1.6 Bandit Lab

Date	@Jun 5, 2020
Progress	approved
Tags	Bandit Post Meeting Practical Week 1

Bandit 0

For the connection to bandit lab, port 2220 is to be connected thus the command is

kali@kali: `ssh -p 2220 bandit0@bandit.labs.overthewire.org -p 2220`

password: bandit0

Bandit 1 : Dashfilename

If we were to access any dash filename then we have to use the fullname of the folder as:

```
cat - [wrong]
cat ./- [Right]
cat /home/- [Right]
```

Bandit 2: Spaces in the filename

If there are files imported from the windows, the filename might consist of spaces in the name. They can be dealt with using a quotation as:

```
cat './home/spaces in this filename'
```

Bandit 3: Hidden File

Some files might be hidden as they start with '.' For e.g. ".hidden". We can view such file using

```
ls -al
```

We can open such file using

```
cat ./.hidden
```

Bandit 4: File to be found with certain size and type

If we were to find a file type in the directories then

```
find / -type f
```

If we were to find files of size greater than 100 Bytes and less than 200 Bytes then

```
find / -type f -size +100c -size -200c
```

- \`c’ for bytes
- ‘w’ for two-byte words
- \`k’ for Kilobytes
- \`M’ for Megabytes
- \`G’ for Gigabytes

If we were to find files that are not executable and human readable then

```
find . -type f -readable -size 1033c ! -executable
```

If we were to find the files owned by group bandit6 and user bandit7

```
find / -group bandit6 -name bandit7
```

Bandit 7

```
bandit1 : boJ9jbbUNNfktd78OOpsqOltutMc3MY1
bandit2 : CV1DtqXWVFXTvM2F0k09SHz0YwRINYA9
bandit3 : UmHadQclWmgdLOKQ3YNgjWxGoRMb5luK
bandit4 : pIwrPrtPN36QITSp3EQaw936yaFoFgAB
bandit5 : koReBOKuIDDepwhWk7jZC0RTdopnAYKh
bandit6 : DXjZPULLxYr17uwoI01bNLQbtFemEgo7
bandit7 : HKBPTKQnIay4Fw76bEy8PVxKEDQRKTzs
```

Bandit 8: Piping and Redirection

Piping and direction tutorial

https://ryanstutorials.net/linuxtutorial/piping.php

Encoding and Decoding overview

https://linuxhint.com/bash_base64_encode_decode/#:~:text=To%20encode%20or%20decode%20standard,be%20easily%20revealed%20by%20decoding

```
cat data.txt | grep -millionth
password : cvX2JJa4CFALtqS87jk27qwqGhBM9plV
```

Bandit 9

```
cat data.txt | sort | uniq -u

password : UsvVyFSfZZWbi6wgC7dAFyFuR6jQQUh
```

Bandit 10

```
cat data.txt | strings | grep =

password : truKLdjsbJ5g7yyJ2X2R0o3a5HQJFuLk
```

Bandit 11

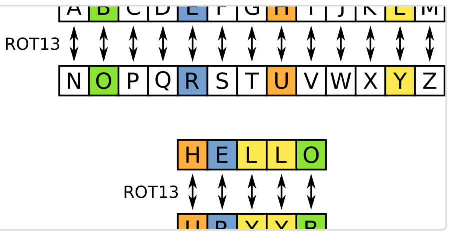
```
base64 -d data.txt | cat

password : IFukwKGsFW8MOq3IRFqrxE1hxTNEbUPR
```

Bandit 12: Rot13

ROT13

ROT13 (" rotate by 13 places", sometimes hyphenated ROT-13) is a simple letter substitution cipher that replaces a letter with the 13th letter after it, in the alphabet. ROT13 is a special case of the Caesar cipher which was

W https://en.wikipedia.org/wiki/ROT13



```
cat data.txt | tr 'A-Za-z' 'N-ZA-Mn-za-m'
password : 5Te8Y4drgCRfCx8ugdwuEX8KFC6k2EUu
```

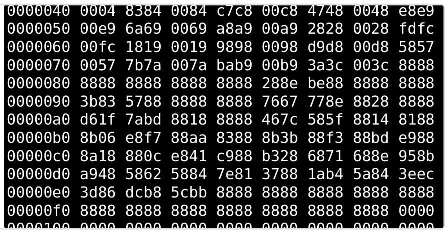
Bandit 13: Hexdump and Multiple File compression

Hexdump

Hex dump

In computing, a hex dump is a hexadecimal view (on screen or paper) of computer data, from RAM or from a computer file or storage device. Looking at a hex dump of data is usually done in the context of either debugging or

W https://en.wikipedia.org/wiki/Hex_dump



```
0000040 0004 8384 0084 c7c8 00c8 4748 0048 e8e9
0000050 00e9 6a69 0069 a8a9 00a9 2828 0028 fdfc
0000060 00fc 1819 0019 9898 0098 d9d8 00d8 5857
0000070 0057 7b7a 007a bab9 00b9 3a3c 003c 8888
0000080 8888 8888 8888 8888 288e be88 8888 8888
0000090 3b83 5788 8888 8888 7667 778e 8828 8888
00000a0 d61f 7abd 8818 8888 467c 585f 8814 8188
00000b0 8b06 e8f7 88aa 8388 8b3b 88f3 88bd e988
00000c0 8a18 880c e841 c988 b328 6871 688e 958b
00000d0 a948 5862 5884 7e81 3788 1ab4 5a84 3eec
00000e0 3d86 dcb8 5cbb 8888 8888 8888 8888 8888
00000f0 8888 8888 8888 8888 8888 8888 8888 0000
```

```
xxd -r data.txt > hello2 —> decodes the hex dump file to original

mv hello2 hello2.gz  —> remane hello2 to hello2.gz

gzip -d hello2.gz —> decompress the compressed file

mv hello2 hello2.bz2 —> rename hello2 to hello2.bz2

bzip2 -d hello2.bz2 —> decompress binary zip file

mv hello2 hello2.gz —> rename hello2 to hello2.gz

gzip -d hello2.gz

mv hello2 hello2.tar

tar -xvf hello2.tar

tar -xvf data5.bin

bzip2 -d data6.bin

tar -xvf data6.bin.out

mv data8.bin hello2.gz

gzip -d hello2.gz

cat hello2
```

```
Password: 8ZjyCRiBWFYkneahHwxCv3wb2a1ORpYL
```

Bandit 14: Open SSH Keys

SSH/Open/Keys

https://help.ubuntu.com/community/SSH/OpenSSH/Keys

```
ssh bandit14@localhost -i sshkey.private

cat /etc/bandit_pass/bandit14

4wcYUJFw0k0XLShlDzztnTBHiqxU3b3e
```

Bandit 15: Netcat

nc = netcat : netcat

The nc (or netcat) utility is used for just about anything under the sun involving TCP or UDP. It can open TCP connections, send UDP packets, listen on arbitrary TCP and UDP ports, do port scanning, and deal with both IPv4 and IPv6

For Port connection and listening:

```
$ nc -l 1234: listening to port 1234

$ nc 127.0.0.1 1234: connecting to the socket 127.0.0.1:1234
```

Now if anything is written in the 2nd terminal then it is printed in the 1st terminal

```
nc [localhost](http://localhost) 30000

4wcYUJFw0k0XLShlDzztnTBHiqxU3b3e

password : BfMYroe26WYalil77FoDi9qh59eK5xNr
```

Bandit 16

Transport Layer Security

1. TLS is the successor of its now-deprecated predecessor secure socket layer(SSL)

2. They are both protocols used for securing the communication over internet.
3. It helps to secure communication between server and web browsers with widespread applications like browsing, email, instant, message and so on.

OpenSSL

OpenSSL is a robust, commercial-grade, and full-featured toolkit for the Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols. It is also a general-purpose cryptography library.

```
openssl s_client -connect localhost:30001
```

here s_client is a SSL/TLS client program

```
Password: cluFn7wTiGryunymYOu4RcffSxQluehd
```

Bandit 17: Port Scanner

1. A port scan or portscan is a process that sends client requests to a range of server port addresses on a host, with the goal of finding an active port;
2. A port scanner is an application designed to probe a server or host for open ports.
3. To portsweep is to scan multiple hosts for a specific listening port.

Port scanner

A port scanner is an application designed to probe a server or host for open ports. Such an application may be used by administrators to verify security policies of their networks and by attackers to identify network services running on a host and exploit vulnerabilities.

W https://en.wikipedia.org/wiki/Port_scanner

nmap = portscanner in linux

Nmap Tutorial

How to Use Nmap to Scan for Open Ports {Updated 2020}

Nmap is the world's leading port security network scanner. The Nmap hosted security tool can help you determine how well your firewall and security configuration is working.

https://phoenixnap.com/kb/nmap-scan-open-ports

What?

Man requires EN-DASH not Hyphen??

Nmap unable to resolve flags

Thanks for contributing an answer to Unix & Linux Stack Exchange! Please be sure to answer the question. Provide details and share your research! Asking for help, clarification, or responding to other answers. Making

https://unix.stackexchange.com/questions/420498/nmap-unable-to-resolve-flags

```
nmap localhost -p31000-32000
```

output:

PORT STATE SERVICE

```
31046/tcp open unknown
31518/tcp open unknown
31691/tcp open unknown
31790/tcp open unknown
31960/tcp open unknown
```

OpenSSL at correct port

```
openssl s_client -connect localhost:31790
```

RSA Key:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAvmOkuifmMg6HL2YPIOjon6iWfbp7c3jx34YkYWqUH57SUdyJ
imZzeyGC0gtZPGujUSxiJSWI/oTqexh+cAMTSMlOJf7+BrJObArnxd9Y7YT2bRPQ
Ja6Lzb558YW3FZl87ORiO+rW4LCDCNd2lUvLE/GL2GWyuKN0K5iCd5TbtJzEkQTu
DSt2mcNn4rhAL+JFr56o4T6z8WWAW18BR6yGrMq7Q/kALHYW3OekePQAzL0VUYbW
JGTi65CxbCnzc/w4+mqQyvmzpWtMAzJTzAzQxNbkR2MBGySxDLrjg0LWN6sK7wNX
x0YVztz/zbIkPjfkU1jHS+9EbVNj+D1XFOJuaQIDAQABAoIBABagpxpM1aoLWfvD
KHcj10nqcoBc4oE11aFYQwik7xfW+24pRNuDE6SFthOar69jp5RlLwD1NhPx3iBl
J9nOM8OJ0VToum43UOS8YxF8WwhXriYGnc1sskbwpXOUDc9uX4+UESzH22P29ovd
d8WErY0gPxun8pbJLmxkAtWNhpMvfe0050vk9TL5wqbu9AlbssgTcCXkMQnPw9nC
YNN6DDP2lbcBrvgT9YCNL6C+ZKufD52yOQ9qOkwFTEQpjtF4uNtJom+asvlpmS8A
vLY9r60wYSvmZhNqBUrj7lyCtXMIu1kkd4w7F77k+DjHoAXyxcUp1DGL51sOmama
+TOWWgECgYEA8JtPxP0GRJ+IQkX262jM3dEIkza8ky5moIwUqYdsx0NxHgRRhORT
8c8hAuRBb2G82so8vUHk/fur85OEfc9TncnCY2crpoqsghifKLxrLgtT+qDpfZnx
SatLdt8GfQ85yA7hnWWJ2MxF3NaeSDm75Lsm+tBbAiyc9P2jGRNtMSkCgYEAypHd
HCctNi/FwjulhttFx/rHYKhLidZDFYeiE/v45bN4yFm8x7R/b0iE7KaszX+Exdvt
SghaTdcG0Knyw1bpJVyusavPzpaJMjdJ6tcFhVAbAjm7enCIvGCSx+X3l5SiWg0A
R57hJglezIiVjv3aGwHwvlZvtszK6zV6oXFAu0ECgYAbjo46T4hyP5tJi93V5HDi
Ttiek7xRVxUl+iU7rWkGAXFpMLFteQEsRr7PJ/lemmEY5eTDAFMLy9FL2m9oQWCg
R8VdwSk8r9FGLS+9aKcV5PI/WEKlwgXinB3OhYimtiG2Cg5JCqIZFHxD6MjEGOiu
L8ktHMPvodBwNsSBULpG0QKBgBAplTfC1HOnWiMGOU3KPwYWt0O6CdTkmJOmL8Ni
```

```
blh9elyZ9FsGxsgtRBXRsqXuz7wtsQAgLHxbdLq/ZJQ7YfzOKU4ZxEnabvXnvWkU
YOdjHdSOoKvDQNWu6ucyLRAWFuISeXw9a/9p7ftpxm0TSgyvmfLF2MIAEwyzRqaM
77pBAoGAMmjmIJdjp+Ez8duyn3ieo36yrttF5NSsJLAbxFpdlc1gvtGCWW+9Cq0b
dxviW8+TFVEBl1O4f7HVm6EpTscdDxU+bCXWkfjuRb7Dy9GOtt9JPsX8MBTakzh3
vBgsyi/sN3RqRBcGU40fOoZyfAMT8s1m/uYv52O6IgeuZ/ujbjY=
-----END RSA PRIVATE KEY-----
```

Change the permission of the file to store the key

```
chmod 600 /tmp/key/b16pkey
```

Store the key in the folder

```
cat /etc/bandit_pass/bandit16 | openssl s_client -connect localhost:31790 -quiet > /tmp/key/b16pkey
```

Use the Key

```
ssh -i /tmp/key/b16pkey bandit17@localhost
```

Password

```
xLYVMN9WE5zQ5vHacb0sZEVqbrp7nBTn
```

Bandit 18

```
diff passwords.new passwords.old
password: kfBf3eYk5BPBRzwjqutbbfE887SVc5Yd
```

Bandit 19

```
ssh bandit18@localhost cat readme
password: IueksS7Ubh8G3DCwVzrTd8rAVOwq3M5x
```

Bandit 20: Setuid

1. setuid and setgid (short for "set user ID" and "set group ID")[1] are Unix access rights flags that allow users to run an executable with the permissions of the executable's owner or group respectively and to change behaviour in directories.

setuid

setuid and setgid (short for "set user ID" and "set group ID") are Unix access rights flags that allow users to run an executable with the permissions of the executable's owner or group respectively and to change behaviour in directories.

W https://en.wikipedia.org/wiki/Setuid

```
./bandit20-do cat /etc/bandit_pass/bandit20
GbKksEFF4yrVs6il55v6gwY5aVje5f0j
```

Bandit 21

Open a port for netcat i.e. network concat in the one terminal

```
nc -nlv -p 31000
```

Then then in the next terminal run the suconnect file at the port 31000, this causes the two terminal to talk through the port.

```
./suconnect 31000
```

Then paste the password of bandit20 in the previous screen and it is done.

```
password : gE269g2h3mw3pwgrj0Ha9Uoqen1c9DGr
```

Bandit 22

Enter the crod.d directory

```
cd  /etc/cron.d
```

print the content of associated 22 named file

```
cat cronjob_bandit22
```

Enter the bin directory to find the cronjob file

```
cd /usr/bin
```

print the content of the file

```
cat cronjob_bandit22.sh
```

print the content of relevant file

```
cat /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv
```

```
Password : Yk7owGAcWjwMVRwrTesJEwB7WVOiILLI
```

Bandit 23

```
cd  /etc/cron.d
cat cronjob_bandit23
cd /usr/bin
cat cronjob_bandit23.sh
(echo I am user bandit23 | md5sum | cut -d ' ' -f 1)
cat  /tmp/8ca319486bfbbc3663ea0fbe81326349
password : jc1udXuA1tiHqjIsL8yaapX5XIAI6i0n
```

Cut command examples

cut command in Linux with examples - GeeksforGeeks

The cut command in UNIX is a command for cutting out the sections from each line of files and writing the result to standard output. It can be used to cut parts of a line by byte position, character and field. Basically the cut

https://www.geeksforgeeks.org/cut-command-linux-examples/

```
echo I am user bandit23 | md5sum | cut -d ' ' -f 1
```

The above code first sends the output "I am user bandit23" as the input to md5sum. All md5sum does is to convert return the 128-bit long checksum. Since all we need is the 128-bit long number only thus the output is cut and required number is received.

Bandit 24

Bash Scripting Tutorial

https://linuxconfig.org/bash-scripting-tutorial

```
mkdir /tmp/kongwenbin      //creating the directory
cd /tmp/kongwenbin
vi shell.sh                 //creating shell script
cp shell.sh /var/spool/bandit24  //copy the script
cat bandit24pass
password : UoMYTrfrBFHyQXmg6gzctqAwOmw1IohZ
```

Bandit 25

Create a directory in tmp named band

Create a file name pass2.sh in the band directory

```
nano pass2.sh
```

write the following code

```
#!/bin/bash
password24=UoMYTrfrBFHyQXmg6gzctqAwOmw1IohZ

for i in {0000..9999}
  do
    echo $password24 $i >> passlist.txt
  done
```

The code generates a file named passlist.txt

Feed the passlist.txt for brute force password cracking in the port 30002

```
nc localhost 30002 < passlist.txt
password : uNG9O58gUE7snukf3bvZ0rxhtnjzSGzG
```

Bandit 26

more → provides a filter for paging   less consider more useful than more for viewing a document

vi → text editor for plain text

id → print the id of the current user

/bin/bash information??

(#!/bin/bash) What exactly is this ?

This first line (#!/bin/bash or #!/bin/sh) has a name. It is known as ' she-bang'. This derives from the concatenation of the tokens sharp (#) and bang (!). It is also called as sh-bang, hashbang, poundbang or hash-pling. In computing, a

https://medium.com/@codingmaths/bin-bash-what-exactly-is-this-95fc8db817bf#:~:text=%2Fbin%2Fbash%20is%20the%20most,well%20developed%20and%20better%20syntax

To check which shell you are using

https://www.cyberciti.biz/tips/how-do-i-find-out-what-shell-im-using.html#:~:text=echo%20%240%20%E2%80%93%20Another%20reliable%20and,you%20open%20a%20terminal%20window

Steps for finding password

first find the shell related to the bandit 26

```
grep "^$USER" /etc/passwd
```

then print the content of the shell of bandit26 as

```
cat /usr/bin/showtext
```

use the sshkey in the home directory as

```
ssh bandit26@localhost -i  bandit26.sshkey
```

narrow down the screen which is a way to break the screen, then type command as

```
:e /etc/bandit_pass/bandit26
5czgV9L3Xx8JPOyRbXh6lQbmIOWvPT6Z
```

Bandit 27

1. first minimize the screen just as before
2. after enter the password for bandit 26, enter v

3. then change the shell by setting it to /bin/bash

```
:set shell=/bin/sh
```

4. Execute the command using :! symbol

5. list all the files in the directory as

```
:! ls -al
```

6. execute the setuid file for bandit27

```
:! ./bandit27-do cat /etc/bandit_pass/bandit27
password : 3ba3118a22e93127a4ed485be72ef5ea
```

Bandit 28

steps for finding the password

1. create a folder in tmp as

```
mkdir /tmp/27
cd /tmp/27
```

2. clone the repo

```
git clone ssh://bandit27-git@localhost/home/bandit27-git/repo
```

3. enter the repo

4. cat the file

```
cat README
password : 0ef186ac70e04ea33b4c1853d2526fa2
```

Bandit 29

1. create a folder in tmp as

```
mkdir /tmp/29
cd /tmp/29
```

2. clone the repo

```
git clone ssh://bandit27-git@localhost/home/bandit27-git/repo
```

3. enter the repo
4. cat the file

```
cat README.md
```

In this exercise we have to roll back to previous version

First check all the commit made

```
git log
```

Then select the commit you want to roll back to and also the name of the file

```
git checkout c086d11a00c0648d095d04c089786efef5e01264 -m "README.md"
cat README.md
password : bbc96594b4e001778eee9975372716b2
```

Bandit 30

After downloading

```
git log -p //check the commit
git branch //check the current branch
git branch -r // check the list of branches available
git checkout dev //switch to branch dev
git log -p //check the commits
password : 5b90576bedb2cc04c86a9e924ce42faf
```

Bandit 31

After downloading the repo

```
git tag //git tagging
git show secret
password: 47e603bb428404d265f59c42920d81e5
```

Bandit 32

After downloading the repo

```
git log //check the log and contents added each time
touch key.txt //create the file to be pushed
echo "May I come in?" > key.txt
rm .gitignore //creating problem so remove .gitignore
git add key.txt //add files to be pushed
git commit -m "Pshing file" //commit
git push //pushing key.txt
password : 56a9bf19c63d650ce78e6ec0354ee45e
```

Bandit 33

```
$0 //to restart the shell
$ cat /etc/bandit_pass/bandit33 //started as bandit33 so get the pass
password : c9c3199ddf4121b10cf581a98d51caee
```