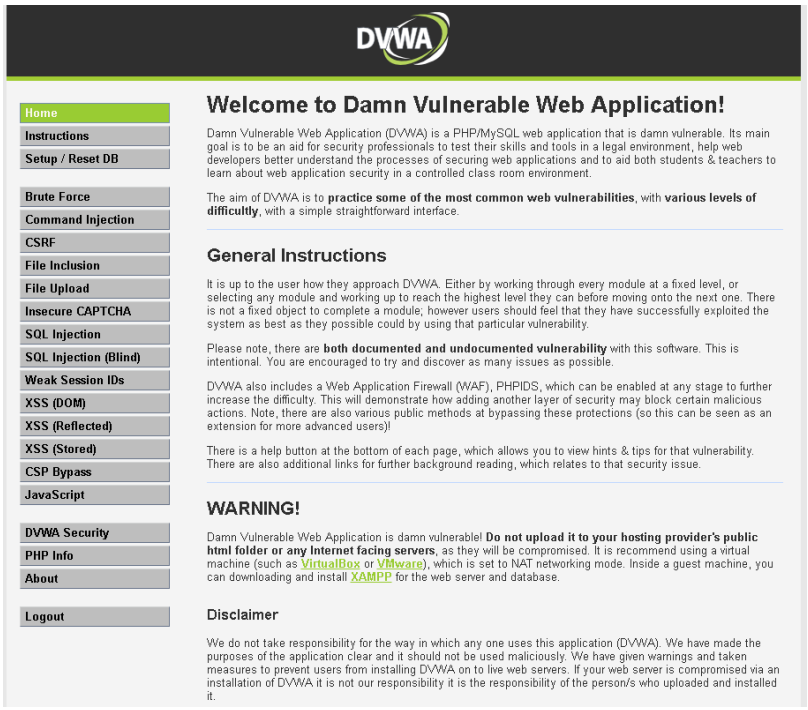
- Solaris
- Android
- OS X
- Zip
- Juniper
- PowerShell

Evasion: La mayoría de los payloads y los códigos de shell que se generan a partir de Metasploit son detectados por software de seguridad como los antivirus. Para evitar la detección, los payloads deben modificarse. La última versión de Metasploit ofrece módulos de evasión especiales que ayudarán a modificar los payloads para evitar la detección.

4.2. ENTORNOS VULNERABLES

4.2.1. Damn Vulnerable Web App (DVWA). Es un entorno de entrenamiento en explotación de seguridad web escrito en el lenguaje de programación PHP y tiene como base de datos MySQL, cuyo objetivo principal es permitir a programadores a investigar sobre las diferentes temáticas existentes en el campo de la seguridad web en un entorno completamente legal.

Figura 4. Damn Vulnerable Web Application “VWA”, aplicación web vulnerable.



Fuente: Autor

Gracias a su programación deliberadamente vulnerable es posible realizar pruebas de cada uno de los diferentes tipos de ataques que se pueden realizar a sitios web y más concretamente sobre páginas web dinámicas implementadas con PHP.

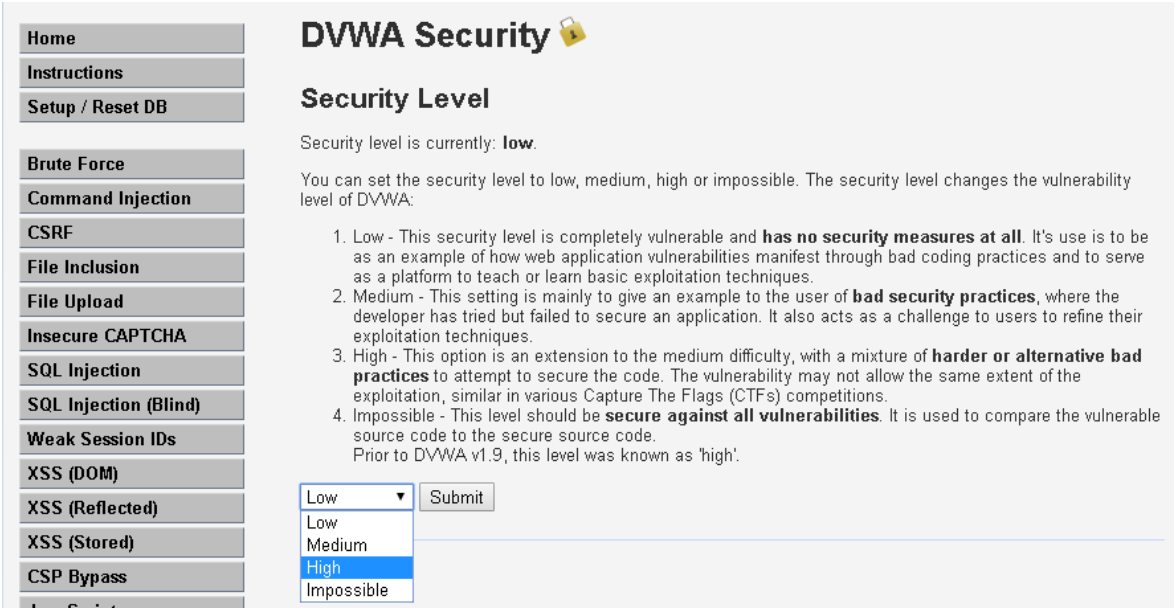
DVWA permite el análisis del código vulnerable a los siguientes ataques:

- Brute Force.
- Command Injection.
- CSRF (Cross-Site Request Forgery).
- File Inclusion (Local File Inclusion y Remote File Inclusion).
- File Upload.
- Insecure CAPTCHA.
- SQL Injection.

- SQL Injection (Blind).
- Weak Sessions IDs.
- XSS (DOM).
- XSS (Reflected).
- XSS (Stored).
- CSP Bypass.
- Javascript.

De igual forma dispone de tres niveles de seguridad diferentes: low, medium, high e imposible (bajo, medio, alto, e imposible respectivamente) que el usuario puede cambiar en cualquier momento.

Figura 5. DVWA, configuración de niveles de seguridad.

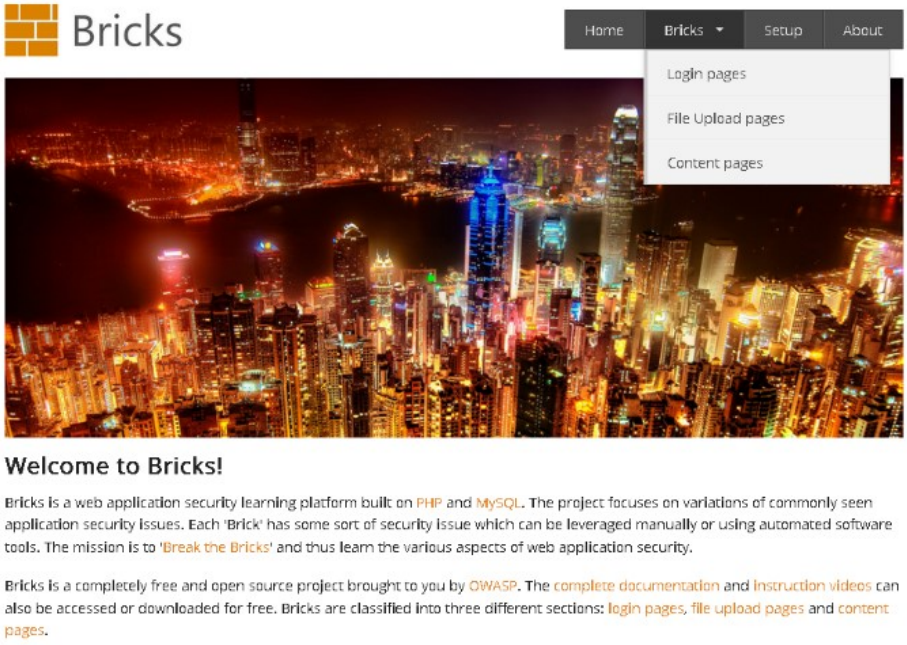


Fuente: Autor

Gracias a los diferentes niveles de seguridad existentes se pueden apreciar las diferencias entre código seguro y bien estructurado, y código vulnerable siguiendo malas prácticas de programación.

4.2.2. Bricks. Es una plataforma de aprendizaje de seguridad de aplicaciones web, está construida en PHP y tiene como base de datos MySQL, el proyecto se centra en las variaciones de los problemas de seguridad que se ven comúnmente en las aplicaciones web. Cada “brick” tiene algún tipo de problema de seguridad que puede aprovecharse manualmente o mediante el uso de herramientas automatizadas. Bricks es un proyecto completamente gratuito y de código abierto que ofrece OWASP, los “bricks” se clasifican en tres secciones diferentes: páginas de inicio de sesión, páginas de carga de archivos y páginas de contenido.

Figura 6. Bricks, aplicación web vulnerable



Fuente: Autor

4.3. ESCANEO DE PUERTOS

La capa de red transfiere datagramas entre dos máquinas por medio de la red utilizando como identificadores las direcciones IP, la capa de transporte incluye el concepto de puerto para distinguir entre los muchos destinos dentro de un mismo

host. Un puerto es una forma de acceder a un servicio en red en un computador, este es un número de 16 bits, por tal razón cada computador tiene 65.535 puertos que pueden estar abiertos o cerrados en cualquier momento. Cada aplicación que se encuentre esperando un mensaje se encuentra escuchando un puerto determinado. Los puertos no solo se usan para la recepción de mensajes, las aplicaciones utilizan los puertos para recibir y también para transmitir mensajes. [5]
Los puertos se identifican comúnmente colocando dos puntos después de una dirección IP, por ejemplo, si se observa la siguiente dirección 127.0.0.1:22, entonces se dice que se está apuntando a la dirección IP 127.0.0.1 y al puerto 22.

Los puertos tienen un buffer situado entre el software de aplicación y la red, de tal forma que las aplicaciones transmiten los datos a los puertos y estos datos se van almacenando en este buffer hasta que pueda enviarse por la red, conforme se va transmitiendo, los datos van llegando al puerto destino donde se va almacenando hasta que la aplicación se encuentre disponible para recibirla.

Las aplicaciones cliente realizan la asignación de puertos de forma dinámica y generalmente utilizan valores superiores al valor de 1024, cuando una aplicación cliente requiere comunicarse con un servidor, busca un número de puerto libre y lo utiliza, en cambio las aplicaciones de servidor utilizan números de puerto ya fijos, algunos servicios como el HTTP o el FTP tienen puertos que están asociados a ellos de forma predeterminada, HTTP se ejecuta en el puerto 80, FTP se ejecuta en el puerto 21. [6]

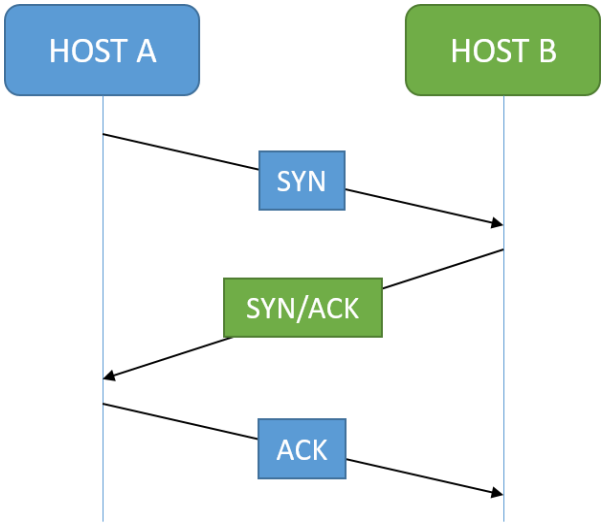
El escaneo de puertos permite realizar auditoría de redes y máquinas para identificar que puertos se encuentran abiertos. Un escáner de puertos es

---

[5] SOLSONA Antonio, HUIDROBO José, JORDAN Julia. Redes de área local: administración de sistemas informáticos. Editorial Paraninfo, 2006.
[6] SHAW, David. Nmap Essentials. Community experience distilled. Packt Publishing Ltd, 2015.

simplemente una herramienta a nivel de software que intenta conectarse a cada puerto del destino que se le especifique y validar si el puerto se encuentra abierto. Para realizar el escaneo de puertos, se utilizan diversas técnicas basadas en el protocolo TCP, que utilizan la activación de flags de la cabecera de este protocolo, una de las maneras más sencillas de identificar el estado de un puerto "abierto, cerrado o filtrado" es intentando conectarse a él, a partir del proceso conocido como "3 way handshake". [7]

Figura 7. 3 way handshake.



Fuente: Autor

El host A solicita una conexión al host B, A envía un SYN "bit de control dentro del segmento TCP, usado para sincronizar una conexión" a B en un puerto específico, si B quiere establecer la conexión, B le envía una respuesta SYN/ACK "mensaje de acuse de recibo de la recepción del mensaje" a A, A recibe esta respuesta y verifica que la conexión se haya establecido enviando a B un ACK.

La forma anteriormente descrita es válida y efectiva, pero desde el punto de vista del atacante es muy ruidosa, debido a que deja muchos registros en el objetivo y

[7] JARA Héctor, PACHECO Federico. Ethical Hacking 2.0, Fox Andina. 1ª Edición, 2012.

se puede detectar con facilidad, por tal razón existe una variante a este tipo de escaneo y es el SYN Scan, este tipo de escaneo difiere en que en vez de responder el último paso con un paquete que tenga el flag ACK activado y finalmente establecer la conexión, envía un RST de modo tal que corta la conexión, dejando menos rastros en el objetivo. [8]

A continuación, se enumeran los puertos más usados:

Tabla 2. Puertos más usados

Protocolo de aplicación	Número de puerto
ftp-data	20
ftp-control	21
telnet	23
smtp	25
dns	53
tftp	69
http	80
https	443
smtps	465
telnets	992
ftps-data	989
ftps-control	990

Fuente: Autor

Al escanear puertos, generalmente el interés no solo radica en saber si un puerto está abierto o no, más que solo entender si un puerto está abierto, es importante entender cuál es el servicio que está utilizando dicho puerto debido a que en los análisis de vulnerabilidades lo que habitualmente se pretender determinar es que versión de un servicio está actuando sobre un puerto particular para poder buscar vulnerabilidades.

Cuando se habla del tema de banners en cuanto al reconocimiento de sistemas, nos estamos refiriendo a cadenas de texto que nos indican explícitamente su

[8] JARA Héctor, PACHECO Federico. Ethical Hacking 2.0, Fox Andina. 1ª Edición, 2012

nombre, versión y otras cosas más, el banner grabbing es un método para poder identificar qué sistema se encuentra detrás de un servicio, este se puede aplicar a cualquier tipo de servicio, pero se resalta que no siempre es posible obtener dicha información ya que los servidores pueden implementar funciones para poder ocultar datos frente a las diferentes peticiones que se le realizan.

4.3.1. Nmap. Es un escáner de puertos de red extremadamente potente que se utiliza para identificar equipos en una red. Nmap es gratuito, flexible, potente y fácil de implementar, lo que lo convierte en una de las utilidades más utilizadas en el área de la seguridad informática. Muchos administradores de sistemas y redes también lo utilizan para tareas como el inventariado de equipos en una red, la gestión de programas de actualización de servicios y la supervisión del tiempo de actividad de los host's y servicios.

Nmap utiliza paquetes IP sin procesar de diferentes formas para determinar qué hosts están disponibles en la red, qué servicios (nombre de la aplicación y versión) ofrecen esos hosts, qué sistemas operativos (y versiones de SO) están ejecutando. Fue diseñado para escanear rápidamente grandes redes, pero funciona bien al escanear hosts individuales. Nmap se ejecuta en todos los principales sistemas operativos existentes y los paquetes binarios oficiales están disponibles para Linux, Windows y Mac OS X.

Esta herramienta de seguridad es utilizada por los pentesters y administradores de sistemas para muchas tareas de red diferentes, hoy en día es aclamado como una de las mejores herramientas para el reconocimiento de redes y de auditoría de seguridad en la industria de la seguridad de la información. La primera versión pública se presentó como un escáner de puertos avanzado, pero en su continua evolución se ha convertido en algo mucho más que eso, se ha convertido en una herramienta esencial que incluye otros grandes subproyectos, como Ncrack (se

centra en ataques de fuerza bruta a servicios de autenticación de red), Ncat (permite a los usuarios leer, escribir, redirigir y modificar datos de red), Nping (se especializa en la elaboración de paquetes de red), Zenmap (es una GUI multiplataforma centrada en la usabilidad de Nmap) y Nmap Scripting Engine (es una funcionalidad de automatización por medio de scripts para que los usuarios escriban sus propias tareas)[9].

4.4. SNIFFING

Es la monitorización del tráfico de red en tiempo real, es una técnica utilizada para poder escuchar todo lo que ocurre dentro de una red. Existen dos técnicas de sniffing: activas y pasivas, que se usan de acuerdo a la estructura de red a evaluar. La técnica pasiva funciona con concentradores pero si hay conmutadores implicados, se utiliza el sniffing activo.

Sniffing pasivo: Al utilizar concentradores no existen mecanismos reguladores que dirijan el tráfico a su destinatario; en lugar de ello, todos los dispositivos reciben todo el tráfico y luego determinan si ese tráfico es relevante o no. Como todos los equipos del concentrador reciben todo el tráfico de la red, el sniffer puede escuchar fácilmente y de forma pasiva todo el tráfico que se envía.

Sniffing activo: Al utilizar conmutadores que regulan el tráfico de la red enviando los datos al dispositivo al que están destinados, un sniffer pasivo en un concentrador de red solo podrá ver los datos que entren y salgan de esa máquina. Para poder acceder a todo el tráfico que circula en la red, un sniffer inyecta tráfico adicional en la red, por este motivo se indica que es un proceso activo, esta

[9] CALDERON, Paulino. Network Exploration and Security Auditing Cookbook. 2ª Edición. Packt Publishing Ltd, 2017.

técnica es más fácil de detectar, porque ese tráfico adicional delata la presencia del sniffer. [10]

4.4.1. Sniffer. Conocidos como analizadores de paquetes, son herramientas de software o hardware destinados para detectar tramas en la red, capturan los paquetes que fluyen por las tramas de red, pero no solo recogen los paquetes destinados al mismo equipo, sino que recogen todos los paquetes que circulen por la red. [11]

El análisis de paquetes es el proceso de examinar los paquetes para entender las características y la estructura del flujo de tráfico. Al monitorear la red para su análisis, se captura el tráfico utilizando algún software especializado, al capturar los datos, el software almacena dicha captura en un archivo que comúnmente se denomina “captura de paquetes” o archivo PCAP. El análisis de paquetes se hace sobre paquetes individuales o con una captura completa de estos y se puede realizar sobre capturas en vivo o utilizando un archivo que contenga paquetes capturado previamente.

Los administradores de red utilizan el análisis de paquetes para obtener información sobre las condiciones actuales de la red, los analistas de seguridad utilizan el análisis de paquetes para determinar si hay algo inusual o sospechoso en el tráfico, desde el punto de vista académico se utiliza el análisis de paquetes como una herramienta de aprendizaje para comprender mejor los protocolos y en muchos casos los hackers utilizan el análisis de paquetes para rastrear el tráfico de la red con el fin de obtener información valiosa sobre la red mientras se

---

[10] ¿Qué es un sniffer y cómo puede protegerse?, Recuperado el 3 de Octubre de 2020 del sitio web de Avast Academy: https://www.avast.com/es-es/c-sniffer

[11] Sniffer, Recuperado el 3 de Octubre de 2020 del sitio web de EcuRed: https://www.ecured.cu/Sniffer

realizan búsquedas y reconocimiento, también se utiliza el análisis de paquetes para solucionar problemas de latencia, probar dispositivos de Internet de las cosas (Iot) y como herramienta para establecer una línea de base de la red.

El análisis de paquetes ha existido en el mundo de las redes durante muchos años, ya en la década de 1990, existían varias herramientas que permitían a los analistas realizar análisis de paquetes en la red para solucionar errores y monitorear el comportamiento de las máquinas dentro de una red. El análisis de paquetes ha existido durante más de 20 años como una herramienta de diagnóstico, para observar datos y otra información que viaja a través de la red, este análisis de paquetes antes también se conocía como rastreo, este término se refiere a los primeros rastreadores de paquetes, que rastreaban o capturaban el tráfico mientras viajaba por la red.

Muchos dispositivos de una red utilizan el análisis y el rastreo de paquetes, en los que se incluyen los enrutadores, conmutadores y los firewalls. A medida que los datos fluyen a través de la red, pasan a través de varios dispositivos de red, que interpretan los bits sin procesar del paquete y examinan los valores de campo en cada paquete para decidir qué acción se debe tomar.

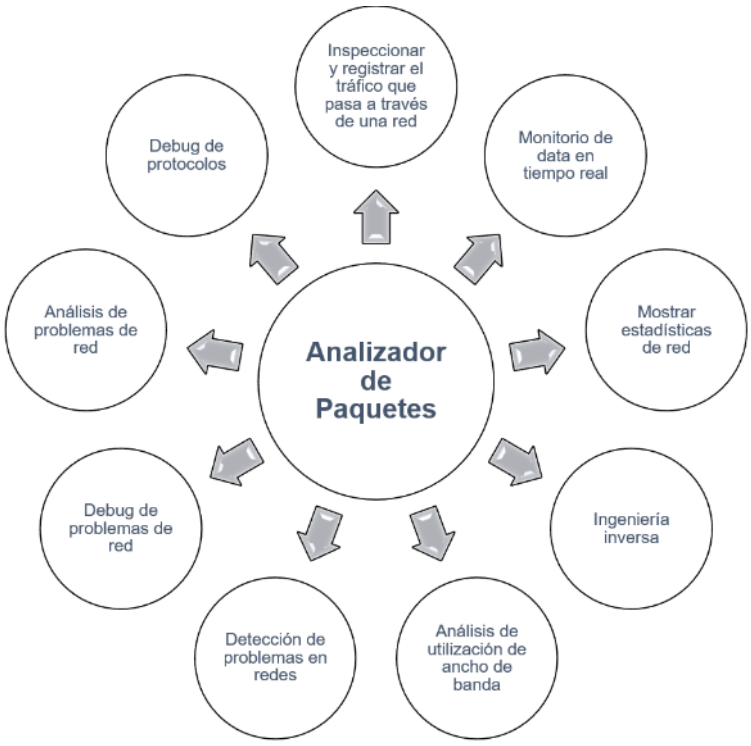
Un enrutador captura el tráfico y examina el encabezado IP para determinar a dónde se debe enviar el tráfico, como parte del proceso de enrutamiento, un dispositivo como el IDS capturará el tráfico y examinará el contenido y alertará al administrador de la red si hay algún comportamiento inusual o sospechoso, un firewall monitorea todo el tráfico y eliminará cualquier paquete que no esté en la lista de control de acceso (ACL). Es importante tener en cuenta que un analizador de paquetes detecta el tráfico, pero no modifica el contenido de ninguna manera, simplemente reúne el tráfico para analizarlo mientras viaja por la red. Los packet sniffers tienen diversos usos como monitorear redes para detectar y analizar fallos o poder realizar ingeniería inversa de protocolos de red, también es habitual su

uso para fines maliciosos, como robar contraseñas, interceptar mensajes de correo electrónico, entre otras cosas. Los principales usos de un sniffer son:

- Conversión del tráfico de red en un formato legible por los humanos.
- Análisis de fallos para descubrir problemas en la red.
- Medición del tráfico, mediante el cual es posible detectar grandes consumos de ancho de banda o poder descubrir cuellos de botella.
- Detección de intrusos, con el fin de descubrir hackers.
- Analizar los datos que se están transmitiendo por la red en tiempo real.
- Captura de contraseñas enviadas en claro y nombres de usuario de la red.
- Detectar errores de conexión o configuración.

La siguiente imagen resume los usos que se le dan a los analizadores de paquetes:

Figura 8. Usos de los analizadores de paquetes.



Fuente: Autor.

El análisis de paquetes se puede utilizar para alguno de los siguientes propósitos:

- Analizar los problemas de una red examinando los paquetes y sus encabezados.
- Detectar y analizar intentos de intrusión en la red mediante patrones de filtrado.
- Para detectar el uso indebido de la red por parte de usuarios internos o externos estableciendo reglas de firewall en su dispositivo de seguridad y luego monitoreando esas reglas.
- Para monitorear y analizar datos en movimiento mientras viajan en vivo por la red.
- Tener un mejor control sobre las categorías permitidas y restringidas de datos que viajan en la red.
- Recopilar e informar estadísticas de la red mediante el filtrado de rutas de paquetes.
- Para saber quién está en una red en tiempo real qué está haciendo (por ejemplo, se puede determinar el ancho de banda que pueda estar consumiendo o si se está intentando conectarse a sitios web restringidos) y saber si alguien está tratando de eludir las restricciones de red configuradas.
- Para depurar las comunicaciones cliente/servidor para que todas las solicitudes y respuestas comunicadas en la red puedan ser auditadas.
- Para depurar implementaciones de protocolos de red y cualquier anomalía que se genere debido a errores de configuración no intencionales o errores humanos.
- Para identificar patrones de tráfico anormales o maliciosos en la red, luego analizar, controlar, supervisar y prepararse para tales eventos.

Todos pueden beneficiarse del uso del análisis de paquetes, incluidos los desarrolladores, administradores de red, estudiantes y analistas de seguridad. Los problemas de rendimiento de las aplicaciones pueden afectar el objetivo de una

organización, especialmente en software de misión crítica, los desarrolladores se esfuerzan por producir software elegante y eficiente, antes de lanzar una aplicación, los desarrolladores ejecutan pruebas funcionales y de regresión, además realizan pruebas de stress en los servidores para garantizar una aplicación óptima. Los desarrolladores suelen probar aplicaciones en un entorno perfecto, con un gran ancho de banda y baja latencia, sin embargo, una vez que la aplicación se mueve del entorno local o de pruebas a la red de producción, los clientes pueden quejarse de los tiempos de respuesta lentos, los programadores revisan cuidadosamente la aplicación, sin embargo, muchas veces no encuentran nada inusual.

El desarrollador debe determinar las razones de los tiempos de respuesta lentos, una vez que las pruebas adicionales determinan que no es la aplicación la que está causando el problema, una herramienta de análisis de paquetes puede ayudar al desarrollador a determinar la causa raíz estos problemas de tiempos de respuesta. Al utilizar un analizador de paquetes, el desarrollador puede descubrir problemas comunes en las transmisiones, como el tiempo de ida y vuelta y detectar signos de congestión dentro de una organización, que pueden ocurrir en una red y afectar el tiempo de respuesta. Los desarrolladores entenderán que simplemente con optimizar una aplicación no es suficiente y todos los ciclos de vida del desarrollo deben incluir ver lo que está sucediendo en la red, ya que los problemas pueden afectar el rendimiento general.

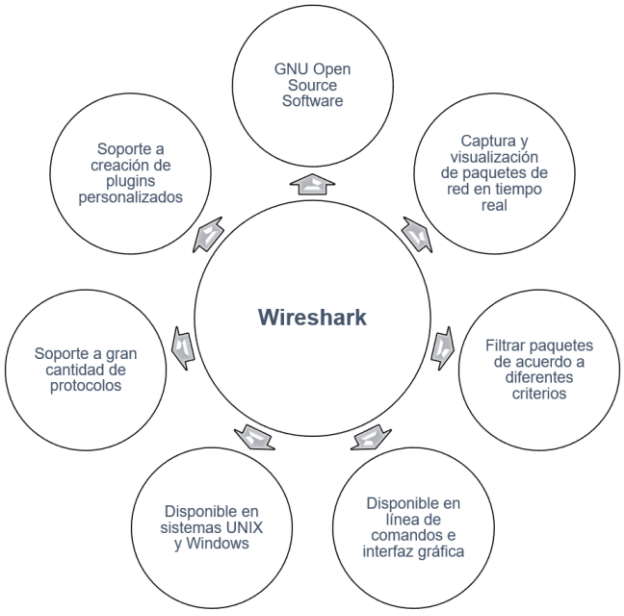
Además de los desarrolladores, los administradores de red suelen utilizar los analizadores de paquetes para solucionar problemas de la red, estos utilizan el análisis de paquetes para obtener información sobre las condiciones actuales de la red, esto puede ayudar a identificar errores o problemas en la red que pueden requerir el ajuste o reemplazo de los dispositivos físicos para mejorar el rendimiento general.

4.4.2. Wireshark. Es uno de los analizadores de red más populares, antes llamado Ethereal, permite identificar y analizar tráfico de una red en un momento determinado, se ha convertido en la nueva referencia de análisis de tráfico de red debido a todos sus módulos de decodificación, filtrado y visualización que permiten un análisis de forma simple de los paquetes capturados. [12]

Entre sus principales características se tienen:

- Permite analizar más de 480 protocolos.
- Captura directamente los paquetes de datos desde una interfaz de red.
- Obtiene información del protocolo utilizado.
- Filtra los paquetes de acuerdo a criterios establecidos por el usuario.
- Ofrece estadísticas del tráfico de una red.

Figura 9. Ventajas Wireshark.



Fuente: Autor.

[12] CHICANO, Ester. Auditoría de Seguridad Informática. IC Editorial, 2019.

A continuación, se listan algunas de las razones por las que la mayoría de los profesionales prefieren usar Wireshark en comparación con otros rastreadores de paquetes:

- Fácil de usar: la interfaz de Wireshark es fácil de usar y comprender, las herramientas y funciones están muy bien organizadas y representadas.
- Solidez: Wireshark es capaz de manejar con facilidad enormes volúmenes de tráfico de red.
- Independiente de la plataforma: Wireshark está disponible para diferentes tipos de sistemas operativos, ya sea Windows, Linux o Macintosh.
- Filtros: hay dos tipos de opciones de filtrado disponibles en Wireshark, filtros de captura y filtros de visualización.
- Costo: Wireshark es un analizador de paquetes de código abierto y gratuito desarrollado y mantenido por una comunidad dedicada de profesionales.
- Soporte: Wireshark está siendo desarrollado continuamente por un grupo de colaboradores que se encuentran dispersos por todo el mundo, las personas se pueden suscribir a la lista de correo de Wireshark o se puede obtener ayuda de la documentación en línea, a la que se puede acceder a través de la propia GUI, también existen foros en línea disponibles para que se obtenga la ayuda más eficaz.

4.4.2.1. Funcionamiento de Wireshark. Wireshark recopila el tráfico de red a través de la interfaz de red del computador, ejecutándose en modo promiscuo para inspeccionar y mostrar información relacionada con protocolos, direcciones IP, puertos, encabezados y longitud del paquete. Wireshark contiene la librería Winpcap/libcap que permite que la interfaz de red NIC se ejecute en modo promiscuo; el único momento en el que no tiene que rastrear en modo promiscuo es cuando los paquetes son generados directamente o intencionalmente hacia y/o desde el propio dispositivo. Hay tres procesos que sigue cada analizador de protocolos: recopilar, convertir y analizar, que se describen a continuación:

- Recopilar: Escuchar el tráfico y capturar paquetes de red de una interfaz de red específica.
- Convertir: Aumentar la legibilidad de los datos que no son legibles por humanos. Los paquetes se convierten en información fácilmente comprensible que se visualizan a través de una GUI.
- Analizar: Analizar el tráfico de la red relacionado con los paquetes, protocolos, datos sin procesar y más mediante el uso de funciones estadísticas y gráficas.

4.5. SEGURIDAD EN APLICACIONES WEB

Cada día es más habitual el uso de aplicaciones web en empresas y gobiernos, su fácil implementación y uso han hecho que sean prácticamente esenciales en todos los comercios. El crecimiento de internet ha impactado directamente en la seguridad de la información, sitios de comercio electrónico, bancos, servicios, redes sociales, entre otros, contienen información sensible y crítica considerada de gran importancia. Actualmente, en gran medida la reputación de muchas empresas está en manos de la aplicación web utilizada, las complejas funcionalidades de las aplicaciones web han movido el perímetro de seguridad de

las organizaciones, los privilegios de acceso a funcionalidades y datos ya no son uniformes y abiertos, sino que requieren de complejos esquemas.

Desde un punto de vista de un administrador de un sitio web, la seguridad de una aplicación web debe ser abordada durante todo el ciclo de vida del proyecto, desde la fase de diseño, hasta la puesta en producción, incluyendo aspectos técnicos, herramientas, metodologías y sin dejar de lado el factor humano.

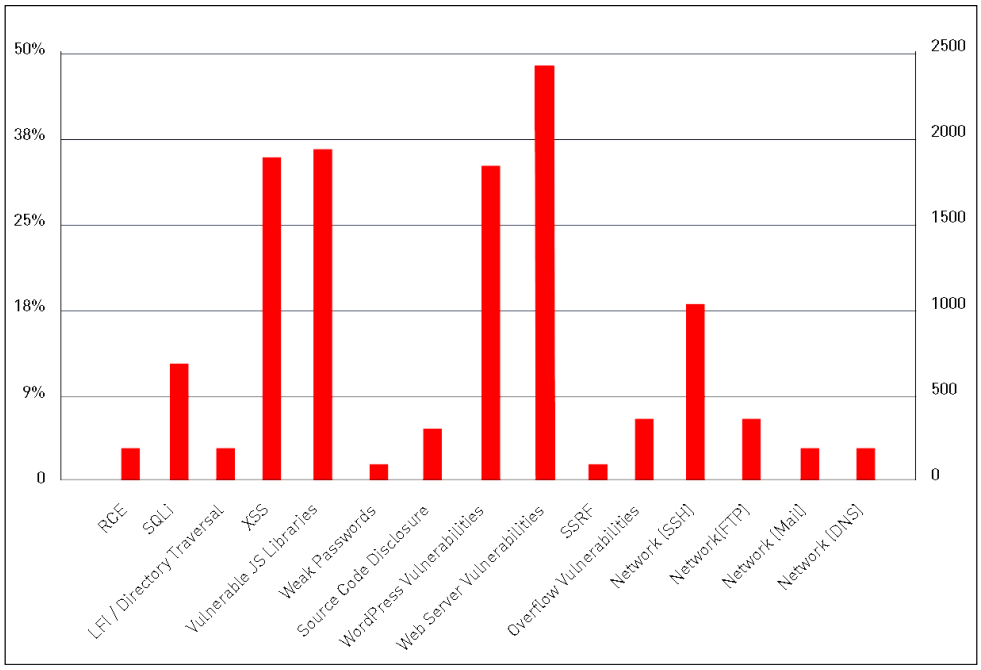
Al igual que con cualquier tecnología nueva, las aplicaciones web vienen acompañadas con una gran variedad de amenazas a la seguridad. El conjunto de los defectos que se detectan con mayor frecuencia ha evolucionado a lo largo del tiempo, nuevos ataques están concebidos de forma tal que resulta casi imposible considerarlos al momento del desarrollo de las aplicaciones, sumado al surgimiento de nuevas tecnologías que introducen nuevas posibilidades de explotación.

Diferentes fuentes como Acunetix[13], compilan en un informe anual, las vulnerabilidades encontradas sobre más de 10.000 aplicaciones web, con el propósito de proporcionar un análisis respecto a:

- ¿Qué vulnerabilidades están aumentando o disminuyendo y en qué frecuencia?
- Preocupaciones de seguridad actuales.
- Hallazgos de nuevos tipos de vulnerabilidades.
- Análisis de cada vulnerabilidad descubierta.

[13] Acunetix Web Application Vulnerability Report 2019, Recuperado el 10 de Octubre de 2020 del sitio web de Acunetix: https://www.acunetix.com/acunetix-web-application-vulnerability-report/

Figura 10. Vulnerabilidades en Aplicaciones Web Con Severidad Alta.



Fuente: Acunetix Web Application Vulnerability Report 2019.

A continuación, se listan las vulnerabilidades con severidad alta encontradas en el informe, este nivel indica que un atacante puede comprometer de forma completa la confidencialidad, integridad o disponibilidad de un sistema objetivo:

- Ejecución Remota de Código (RCE) (2%).
- SQL Injection (14%).
- Inclusión de Archivos Locales y Recorrido de Directorio (LFI/Directory Traversal) (3%).
- Cross-site Scripting (XSS) (32%).
- Vulnerabilidades en Librerías JS (33%).
- Contraseñas Inseguras (1%).
- Divulgación de Código Fuente (4%).
- Falsificación de Solicitudes del Lado del Servidor (SSRF) (1%).
- Vulnerabilidades de Overflow (5%).

- Vulnerabilidades Wordpress (30%).
- Vulnerabilidades Servidores Web y Errores de Configuración (47%).
- SSH (19%).
- FTP (6%).
- Mail (2%).
- DNS (2%).

Uno de los puntos críticos de la seguridad en la web son las herramientas con las que el usuario interactúa de forma directa, en este caso los servidores web. Es común escuchar sobre vulnerabilidades en los sistemas de protección de los servidores web más frecuentemente utilizados, por ejemplo, Apache, NGINX, IIS, etc., o en los lenguajes de programación en las que son escritas las aplicaciones.

Sin embargo, la mayoría de los problemas detectados en aplicaciones web no son provocados por fallas de algunas de estas partes mencionadas anteriormente, si no que los problemas se generan debido a las malas prácticas de parte de los programadores. Se debe entender que desarrollar aplicaciones web seguras no es una tarea trivial, ya que requiere por parte del programador, no sólo cumplir con los requerimientos de la aplicación, sino que es necesario tener una concepción general de los riesgos que puede correr la información procesada por el sistema.

A la hora de construir aplicaciones web la seguridad es por lo general, un aspecto obviado por la mayor parte de programadores, planificar los mecanismos necesarios para evitar accesos no autorizados a las aplicaciones se convierte en algo que todo programador debería implementar correctamente.

Al momento de construir una aplicación web especialmente sitios con fines comerciales, es imprescindible tener una traza de los pasos que realiza cada usuario y poder controlar que tipos de acciones puede realizar un usuario concreto.

Aunque el tema de la creación de sitios web seguros es muy amplio y requiere realizar un estudio bastante minucioso para comprender los puntos vulnerables, existen ciertas medidas básicas que se deberían adoptar para proteger cualquier aplicación web, la siguiente lista proporciona algunas pautas de seguridad mínimas que se deberían seguir:[14]

- Ejecutar aplicaciones con privilegios mínimos: No se debe ejecutar una aplicación con la identidad de un usuario administrador del sistema, las aplicaciones web se deben ejecutar en el contexto de un usuario con los mínimos privilegios factibles, se deben establecer permisos en todos los recursos requeridos por la aplicación utilizando la configuración menos permisiva posible, no se debe dar la opción a los usuarios a especificar rutas que permitan tener acceso a ningún archivo de la aplicación.

- Protegerse contra entradas malintencionadas: Como regla general nunca se debe dar por sentado que la entrada ingresada por parte del usuario es segura, se deben filtrar todas las entradas de los usuarios, no se deben mostrar las entradas de los usuarios sin filtrar, antes de mostrar cualquier tipo de información, se deben codificar los elementos para evitar la ejecución de scripts malintencionados, no se debe almacenar información sin filtrar en una base de datos, se deben proteger las cadenas de consulta, cookies y si es posible no se debe almacenar información confidencial en campos ocultos o cookies.

- Crear mensajes de error seguros: Si no se es cuidadoso en la forma de mostrar los mensajes de error, un usuario malintencionado puede deducir

[14] Procedimientos de seguridad básicos para Aplicaciones Web, Recuperado el 5 de Octubre de 2020 del sitio web de Microsoft Developer Network: https://msdn.microsoft.com/es-es/library/zdh19h94(v=vs.100).aspx

información importante sobre la aplicación a partir de los mensajes de error, se debe configurar la aplicación para que no se muestren errores detallados a los usuarios, en lo posible se debe crear un sistema de administración de errores personalizados para las acciones que sean propensas a errores como el acceso a archivos o a bases de datos.

- Mantener segura la información confidencial: Si la aplicación web transmite información sensible, se debe plantear el uso del protocolo SSL, se deben utilizar los algoritmos de cifrado de alta seguridad.

Antes del surgimiento de las aplicaciones web, las organizaciones se esforzaban en asegurarse a sí mismas contra ataques externos enfocándose principalmente sobre el perímetro de la red; defender este perímetro significaba fortalecer los servicios que se exponían y filtrar los accesos hacia los demás. Las aplicaciones web han evolucionado y para que una aplicación sea accesible por los usuarios, el filtrado perimetral debe permitir conexiones entrantes hacia el servidor y para que la aplicación funcione, el servidor debe tener permitido conectarse con los sistemas que soportan la funcionalidad del backend; generalmente estos sistemas soportan operaciones de misión crítica para la organización y residen detrás de varias capas de defensas a nivel de seguridad en la infraestructura de comunicaciones. Si existe una vulnerabilidad dentro de una aplicación web sobre Internet, entonces un atacante puede ser capaz de comprometer los sistemas esenciales de la organización simplemente mediante el envío de datos desde su navegador web, estos datos atravesarán todas las defensas de la infraestructura de comunicaciones de la organización, de la misma manera que lo hace el tráfico normal hacia la aplicación web. El perímetro de seguridad de una organización se ha movido, parte de ese perímetro aún se encuentra en los firewalls y similares pero una parte significativa del mismo ahora está ocupado por las aplicaciones web de la organización. Debido a las múltiples formas en las que las aplicaciones web reciben datos por medio de las entradas del usuario y las pasan hacia

sistemas sensibles, son las potenciales puertas de entrada de un amplio rango de ataques, por tal motivo las defensas contra los mismos se deben implementar dentro de las propias aplicaciones. Un simple error en alguna validación de una única aplicación web puede convertir en vulnerables los sistemas internos de una organización.

Otra forma en la que las aplicaciones web han movido su perímetro de seguridad surge de las amenazas que los propios usuarios enfrentan cuando acceden a una aplicación vulnerable. Una atacante puede hacer uso de una aplicación web vulnerable para atacar a cualquier usuario que la visite, si el usuario ingresa desde una intranet organizacional, el atacante puede lanzar un ataque contra la red interna desde la posición de confianza del usuario pudiendo ser capaz de realizar cualquier acción que pudiera ejecutar el usuario, si fuera malicioso. Más a menudo los administradores de red están familiarizados con la idea de impedir que los usuarios visiten sitios web maliciosos y los mismos usuarios gradualmente están adquiriendo mayor conciencia de este tipo de amenazas, pero no se puede decir lo mismo en relación al desarrollo e implementación de aplicaciones web y la implicancia que una aplicación vulnerable puede representar para sus usuarios y para la organización a la que pertenecen siendo una amenaza no menor que un sitio web que es claramente malicioso.

4.6. SEGURIDAD EN BASES DE DATOS

Las bases de datos son el almacén de datos de uso diario para prácticamente todo, estas se encuentran en todas partes, almacenan datos médicos, datos bancarios, datos de antecedentes penales, datos geográficos, etc., no hay ninguna empresa que no haga uso de algún modo de una base de datos, por lo que se les considera como una pieza clave para cualquier atacante.

Las bases de datos son siempre parte fundamental de los sistemas de información donde casi siempre encontramos un aplicativo web, es por esta razón que ambos están íntimamente interrelacionados y cuando se piensa en seguridad no se pueden considerar dichos elementos como asilados. Más allá del tipo de datos almacenados en la base de datos, las aplicaciones o servicios web en sí mismos suponen una puerta de entrada para un potencial atacante, un servicio con una pobre seguridad puede ser considerado como una amenaza para la organización.

La seguridad de las bases de datos heredan las mismas dificultades de seguridad a las que se enfrenta la información, que es garantizar la integridad, la disponibilidad y la confidencialidad. El acceso no autorizado, la pérdida o publicación de los datos privados de una entidad, suponen un daño irreparable para la reputación de una organización, ante una pérdida de datos se pueden adoptar medidas y precauciones como copias de seguridad, pero el simple acceso a la información clave de una organización puede suponer una pérdida irreparable.

Los Sistemas Gestores de Bases de Datos SGBD implementan mecanismos que restringen o permiten accesos a los datos de acuerdo con los perfiles o roles suministrados por el administrador, a modo general el SGBD verifica todas las solicitudes de acceso, comparándolas con las restricciones de seguridad almacenadas en el catálogo del sistema, sin embargo existen brechas en el sistema y amenazas externas que pueden comprometer a un servidor de base de datos, estas pueden resultar en la pérdida o degradación de algunos de los objetivos de seguridad aceptados como los son: la integridad, la disponibilidad, o confidencialidad. La integridad se refiere al requisito de que la información esté protegida contra modificaciones impropias, la disponibilidad se refiere a hacer que los objetos estén disponibles a un usuario al cual estos tienen un derecho legítimo y la confidencialidad se refiere a la protección de datos contra la exposición no autorizada. El impacto en la degradación o pérdida de cualquiera de los anteriores

objetivos de seguridad puede dar como resultado en la pérdida de confianza pública, vergüenza o acciones legales contra la organización [15].

La preocupación con la creación y mantenimiento de entornos seguros es una de las principales preocupaciones de los administradores de redes, de sistemas operativos y de bases de datos. Las investigaciones muestran que la mayoría de los ataques, robos de información y accesos no autorizados los realizan personas que pertenecen a la propia organización. Los controles de acceso en sistemas de información deben certificar que todos los accesos directos al sistema sucedan exclusivamente de acuerdo a las modalidades y las reglas preestablecidas y observadas por las directivas y políticas de protección.

4.6.1. Inyección SQL. Las aplicaciones web en general suelen presentar defectos, errores de programación, algunos de los cuales pueden afectar su seguridad, esto no es nada nuevo, ni debería ser extraño para nadie, dado el gran nivel de complejidad que con frecuencia se maneja en estas aplicaciones, cuando aumenta el número de líneas de código, es fácil que existan errores y en muchos casos estos sean difíciles de detectar y del mismo modo difíciles para darles solución.

La mayoría de aplicaciones y sitios web desarrollados hoy en día hacen uso de una base de datos para almacenar información tanto de la propia aplicación como de los usuarios, datos a los que comúnmente se acceden por medio del lenguaje de bases de datos relacionales SQL. Está característica común ha traído consigo la aparición a las vulnerabilidades que tienen relación con ataques por inyección SQL, vulnerabilidad que pone en riesgo la propia aplicación y los datos almacenados.

---

[15] GALLARDO, Gabriel. Seguridad en Bases de Datos y Aplicaciones Web: 2ª Edición. IT Campus Academy, 2016. 18 p.

El ataque de inyección SQL es posible debido a ciertas características del lenguaje SQL como las siguientes:

- Se pueden colocar comentarios en una sentencia SQL.
- Se pueden escribir varias sentencias SQL juntas y se pueden ejecutar en bloque.
- Se pueden realizar consultas a metadatos almacenados en tablas del sistema.

Un ataque de inyección SQL consiste en la inserción o "inyección" de consultas SQL por medio de los datos de entrada desde el cliente hacia la aplicación. Un ataque por inyección SQL exitoso puede leer datos sensibles de una base de datos, modificar datos, ejecutar operaciones de administración sobre la misma y en algunos casos emitir comandos al sistema operativo. En los ataques por inyección los comandos SQL son insertados en la entrada de datos con la finalidad de efectuar la ejecución de comandos SQL predefinidos.

Los ataques de inyección SQL permiten a los atacantes suplantar identidades, alterar datos existentes, causar problemas de repudio como anular transacciones, permite la revelación de datos de un sistema, destruir los datos o volverlos inasequibles y convertirse en administradores del servidor de base de datos e incluso acceder a datos almacenados en otras bases de datos almacenados en el mismo servidor.

La inyección SQL es muy común con aplicaciones implementadas en lenguajes de programación o frameworks de desarrollo un poco antiguos como PHP ó ASP. Debido a la naturaleza de las interfaces programáticas disponibles, las aplicaciones J2EE y ASP.NET tienen menor probabilidad de ser fácilmente atacadas por una inyección SQL. La gravedad de una inyección SQL está limitada por la habilidad e imaginación del atacante y en general, se considera a la inyección SQL como un ataque de alto impacto.

La inyección se caracteriza cuando el atacante, incluyendo usuarios internos, externos y administradores, envían datos no confiables al sistema, es decir, sin el tratamiento adecuado, estos datos que en realidad se tratan de cadenas de texto que forman consultas “queries”, llegan hasta el sistema y logran alcanzar algún interpretador de comandos.

Según el proyecto OWASP, la clasificación del riesgo está encuadrado de la siguiente forma: el vector de ataque se considera fácil ya que puede ser constituido por cualquier fuente de datos. La detección se considera media porque es fácil encontrarla cuando se hace una verificación de código fuente de la aplicación, sin embargo, es más difícil a través de pruebas. Los scanners y fuzzers podrán ayudar a los atacantes a encontrarlas. El impacto para el negocio es severo ya que puede, por ejemplo, perjudicar a toda la base de datos y puede también dar acceso total del sistema al atacante. La siguiente tabla sintetiza la clasificación del riesgo.

Tabla 3. Clasificación del riesgo

Vector de Ataque	**Vulnerabilidad de Seguridad**		**Impacto Técnico**
Exploración	Predominio	Detección	
Fácil	Común	Medio	Severo

Fuente: Autor

Todo formulario web puede servir como puerta de entrada “una vulnerabilidad” para el ataque de Inyección de SQL. Es muy común que este ataque suceda en la pantalla de login, ya que este es el primer formulario del sistema y normalmente está más expuesto que los demás formularios. Pero esto no significa que los formularios internos “posteriores a la pantalla de login” del sistema no necesiten prevención.

Es importante recordar que el atacante puede ser externo (probablemente atacando la pantalla de login) o interno (usuario del sistema mal intencionado). El

atacante intentará, a través de varias tentativas, descubrir la estructura de la base de datos, por eso es importante que los nombres de los campos de los formularios no concuerden con los nombres de los campos de la base de datos.

Es necesario que el atacante realice tentativas de acceso para conocer la estructura de la base de datos. Esta tarea se hace más fácil cuando los nombres de las variables usadas en el formulario HTML se usan en la estructura de la base de datos, finalmente el código HTML es legible para los usuarios de la Web.

Mediante la inyección SQL un atacante podría realizar alguna de las siguientes acciones contra el sistema:

- Descubrimiento de información: Un atacante puede acceder a registros u objetos de la base de datos a los que inicialmente no tenía acceso.
- Elevación de privilegios: Un atacante puede acceder a los identificadores de usuarios más privilegiados y cambiar las credenciales.
- Denegación de servicio: La modificación de comandos SQL puede llevar a la ejecución de acciones destructivas como el detenimiento de servicios o borrado de datos u objetos, de igual forma se pueden ejecutar comandos que requieran gran cantidad de tiempo de máquina para colapsar el motor de base de datos o hacer que el servicio no responda en tiempos adecuados.
- Suplantación de usuarios: Un atacante al poder acceder al sistema de credenciales puede obtener las credenciales de otro usuario y realice acciones con la identidad robada.

Las principales consecuencias de los ataques de inyección SQL son:

- Autenticación: Mediante este ataque es posible conectarse a un sistema o aplicación como otro usuario sin conocimiento previo de la contraseña.
- Autorización: Mediante este ataque es posible cambiar datos de autorización que son almacenados en las bases de datos.

- Confidencialidad: Debido a que las bases de datos generalmente almacenan datos sensibles y la pérdida de estos genera una pérdida de confiabilidad.
- Integridad: Mediante este ataque es posible modificar datos sensibles o incluso eliminarlos.

Existen diferentes tipos de ataques de inyección SQL, estos se listan a continuación:

- Inyección SQL por error “Error-Based SQL Injection”: Este tipo de ataque se aprovecha de que las aplicaciones van mostrando los errores que da la base de datos, con este error se consigue información valiosa referente a “estructuras, tablas, nombre de campos e incluso datos.” Es el tipo más simple.
- Inyección SQL por UNION “Union-Based SQL Injection”: Es el tipo más popular de inyección SQL, este ataque utiliza el operador UNION para poder obtener datos de la base de datos, este operador combina los resultados de dos o más sentencias SELECT en un único resultado que es retornado como parte de la respuesta.
- Inyección SQL ciega “Blind SQL Injection”: Este es un tipo de ataque más avanzado, se da cuando las aplicaciones no muestran ningún mensaje de error al ejecutar una sentencia SQL errónea, por lo que el atacante va realizando más ataques hasta dar con datos con los cuales pueda actuar. Es el tipo de ataque más difícil, se extraen los datos realizando preguntas a la base de datos. Esta técnica se utiliza en combinación de diccionarios y fuerza bruta para la búsqueda carácter por carácter de una contraseña, nombre de usuario o cualquier dato que albergue la base de datos. Este ataque se subdivide en dos tipos, la inyección SQL basada en tiempo y la inyección SQL basada en booleanos o contenido. La inyección SQL basada en tiempo se fundamenta en poder pausar la base de datos por un tiempo especificado, para que posteriormente devuelva los resultados.

4.6.2. SQLMap. Es una herramienta de código abierto para realizar inyección de código SQL de forma automatizada “detección y explotación” en aplicaciones web, está desarrollado en el lenguaje de programación Python, tiene soporte completo para los DBMS: MySQL, Oracle, PostgreSQL, Microsoft SQL Server, Microsoft Access, IBM DB2, SQL Lite, Firebird, Sybase, SAP MaxDB, HSQLDB e Informix.

Soporta las técnicas boolean based blind inyection, time based blind inyection, error based inyection, UNION query based inyection, stacked queries y out of band inyection.

Tiene soporte para enumerar usuarios, privilegios, roles, bases de datos, tablas y columnas, tiene reconocimiento automático de los formatos password hash y soporte para poder descifrar usando ataques basados en diccionario.
Ofrece la posibilidad de realizar vaciados completos de las tablas de una base de datos, tiene soporte para hacer búsquedas de datos específicos en tablas y columnas.

Tiene soporte para descargar y cargar cualquier archivo del sistema de archivos subyacente del servidor de base de datos, puede ejecutar comandos arbitrarios y trabajar con la salida estándar del sistema operativo subyacente del servidor de base de datos.

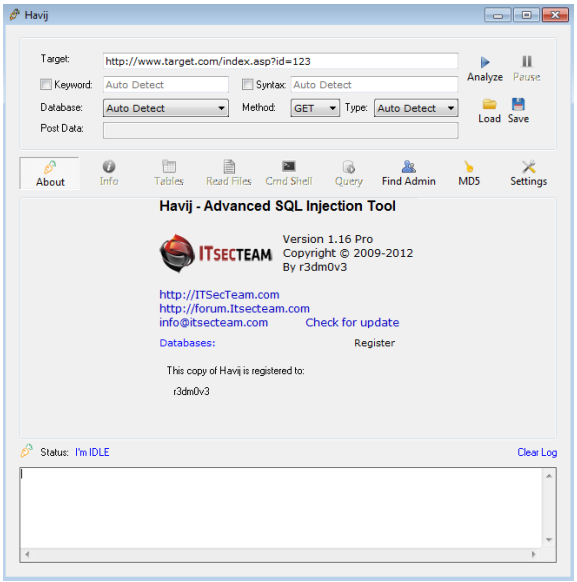
Los comandos que posee sqlmap se agrupan en las siguientes categorías: target, request, optimization, injection, detection, techniques, fingerprint, enumeration, brute force, user-defined function injection, file system access, operating system access, windows registry access, general y miscellaneous.

Tabla 4. Sqlmap, herramienta de inyección SQL.

Fuente: Sitio web sqlmap http://sqlmap.org/

4.6.3. Havij. Es una herramienta automática que facilita la explotación de vulnerabilidades de inyección SQL en aplicaciones web, es de fácil uso, permite hacer fingerprint de las bases de datos, se pueden obtener datos de las tablas, se pueden ejecutar sentencias SQL y algunas veces también es posible acceder al sistema de ficheros y ejecutar comandos en el sistema operativo. Es una de las herramientas más populares en entornos Windows por su facilidad de uso. Es desarrollada por la compañía iraní de seguridad ITSecTeam.

Figura 11. Havij. Herramienta de inyección SQL.



Fuente. Autor.

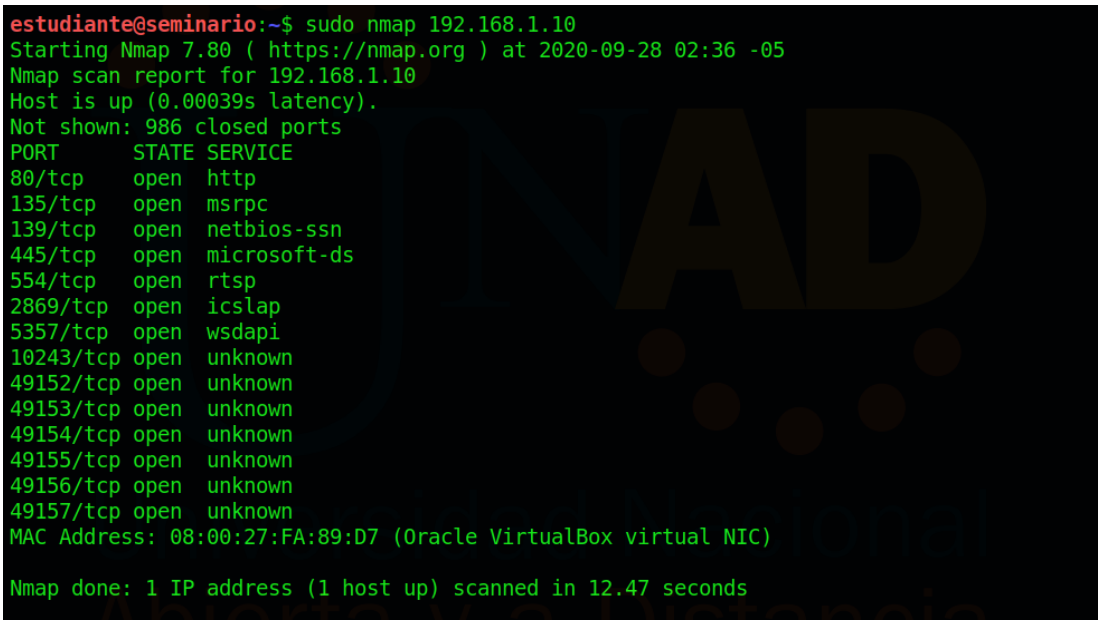
5. DESARROLLO

5.1. PRÁCTICAS NMAP

Escaneo de una Dirección IP o escaneo por defecto

Por defecto Nmap utiliza un escaneo del tipo SYN. En el escaneo por defecto la columna SERVICE se obtiene de acuerdo a la especificación de puertos de Linux "/etc/services", en lugar de analizar profundamente el protocolo.

Figura 12. Nmap escaneo por defecto. nmap 192.168.1.10



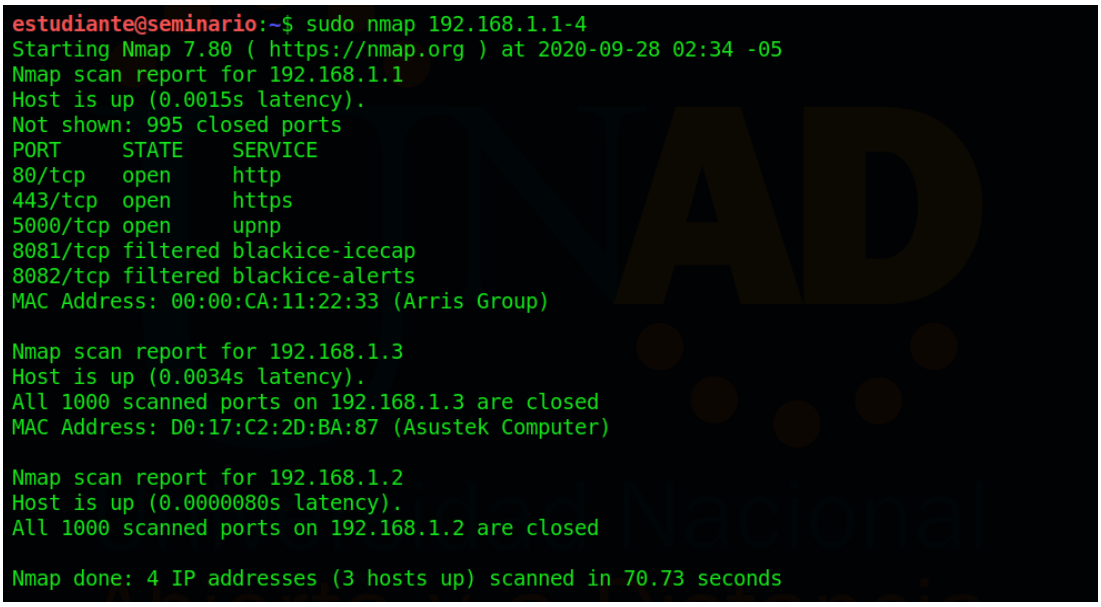
Fuente. Autor.

Escaneo de Rangos de Direcciones IP

Muchas veces se necesita escanear más de una máquina, Nmap permite realizar escaneos de rangos de direcciones IP, a continuación, se ilustran algunos ejemplos.

Escanear desde la dirección IP 192.168.1.0 a la 192.168.1.4 es decir las direcciones 192.168.1.1, 192.168.1.2, 192.168.1.3 y 192.168.1.4

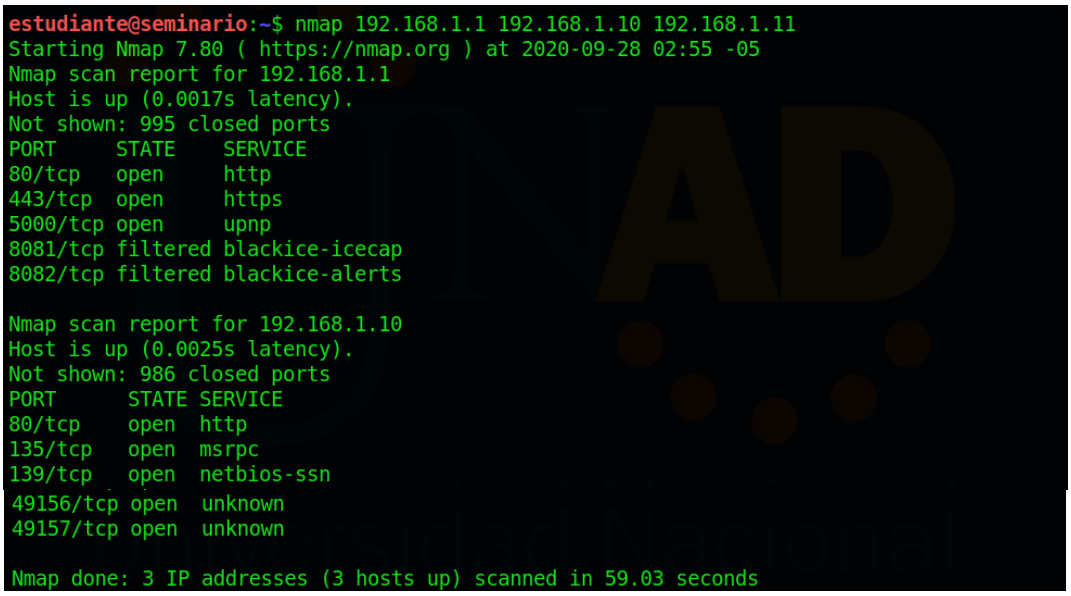
Figura 13. Escaneo de direcciones IP por rango. nmap 192.168.1.1-4



Fuente. Autor.

Escanear múltiples direcciones IP: 192.168.1.1, 192.168.1.10, 192.168.1.11

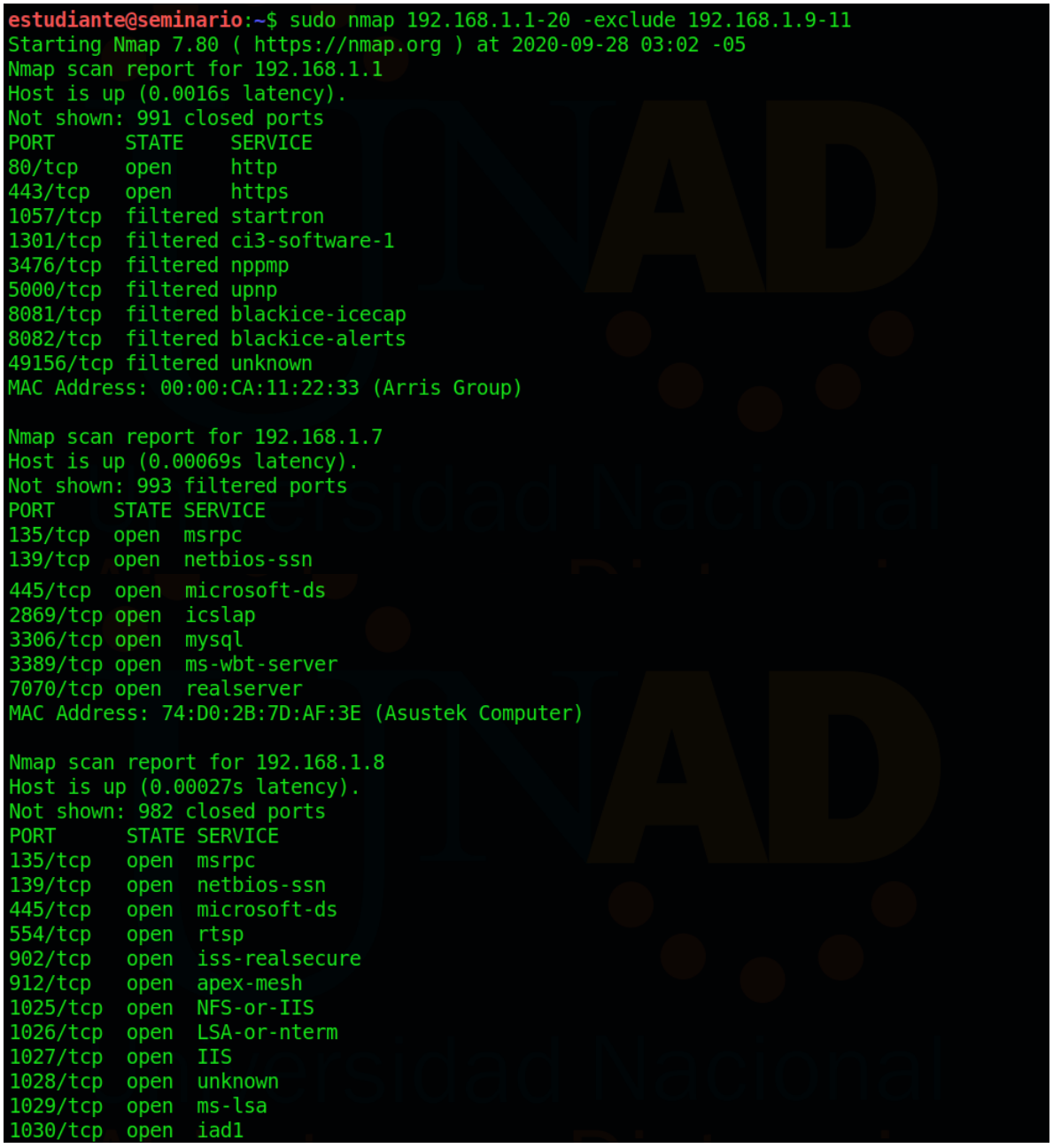
Figura 14. Escaneo de múltiples direcciones IP separadas por comas. nmap 192.168.1.1 192.168.1.10 192.168.1.11



Fuente. Autor.

Escanear desde la dirección IP 192.168.1.1. a la 192.168.1.20 excluyendo el rango de direcciones 192.168.1.9, 192.168.1.10, 192.168.1.11

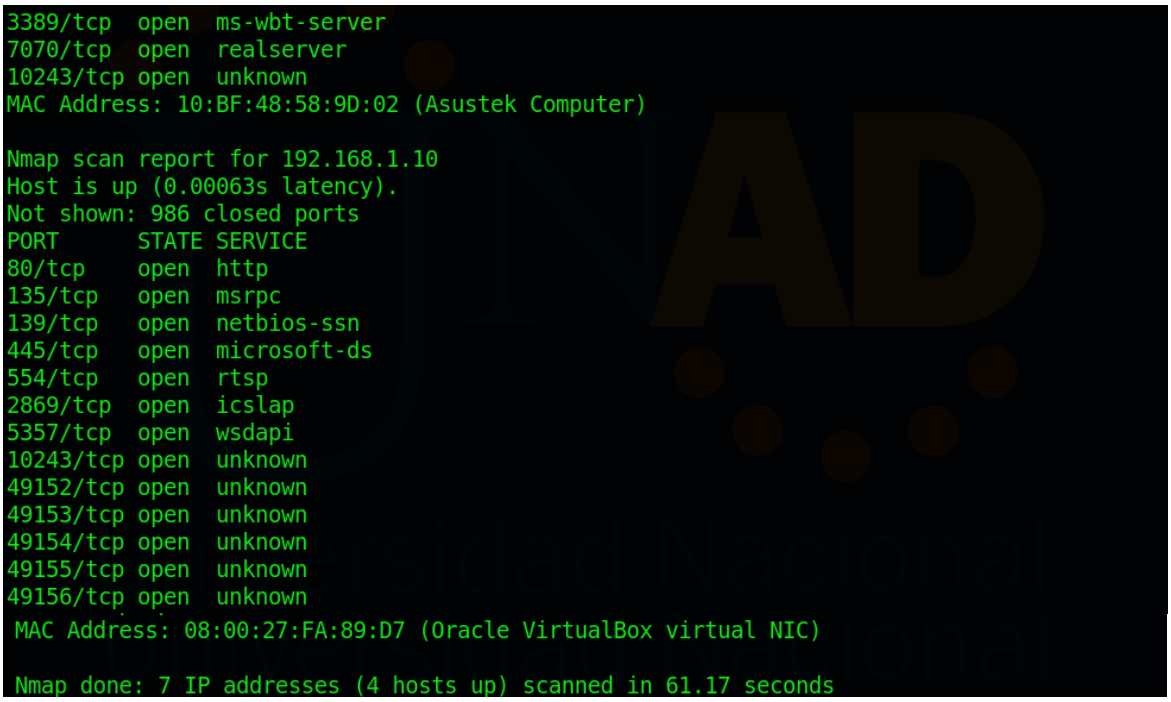
Figura 15. Escaneo de múltiples direcciones IP, excluyendo un rango. nmap 192.168.1.1-20 –exclude 192.168.1.9-11



Fuente. Autor.

Escanear desde la dirección IP 192.168.1.1 a la 192.168.1.10 excluyendo las direcciones 192.168.1.2, 192.168.1.5 y 192.168.1.9

Figura 16. Escaneo de múltiples direcciones IP, excluyendo una lista de direcciones IP separadas por coma.

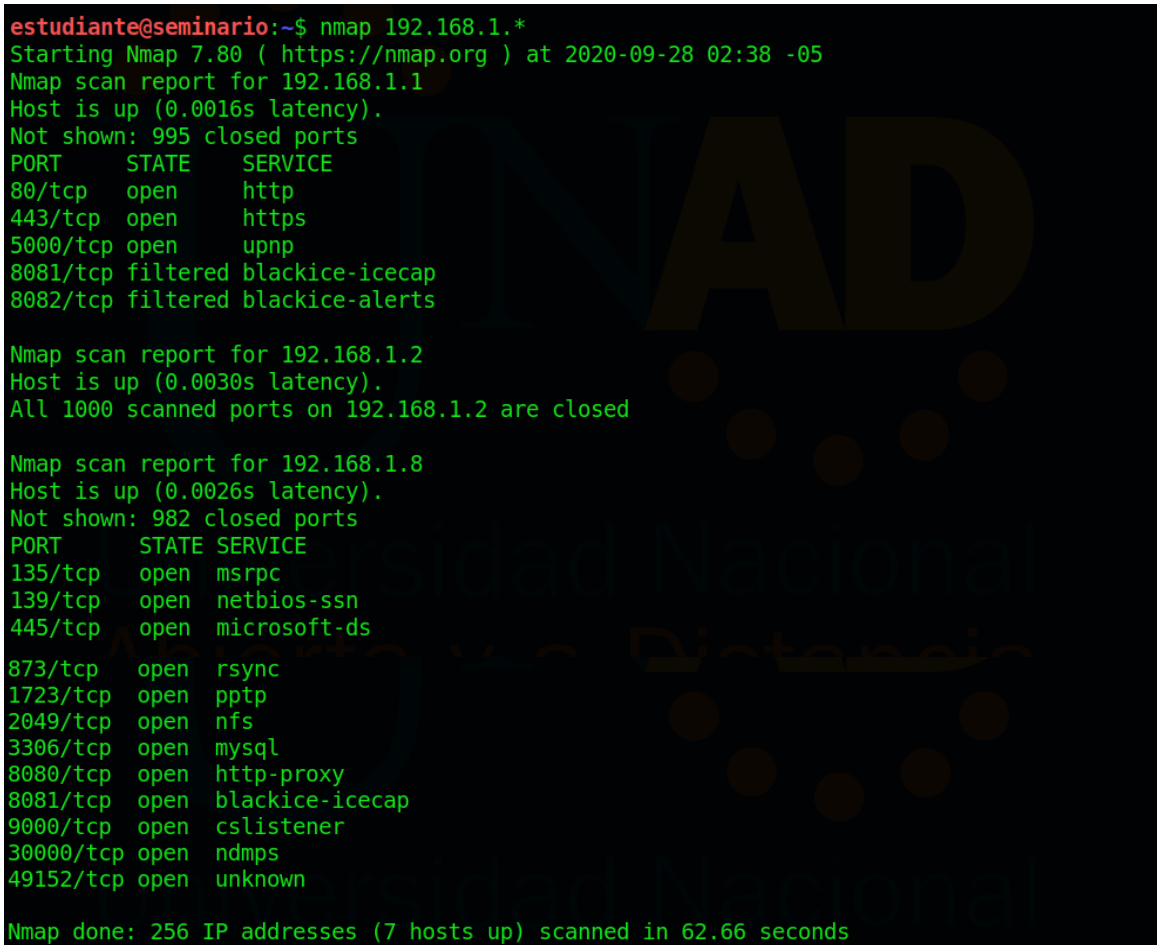


Fuente. Autor.

Otra forma de realizar escaneos de rangos de direcciones IP's es utilizando la notación "Octect Range", con esta notación en lugar de especificar una lista de direcciones IP separadas por comas, se pueden utilizar rangos específicos de direcciones.

A continuación, se ilustra un ejemplo para escanear todas las direcciones IP del rango 192.168.1 utilizando la notación "Octect Range".

Figura 17.Escaneo de direcciones IP por rango con notación "Octate Range". Nmap 192.168.1.1.*.



Fuente. Autor.

Notación CIDR

La notación CIDR “Classless Inter-domain Routing” es un método compacto para especificar direcciones IP junto con un sufijo de enrutamiento, esta notación se especifica mediante una dirección IP y un sufijo de red, este sufijo representa el número de bits de red, las direcciones IPV4 son de 32 bits, por lo que la red puede estar entre 0-32. Los sufijos más comunes son 8, 16, 24 y 32.

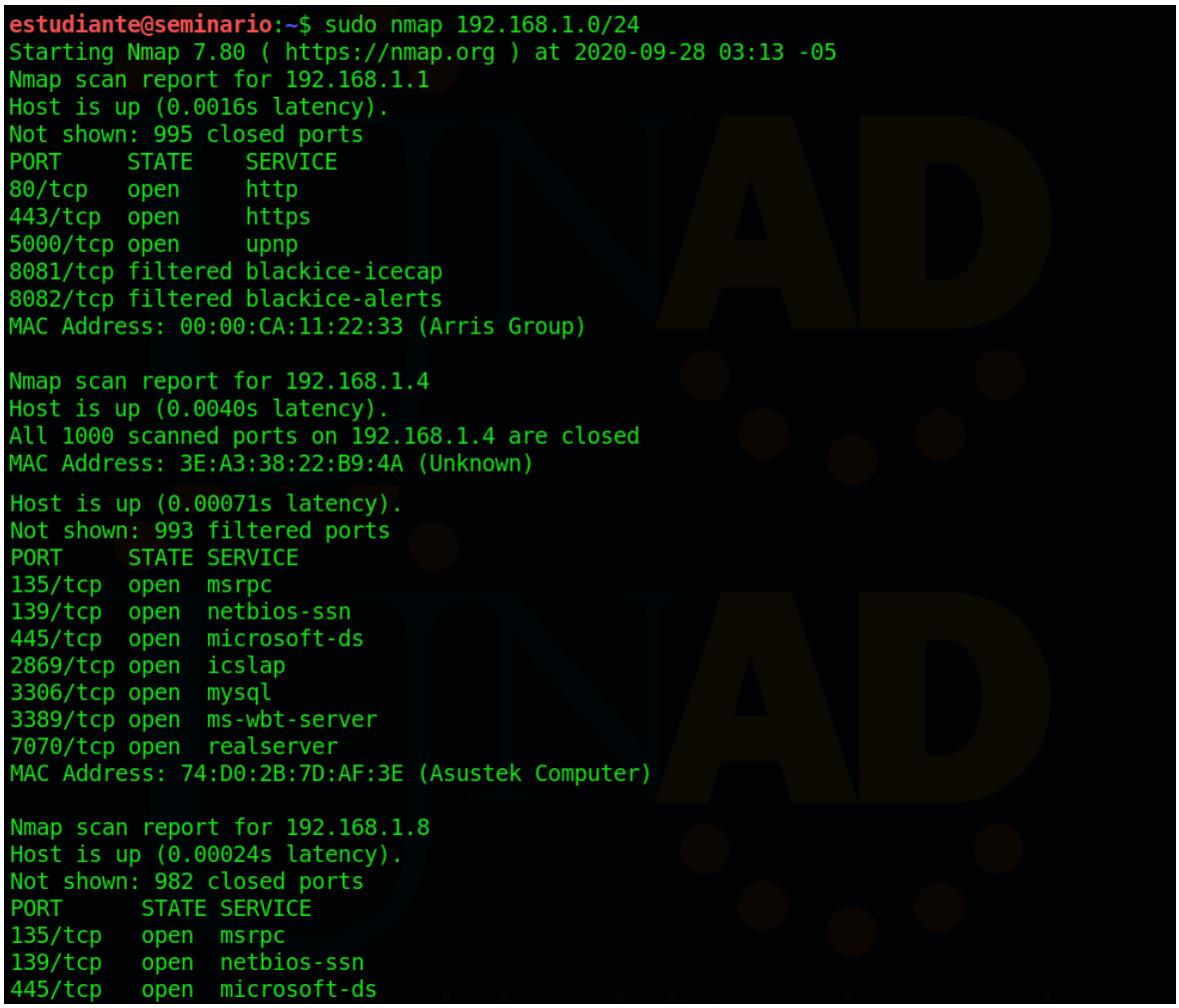
Tabla 5. CIDR

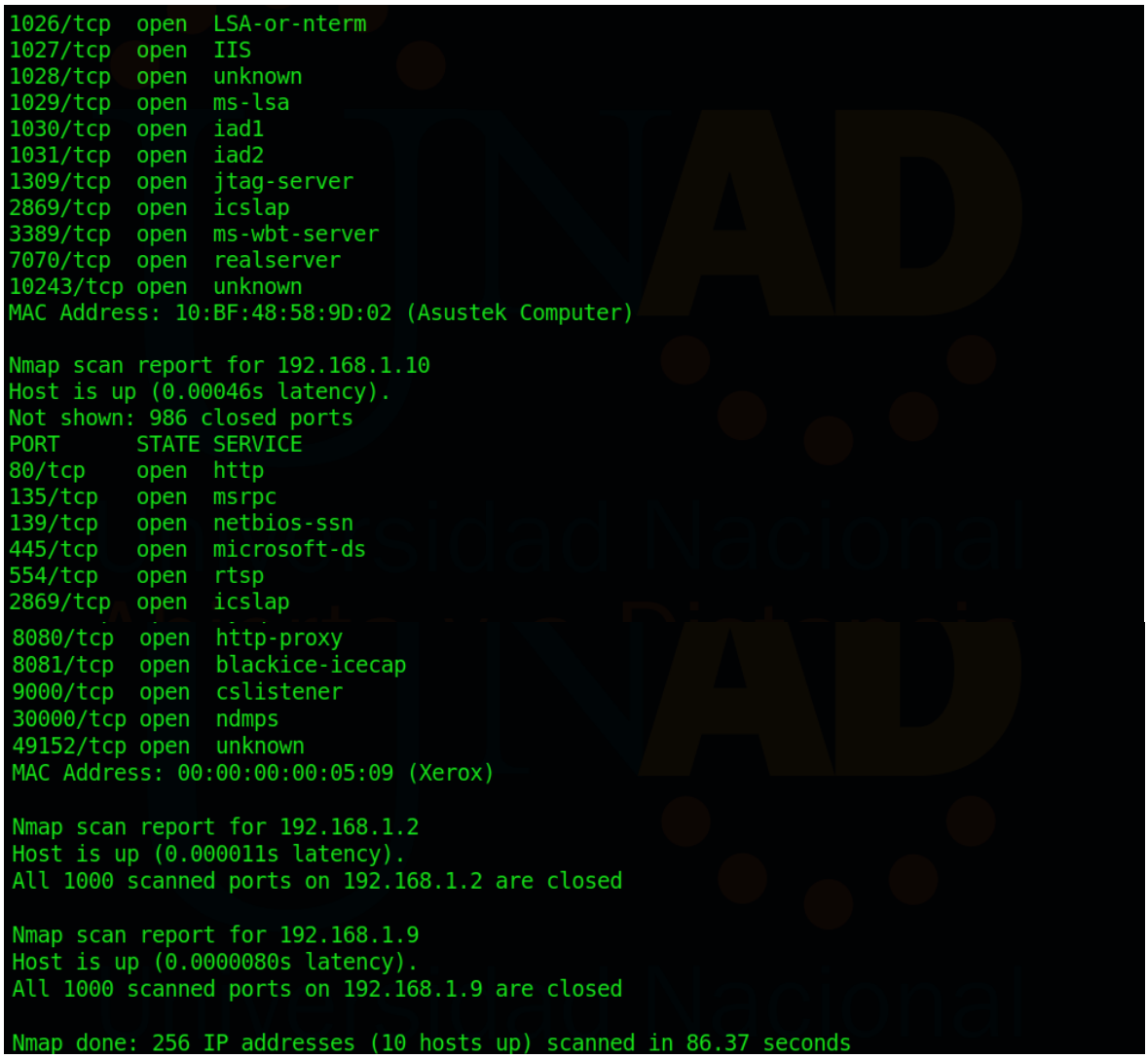
CIDR	Máscara de red
/8	0.0.0
/16	255.0.0
/24	255.255.0
/32	255.255.255

Fuente. Autor.

Por ejemplo, el valor 192.168.1.0/24 representa las 256 direcciones IP que están en el rango de 192.168.1.0 al 192.168.1.255, a continuación, se ilustra un ejemplo.

Figura 18. Escaneo de direcciones IP con notación CIDR. Escaneo de direcciones IP por rango con notación CIDR. nmap 192.168.1.0/24





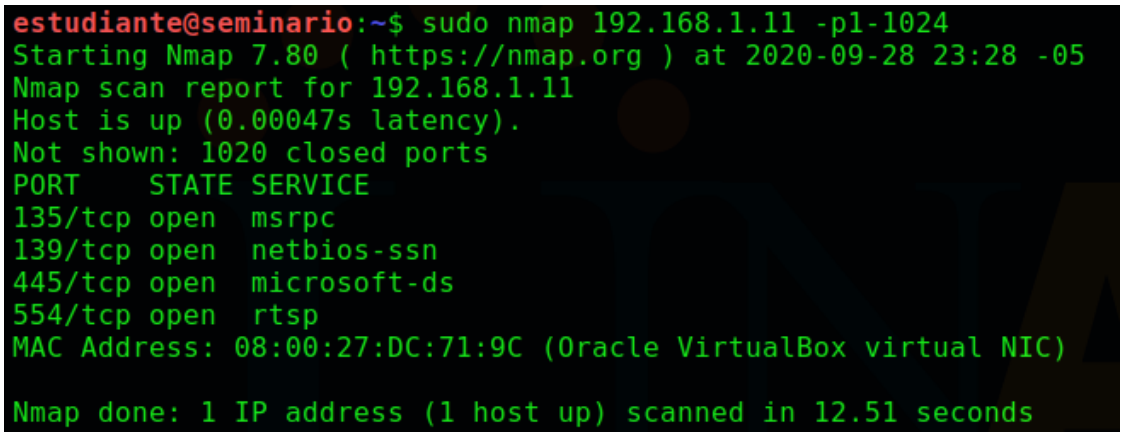
Fuente. Autor.

Escaneo por rango de puertos

Por defecto Nmap solamente escanea los 1000 puertos principales, sin embargo, los servicios se pueden poner en línea en cualquiera de los 65535 puertos disponibles, no necesariamente en los más comunes. Muchos administradores de redes deciden ejecutar los servicios en puertos muy altos, de modo que no son detectados en análisis normales. En Nmap es posible especificar un puerto específico utilizando el flag –p, la especificación de puertos se puede realizar también por medio de rangos colocando después del flag –p el intervalo deseado,

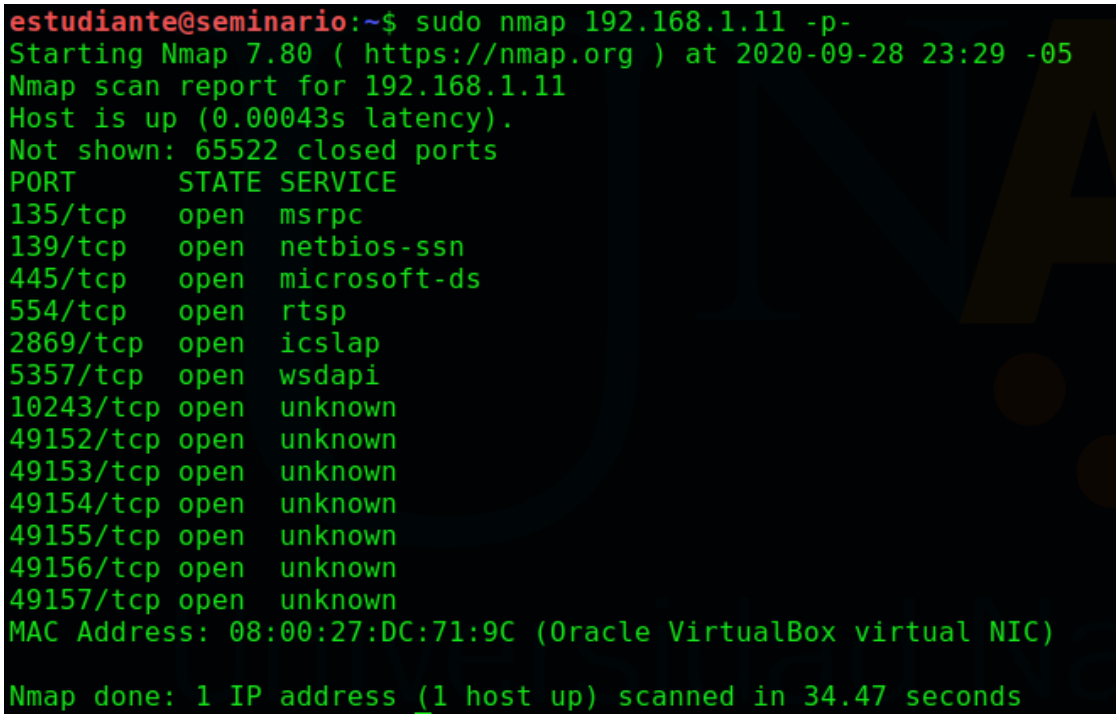
ejemplo “-p1-1024”. Para escanear los 65535 puertos se puede escribir –p1-65535 o simplemente utilizar el acceso directo –p-.

Figura 19. Escaneo por rango de puertos. nmap 192.168.1.11 –p1-1024



Fuente. Autor.

Figura 20. Escaneo por rango de puertos. nmap 192.168.1.11 –p-

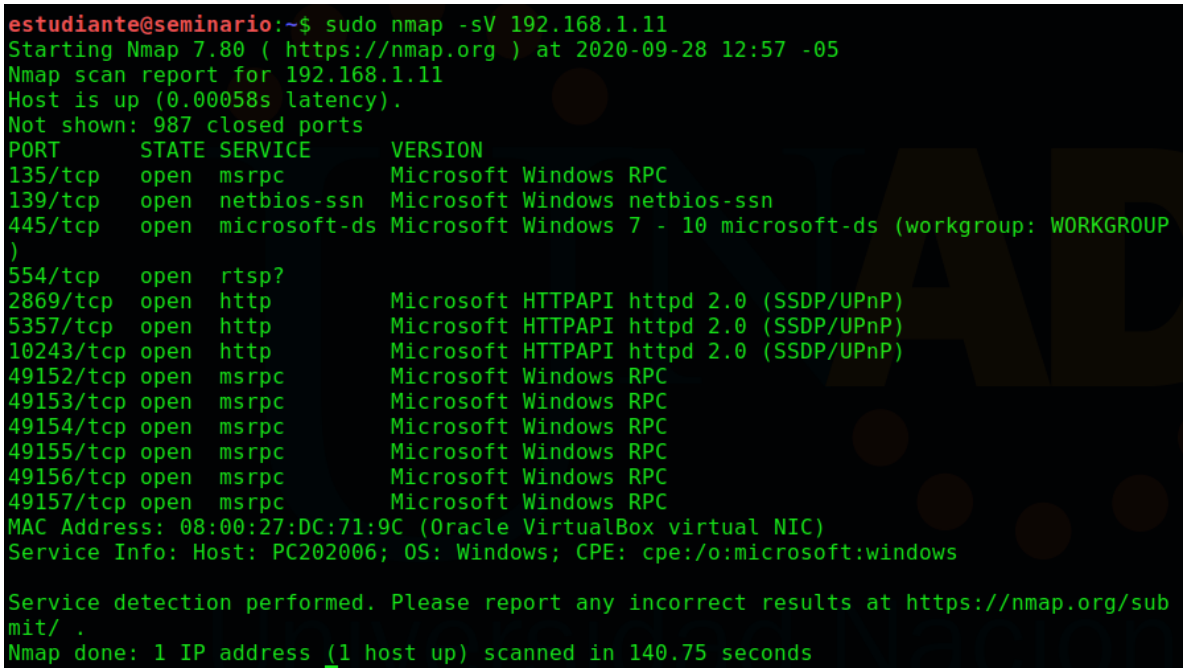


Fuente. Autor.

Escaneo de servicios

Para realizar un escaneo de servicios adicionamos el flag –sV, esto indica al Nmap que estamos realizando un análisis de la versión del servicio, esto es de gran utilidad si un sistema está ejecutando un servicio en un puerto no predeterminado, es decir un servicio que no coincide con los configurados en "/etc/services", en tales casos, es aún más importante poder averiguar exactamente qué se está ejecutando.

Figura 21. Escaneo de servicios. Nmap –sV 192.168.1.11



Fuente. Autor.

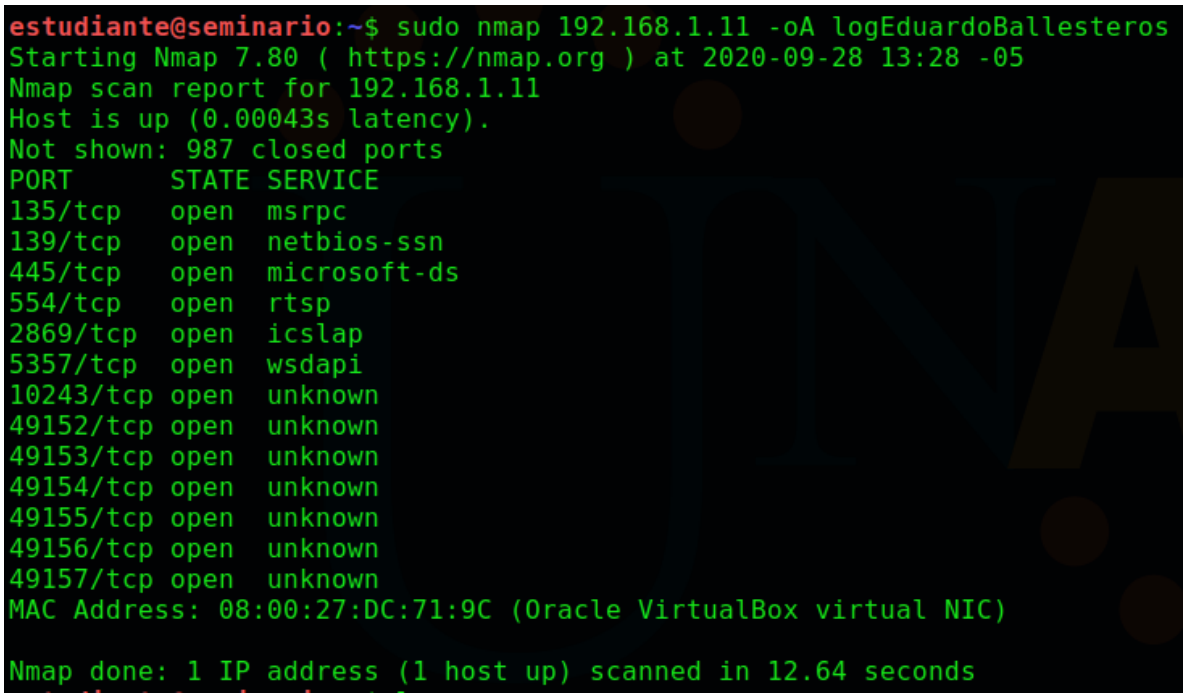
En la anterior imagen se observa que se muestra mayor número de datos, ahora se listan los puertos abiertos y los nombres reales de las versiones de los servicios. Esto es demasiado útil en el contexto de una evaluación de seguridad debido a que las vulnerabilidades se encuentran muchas veces solo en versiones específicas de software. Un punto a tener en cuenta es que, si el administrador del sistema a evaluar restringe la visualización de la versión del servicio, no es posible

saber exactamente que se encuentra ejecutando, siendo esto una muy buena técnica desde una perspectiva defensiva.

Registro de Escaneos

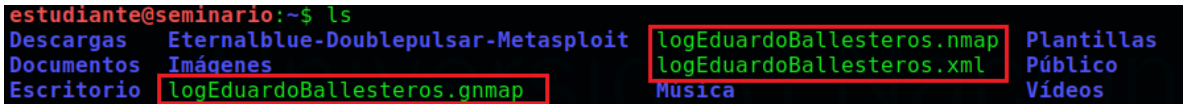
Para escaneos demasiado grandes o escaneos que tardan gran cantidad de tiempo, se hace necesario poder registrar o exportar los resultados en logs, debido a la dificultad de leer la gran cantidad de resultados que se muestran en la consola. Utilizando el flag –oA, es posible generar los 3 tipos de archivos de registro “Nmap admite tres tipos diferentes de registro, cada tipo tienen una bandera diferente para registrar registros específicos” y el segundo parámetro es simplemente el nombre que queremos que tenga el log, Nmap automáticamente les coloca su extensión de archivo.

Figura 22. Registro de escaneos. Nmap 192.168.1.11 –oA logEduardoBallesteros



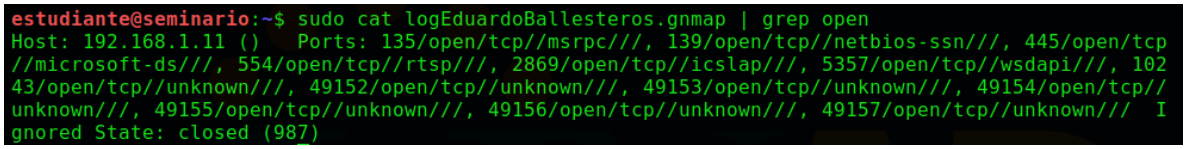
Fuente. Autor.

Figura 23. Registro de escaneos. Archivos creados por Nmap



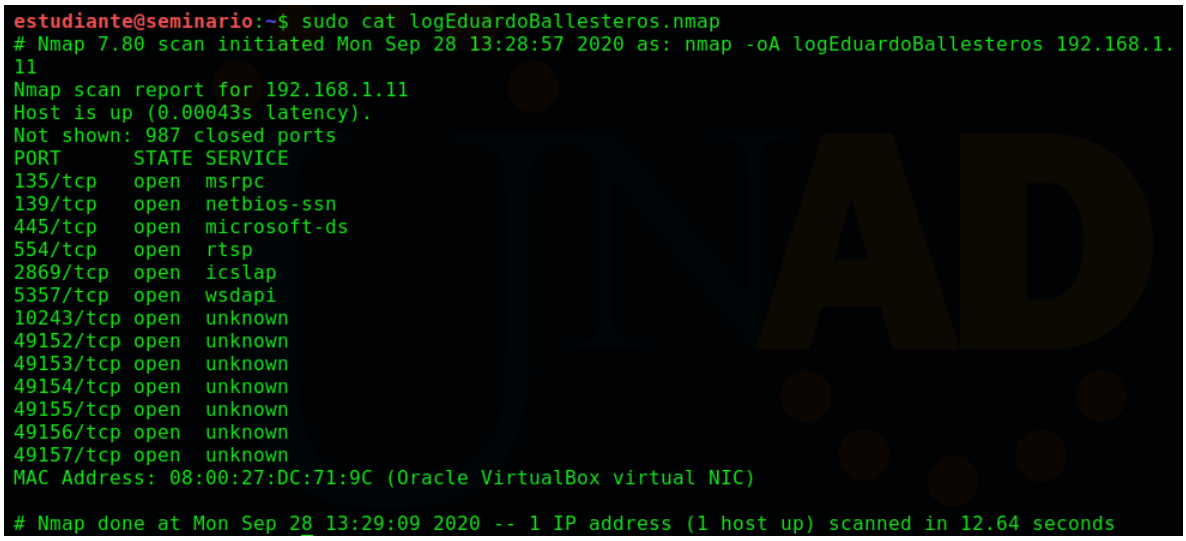
Fuente. Autor.

Figura 24. Registro de escaneos. Visualización de archivos creados por Nmap.



Fuente. Autor.

Figura 25. Registro de escaneos. Visualización de archivos creados por Nmap.



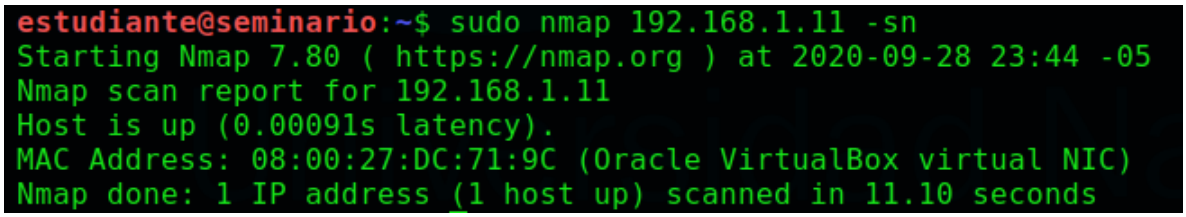
Fuente. Autor.

Como se observa en las anteriores imágenes Nmap automáticamente creo 3 archivos de log con las extensiones “.xml, .nmap y .gnmap”. El archivo xml contiene los resultados del escaneo junto con los detalles de tiempo, el archivo .nmap contiene los resultados del escaneo de forma legible y el archivo .gnmap es un archivo compatible con la herramienta de línea de comandos grep, que en archivos grandes se ve la facilidad de búsqueda con este tipo de archivo.

Detección de Hosts

Debido a que muchos administradores de sistemas intentan ocultar las máquinas, ciertos hosts pueden parecer estar fuera de línea, Nmap tiene varias formas de detectar qué hosts están en línea, la forma más sencilla de detectar hosts en línea es ejecutando un barrido de ping, para ejecutar un barrido con Nmap se debe usar el flag -sn ,esto asegura que solo se ejecute un barrido de ping, en lugar de un escaneo completo de puertos, lo cual es excelente solo para averiguar qué hosts se encuentran en línea.

Figura 26. Detección de hosts en línea. nmap 192.168.1.11 -sn



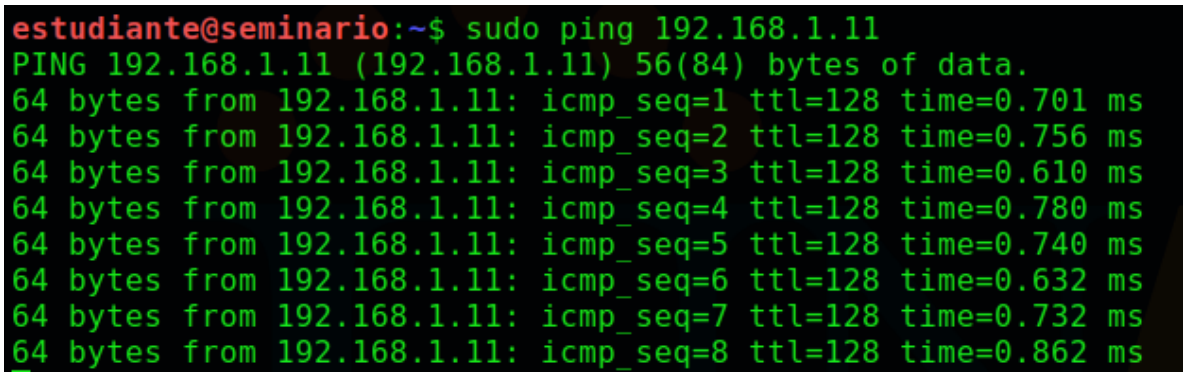
Fuente. Autor.

Cómo se observa en la figura anterior Nmap nos indica si el host se encuentra en línea mediante el mensaje “Host is up (0.00091s latency)”. Se debe tener en cuenta que muchas veces los administradores de red, deshabilitan el poder hacer ping a ciertas máquinas. Cuando un sistema no responde a los barridos de ping se torna difícil identificar si se encuentra en línea, Nmap proporciona un método agnóstico de ping para enfrentarnos ante estas situaciones, cuando Namp realiza un escaneo normal, lo que primero se ejecuta es un barrido de ping y posteriormente se realiza un escaneo de puertos, si las máquinas no responden al ping, no se realizará el barrido de servicios, lo que significa que incluso si la máquina tiene servicios en línea, estos no se detectarán.

Al ejecutar un escaneo con el flag –Pn, Nmap omite el barrido inicial de ping, esto generalmente toma más tiempo en ejecutarse debido a que realizar un análisis a hosts que realmente si están fuera de línea es una gran pérdida de tiempo, pero

es de gran utilidad encontrando hosts que no se hubieran detectado mediante un escaneo normal. Para este ejemplo, se deshabilitará el ping al host 192.168.1.11

Figura 27. Detección de host en línea. Validación de ping a host 192.168.1.11



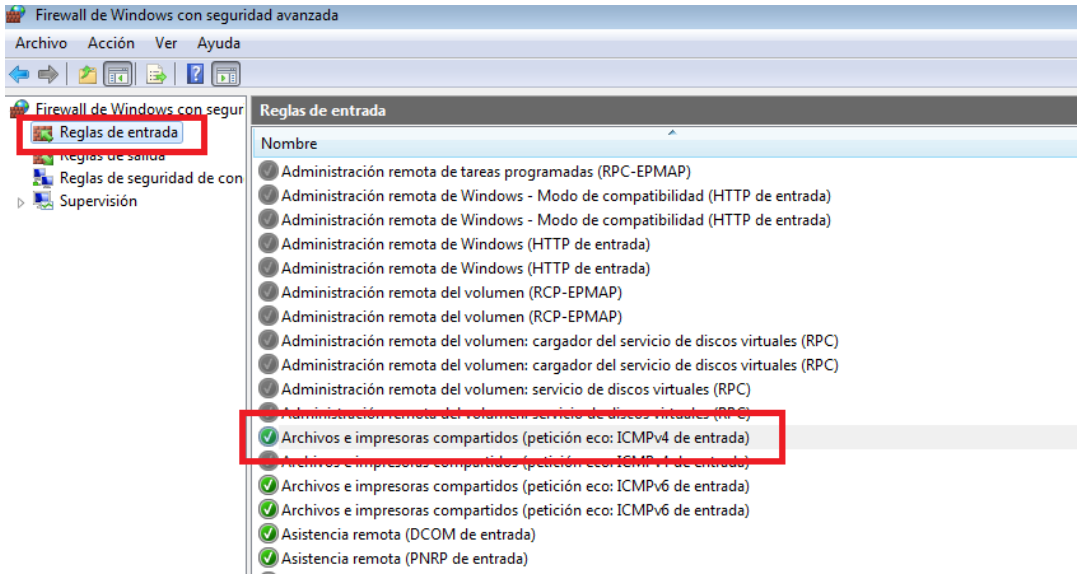
Fuente. Autor.

Se procede a deshabilitar el ping en la máquina destino Windows 7 como se muestra en las figuras 23, 24 y 25.

Figura 28. Detección de host en línea. Deshabilitar ping en máquina destino Windows 7

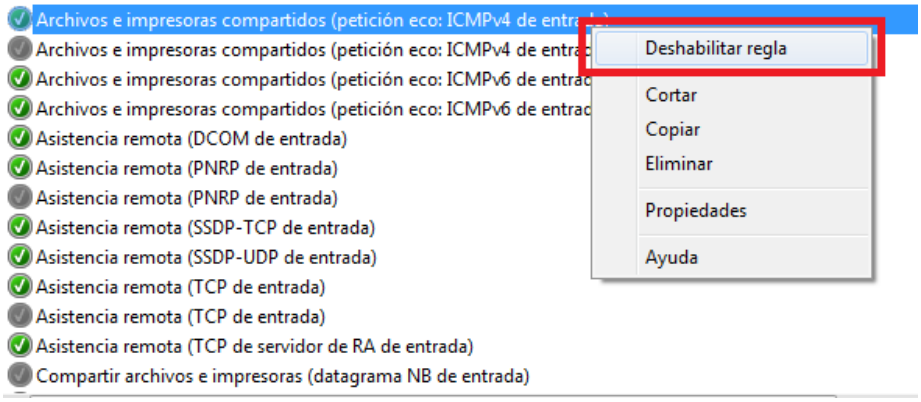
Fuente. Autor.

Figura 29. Detección de host en línea. Deshabilitar ping en máquina destino Windows 7



Fuente. Autor.

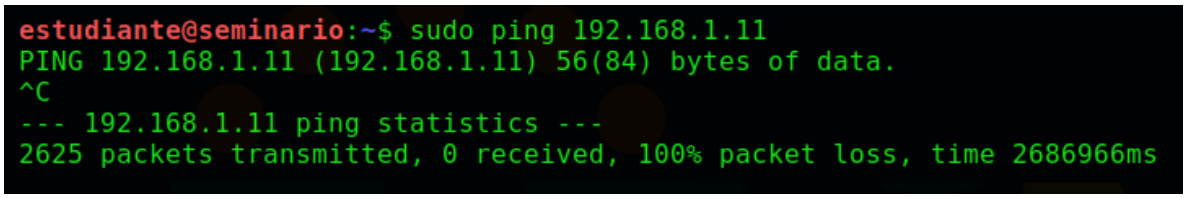
Figura 30. Detección de host en línea. Deshabilitar ping en máquina destino Windows 7



Fuente. Autor.

Después de deshabilitar el ping en la máquina destino, realizamos una validación de barrido de ping, como se ilustra en la siguiente figura.

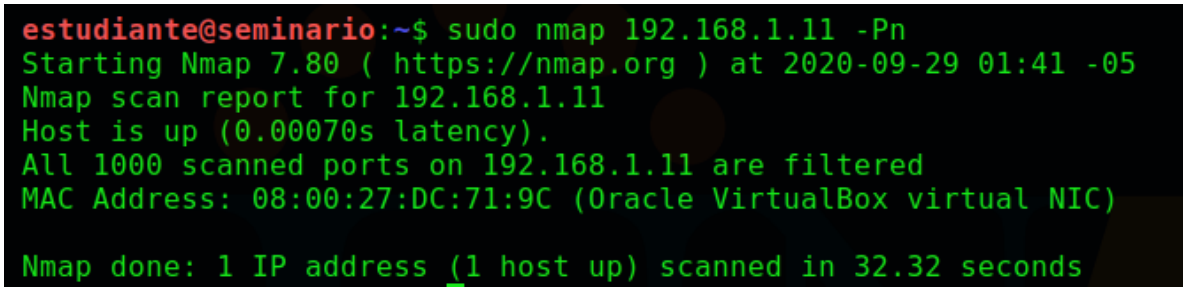
Figura 31. Detección de host en línea. Validación de ping a host 192.168.1.11



Fuente. Autor.

Como se observa en la anterior figura, ya no es posible realizar un ping al host 192.168.1.11, pero si realizamos la detección de host en línea con Nmap podemos validar que el host si se encuentra en línea.

Figura 32. Detección de host en línea con flag –sn. nmap 192.168.1.11 -Pn



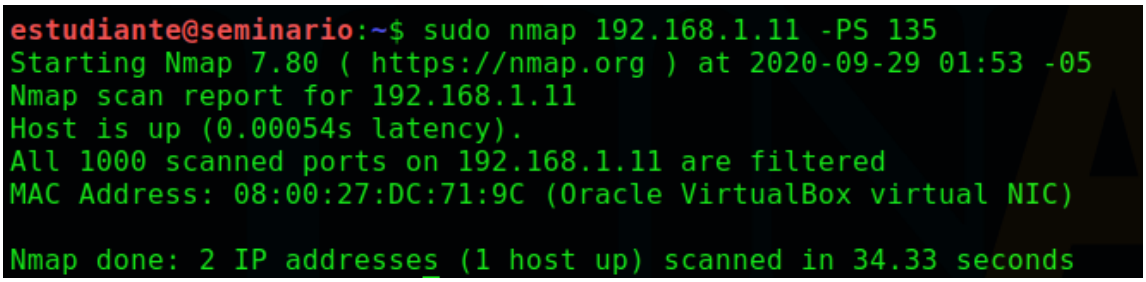
Fuente. Autor.

Cómo se observa en la figura anterior, a pesar de haber deshabilitado el ping, Nmap detecta que el host se encuentra en línea.

Otra gran característica de Nmap que nos puede ayudar en la detección y descubrimiento de hosts es el escaneo de ping TCP SYN, en lugar de enviar una solicitud de ping ICMP “los administradores pueden deshabilitar la respuesta del ping”, el escaneo TCP SYN puede detectar host en línea si responden a una solicitud TCP SYN en un puerto determinado, por ejemplo se puede dar el caso de un servidor web con el ping deshabilitado pero al utilizar el flag –PS 443 detecta si hay una respuesta ante el puerto 443, esto es de gran utilidad y es de las características más apreciadas de Nmap, en la siguiente figura se ilustra el uso de

esta técnica al mismo host del ejemplo anterior con el ping deshabilitado, como se puede observar, Nmap detecta que el host se encuentra arriba.

Figura 33. Detección de host en línea con flag –Pn. nmap 192.168.1.11 –PS 135

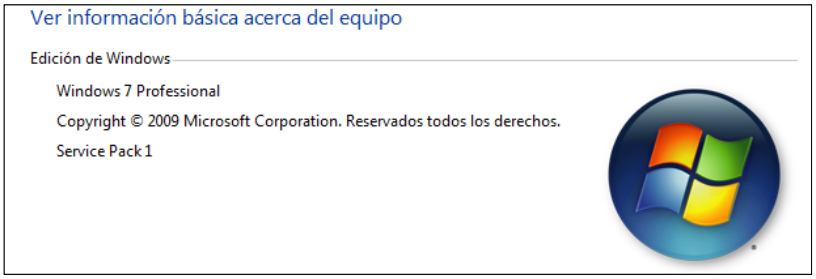


Fuente. Autor.

Detección del Sistema Operativo

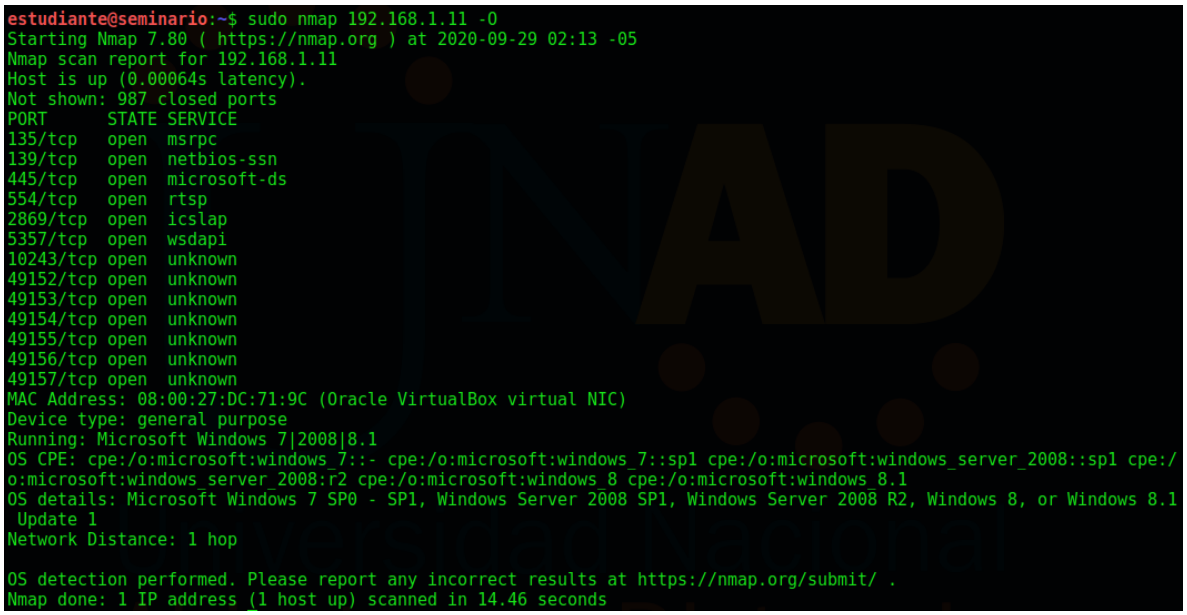
Uno de los elementos más importantes de Nmap es el poder realizar la detección del sistema operativo de una máquina. Al intentar identificar y atacar un objetivo, una de las piezas de información más útiles es qué sistema operativo está ejecutando esa máquina. Esto tradicionalmente es una cosa difícil de resolver, sin embargo, los desarrolladores de Nmap, con la ayuda de la comunidad de seguridad de la información en general, han podido crear una base de datos de los fingerprint de sistemas operativos, lo que puede ayudar en buena forma a identificar qué sistema operativo se está ejecutando en la máquina objetivo. Para realizar esto se utiliza el flag -O.

Figura 34. Sistema operativo del host destino. Windows 7 Pro



Fuente. Autor.

Figura 35. Detección del Sistema Operativo. nmap 192.168.1.11 -O



Fuente. Autor.

Como se puede observar en la anterior figura, primero se identifica la dirección MAC de la máquina “solo se puede ver la dirección MAC cuando se realizan escaneos en una red local no sobre internet”, después se observa el campo OS CPE y OS details, en estos campos se encuentra el sistema operativo y su respectiva versión. En muchos casos la detección del sistema operativo no se realiza de forma sencilla, sin embargo, el mismo Nmap nos indica que tan seguro se encuentra en los resultados que nos está brindando, para este punto es importante siempre estar trabajando con la última versión de Nmap para garantizar que se tengan las últimas bases de datos de los fingerprint de sistemas operativos.

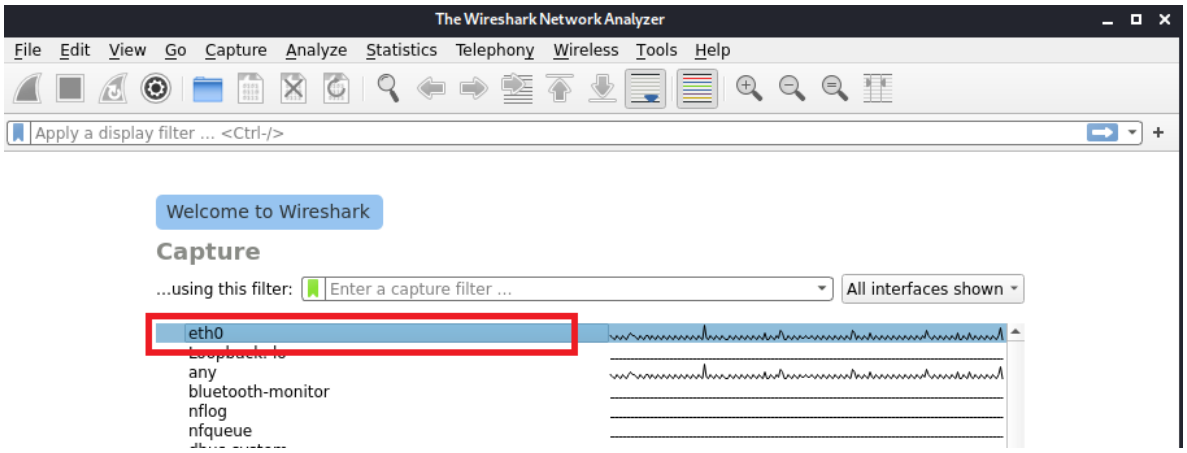
5.2. PRÁCTICAS WIRESHARK

Cómo Capturar Tráfico

Para capturar el tráfico en tiempo real con Wireshark se debe colocar algún tipo de dispositivo en el cable donde se pueda ver el tráfico que se envía y recibe y luego replicar ese tráfico a las máquinas de diagnóstico que posiblemente estén ejecutando Wireshark. En una red moderna si se tiene un switch administrado, como un switch de Cisco, puede ingresar al switch y decirle que replique el tráfico que ve en un puerto a otro puerto diferente, este puerto podría conectarse a una máquina con Wireshark para capturar el tráfico siendo de gran utilidad para este tipo de redes porque no se requiere ningún otro hardware, simplemente se ingresa al switch y se le indica que replique los datos en el sistema de monitoreo. Para capturar el tráfico de forma inalámbrica, se debe tener en cuenta que existen varios modos, el modo de monitorización es un modo que recibe todos los paquetes en un canal específico, entonces, se le dice a la tarjeta de red o tarjeta inalámbrica que reciba todo el tráfico en un canal específico y luego capturará todo ese tráfico para cualquier SSID y cualquier red que esté en ese canal, el modo promiscuo que es el modo más común de encontrar en los controladores inalámbricos y permite recibir todos los paquetes en un SSID conectado, en una red conectada, se capturará cualquier cosa que atraviese ese nombre de red y ese SSID.

Para capturar el tráfico, todo lo que se tiene que hacer en la última versión de Wireshark es hacer doble clic en la interfaz y comenzará a capturar tráfico, después se podrá ver que el tráfico comienza a visualizarse en pantalla, este proceso se ilustra en las siguientes figuras.

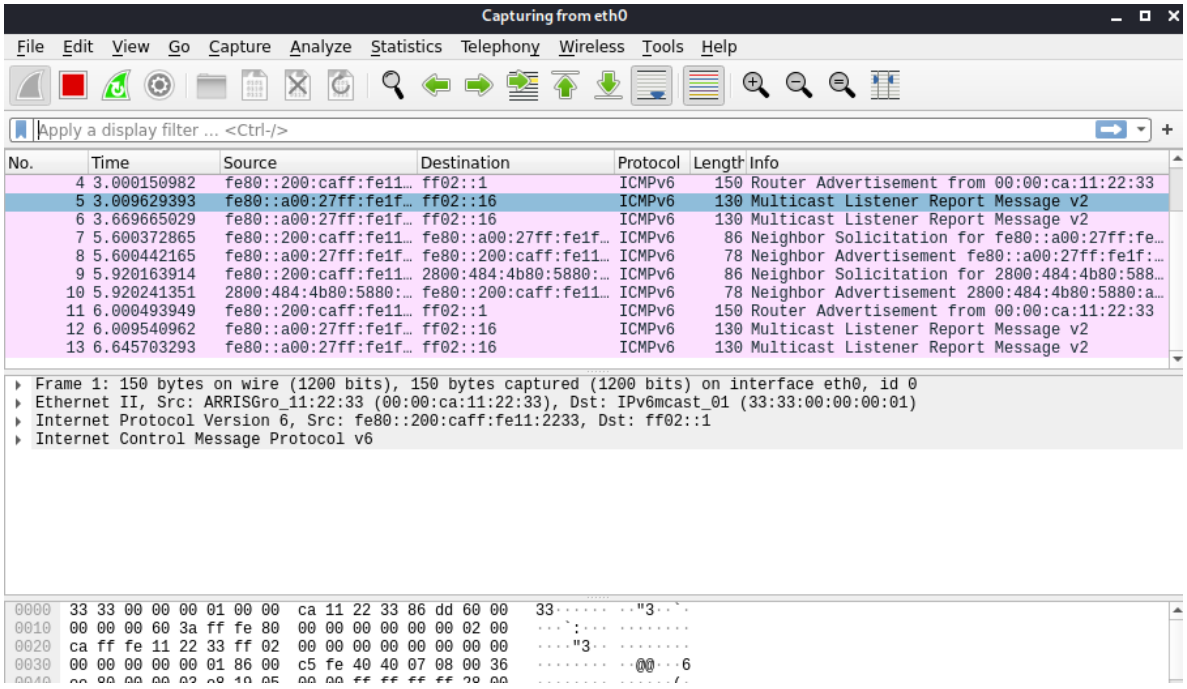
Figura 36. Wireshark, captura de tráfico



Fuente. Autor.

Se debe tener en cuenta que hay servicios que se ejecutan en segundo plano y posiblemente navegadores web minimizados, y cosas así. Pero verá que hay bastantes comunicaciones solo en una computadora estándar inactiva.

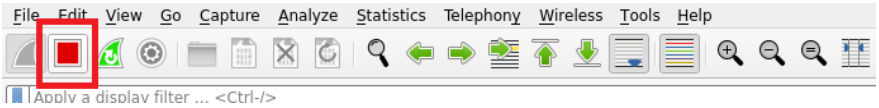
Figura 37. Wireshark, captura de tráfico



Fuente. Autor.

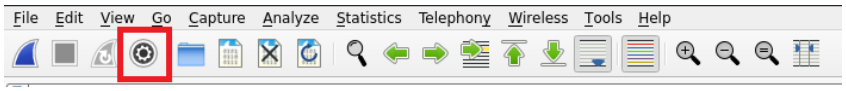
Como se visualiza en la figura anterior los paquetes se desplaza y se actualizan en tiempo real, esto es útil para algunas situaciones, aunque puede que no sea útil para todas si se tiene un sistema que recibe una gran cantidad de datos, esa podría no ser una situación ideal, especialmente si se está usando la GUI, pero esto se puede apagar para que no use la tarjeta gráfica y la potencia del procesador para poder realizar esta actualización en tiempo real. Para hacerlo esto se da clic en el botón “Stop”, luego se da clic en el botón “Capture Options”, al hacer esto se despliega una ventana con las opciones, posteriormente se deshabilitan los checks de “Update list of packets in real-time” y “Automatically scroll during live capture”.

Figura 38. Wireshark, detener captura de tráfico en tiempo real.



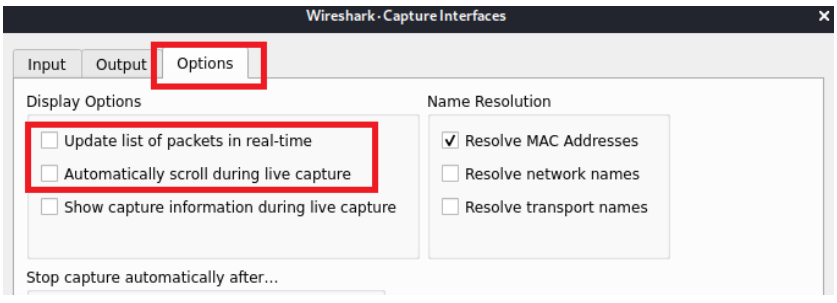
Fuente. Autor.

Figura 39. Wireshark, detener captura de tráfico en tiempo real. Capture Options.



Fuente. Autor.

Figura 40. Wireshark, detener captura de tráfico en tiempo real. Capture Options.

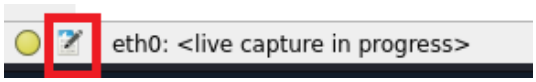


Fuente. Autor.

Creación de anotaciones

Para crear comentarios a un paquete específico se da clic en el botón “Open the capture File Properties dialog” ubicado en la parte inferior izquierda de Wireshark.

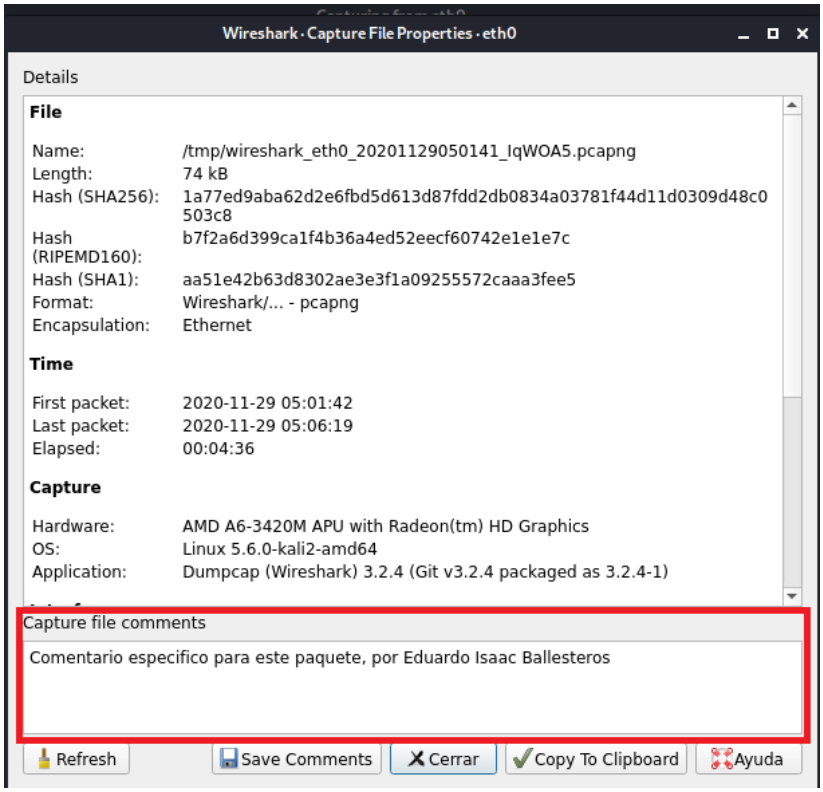
Figura 41. Wireshark, creación de anotaciones. Capture File Properties.



Fuente. Autor.

Al dar clic se despliega una ventana con información relacionada al paquete capturado, en la parte inferior se encuentra una caja de texto con el título “Capture File Comment”, en esta caja de texto se ingresa el respectivo comentario o descripción que se necesita de la captura del paquete, se da clic en el botón “Save Comments”.

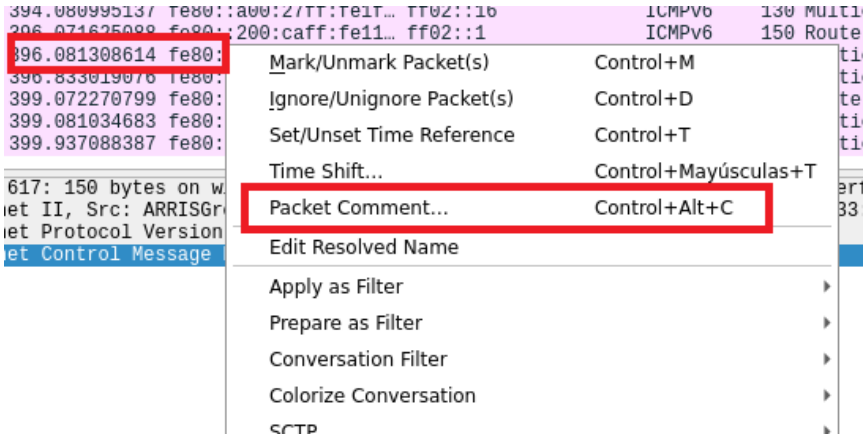
Figura 42. Wireshark, creación de anotaciones. Capture File Properties.



Fuente. Autor.

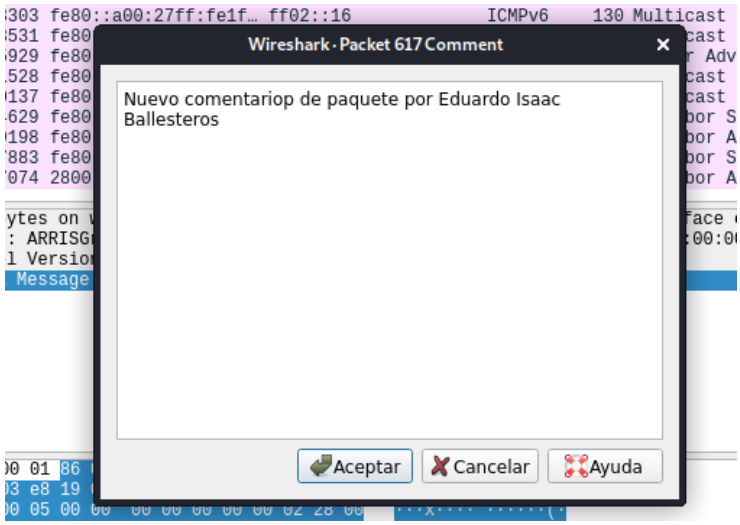
Otra forma de adicionar comentarios a los paquetes es dando clic derecho sobre el paquete en particular y seleccionado la opción “Packet Comment”, al dar clic se despliega una ventana con un campo de texto donde se ingresa el respectivo comentario, después de guardar el comentario en la ventana “Packet Details” se observa de color verde la opción de “Packet comments”.

Figura 43. Wireshark, creación de anotaciones. Packet Comment.



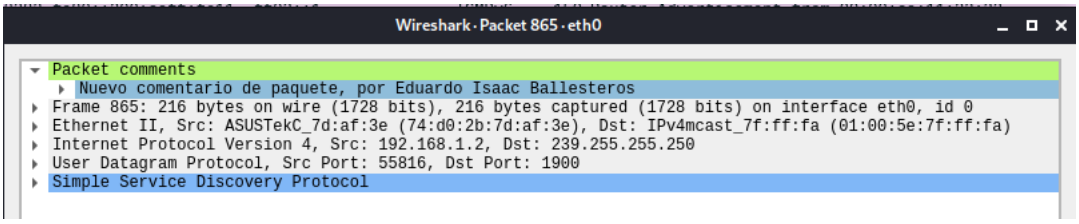
Fuente. Autor.

Figura 44. Wireshark, creación de anotaciones. Packet Comment.



Fuente. Autor.

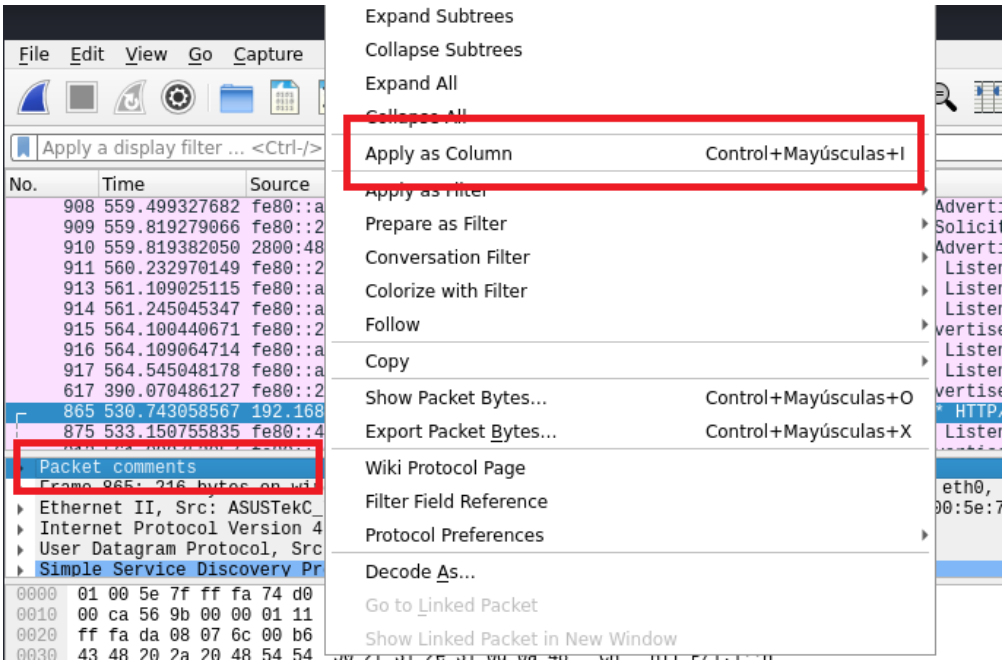
Figura 45. Wireshark, creación de anotaciones. Packet Comment.



Fuente. Autor.

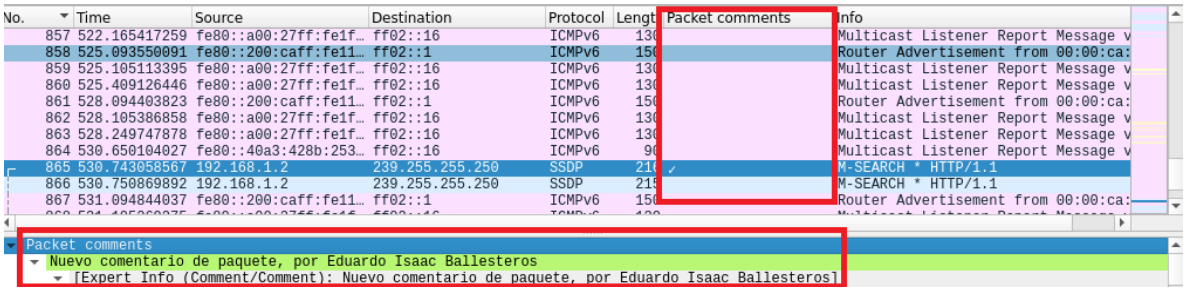
Existen diversas formas para poder filtrar los paquetes a los cuales les hemos adicionado comentarios, una de estas es dando clic en el apartado de “Packet comments” y seleccionando la opción de “Apply as Column”, después de hacer esto, en la ventana principal o ventana “Packet List” se adiciona automáticamente una nueva columna con el nombre de “Packet comments”, para los paquetes que contengan comentarios se visualiza un check, como se muestra en las siguientes imágenes.

Figura 46. Wireshark, creación de filtros de anotaciones. Packet Comments.



Fuente. Autor.

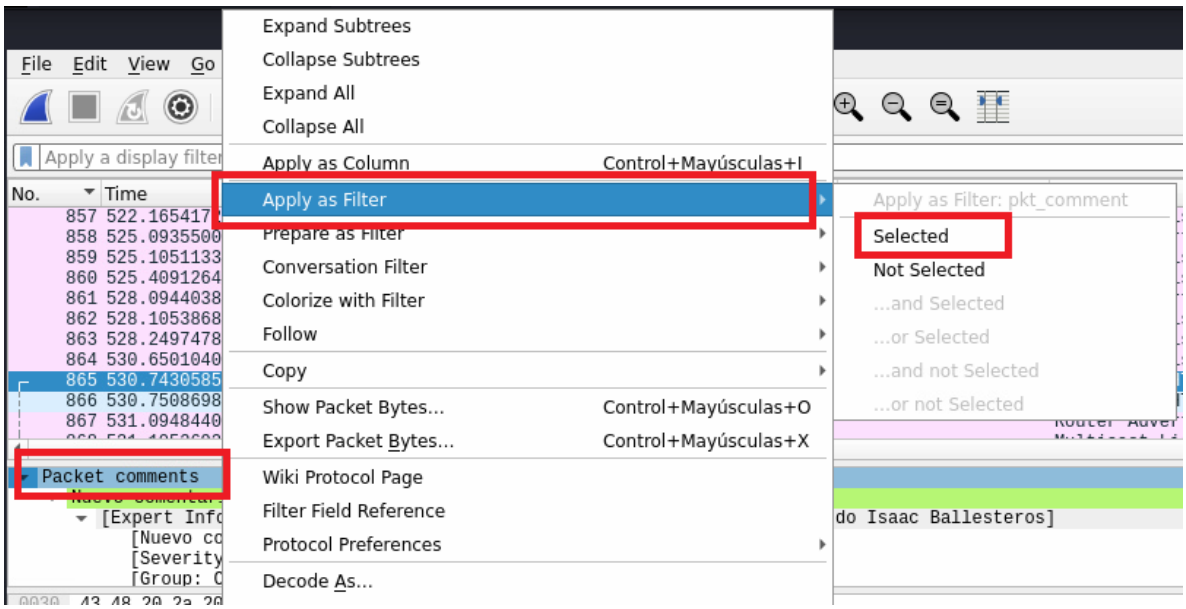
Figura 47. Wireshark, creación de filtros de anotaciones, nueva columna Packet comments



Fuente. Autor.

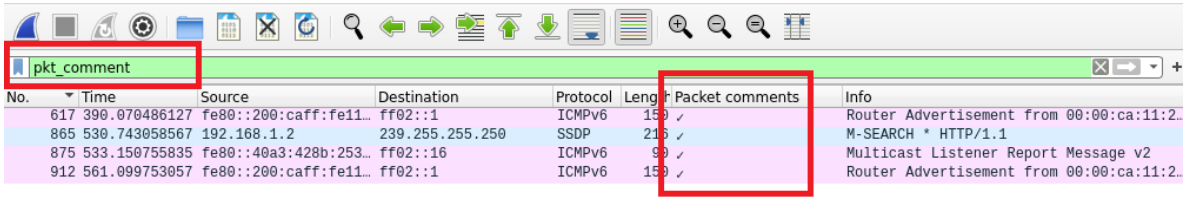
Otra forma de filtrar los paquetes que contienen comentarios es dando clic derecho en el apartado de “Packet comments” y seleccionando la opción “Apply as Filter - Selected”, al hacer esto, en la parte superior de Wirsehark en el campo de los filtros se visualiza el texto “pkt_comment” y automáticamente en la venta aprincipal “Packet List” solo se listan los paquetes que tienen comentarios.

Figura 48. Wireshark, creación de filtros de anotaciones.Apply as Filter.



Fuente. Autor.

Figura 49. Wireshark, creación de filtros de anotaciones.Apply as filter.



Fuente. Autor.

Manejo de filtros

Wireshark utiliza una notación con nombre “BPF Berkeley Packet Filter”, una expresión BPF contienen una o más primitivas como lo pueden ser un ID, un nombre, un número, una dirección IP, etc, más un calificador. Un calificador tiene los siguientes tres componentes: tipo, dirección y protocolo.

Un tipo puede ser un host, un puerto o un rango de puertos, la dirección puede ser el origen o el destino o ambas y el protocolo puede ser TCO, UDP, etc. Ejemplo: ip host 192.168.1.2, donde ip es el protocolo, host es el tipo y el ID es la dirección IP, este filtro en la notación BPF filtra todos los paquetes de esa dirección IP, sin importar si es origen o destino. Para aplicar un filtro en particular se ingresa la expresión BPF en la caja de texto ubicada en la parte superior que tiene el texto “…using this filter”, al ingresar cualquier filtro, la captura de paquetes se inicializa con la expresión indicada, el proceso se ilustra en las siguientes figuras.

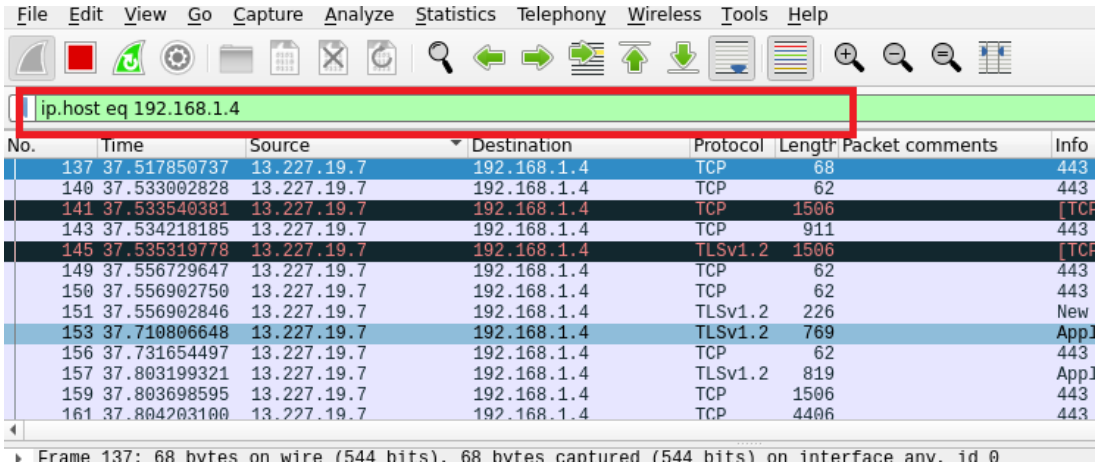
Figura 50. Wireshark, manejo de filtros. Using this filter.

Fuente. Autor.

Figura 51. Wireshark, manejo de filtros.Enter a capture filter.

Fuente. Autor.

Figura 52. Wireshark, manejo de filtros en interfaz principal.



Fuente. Autor.

Otra forma de utilizar los filtros es empezar a capturar todo y después utilizar algún filtro para visualizar exactamente lo que se desea ver, ya existen una serie de operadores de filtro que se pueden usar para combinar y crear expresiones completas. Se pueden combinar filtros junto con sus operadores, como si fuese una ecuación matemática. Para aplicar estos filtros descritos anteriormente, iniciamos una captura de forma normal y en la parte superior se encuentra una caja de texto con el texto “Apply a display filter”, en esta parte escribimos el filtro a utilizar, utilizamos algún filtro guardado previamente o utilizamos algunos de los que están preconstruidos por el propio Wireshark. En la siguiente tabla se listan los operadores más usados para poder realizar los filtros.

Tabla 6. Operadores para filtros

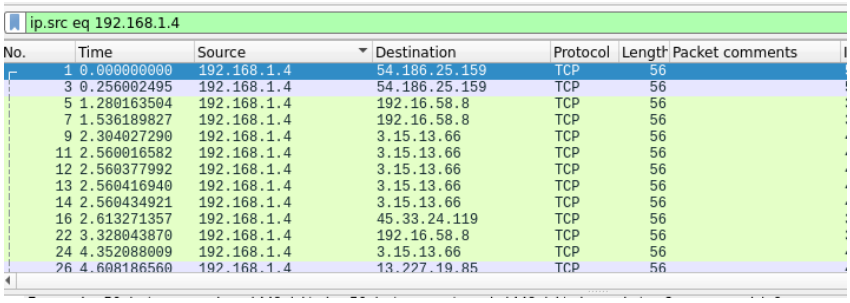
Operador	Equivalencia
eq	==
le	<=
or	||
not	!
and	&&
ne	!=
gt	>
ge	>=
contains	
matches	

Fuente. Autor.

El proceso descrito anteriormente se ilustra con los siguientes ejemplos:

Paquetes con ip de origen 192.168.1.4 con operador eq: ip.src eq 192.168.1.4

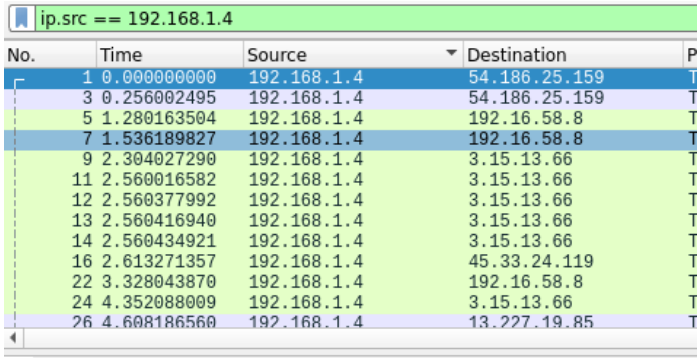
Figura 53. Wireshark, manejo de filtros. Paquetes con IP origen y operador eq.



Fuente. Autor.

Paquetes con ip de origen 192.168.1.4 con operador ==: ip.src == 192.168.1.4

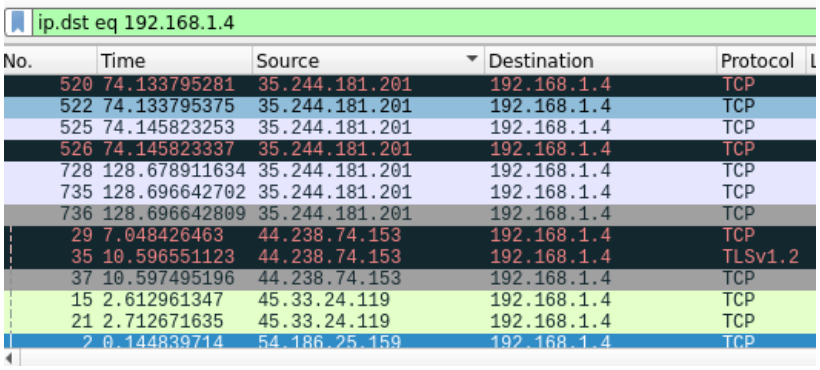
Figura 54. Wireshark, manejo de filtros. Paquetes con IP origen y operador ==.



Fuente. Autor.

Paquetes con destino 192.168.1.4 con operador eq: ip.dst eq 192.168.1.4

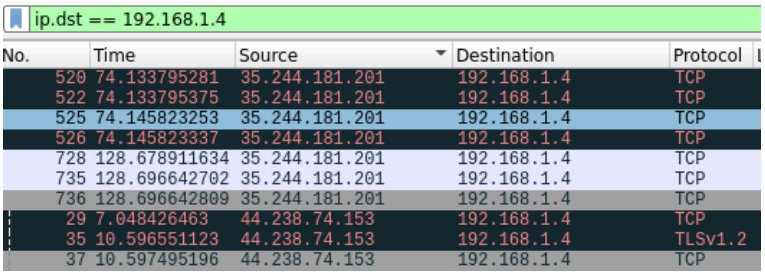
Figura 55. Wireshark, manejo de filtros. Paquetes con IP destino y operador eq.



Fuente. Autor.

Paquetes con destino 192.168.1.4 con operador ==: ip.dst == 192.168.1.4

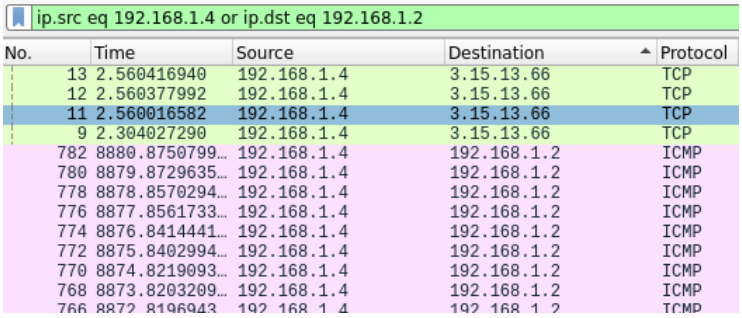
Figura 56. Wireshark, manejo de filtros. Paquetes con IP destino y operador ==.



Fuente. Autor.

Paquetes con ip de origen 192.168.1.4 o paquetes con destino 192.168.1.2: ip.src eq 192.168.1.4 or ip.dst eq 192.168.1.2

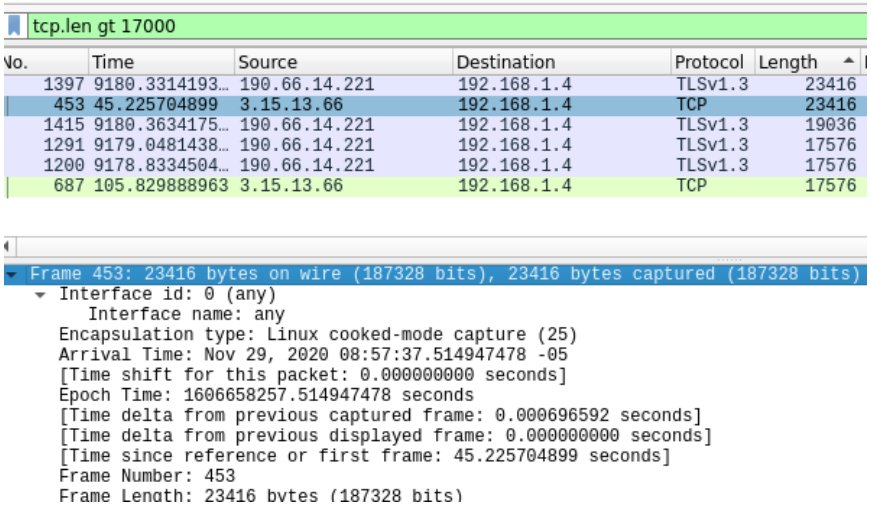
Figura 57. Wireshark, manejo de filtros. Paquetes con IP origen o IP destino.



Fuente. Autor.

Paquetes con longitudes mayores a 17000 bytes: ip.len gt 17000

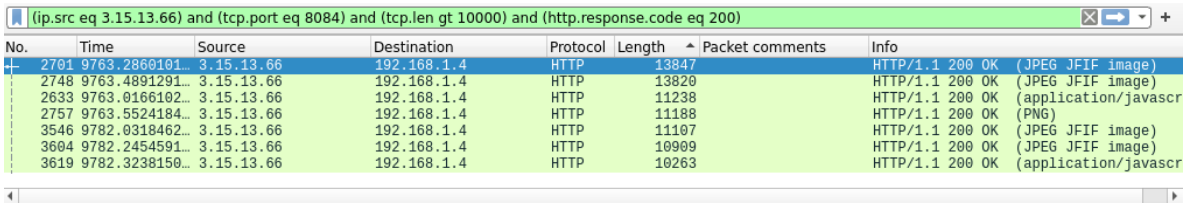
Figura 58. Wireshark, manejo de filtros. Paquetes con longitud mayor a 17000 bytes.



Fuente. Autor.

Paquetes con ip de origen 3.15.13.66 por el puerto 8084 con longitud mayor a 10000 bytes y de código de respuesta http 200: (ip.src eq 3.15.13.66) and (tcp.port eq 8084) and (tcp.len gt 10000) and (http.response.code eq 200)

Figura 59. Wireshark, manejo de filtros. Paquetes con IP origen o IP destino.



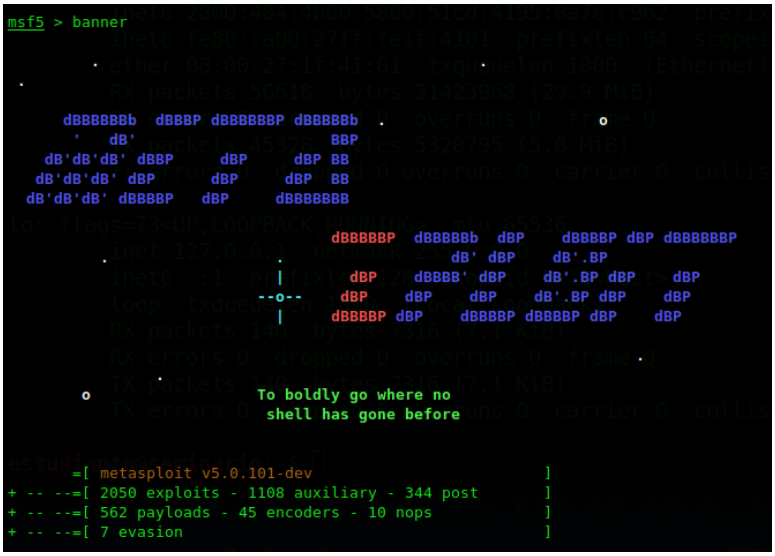
Fuente. Autor.

5.3. PRÁCTICAS METASPLOIT

Banner

Es un comando que se utiliza para mostrar la información del banner de Metasploit Framework. Esta información generalmente incluye los detalles de la versión y la cantidad de exploits, auxiliares y payloads disponibles en la versión instalada.

Figura 60. Metasploit, comando banner

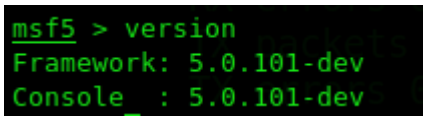


Fuente. Autor.

Version

Este comando se utiliza para verificar la versión de la instalación actual de Metasploit Framework.

Figura 61. Metasploit, comando version

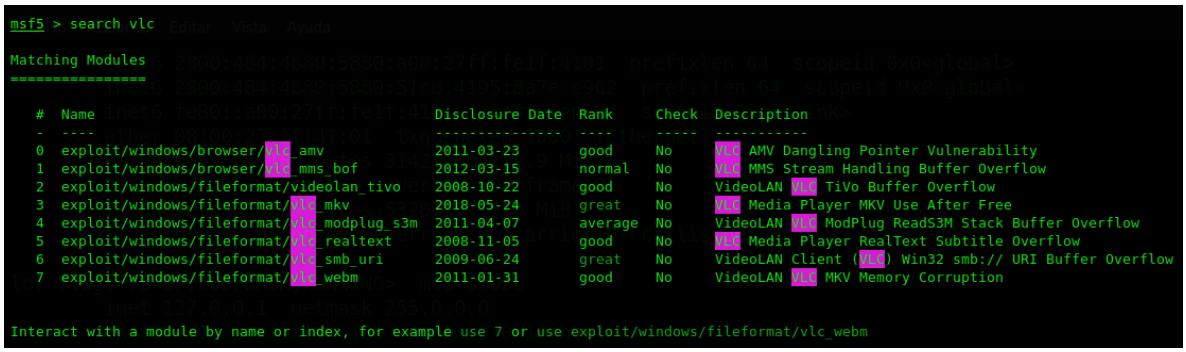


Fuente. Autor.

Search

El comando search nos ayuda a encontrar exploits o módulos.

Figura 62. Metasploit, comando search



Fuente. Autor.

Variables

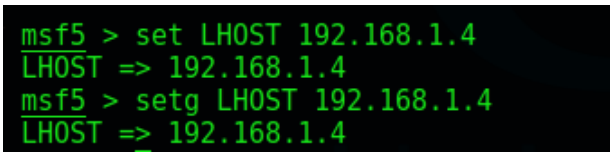
Para la mayoría de los exploits que se usan dentro de Metasploit, se deben establecer valores para algunas de las variables. Las siguientes son algunas de las variables comunes y más importantes en Metasploit Framework:

- LHOST “Local host”: Esta variable contiene la dirección IP del host atacante, es decir la IP del sistema que inicia el exploit.
- LPORT “Local port”: Esta variable contiene el número de puerto local del host atacante. Este valor es necesario cuando el exploit usa una shell inversa.
- RHOST “Remote host”: Esta variable contiene la dirección IP del host objetivo.
- RHOSTS “Remote hosts”: Esta variable se establece cuando se quiere lanzar un exploit en varios host al mismo tiempo, por ejemplo se puede setear el valor en 192.168.0.1/24, o de manera alternative se puede asignar un archive que contiene varias IPs de sistemas objetivo.
- RPORT “Remote port”: Esta variable contiene el número de puerto en el sistema objetivo.

Set y Setg

El comando set asigna un nuevo valor en una variable local como “RHOST, RPORT, LHOST o LPORT”, esta asignación local solo es válida para la sesión actual. El comando setg asigna un nuevo valor a la variable, pero de forma global, para que pueda utilizarse repetidamente cuando sea necesario.

Figura 63. Metasploit, comando set y setg

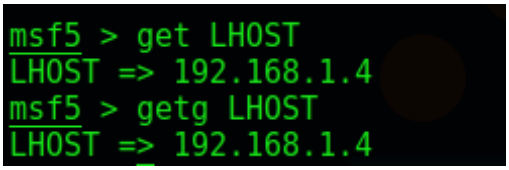


Fuente. Autor.

Get y getg

Este comando es usado para retornar el valor contenido en una variable específica. Getg se usa para retornar el valor contenido en un variable global.

Figura 64. Metasploit, comando get y getg

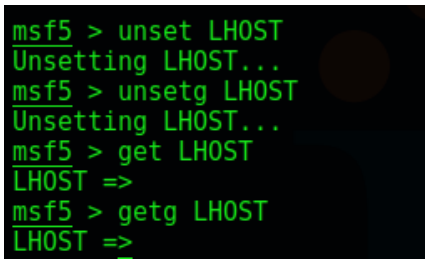


Fuente. Autor.

Unset y unsetg

El comando unset borra el valor almacenado en una variable local que se asignó por medio del comando set, el comando unsetg borra el valor almacenado previamente en una variable local por medio del comando setg.

Figura 65. Metasploit, comando unset y unsetg



Fuente. Autor.

Para usar la mayoría de módulos dentro de Metasploit, se realiza la siguiente secuencia:

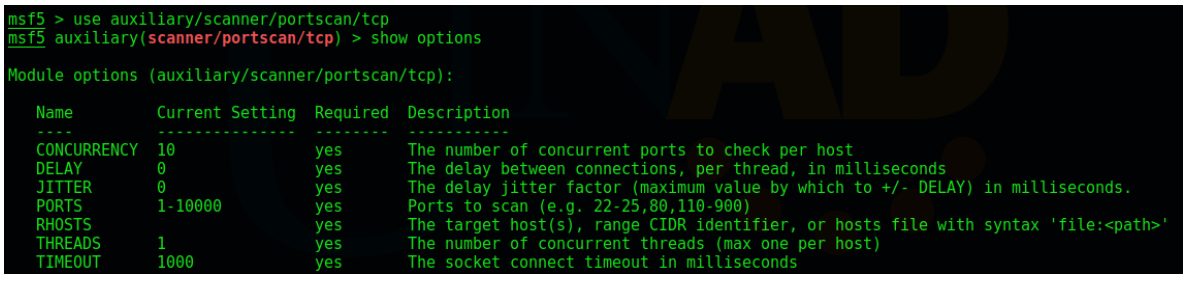
Uso del comando “use” para seleccionar el módulo requerido.

Uso del comando “show options” para enumerar todas las variables que se requieren para ejecutar el módulo seleccionado.

Uso del comando “set” para establecer los valores de cada una de las variables necesarias.

Uso del comando “run” para ejecutar el módulo con las variables configuradas anteriormente.

Figura 66. Metasploit, comando show options



Fuente. Autor.

6. CONCLUSIONES

Actualmente, las organizaciones soportan gran parte de sus procesos de negocio con sistemas e infraestructuras informáticas que deben implementar políticas y medidas de protección que garanticen el continuo desarrollo y sostenibilidad de sus actividades; de acuerdo a esto, surge la necesidad de contar con profesionales especializados en seguridad informática idóneos para poder gestionar e implementar de manera oportuna dichos requerimientos.

Como consecuencia de lo anterior, la información es el activo de más valor presente en una organización y por tal motivo se debe proteger y asegurar para garantizar su confidencialidad, integridad y disponibilidad, por tal razón como profesionales de la seguridad informática debemos poder seleccionar e implementar sistemas de seguridad para proteger a este activo de eventuales amenazas que puedan causar un impacto negativo dentro de las actividades del negocio.

Como profesionales de la seguridad informática debemos tener en cuenta que cada vez los ataques cibernéticos son más sofisticados y son más recurrentes, por tal motivo debemos estar en una continua formación y actualización tanto de conocimientos teóricos como técnicos para poder responder de manera adecuada a estas nuevas amenazas.

Es de vital importancia para los profesionales de la seguridad informática tener muy claros los conceptos teóricos y el poseer un alto conocimiento en el uso y fortalezas de cada una de las herramientas existentes para pruebas de penetración, contención y detección, además de ser conscientes de la importancia de tener un monitoreo periódico de las infraestructuras informáticas para poder asegurar los activos de las organizaciones.

Por medio del uso de herramientas tanto para seguridad ofensiva y defensiva se pueden realizar estudios de análisis de vulnerabilidades y riesgos en los sistemas informáticos presentes en una organización.

7. RECOMENDACIONES

Se recomienda a nivel organizacional tener una cultura de cambio comprendiendo que toda la información y los procesos realizados por cada una de las áreas son los activos más importantes y son vitales para la continuidad del negocio.

Debe existir un compromiso con todos los integrantes de una organización en cumplir y hacer cumplir las políticas de seguridad planteadas para poder mitigar los riesgos dentro de la infraestructura y sistemas informáticos.

Los profesionales de la seguridad informática deben tener claridad sobre las leyes que rigen los delitos de la seguridad de la información y los códigos de ética profesional establecidos con el fin de ejercer de forma correcta la profesión.

Se debe mantener la información de los activos correctamente clasificada y ordenada para poder reducir los tiempos de atención y reacción ante la materialización de cualquier amenaza.

Se recomienda la implementación de sistemas de detección de intrusos “IDS” para poder prevenir diferentes ataques.

Se recomienda mantener los equipos tanto de software como de hardware actualizados, aplicando los últimos parches o actualizaciones del fabricante.

Se recomienda tener un software antivirus actualizado, además de realizar escaneos de forma regular.

Se recomienda tener un firewall, tanto en hardware como en software.

Se recomienda realizar sesiones informativas periódicas donde se ilustre a todo el personal de todas las áreas acerca de las normas para evitar riesgos informáticos.

Se debe realizar un proceso de hardening sobre todos los elementos de la infraestructura informática de la organización que pueda comprometerse en un ataque, para tener una disminución en los incidentes de seguridad.

8. BIBLIOGRAFÍA

Acunetix Web Application Vulnerability Report 2019, Recuperado el 10 de Octubre de 2020 del sitio web de Acunetix: https://www.acunetix.com/acunetix-web-application-vulnerability-report/

CALDERON, Paulino. Network Exploration and Security Auditing Cookbook. 2ª Edición. Packt Publishing Ltd, 2017.

CHICANO, Ester. Auditoría de Seguridad Informática. IC Editorial, 2019.

GALLARDO, Gabriel. Seguridad en Bases de Datos y Aplicaciones Web: 2ª Edición. IT Campus Academy, 2016. 18 p.

JARA Héctor, PACHECO Federico. Ethical Hacking 2.0, Fox Andina. 1ª Edición, 2012.

GÓMEZ, Alberto y DE ABAJO, Nicolás. Los Sistemas De Información En La Empresa. 1ª Edición. Oviedo Principado De Asturias: Servicio De Publicaciones De La Universidad De Oviedo, 1997. 3 p.

NATH, Anish. Packet Analysis with Wireshark. Packt Publishing Ltd, 2015.

La Sociedad de la Información, Recuperado el 3 de Octubre de 2020 del sitio web de McGraw-Hill Interamericana de España, SL: http://www.mcgraw-hill.es/bcv/guide/capitulo/8448146905.pdf.

Procedimientos de seguridad básicos para Aplicaciones Web, Recuperado el 5 de Octubre de 2020 del sitio web de Microsoft Developer Network: https://msdn.microsoft.com/es-es/library/zdh19h94(v=vs.100).aspx

SAGAR, Rahalkar. Metasploit 5.0 for Beginners, Perform Penetration Testing to Secure Your IT Environment Against Threats and Vulnerabilities. 2ª Edición. Packt Publishing Ltd, 2020.

SHAW, David. Nmap Essentials. Community experience distilled. Packt Publishing Ltd, 2015.

Sniffer, Recuperado el 3 de Octubre de 2020 del sitio web de EcuRed: https://www.ecured.cu/Sniffer

SOLSONA Antonio, HUIDROBO José, JORDAN Julia. Redes de área local: administración de sistemas informáticos. Editorial Paraninfo, 2006.

¿Qué es un sniffer y cómo puede protegerse?, Recuperado el 3 de Octubre de 2020 del sitio web de Avast Academy: https://www.avast.com/es-es/c-sniffer