RECONOCIMIENTO

I. Herramientas: En el contexto de la ciberseguridad, la fase de reconocimiento es crucial para recopilar información sobre un objetivo antes de un posible ataque. Algunas herramientas comunes utilizadas en esta fase incluyen:

- **SHODAN:** Motor de búsqueda que descubre dispositivos conectados a internet, como servidores, enrutadores y cámaras web.

    a. **shodan search <query>:** Realiza búsquedas específicas en la base de datos de Shodan.
    b. **shodan host <target>:** Muestra información detallada sobre un host específico.
    c. **shodan stats:** Proporciona estadísticas sobre la base de datos de Shodan.

- **RECON-NG:** Es un marco de trabajo de código abierto que automatiza tareas de reconocimiento en pruebas de penetración y ciberseguridad.

    a. **show modules:** Muestra todos los módulos disponibles para la recopilación de información.
    b. **use <module_name>:** Selecciona un módulo específico para utilizarlo.
    c. **run:** Ejecuta el módulo seleccionado.

- **MALTEGO:** Herramienta de inteligencia de activos que visualiza y relaciona información sobre activos en línea para ayudar en la recopilación, cotejo y análisis de datos en investigaciones de seguridad.
    a. Dentro de la interfaz de Maltego, puedes arrastrar y soltar entidades (como direcciones IP, nombres de dominio, direcciones de correo electrónico, etc.) y luego ejecutar transformaciones para analizar la relación entre estas entidades.

- **NMAP (escaneo de redes y la detección de hosts y vulnerabilidades en servicios)**

    Ayuda
    # nmap --help

    Escaneo de puertos más usados comúnmente
    # nmap -F direcciónIP

    Escanear puerto X específico

nmap -p X direcciónIP

Escanear puerto X UDP
nmap -sU X direcciónIP

Escanear varios puertos X
nmap -p X,X,X,X direcciónIP

Versiones de los servicios
nmap -sVC T1 direcciónIP

Averiguar Sistema Operativo
nmap -O T1 direcciónIP

Realizar análisis con los scripts por defecto
nmap -sC T1 direcciónIP

Script de vulnerabilidades
--script <filename>|<category>|<directory>|<expression>[,...]
nmap --script vulners_enterprise direcciónIP

Análisis con más información(verbose)
nmap -v ejemplo.com

Comprobar si el objetivo tiene activado algún filtrado de paquetes o Firewall
nmap -sA direcciónIP

Equipos activos en la red
nmap -sP direcciónIP

CRIPTOGRAFIA

II. Herramientas: En el contexto de la ciberseguridad, la fase de criptografía es crucial para. Algunas herramientas comunes utilizadas en esta fase incluyen:

- **CrackStation** es un servicio en línea que ofrece una base de datos de contraseñas filtradas y comprometidas

- **CyberChef** - Ofrece una amplia gama de capacidades de manipulación de datos para descifrar y codificar mensajes. Su interfaz intuitiva y fácil de usar permite a los usuarios explorar y aplicar una variedad de operaciones criptográficas y de codificación de manera rápida y eficiente. (página web).

- **Pkcracj** - Herramienta utilizada para realizar ataques de fuerza bruta y descifrar contraseñas de archivos PKZIP mediante el método de ataque conocido como "Known Plaintext Attack".

  Ejemplo de uso: `pkcrack -C <archivo_cifrado> -P <archivo_plaintext> -c <compresion_utilizada> -a <algoritmo_utilizado> -d <diccionario>`

  Donde:

  -C especifica el archivo cifrado que quieres atacar.

  -P es el archivo de texto plano que corresponde al archivo cifrado.

  -c indica el tipo de compresión utilizado por el archivo cifrado.

  -a especifica el algoritmo de cifrado utilizado.

  -d apunta al diccionario a utilizar para el ataque de fuerza bruta.

- **Xortools** - Conjunto de herramientas para realizar operaciones de cifrado y descifrado XOR en archivos y cadenas de datos.

- **OpenSSL** - Biblioteca de código abierto que implementa protocolos de seguridad como SSL/TLS y provee funciones criptográficas para asegurar comunicaciones seguras en redes.

- **Fcrackzip** - Utilidad diseñada para romper contraseñas de archivos ZIP mediante ataques de fuerza bruta.

  Ejemplo de uso: `fcrackzip -u -D -p /ruta/diccionario.txt archivo.zip`

- **RSACTFTool** - Herramienta especializada en la realización de operaciones criptográficas relacionadas con el algoritmo RSA, como cifrado, descifrado, generación de claves y firmas digitales.

- **Multisolver** - Plataforma en línea que proporciona una variedad de herramientas para el cifrado, descifrado y análisis de mensajes y desafíos criptográficos. Esta plataforma está diseñada específicamente para resolver rompecabezas de geocaching y desafíos similares que requieren habilidades en criptografía y resolución de acertijos. (página web).

- **Hashcat** - Herramienta de cracking de contraseñas que admite una amplia variedad de algoritmos de hash (herramienta de línea de comandos).

  Ejemplo de uso: `hashcat -m 0 -a 0 hash.txt rockyou.txt` donde `hash.txt` es el archivo que contiene los hashes a crackear y `rockyou.txt` es el archivo que contiene la lista de contraseñas.

- **John the Ripper** - Herramienta de cracking de contraseñas que admite varios tipos de cifrado y formatos de archivo (herramienta de línea de comandos).

  Ejemplo de uso: `john hash.txt` donde `hash.txt` es el archivo que contiene los hashes a crackear.

- **CryptoCorner** - Herramientas criptográficas y de codificación para ayudar a cifrar y descifrar datos. Algunas de las herramientas que ofrece son generadores de claves criptográficas, cifradores y descifradores de mensajes, y herramientas para el análisis de criptogramas. (página web)

- **AsecuritySite** - Sitio web que trata sobre seguridad de la información y criptografía. Contiene muchísima información y diferentes herramientas para el análisis y la protección de la información. (página web).

- **Boxentriq** - Plataforma en línea que ofrece diversas herramientas y desafíos para el aprendizaje y la práctica de habilidades en áreas como la criptografía, la esteganografía y la resolución de acertijos. Entre sus herramientas se incluyen generadores de claves criptográficas, cifradores y descifradores de mensajes, y herramientas de análisis de criptogramas. (página web) https://www.boxentriq.com/code-breaking/cipher-identifier

- **Dcode** - Plataforma en línea tipo cyberchef que proporciona una amplia variedad de herramientas criptográficas y de codificación para ayudar a cifrar y descifrar

datos de forma segura. También ofrece herramientas para la resolución de acertijos y problemas matemáticos, así como para el análisis de criptogramas y códigos. Muy útil para cifrados RSA (página web).

- **Hashes.org (pagina caída)** es un sitio web que se dedica a la base de datos de hashes y contraseñas 1 . Este tipo de base de datos es utilizada para almacenar hashes de contraseñas obtenidas a partir de filtraciones de datos y otros tipos de incidentes de seguridad

WEB

- **Recolección de información:** recopilar toda la información relevante sobre la aplicación web, como la arquitectura, las tecnologías utilizadas, las URL disponibles, etc. Estas herramientas puede ser:
    - **Nmap:** escáner de puertos.
    - **Nessus:** escáner de vulnerabilidades de red.
    - **BurpSuite:** herramienta utilizada principalmente para pruebas de penetración en aplicaciones web.
- **Análisis de la superficie de ataque:** examinar la aplicación web para identificar todas las áreas que podrían ser vulnerables a ataques incluyendo formularios de entrada, puntos de acceso a la base de datos, APIs, archivos subidos, etc.
    - **Escáneres de vulnerabilidades**: Estas herramientas examinan sistemas informáticos en busca de posibles puntos débiles o vulnerabilidades que podrían ser explotados por atacantes. Ejemplos de escáneres de vulnerabilidades incluyen Nessus, OpenVAS y Qualys.
    - **Herramientas de enumeración**: Estas herramientas se utilizan para recopilar información sobre sistemas, redes o aplicaciones que podrían ser utilizadas en ataques posteriores. Ejemplos incluyen Nmap, SNMPWalk y DNSenum.
    - **Herramientas de análisis de tráfico**: Estas herramientas se utilizan para monitorear y analizar el tráfico de red con el fin de identificar posibles patrones maliciosos o actividades sospechosas. Wireshark es un ejemplo popular de una herramienta de análisis de tráfico.
- **Utilizar herramientas de escaneo automatizado:** usar herramientas que pueden ayudar a escanear la web en busca de vulnerabilidades comunes, como inyecciones

SQL, XSS, CSRF, etc. Algunas de estas herramientas populares incluyen Burp Suite, OWASP ZAP, Nikto, etc.

  - **BurpSuite**: suite de herramientas de prueba de seguridad diseñada específicamente para probar la seguridad de aplicaciones web. Características: escaneo de proxy web, escáner de vulnerabilidades, SpiderWeb (rastrear y mapear la estructura de una aplicación web), etc. https://portswigger.net/burp
  - **SQLMap:** herramienta de código abierto diseñada para automatizar la detección y explotación de vulnerabilidades de inyección SQL en aplicaciones web. https://github.com/sqlmapproject/sqlmap
  - **Nmap:** herramienta de escaneo de puertos y seguridad de código abierto, así como una herramienta de exploración de redes. https://nmap.org/
  - **Aircrack-ng:** herramientas de línea de comandos que verifican y evalúan la seguridad de la red Wi-Fi donde se dedica a actividades como ataque, monitoreo, prueba y craqueo descifrar contraseñas inalámbricas WEP/WAP/ WPA2. https://www.aircrack-ng.org/

- **Realizar pruebas manuales:** realizar pruebas manuales para detectar vulnerabilidades que las herramientas automatizadas podrían pasar por alto. Esto implica probar diferentes áreas de la aplicación con diversos vectores de ataque para ver si se pueden encontrar vulnerabilidades.
  - **Burp Suite**: Esta es una suite de herramientas de prueba de penetración diseñada específicamente para pruebas de seguridad web. Permite realizar pruebas de inyección SQL, cross-site scripting (XSS), entre otras técnicas de ataque más comunes.
  - **Metasploit Framework**: Es una herramienta de código abierto utilizada para el desarrollo y ejecución de exploits contra sistemas informáticos. Facilita la automatización de tareas comunes en pruebas de penetración y permite probar la seguridad de sistemas y redes.
  - **Nmap**: Aunque Nmap es una herramienta de escaneo de redes automatizada, también se puede utilizar de manera manual para realizar exploraciones detalladas de sistemas y redes, identificando puertos abiertos, servicios en ejecución y otras características importantes para la seguridad.

ANÁLISIS FORENSE

- **Autopsy**: herramienta forense digital de código abierto que analiza discos duros y dispositivos de almacenamiento para recuperar evidencia digital, como archivos eliminados o datos ocultos, facilitando investigaciones criminales y de seguridad..

- **Volatility**: herramienta de análisis forense digital especializada en la extracción y examen de información volátil de sistemas informáticos, como la memoria RAM, para identificar procesos, conexiones de red, malware y evidencia de actividad sospechosa o delictiva.

- **Wireshark:** herramienta de análisis de tráfico de red que también se puede utilizar con propósitos forenses, permitiendo la captura y examen detallado de paquetes de datos para investigar incidentes de seguridad, identificar patrones de comportamiento malicioso y reconstruir eventos de red.

- **Forensic Toolkit (FTK):** permite adquirir, analizar y presentar evidencia digital de manera integral. Ofrece herramientas para examinar discos duros, dispositivos móviles y otros medios de almacenamiento, facilitando la investigación de delitos cibernéticos y la recuperación de datos relevantes para casos judiciales.

- **Forensically:** colección de herramientas basada en la web que se pueden utilizar para "análisis forense de imágenes digitales". Algunas funcionalidades incluyen funciones de ampliación, detección de clones, análisis de nivel de error, análisis de ruido, barrido de nivel y muchas más.

- **Binwalk:** herramienta rápida y fácil de usar para: analizar en busca de código malicioso, realizar ingeniería inversa y extraer imágenes de firmware. Binwalk hace un buen trabajo analizando posibles firmas de archivos y filtrando falsos positivos obvios, pero no es perfecto.

- **Pdf-Parser:** herramienta construida sobre PDF Miner para ayudar a extraer información de archivos PDF en Python. La idea principal era crear una herramienta que pudiera ser impulsada por código para interactuar con los elementos del PDF y clasificarlos lentamente creando secciones y agregándoles etiquetas.

EXPLOTACIÓN

- **Herramientas / Comandos**

Metasploit: Metasploit es una plataforma de código abierto para probar y ejecutar exploits en sistemas informáticos, diseñada para evaluar la seguridad y encontrar vulnerabilidades.

msfconsole (iniciar la consola de metasploit)
search (buscar el módulo que se necesite)
use (usar dicho módulo)
options (mostrar las opciones del módulo)
set (establecer dichas opciones)
run / exploit (empezar el exploit)

Msfvenom: Es una herramienta incluida en el marco de trabajo de Metasploit que se utiliza para generar payloads, también conocidos como shellcodes o códigos maliciosos, de manera rápida y sencilla.

-p TIPO_PAYLOAD
–list payloads
–help

Burpsuite: Burp Suite es una suite de herramientas de prueba de seguridad, se utiliza principalmente para realizar pruebas de penetración y evaluaciones de seguridad en aplicaciones web.

Nessus: Una herramienta de escaneo de vulnerabilidades que puede identificar y clasificar las vulnerabilidades en sistemas remotos. Nessus utiliza un conjunto de plugins para realizar escaneos exhaustivos en busca de vulnerabilidades conocidas.

OpenVAS: Un escáner de vulnerabilidades de código abierto que realiza evaluaciones de seguridad automatizadas en sistemas y redes. OpenVAS puede identificar vulnerabilidades conocidas y configuraciones erróneas que podrían ser explotables.

Nmap: Es una herramienta de escaneo de red ampliamente utilizada para descubrir hosts y servicios en una red, así como para identificar puertos abiertos y realizar evaluaciones de seguridad.

-sV (Muestra que servicio está ejecutándose y su versión)
-O (Muestra el sistema operativo de la máquina objetivo)
–script SCRIPT (permite ejecutar diferentes scripts para el análisis)
-sC (Igual que el anterior pero ejecuta los scripts por defecto)

Searchsploit: Es una herramienta incluida en el marco Metasploit que se utiliza para buscar exploits y shellcodes en la base de datos de exploits de Exploit Database (EDB).

-m (Sirve para descargar el exploit**) código exploit**

uname -a: Se utiliza en sistemas Linux para obtener información detallada sobre el sistema operativo y el hardware en el que se está ejecutando. Como el kernel y la arquitectura.

lsb_release -a: Se utiliza en sistemas Linux para obtener información detallada sobre la distribución y la versión del sistema operativo.

- **Páginas web**

Exploit-db: **https://www.exploit-db.com/** (Exploit-DB es una base de datos de exploits y vulnerabilidades conocidas que proporciona acceso a una amplia colección de exploits, shellcodes y documentos relacionados con la seguridad informática.)

Gtfobins: **https://gtfobins.github.io/#** (Es un recurso en línea que enumera binarios y comandos comunes en sistemas Unix y Linux que pueden ser utilizados de manera maliciosa para obtener una shell de sistema o escalar privilegios en un entorno comprometido. GTFOBins proporciona ejemplos de uso para cada comando, lo que facilita su explotación en escenarios de pentesting y hacking ético.)

Attackerb.com: **https://attackerkb.com/**

Shodan: **https://www.shodan.io/** (Es un motor de búsqueda especializado que se centra en la identificación y el análisis de dispositivos conectados a Internet.)

- **Scripts**

linpeas: **https://github.com/carlospolop/PEASS-ng/tree/master/linPEAS** (Es un script de enumeración de seguridad para sistemas Linux que ayuda a identificar de manera rápida y automática la configuración del sistema, los servicios en ejecución, los permisos de archivos…El objetivo principal de LinEnum es ayudar a los administradores de sistemas y a los evaluadores de seguridad a identificar posibles puntos débiles y configuraciones incorrectas en sistemas Linux.)

winpeas: https://github.com/carlospolop/PEASS-ng/tree/master/winPEAS (Es una herramienta de enumeración y escalada de privilegios para sistemas Windows. Similar a LinPeas para sistemas Linux.)

INGENIERÍA INVERSA

1. Análisis Inicial

- Se inicia identificando y extrayendo los archivos binarios relevantes del software.
- `Hopper Disassembler`: Facilita el análisis y la comprensión de programas mediante la descompilación de código en un formato legible.
- `Binary Ninja`: Proporciona un entorno de análisis estático personalizable para comprender la estructura y el flujo del código binario.

2. Análisis Dinámico

- El siguiente paso consiste en ejecutar el programa en un entorno controlado, haciendo uso de herramientas como `OllyDbg` y `GDB`.
- Se interceptan llamadas de sistema y eventos importantes para comprender el comportamiento del software en tiempo de ejecución.
- `WinDbg`: Herramienta avanzada de depuración para Windows que permite el análisis a bajo nivel de sistemas y aplicaciones.
- `x64dbg`: Depurador de código abierto que proporciona capacidades avanzadas para el análisis dinámico de programas.

3. Análisis de Fuzzing y Entrada Aleatoria

- Se somete el software a pruebas de fuzzing con el objetivo de identificar posibles vulnerabilidades.
- `AFL (American Fuzzy Lop)`: Realiza fuzzing automático, generando entradas aleatorias de forma inteligente para identificar vulnerabilidades de software.
- `Peach Fuzzer`: Una plataforma de pruebas de vulnerabilidad que permite la generación y el análisis de casos de prueba para encontrar fallos de seguridad.

4. Identificación de Algoritmos y Protocolos de Comunicación

- Se busca identificar y comprender los algoritmos criptográficos utilizados en caso de comunicaciones seguras.
- `Wireshark`: Herramienta de análisis de redes que facilita la comprensión de protocolos de comunicación y la inspección del tráfico de red.
- `Crypto++`: Biblioteca de algoritmos criptográficos de código abierto que proporciona una amplia gama de algoritmos.

5. Análisis de Protecciones y Ofuscación

- Se analiza cualquier medida de protección contra ingeniería inversa presente en el binario. Se emplean técnicas de desofuscación para simplificar la comprensión del código.
- `Radare2`: Proporciona una amplia gama de herramientas para analizar protecciones contra ingeniería inversa.
- `Unicorn Engine`: Emulador de código multiarquitectura que ayuda en el análisis de código ofuscado y protegido.

6. Búsqueda de Vulnerabilidades de Seguridad

- `Frida`: Herramienta para hacer inyección de código, monitorear y manipular aplicaciones en tiempo de ejecución, útil para identificar vulnerabilidades de seguridad.
- `Angr`: Plataforma de análisis de binarios que ayuda en la búsqueda y explotación de vulnerabilidades de seguridad.

7. Análisis Estático del Binario

- `Ghidra`: Se utiliza para llevar a cabo un análisis estático detallado del software, permitiendo la exploración del código ensamblador, la identificación de funciones y la comprensión de la estructura del programa. Ghidra también ofrece capacidades avanzadas para el análisis de malware y la identificación de vulnerabilidades en binarios de Windows y Linux.

PROGRAMACIÓN

Los desafíos de programación en CTFs bajo la categoría PPC son fundamentales en muchos CTFs. Requieren que los participantes desarrollen un programa o script para una tarea específica. Puede implicar desde la manipulación de datos hasta la resolución de problemas algorítmicos complejos. La programación de scripts en Bash y en Batch son habilidades valiosas. Aquí hay ejemplos de cómo pueden aparecer estos desafíos y tres herramientas comunes utilizadas para abordarlos.

Ejemplos de desafíos de PPC:

1. **Manipulación de cadenas o datos:** Se proporciona un conjunto de datos en bruto o una cadena de texto, y se pide a los participantes que realicen ciertas operaciones (como filtrado, manipulación, extracción, etc.) para obtener información específica o lograr un resultado deseado.
2. **Resolución de problemas algorítmicos:** Se plantea un problema algorítmico (por ejemplo, encontrar el camino más corto en un grafo, implementar un algoritmo de ordenación eficiente, resolver un rompecabezas matemático, etc.) y se solicita a los participantes que escriban un programa que resuelva ese problema de manera óptima.

3. **Ingeniería de software:** Los participantes pueden recibir un fragmento de código incompleto o mal escrito y se les desafía a corregir los errores, completar la implementación o mejorar la eficiencia del código.

Herramientas comunes para abordar desafíos de PPC:

1. **Entornos de desarrollo integrado (IDE):** Herramientas como Visual Studio Code, PyCharm, IntelliJ IDEA o Eclipse proporcionan un entorno completo para escribir, depurar y ejecutar código. Los participantes pueden aprovechar las funciones de autocompletado de código, resaltado de sintaxis y depuración para desarrollar programas de manera efectiva y eficiente.
2. **Lenguajes de programación versátiles:** Lenguajes como Python, C++, Java, JavaScript y Ruby son comunes en los desafíos de PPC. Los participantes deben elegir el lenguaje más adecuado para el desafío y estar familiarizados con sus características y bibliotecas estándar para completar la tarea de manera eficiente.
3. **Bibliotecas y herramientas especializadas:** Dependiendo del tipo de desafío, los participantes pueden necesitar utilizar bibliotecas o herramientas especializadas. Por ejemplo, para la manipulación de datos en Python, podrían utilizar bibliotecas como Pandas o NumPy. Para resolver problemas algorítmicos, podrían recurrir a bibliotecas de matemáticas como SciPy o implementaciones de estructuras de datos avanzadas.

Para analizar vulnerabilidades de un programa o una parte de código, podemos hacer uso de la aplicación web Vulert Scanner. Es un escáner en línea diseñado para detectar dependencias de código abierto vulnerables en tus proyectos. Opera sin necesidad de instalación y abarca una variedad de lenguajes de programación, incluyendo PHP, JavaScript, Java, Python y C/C++. Además, Vulert Playground es capaz de escanear los SBOMs.

Otra herramienta muy útil y que es capaz de analizar multitud de lenguajes de programación es Snyk Code Checker. Ofrece 3 escaneos sin necesidad de crearse una cuenta en el sitio.

Otra herramienta es BlackBox, la cual responde cualquier pregunta acerca de cualquier lenguaje de programación que solicitemos.

Otra herramienta es Dotenv. Esta es una extensión de Visual Studio Code la cual permite encontrar fallos en el código.