My HackTheBox CTF Methodology - From fresh box to root!

CTF ctf, root, methodology, hackthebox

pry0cc Leader & Offsec Engineer & Forum Daddy 2 May 28

Hey 0x00ers!

I'm so sorry that it's been such a long time since I've dropped an article here! I've been writing for my current company navisec.io 190 @ delta.navisec.io 89 and I've not had the chance to drop a good article for 0x00sec for a little while.

Today that is changing! Whoop!

In this article I'm going to discuss CTF methodology, really, this links in so closely to real life penetration methodology (if you were scoped down to an internal or to a single machine).

Specifically, we're going to be discussing boot2root CTF's, things such as HackTheBox.eu, and how I generally go about pwning a box.

The Basics - what is our objective?

Usually, the objective of these CTF's is to obtain a shell, usually unprivileged, and then escalate your privileges to gain access to root. Occasionally this doesn't necessarily mean you have to obtain a root shell, but be people to read/write files as root, (which with more time could be used to gain a root shell).

As per hackthebox, you usually have these two files known as flags stored on the machine. On Linux machines the "user.txt" flag denotes a user own, and is stored in /home/someusername/user.txt, and on Linux, the "root.txt" flag file is stored in /root/root.txt.

On windows boxes, these files are usually stored in C:\Users\Username\Desktop\user.txt, and C:\Users\Administrator\Desktop\root.txt respectively.

When you gain access to these files, you just need to view the contents (`cat`/`type`) and copy the code into the HackTheBox page for that box.

How do we even start?

Ok, so now you know what you're trying to achieve, you need to know where to start.

I recommend now if you're reading so far, and you are not familiar with basic networking, Linux or Windows basic usage, and scripting languages such as python/ruby, then you're not ready for CTF's. For networking watch this 407 and for Linux watch this 265. Obviously, this is nowhere near enough to get started but it should provide some structure for absolute newbies!

The first stage of hacking is enumeration.

Enumeration

Enumeration is the stage where we attempt to collect as much information as we can from the host before even trying to exploit anything. I will easily spend 1-2 hours of enumeration on a medium box. Take your time with this step, never rush this. Go very slow, read everything, save anything and everything in your notes that might be relevant later.

For a quick reference containing everything here, read this 107.

Port scanning

Typically in a CTF, I will begin with a portscan.

This is the real nmap scan I use for SwagShop this weekend.

```
sudo nmap -v -sS -A -Pn -T5 -p- -oN swagshop.txt 10.10.10.140
```

Just to make sure you're not missing anything a quick UDP scan, since this is so slow I generally won't wait for it and will just go ahead while it's running.

```
sudo nmap -v -sU -T5 -oN swagshop-udp.txt 10.10.10.140
```

You rarely will find UDP ports open & relevant in CTF's, but they can catch you out (for example Access).

This command will return your nmap results, very often if it's a Linux box, you'll end up with a webserver on port 80, maybe 443, and port 22 open. Bruteforcing SSH is rarely fruitful and something I will only ever do if i'm scrabbling for access (I've never bruteforced successfully on a HTB before).

Now that you've done a network map, you need to enumerate each service. So in our case:

```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol
| ssh-hostkey:
|   2048 b6:55:2b:d2:4e:8f:a3:81:72:61:37:9a:12:f6:24:ec (RSA)
|   256 2e:30:00:7a:92:f0:89:30:59:c1:77:56:ad:51:c0:ba (ECDSA)
|_  256 4c:50:d5:f2:70:c5:fd:c4:b2:f0:bc:42:20:32:64:34 (ED25519)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-favicon: Unknown favicon MD5: 88733EE53676A47FC354A61C32516E82
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-title: Home page
```

Service Enumeration

Now I will write down things such as the OpenSSH version, the distribution discovered via this fingerprint. I will also do the same for HTTP. Generally, to ensure I'm not missing anything dumb, I will search https://www.exploit-db.com/ 156 for these version numbers.

(Tip, if you get a version like OpenSSH 7.2p2, search SSH 7.2, or OpenSSH 7 and look over the results. You might find that the description on the exploit is like "< 7.3".)

Quite often, I will not find a full on RCE here, if I do spot an RCE, I'll fire up metasploit and search the software as well. Sometimes there are exploits in here (and vice versa). You're probably noticing the trend here is "just check in case". As @cry0lit3 always says, leave no stone unturned.

If you're on a Windows box, you might find that port 21 is open, port 139, 445 are open. If this is the case, try and connect to FTP (I use ncftp thanks to @guly), note down the version numbers, try and

connect with the “Anonymous” user. For 139 and 445, try and enumerate SMB.

```
smbclient -L 10.10.10.140
```

This will return shares listed in SMB. Once you’ve got a few, use smbclient to try and connect to each one.

```
smbclient //10.10.10.140/TheShare
```

This will prompt you for a password, you can also specify a username with the -U flag. If your credentials are successful, you will be dropped into an interactive prompt similar to that of FTP and you’ll be able to navigate files and download files.

You can also use the SMB script that comes with nmap,

```
nmap -p445 -sV --script smb-enum-services 10.10.10.140
```

This script works ok, but it’s best to try both.

HTTP Enumeration

A crucial part of lots of hacktheboxes and CTF’s is HTTP enumeration. I love using Burpsuite for this, setup burpsuite and proxy all your requests, if you have pro, do an active spider. Click through the entire website and click everything, every link, the file structure will be populated in the left-hand side of the Burpsuite window. I strongly recommend flipping through each request and nothing things such as the request cookies, the response headers, and scanning through the web source. You might get lucky and find some comments containing a hint!

Now that you’ve completed some level of passive reconnaissance, go in hard with the active. I will generally begin this step before I do passive enum just for the sake of efficiency.

There are many different tools you can use here, but I really like gobuster.

```
gobuster -w SecLists/Discovery/Web_Content/big.txt -u http://10.10.10.140/
```

You can also chuck in the -x flag and supply some different extensions if the app has a lot of aspx pages I will chuck on `-x aspx`. Focus this bit depending on what you're dealing with, is it a NodeJS app? Try `-x js,json`. It's unlikely you'll find a .php page on a NodeJS app.

For good measure, once I've run this and begin investigating further, I will chuck this at it:

```
gobuster -w SecLists/Discovery/Web_Content/raft-large-files.txt -u http://10
```

Once I've gotten a list of different directories that have been discovered, I will visit them, curl them, and investigate further. Usually, on an easy box, you'll find something you might be able to exploit here, or maybe just more hints!

What you find might happen is that the developer has been crafty and enabled wildcard directories. This is a common thing on harder boxes and you will often notice that this the developer's way of telling you "its not by dirbusting!". Sometimes, though, they might hide something inside this, maybe a comment or a page with a 404 not found status code, but with something inside the source, you can use burp intruder and examine the response lengths if you feel this might be the case.

Again - examine all of your traffic through Burp, its so invaluable to be able to read the source. Doing this would of gotten me a foothold on OneTwoSeven so much quicker!

Finding subdomains with SSL certs

If you find that port 443 is open, or an SSL port is open, you might be able to leverage this to get yourself some subdomains! You can either click on the cert and navigate and read all the information until you find some other valid certnames.

Or you can use this one-liner, (I recommend doing both though).

```
echo | openssl s_client -connect 0x00sec.org:443  | openssl x509 -noout -tex
```

Once you've got these, put them in your hosts file or try and request with curl:

```
curl -vv 'Host: yournewdomain.com' 10.10.10.40
```

You might get lucky! If you want to bruteforce these, you can use the `auxiliary/scanner/http/vhost_scanner` module in Metasploit.

Exploitation

Now that you've enumerated a bunch, and hopefully organized this into a nice set of notes (I usually just use a text file), I recommend (taking your time), and going over the data you've collected. Focus on trying to use your current gathered information to gain and gather more information. Perhaps you read some text on the website that suggested the username or potential name of the fictional user of the system.

If you've obtained creds, try them everywhere! And note down that you've tried them. Noting down what you've tried and being methodical is so useful.

Now after you've reviewed your notes, you will likely find that something pops into your head; an "OH OF COURSE" moment. If you've done your enumeration well enough, you should see the path that has been laid out for you pretty clearly. Whether that is a vulnerable webapp, a wordpress instance located in a sub path, a status page giving you more information, an LDAP instance that is leaking hashes (that you can crack!), an LFI that you can get ssh private keys from!

Try and think through what the app does, what the system is, and how it works. Where is it communicating? How did the developer design this? How might this be vulnerable?

Once you've found this path, try and exploit it. Depending on the difficulty of the box, you might need to spend a little while on this. Google is your best friend! If you're really struggling, you can look over some hints from the HackTheBox forums, but remember, if you're planning on doing the OSCP, you won't get the luxury of these hints. Usually, this involves getting a shell, for a quick shell reference this 289 can be very helpful.

Getting the Root!

Now really, since this is not intended to be an exhaustive article on enumeration, but more of a quick overview of CTF methodology. I won't go too deep into rooting and privesc. The methodology is exactly the same for me. Collect as much information as possible, organize your collected data and then the path should reveal itself.

If you want a privesc reference, check this 488 out!

Conclusion

In conclusion, the methodology of gaining access to a CTF box contains a few steps:

- Information gathering
- More information gathering
- Review your collected information
- Exploit the path that you've been shown by the hacker gods
- Repeat

It's dead easy when you think about it like this, and really taking your time is invaluable. If you don't see a path, just keep enumerating. You'll find something, it just might take a LOT of enumeration. In many ways, knowing the path to exploitation but just struggling on the actual exploitation execution is a really rewarding feeling, because finding that flaw to exploit is literally half the battle.

I hope this was helpful, and not too scrappy, this was thrown together in a mad blaze of creativity before my workday. If you spot any typo's let me know and I'll fix it.

What does your CTF methodology look like? Do you have a different approach or tips for others on this topic? Please let me know!

Thanks, 0x00ers!

Stay snappy 😉

2 Replies

53

created	last reply	19	27.2k	10	70	9
May 28	29d	replies	views	users	likes	links

Alice May 28

For nmap, I usually use the -sV (service scan) instead of -sS and if need be, I run a FIN scan (-sF), but that is unreliable because the nature of how the packets are sent and received. Of course there is also the X-mas and NULL scans. I think that sV is the same as sS packet wise, but whatever.

I should note that **MOST** CTF's that I have seen aren't that realistic especially when we are talking the real world threats that are actually problematic.

1 Reply

Kage May 28

That's a great resource, thanks pry! The links you've shared are as invaluable as the rest of your content. This post boosted my motivation to boot up my Linux machine and work on some HTB machines.

1 Reply 1

pry0cc Leader & Offsec Engineer & Forum Daddy Alice May 28

I really have no idea what you're talking about.

There are some really realistic boxes on HTB. Even ctfy boxes teach you a lot about enumeration.

Boxes like Querier & Access are hella realistic. But then again, I guess it's never going to be as easy as "hacking" a printer that's completely open with a web GUI 😛

With regards to realistic threats, the actual realistic threats are unpatched eternalblue and open FTP shares. It's the stupid stuff that causes stuff like Mirai, not some 1337 0day.

1 Reply

pry0cc Leader & Offsec Engineer & Forum Daddy Kage May 28

I'm glad man! I cannot wait for you to get back into HTB! It's so much fun and getting that root is the best feeling ever.

Don't put too much pressure on yourself, go slow and enjoy the ride. Get lost in the rabbit holes and don't be too hard on yourself. It's like a muscle, you gotta go easy when you've not flexed it for a while.

Alice pry0cc May 28

I disagree, but I doubt you and I both have the experience to back our claims up. Mirai for one thing infected IoT devices and last I checked, IoT devices are now being secured via a key encryption thing… Eternalblue was a joke in my opinion. Only reason why it was effective was because it was easy to re-arm. My suspicions are more toward HVAC and SCADA as the next 'disaster in the making'. The ICS devices in general are vulnerable as far as I can tell and also, controls the infrastructure of a country which is a necessity of a country to function.

I do suggest you take a couple of hours or so and read Hacker's Playbook Volume 3. There are a lot of stuff in there that I feel can better explain what I am trying to well… explain.

2 Replies

overcast May 29

I'd agree with @pry0cc on this one. While HVAC and SCADA hacking does happen, there is a reason why MITRE ATT&CK and the OWASP Top 10 exist - these things are far more common, and those vulnerabilities appear on many HTB boxes.

4

Sea First 100 May 31

> Alice:
>
> I do suggest you take a couple of hours or so and read Hacker's Playbook Volume 3

The Hacker's Playbook 3 shows you how to re-create real attacks for red teaming. It was written with that objective in mind, so it goes into far more detail than a post like this to enable you to explain to your client what you are doing and how they can improve their defenses.

2

c0wm1lk Jun 2

Thanks for this amazing guide, I'm keeping this pinned for later, I currently haven't pwn'd any boxes yet but I've been making work of it by reading writeups and watching walkthroughs and trying it myself and failing.

However I think there might be a little error in your guide:

> pry0cc:
>
> For a quick reference containing everything here, read this.

The "this" shows up as if it was a link but it's unclickable.
I'd love to know where it leads to so therefore the little remark 😛

2

hostile.node Jun 7

Gobuster is excellent and is my go-to tool for HTTP enumeration as well. It's important to keep in mind that it is not recursive though, and that adding 401 to the list of status codes to report is useful. And bump the threads, *-t 50* has worked nicely on HTB so far.

As you've said, enumerating is a crucial step. The "enumerate, enumerate, enumerate" mantra that gets thrown around can't be overstated: comprehensive information gathering can save many headaches.

Good shout about extracting subdomains from the certificate, too.

3

pry0cc Leader & Offsec Engineer & Forum Daddy Jun 7

Hey man! I'm glad you got some value from this.

I completely agree, enumeration is the name of the game. If you can't pop it, you've not enumerated enough - generally.

2

SecurityFlaw Jun 7

Nice one. More of this texts please.
Really informative!

c0wm1lk Jun 8

> pry0cc:
>
> For a quick reference containing everything here, read this.

Could you please tell me where the clickable “this” leads to? It doesn’t open any page when I click on it and I’m curious to what it leads.

pry0cc Leader & Offsec Engineer & Forum Daddy Jun 9

Ugh! I thought I fixed it. I will fix this when I get a reliable internet connection.

It’s the DNS and web enumeration reference on delta.navisec.io 30

1

c0wm1lk Jun 10

Thanks for letting me know, and if there are no clicks on a link that probably means it’s broken somehow 😛

xYantix Jun 10

Great guide! A really good base on how to tackle these boxes. This approach might not suit everyone but I think it's good to understand how others tackle various situations.

1

Cgboal Jun 12

Pretty awesome man, thanks for this. Hadn't considered trying SSL cert domain enum for virtual hosts on HTB.

1

15 DAYS LATER

CLOSED JUN 27

This topic was automatically closed after 30 days. New replies are no longer allowed.

3 MONTHS LATER

OPENED SEP 17

pry0cc Leader & Offsec Engineer & Forum Daddy 29d

Cheeky bump - does anybody have any questions regarding HTB and their methodology? Any suggestions? Now is the time to ask!

Reply

Suggested Topics

Topic	Replies	Activity
HackTheBox for Learning Hacking **CTF** ctf, 0x00sec, pentesting, hackthebox, partnership	50	Mar 9
HackTheBox Write-Up - Access **Hackthebox Writeups**	18	May 28
HackTheBox Writeup: Frolic **Hackthebox Writeups**	8	Mar 25
HackTheBox Write-Up - Curling **Hackthebox Writeups**	11	May 28
0x00sec / LeapSecurity CTF 2018 Results **0x00sec Announcem…** ctf, leapsecurity, 0x00ctf2018, results	4	Dec '18

Want to read more? Browse other topics in CTF or view latest topics.