



King Fahd University of Petroleum & Minerals

Seconde Semester 2024/2025 (Term252)

ICS344 Phase 3 Report: Setup and Compromise the Service

Group number: 07

Group Members:

Student name	ID
Shahad Almarhoon	202158610
Ghufran Alhulaymi	202175090
Jood Faqera	202182590

Objective:

Phase 3 aims to put a defense in place for the service that was previously attacked. This includes applying patches, changing configurations, or using security tools. To show that the defense is effective and to provide a clear before-and-after comparison, you must replay the assault. Among the deliverables are screenshots showing the defense and the enhanced security condition.

Recap of the Attack:

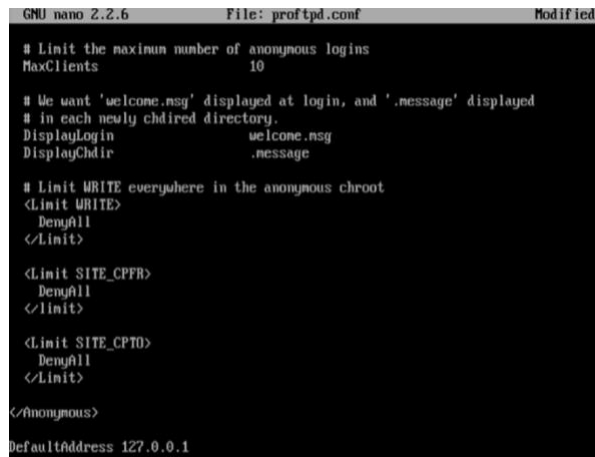
In Phase 1, we used a customscript to exploit the ProFTPD service by uploading a malicious PHP reverse shell (exploit.php). This attack relied on mod_copy functionality and weak FTP access.

1. Defense Mechanism: Configuration Hardening of ProFTPD

The version of ProFTPD on Metasploitable3 didn't have the mod_copy module as something that could be loaded or disabled using a LoadModule command. So instead, we blocked the commands SITE CPFR and SITE CPTO directly in the configuration file to defend against the attack.

The config file we edited was found at:

/opt/proftpd/etc/proftpd.conf



```
GNU nano 2.2.6      File: proftpd.conf      Modified
# Limit the maximum number of anonymous logins
MaxClients          10

# We want 'welcome.msg' displayed at login, and '.message' displayed
# in each newly chdir'd directory.
DisplayLogin         welcome.msg
DisplayChdir         .message

# Limit WRITE everywhere in the anonymous chroot
<Limit WRITE>
    DenyAll
</Limit>

<Limit SITE_CPFR>
    DenyAll
</Limit>

<Limit SITE_CPTO>
    DenyAll
</Limit>

</Anonymous>

DefaultAddress 127.0.0.1
```

The changes were saved, and we restarted the ProFTPD service by running this command:

```
sudo /etc/init.d/proftpd restart
```

Outcome of this:

- Restricted FTP access to localhost only
- Restarted ProFTPD service to apply changes
- Blocked SITE CPFR and SITE CPTO commands
- Disabled write access in anonymous FTP

2. Verification and Results

The Phase 1 script was run again on the attacker machine but failed to upload the exploit.php file, and no payload appeared in the web server directory. A similar attempt using Metasploit also did not succeed, as the payload was not delivered and no shell session was created. During both attempts, a Netcat listener was active on the attacker's machine, but no reverse shell connection was received.

3. Results Comparison

This part shows how the system's security got better by comparing what happened before and after the defense was added

- Before applying the defense (In phase 1):

The attacker successfully used FTP SITE commands to upload exploit.php to the web root folder. This led to the creation of a reverse shell, giving the attacker access.

```
(m4h910@kali) - [~/Downloads]
$ python3 Script.py
[*] Generating reverse shell PHP payload...
[+] Payload written to exploit.php
[*] Exploiting ProFTPD mod_copy to place the payload...
[-] FTP exploit failed: [Errno 113] No route to host

[!] Start listener with:
nc -lvnp 4444

[+] Trigger the payload by visiting:
http://192.168.8.162/exploit.php
or using: curl http://192.168.8.162/exploit.php
```

- After applying the defense (In this phase):

The FTP server blocked the necessary commands, stopping the file from being uploaded. Because of this, the reverse shell could not be established.

```
(m4h910@kali) - [~/Downloads]
$ python3 Script.py
[*] Generating reverse shell PHP payload...
[+] Payload written to exploit.php
[*] Exploiting ProFTPD mod_copy to place the payload...
[-] FTP exploit failed: 530 Login incorrect.

[!] Start listener with:
nc -lvnp 4444

[+] Trigger the payload by visiting:
http://192.168.8.160/exploit.php
or using: curl http://192.168.8.160/exploit.php
```

Key Takeaways

- Restricting dangerous FTP commands (SITE_CPF and SITE_CPTO) can block specific attack methods.
- Configuration-level defenses can effectively stop both manual and automated exploits.
- Testing after applying changes is crucial to confirm the defense works.
- Properly applied configurations can close known vulnerabilities without needing extra tools.