Penetration Test Report

TryHackeMe

Project: 1
Version: 1.0
Date: July 30, 2023
Prepared for: TryHackMe
Prepared by: Lawson Baldwin

BUSINESS CONFIDENTIAL

Table of Contents

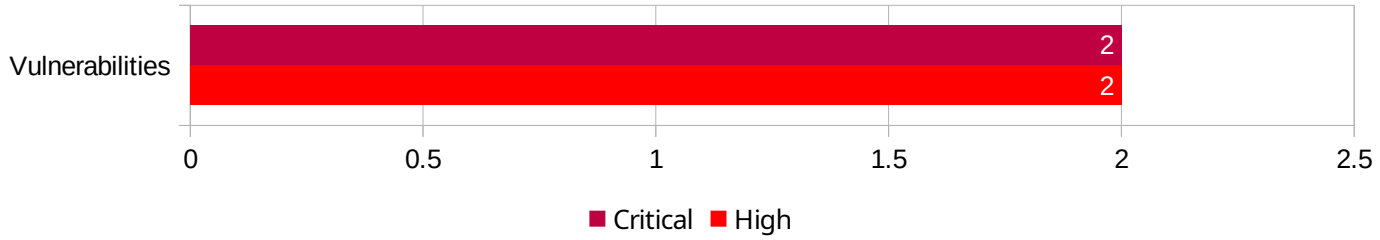
Executive Summary

Introduction

TryHackMe reached out to the penetration tester to conduct an evaluation on their Linux servers. The assessment started on July 22, 2023 and ended July 24, 2023. The tester examined the external and internal servers.

Findings Summary

Four vulnerabilities were found and two were critical. The severe vulnerabilities will cause catastrophic damage if not fixed immediately.



The tester however did encounter a positive finding that prevented files from being manipulated on the FTP server.

High-Level Recommendations

It is recommended to address the critical and high vulnerabilities first. **Appendix B – Remediation Checklist** should be used to ensure all areas are covered.

Methodology

Overview

This assessment was conducted using the Penetration Testing Execution Standard (PTES) and includes 6 phases: Pre-engagement, Reconnaissance, Vulnerability Assessment, Exploitation, Post Exploitation, and Reporting.

Scope

The scope was to obtain root access to a Linux system using the IP 10.10.246.215.

Approach

The tester used Nmap to scan for open ports which resulted in port 21, 22, and 80 being found. Port 80 did not have much to go on except that the server was hosted using Nginx. Port 21 allowed the tester to login to the FTP server using an anonymous login.

A password file was found on the server and was downloaded. It contained the root password.

The tester then logged into port 22 using SSH and the username root with the password found on the FTP server.

The SSH server showed system information in the banner which could be useful for further attacks. The tester went to the root directory and found the root flag. The account flag was then located in the /home/librarian directory.

Afterwards, the tester covered his tracks by deleting the command history.

Tools and Techniques

Below are a list of tools used and a brief description:

- Nmap was used to reveal ports 21, 22, and 80.
- TNFTP was used to login to the FTP server on port 21.
- SSH was used to login to the SSH server on port 22.
- Various commands were used to disable, delete, and enable the command history.

Technical Findings Details

Findings Overview

All of the issues during the penetration test are listed below and risk rated. Details on the risk rating can be found in **Appendix A – Finding Severities**.

Finding #	Description	Risk
1	Observable Discrepancy	CRITICAL
2	Plaintext Storage of a Password	CRITICAL
3	Lack of FTP Server Login Restrictions for Anonymous Users	HIGH
4	Improper Restriction of Administrator Login	HIGH
5	Display of Web Server Software on Website	INFO

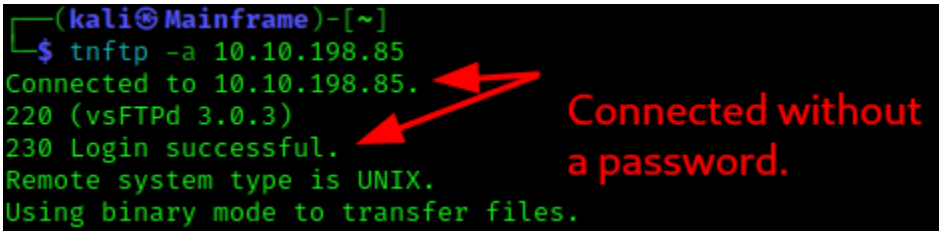
Technical Details

1. Observable Discrepancy - Critical

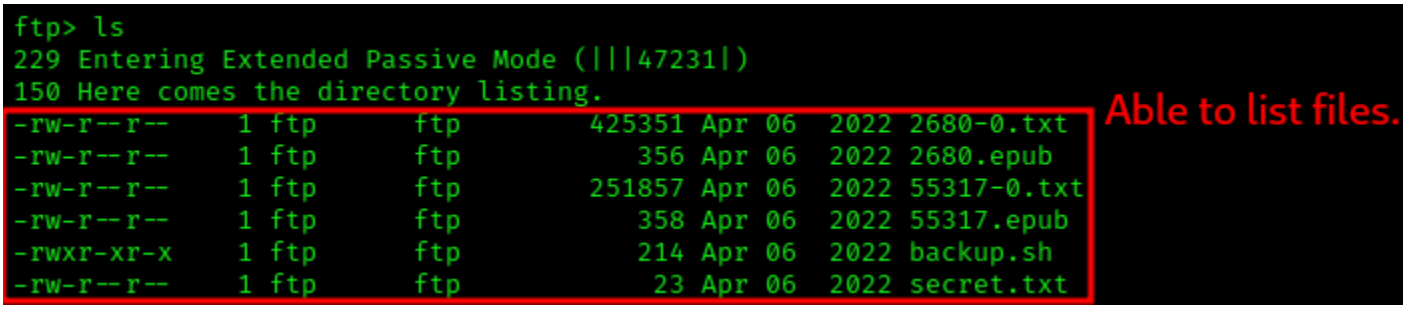
CWE	https://cwe.mitre.org/data/definitions/203.html
Description	A confidential password file was located on a public FTP server. It was stored here by an administrator on accident.
Security Impact	This is an immediate cause for concern, because someone with very little technical knowledge could retrieve the password and use it to login to root on the corresponding SSH server. As a result, the SSH server is very likely to be attacked.
Remediation	Compartmentalize safe and unsafe zones for storing information to avoid confusion. Having dedicated storage locations for the appropriate users ensures information is not sent to the wrong users.

Finding Evidence:

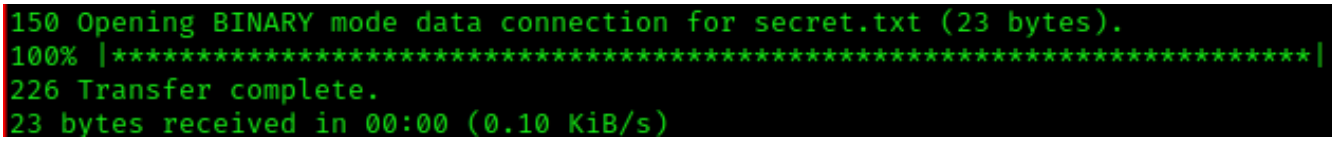
The attacker logged into the FTP server anonymously.



The attacker listed all of the files on the server.



She then downloaded the one named “secret.txt”.



The attacker then viewed the contents of the file on her computer.

2. Plaintext Storage of a Password - Critical

CWE	https://cwe.mitre.org/data/definitions/256.html
Description	A password file was found on a public FTP server and is readable in plain text.
Security Impact	This is a major security risk. Not just because the file was stored publicly, but because the password is not encrypted. Regardless of where it is stored, if someone were to retrieve the password, they could easily read it.
Remediation	Encrypt not just passwords, but any confidential files. Double check that private files are stored securely.

Finding Evidence:

See “Finding #1” for evidence. The steps are the same and portray how an attacker could easily read the password without any effort.

3. Lack of FTP Server Login Restrictions for Anonymous Users - High-Severity

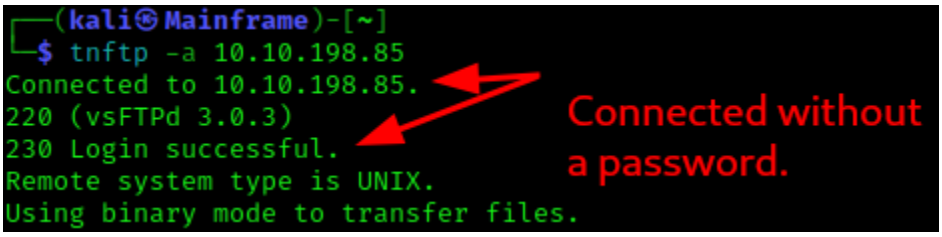
Description	The FTP server does not restrict which users or computers can access the contents of the server.
Security Impact	This is a major security risk because anyone who wants access can have access. If there are any accidental data leaks like the “secret.txt” file located on the server, attackers could use this for privilege escalation.
Remediation	Require users to create a user account. This ensures the perpetrator cannot easily hide themselves. Require users to create a password. This prevents unauthorized users from using known usernames or brute-forcing unknown ones.

Finding Evidence:

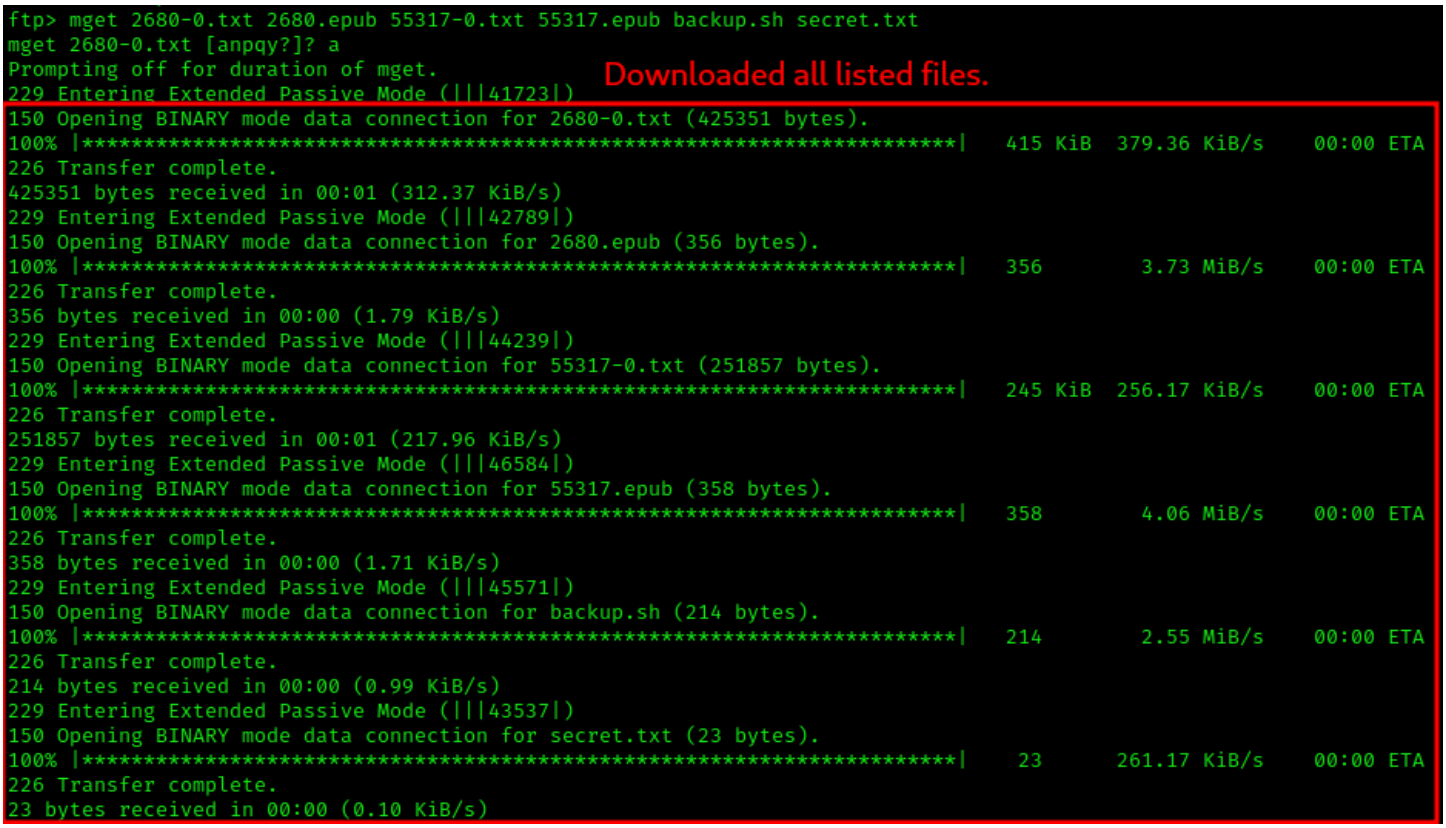
An attacker discovered the IP address of the server and used Nmap to scan for open ports.

```
┌──(kali㉿Mainframe)-[~]
└─$ nmap 10.10.246.215
Starting Nmap 7.94 ( https://nmap.org ) at 2023-07-22 15:19 CDT
Nmap scan report for 10.10.246.215
Host is up (0.13s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 5.22 seconds
```

The attacker used port 21 because it was a simple way to extract data. She also logged in anonymously to avoid the need for a username or password.



The attacker then downloaded all the files on the server.

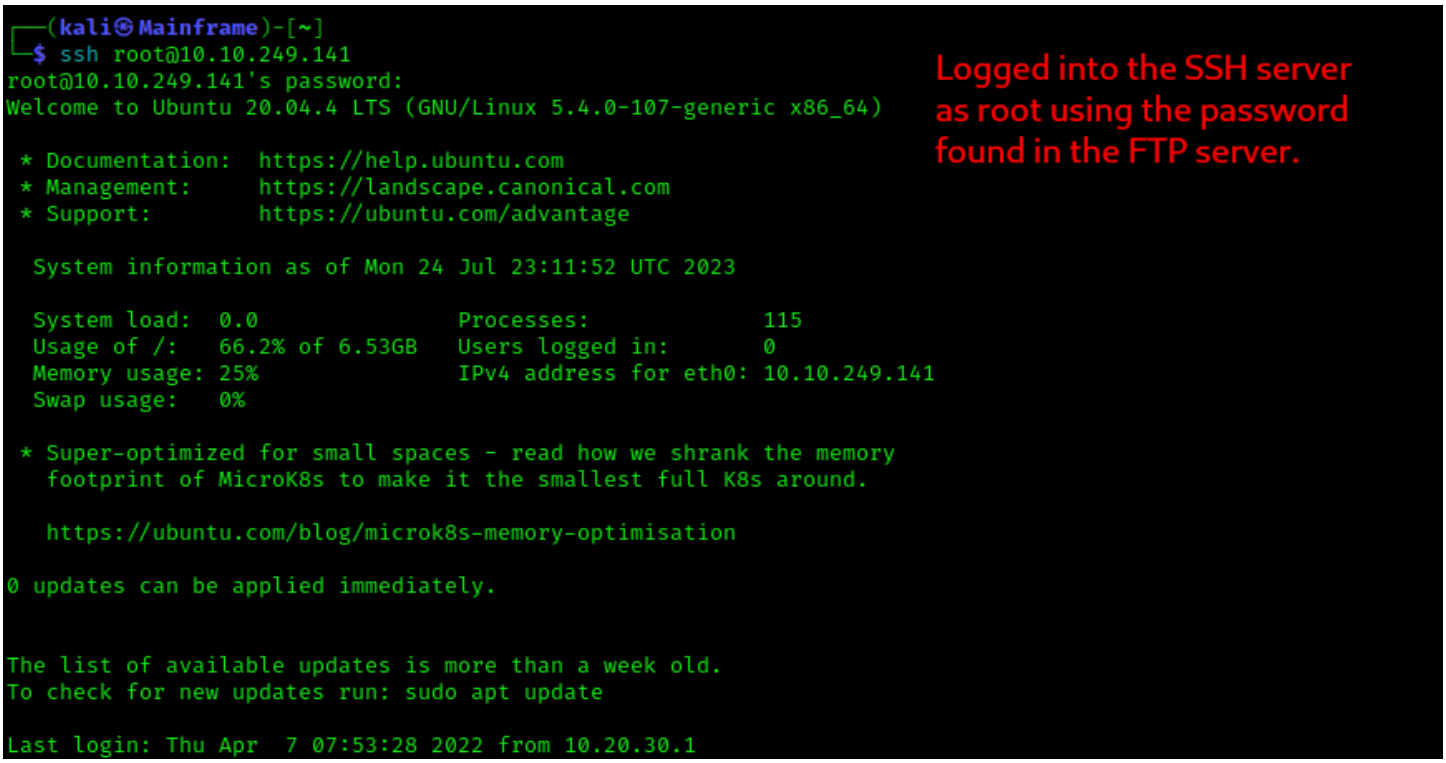



```
ftp> mget 2680-0.txt 2680.epub 55317-0.txt 55317.epub backup.sh secret.txt
mget 2680-0.txt [anpqy?]? a
Prompting off for duration of mget.
229 Entering Extended Passive Mode (|||41723|)
150 Opening BINARY mode data connection for 2680-0.txt (425351 bytes).
100% |*****************************************************************|   415 KiB  379.36 KiB/s    00:00 ETA
226 Transfer complete.
425351 bytes received in 00:01 (312.37 KiB/s)
229 Entering Extended Passive Mode (|||42789|)
150 Opening BINARY mode data connection for 2680.epub (356 bytes).
100% |*****************************************************************|   356        3.73 MiB/s    00:00 ETA
226 Transfer complete.
356 bytes received in 00:00 (1.79 KiB/s)
229 Entering Extended Passive Mode (|||44239|)
150 Opening BINARY mode data connection for 55317-0.txt (251857 bytes).
100% |*****************************************************************|   245 KiB  256.17 KiB/s    00:00 ETA
226 Transfer complete.
251857 bytes received in 00:01 (217.96 KiB/s)
229 Entering Extended Passive Mode (|||46584|)
150 Opening BINARY mode data connection for 55317.epub (358 bytes).
100% |*****************************************************************|   358        4.06 MiB/s    00:00 ETA
226 Transfer complete.
358 bytes received in 00:00 (1.71 KiB/s)
229 Entering Extended Passive Mode (|||45571|)
150 Opening BINARY mode data connection for backup.sh (214 bytes).
100% |*****************************************************************|   214        2.55 MiB/s    00:00 ETA
226 Transfer complete.
214 bytes received in 00:00 (0.99 KiB/s)
229 Entering Extended Passive Mode (|||43537|)
150 Opening BINARY mode data connection for secret.txt (23 bytes).
100% |*****************************************************************|    23      261.17 KiB/s    00:00 ETA
226 Transfer complete.
23 bytes received in 00:00 (0.10 KiB/s)
```

The attacker used “ls” to ensure the files were downloaded on her computer and opened the file “secret.txt” to view the root password for the SSH server on port 22.

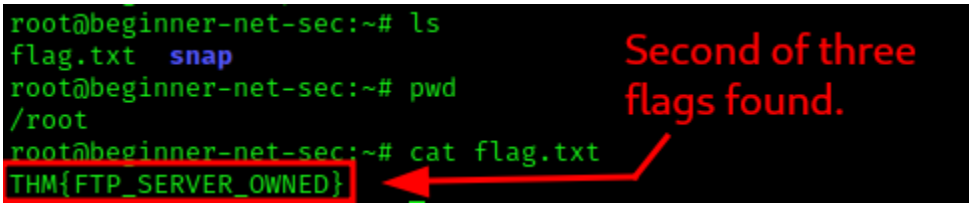

```
┌──(kali㉿Mainframe)-[~]
└─$ ls
2680-0.txt   55317-0.txt  backup.sh  Documents  MALICIOUS-FILE.TXT.EXE  Pictures  secret.txt  Templates
2680.epub    55317.epub   Desktop    Downloads  Music                   Public    simple      Videos

┌──(kali㉿Mainframe)-[~]
└─$ cat secret.txt
password: ABC789xyz123
```

The attacker assumed the username “root” and used the given password to login and privilege escalate.



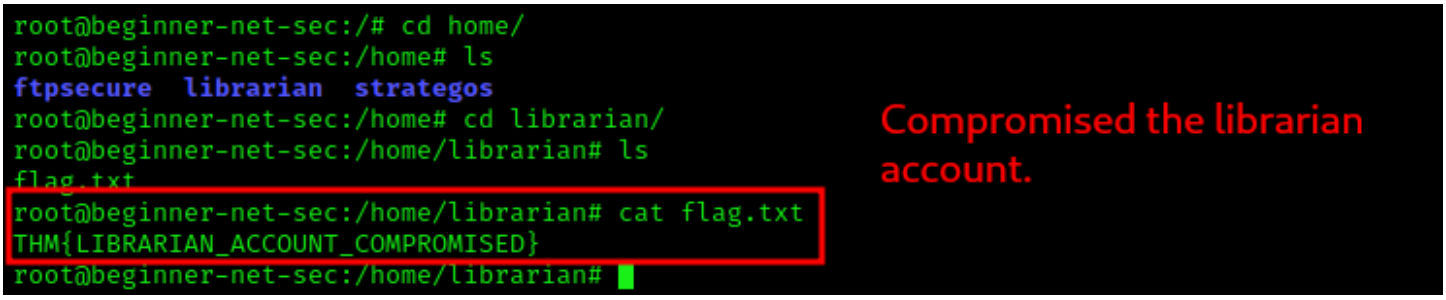

```
┌──(kali㉿Mainframe)-[~]
└─$ ssh root@10.10.249.141
root@10.10.249.141's password:
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-107-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon 24 Jul 23:11:52 UTC 2023

  System load:  0.0               Processes:             115
  Usage of /:   66.2% of 6.53GB   Users logged in:       0
  Memory usage: 25%               IPv4 address for eth0: 10.10.249.141
  Swap usage:   0%

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

0 updates can be applied immediately.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Thu Apr  7 07:53:28 2022 from 10.20.30.1
```

The attacker stole sensitive information on the root directory.




```
root@beginner-net-sec:~# ls
flag.txt  snap
root@beginner-net-sec:~# pwd
/root
root@beginner-net-sec:~# cat flag.txt
THM{FTP_SERVER_OWNED}
```

The attacker fully compromised the server by retrieving account information from the “/home/librarian” directory.




```
root@beginner-net-sec:/# cd home/
root@beginner-net-sec:/home# ls
ftpsecure  librarian  strategos
root@beginner-net-sec:/home# cd librarian/
root@beginner-net-sec:/home/librarian# ls
flag.txt
root@beginner-net-sec:/home/librarian# cat flag.txt
THM{LIBRARIAN_ACCOUNT_COMPROMISED}
root@beginner-net-sec:/home/librarian#
```

4. Improper Restriction of Administrator Login - High-Severity

Description	Any computer providing the correct credentials is able to login to the root

	account of the SSH server
Security Impact	Not limiting who can login to root allows an attacker easy access to the account once they retrieve the correct credentials. This makes it likely that the account will be compromised because the account will be susceptible to brute-force attacks.
Remediation	Require more than just a password and username to prove a user has authorization. Block unknown IP addresses from logging into root. Disable access to the root account when not in use. Have alerts go off if suspicious activity is occurring on the root account.

Finding Evidence:

See “Finding #3” for evidence. The process is the same and shows how an attacker can retrieve the correct credentials to make an unauthorized login.

5. Display of Web Server Software on Website - Informational

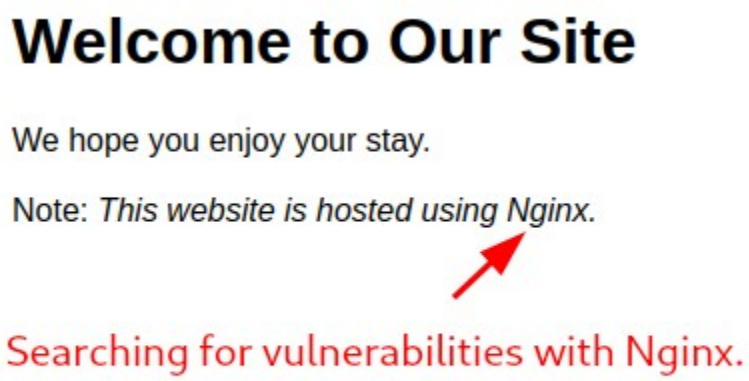
Description	The home page of the website shows that it is hosted using Nginx.
Security Impact	While this does not pose an immediate threat, an attacker can use this information to find potential known vulnerabilities for the software. A vulnerability may pop up in the near future.
Remediation	It is advised to hide unnecessary information to mitigate this risk entirely.

Finding Evidence:

The attacker scanned the IP 10.10.246.215 to search for open ports.

```
┌──(kali㉿Mainframe)-[~]
└─$ nmap 10.10.246.215
Starting Nmap 7.94 ( https://nmap.org ) at 2023-07-22 15:19 CDT
Nmap scan report for 10.10.246.215
Host is up (0.13s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 5.22 seconds
```

The attacker saw that port 80 was open and typed in the IP address into her computer's web browser.



See saw that the website was using Nginx as its host and began doing an open source intelligence gathering to find its vulnerabilities.

Positive Findings

Below is a list of security strengths that were found during testing.

1. Prevention of File Manipulation for Anonymous Users in FTP Server

The tester was unable to delete, move, add, or rename files on the server. This is great, as it prevents unauthorized users from directly manipulating the contents of the server.

Appendices

Appendix A – Finding Severities

Each finding has been assigned a rating of critical, high, medium, low, or informational. The ratings were made to prioritize which findings should be addressed first.

Rating	Definition
CRITICAL	This vulnerability is well known and can be easily exploited. Exploitation likely results in a breach of information, unauthorized system modifications, and a disruption of services. It is likely that multiple or all systems on the network will be affected on the root-level (e.g., Remote Code Execution, Privilege Escalation, SQL Injection).
HIGH	This vulnerability while less direct, is still widely known by attackers. Exploitation likely results in a breach of information and disruption of services. While root-level compromise is not likely, it can still significantly damage to the network (e.g. Remote Code Execution, Privilege Escalation, Cross-Site Scripting).
MEDIUM	This vulnerability could potentially be exploited by attackers, but may require more effort. Exploitation likely results in minor data leakage, limited unauthorized access, or partial system disruption (e.g. Cross-Site Request Forgery, Information Leakage, Insecure Direct Object References).
LOW	This vulnerability is not well known and may require other vulnerabilities to be successfully exploited. Exploitation may grant minimal unauthorized access, minor information leakage, or negligible system disruption (e.g. Weak Password Policy, Lack of Secure Transport Encryption, Lack of Input Sanitation).
INFORMATIONAL	This vulnerability does not directly allow for exploitation. This is applicable if an attacker discovers information about the network or system that could potentially be useful (e.g. Information Disclosure through HTTP Headers, Unnecessary Open Ports, Disclosed Software or Version Information).

Appendix B – Remediation Checklist

1. Observable Discrepancy - Critical
   - ☐ Compartmentalize safe and unsafe zones for storing information to avoid confusion.
2. Plaintext Storage of a Password - Critical
   - ☐ Encrypt any confidential files.
   - ☐ Double check that private files are stored securely.
3. Lack of FTP Server Login Restrictions for Anonymous Users - High-Severity
   - ☐ Require users to create a user account and password
4. Improper Restriction of Administrator Login - High-Severity
   - ☐ Require more than a password and username to prove authorization.
   - ☐ Block unknown IP addresses from logging into root.
   - ☐ Disable access to the root account when not in use.
   - ☐ Have alerts go off if suspicious activity is occurring on the root account.
5. Display of Web Server Software on Website - Informational
   - ☐ Hide unnecessary information.

Appendix C – Exploited Hosts

Host	Method	Notes
10.10.198.85	Anonymous login through FTP.	Second IP issued (given).
10.10.249.141	Root login with default username.	Third IP issued (given).

Appendix D – Compromised Users

Username	Method	Notes
root	Default username and discovered password login.	Password file on FTP server.