SECURITY FIRST PRINCIPLES

LEAST PRIVILEGE

LAB: PRIVILEGE ESCALATION

Lab Description: In this lab, you will investigate, exploit, and correct many common vulnerabilities that may result from not adhering to the security first principle of least privilege. This lab will span both Windows and Linux privilege escalation techniques.

Lab Environment: The instructor has shared the "csc840-securityfirst" vApp with you in DSU's IALab. Clone the vApp to your account before starting it or making any changes. Do not modify or overwrite the snapshot for the base vApp.

Box Name	Type	Lab Credentials
Linux Exercises	Kali 2023.3	student / Password1!
Windows Exercises	Windows 10 Pro build 19044	student / Password1!

You are free to install whatever tools you would like to complete the assignment. Several potentially helpful tools are pre-installed on the Linux Exercises VM if needed:

- gcc/make
- Metasploit
- SCP server

LAB EXERCISE 1: WINDOWS HORIZONTAL/VERTICAL PRIVESC

Windows systems contain many services, which are programs that run at startup and in the background during routine use. These services exist in the file system as executables, with the Services MMC snap-in controlling how they run. Services can run with user-level or system-level privileges,

which means it is critical to control access to both the service configuration itself and the service's entire execution path to ensure users do not have access they do not need.

There is a common service permissions misconfiguration in this exercise, which creates a vulnerability. It is up to you to determine the issue and use it to escalate privileges horizontally to become a different user.

Perform the following actions on the Windows Exercises VM:

1. Review the lecture and resources documentation for information on vulnerabilities related to configuring permissions for Windows services and associating them with execution paths.

   Hint: This vulnerability is a common poor practice related to services. It does not have a specific CVE associated with it.

2. Depending on which vulnerability you suspect to be present in this exercise, determine a method to search for services that exhibit the problem.

3. After you have identified the service in question, develop a method to exploit the aspect making it vulnerable. There are many tools pre-installed on the Linux Exercises VM to assist you with this if desired.

   Hint: You will need an exploit/payload and a listener. Restart the box to run your exploit.

4. Once your exploit is running and you have become the other user, create a file named "(your name).txt" in the "Step 4" folder on their desktop. Operations are logged, so do not edit or delete other files in the folder.

5. Now that you have access to an Administrator account, it's time to get SYSTEM. Create a new service with SYSTEM-level privileges and exploit it in the same manner.

6. Once your exploit is running and you have become SYSTEM, create a file named "(your name).txt" in the "Step 6" folder in the directory

where the SYSTEM-level shell opens. Operations are logged, so do not edit or delete other files in the folder.

LAB EXERCISE 2: LINUX HORIZONTAL PRIVESC

Linux commands and programs typically run with the permissions and privileges of the user that launches them. However, there are many commands that may need to run with the permissions of a more privileged user (such as passwd). Linux incorporates several security features to make this possible. However, when configured incorrectly, programs that run with more privileges than necessary can cause a variety of security vulnerabilities.

You'll investigate one of these vulnerabilities in this exercise. Use it to escalate your privileges horizontally and become a different user!

Perform the following actions on the Linux Exercises VM:

1. Review the lecture and the manpages for vulnerabilities related to Linux file permissions and attributes. There are three primary permissions that apply to this situation. Determine which one is most relevant and could be related to the vulnerability.

2. Once you've identified the vulnerability suspected to be at play in this exercise, determine a method to search for files that leverage the problematic permission.

   Hint: Depending on your familiarity with Linux security configurations, you may need to run the command on a baseline system before you are able to identify the vulnerable file in question.

3. Now that you've identified the vulnerable file, run it! Observe the results. Pay special attention to the permissions you have before and after running the file.

4. Use your newfound access to place a text file "(your name)" in the "Step 4" folder in the Desktop of the user you now have access to. Operations are logged, so do not edit or delete other files in the folder.

LAB EXERCISE 3: LINUX VERTICAL PRIVESC

Typically, attackers will use horizontal privilege escalation for another goal: to find an account that has permission to become the administrative or root user. Can you use your access from the previous exercise to get root?

If you tried to use the `sudo` command while acting as the other user, you will have noticed that it failed with the error message:

```
Sorry, user student is not allowed to execute '<your command>' as root…
```

This is because the `sudo` binary follows the same security principles as the file you investigated in the previous exercise, only they aren't disabled this time. Darn! You'll have to find another way to get root privileges.

Fortunately, there is another configuration error related to this command that will allow you to obtain root access. Investigate this vulnerability and use it to become root!

1. Review the lecture and the manpages for vulnerabilities related to sudo configuration and who can run the `sudo` command.

   Hint: Once again, this is not a CVE. It is a configuration error, so sudo is technically working as intended, even though this configuration introduces a significant security issue.

2. Once you think you have an idea of the configuration error, verify it before attempting to run `sudo` on any further commands. As you know, running `sudo` without the proper permissions generates log entries, which are undesirable since they give defenders a means of identifying the attack. Operations are logged, so avoid this.

   Hint: Look at the flags for the `sudo` command.

3. Make a change via the configuration error to allow your user to act as the root user. Ensure that you grant only this privilege to only your user. Do not grant any further privileges to your user or any privileges to any other users.

4. Execute the command you gave your user permission to run! You should get a root terminal.

5. Use your newfound access to place a text file "(your name)" in the "Step 5" folder in the root user's home directory. Operations are logged, so do not edit or delete other files in the folder.

PUZZLER: CORRECTING PRIVESC VULNERABILITIES

This is an advanced activity for students who complete the regular lab early. It is optional.

Now that you've identified and exploited some common privilege escalation vulnerabilities, it's time to correct them. Create script(s) that can scan Windows and Linux hosts for the vulnerabilities studied in this lab, create a report that lists the vulnerabilities, and (if possible) fix the underlying configuration issues. In addition, choose one or two of your own privilege escalation abilities to implement and defend against.

WHAT TO SUBMIT

For the lab exercises, place your answers to the following questions in a PDF file along with supporting evidence (if requested). Upload the PDF to the drop box.

Lab Exercise 1:

1. What service configuration error caused this Windows vertical privilege escalation vulnerability?
2. What cmd command did you use to identify the Windows vertical privilege escalation vulnerability?
3. What aspect of the service was configured improperly? State the vulnerable value exactly.

   Hint: This value is significant to further success in this exercise. Ensure that it is specific and correct.

4. Describe your method to exploit this vulnerability and get system-level access. Provide screenshots as evidence for your explanation.
   a. How did you create your exploit? Did you use any tool(s)?
   b. What step(s) did you take to place and run the exploit?

5. What command did you use to create the new service with SYSTEM-level privileges?

Lab Exercise 2:

1. What file permission or attribute caused this Linux horizontal privilege escalation vulnerability?
2. What Bash command did you use to identify the vulnerable file?
3. Analyze the file that was configured improperly. Provide screenshots as evidence for your explanation.
    a. What is the path of the file was configured improperly?
    b. What is this file? What is its function?
    c. What are the special modifications made to this file so that it functions this way?

    Hint: You may need to compare this file to the original file on the box, read the manpages related to how the original file handles privileges, or review the original file's source code.

Lab Exercise 3:

1. What is the configuration error that makes this command vulnerable?
2. What command did you use to determine the vulnerability?
3. What change(s) did you make so that you can become the root user?

For the puzzler, upload an archive to the drop box containing your script, any supporting components (i.e., requirements.txt, Dockerfile, etc.), and a readme text or Markdown file explaining how to run it.