Limited Executive Summary

The aim of this assessment was to enhance the protection of sensitive data held by the company. Below are some of the vulnerabilities that were identified during the test:

1. FTP server allows anonymous logins.
2. HTTP used for website traffic.
3. Anyone on the network is allowed access to the staging server.
4. In production, the username parameter is vulnerable to XSS attacks.
5. In staging, the username parameter is vulnerable to SQLi attacks.

The fixes to these vulnerabilities are as follows:

1. Disable FTP Anonymous Authentication
2. Switching to HTTPS
3. Add firewall rules so that the staging server can only communicate with company employees.
4. Sanitize username input and use CSP.
5. Use prepared statements with parameterized queries

Overall, the average security score of the organization is 7 because there are some very important security flaws to be addressed.

Introduction

The primary objective of this penetration test is to assess the susceptibility of company assets to unauthorized access and to aid the company in enhancing its security posture through actionable guidance. An external penetration test of the network, an external penetration test of the web applications, and an internal penetration test of the company network (if access is obtained) will be conducted.

The parties involved in the test and their contact information is outlined here:

Position	Name	Contact Information
Principal Penetration Tester	Areej Ali	areeja132000@gmail.com

The timeline of the penetration test will be as follows:

Phase of penetration test	Date
Pre-engagement interactions	3 days
Intelligence gathering	1 week
Threat Modelling	1 day
Vulnerability Analysis	1 week
Exploitation	1 week
Post Exploitation	1 week
Reporting	3 days

The cost of the penetration test will be $4000.

Scope

Targets include all internet facing hosts, services, and web applications. The internal network can also be made subject to the penetration test if access is obtained.

As outlined by the company, the following items are out of scope:

- Vendor-hosted VPN provider
- Physical penetration test
- Existing NBN Subs and BP accounts
- Distributed Denial of Service attacks

Rules of engagement will be determined in the pre-engagement interactions.

Methodology

Testing

The types of tests that will be carried out are as follows:

1. External Network Pen Testing
    - All internet facing hosts and services
2. External Web App Pen Testing
    - All internet facing web applications
3. Internal Network Pen Test
    - If penetration tester can gain internal network access

Steps

The steps followed in this penetration test are based on the Penetration Testing Execution Standard (PTES)

1. Pre-engagement interactions

This will allow the penetration tester to gain further details about the test. Some questions to be addressed include the following:

a. What is in/out of scope (e.g., social engineering)
b. Confirm what equipment is owned by the company (e.g., the DNS server, the email server, the hardware on which the web servers run, firewall/IDS/IPS)
c. Confirm the dates and times (after hours, during weekends) during which it is acceptable for the penetration tester to engage in port scanning, enumeration, and exploitation
d. Ask if the penetration test is being conducted to meet a specific compliance requirement

2. Intelligence gathering

Intelligence gathering can be further separated into roughly three steps. The steps are as follows:

a. OSINT and recon-ng to obtain list of users, emails

b. Determine domains and network blocks owned (WHOIS lookup, BGP looking glass, recon-ng)
c. Port scanning (with Nmap) and banner grabbing to obtain information about which applications, hosts, and services are being run. Shodan will be used to identify further services

3. Threat Modelling

Threat modelling is done to determine which assets hold the most importance to the company and what threat actors could be looking for. It provides additional context to determine the risk score in later steps. Threat modelling can be further divided into the following four steps:

a. Get company documents outlining policies, plans, and procedures
b. Identify assets and determine rank them in order of importance
c. Identify threat actors and categorize them by capability
d. Determine likely targets of relevant threat actors

4. Vulnerability Analysis
    a. Look through port scanning output (generated by **nmap**) to manually determine if there are any other services that could be vulnerable.
    b. Look through the **metsploit framework** to see if any of the services running have any published vulnerabilities.
    c. Look through **exploit db** to see if any of the services running have any published vulnerabilities.
    d. Determine if **hydra** can guess client passwords
    e. Examine web applications for vulnerabilities like SQL injections, XSS, CSRF, and more (manually and through automated tools like **zaproxy**).

5. Exploitation

This phase of the penetration test will focus on bypassing security.

6. Post Exploitation

This phase will focus on determining whether the machine compromised can lead to a compromise of a valued asset of the company. Furthermore, the penetration tester will look to maintain control of the machine compromised for future use.

7. Reporting

The results of the penetration test are recorded.

Risk Scoring Methodology

Risk is determined by considering the factors of likelihood and impact (Risk = Likelihood * Impact). The following table incorporates likelihood/ease-of-exploitation (easy, moderate, or requiring social engineering) and the resulting impact (usually being a compromise of company assets), to assign a score indicating the severity of the security flaw.

The following table and CVSS V3 score ranges will be used to assess risk:

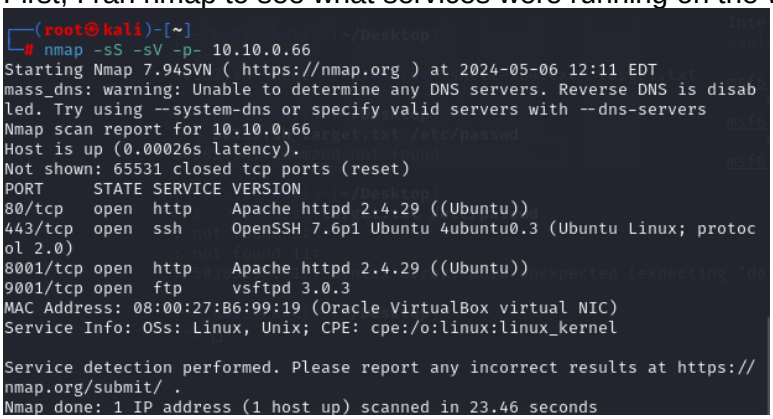
CVSS Score Range	Severity	Definition
9-10	Critical	Exploitation is easy and attacker gains full control of a system. This control gives them access to company assets. Should be patched immediately.
7-8.9	High	Exploitation is more difficult, but elevated privileges can be obtained, and company assets could be compromised. Should be patched as soon as possible.
4-6.9	Moderate	Exploitation can be achieved, but social engineering is required. It is recommended that higher priority security flaws be addressed first.
0.1-3.9	Low	The vulnerability is not exploitable because of the way the system is set up or other security controls that are in place. It is recommended that these be solved whenever maintenance is done in the future

Findings

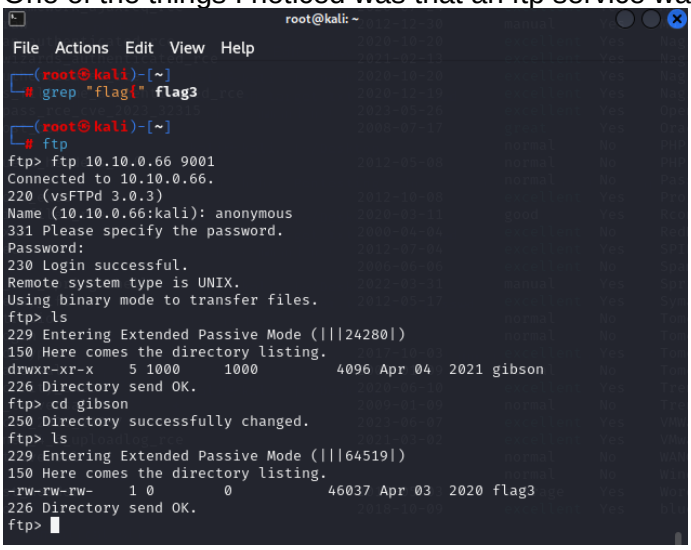
My findings are listed below:

1. FTP server allows anonymous logins (CVSS Score: 9)
   I. How this was discovered and exploited

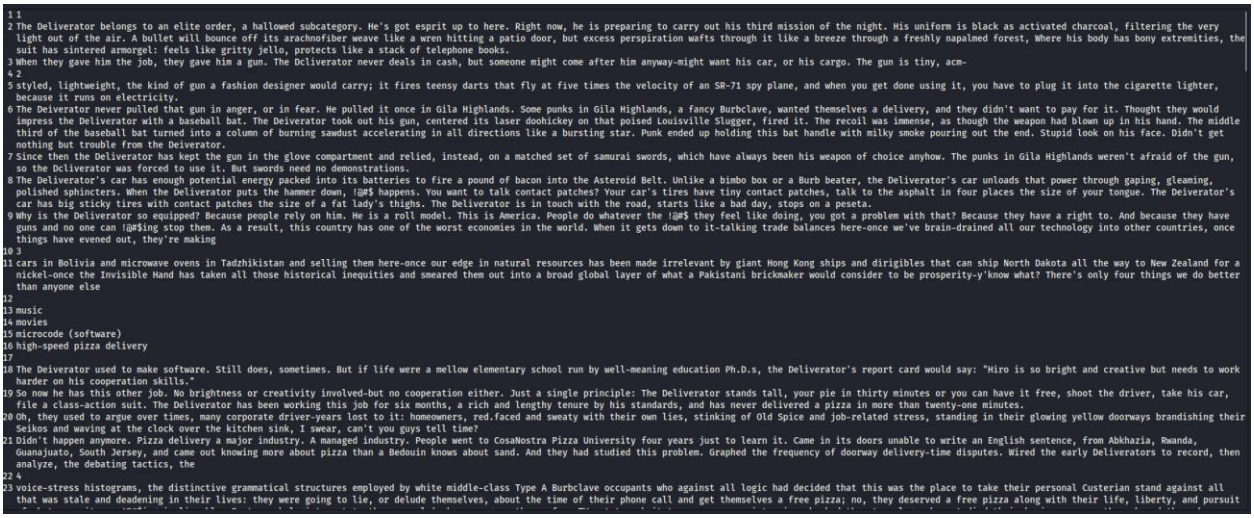
First, I ran nmap to see what services were running on the web server



One of the things I noticed was that an ftp service was running on port 9001.



We can try logging into ftp anonymously. Specifically, by passing in “anonymous” as the username and anonymous@gmail.com as the password. Ftp will allow access. Now, we can use the “ls” command to see what is inside the directories. As you can see, there is a flag3 file inside “gibson”. We can download this file. If we look at what is inside flag3, we see the following:



In this text, there is the following flag: flag3{brilliantly_lit_boulevard}. According to the contract, this is sensitive data. Thus, this is an important vulnerability.

II. Rationale behind CVSS Score

The rationale behind this vulnerability's score is that it is very easy to execute this attack and it allows the attacker access to sensitive data.

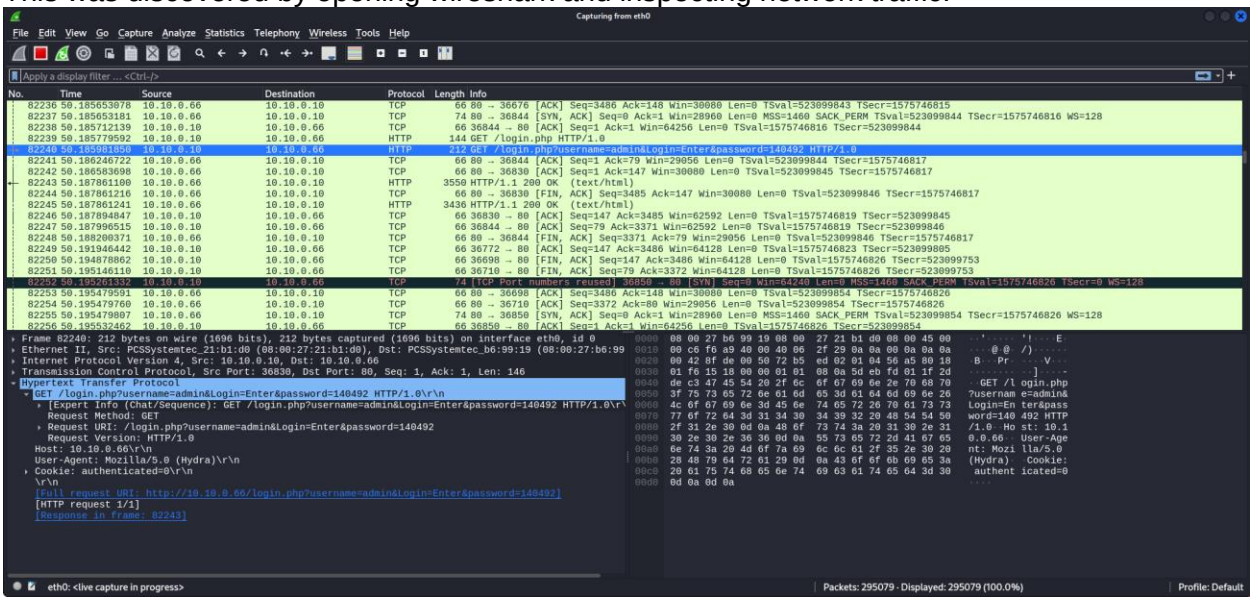
III. How to fix

To fix this, the company would need to disable FTP Anonymous Authentication. This can be done by following the outlined steps:
1. From the Internet Information Services (IIS) Manager, go to the Connections pane.
2. Now, expand Sites and then click on the site name.
3. Click on the FTP Authentication feature, select Anonymous Authentication, and disable Anonymous Authentication in the Actions pane.

2. HTTP used for website traffic (CVSS Score: 8.9)

I. How this was discovered and could be exploited

This was discovered by opening wireshark and inspecting network traffic.



As you can see from this screenshot of Wireshark, this website uses HTTP instead of HTTPS. This means that the credentials being entered by clients are in plaintext and vulnerable to Man in the Middle attacks. Due to this, an attacker can just open wireshark, sit on the network, and collect client credentials.

II. Rationale behind CVSS Score

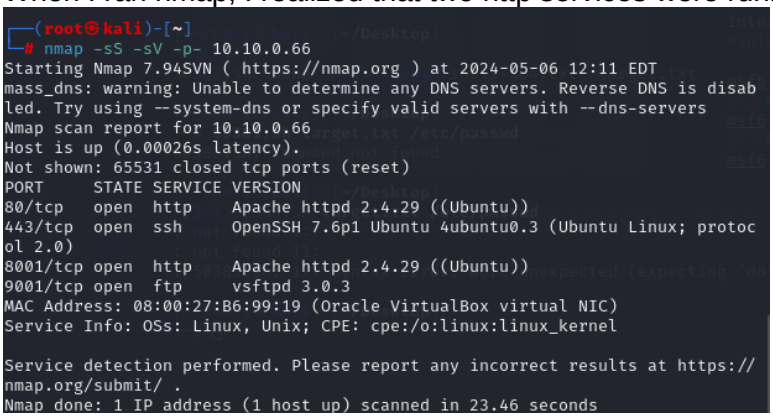
The rationale behind this vulnerability's score is that it is very easy to execute this attack and it allows the attacker access to sensitive data (client account credentials).

III. How to fix

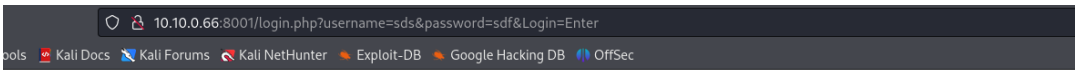
Configure website to use HTTPS instead of HTTP. This will make it so that traffic is encrypted with an SSL certificate, which will prevent a man in the middle from being able to read sensitive data from the traffic being exchanged.

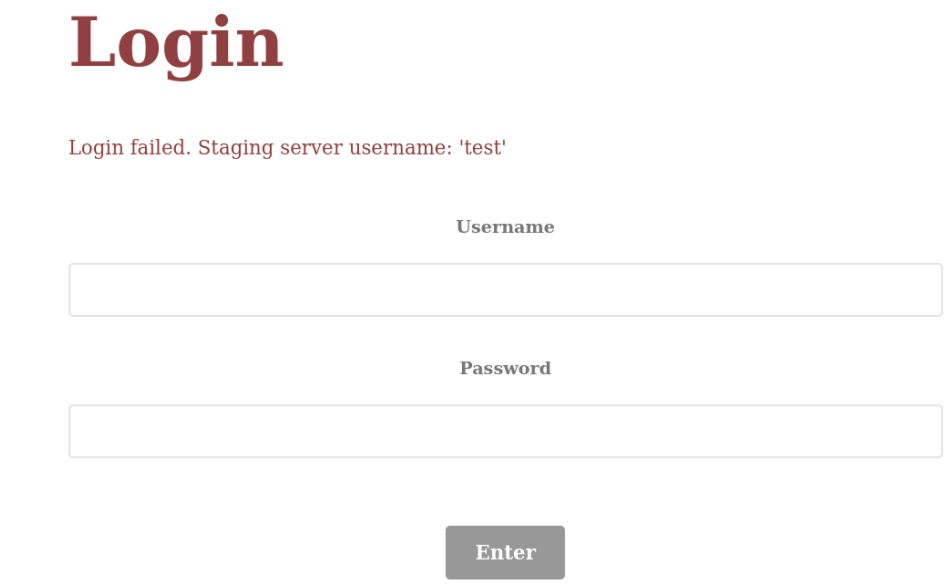
3. Anyone on the network is allowed access to the staging server (CVSS Score: 7)
   I. How this was discovered and could be exploited

When I ran nmap, I realized that two http services were running:

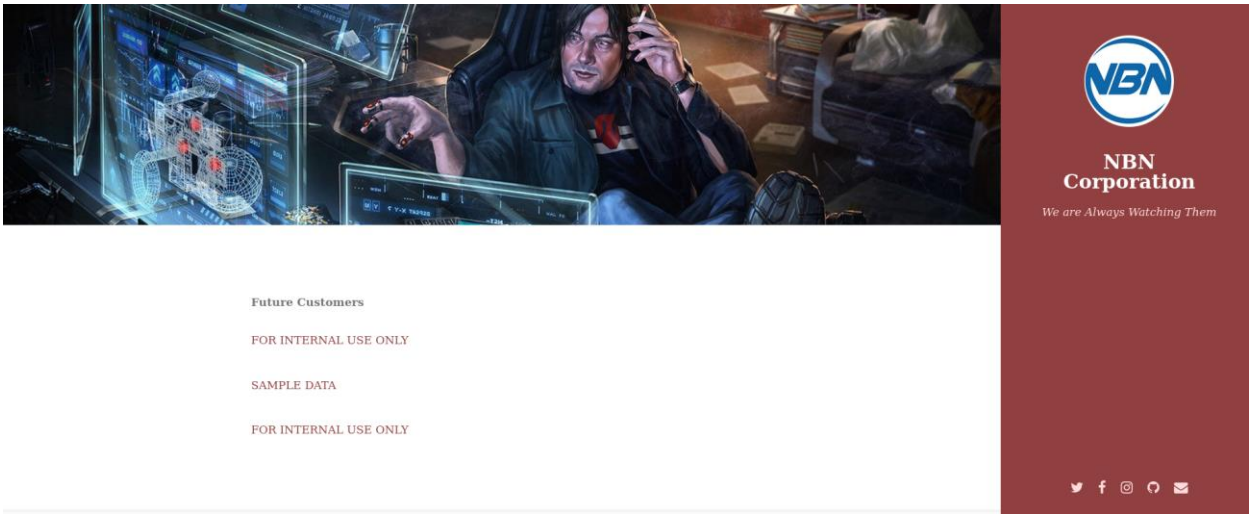


The service running on port 80 is more secure than the one running on port 8001. Port 8001 hosts the staging server. To visit the staging server, you can go to 10.10.0.66/8001. After that, you can click on employee login. Logging in from here is very simple because it does not matter whether you know the password or not. If you fail once, the server will tell you that the username needs to be “test” and the password can be anything.





Now, once you get in, you can click on “Future Customer List” and then you get to see a page that should only be seen by company employees.



The sample data could be a placeholder for sensitive information at some point during staging. For example, a developer might be using their own information as sample data to test something. Thus, we have found a vulnerability.

II. Rationale behind CVSS Score

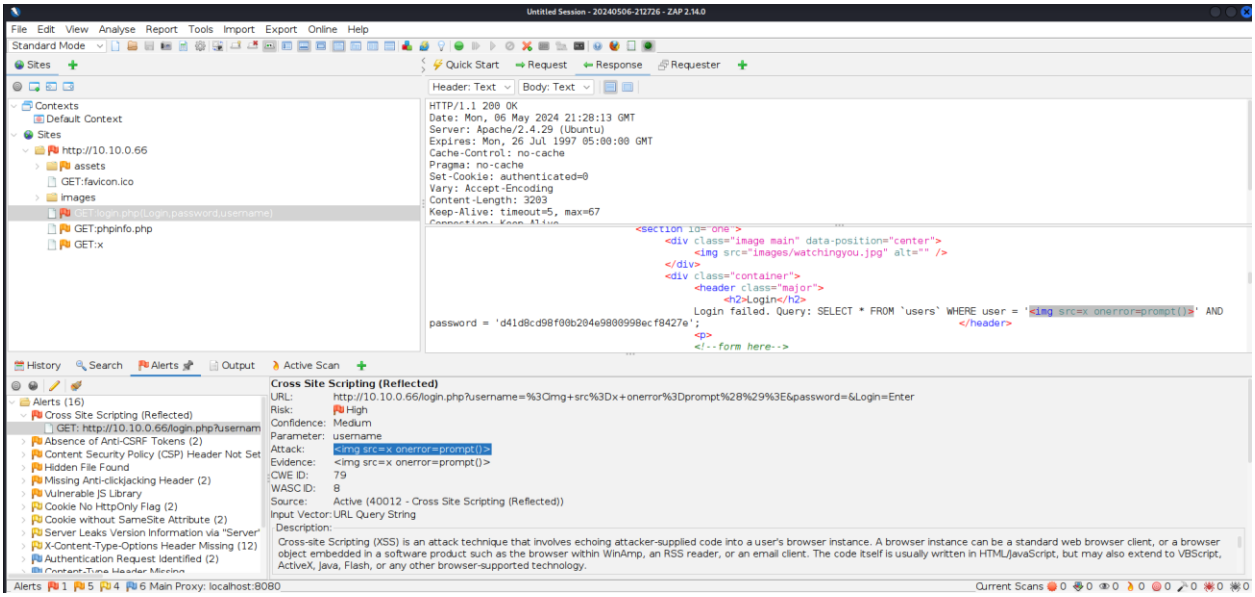
The rationale behind this vulnerability's score is that it is very easy to execute this attack and it allows the attacker access to potentially sensitive data ("SAMPLE DATA"). However, if best practices are followed, then "SAMPLE DATA" may not have any sensitive information, which is why the CVSS Score is lower than previous vulnerabilities.

III. How to fix

Add a firewall and configure it so that the staging server cannot communicate with anyone but company employees (as opposed to anyone over the web). Also, I would recommend putting the staging server on its own separate VM so that it has its own IP. This will make it so that the entire product does not have a single point of failure. Also, production will be touched as little as possible this way.

4. In production, the username parameter is vulnerable to XSS attacks (CVSS Score: 6)
   I. How this was discovered and could be exploited

When I got to the login portion of the staging server, I ran zaproxy. Then, I submitted some input on the website and used zaproxy to fuzz and see if any input indicated a successful attack:



As you can see, if “<img src=x onerror=prompt()>” is entered into the username field, a prompt comes up. This indicates that the page is vulnerable to XSS. A man in the middle could utilise this vulnerability in conjunction with the fact that they can see all traffic to send phishing links to clients (via an MitM attack).

   II. Rationale behind CVSS Score

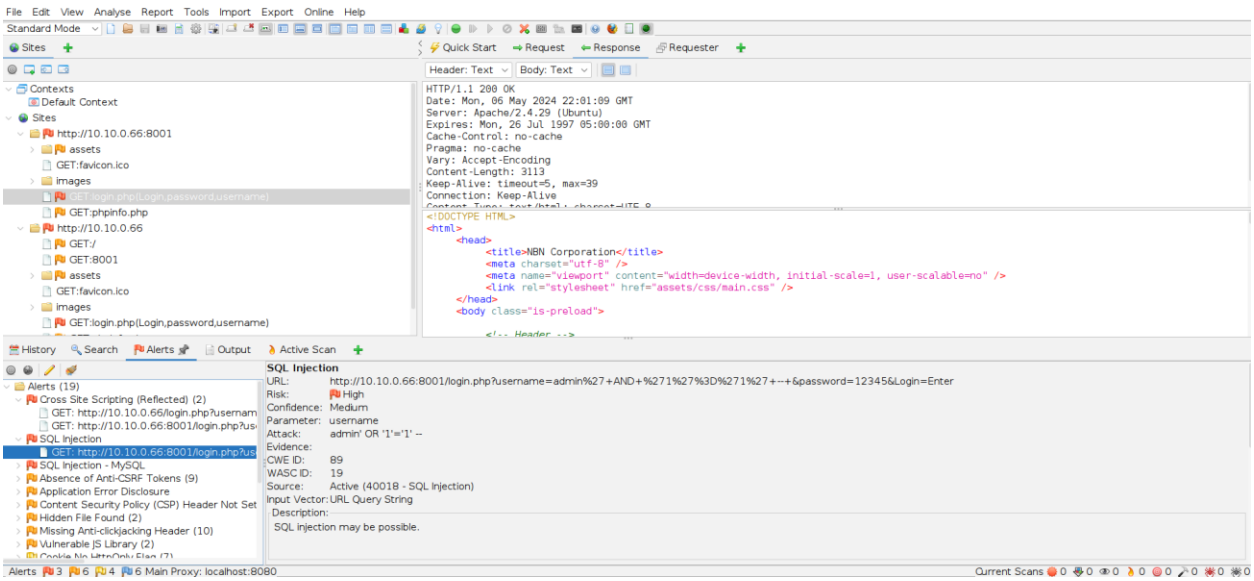
The rationale behind this vulnerability’s score is that it is somewhat more difficult to detect and execute this attack. However, it would allow the attacker to potentially get the client’s credentials or make them inadvertently download a virus on their computer. The fact that this attack is somewhat more difficult lowered its CVSS Score, but the impact still seems significant, which is why the score did not drop below 7.

   III. How to fix

Sanitize the username input once it is received as strictly as possible. Make sure that any special characters that make it likely that the input is an XSS attack are removed or neutralized. Also, use CSP as further defense against XSS.

5. In staging, the username parameter is vulnerable to SQLi attacks (CVSS Score: 5)
   I. How this was discovered and could be exploited

When I got to the login portion of the website, I ran zaproxy. Then, I submitted some input on the website and used zaproxy to fuzz and see if any input indicated a successful attack:



As you can see, if “admin’ OR 1=1 – ” is entered into the username field, then we are able to execute a successful SQLi attack. Although this is not useful in this case, since we do not need a password to log in to the staging server, it has the potential to become a problem. It is possible that an attacker could leak the server’s database through blind SQL injections. For example, consider this input: "admin' UNION SELECT IF(SUBSTR((SELECT SCHEMA_NAME FROM information_schema.SCHEMATA LIMIT 1 OFFSET {x}), {y}, 1) = '{z}' , SLEEP(5), 0), 2, 3; -- ". If an attacker can automate sending input like these to the server and iterates over all ascii characters ({z}) that could be at the y’th position of the x’th database name, then the attacker could slowly leak the database names. After that, he could similarly leak table names, column names, and then finally, what is inside the tables.

   II. Rationale behind CVSS Score

The rationale behind this vulnerability’s score is that it is a lot more difficult to execute this attack (especially if the goal is to leak the database). However, it would allow the attacker to potentially get the entire database, which may have more than just the staging server’s information in it. This increased its score. I left the CVSS Score of this vulnerability at 5 even through the CVSS Scores of theoretical vulnerabilities should be low because although this vulnerability has theoretical aspects, the SQLi is shown to work.

III. How to fix

Use prepared statements with parameterized queries. This helps mitigate SQLi attacks because the reason SQLi attacks work is because data can be mixed with the code. Prepared statements keep the data and the code separate and thus help make SQLi attacks far more difficult.

Conclusion

The goal of this test was to further secure the sensitive data that this company is in possession of. The targets of this test included all external facing hosts and services, all external facing Web Apps, and internal network vulnerabilities (if internal network access was achieved).

Here is a summary of my results and their associated risks:

Description of vulnerability	CVSS Score
FTP server allows anonymous logins	9
HTTP used for website traffic	8.9
Anyone on the network is allowed access to the staging server	7
In production, the username parameter is vulnerable to XSS attacks	6
In staging, the username parameter is vulnerable to SQLi attacks	5

Immediate fixes that should be implemented right away are disabling FTP Anonymous