Seguridad en los Sistemas Informáticos
Tema 3: Hacking ético

Grado en Ingeniería Informática

Departamento de Ingeniería Informática
Universidad de Cádiz

Curso 2024–2025

Índice

Hacking ético

Definición

- Se conoce hacking ético al conjunto de pruebas de intrusión (*penetration testing*) llevadas a cabo por profesionales de seguridad con una metodología rigurosa.
- Un hacker ético analiza un sistema o una red para identificar y explotar sus debilidades.
- El hacker ético para su actividad cuando sabe que sus acciones pueden repecutir negativamente en los datos o el sistema auditado.

Tipos de hackers (I)

White hat, *sneaker* o hacker ético

- El hacker ético o hacker de sombrero blanco es un experto en seguridad informática.
- Está especializado en realizar pruebas de intrusión para garantizar la seguridad de la información de los sistemas de una organización.
- Suele comunicar a las empresas las brechas de seguridad encontradas.

Tipos de hackers (II)

Black hat o *cracker*

- El cracker o hacker de sombrero negro es un experto en atacar sistemas.
- Los ataques se llevan a cabo con un propósito malicioso o beneficio personal.
- Su principal objetivo es causar daño al sistema: borrado de ficheros, robo de información, modificación de sitios web, etc.

Tipos de hackers (III)

Grey hat

- El hacker de sombrero gris es un hacker que a veces respeta las leyes y otras veces actúa ilegalmente.
- Es una mezcla de hacker ético y cracker.
- Suele alertar de las vulnerabilidades encontradas a la comunidad hacker.
- Entre sus motivaciones se encuentran el ánimo de lucro, protestas o desafíos personales.

Tipos de hackers (IV)

Insider

- Trabaja como empleado o consultor en una organización.
- Tiene acceso a los sistemas de la organización.
- Puede causar grandes daños, repercutiendo en pérdidas de grandes sumas de dinero.
- Entre sus motivaciones se encuentran la venganza (tras haber sido despedido), la obtención de información estratégica, confidencial o con ánimo de lucro.

Tipos de hackers (V)

Script kiddie

- Un atacante inexperto capaz de ejecutar un *script*/aplicación desarrollado por un tercero para llevar a cabo un ataque.
- Carece de un buen nivel de programación o de las aplicaciones usadas.
- Dada su inexperiencia, deja muchas huellas durante los ataques, siendo fácilmente rastreable.

Phreaker

- Un atacante que consigue acceder a sistemas o centrales telefónicos.
- Se requiere un gran conocimiento en telecomunicaciones.

Metodologías (I)

Tipos de metodologías

- En un test de intrusión las tareas deben ejecutarse siguiendo un orden que permita ir conociendo cada vez más información sobre el sistema objetivo.
- Las metodologías pueden clasificarse en:

Caja negra	No se dispone de ningún tipo de información, por lo que la tarea más importante es la búsqueda de información.
Caja gris	El atacante parte de alguna información.
Caja blanca	Se dispone de muchos tipos de información, incluso credenciales de acceso.

Metodologías (II)

Metodologías más conocidas a nivel práctico

- **Open Web Application Security Project (OWASP) testing guide.**
- **Open Source Security Testing Methodology Manual (OSSTMM).**
- **Penetration Testing Execution Standard (PTES).**
- **MITRE Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK).**
- PCI Penetration testing guide.
- NIST 800-115.
- Penetration Testing Framework.
- Information Systems Security Assessment Framework (ISSAF).

OWASP (I)

Descripción

- OWASP es una fundación sin ánimo de lucro.
- Supone una referencia para aspectos que afectan a la seguridad de aplicaciones web.
- Publica el *OWASP top ten*, un listado con las 10 vulnerabilidades más importantes y herramientas y consejos para subsanarlos.
- También desarrollan guías sobre seguridad como la Guía OWASP de Pruebas: `https://www.owasp.org/index.php/OWASP_Testing_Project`

OWASP (II)

Metodología OWASP

Esta metodología dirigida a aplicaciones web se divide en 12 apartados:

- Introducción y objetivos.
- Recopilación de la información.
- Pruebas de configuración y gestión de despliegue.
- Pruebas de gestión de identidades.
- Pruebas de autenticación.
- Pruebas de autorización.
- Pruebas de gestión de sesiones.
- Pruebas de validación de datos.
- Pruebas de manejo de errores.

OWASP (III)

Metodología OWASP (cont.)

- Pruebas de métodos de cifrados débiles.
- Pruebas sobre lógica de negocio.
- Pruebas del lado del cliente.

OWASP (IV)

Ejercicio 3.1

- Dividimos la clase en cuatro grandes grupos.
- Cada grupo deberá elegir a un portavoz.
- Cada uno de los cuatro portavoces debe crear y compartir un documento Google Docs con todos los miembros de su grupo, publicando la URL en el foro de teoría.
- Cada grupo deberá buscar y describir brevemente cada una de las siguientes 10 vulnerabilidades.

OWASP (IV)

Ejercicio 3.1 (cont)

- Grupo 1: *OWASP top ten*
  `https://owasp.org/www-project-top-ten/`
- Grupo 2: *OWASP Mobile top ten*:
  `https://owasp.org/www-project-mobile-top-10/`
- Grupo 3: *OWASP IoT*:
  `https://owasp.org/www-project-internet-of-things/`
- Grupo 4: *OWASP Top 10 for Large Language Model Applications*:
  `https://owasp.org/www-project-top-10-for-large-language-model-applications/`
- Finalmente, cada grupo expondrá a toda la clase la información recabada.

PTES

Descripción

- El estándar PTES (`http://www.pentest-standard.org/index.php/Main_Page`) propone 7 secciones principales:
  - Interacciones previas al compromiso.
  - Recopilación de inteligencia.
  - Modelado de amenazas.
  - Análisis de vulnerabilidades.
  - Explotación.
  - Post-explotación.
  - Informes
- Existe una guía técnica para llevar a cabo la ejecución de las pruebas: `http://www.pentest-standard.org/index.php/PTES_Technical_Guidelines`

OSSTMM (I)

Descripción

- La metodología incluye en su alcance a personas, sistemas, aplicaciones y procesos:
  `https://www.isecom.org/research.html`
- Se estructura en 15 capítulos sobre conceptos básicos de seguridad, procesos para definir una prueba, aspectos legales, métricas, entre otros.
- Propone distintos tipos de pruebas:
  - Pruebas de seguridad de los factores relacionados con las personas.
  - Pruebas de seguridad física.
  - Pruebas de seguridad inalámbrica.
  - Pruebas de seguridad en telecomunicaciones.
  - Pruebas de seguridad en redes de datos.
  - Pruebas de cumplimiento normativo.

OSSTMM (II)

STAR

- El manual proporciona un modelo de informe denominado *Security Test Audit Report* (STAR).
- Este informe recapitula todas las evidencias encontradas durante las pruebas.
- Los auditores de ISECOM pueden verificar este informe.

MITRE ATT&CK

Descripción

- Es una base de conocimiento de técnicas adversas.
- Permite gestionar:
  - Comportamientos adversos.
  - Modelos de ciclo de vida.
  - Aplicación a entornos reales.
  - Taxonomía común.
- Las tácticas representan el porqué de una técnica ATT&CK (`https://attack.mitre.org/`).
- Las técnicas representan el cómo un adversario consigue un objetivo táctico realizando una acción.
- Las técnicas también pueden representar el qué gana un adversario realizando una acción.
- La relación entre tácticas y técnicas se visualizan en la matriz ATT&CK.

Reconocimiento y recolección de información en fuentes abiertas

Descripción

- La primera fase del *pentesting* es el reconocimiento (*footprinting*).
- Consiste en descubrir toda la información relevante de la organización víctima o cliente.
- Existen dos tipos de reconocimiento:

  Pasivo No tenemos interacción directa con la víctima o el cliente. Por ejemplo, investigación de una empresa a través de Google.

  Activo Existe una interacción directa con la víctima o el objetivo. Por ejemplo, realizar un mapeo de la red o un barrido de ping.

Escaneo y enumeración

Descripción

- La segunda fase del *pentesting* es el escaneo y enumeración (*fingerprinting*).
- Tras obtener el rango de direcciones IP de nuestro objetivo durante la fase de reconocimiento, en esta fase de escaneo:
  - Identificaremos los hosts que estén activos.
  - Buscaremos los puertos abiertos en dichas máquinas.
  - A partir de los puertos abiertos, obtendremos información del sistema operativo, aplicaciones, servicios...
- Una vez identificados el sistema operativo, servicio, etc. podremos decidir si realizar la enumeración (escaneo más intenso) para tratar de obtener cuentas de usuario, procesos...
- Si existen vulnerabilidades, estas podrán ser posteriormente explotadas.

Análisis de vulnerabilidades (I)

Vulnerabilidad

- "*Un fallo o una debilidad en los procedimientos, diseño, implementación o controles internos de seguridad del sistema, que puede ser activado (accidentalmente o de forma intencionada) y resultar en una brecha de seguridad o la violación de la política de seguridad del sistema*" (Guía de Gestión de Riesgos de las Tecnologías de la Información publicada por el NIST).

Análisis de vulnerabilidades (II)

Origen de las vulnerabilidades

- Técnico, debido a fallos en la configuración de sistemas o errores de implementación.
- Factor humano y falta de cobertura de las políticas que las regulan.

Resumen de la anatomía de un ataque

- Reconocimiento.
- Identificación de vulnerabilidades.
- Explotación.

Clasificación y fuentes (I)

CVE

- *Common Vulnerabilities and Exposures* (CVE) es una base de datos de vulnerabilidades públicamente conocidas.
- De libre acceso internacional: `https://cve.mitre.org/`
- Creada por el MITRE y financiada parcialmente por el *Department of Homeland Security* (DHS) de los EE.UU.
- Ofrece un identificar único y público (CVE-ID) para cada vulnerabilidad registrada:
  - Hasta el 01/01/2014, el CVE-ID tenía el siguiente formato fijo: CVE-AAAA-NNNN
  - A partir de esta fecha, el CVE-ID tiene el formato: CVE-AAAA-NNNN...N
- Algunas bien documentadas: CVE-2021-26855, CVE-2021-28316, CVE-2021-44228, CVE-2020-1472, CVE-2017-0144, CVE-2021-34527, CVE-2021-42013.

Clasificación y fuentes (II)

Ejercicio 3.2

Cada grupo debe buscar en CVE diez vulnerabilidades al azar, identificando:

- Resumen de en qué consiste.
- Cómo se podría mitigar o evitar.
- Un programa/sistema que la sufra o la haya sufrido.

Clasificación y fuentes (III)

NVD

- *National Vulnerability Database* (NVD) es una base de datos de vulnerabilidades públicamente conocidas.
- De libre acceso internacional: `https://nvd.nist.gov/`
- Creada por el NIST y mantenida por el gobierno de EE.UU.
- Implementa el protocolo *Security Content Automation Protocol* (SCAP) que facilita la clasificación y gestión automática de vulnerabilidades.
- Permite:
  - Realizar búsquedas de vulnerabilidades específicas.
  - Descargar la base de datos completa.
  - Utilizar gratuitamente herramientas de consulta y alerta. Destaca la fuente RSS: `https://nvd.nist.gov/download/nvd-rss-analyzed.xml`

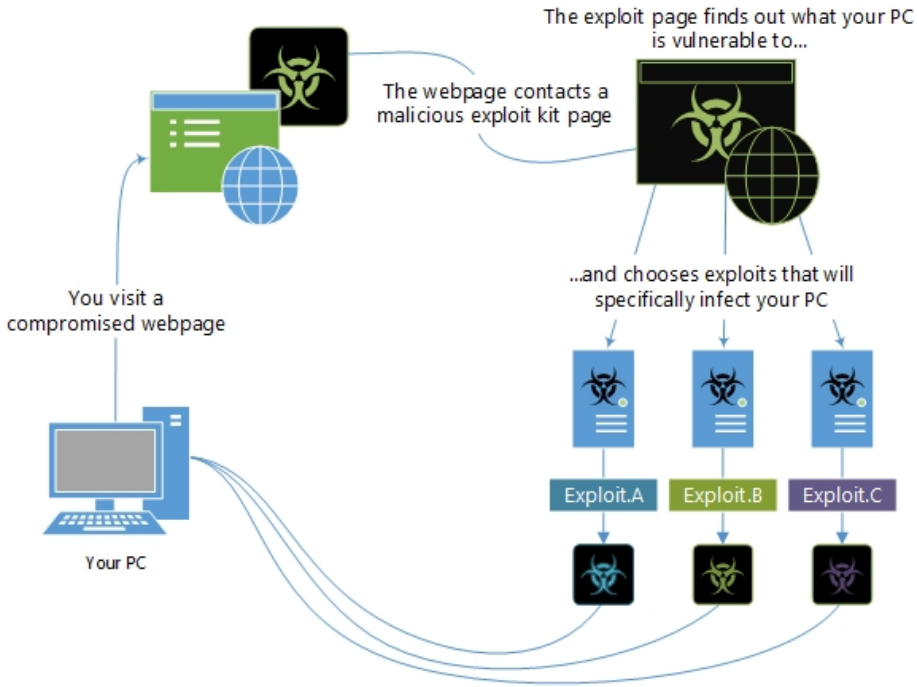
Explotación de vulnerabilidades (I)

Explotación

- Un *exploit* es una pequeña aplicación implementada para aprovecharse de una vulnerabilidad conocida en un software.
- El *pentester* utiliza un *exploit* para tratar de conseguir el control de la máquina remota.
- Los exploits se implementan en distintos lenguajes: C, Ruby, Java, Python...
- Un *payload* o *shellcode* es la parte del código de un *exploit* cuya finalidad es ejecutarse en la máquina víctima para llevar a cabo una acción, normalmente maliciosa.

Explotación de vulnerabilidades (II)

Ejemplo de ataque mediante *exploits* entrando en una web comprometida:



Fuente: `https://docs.microsoft.com/en-us/microsoft-365/security/intelligence/exploits-malware?view=o365-worldwide`

Explotación de vulnerabilidades (III)

Social Engineering Toolkit (SET)

- SET es un framework para automatizar ataques basados en ingeniería social: `https://www.social-engineer.org/framework/general-discussion/`
- Permite realizar ataques como:
  - *Phishing*
  - Vectores de ataques web.
  - Generación de payloads para dispositivos USB.
  - Ataques de *mailing* masivo.

Explotación de vulnerabilidades (IV)

Metasploit

- Metasploit es un *framework* que proporciona un conjunto de herramientas con las que el auditor pueda desarrollar, ejecutar y lanzar *exploits* contra máqinas para comprobar si son seguras.
- Es uno de los principales *frameworks* de Kali Linux.
- En */usr/share/metasploit-framework* se encuentran los binarios de tipo msf, i.e. herramientas como:
  - Línea de comandos para interactuar con Metasploit.
  - Interfaz gráfica para interactuar con Metasploit.
  - Generación de payloads.
  - Análisis de binarios.

Post-explotación (I)

Post-explotación

- Una vez que el hacker ha vulnerado un sistema con éxito, podrá llevar a cabo tareas de post-explotación para continuar con su ataque:
  - Elevar privilegios desde una cuenta sin privilegios para obtener permisos administrativos. Proceso por el que, desde una cuenta de usuario con unos permisos determinados, se obtienen permisos superiores para realizar acciones que previamente estaban restringidas.
  - Obtener los *hashes* de los usuarios de un sistema para posteriormente efectuar una rotura de contraseñas.
  - Migrar un proceso a otro.
  - Descargar o subir información desde y hacia un equipo, que ha sido vulnerado.
  - Eliminar pistas que podrían revelar la explotación del sistema.

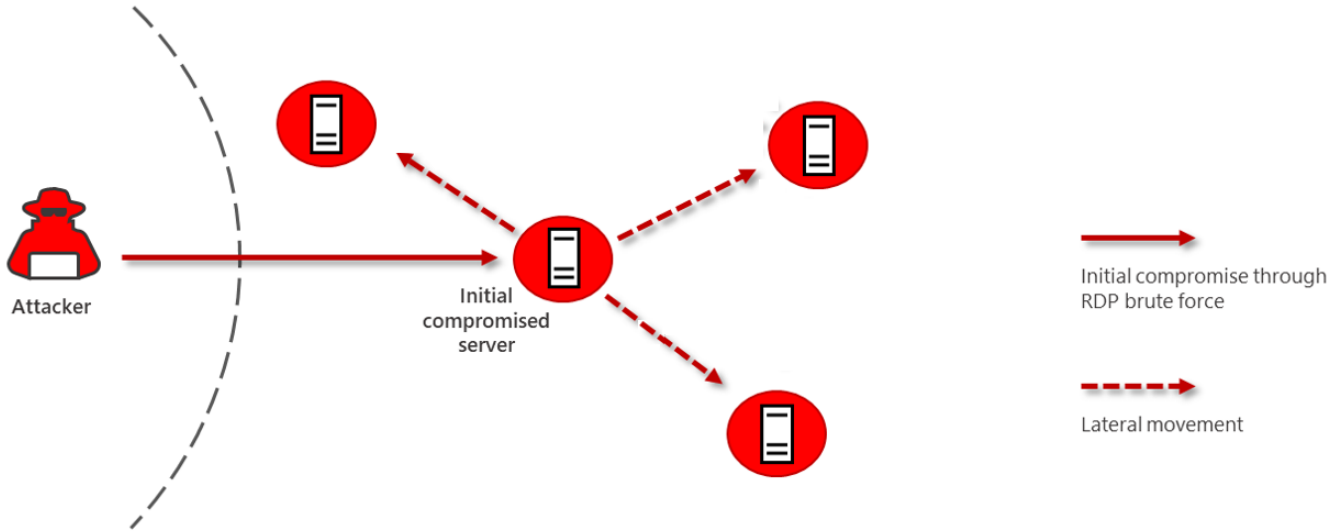
Post-explotación (II)

Post-explotación (cont.)

- Levantar un proceso como un *keylogger* (*software* encargado de capturar la información que recibe el equipo a través del teclado y ratón) u otro *software* espía para capturar información.
- Instalar un *rootkit* (*software* dañino que permite el acceso privilegiado a áreas de una máquina, mientras que al mismo tiempo se oculta su presencia mediante la corrupción del Sistema Operativo u otras aplicaciones) o *backdoor* en un sistema para poder acceder posteriormente sin tener que volver a explotar la vulnerabilidad.
- Utilizar un sistema explotado como pivote para poder escanear *host* internos desde la red externa.

Post-explotación (III)

Ejemplo de movimiento lateral:



Fuente: `https://www.microsoft.com/security/blog/2020/06/10/the-science-behind-microsoft-threat-protection-attack-modeling-for-finding-and\-stopping-evasive-ransomware/`

Bibliografía

K. Astudillo B.
Hacking Ético: Cómo Convertirse en Hacker Ético en 21 Días o Menos.
3ª edición. ISBN 978-84-9964-767-8. Ra-Ma, 2018.

M. Á. Caballero Velasco y D. C. Serrano.
El Libro del Hacker.
ISBN 978-84-415-3964-8. Anaya Multimedia, 2018.

P. González, G. Sánchez y J. M. Soriano
Pentesting con Kali Linux Rolling Release 2017.
2ª edición. ISBN 978-84-697-6035-2. 0xWORD, 2017.

R. Hertzog, J. O'Gorman y M. Aharoni.
Kali Linux Revealed: Mastering the Penetration Testing Distribution.
ISBN 978-0-9976156-0-9. Offsec Press, 2017.

Otros recursos I

Sitios web sobre virus y botnets

- https://www.incibe.es/
- http://www.virustotal.com
- https://www.f-secure.com/
- http://www.securelist.com/

Otros recursos II

Sitios web sobre vulnerabilidades

- http://cve.mitre.org
- https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project
- https://www.owasp.org/index.php/OWASP_Mobile_Security_Project
- https://owasp.org/www-project-internet-of-things/
- https://owasp.org/www-project-top-10-for-large-language-model-applications/
- http://www.kb.cert.org/vuls

Otros recursos III

Herramientas

- Herramientas gratuitas para proteger los dispositivos (ordenador, *smartphone*, *tablet*): `https://www.osi.es/herramientas`
- Herramientas de detección y desinfección de botnets: `https://www.osi.es/es/servicio-antibotnet`
- *Social-Engineer Framework*: `https://www.social-engineer.org/framework/general-discussion/`

Otros recursos IV

Guías y herramientas que ofrecen diferentes fuentes de consultas según el dato origen (teléfono, nombre y apellidos...)

- https://osintframework.com/
- https://ciberpatrulla.com/links/
- https://inteltechniques.com/tools/index.html
- https://ciberpatrulla.com/guias/
- https://web.telegram.org/k/#@UniversalSearchRobot
- https://osint.industries/
- https://epieos.com/
- https://github.com/sundowndev/PhoneInfoga
- https://ciberpatrulla.com/como-saber-quien-numero-telefono/
- Google, Bing, Duckduckgo, Yandex, Baidu, etc.