Exercises on Penetration Testing

Solve the following exercises and submit them until the communicated date.

Reminder: All VMs must be running at the same time and your host machine **must** be disconnected from the Internet at any time during the subsequent exercises unless explicitly noted otherwise. For details, refer to the respective VM tutorials and the instructions on the previous exercise sheets.

LB-PT 01.

a) Based on your observations from the reconnaissance performed in LB-R 02. c), identify a shortlist of targets for more detailed penetration testing. Consider your results from LB-MA 01. b).

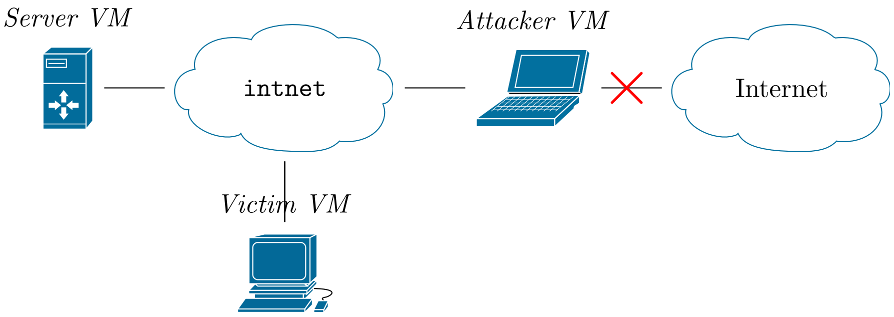
b) In the *Attacker VM*, use *telnet* via the command `telnet mail.lab-unterweger.it-security smtp` to connect to the local SMTP server. Assess whether it is possible to send an e-mail as the *maintenance* user without credentials, i.e., try to send an e-mail from `maintenance@lab-unterweger.it-security` to your own e-mail account, `stud@lab-unterweger.it-security`.
The (E)SMTP protocol is text based. To send an e-mail, input the following command sequence[1] and hit return after each command:

```
EHLO dummy
MAIL FROM:$sender
RCPT TO:$recipient
DATA
From:$sender
To:$recipient
Subject: $subject

$message
.
QUIT
```

Replace `$sender` with the sender's e-mail address, `$recipient` with the recipient's e-mail address, `$subject` with the subject of the e-mail and `$message` with the message body, which may span multiple lines. Note the final period at the end of the message. Each (upper case) command must be confirmed by the server. On errors, you must restart the sequence.

c) Use the command `ssh ssh.lab-unterweger.it-security` to initiate a remote session on the e-mail server and access your e-mails with the `mail` command (see LB-MA 00. b)). Verify that you received the e-mail from b) and that the sender appears authentic. Use the `exit` command to terminate the session.



LB-PT 02.

a) Use *Metasploit* and its module `auxiliary/server/browser_autopwn2` to set up a malicious Web server which automatically detects a visitor's Web browser and system version and attempts to exploit any known vulnerabilities. Set the module option `MaxExploitCount` to 100 and start the module. After some time, the module outputs a link which can be sent to a victim, e.g., via e-mail from a trustworthy, but spoofed sender (see exercise 01.). Send such an e-mail to `victoria.victim@lab-unterweger.it-security` to initiate the aforementioned attack as illustrated in the figure above. As the user *Victoria Victim* in the *Victim VM*, start *Mozilla Thunderbird*, open the e-mail and click the link. Discuss the output of the *Metasploit* module after that. A tutorial on how to use *Metasploit* is available in the How to use *Metasploit* Section.
Usage note: Background tasks in Metasploit*, e.g., those from the Web server started in this task, print into the command prompt. It is recommended to wait for the printing to complete before hitting the return key and inputting commands again. If part of a command has already been input before printing started, it is recommended to press `Alt+Backspace` multiple times first to avoid garbling and/or duplication of commands.*

b) Abort the exploit attempt in *Metasploit* by terminating all sessions and additionally stopping all background tasks via `jobs -K`. Restart the attack, but use an e-mail text which makes it more likely for a potential victim to click the link. Similarly, set a URI and Web site content (`HTMLContent` parameter) so that a victim is more likely to stay on the Web site longer. Explain why this is practically relevant for an attack.

c) Resume your attack from a) with the new settings from b) by accessing one of the opened *meterpreter* sessions automatically. In this session, use the `screenshot` command to take a screenshot of the victim's machine and view it in the *Attacker VM*. Discuss the security implications of taking screenshots in such a way. Furthermore, use and document at least one more attack method which is available via the commands in the established `meterpreter` session. The `help` command lists all available commands.

LB-PT 03.

a) Based on the information collected about the target machine and operating system, search for a potentially matching exploit (module) in the `exploit/windows/local/` category of *Metasploit* (**except** for `exploit/windows/local/cve_2017_8464_lnk_lpe`). The exploit must be capable of further escalating your privileges, i.e., to give you *Administrator* or even *SYSTEM* privileges. Explain and document your selection criteria and start the exploit by using the *meterpreter* session from exercise 02. c). Explain whether or not the exploit was successful and why. *Note: Before starting with this exercise, it is necessary to stop all background tasks (see exercise 02. b)).*

b) Use the `exploit/windows/local/cve_2017_8464_lnk_lpe` exploit (module) and set its `TARGET` parameter to 1 and its `LHOST` parameter to your local IP address, if it is not yet set. Explain why the attack is unsuccessful initially and how setting the `PAYLOAD` parameter to `windows/meterpreter/reverse_winhttps` (which sends the exploit code via an HTTPS connection instead of a plaintext connection) makes a difference. Furthermore, explain why this attack is successful in the first place, i.e., which software component was vulnerable and could be exploited.

c) Based on the results and conclusions of the exercises 01. to 03. b), provide a list of recommendations for the maintainer of the attacked machines. The recommendations must cover both, short-term and long-term actions which are necessary to mitigate the attacks from exercises 01. to 03. b).

LB-PT 04. (bonus exercise – voluntary)

a) Use the `post/windows/gather/smart_hashdump` module to collect and export user names, their password hashes and further related information by using the *SYSTEM*-level *meterpreter* session from exercise 03. b). Convert the password hash file (its path is printed after the module has been started successfully) into a format supported by *hashcat* with the command (without line breaks or dollar signs!) `cat "$hashfile" | cut -d: -f4 > "$convertedfile"`. Use *hashcat* on the converted file in an attempt to recover the passwords of all users. Choose an appropriate attack mode and consider all information available to you. Discuss the implications of a successful password recovery on the attacked machine in general and your actual recovery results in particular.

b) Amend the list of recommendations from exercise 03. b) based on your results and conclusions from a).

How to use *Metasploit*

This section described how to use *Metasploit*. The How to start *Metasploit* Section describes how to start *Metasploit*. The How to use *Metasploit* modules and How to manage *Metasploit* sessions Sections explain two selected core concepts of *Metasploit*.

How to start *Metasploit*

Before the very first start of *Metasploit*, its database needs to be initialized **once** with the command `msfdb init`. Any prompts may be answered by hitting the return key to use sensible defaults.
To start the console version of the *Metasploit Framework* (MSF for short) with *root* privileges, use the command `sudo msfconsole`. From within the console, *Metasploit*'s functionality can accessed via commands. The commands which are relevant in the context of this laboratory are explained in the subsequent sections. The `exit` command can be used to terminate *Metasploit*.

How to use *Metasploit* modules

Metasploit consists of a collection of categorized modules which can be used via the `use` command, followed a space and the module name. The `back` command can be used to exit a module.
In order to search for modules before using one, the `search` command, followed by a space and a search term, can be used. The search result is a list of modules. For example, the command `search mozilla` yields a list of modules related to software from *Mozilla*.
In addition to searching, *Metasploit* supports auto completion via the tab key. For example, entering `use exploit/windows/browser/mozilla_` (without hitting return) and pressing the tab key (it needs to be pressed twice when there are multiple possibilities) provides a list of proposals for completion. If only one proposal remains, it is completed automatically, which significantly reduces the required input for typing out module names.
If a module is in use, e.g., after entering the command `use exploit/windows/browser/mozilla_firefox_onreadystatechange`, the commands `info` and `options` can be used to retrieve a short module description and a list of module parameters, respectively. All required parameters (in the column of the same name) must be set, i.e., they must not be empty (in the *Current setting* column). Setting parameters is done via the `set $name $value` command, where `$name` has to be replaced by the parameter name and `$value` by the desired value, e.g., `set SRVHOST 192.168.0.101`.
After setting all required and desired parameters, the used module can be started via the `run` command. Many modules open sessions when they are started (successfully). Sessions are explained in the next section.

How to manage *Metasploit* sessions

Metasploit can manage multiple active sessions via the `sessions` command, followed by a space, a command flag and an optional parameter. The flag `-l` (in the full command `sessions -l`) lists all active sessions and their properties. The flag `-K` terminates all active sessions. The flag `-i`, followed by a session number (which can be seen in the output from `sessions -l`) starts an interactive command prompt for a session. Note that some exploits (exploit modules) automatically start such a command prompt when they create a new session.

Inside a session, commands can be entered and executed. The `exit` command can be used to terminate both, the session and the command prompt, while the `background` command can be used to terminate only the command prompt, but not the session. Note, however, that the `background` command and many others are only available in special, so-called *meterpreter* sessions which are the only type of session that is used in this laboratory.

Modules often require an active session with defined privileges before they can be started. For this purpose, such modules usually provide a `SESSION` parameter which is to be set to the session number (see above). Privilege levels can typically be derived from the user name printed in the output of `sessions -l` or from the output of the `getuid` command inside the *meterpreter* session.

Notes

[1] The `EHLO` command identifies the connecting machine's domain name (which does not necessarily need to be its actual domain name) and requests the server to use the ESMTP protocol (as opposed to the `HELO` command which requests classical SMTP). The `MAIL` command starts composing a new e-mail with the sender specified after `FROM:`. The `RCPT` command adds the recipient specified after `TO:` to the recipient list of the composed e-mail. The `DATA` command expects the content of the e-mail. The *From:* and *To:* directives are for display purposes only, while the *Subject:* directive is used for specifying the e-mail subject. The message itself may contain multiple lines and is terminated by a line which contains only a period (`.`), followed by a line break. The `QUIT` command terminates the (E)SMTP session.