UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Análisis, diseño e implementación de framework de herramientas de reconocimiento perimetral y generación de informes orientado a tareas de auditoría Red Team

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Frejo Ballesteros, Daniel

Tutor/a: Martínez Hinarejos, Carlos David

CURSO ACADÉMICO: 2022/2023

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

Escola Tècnica Superior d'Enginyeria Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Análisis, diseño e implementación de *framework* de herramientas de reconocimiento perimetral y generación de informes orientado a tareas de auditoría Red Team

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Autor: Daniel Frejo Ballesteros

Tutor: Carlos D. Martínez Hinarejos

Curso 2022-2023

Resumen

En el panorama actual de la ciberseguridad, las organizaciones y empresas enfrentan desafíos cada vez mayores para proteger sus activos digitales e información confidencial. Con la creciente complejidad y sofisticación de los ciberataques, es esencial contar con medidas de seguridad efectivas y adaptativas para enfrentar estas amenazas. Las auditorías Red Team se han convertido en una práctica valiosa para evaluar y mejorar las defensas de las organizaciones al simular ataques cibernéticos reales y probar la eficacia de las medidas de seguridad implementadas.

El reconocimiento perimetral es un componente clave en cualquier auditoría Red Team, ya que se enfoca en identificar y analizar los sistemas, servicios y vulnerabilidades presentes en la red de una organización. Esta información es crucial para determinar el nivel de riesgo y exposición, así como para planificar y ejecutar ataques simulados que evalúen las capacidades de defensa de la organización. Para llevar a cabo el reconocimiento perimetral, los profesionales de la seguridad emplean diversas herramientas y técnicas que pueden variar en alcance, funcionalidad y eficacia.

En este contexto, el desarrollo de un *framework* de herramientas de reconocimiento perimetral y generación de informes que facilite la realización de auditorías Red Team se vuelve fundamental para mejorar la eficiencia y efectividad de los profesionales de la seguridad en la evaluación de la postura de seguridad de una organización. Un *framework* de este tipo permitiría la integración y automatización de las herramientas y procesos involucrados en el reconocimiento perimetral, lo que mejoraría la calidad y coherencia de los resultados obtenidos.

El objetivo de este Trabajo de Fin de Grado (TFG) es analizar, diseñar e implementar teóricamente un *framework* de herramientas de reconocimiento perimetral y generación de informes orientado a tareas de auditoría Red Team. Para lograr esto, se investigarán las herramientas y técnicas existentes en el ámbito del reconocimiento perimetral, se identificarán las necesidades y requerimientos para un *framework* de reconocimiento perimetral y se diseñará e implementará un sistema que integre las herramientas y procesos necesarios para llevar a cabo un reconocimiento perimetral eficiente y efectivo.

Palabras clave: Reconocimiento, Descubrimiento, Vulnerabilidad, Análisis

Abstract

In today's cybersecurity landscape, organizations and businesses face increasíng challenges to protect their digital assets and sensitive information. With the increasíng complexity and sophistication of cyber attacks, it is essential to have effective and adaptive security measures in place to deal with these threats. Red Team audits have become a valuable practice to assess and improve organizations' defenses by simulating real cyberattacks and testing the effectiveness of implemented security measures.

Perimeter reconnaissance is a key component in any Red Team audit, as it focuses on identifying and analyzing the systems, services, and vulnerabilities present in an organization's network. This information is crucial for determining the level of risk and exposure, as well as for planning and executing simulated attacks that assess the organization's defense capabilities. To perform perimeter reconnaissance, security professionals use a variety of tools and techniques that can vary in scope, functionality, and effectiveness.

In this context, the development of a framework of perimeter reconnaissance and reporting tools that facilitates the performance of Red Team audits becomes essential to improve the efficiency and effectiveness of security professionals in evaluating the security posture of an organization. A framework of this type would allow the integration and automation of the tools and processes involved in perimeter recognition, which would improve the quality and consistency of the results obtained.

The objective of this Final Degree Project is to analyze, design and theoretically implement a framework of perimeter reconnaissance tools and report generation oriented to Red Team audit tasks. To achieve this, existing tools and techniques in the field of perimeter reconnaissance will be investigated, the needs and requirements for a perimeter reconnaissance framework will be identified, and it will implement a system that integrates the tools and processes necessary to carry out a perimeter reconnaissance in an efficient and effective manner.

Key words: Reconnaissance, Discovery, Vulnerability, Analisys

Índice general

Índice de figuras

CAPÍTULO 1
Introducción

En el panorama actual de la ciberseguridad, las organizaciones y empresas enfrentan desafíos cada vez mayores para proteger sus activos digitales e información confidencial. Con la creciente complejidad y sofisticación de los ciberataques, es esencial contar con medidas de seguridad efectivas y adaptativas para enfrentar estas amenazas. Las auditorías Red Team [1] se han convertido en una práctica valiosa para evaluar y mejorar las defensas de las organizaciones al simular ataques cibernéticos reales y probar la eficacia de las medidas de seguridad implementadas.

El reconocimiento perimetral es un componente clave en cualquier auditoría Red Team, ya que se enfoca en identificar y analizar los sistemas, servicios y vulnerabilidades presentes en la red de una organización. Esta información es crucial para determinar el nivel de riesgo y exposición, así como para planificar y ejecutar ataques simulados que evalúen las capacidades de defensa de la organización. Para llevar a cabo el reconocimiento perimetral, los profesionales de la seguridad emplean diversas herramientas y técnicas que pueden variar en alcance, funcionalidad y eficacia.

En este contexto, el desarrollo de un *framework* de herramientas de reconocimiento perimetral y generación de informes que facilite la realización de auditorías Red Team se vuelve fundamental para mejorar la eficiencia y efectividad de los profesionales de la seguridad en la evaluación de la postura de seguridad de una organización. Un *framework* de este tipo permitiría la integración y automatización de las herramientas y procesos involucrados en el reconocimiento perimetral, lo que mejoraría la calidad y coherencia de los resultados obtenidos.

1.1 Objetivos del TFG.

El objetivo de este Trabajo de Fin de Grado (TFG) es analizar, diseñar e implementar teóricamente un *framework* de herramientas de reconocimiento perimetral y generación de informes orientado a tareas de auditoría Red Team. Para lograr esto, se investigarán las herramientas y técnicas existentes en el ámbito del reconocimiento perimetral, se identificarán las necesidades y requerimientos para un *framework* de reconocimiento perimetral y se diseñará e implementará un sistema que integre las herramientas y procesos necesarios para llevar a cabo un reconocimiento perimetral eficiente y efectivo.

1.2 Estructura de la memoria

Esta memoria se encuentra estructurada en 6 bloques principales:

1. Introducción. En este capítulo se proporciona un resumen del contexto actual y las necesidades a cubrir en las fases iniciales en una auditoria de Red Team, así como los objetivos del trabajo.

2. Estado del arte. Aquí se proporciona una visión general de las herramientas relevantes para la realización del trabajo, en particular lo referente a la seguridad de la red, las auditorías Red Team y las herramientas existentes para el reconocimiento perimetral.

3. Análisis y diseño del *framework*. Este capítulo esta centrado en la conceptualización y diseño del *framework*. Se encuentra dividido en 4 secciones:

    - Sección 3.1. Se identifican los requerimientos y necesidades para el *framework*
    - Sección 3.2. Se realiza una comparación de las herramientas existentes, incluyendo sus fortalezas y debilidades. Cada subapartado se centra en una herramienta específica.
    - Sección 3.3. Se justifica la seleccion de las herramientas más adecuadas para la integración en el *framework*.
    - Sección 3.4. Se presenta el diseño del *framework* propuesto.

4. Implementación del *framework*. Este capítulo cubre la implementación teórica del *framework*. Se encuentra dividido en 4 secciones:

    - Sección 4.1. Se detalla cómo se realiza la implementación del módulo de reconocimiento perimetral externo.
    - Sección 4.2. Se describe cómo se lleva a cabo la implementación del módulo de reconocimiento perimetral interno y el análisis de vulnerabilidades.
    - Sección 4.3. Se aborda la implementación del mecanismo de captura y análisis de datos.

- Sección 4.4. Se explica cómo se implementará la generación de informes.

5. Casos de uso. Este capítulo proporciona ejemplos de cómo el *framework* puede ser aplicado en entornos reales. Cada subsección expone un caso de uso específico.

6. Conclusiones. Este último capítulo resume los hallazgos del trabajo y subraya su importancia y relevancia. Se discuten las ventajas del *framework* propuesto y cómo podría mejorar la seguridad de las redes.

CAPÍTULO 2
Estado del arte

En este capítulo se procederá a enumerar brevemente las diferentes herramientas y *frameworks* comúnmente empleados hasta la fecha, mediante la integración de los cuales se podrá realizar el diseño de un nuevo *framework* para satisfacer todas las necesidades propias de las primeras fases de auditoría Red Team.

Estos *frameworks* se utilizan para realizar análisis de seguridad en redes y sistemas, y son esenciales para las tareas de auditoría Red Team. A continuación, se presentan algunos de los principales *frameworks* y herramientas utilizadas en el reconocimiento perimetral.

- NMAP (Network Mapper)(https://nmap.org/): Es una de las herramientas de exploración de red más conocidas y utilizadas en la comunidad de seguridad. NMAP permite a los usuarios descubrir dispositivos y servicios en una red, proporcionando información sobre el sistema operativo, puertos abiertos, servicios en ejecución y posibles vulnerabilidades. [4]

- Metasploit *framework*(https://www.metasploit.com/): Metasploit es un *framework* de penetración ampliamente utilizado que ofrece a los profesionales de la seguridad una gran cantidad de herramientas y módulos para realizar pruebas de penetración en redes y sistemas. Metasploit incluye capacidades de reconocimiento perimetral, como el escaneo de puertos y la identificación de servicios y versiones. [3, 5]

- Nessus(https://es-la.tenable.com/products/nessus): Nessus es un escáner de vulnerabilidades muy popular que permite a los usuarios evaluar la seguridad de sus sistemas y redes. Nessus puede realizar reconocimiento perimetral al escanear puertos, detectar vulnerabilidades conocidas, identificar configuraciones incorrectas y mucho más.

- OpenVAS (Open Vulnerability Assessment System)(https://openvas.org/): OpenVAS es un *framework* de escaneo de vulnerabilidades de código abierto que se utiliza para evaluar la seguridad de redes y sistemas. OpenVAS incluye una serie de herramientas de reconocimiento perimetral, como escaneo de puertos y servicios, análisis de vulnerabilidades y generación de informes.

- ZMap(https://zmap.io/): ZMap es un escáner de red de código abierto que permite a los usuarios realizar escaneos de red rápidos y exhaustivos. ZMap es útil para el reconocimiento perimetral, ya que puede identificar rápidamente dispositivos y servicios en una red.
- Recon-ng(https://github.com/lanmaster53/recon-ng): Recon-ng es un *framework* de reconocimiento de código abierto diseñado específicamente para realizar tareas de reconocimiento perimetral. Recon-ng incluye una serie de módulos y herramientas que pueden ayudar a los profesionales de la seguridad a obtener información sobre redes y sistemas, como la identificación de dispositivos, la recopilación de información sobre dominios y la enumeración de servicios.
- Shodan(https://www.shodan.io/): Shodan es un motor de búsqueda que permite a los usuarios buscar dispositivos, servicios y sistemas conectados a Internet. Shodan puede ser utilizado para realizar reconocimiento perimetral al proporcionar información sobre dispositivos, puertos, servicios y vulnerabilidades en una red. [7]

En resumen, los *frameworks* y herramientas de reconocimiento perimetral mencionados aquí son esenciales en el análisis, diseño e implementación de sistemas de seguridad y auditoría Red Team. Estas herramientas ayudan a los profesionales de la seguridad informática a identificar y evaluar la seguridad de las redes y sistemas, lo que es fundamental para protegerlos contra posibles amenazas y vulnerabilidades.

Más adelante, en la sección 3.2 se realizará un análisis exhaustivo de cada herramienta, detallando sus ventajas e inconvenientes.

CAPÍTULO 3
Análisis y diseño del *framework*

3.1 Identificación de los requerimientos y necesidades

Para desarrollar un *framework* eficiente y efectivo de herramientas de reconocimiento perimetral y generación de informes orientado a tareas de auditoría Red Team, es fundamental identificar los requerimientos y necesidades específicas que debe cumplir dicho *framework*.

A continuación, se enumeran algunos de los principales requerimientos y necesidades identificados:

- Integración de herramientas: El *framework* debe ser capaz de integrar diversas herramientas de reconocimiento perimetral y generación de informes, permitiendo a los profesionales de la seguridad utilizar y combinar diferentes técnicas y enfoques para evaluar la postura de seguridad de una organización.
- Automatización de procesos: El *framework* debe facilitar la automatización de los procesos de reconocimiento perimetral y generación de informes, reduciendo la necesidad de intervención manual y aumentando la eficiencia y consistencia de los resultados obtenidos.
- Flexibilidad y escalabilidad: El *framework* debe ser lo suficientemente flexible y escalable como para adaptarse a diferentes entornos y necesidades de las organizaciones, permitiendo agregar, modificar o eliminar herramientas y funcionalidades según las necesidades específicas de cada auditoría Red Team.
- Compatibilidad e interoperabilidad: El *framework* debe ser compatible e interoperable con diferentes sistemas operativos, plataformas y tecnologías, facilitando su implementación y uso en una amplia variedad de entornos y redes.
- Facilidad de uso y configuración: El *framework* debe ser fácil de usar y configurar, permitiendo a los profesionales de la seguridad personalizar y adaptar las herramientas y procesos de reconocimiento perimetral y generación

de informes a las necesidades específicas de cada organización y auditoría Red Team.

- Seguridad y privacidad: El *framework* debe garantizar la seguridad y privacidad de la información recolectada y generada durante el proceso de reconocimiento perimetral y generación de informes, cumpliendo con las regulaciones y estándares de la industria en materia de ciberseguridad y protección de datos.
- Generación de informes y visualización de resultados: El *framework* debe incluir funcionalidades para generar informes detallados y fácilmente comprensibles que muestren los resultados del reconocimiento perimetral, así como herramientas de visualización que permitan a los profesionales de la seguridad analizar y comprender rápidamente los riesgos y vulnerabilidades identificados en la red de una organización.

Estos requerimientos y necesidades representan aspectos clave que deben ser considerados en el diseño e implementación del *framework* propuesto. Al abordar estos aspectos de manera efectiva, el *framework* desarrollado podrá proporcionar a los profesionales de la seguridad una solución integral y eficiente para llevar a cabo tareas de reconocimiento perimetral y generación de informes en el contexto de auditorías Red Team.

3.2 Comparación de las herramientas existentes

Para seleccionar de manera apropiada las herramientas a implementar en el *framework*, se ha realizado un análisis exhaustivo de las herramientas destacando los puntos fuertes y débiles de cada una de ellas:

3.2.1. NMAP [4]

- Ventajas

  1. Código abierto: Como una herramienta de código abierto, NMAP es completamente gratuito para usar y su código fuente está disponible para su revisión y modificación, lo que permite a los usuarios personalizarlo según sus necesidades.
  2. Potente y versátil: NMAP es extremadamente potente y flexible. Puede realizar una variedad de tareas, incluyendo escaneo de puertos, detección de servicios y versiones, detección de sistemas operativos, detección de vulnerabilidades y más.
  3. Extensible a través de *scripts*: NMAP permite la utilización de *scripts* a través del *NMAP scripting Engine (NSE)*, lo que amplía enormemente su funcionalidad y permite a los usuarios crear sus propios *scripts* para realizar tareas específicas.
  4. Documentación exhaustiva: NMAP tiene una gran cantidad de documentación disponible, lo que puede facilitar el aprendizaje y la resolución de problemas.
  5. Ampliamente utilizado y reconocido: NMAP es ampliamente utilizado en la industria de la ciberseguridad, lo que significa que hay una gran comunidad de usuarios y una gran cantidad de recursos de aprendizaje disponibles.

- Desventajas

  1. Curva de aprendizaje empinada: NMAP tiene una gran cantidad de opciones y características, y aprender a usarlo efectivamente puede llevar tiempo. Para los usuarios no iniciados, emplear la línea de comandos y comprender y memorizar la cantidad de opciones que posee puede ser una tarea realmente tediosa.
  2. Interfaz de línea de comandos: Aunque la interfaz de línea de comandos proporciona una gran flexibilidad, puede ser intimidante para los usuarios menos técnicos. Sin embargo, NMAP ofrece una interfaz gráfica de usuario llamada ZeNMAP, visible en la figura 3.1, que puede facilitar su uso.
  3. Velocidad: Los escaneos de NMAP pueden ser lentos, especialmente si se están escaneando un gran número de puertos o si se están utilizando técnicas de escaneo más completas.

4. Posibles falsos positivos y falsos negativos: Como con cualquier herramienta de escaneo de red, NMAP puede producir resultados falsos positivos o falsos negativos. Los resultados del escaneo deben ser verificados y requieren de un correcto análisis para tomar decisiones que impacten a la seguridad.

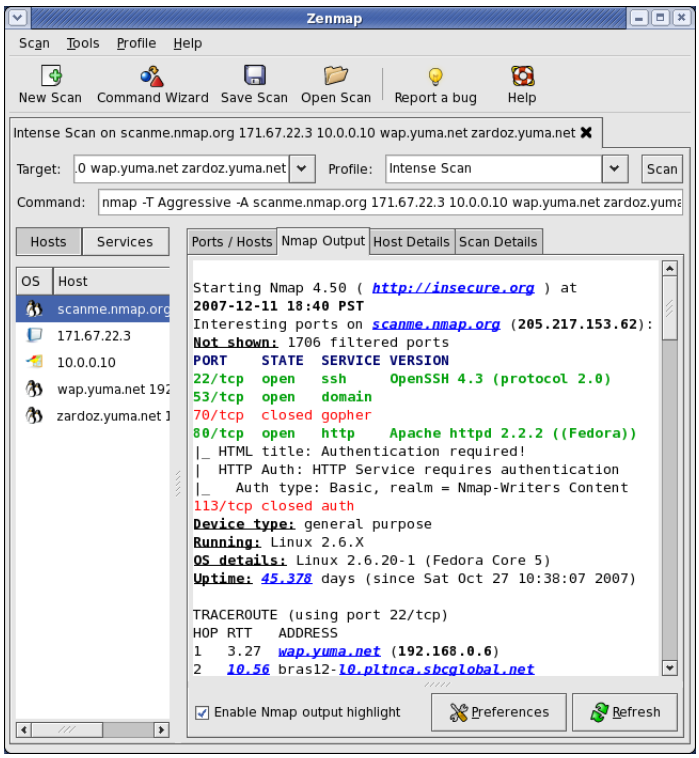

Figura 3.1: Zenmap GUI

3.2.2. Metasploit *framework* [3]

- Ventajas

1. Código abierto: Metasploit es una herramienta de código abierto, lo que significa que es gratuita para usar y su código es accesible para revisión y modificación.
2. Amplia gama de módulos de *exploits*: Un *exploit* es un fragmento de código o secuencia de comandos o acciones empleada con la finalidad de aprovechar una vulnerabilidad en un sistema determinado para lograr un comportamiento diferente al cual ha sido diseñado. Metasploit tiene una amplia base de datos de *exploits* que se actualiza con regularidad, lo que lo hace muy efectivo para encontrar y explotar vulnerabilidades.
3. Automatización: Metasploit permite la automatización de ciertas tareas de pruebas de penetración, lo que puede ahorrar tiempo y esfuerzo.
4. Integración con otras herramientas: Metasploit se integra bien con una variedad de otras herramientas de seguridad, lo que puede facilitar un flujo de trabajo de pruebas de penetración más eficiente.

5. Amplia comunidad: Metasploit tiene una gran comunidad de usuarios y desarrolladores que contribuyen regularmente con nuevos módulos y actualizaciones.

- Desventajas

1. Curva de aprendizaje empinada: A pesar de su poder, Metasploit puede ser bastante complejo de usar para los principiantes, especialmente aquellos que no están familiarizados con el lenguaje de programación Ruby, que se utiliza para escribir módulos para Metasploit.
2. Posibles detecciónes por medidas de seguridad perimetral: Aunque Metasploit contiene una variedad de *payloads* y técnicas de evasión, algunos de sus *exploits* y *payloads* pueden ser fácilmente detectados por sistemas modernos de detección de intrusos y software antivirus. Cabe destacar que mientras un *exploit* hace referencia al fragmento de código completo que engloba todos los pasos necesarios para explotar una vulnerabilidad, un *payload* es únicamente la carga útil, es decir, aquella parte de la comunicación que debe ser interpretada por el sistema sujeto a explotación.
3. Interfaz de línea de comandos: Aunque Metasploit tiene una interfaz gráfica (Metasploit Community Edition), la mayoría de las funcionalidades avanzadas se encuentran en la interfaz de línea de comandos, como puede observarse en la figura 3.2, lo cual puede ser intimidante para los usuarios menos técnicos.
4. Limitaciones en la versión gratuita: Aunque Metasploit *framework* es de código abierto, Rapid7 también ofrece versiones comerciales de la herramienta (como Metasploit Pro) que contienen funcionalidades adicionales no disponibles en la versión gratuita.

Figura 3.2: Interfaz de línea de comandos de Metasploit *Framework*

3.2.3. Recon-NG

- Ventajas

  1. Modularidad: Recon-ng se basa en una arquitectura modular, lo que permite a los usuarios agregar y actualizar módulos para expandir sus capacidades. Esto también significa que si una parte de la herramienta se vuelve obsoleta, puede ser actualizada sin afectar al resto de la herramienta.
  2. Amplia gama de módulos: Recon-ng viene con una amplia gama de módulos que permiten la recolección de información de una variedad de fuentes. Esto incluye módulos para la recolección de datos de redes sociales, bases de datos de WHOIS, servicios de geolocalización y más.
  3. Interfaz de línea de comandos: Recon-ng tiene una interfaz de línea de comandos intuitiva y fácil de usar que se asemeja a la de Metasploit, una herramienta con la que muchos profesionales de la seguridad están familiarizados, tal y como muestra la figura 3.3.
  4. Búsqueda automatizada: Recon-ng puede automatizar la búsqueda de información en varias fuentes, lo que puede ahorrar tiempo y esfuerzo en comparación con la búsqueda manual de información.

- Desventajas

  1. Curva de aprendizaje: Recon-ng puede ser un poco intimidante para los nuevos usuarios debido a la gran cantidad de módulos y opciones disponibles.
  2. Dependencia de las APIs: Muchos módulos de Recon-ng dependen de las APIs de varios servicios en línea. Si estos servicios cambian sus APIs o las descontinúan, los módulos correspondientes pueden dejar de funcionar hasta que sean actualizados.
  3. Falta de interfaz gráfica: Recon-ng es una herramienta de línea de comandos, lo que puede ser desafiante para los usuarios menos técnicos que prefieren una interfaz gráfica.
  4. Límites de la API: Algunos módulos pueden estar sujetos a los límites de la API del servicio que están utilizando, lo que puede limitar la cantidad de información que pueden recopilar en un período de tiempo determinado.

Figura 3.3: Interfaz de línea de comandos de Recon-ng

3.2.4. Nessus

- Ventajas

  1. Amplia base de datos de vulnerabilidades: Nessus tiene una de las bases de datos de vulnerabilidades más grandes y actualizadas del mercado, lo que la hace muy efectiva para descubrir vulnerabilidades conocidas en una red.
  2. Interfaz de usuario intuitiva: Nessus ofrece una interfaz de usuario gráfica fácil de usar que permite a los usuarios configurar y ejecutar escaneos de manera más sencilla, como se muestra en la figura 3.4.
  3. Informes detallados: Los informes generados por Nessus son muy detallados y proporcionan descripciones comprensibles de las vulnerabilidades detectadas, los riesgos asociados, y recomendaciones de remediación.
  4. Escaneos programados: Nessus permite programar escaneos a intervalos regulares, lo que facilita el monitoreo continuo de la seguridad de la red.
  5. Integración con otras herramientas: Nessus se integra bien con una variedad de otras herramientas de seguridad y sistemas de gestión de información y eventos de seguridad (SIEM), lo que permite un flujo de trabajo de seguridad más eficiente.

- Desventajas

  1. Costo: Aunque Nessus ofrece una versión gratuita con funcionalidad limitada, las características más avanzadas requieren la compra de una licencia, que puede ser costosa para algunas organizaciones.
  2. Complejidad: A pesar de su interfaz de usuario, Nessus todavía puede ser complejo de usar para los usuarios menos técnicos, especialmente cuando se trata de la interpretación de los resultados del escaneo.
  3. Falsos positivos: Como con cualquier herramienta de escaneo de vulnerabilidades, Nessus puede generar falsos positivos que pueden requerir tiempo y esfuerzo adicionales para validar.
  4. Limitaciones de la versión gratuita: La versión gratuita de Nessus, Nessus Essentials, tiene limitaciones significativas, incluyendo una restricción en el número de IPs que pueden ser escaneadas.

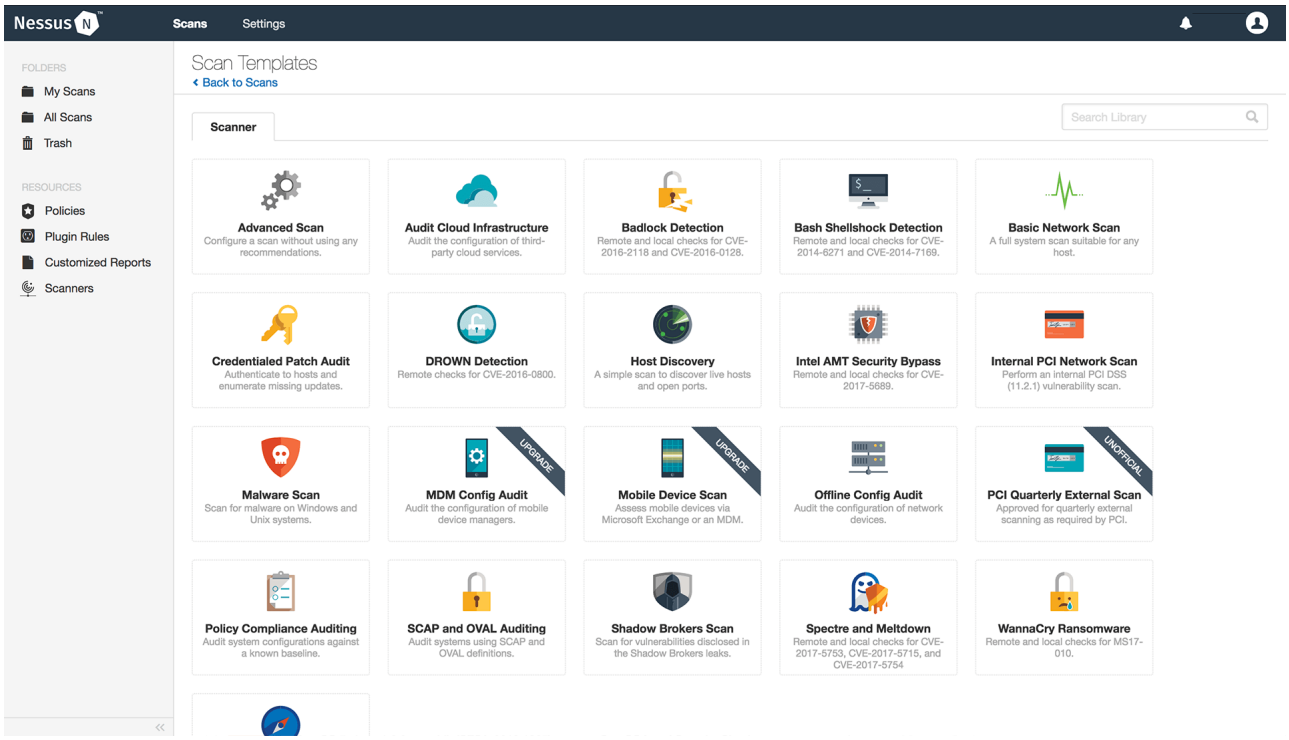

Figura 3.4: Interfaz gráfica de Nessus (selector de plantillas de escaneo)

3.2.5. Wireshark [8]

- Ventajas

  1. Código abierto: Wireshark es una herramienta de código abierto, lo que significa que es gratuita para usar y su código es accesible para revisión y modificación.
  2. Análisis profundo de protocolos: Wireshark permite a los usuarios capturar y analizar el tráfico de la red a nivel de paquetes, proporcionando una visión detallada de lo que está ocurriendo en la red.
  3. Soporte para múltiples protocolos: Wireshark puede interpretar y decodificar una amplia gama de protocolos de red, lo que lo hace muy versátil.
  4. Interfaz de usuario gráfica: A pesar de ser una herramienta técnica, Wireshark ofrece una interfaz de usuario gráfica que facilita la visualización y el análisis de los datos capturados. Dicha interfaz se muestra en la figura 3.5
  5. Filtrado: Wireshark proporciona potentes capacidades de filtrado que permiten a los usuarios centrarse en aspectos específicos del tráfico de la red.

- Desventajas

  1. Curva de aprendizaje empinada: Aprender a usar Wireshark efectivamente puede ser un desafío, especialmente para aquellos que no están familiarizados con los protocolos de red y el análisis de paquetes.
  2. Requiere acceso físico: Para capturar tráfico de red, Wireshark necesita ser ejecutado en una máquina que tenga acceso a la red que se está analizando, lo que puede limitar su utilidad en ciertos escenarios.
  3. Problemas de privacidad: Capturar y analizar el tráfico de red puede tener implicaciones de privacidad, ya que puede incluir la interceptación de datos sensibles.
  4. Volumen de datos: Wireshark puede generar una gran cantidad de datos, especialmente cuando se captura tráfico de red durante un período de tiempo prolongado. Esto puede hacer que el análisis de los datos sea abrumador y consume mucho tiempo.

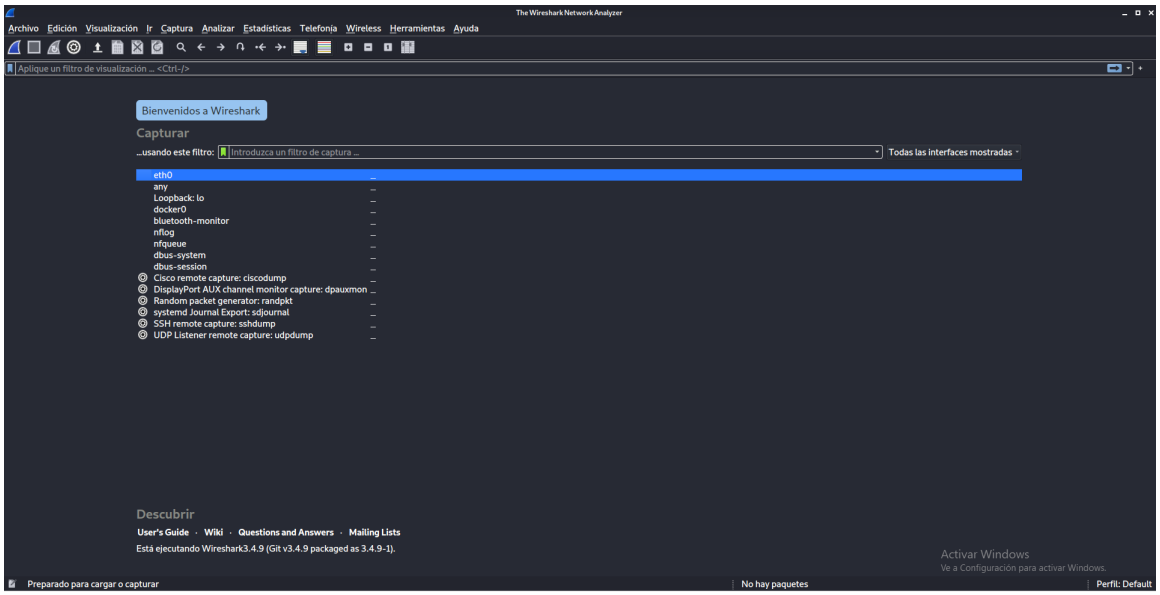

Figura 3.5: Interfaz gráfica de Wireshark

3.2.6. OpenVAS

- Ventajas

  1. Código abierto: OpenVAS es una herramienta de código abierto, lo que significa que es gratuita para usar y su código es accesible para revisión y modificación.
  2. Extensa base de datos de pruebas de vulnerabilidades: OpenVAS cuenta con una amplia gama de pruebas de vulnerabilidades que se actualizan regularmente, permitiendo la detección de una variedad de vulnerabilidades conocidas.
  3. Interfaz de usuario gráfica: Aunque OpenVAS puede ser usado a través de la línea de comandos, también ofrece una interfaz de usuario gráfica que puede facilitar la configuración y ejecución de escaneos, tal y como se ejemplifica en la figura 3.6.
  4. Escaneos programados: OpenVAS permite la programación de escaneos, lo que puede facilitar el monitoreo regular de la seguridad de la red.
  5. informes detallados: OpenVAS genera informes detallados de las vulnerabilidades encontradas, proporcionando información sobre el riesgo y las posibles soluciones.

- Desventajas

  1. Instalación y configuración complejas: Aunque es una herramienta poderosa, OpenVAS puede ser bastante complicado de instalar y configurar correctamente, especialmente para los usuarios menos técnicos.
  2. Escaneos lentos: Algunos usuarios han informado que los escaneos de OpenVAS pueden ser relativamente lentos, especialmente en comparación con algunas otras herramientas de escaneo de vulnerabilidades.
  3. Falsos positivos: Como con cualquier herramienta de escaneo de vulnerabilidades, OpenVAS puede generar falsos positivos. Esto significa que los resultados del escaneo pueden requerir una validación adicional.
  4. Interfaz de usuario menos pulida: Aunque OpenVAS tiene una interfaz de usuario gráfica, algunos usuarios consideran que no es tan intuitiva o fácil de usar como las de algunas otras herramientas de escaneo de vulnerabilidades comerciales

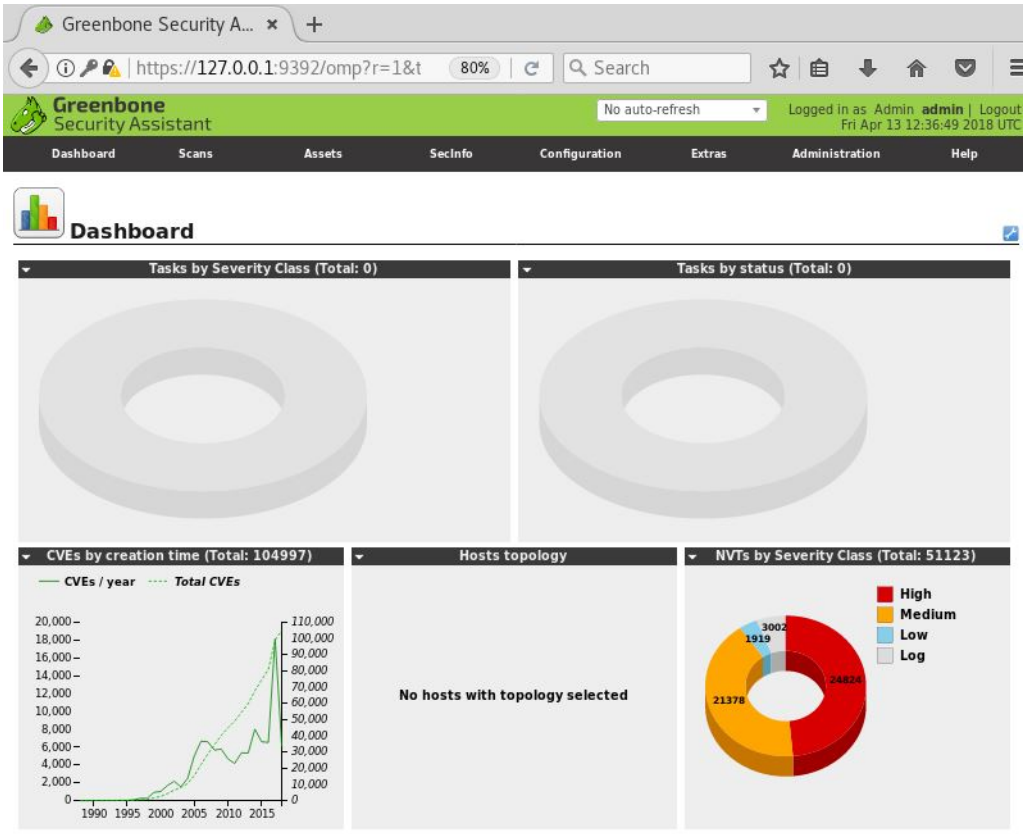

Figura 3.6: Interfaz gráfica de OpenVAS

3.2.7. Shodan [7]

- Ventajas

  1. Amplia cobertura: Shodan puede escanear y indexar una amplia gama de dispositivos conectados a Internet, desde computadoras y servidores hasta dispositivos IoT, sistemas de control industrial, y más.
  2. Fácil de usar: Shodan tiene una interfaz web fácil de usar, como se muestra en la figura 3.7, que permite a los usuarios realizar búsquedas complejas sin necesidad de escribir comandos de línea de comandos.
  3. Información detallada: Shodan proporciona información detallada sobre los dispositivos que encuentra, incluyendo el sistema operativo, los servicios en ejecución, las vulnerabilidades conocidas, la ubicación geográfica, y más.
  4. Información potencialmente sensible: Shodan puede revelar información potencialmente sensible sobre dispositivos que pueden estar mal configurados o no estar adecuadamente protegidos.
  5. API disponible: Shodan proporciona una API que el usuario puede utilizar para automatizar búsquedas y analizar los resultados de las búsquedas en programas de terceros.

- Desventajas

  1. Costo: Mientras que Shodan ofrece un nivel básico de servicio de forma gratuita, muchas de sus características más avanzadas requieren una suscripción de pago.
  2. Actualizaciones de datos: Los datos de Shodan se actualizan regularmente, pero no en tiempo real, lo cual significa que la información obtenida a través de Shodan puede estar desactualizada.
  3. Interpretación de resultados: Aunque Shodan proporciona mucha información, puede requerir un cierto nivel de conocimiento técnico para interpretar los resultados de las búsquedas y comprender su relevancia.

+

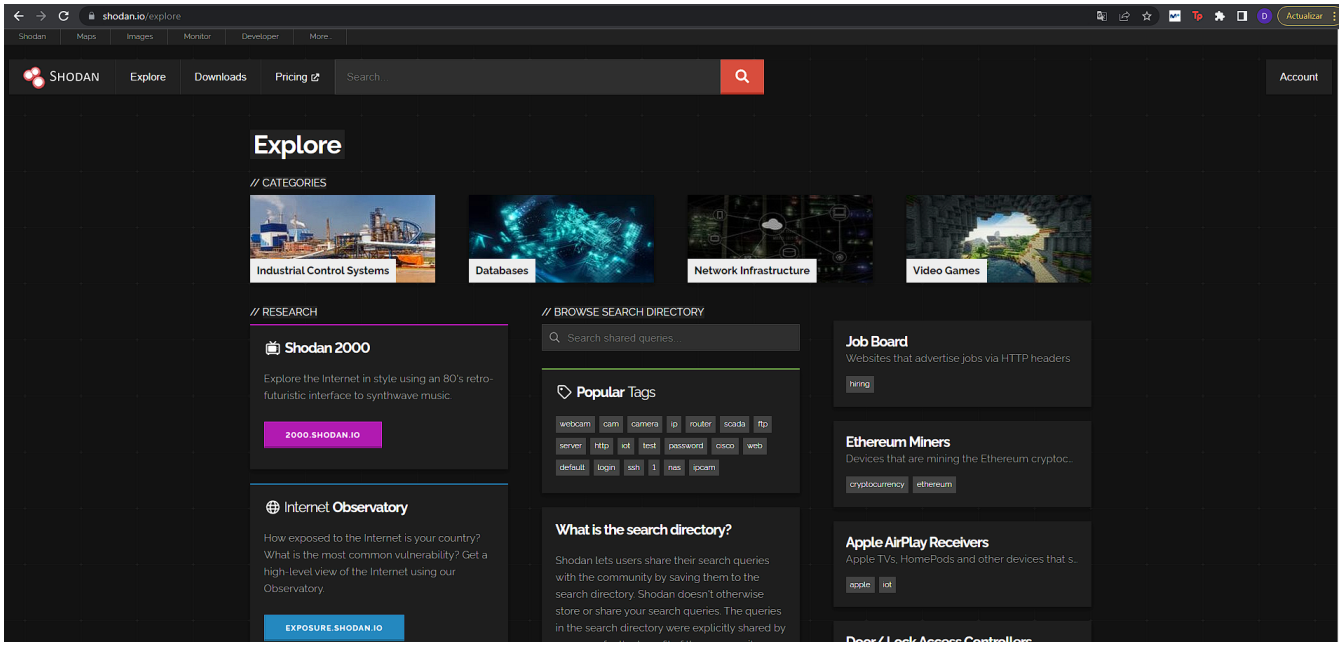

Figura 3.7: Explorador de la interfaz gráfica de Shodan

3.2.8. Nuclei [6]

- Ventajas

  1. Rápida y eficiente: Nuclei es conocida por su velocidad y eficiencia. Puede realizar un escaneo rápido de vulnerabilidades, lo que facilita su uso en grandes redes.
  2. Plantillas personalizables: Nuclei proporciona un marco de trabajo para escribir tus propias plantillas, lo que permite personalizar los escaneos según las necesidades específicas de tu red.
  3. Base de fatos de plantillas: Nuclei viene con una base de datos actualizada de plantillas que cubren una amplia gama de vulnerabilidades conocidas.
  4. Interfaz de línea de comandos simple: Nuclei tiene una interfaz de línea de comandos que es fácil de usar y entender, lo cual se puee observar en la figura 3.8.
  5. Integración con otras herramientas: Nuclei se puede integrar con otras herramientas populares de seguridad y pruebas de penetración.[6]

- Desventajas

  1. Curva de aprendizaje: Aunque Nuclei tiene una interfaz de línea de comandos simple, puede llevar algún tiempo aprender a usar la herramienta de manera eficaz, especialmente cuando se trata de escribir y personalizar las plantillas.
  2. Falsos positivos: Como cualquier herramienta de escaneo de vulnerabilidades, Nuclei puede generar falsos positivos. Esto significa que los resultados del escaneo pueden requerir una validación adicional.

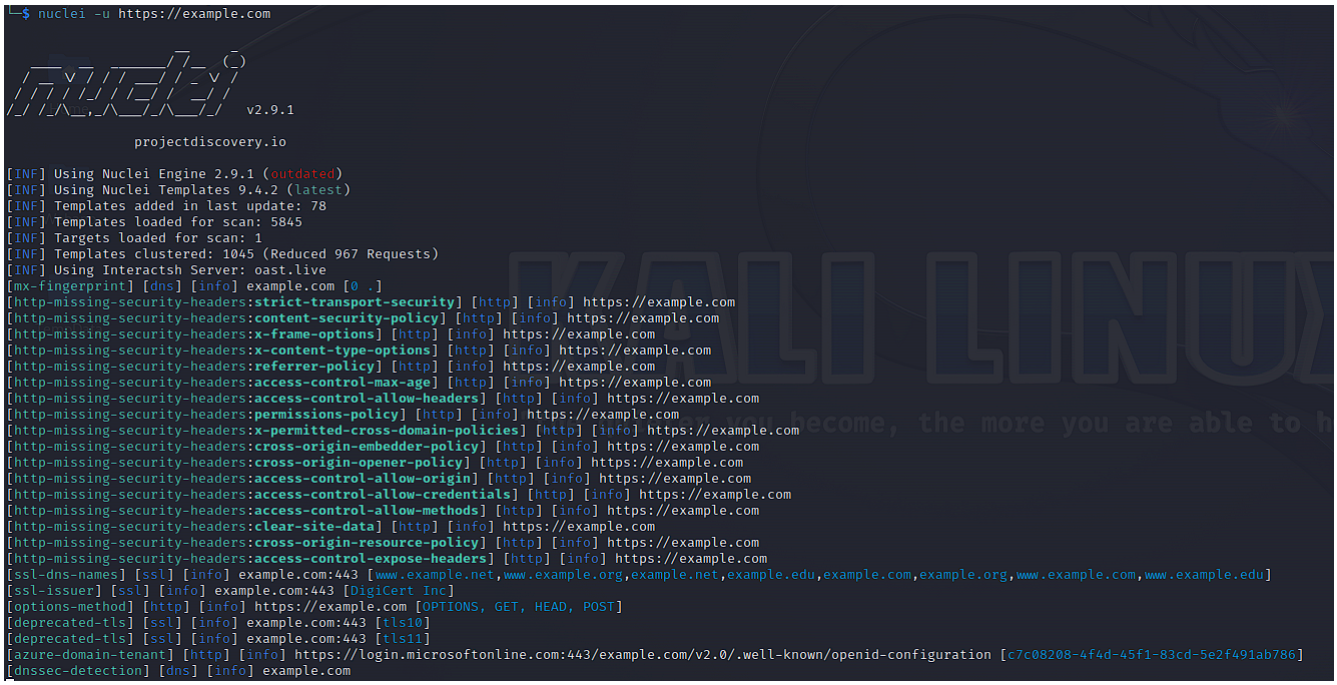

Figura 3.8: Ejemplo uso Nuclei en interfaz de línea de comandos

3.3 Herramientas seleccionadas para el *framework*

Para la seleccion de herramientas empleadas en la implementación del *framework*, se ha tenido en cuenta que, por lo general, podemos diferenciar la superficie de ataque entre interna y externa, y es importante focalizar las comprobaciones más pertinentes en cada caso, ya que la granularidad de los escaneos puede variar entre la red interna y la externa. Además, hay que ser especialmente cautelosos escaneando activos expuestos a internet, ya que en algunos casos podríamos encontrarnos causando algun tipo de disrupción en el servicio de alguna compañía externa, enfrentando así posibles demandas por daños y perjuicios.

De este modo, la selección de estas herramientas tiene varias justificaciones basadas en sus características individuales y cómo pueden complementarse entre sí para llevar a cabo una auditoría Red Team completa y eficaz. Además, todas ellas son de código abierto, por lo que podemos tener control total sobre el comportamiento de cada módulo implementado.

En base a lo expuesto hasta ahora, la selección de herramientas se ha llevado a cabo diferenciando la superficie sujeta a análisis de la siguiente manera:

- Superficie interna
  - NMAP: Esta herramienta se selecciona debido a su versatilidad y eficiencia en la exploración de redes. Puede identificar dispositivos activos, puertos abiertos, servicios en ejecución y sus versiones, lo cual es crucial para mapear el entorno de la red interna y descubrir posibles puntos de entrada.
  - Metasploit *framework*: Metasploit se selecciona para la explotación y post-explotación. Gracias a su amplia base de datos de *exploits*, esta herramienta puede ayudar a aprovechar las vulnerabilidades descubiertas por NMAP. También proporciona capacidades de post-explotación para profundizar en el entorno comprometido, lo que puede ser crucial para la fase de movimiento lateral de la auditoría. [2]
  - Wireshark: Esta herramienta de análisis de protocolos de red se utiliza para inspeccionar el tráfico de la red interna en un nivel de detalle más granular. Puede ayudar a identificar anomalías, capturar intentos de exfiltración de datos y proporcionar una visión más profunda de las actividad que tiene lugar en la red.
- Superficie externa
  - Shodan: Es un motor de búsqueda que escanea Internet en busca de dispositivos conectados y sus servicios asociados. Esto permite a los auditores Red Team descubrir componentes de la superficie de ataque externa de la organización que podrían ser vulnerables a la explotación. Shodan proporciona una visión de alto nivel de la presencia de una organización en Internet.
  - Nuclei: Esta herramienta se utiliza para un escaneo rápido de vulnerabilidades en la superficie de ataque externa identificada por Shodan.

Nuclei utiliza plantillas personalizables para buscar una amplia variedad de vulnerabilidades y errores de configuración. Esto permite un escaneo más específico y dirigido que puede descubrir vulnerabilidades que Shodan puede pasar por alto.

3.4 Diseño

La implementación del *framework* será comprendida por cuatro módulos empleables de manera independiente, empleando como lenguaje base Python gracias a su gran versatilidad.

A continuación se describen las herramientas empleadas para cada módulo a implementar:

1. Reconocimiento perimetral externo

   - Shodan: Empleada para descubrir y enumerar dispositivos o sistemas expuestos en Internet en base a los parámetros definidos por el usuario. Esta información puede incluir servidores web, bases de datos, dispositivos IoT, etc., que serán empleados para lanzar el escaneo de vulnerabilidades automáticamente.
   - Nuclei: Utilizado para realizar un escaneo rápido de las vulnerabilidades en la superficie de ataque externa identificada por Shodan. Puede identificar vulnerabilidades específicas y problemas de configuración que pueden ser explotados.

2. Reconocimiento perimetral interno y análisis de vulnerabilidades

   - NMAP: Una vez definidos los objetivos por el usuario (lo cual puede ser proporcionado como un fichero en formato CSV de entrada como direcciones IP únicas o rangos CIDR), NMAP se utiliza para realizar un escaneo de descubrimiento y un análisis más detallado de los puertos y servicios abiertos en los sistemas identificados.
   - Metasploit: Posteriormente, se utiliza Metasploit para explotar las vulnerabilidades identificadas. Esto puede implicar la ejecución de código, la elevación de privilegios, o la creación de una puerta trasera para un acceso más profundo. En caso de detección positiva, se captura el *payload* enviado mediante Wireshark como evidencia de la detección.

3. Captura y análisis de datos

   - Wireshark: Durante la fase de explotación, Wireshark puede ser utilizado para capturar y analizar el tráfico de la red. Esto puede proporcionar información valiosa sobre los datos que se están transmitiendo, así como sobre cualquier anomalía o actividad sospechosa; además, puede aportar evidencias del tráfico real en la explotación de vulnerabilidades.

4. Generación de informes

   Todas las actividades y hallazgos son documentados durante cada fase del proceso, almacenando el registro en ficheros CSV. Esta documentación puede luego ser consolidada en un informe que proporciona detalles sobre las vulnerabilidades descubiertas, la eficacia de los controles de seguridad existentes, y las recomendaciones para mitigar las vulnerabilidades y mejorar la seguridad. Dicho informe podrá ser exportado en formato XLSX con tablas

que permitan focalizar de manera ordenada la información más relevante, o bien en formato PDF para generar un informe de carácter ejecutivo.

La fase de generación de informes se realizará principalmente en Python dada la simplicidad para tratar los *outputs* generados por los comandos lanzados por cada herramienta de cara a la generación de registros en CSV.

Para la generación de ficheros XLSX y aplicación de macros para la generación de las diferentes tablas en base a las plantillas predefinidas (o proporcionada por el usuario), se empleará PowerShell, dada la simplicidad de trabajar con esta clase de documentos empleando dicha herramienta

CAPÍTULO 4
Implementación del *framework*

Para analizar en detalle la implementación del *framework*, analizaremos la implementación de cada módulo de manera independiente.

Todas las dependencias serán instaladas mediante un archivo de instalación inicial.

4.1 Reconocimiento perimetral externo

Primero, se requerirá instalar y configurar las bibliotecas y módulos necesarios en Python. A continuación, podemos usar estos módulos para interactuar con las APIs de Shodan y realizaremos las llamadas a Nuclei mediante la creación de subprocesos en Python.

Afortunadamente, Shodan cuenta con su propia biblioteca en Python, lo cual permite interactuar con la herramienta de manera muy sencilla, únicamente se deberá configurar la clave de acceso personal.

Principalmente se emplearán los siguientes métodos disponibles en la biblioteca de Shodan para la detección de posibles activos de nuestro interés:

- `search(query, kwargs)` : Este método permite realizar búsquedas en Shodan utilizando una cadena de consulta. Devuelve un diccionario que contiene información sobre los resultados de la búsqueda.
- `search_cursor(query, kwargs)` : Este método permite iterar sobre los resultados de la búsqueda. Es útil para las consultas que devuelven una gran cantidad de resultados.

Para recopilar toda la información disponible sobre cada dispositivo resultante de la búsqueda, se ejecutarán dos procesos en paralelo, utilizando como entrada cada IP obtenida en el paso anterior:

1. Se empleará el siguiente metodo de la biblioteca de Shodan para recopilar toda la información disponible de cada activo en su base de datos
    - `host(ip, kwargs)` : Este método recupera toda la información que Shodan tiene sobre una IP dada.
2. Se realizará un escaneo con Nuclei sobre la lista de dispositivos detectados

4.2 Reconocimiento perimetral interno y análisis de vulnerabilidades

Para la implementación de esta fase se requerirá de las bibliotecas para Python "Python-NMAP" y "pyMetasploit3"

En base a los datos de entrada proporcionados por el usuario, se lanzará un escaneo completo para identificar dispositivos activos y realizar un primer escaneo de vulnerabilidades el cual se complementará con una posterior explotación con Metasploit. Para ello emplearemos el metodo `scan` con los siguientes parámetros:

```
scan(target, '1-65535', arguments='-sV -O --script=vuln')
```

Una vez finalizado el escaneo, se emplean los datos de servicios y vulnerabilidades obtenidos del escaneo para realizar una búsqueda de posibles *exploits* que se ejecutarán sobre los dispositivos detectados

4.3 Captura y análisis de datos

Para esta fase emplearemos Wireshark para el análisis de paquetes y Python para el procesamiento y la identificación de patrones anómalos.

Para ello se lanzará mediante la instrucción `subprocess` una captura de paquetes sobre la interfaz definida por el usuario utilizando la utilidad `tshark` con lo que se generará un paquete `.pcap` que será posteriormente analizado, así como almacenado para generación de informes.

Con Python, y haciendo uso de la biblioteca `scikit-learn`, se realizará un análisis de comportamientos anómalos con la creación de un modelo de aprendizaje automático que, a medida que se emplee el *framework* en un determinado entorno, desarrollará un mayor nivel de confianza en los resultados con la ingesta de paquetes de red.

4.4 Generación de informes

En esta fase se crearán los informes que incluirán el detalle acerca de los activos descubiertos, los resultados de los escaneos, las pruebas de explotación y el análisis del tráfico de red.

Para esto contaremos con los CSV que se han ido generando en la ejecución de cada módulo y, en función de la plantilla y los parámetros seleccionados por el usuario, se realiza un procesamiento de los datos para seleccionar aquellos de interés, creando un fichero temporal que será utilizado para la aplicación de macros mediante PowerShell.

CAPÍTULO 5
Casos de uso

En esta sección se exponen diferentes casos de uso en los que el empleo de este *framework* resultaría de especial utilidad.

5.1 Auditorías de seguridad internas

Este *framework* puede ser empleado por las organizaciones para realizar auditorías de seguridad internas regulares. Esto puede ayudar a identificar vulnerabilidades y problemas de seguridad en su infraestructura antes de que sus sistemas sean explotados por actores maliciosos.

En este caso, el alcance de los escaneos vendría definido por la topología de red de la organización en cuestión, siendo especialmente cauteloso con aquellos segmentos de red que puedan albergar sistemas sensibles sobre los que un escaneo agresivo podría causar una interrupción del servicio.

A continuación, el *framework* podría ser empleado para realizar las tareas de detección de dispositivos activos, puertos abiertos y servicios en ejecución, así como generar informes de vulnerabilidades detectadas y explotables, e incluso ejecutar un análisis continuo del tráfico de red, gracias a la integración de las diferentes herramientas en uso.

Finalmente, la generación de informes permitirá a la organización generar diferentes formatos de informe de manera que se adapte de la manera más óptima posible a las necesidades y limitaciones de la audiencia a la que sea remitido.

5.2 Evaluación del estado de seguridad previo al despliegue

Antes de lanzar un nuevo sistema o aplicación (o incluso posteriormente a la implementación de cambios en la topología de red), las organizaciones pueden usar este *framework* para evaluar su estado de securización. Esto puede ayudar a identificar y corregir cualquier vulnerabilidad o problema de seguridad antes del despliegue.

En este caso, el alcance de los escaneos vendrá determinado generalmente por el número de sistemas que se vayan a desplegar o, en aquellos casos en los que se vayan a desplegar dispositivos replicados, será suficiente con escanear uno de ellos para lograr una visión general del estado de los dispositivos.

Una vez determinado el alcance, se procedería con el primer escaneo mediante NMAP (o Nuclei si el despliegue fuese en entorno *cloud*) y las posteriores comprobaciones de explotación mediante Metasploit o el enriquecimiento de los datos mediante la adición de la información disponible en la base de datos de Shodan.

5.3 Cumplimiento normativo

Algunas industrias y sectores jurisdiccionales requieren auditorías de seguridad regulares para demostrar el cumplimiento de diversas normativas y estándares de seguridad, además de la proactividad y capacidad de las diferentes organizaciones de identificar y trabajar sobre los problemas de seguridad. Además, este tipo de auditorías también pueden ser requeridas por ejemplo en el proceso de contratación de seguros de Ciber Riesgo. Este *framework* puede ser una herramienta valiosa para ayudar a las organizaciones a medir el cumplimiento con estos requisitos y poder detectar aquellos puntos a resolver para poder superar satisfactoriamente la auditoría.

En este caso, el alcance de los escaneos vendrá definido por la superficie auditada y se procedería a realizar el escaneo inicial y posterior análisis de vulnerabilidades.

CAPÍTULO 6
Conclusiones

La creciente prevalencia de las amenazas cibernéticas en la actualidad subraya la necesidad de estrategias de seguridad sólidas y robustas. Un aspecto esencial de cualquier estrategia de seguridad efectiva es la capacidad de realizar auditorías y pruebas regulares de la infraestructura de red de una organización para identificar y abordar las vulnerabilidades. Este trabajo ha explorado el diseño e implementación de un *framework* para facilitar estas auditorías, especialmente desde la perspectiva de un equipo Red Team.

En este trabajo, se ha realizado un estudio detallado del estado del arte en lo que respecta a las herramientas de reconocimiento perimetral. Se han evaluado y comparado varias herramientas, destacando sus fortalezas y debilidades, y se han seleccionado aquellas que ofrecían las mejores capacidades para su integración en el *framework* propuesto.

Se ha descrito la estructura de un *framework* enfocado a auditorías de Red Team, dividido en cuatro fases clave:

- Descubrimiento de activos
- Escaneo, enumeración y pruebas de vulnerabilidad
- Análisis de tráfico de red
- Generación de informes

En cada fase, se han empleado las herramientas seleccionadas para llevar a cabo tareas específicas, y se han proporcionado ejemplos de cómo estas herramientas podrían ser utilizadas en un contexto real.

Además, se han explorado casos de uso potenciales donde entraría en juego el uso de este *framework*, mostrando cómo podría ser útil en una variedad de contextos, desde auditorías de seguridad internas hasta la evaluación del estado de seguridad de un sistema específico previo a su depliegue, y el cumplimiento normativo.

En conclusión, la implementación de un *framework* de herramientas de reconocimiento perimetral y generación de informes tiene el potencial de ser un aporte de valor a las estrategias de ciberseguridad de las organizaciones. Al proporcionar un mecanismo útil para satisfacer las necesidades de las auditorías de seguridad, este *framework* puede ayudar a las organizaciones a identificar y abordar

las vulnerabilidades activas en sus sistemas, mejorar su estado de securización general y, en última instancia, proteger mejor sus activos y operaciones ante las amenazas cibernéticas.

El desarrollo de este Trabajo de Fin de Grado se sustenta de los conocimientos obtenidos en las diferentes asignaturas de programación (tanto más teóricas como puede ser Lenguajes, tecnologías y paradigmas de la programación, asi como las más practicas como Integración de aplicaciones) para la correcta integración entre las herramientas y construccion de módulos sólidos. Importante mencionar también los conocimientos en redes y computadores obtenidos en las asignaturas Estructura de computadores, Redes de computadores y Arquitectura e ingeniería de computadores, asi como Redes de computadores y por supuesto las asignaturas optativas enfocadas a la seguridad, de donde se obtiene una buena base para continuar la investigación y el desarrollo personal en tal materia, destacando Seguridad en redes y sistemas y Hacking Ético.

Cabe destacar que este *framework*, dada su estructura modular y gracias a estar totalmente integrado mediante el uso de herramientas de código abierto existentes, permite la ampliación de su funcionalidad en el futuro, siendo posible incorporar tantas nuevas funcionalidades como sea necesario para la integración de un *framework* más completo capaz de ejecutar y automatizar todas las acciones requeridas en las diferentes fases de un proceso de auditoría de Red Team e incluso derivar de manera automatizada los informes generados a su pertinente audiencia.

Bibliografía

[1] Seemant Sehgal. Red Teaming for Cybersecurity, *ISACA JOURNAL VOL 5*, pp 1-6, 2018.

[2] Benjamin Bowman, Craig Laprade, Yuede Ji, and H. Howie Huang, Red Teaming for Cybersecurity, *Graph Computing Lab, George Washington University*, pp 257-268, 2020.

[3] David Kennedy, Jim O'Gorman, Devon Kearns, and Mati Aharoni. *Metasploit: The Penetration Tester's Guide*. No Starch Press, Inc, 2011.

[4] Gordon "Fyodor"Lyon. *Nmap Network Scanning: The Official Nmap Project Guide*. Nmap Project; 12.2.2008 edición (1 Enero 2009)

[5] Documentación oficial Metasploit. Consultado en https://docs.rapid7.com/metasploit/manual-exploitation/.

[6] Documentación oficial Nuclei. Consultado en https://github.com/projectdiscovery/nuclei.

[7] Documentación oficial API Shodan. Consultado en https://developer.shodan.io/api.

[8] Documentación oficial de Wireshark Consultado en https://www.wireshark.org/docs/

APÉNDICE A
OBJETIVOS DE DESARROLLO SOSTENIBLE

A.1 Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenible	**Alto**	**Medio**	**Bajo**	**No procede**
ODS 1. **Fin de la pobreza.**				X
ODS 2. **Hambre cero.**				X
ODS 3. **Salud y bienestar.**				X
ODS 4. **Educación de calidad.**				X
ODS 5. **Igualdad de género.**				X
ODS 6. **Agua limpia y saneamiento.**				X
ODS 7. **Energía asequible y no contaminante.**				X
ODS 8. **Trabajo decente y crecimiento económico.**		X		
ODS 9. **Industria, innovación e infraestructuras.**	X			
ODS 10. **Reducción de las desigualdades.**				X
ODS 11. **Ciudades y comunidades sostenibles.**				X
ODS 12. **Producción y consumo responsables.**			X	
ODS 13. **Acción por el clima.**				X
ODS 14. **Vida submarina.**				X
ODS 15. **Vida de ecosistemas terrestres.**				X
ODS 16. **Paz, justicia e instituciones sólidas.**			X	
ODS 17. **Alianzas para lograr objetivos.**			X	

A.2 Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados.

El resultado del trabajo expuesto, centrado en el desarrollo de un *framework* de herramientas de reconocimiento perimetral y generación de informes, podría verse implicado de manera significativa en varias áreas que podrían verse alineadas con los ODS.

Puesto que el *framework* descrito supone un activo intangible de aplicación a entornos cibernéticos, no es posible su relación con objetivos tales como Fin de la Pobreza, Hambre Cero o Educación de Calidad, así como objetivos relativos al impacto sobre el ecosistema, donde este *framework* supondría un impacto 0 pero no se ve involucrado en la tarea de lograr un mejor entorno.

Sin embargo, podríamos relacionar de manera más o menos directa nuestro trabajo como herramienta para cumplir con los siguientes Objetivos de Desarrollo Sostenible:

- Trabajo decente y crecimiento económico. Una correcta securización de la red de una determinada organización supone también una mejor securización de la información transmitida mediante esta, permitiendo un flujo de información eficiente y seguro, lo cual es fundamental para asegurar el crecimiento económico de la organización en cuestion. Asimismo, la profesionalización en el uso de este *framework* podría suponer un factor de valor añadido que permita generar nuevas oportunidades de empleo.
- Industria, innovación e infraestructura. La integración de este *framework* podría considerarse como una innovación en materia de reconocimiento perimetral y generación de reportes, ya que simplificaría el uso de herramientas ya existentes, enfocado en la realización de tareas más especificas directamente relacionadas con las necesidades de un equipo de Red Team en las primeras fases de su proceso de auditoría, contribuyendo de tal manera a la creación de infraestructuras más seguras.
- Producción y consumo responsables. Nuestro *framework* es capaz de ser replicado para su uso tantas veces como sea necesario con un coste de producción nulo, siendo posible su ejecución posible en entornos virtualizados, minimizando la necesidad de recursos para su empleo. Además, gracias a su funcionalidad de analisis de red, es posible detectar dispositivos en desuso que permanecen conectados a la red, asi como servicios que puedan estar consumiendo un exceso de recursos en la red, de manera que se puedan identificar problemas cuya solución contribuye a un consumo responsable de los recursos de la organización.
- Paz, justicia e instituciones sólidas. Aunque en menor medida, nuestro trabajo también podría verse relacionado con este ODS, dado que entre sus aplicaciones se encuentra la prevención y detección activa de ciberataques para obtener el mayor grado posible de seguridad en la información con la que se trabaja, lo cual es esencial para mantener la justicia y la paz relativa a los datos que se encuentran en el ciberespacio.

- Alianzas para lograr objetivos. De manera muy indirecta, este *framework* podría tener implicaciones positivas en el cumplimiento de este ODS ya que podría obtenerse, por ejemplo de un escaneo externo en base a activos descubiertos mediante Shodan, resultados de activos que perteneciesen a otras organizaciones o incluso vulnerabilidades que afecten a dispositivos de terceros que se encuentren en la propia red interna de la organización, lo cual podría dar pie a iniciar relaciones con dicha audiencia, reportando los resultados y trabajando en colaboración para lograr un entorno más seguro.

En conclusión, pese a que este trabajo sea un estudio técnico específico en materia de ciberseguridad, sus diferentes aplicaciones en función del contexto en el que se emplee, pueden contribuir potencialmente con el cumplimiento de varios de los Objetivos de Desarrollo Sostenible definidos por la organización de las Naciones Unidas.