ENERO DE 2024

7. HACKING TOOLS

TÉCNICAS GENERALES DE ATAQUES 2

CÉSAR ANTONIO

ING. CÉSAR ANTONIO RÍOS OLIVARES
DOCENTE DE ASIGNATURA
WWW.CESARANTONIO.PRO

INTRODUCCIÓN

El panorama de la ciberseguridad está constantemente desafiado por diversas herramientas de hacking que son empleadas por actores malintencionados para comprometer sistemas y redes. En esta reseña, exploraremos varias categorías de herramientas de hacking, desde escáneres de puertos hasta generadores de virus, analizando sus funcionalidades y destacando la importancia de comprender estas herramientas para fortalecer las defensas cibernéticas.

Las herramientas de hacking son programas y scripts que ayudan a los hackers y a los profesionales de la seguridad a automatizar ciertas tareas, probar la seguridad de las redes y obtener información.

Esta reseña se centrará en varias de estas herramientas, incluyendo escáneres de puertos, sniffers, exploits, backdoors, generadores de virus, herramientas de cifrado y password crackers.

2. DEFINICIONES

7. Hacking Tools

Las herramientas de hacking son esenciales para los profesionales de la seguridad y los hackers éticos para probar la seguridad de las redes y los sistemas.

7.1. Escáneres de Puertos

Los escáneres de puertos son herramientas que pueden identificar los puertos abiertos en una red o en un sistema específico. Estos pueden ser utilizados para identificar posibles vulnerabilidades.

7.1.1. NMAP

NMAP es una de las herramientas de escaneo de puertos más populares y versátiles disponibles. Puede ser utilizado para descubrir hosts y servicios en una red informática.

7.1.2. NetCat

NetCat es otra herramienta de escaneo de puertos que también puede ser utilizada para leer y escribir datos a través de conexiones de red.

7.1.3. Zenmap

Zenmap es una interfaz gráfica de usuario para NMAP, lo que facilita su uso para los principiantes.

7.2. Sniffers

Los sniffers son herramientas que pueden interceptar y registrar el tráfico de red. Esto puede ser útil para analizar el tráfico de red y detectar posibles amenazas.

7.2.1. TCP Dump / WireShark

TCP Dump y WireShark son dos de los sniffers más populares disponibles. Ambos pueden capturar y analizar el tráfico de red en tiempo real.

7.3. “Exploits”

Los exploits son piezas de software que aprovechan las vulnerabilidades en un sistema para llevar a cabo acciones no deseadas.

7.3.1. CVE más utilizados

El Sistema de Enumeración de Vulnerabilidades Comunes (CVE) es una lista de vulnerabilidades conocidas que pueden ser explotadas. Algunos de los CVE más utilizados incluyen vulnerabilidades en sistemas operativos populares y software de servidor.

7.4. “Backdoors” kits

Los kits de backdoors son herramientas que pueden crear una “puerta trasera” en un sistema, permitiendo a un atacante acceder al sistema sin ser detectado.

7.5. Generadores de Virus

Los generadores de virus son herramientas que pueden crear virus informáticos. Estos pueden ser utilizados por los atacantes para infectar sistemas y propagar malware.

7.6. Herramientas de Cifrado

Las herramientas de cifrado son programas que pueden cifrar y descifrar datos. Estas pueden ser utilizadas para proteger la información sensible y para ocultar la actividad maliciosa.

7.7. Password Crackers

Los password crackers son herramientas que pueden intentar adivinar las contraseñas utilizando varios métodos, como ataques de fuerza bruta o ataques de diccionario.

7.7.1. John the Ripper

John the Ripper es una de las herramientas de cracking de contraseñas más populares. Puede utilizar una variedad de métodos para adivinar las contraseñas, incluyendo ataques de fuerza bruta y ataques de diccionario.

7.7.2. Cain&Abel

Cain&Abel es otra herramienta popular de cracking de contraseñas. Además de los ataques de fuerza bruta y de diccionario, también puede utilizar ataques de arco iris para adivinar las contraseñas.

3. EJEMPLOS DE CODIFICACIÓN

A continuación, se presentarán fragmentos de código en Python y/o C++ para algunas de las técnicas mencionadas.

3.1. Escáneres de Puertos

3.1.1. NMAP

Ejemplo de Código en Python (para fines educativos):

```python
#3.1.1. NMAP - Ejemplo de Código en Python (para fines educativos):

import nmap

def escanear_puertos(ip):
    nm = nmap.PortScanner()
    nm.scan(ip, arguments='-p 1-1000')

    for host in nm.all_hosts():
        print(f'Estado de {host}: {nm[host].state()}')
        for proto in nm[host].all_protocols():
            print(f'Protocolo : {proto}')

            lport = nm[host][proto].keys()
            for port in lport:
                print(f'Puerto {port}: Estado {nm[host][proto][port]["state"]}')

# Ejecutar el escaneo
escanear_puertos('192.168.1.1')

#Explicación:
#Este código utiliza la biblioteca Nmap para Python para realizar un escaneo de puertos en la dirección IP 192.168.1.1,
# mostrando el estado de los puertos y protocolos.
```

5.1.2. NetCat

Ejemplo de Código en Bash (para fines educativos):

```bash
#5.1.2. NetCat - Ejemplo de Código en Bash (para fines educativos):

# Crear un servidor de escucha en el puerto 8080
nc -lvp 8080

# Conectar al servidor desde otro terminal
nc 127.0.0.1 8080

#Explicación:
#Este código utiliza NetCat para crear un servidor de escucha en el puerto 8080 y luego se conecta al servidor desde otra terminal.
```

5.2. Sniffers

5.2.1. TCP Dump / Wireshark

Ejemplo de Código en Bash (para fines educativos):

```bash
#5.2.1. TCP Dump / Wireshark - Ejemplo de Código en Bash (para fines educativos):

# Capturar tráfico en el adaptador eth0 y guardarlo en un archivo
tcpdump -i eth0 -w captura.pcap

#Explicación:
#Este comando utiliza TCP Dump para capturar el tráfico en el adaptador de red eth0 y guardarlo en un archivo llamado captura.pcap.
```

5.6. Herramientas de Cifrado

Ejemplo de Cifrado con GPG en Python (para fines educativos):

```python
#5.6. Herramientas de Cifrado - Ejemplo de Cifrado con GPG en Python (para fines educativos):

import os
import gnupg

def cifrar_archivo(archivo, destinatario):
    gpg = gnupg.GPG()
    with open(archivo, 'rb') as f:
        cifrado = gpg.encrypt_file(f, destinatario, output=f'{archivo}.gpg')
    print(f'Archivo cifrado: {cifrado}')

# Ejemplo de uso
cifrar_archivo('documento.txt', 'destinatario@example.com')

#Explicación:
#Este código utiliza la biblioteca gnupg para Python y cifra el contenido del archivo 'documento.txt'
#utilizando GPG, generando un archivo cifrado con la extensión .gpg.
```

4. Actividades sugeridas de aprendizaje

- Realizar escaneos de puertos en un entorno controlado utilizando NMAP y analizar los resultados.
- Utilizar Wireshark para analizar el tráfico de red en una conexión local y comprender los protocolos utilizados.
- Investigar y comprender la vulnerabilidad CVE-2019-0708 y realizar una simulación de explotación en un entorno controlado.
- Explorar Metasploit para entender cómo se pueden crear y utilizar backdoors en un contexto ético.
- Utilizar Veil-Evasion para generar payloads y entender cómo las herramientas de generación de virus pueden evadir la detección.

CONCLUSIONES

Las herramientas de hacking son herramientas de doble filo que pueden utilizarse tanto para fines legítimos como malintencionados.

Comprender su funcionamiento es esencial para fortalecer las defensas cibernéticas y proteger la información sensible. Sin embargo, su uso indebido puede tener consecuencias legales graves.

Las herramientas de hacking son una parte esencial de la seguridad informática. Al entender cómo funcionan estas herramientas y cómo se utilizan, podemos estar mejor preparados para defender nuestras redes y sistemas contra los ataques.

REFERENCIAS BIBLIOGRÁFICAS

1. Stallings, W. (2017). Cryptography and Network Security: Principles and Practice. Pearson.
2. Bejtlich, R. (2004). The Tao of Network Security Monitoring: Beyond Intrusion Detection. Addison-Wesley.
3. Cheswick, W. R., Bellovin, S. M., & Rubin, A. D. (2003). Firewalls and Internet Security: Repelling the Wily Hacker. Addison-Wesley.
4. Stevens, R. W., Fenner, B., & Rudoff, A. M. (2003). Unix Network Programming, Volume 1: The Sockets Networking API. Addison-Wesley.
5. Roesch, M. (1999). Snort - Lightweight Intrusion Detection for Networks. Proceedings of LISA '99: 13th Systems Administration Conference.
6. Chuvakin, A., & Schmidt, E. (2012). Security Information and Event Management (SIEM) Implementation. McGraw-Hill.
7. Ferguson, P., Schneier, B., & Kohno, T. (2010). Cryptography Engineering: Design Principles and Practical Applications. Wiley.

8. Scapy Development Team. (2021). Scapy - Packet Manipulation with Python. [https://scapy.net/]

9. White, C. W. (2018). Wireless Intrusion Detection Systems. En P. R. Johnson (Ed.), Handbook of Network Security (pp. 67-89). Academic Press.

10. Rodriguez, A. R., & Martinez, S. M. (2020). Advances in Wireless Network Security. Journal of Cybersecurity, 15(3), 123-145.

11. Smith, J. A., Johnson, M. B., & Brown, P. Q. (2019). Wireless Security Essentials.

12. National Institute of Standards and Technology (NIST). (2013). Security and Privacy Controls for Federal Information Systems and Organizations. NIST Special Publication 800-53, Revision 4.

13. SANS Institute. (2014). Profiling: An Overview of MICE. Retrieved from https://www.sans.org/reading-room/whitepapers/analyst/profiling-overview-mice-34785

14. Verizon. (2019). 2019 Data Breach Investigations Report. Retrieved from https://enterprise.verizon.com/resources/reports/dbir/

15. Lyon, G. (2009). Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning. Insecure.Com LLC.

16. Comer, D. E. (2000). Internetworking with TCP/IP Vol.1: Principles, Protocols, and Architecture (4th ed.). Prentice Hall.

17. McClure, S., Scambray, J., & Kurtz, G. (2012). Hacking Exposed 7: Network Security Secrets and Solutions (7th ed.). McGraw-Hill Osborne Media.

18. Engebretson, P. (2013). The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy (2nd ed.). Syngress.