Lab Session 2 (L2)

Secure Deployment of Virtualized Environments

Part 2 – Deploying Secure VMs (continuation)

Objectives

Since part 2 is continuation of part 1, the objectives of the L2 remain invariable:

- To practice the configuration and deployment of a secure virtual machines (VM), applying measures explained in guidelines and recommendations available in the reference documentation links.
- To understand the weaknesses and vulnerabilities that each applied measure mitigate/cancel.
- To play with a penetration testing (pentesting) engine in order to test the applied measures and evaluate the security of the deployed VM.
- In addition, practice with the implementation of measures that create intended vulnerabilities in the VM, which is can be exploited by the pentesting engine.

Statement

Preliminaries

For doing this lab, it is required to have ready and working a minimum configuration of the two VMs deployed during part 1 of this lab L2. This minimum setup will be achieved by completing the following points of previous part 1:

- Task 0: Creating an initial VM
- Task 2: Adding a Vulnerable Service
- Task 3: Exploiting Vulnerabilities
    - 5 steps for configuring Kali VM

Important: In this lab, we are not going to evaluate the quality of your proposed solutions. As mentioned before, we are going to evaluate your effort and learning process. For this reason, try to do your best to complete **Q3, Q4, Q5, and Q6 of part 1** with your proposals before the beginning of part 2. To this aim, fill the deliverable **L2_D1** with your answers.

Task 4: Guided vulnerability exploitation (2p)

In this task, the lecturer will ask you the solutions you proposed for questions **Q3, Q4, Q5, and Q6 of part 1.** Depending on your progress, proposed solutions, and rationale of the choices, the lecturer will guide you with some additional hints and leave you additional time (if needed) to complete the questions. During the session and according to the best timing, the lecturer will propose a solution for each of the aforementioned questions.

[Q10] Write a critical analysis of your progress after lecturer's guided vulnerability exploitation. For each of the abovementioned Q3, Q4, Q5, and Q6 of part 1, write:

- **In case you could not find a solution: identify the reasons and what you learnt after lecturer exposition. If you were blocked at some stage, explain the blockage and how this session allowed you to unblock your issue.**
- **In case you proposed a different solution: identify the similarities/differences w.r.t. the lecturer's proposed solution.**

Take the opportunity of this guided session to ask your doubts about the rest of questions of part 1 that are not included in this exercise, i.e. Q7, Q8, and Q9.

Task 5: SQL injection vulnerability (8p)

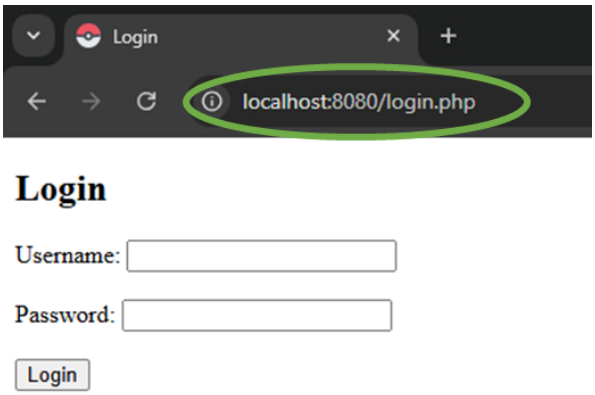
At this stage, we want to add a new component to our application stack that requires authentication. Therefore, users must login via the "/login.php" endpoint, using a basic form. Given your recently proven cybersecurity skills, a colleague asked you to test their implementation to ensure it is flawless. He told us he is using a **MySQL** database to store the users.

First, install your colleague's component by running the following command (as root):

```
#> curl -s https://raw.githubusercontent.com/TheMatrix97/Insecure-WebStack/refs/tags/2.1.0/script_part2.sh | bash
```

[Hint] You should be able to access the web service from your host computer via http://localhost:8080 (if the port-forwarding rules are set) and from the Kali VM via http://10.0.2.10

Browse to the `/login.php` endpoint, where you should see the following form:

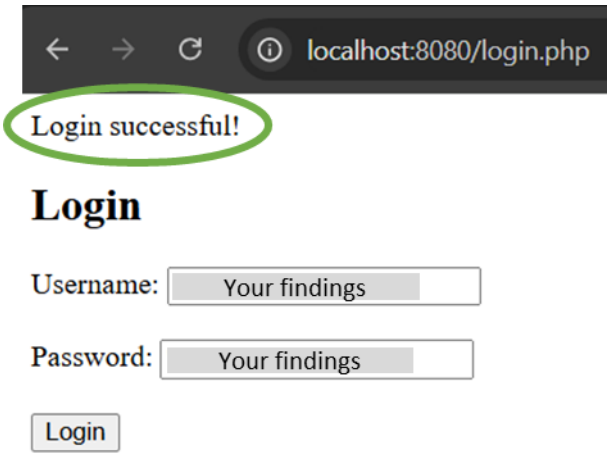


[Q11] Is the form vulnerable to SQL injection? Provide a payload example to prove it.

[Q12] Did you find a way to bypass the login with a well-known username?

[Hint] https://book.hacktricks.wiki/en/pentesting-web/sql-injection/index.html#entry-point-detection

If you find the way to bypass the authentication, you will be able to log in successfully:



[Q13] Which type of SQLi are we facing? Could we steal information using time-based exploitation? Explain what that means, and provide a payload to prove the service is vulnerable to time-based SQLi

[Hint] https://book.hacktricks.wiki/en/pentesting-web/sql-injection/index.html?highlight=sqli#confirming-with-timing

[Hint+]: https://book.hacktricks.wiki/en/pentesting-web/sql-injection/index.html?highlight=sqli#exploiting-time-based-sqli

[Q14] We would like to exploit the vulnerability and dump the table where users are stored. Are you able to obtain all usernames and passwords stored in the system? Explain the procedure you followed, the tools you used and the injection method behind the exploit.

[Hint] https://book.hacktricks.wiki/en/pentesting-web/sql-injection/sqlmap.html#post-request-injection

Delivery

Don't forget to save your progress on the L2 Delivery. More information about the delivery will be provided in the next sessions.

Please recall at this moment L2 report should contain the answers to questions:

- Q1 to Q9 from Part 1
- Q10 to Q14 from Part 2

Supporting Material

- https://book.hacktricks.wiki/en/pentesting-web/sql-injection/index.html
- https://owasp.org/www-community/attacks/SQL_Injection
- https://github.com/sqlmapproject/sqlmap