Penetration Test Report

NBN Corp

December 16th, 2022

Ping Chang

NYU Penetration Testing Student

ypc231@nyu.edu

Executive Summary

NYU Penetration Testing student was approached by NBN Corp to perform a red team style test that would allow the company to recognize what a threat source on the outside can achieve through exploiting the vulnerabilities that still exist within the system. This penetration test aims to provide NBN with major findings of vulnerabilities in the overall system and how they can be fixed. The goal is to get shell and eventually root on each machine. This penetration test began on Nov 16, 2022 and ended on Dec 16, 2022.

The student was provided with a client VM and a server VM. NBN did not provide any system access or credentials. These machines were attacked over the network only, by pivoting through the web server. System passwords and configurations were not changed, and no software were installed. A denial-of-service attack is outside the scope of this penetration test as NBN did not want the test to intentionally break anything or risk doing so.

Types of tests that were performed for this penetration test include a port scan to determine how one can gain access to the server, followed by examining the web pages for potential vulnerabilities. Attempts were made to crack the passwords of a client, CEO of the company, as well as the root of the server and client machines by using Hydra and John the Ripper. Proxychains was utilized to access information of the client machine.

The overall risk score of the findings from this penetration test was HIGH with immediate actions recommended. Major findings are remote code injection, cross-site scripting, sensitive data accessible to the public, authentication levels being easily changeable, account and session lockouts not implemented, no web server access encryption, no multi-factor authentication required, phpinfo page being publicly accessible as well as users having weak passwords.

Suggested fixes include implementing OWASP Enterprise Security API libraries, attaching timeout sessions for all users, using buffer overflow protection, sanitizing user input, adding website encryption, enforcing multi-factor authentication, and making sure password strengths are strong.

Introduction

NBN Corp approached a NYU Penetration Testing student after the company suffered a massive breach. The company was accessed from their internal facing servers and lost customers as well as employee data.

The company wishes for a red team styled penetration test to be performed on two images of systems from their network, a web server under construction that will be used for customer online account access and employee customer service as well as a client machine that is used for managing customer accounts with custom application, before being deployed for their website.

NBN believes that their adversaries are still targeting their external-facing web server. The goal of this penetration test is to identify vulnerabilities that still exist within the system and to provide mitigating actions the company can take to establish a secure posture.

It is important to know what a threat source on the outside can achieve, hence no system access or credentials were provided. NBN Corp had also specified that no changes to the system passwords and configurations were allowed, and no software can be installed. A denial-of-service attack, however, is outside the scope of this test.

This penetration test started on Nov 17, 2022 and finished on Dec 16, 2022 where multiple attempts at attacking the machines were made. Several flags in the system were found to illustrate that the system is vulnerable.

The findings from this penetration test show that the system is in HIGH risk and immediate actions are recommended. Major findings are remote code injection, cross-site scripting, sensitive data accessible to the public, authentication levels being easily changeable, account and session lockouts not implemented, no web server access encryption, no multi-factor authentication required, phpinfo page being publicly accessible as well as users having weak passwords.

Suggested fixes include implementing OWASP Enterprise Security API libraries, attaching timeout sessions for all users, using buffer overflow protection, sanitizing user input, adding website encryption, enforcing multi-factor authentication, and making sure password strengths are strong.

The NYU student can be contacted through ypc231@nyu.edu

Methodology

A Kali Linux Virtual Machine was used as the primary attack source for this penetration test as it has all the tools needed, including Nmap, Hydra, John the Ripper, Proxychains, and Nikto. The order of the exploitation has been slightly modified to match the order of the flags found as this makes for a more coherent explanation and report.

Risk ratings for vulnerabilities are interpreted using the OWASP Risk Rating Methodology where the score given to each finding is evaluated through the different factors as presented on the site. Only an impact level of over medium will be reported as they require more immediate responses.

The general approach of this penetration test was to perform a port scan first to determine where the attack can begin, then using that knowledge to gain access to different accounts.

The Kali machine was set up to create the route that allows for pinging the other interfaces to create tests:

```
┌──(kali㉿kali)-[~]
└─$ sudo ip route add 172.16.1.0/24 via 10.10.0.66
[sudo] password for kali:

┌──(kali㉿kali)-[~]
└─$ ping 10.10.0.66
PING 10.10.0.66 (10.10.0.66) 56(84) bytes of data.
64 bytes from 10.10.0.66: icmp_seq=1 ttl=64 time=0.362 ms
64 bytes from 10.10.0.66: icmp_seq=2 ttl=64 time=0.259 ms
64 bytes from 10.10.0.66: icmp_seq=3 ttl=64 time=0.183 ms
^C
── 10.10.0.66 ping statistics ──
3 packets transmitted, 3 received, 0% packet loss, time 2049ms
rtt min/avg/max/mdev = 0.183/0.268/0.362/0.073 ms

┌──(kali㉿kali)-[~]
└─$ ping 172.16.1.1
PING 172.16.1.1 (172.16.1.1) 56(84) bytes of data.
64 bytes from 172.16.1.1: icmp_seq=1 ttl=64 time=0.204 ms
64 bytes from 172.16.1.1: icmp_seq=2 ttl=64 time=0.221 ms
64 bytes from 172.16.1.1: icmp_seq=3 ttl=64 time=0.287 ms
^C
── 172.16.1.1 ping statistics ──
3 packets transmitted, 3 received, 0% packet loss, time 2057ms
rtt min/avg/max/mdev = 0.204/0.237/0.287/0.035 ms

┌──(kali㉿kali)-[~]
└─$ ping 172.16.1.2
PING 172.16.1.2 (172.16.1.2) 56(84) bytes of data.
64 bytes from 172.16.1.2: icmp_seq=1 ttl=63 time=0.593 ms
64 bytes from 172.16.1.2: icmp_seq=2 ttl=63 time=0.399 ms
64 bytes from 172.16.1.2: icmp_seq=3 ttl=63 time=0.440 ms
^C
── 172.16.1.2 ping statistics ──
3 packets transmitted, 3 received, 0% packet loss, time 2056ms
rtt min/avg/max/mdev = 0.399/0.477/0.593/0.083 ms
```

Figure 1: Pinging Target Machine

A ping test was then deployed to check connections to the server and remote clients. After successful connections, the testing began with attacking the server VM.

A Nmap scan was conducted to see what the open ports on the NBN server are:

```
┌──(root㉿kali)-[~]
└─# nmap -sV 10.10.0.66 -p-
Starting Nmap 7.93 ( https://nmap.org ) at 2022-12-05 23:33 EST
Nmap scan report for 10.10.0.66
Host is up (0.000094s latency).
Not shown: 65531 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.29 ((Ubuntu))
443/tcp   open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
8001/tcp  open  http    Apache httpd 2.4.29 ((Ubuntu))
65534/tcp open  ftp     vsftpd 3.0.3
MAC Address: 08:00:27:BC:64:54 (Oracle VirtualBox virtual NIC)
Service Info: OSs: Linux, Unix; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 25.59 seconds
```

Figure 2: Nmap Open Ports

From this Nmap scan, the ports that are open include port 80 which is serving up a website. The site was browsed, by opening a browser and entering the port number, to see if any vulnerabilities can be found. The page http://10.10.0.66, is the home page for NBN Corp with an option allowing users to log in. Looking at the source code by right-clicking and choosing the source code option, there are some comments written by the developer. This is a command injection attack that has been commented out.

```
<!-- Main -->
    <div id="main">

        <!-- One -->
            <section id="one">
                <div class="image main" data-position="center">
                    <img src="images/watchingyou.jpg" alt="" />
                </div>
                <div class="container">
                    <header class="major">
                        <p><!--DEBUG
$cmd = shell_exec( "echo '" . $_GET['email'] . " : " . $_GET['name'] . "' >> /var/www/html/data/customer.list " );
                        <h2>Near-Earth Broadcast News</h2>
                        <p>Connecting You to the World</br>
                        </p>
                    </header>
                    <p>The largest media conglomerate in the world, NBN operates five of the six top-rate

                    <p>We do more than simply read the communcation and entertainment market; we steer it
                </div>
            </section>
```

Figure 3: Command Injection

Through clicking on all the links available on the homepage, the employee login page source code gives information about how the passwords can be cracked, that the rockyou file on the Kali VM should be enough to guess/crack all passwords on the servers and that no additional mangling rules are needed.

```
<br />
(Hey, you, yes you! You can guess/crack all passwords on these servers <br />
using rockyou and without mangling rules. You should not have to spend <br />
days cracking and burning up your hardware. If you are, you're doing it wrong)<br />
    </div>
<hr />
```

Figure 4: How to Crack All Passwords

Continuing looking at possible vulnerabilities through the webpages, a robots.txt file (10.10.0.66/robots.txt) seems to be protecting two other directories, /internal/ and /data/. These two directories can be browsed directly without a need for passwords or certain admin rights.

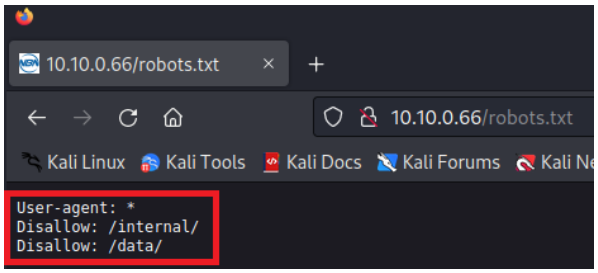

Figure 5: robots.txt

Looking at one of the mentioned directory from robots.txt, the data directory, there are two flag files, flag1 and a jpg file for flag4. Flag4 can't be accessed yet as clicking on it results in a permission deny page, so flag1 was examined first.

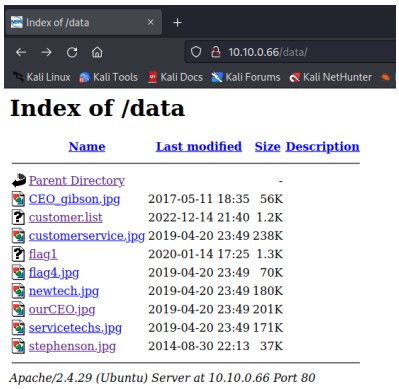

Figure 6: /data/ directory

Zooming out enough on the message presented on the flag1 directory, the message can be seen, **FLAG1{CYBERFELLLOWS_GOODLUCK}**. Since the flag is in the desired structure, no extra modifications were conducted.

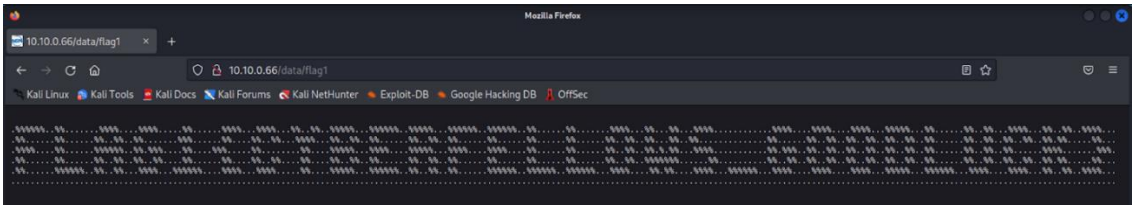

Figure 7: FLAG1{CYBERFELLOWS_GOODLUCK}

Other important information from the /data directory includes a customer.list, which shows a list of customers along with the emails used for their data. There is also a jpg file that is named CEO_gibson in the /data/ directory. The CEO of the company may be named gibson and that important information may be stored in his account. A Hydra command was used to see if a password can be cracked using gibson as the username:

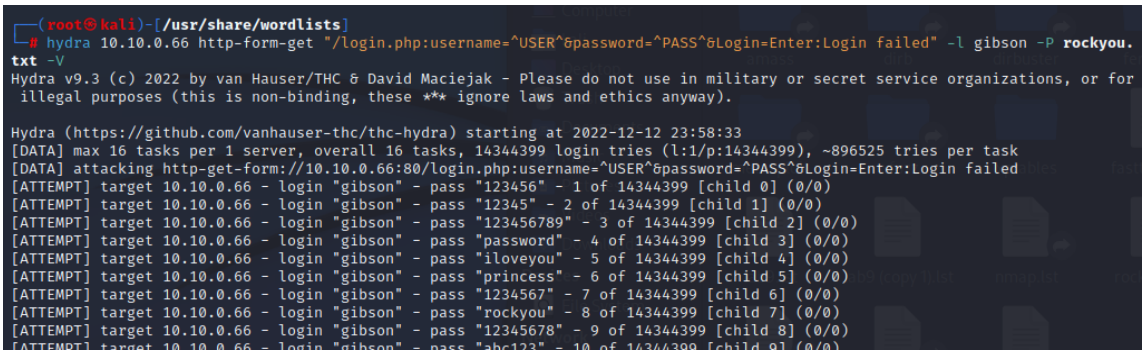

Figure 8: Hydra gibson

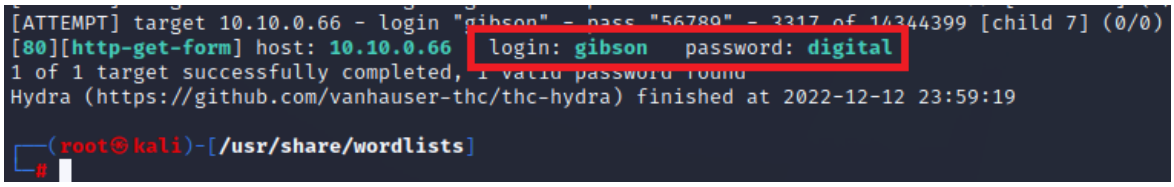

Figure 9: Password Cracked for gibson

The attempt has led to knowing the password for gibson is “digital”. Login with gibson’s password through the website, a “Future Customer List” link takes the user to flag2, **flag2{down_a_rabbithole}**. Since flag2 is in the desired format, no additional changes were made.

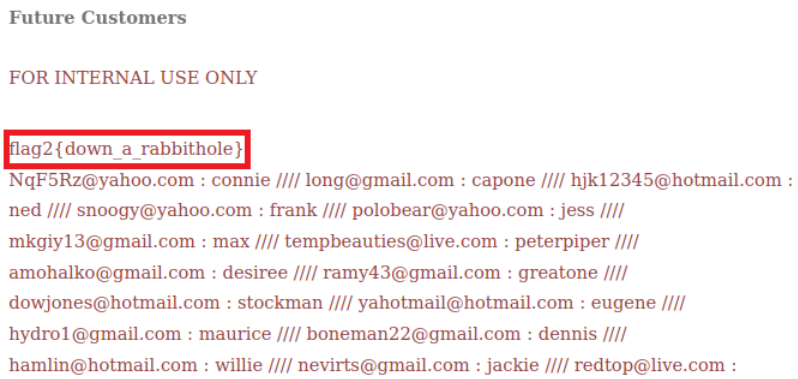

Figure 10: flag2{down_a_rabbithole}

A list of names and emails of future customers can also be seen on this page even though it is labeled as "for internal use only".

Using the Kali machine, recall that from Figure 2, port 443 was also open after the Nmap scan and it was running through ssh. With the newly acquired password for gibson, an attempt at getting a shell was made through port 443.

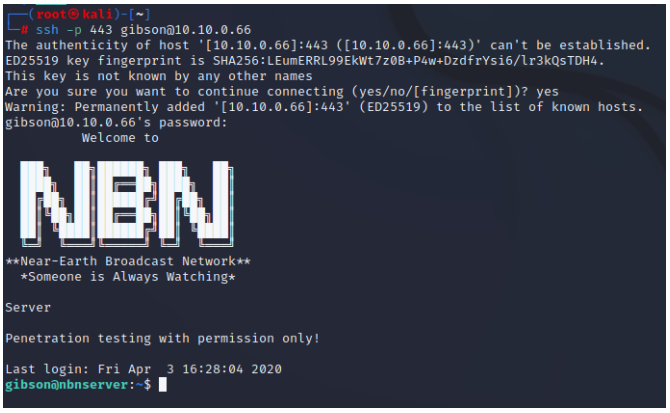

Figure 11: gibson ssh

Once gibson's page can be entered, any user can look at all pages and directories the CEO has access to. Flag3 is the only file listed in the list of files, **flag3{brilliantly_lit_boulevard}**. Again, since flag3 is in the correct format, no additional changes were needed.

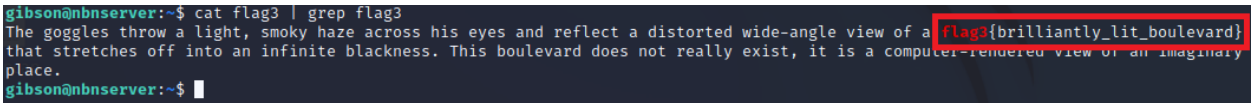

Figure 12: flag3{brilliantly_lit_boulevard}

Not all sudo rights are available through using gibson's server, where only three commands can truly be run. Additionally, it seems that gibson does not have access to Nmap. However, privilege escalation can be used to view the shadow file containing an encrypted password for the root.

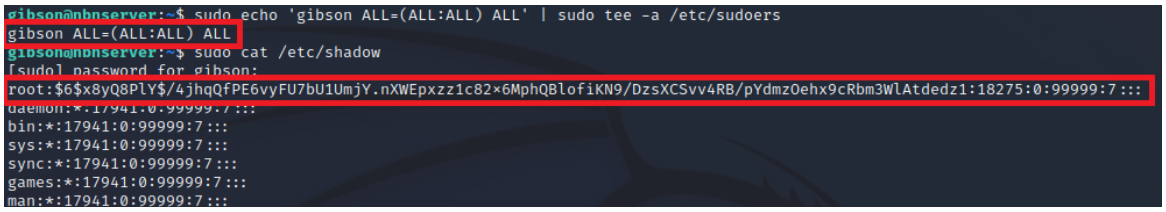

Figure 13: Encrypted root Password

The password can then be cracked using John the Ripper with the rockyou text file again:

```
┌──(kali㉿kali)-[~/Desktop]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt password
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
digital          (gibson)
1986angeles      (root)
2g 0:00:57:53 DONE (2022-12-12 08:54) 0.000575g/s 3764p/s 3764c/s 3764C/s 1986c03..1986805
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Figure 14: Cracked root Password

Logging in through root is now possible with the cracked password. With root, more files can be explored, such as the file location for flag4. Using the same method as how flag3 was found, flag4 can be viewed within the /var/www/html/data directory, **flag4{youre_going_places}**.

```
gibson@nbnserver:/var/www/html/data$ sudo 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:Descrip
tion flag4="flag4{youre_going_places}" xmlns:MicrosoftPhoto="http://ns.microsoft.com/photo/1.0/" /></rdf:RDF></x:x
mpmeta>
gibson@nbnserver:/var/www/html/data$
```

Figure 15: flag4{youre_going_places}

Recall that from the /data/ directory, there was someone named stephenson. With the mysql file in the shadow file, the password of stephenson can be found to be used later when attacking the client VM.

```
gibson@nbnserver:~$ sudo cat /root/.mysql_history
connect
connect 127.0.0.1
show databases;
create
help;
connect;
connect
create database nbn;
use nbn;
```

Figure 16: gibson mysql_history

```
select * from users;
insert into users  values (1, "stephenson", "stephenson", "stephenson", "942cbb4499d6a60b156f39fcbaacf0ae", "data
/stephenson.jpg", "2029-12-12 01:23:45", 123);
insert into users  values (3, "stephenson", "stephenson", "stephenson", "942cbb4499d6a60b156f39fcbaacf0ae", "data
/stephenson.jpg", "2029-12-12 01:23:45", 123);
select * from users;
update users set avatar = "data/ourCEO.jpg" where user_id = 1;
select * from users;
gibson@nbnserver:~$
```

Figure 17: stephenson Hashed Password

Using a password cracker online, the encrypted password was in the form of a md5 string, where after converting the text, the password of stephenson can be seen in the figure below.

Hash	Type	Result
942cbb4499d6a60b156f39fcbaacf0ae	md5	pizzadeliver

Figure 18: stephenson Cracked Password

The keys can also be found within gibson's server, in a backup directory, where after adding the keys into the Kali VM, the user may gain root without having to login using a password:

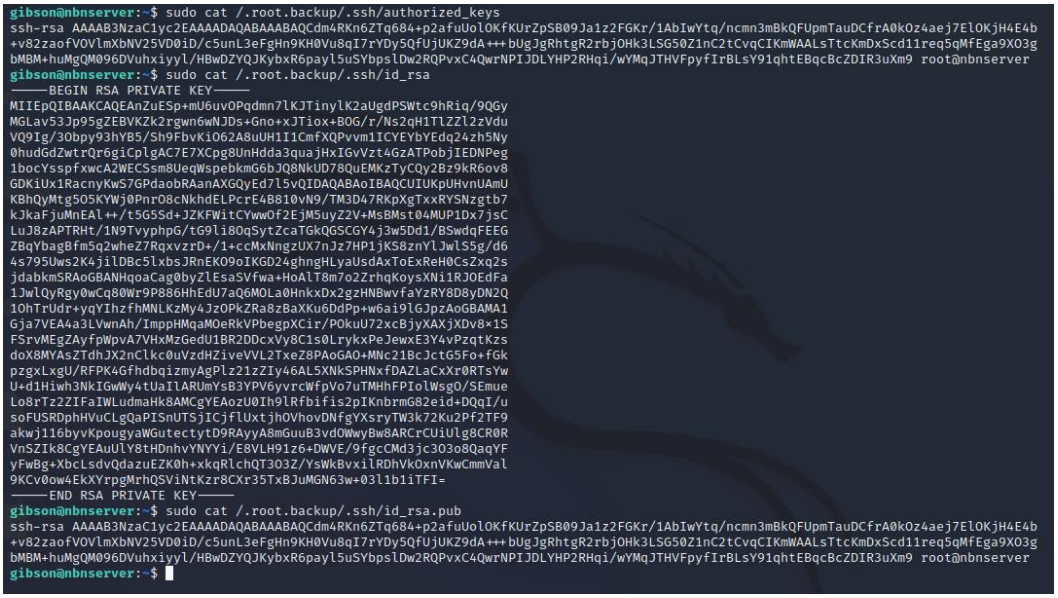

Figure 19: ssh keys

After getting root on the server VM, it is now time to attack the client VM. A proxy can be set up using gibson's root access as well as proxychains to scan the client with Nmap. With the password for stephenson, the next step is to ssh to the client to see what files from the client side are exploitable.

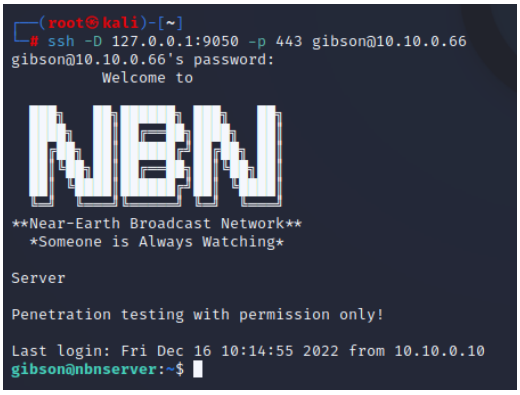

Figure 20: Kali Machine gibson

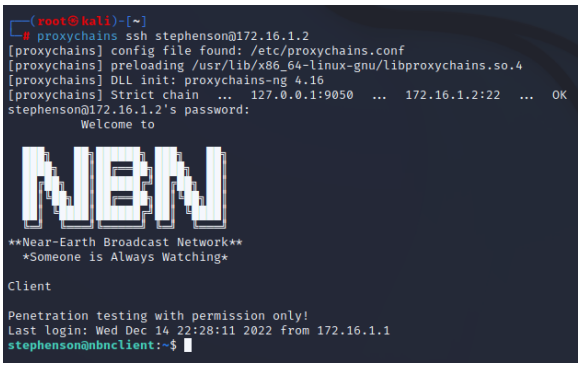

Figure 21: proxychains ssh stephenson

On the client side, the user can view the status of the active processes and see there is a ping command from the client to the NBN server that is sending a hex message. This message can be decoded to reveal that it is **flag6{listen}**.

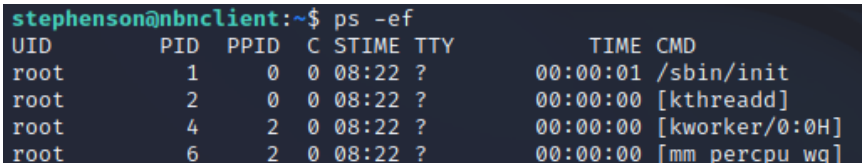

Figure 22: View Status of Active Processes

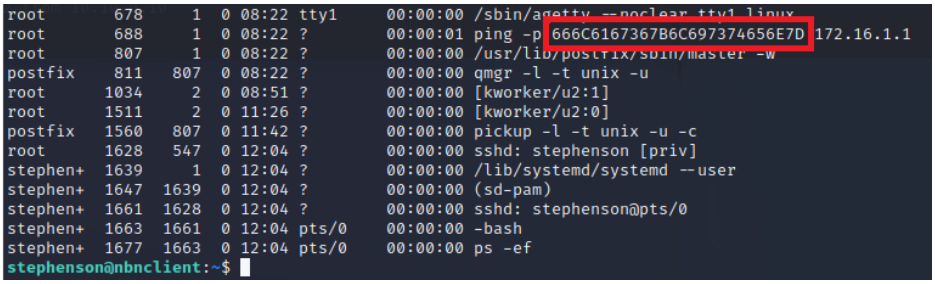

Figure 23: Hex Message Ping Command

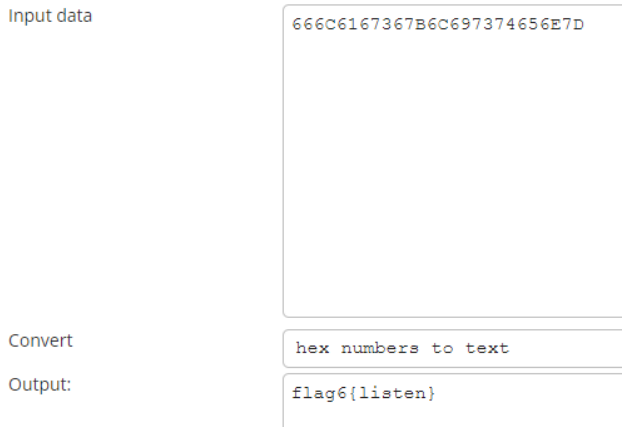

Figure 24: flag6{listen}

Similar to seeing what files gibson has access to, Flag7 seems to be in the list of files for stephenson, but when viewing the contents, the flag is not in the desired structure. The code seems to be in a base64 format that can be converted to a png image file. By running the Base64 code through a converter, the below image shows the flag in the form that is the same as the other flags found, **flag7{worlds_within_worlds}**.

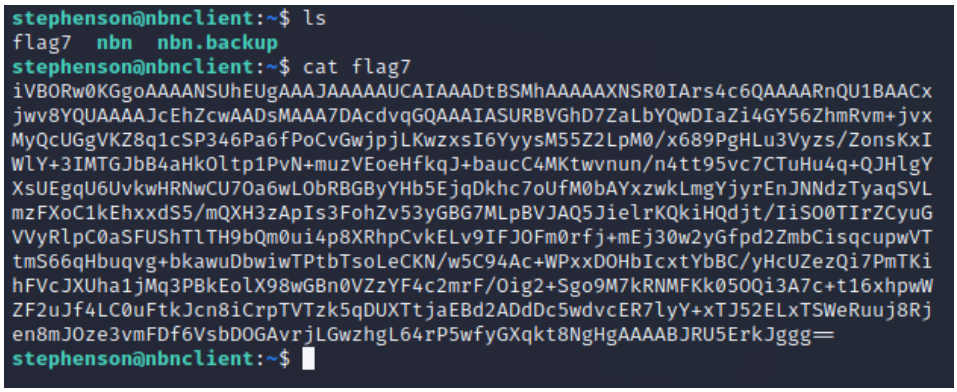

Figure 25: base64 flag7

Figure 26: flag7{worlds_within_worlds}

Same with gibson, the client stephenson only have a limited amount of sudo rights. But by going into the ./nbn directory, the user is presented with the NBN Customer Management Portal where anyone can perform the actions as presented in the figure below.

```
stephenson@nbnclient:~$ sudo ./nbn

***** 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 :
```

Figure 27: NBN Customer Management Portal

Findings

Major findings/vulnerabilities of this penetration test are as follow:

Weak passwords, hidden directories with sensitive data, authentication levels can be easily changed, no account lockout implemented, no ssh session timeout, remote code injection, cross site scripting, no web server access encryption, publicly accessible phpinfo page, and no multi-factor authentication required.

Each finding is broken down into its own section with a description of the vulnerability, how it was found, the impact it has, the risk rating, as well as remediation recommendations.

Weak Passwords

Description	All users that were used as attempts to login for this penetration test, root, Gibson, and Stephenson, have weak passwords that could be cracked easily with the rockyou.txt file in Kali.
Method of Discovery and Exploitation	By using Hydra or John the Ripper, these passwords can be cracked through brute force methods by entering the username into the system.
Impact to Assets	Outsiders can easily access inside information resulting in data leak.
Risk Rating	High
Remediation Recommendations	Complex passwords should be used, including upper and lowercase letters, numbers, as well as special characters (!, ?. #, etc.) The length of password should also increase to add difficulty to outsiders trying to crack passwords.

Hidden Directories with Sensitive Data

Description	robots.txt is used to hide sensitive directories of /internal/ and /data/
Method of Discovery and Exploitation	By going to the /robots.txt directory after getting on the NBN website through the port 10.10.0.66
Impact to Assets	The contents of both the directories can be viewed even if they are hidden. Sensitive customer information from the customer.list file should not be accessible to the public.
Risk Rating	High
Remediation Recommendations	Use OWASP ESAPI applications and services to protect the contents of the files from attackers

Authentication Levels

Description	The authentication cookie only controls the login authentication with a header of either 0 or 1.
Method of Discovery and Exploitation	A web proxy tool can be used to manually change the authenticated value from 0 to 1.
Impact to Assets	Can gain access to unauthorized web pages and data.
Risk Rating	High
Remediation Recommendations	Use OWASP ESAPI Authentication features.

No Account Lockout Implemented

Description	Outsiders can repeatedly try passwords to attempt logging in.
Method of Discovery and Exploitation	Users can repeatedly try logging in through the login page.
Impact to Assets	Outsiders have infinite chances at cracking the passwords to get valuable information stored within the accounts.
Risk Rating	Medium
Remediation Recommendations	After a certain number of failed login attempts, the session should timeout and set a period where no one can attempt to log in through the machine used.

No SSH Session Timeout

Description	ssh sessions can persist indefinitely allowing outsiders to access information.
Method of Discovery and Exploitation	ssh sessions ran on Kali VM did not have to restart throughout the entire duration of the testing process.
Impact to Assets	Outsiders have infinite time at cracking the passwords to get valuable information stored within the accounts.
Risk Rating	Medium
Remediation Recommendations	A time limit should be set for inactivity and log the user out of the session.

Remote Code Injection

Description	Unauthorized execution of the server through injecting malicious code.
Method of Discovery and Exploitation	An example was shown on the source code of the NBN web homepage.
Impact to Assets	Outsiders can steal confidential information, modify, and destroy files.
Risk Rating	High
Remediation Recommendations	Use buffer overflow protection and sanitize user input.

Stored XSS Attack

Description	Malicious scripts that are injected into website pages.
Method of Discovery and Exploitation	customer.list file include malicious scripts such as UNION ALL select NULL
Impact to Assets	The contents of the customer list and database may be damaged, and information cannot be retracted.
Risk Rating	High
Remediation Recommendations	Inputs should be filtered on arrival, where the expected outputs are clearly stated, and the machine should reject any code that is not in the correct format.

Web Server Access Encryption

Description	The website is not encrypted. There is no protocol when sending data, such as logging in.
Method of Discovery and Exploitation	No HTTPS web server access.
Impact to Assets	Data transfer is unsafe, and the transfer traffic can be snooped.
Risk Rating	High
Remediation Recommendations	Use website encryption such as HTTPS instead of HTTP.

PHPinfo Page Leak

Description	The php info page can be accessed
Method of Discovery and Exploitation	Accessing phpinfo()
Impact to Assets	Knowing the structure of the filesystem may allow outsiders to execute directory traversal attacks.
Risk Rating	Medium
Remediation Recommendations	Check if the phpinfo() function was not removed. Hide php using html types for php extensions.

Multi-Factor Authentication

Description	No multi-factor authentication methods are needed to login as any user.
Method of Discovery and Exploitation	Anyone can login using the webpage if the username and password are correct.
Impact to Assets	If a password is leaked, any outsider can login to the system and retrieve valuable information.
Risk Rating	Medium
Remediation Recommendations	Implement multi-factor authorization methods such as using a phone-number, a trusted device, etc.

Conclusion

This penetration test aims to provide NBN with major findings of vulnerabilities in the overall system and how they can be fixed. The goal is to get shell and eventually root on each machine.

NBN did not provide any system access, with only a client VM and a server VM given. These machines were attacked over the network only. System passwords and configurations were not changed, and no software were installed. A denial-of-service attack is outside the scope of this penetration test.

Types of tests that were performed for this penetration test include a port scan to determine how one can gain access to the server, followed by examining the web pages for potential vulnerabilities. Attempts were made to crack the passwords of a client, CEO of the company, as well as the root of the server and client machines by using Hydra and John the Ripper. Proxychains was utilized to access information of the client machine.

The student was able to gain root on the server machine and access the customer management portal through the client machine. Major vulnerabilities were found where the overall risk score of the findings from this penetration test was HIGH with immediate actions recommended. Major findings are remote code injection, cross-site scripting, sensitive data accessible to the public, authentication levels being easily changeable, account and session lockouts not implemented, no web server access encryption, no multi-factor authentication required, phpinfo page being publicly accessible as well as users having weak passwords.

Suggested fixes include implementing OWASP Enterprise Security API libraries, attaching timeout sessions for all users, using buffer overflow protection, sanitizing user input, adding website encryption, enforcing multi-factor authentication, and making sure password strengths are strong.

Appendix

Appendix A: Flags

FLAG	HOW TO ACCESS
FLAG1{CYBERFELLOWS_GOODLUCK}	By going to the data directory (10.10.0.66/data), there is a file called flag1. A text of flag1 will be seen upon clicking on the file after scaling out.
flag2{down_a_rabbithole}	Login with gibson's credentials on the site, then click on the "Future Customer List" link.
flag3{brilliantly_lit_boulevard}	Login with gibson's password with ssh to the server, then use the command "cat flag3 | grep flag3".
flag4{youre_going_places}	After getting root on the server VM, use the command "sudo strings flag4.jpg | grep flag" to view the text of the flag4 jpg file.
flag6{listen}	With stephenson's server, use the command "ps -ef" and a ping from the client to nbn server in the form of a hex message can be converted to the flag6 text.
flag7{worlds_within_worlds}	Login with stephenson as a client, then use the command "cat flag7" to find a base64 message that can be converted to the flag7 text.

Appendix B: Open Ports

PORT	SERVICE	VERSION
80	http	Apache httpd 2.4.29 ((Ubuntu))
443	ssh	OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
8001	http	Apache httpd 2.4.29 ((Ubuntu))
65535	ftp	vsftpd 3.0.3

Appendix C: Usernames and Passwords

USERNAME	PASSWORD	HOW TO GET
gibson	digital	Using Hydra along with the rockyou.txt file in Kali on the username "gibson"
root	1986angeles	Using John the Ripper with the rockyou.txt file in Kali
stephenson	pizzadeliver	Getting a hashed password from /root/.mysql_history, then convert it to text

Appendix D: Nikto Vulnerability Scan Report

```
┌──(root㉿kali)-[~]
└─# nikto -h 10.10.0.66
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          10.10.0.66
+ Target Hostname:    10.10.0.66
+ Target Port:        80
+ Start Time:         2022-12-15 00:40:02 (GMT-5)
---------------------------------------------------------------------------
+ Server: Apache/2.4.29 (Ubuntu)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of X
SS
+ 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
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Entry '/internal/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ OSVDB-3268: /data/: Directory indexing found.
+ Entry '/data/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ "robots.txt" contains 2 entries which should be manually viewed.
+ Apache/2.4.29 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch
.
+ Cookie authenticated created without the httponly flag
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ /phpinfo.php: Output from the phpinfo() function was found.
+ OSVDB-3092: /data/: This might be interesting...
+ OSVDB-3092: /internal/: This might be interesting...
+ OSVDB-3092: /manual/: Web server manual found.
+ OSVDB-3233: /phpinfo.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of sys
tem information.
+ OSVDB-3268: /manual/images/: Directory indexing found.
+ OSVDB-3268: /images/: Directory indexing found.
+ OSVDB-3233: /icons/README: Apache default file found.
+ /login.php: Admin login page/section found.
+ 7891 requests: 0 error(s) and 19 item(s) reported on remote host
+ End Time:           2022-12-15 00:41:01 (GMT-5) (59 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```