**BANDIT WARGAME (OVER THE WIRE)**

**LEVEL 0**

I OPENED TERMINAL AND TYPED THE FOLLOWING COMMAND

ssh bandit0@bandit.labs.overthewire.org -p2220

-p220 is the port and username & password is bandit 0

**LEVEL 0 -> LEVEL 1**

I ran the ls command to view all folders listed. The output showed a file called “readme”. I read the file

using cat readme and it had the password for the next level.

Password: NH2SXQwcBdpmTEzi3bvBHMM9H66vVXjL

**LEVEL 1 -> LEVEL 2**

ssh bandit1@bandit.labs.overthewire.org -p2220

Using the password found in the earlier level I entered. The next level's password is stored in a file in the

home directory. I accessed the file using cat<- command and I saw that the password is given.

Password: rRGizSaX8Mk1RTb1CNQoXTcYZWU6lgzi

**LEVEL 2 -> LEVEL 3**

The next level's password is stored in a file called spaces in this filename in the home directory.

ssh bandit2@bandit.labs.overthewire.org -p2220 and entered the password.

ls

spaces in the file name

cat “spaces in the file name” and I got the password: aBZ0W5EmUfAf7kHTQeOwd8bauFJ2lAiG for the next level.

**LEVEL 3 -> LEVEL 4**

The password for the next level is stored in a hidden file in the inhere directory.

ssh bandit3@bandit.labs.overthewire.org -p2220 and entered the password.

Ls

Inhere

Cd inhere/

Cat. hidden

Password found: 2EW7BBsr6aMMoJ2HjW067dm8EgX26xNe

**LEVEL 4 -> LEVEL 5**

ssh bandit4@bandit.labs.overthewire.org -p2220 and entered the password.

The password for the next level is stored in the only human-readable file in the inhere directory. Tip: if your terminal is messed up, try the “reset” command.

When trying to read the binary files with "cat" I would get something I could not read.

ls

cd inhere/

ls

file ./\*

cat ./-file07

found the password: lrIWWI6bB37kxfiCQZqUdOIYfr6eEeqR

**LEVEL 5 -> LEVEL 6**

ssh bandit5@bandit.labs.overthewire.org -p2220 and entered the password.

The password for the next level is stored in a file somewhere under the inhere directory and has all of

the following properties:

• human-readable

• 1033 bytes in size

• not executable

ls

cd inhere/

ls

find -type f -size 1033c ! -executable

Cat ./maybehere07/.file2

Found the password P4L4vucdmLnm8I7Vl7jG1ApGSfjYKqJU

**LEVEL 6 -> LEVEL 7**

ssh bandit6@bandit.labs.overthewire.org -p2220 and entered the password.

The password for the next level is stored somewhere on the server and has all of the following

properties:

• owned by user bandit7

• owned by group bandit6

• 33 bytes in size

Password is stored somewhere else in the server so finding it with ls will br difficult so we will use find

command and use the hints mentioned above.

find / -user bandit7 -group bandit6 -size 33c

Found a file containing the password

cat /var/lib/dpkg/info/bandit7.password

Found password Z7WtoNQU2XfjmMtWA8u5rN4vzqu4v99S

**LEVEL 7 -> LEVEL 8**

ssh bandit7@bandit.labs.overthewire.org -p2220 and entered the password.

The password for the next level is stored in the file data.txt next to the word millionth.

Ls

Cat data.txt - don’t use

The hint was that the password is next to the word millionth, so I used the command below to read the

file and then grep the word millionth.

cat data.txt | grep millionth

Found password TESKZC0XvTetK0S9xNwm25STk5iWrBvP

**LEVEL 8 -> LEVEL 9**

ssh bandit8@bandit.labs.overthewire.org -p2220 and entered the password.

The password for the next level is stored in the file data.txt and is the only line of text that occurs only

once. We will use sort command to sort the text inside the file and uniq command to print the not

repeating statements.

cat data.txt | sort | uniq –u

Found password EN632PlfYiZbn3PhVK3XOGSlNInNE00t

**LEVEL 9 -> LEVEL 10**

ssh bandit9@bandit.labs.overthewire.org -p2220 and entered the password.

The password for the next level is stored in the file data.txt in one of the few human-readable strings,

preceded by several ‘=’ characters.

I used the strings command which prints character sequences that are at least 4 characters long

And I used grep to get the exact location of the password.

Ls

cat data.txt | strings | grep ^=

Found the password G7w8LIi6J3kTb8A7j9LgrywtEUlyyp6s

**LEVEL 10 -> LEVEL 11**

ssh bandit10@bandit.labs.overthewire.org -p2220 and entered the password

The password for the next level is stored in the file data.txt, which contains base64 encoded data.

To decode the file I ran the command below:

cat data.txt | base64 –decode

Found the password 6zPeziLdR2RKNdNYFNb6nVCKzphlXHBM

**LEVEL 11 -> LEVEL 12**

The password for the next level is stored in the file **data.txt**, where all lowercase (a-z) and uppercase (A-Z) letters have been rotated by 13 positions.

I searched for rot 13 on Wikipedia and found that it is **a simple letter substitution cipher that replaces a letter with the 13th letter after it in the latin alphabet**.

In order to decrypt it, I have to replace every letter by the letter 13 positions ahead.

cat data.txt | tr '[A-Za-z]' '[N-ZA-Mn-za-m]'

I used **n-z** and **a-m** because tr won’t continue to translate after the Z and **tr** command to translate depending on parameters given.

The password is JVNBBFSmZwKKOP0XbFXOoW8chDz5yVRv

**LEVEL 12 -> LEVEL 13**

The password for the next level is stored in the file **data.txt**, which is a hexdump of a file that has been repeatedly compressed. For this level it may be useful to create a directory under /tmp in which you can work using mkdir. For example: mkdir /tmp/myname123. Then copy the datafile using cp, and rename it using mv (read the manpages!)

Ls

Cat data.txt

It is not readable at all. The tmp directory in root contains the required permissions.

Mkdir /tmp/pc

Cp data.txt /tmp/pc

Cd /tmp/pc

Ls

I googled a bit and found that we can use xxd command to reverse the hex dump.

Xxd –r data.txt data

File data

I received a gzip compressed file. And to decrypt I need to rename the file with the .gz extension and proceed.

Mv data data.gz

Gzip –d data.gz

Ls

File data

Bzip2 –d data

Bzip2 command is used to compress and decompress data,

Ls

File data.out

Mv data.out data1.gz

Gzip –d data1.gz

Ls

File data1

data1: POSIX tar archive (GNU)

Tar command is used to view, create and extract file from archives.

Tar –xvf data1 data5.bin

File data5.bin

Tar –xvf data5.bin data6.bin

File data6.bin

Bzip2 –d data6.bin

File data6.bin.out

Tar –xvf data6.bin.out data8.bin

File data8.bin

Mv data8.bin data8.gz

Gzip –d data8.gz

Ls

File data8

Cat data8

The password is wbWdlBxEir4CaE8LaPhauuOo6pwRmrDw

I used google to learn about tar, gzip and bzip.

**LEVEL 13 -> LEVEL 14**

The password for the next level is stored in **/etc/bandit\_pass/bandit14 and can only be read by user bandit14**. For this level, you don’t get the next password, but you get a private SSH key that can be used to log into the next level. **Note:** **localhost** is a hostname that refers to the machine you are working on.

Ls

I saw a file named as sshkey.private

I ran ssh –i sshkey.private bandit14@hostname –p2220

Asked google.

And searched for file 14 using the hint and read that file,

Cat bandit14

Found the password fGrHPx402xGC7U7rXKDaxiWFTOiF0ENq

**LEVEL 14 -> LEVEL 15**

The password for the next level can be retrieved by submitting the password of the current level to **port 30000 on localhost**.

Netcat command “nc” is a command line utility that allows users to read and write data over a network connection. (Google)

Nc localhost 30000

And I entered this level's password to get the password for the next level.

The password is jN2kgmIXJ6fShzhT2avhotn4Zcka6tnt

Can also use telnet localhost 30000 and enter the password and get the password for the next level.

**LEVEL 15 -> LEVEL 16**

The password for the next level can be retrieved by submitting the password of the current level to **port 30001 on localhost** using SSL encryption.

**Helpful note: Getting “HEARTBEATING” and “Read R BLOCK”? Use -ign\_eof and read the “CONNECTED COMMANDS” section in the manpage. Next to ‘R’ and ‘Q’, the ‘B’ command also works in this version of that command…**

After some surfing on the internet I tried this command and it worked.

openssl s\_client -connect localhost:30001

Found the password : JQttfApK4SeyHwDlI9SXGR50qclOAil1

**LEVEL 16 -> LEVEL 17**

The credentials for the next level can be retrieved by submitting the password of the current level to **a port on localhost in the range 31000 to 32000**. First find out which of these ports have a server listening on them. Then find out which of those speak SSL and which don’t. There is only 1 server that will give the next credentials, the others will simply send back to you whatever you send to it.

I didn't get this level at all so i referred my code camp to solve this level.

Nmap localhost –p 31000-32000

nmap localhost -p 31046,31518,31691,31790,31960 -sV –T4

-sV for service versions and T is for time.

Copy the password for bandit 16 and paste when prompted.

openssl s\_client -connect localhost:31518 i tried this command since it showed only two ports with ssl so i tried one by one and –p31518 didnt work.

openssl s\_client -connect localhost:31790

And this worked. I copied the entire rsa key and

Cd /tmp

Mkdir /game12

Cd game12

And i created a nano file and pasted the entire rsa key in it and saved it as rsafile

Nano rsafile

Ls –la

And i changed the viewing permissions

Chmod 600 rsafile

600 means you can read and write the file or directory and other users have no access to it.

ssh -i rsafile bandit17@localhost -p2220

Yes

And you are logged into bandit 17 automatically. And for the password

Cat /etc/bandit\_pass/bandit17

VwOSWtCA7lRKkTfbr2IDh6awj9RNZM5e

**LEVEL 17 -> LEVEL 18**

There are 2 files in the homedirectory: **passwords.old and passwords.new**. The password for the next level is in **passwords.new** and is the only line that has been changed between **passwords.old and passwords.new**

* 4 stands for "read" [r],
* 2 stands for "write" [w],
* 1 stands for "execute" [x], and
* 0 stands for "no permission."

Ls

Passwords.new passwords.old

So i used the “diff” command to differentiate between those two.

diff passwords.new passwords.old

The top one is the password

Hga5tuuCLF6fFzUpnagiMN8ssu9LFrdg

**LEVEL 18 -> LEVEL 19**

The password for the next level is stored in a file **readme** in the home directory. Unfortunately, someone has modified **.bashrc** to log you out when you log in with SSH.

After reading the ssh manual i tried with command argument

Ssh bandit18@bandit.labs.overthewire.org -p2220 ls

Readme

Ssh bandit18@bandit.labs.overthewire.rog -p2220 cat readme

AwhqfNnAbc1naukrpqDYcF95h7HoMTrC – password found

**LEVEL 19 -> LEVEL 20**

To gain access to the next level, you should use the setuid binary in the homedirectory. Execute it without arguments to find out how to use it. The password for this level can be found in the usual place (/etc/bandit\_pass), after you have used the setuid binary.

Ls –la

S stands for setuid binary file

./bandit20-do cat etc/bandit\_pass/bandit20

And found the password VxCazJaVykI6W36BkBU0mJTCM8rR95XT

**LEVEL 20 -> LEVEL 21**

There is a setuid binary in the home directory that does the following: it makes a connection to localhost on the port you specify as a command line argument. It then reads a line of text from the connection and compares it to the password in the previous level (bandit20). If the password is correct, it will transmit the password for the next level (bandit21).

**NOTE:** Try connecting to your own network daemon to see if it works as you think

After entering bandit20’s shell, we see a binary that connects to a local port, and waits for a connection and the password to be sent to it from outside. We need two shells to accomplish this.

On the other window, we listen to the server running, which when I send the current password returns the next.

Ls

Suconnect

In the 1st shell i type

./suconnect 3242

In the 2nd shell i type

Nc -l 3242

And when the connection is made i paste the password to this level in netcat.

VxCazJaVykI6W36BkBU0mJTCM8rR95XT

NvEJF7oVjkddltPSrdKEFOllh9V1IBcq - this is the password for the next level.

**LEVEL 21 -> LEVEL 22**

A program is running automatically at regular intervals from cron, the time-based job scheduler. Look in /etc/cron.d/ for the configuration and see what command is being executed.

Cron is a program that will run a specific command or script at a specific time or intervals of time.

Cd /etc/cron.d/

Ls

cronjob\_bandit15\_root cronjob\_bandit23 e2scrub\_all

cronjob\_bandit17\_root cronjob\_bandit24 otw-tmp-dir

cronjob\_bandit22 cronjob\_bandit25\_root sysstat

I thought of looking into cronjob\_bandit22 since we need to get bandit 22 level password.

I used the below command to see what the cron job is doing I saw that it is executing a bash script at every minute of every hour as explained by the stars in the beginning.

Cat cronjob\_bandit22

@reboot bandit22 /usr/bin/cronjob\_bandit22.sh &> /dev/null

\* \* \* \* \* bandit22 /usr/bin/cronjob\_bandit22.sh &> /dev/null

Since there is all \* then the script run at every minute.

I tried cd /usr/bin/ and read the bash file using "cat" to see what it does.

Cat cronjob\_bandit22.sh

#!/bin/bash

chmod 644 /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv

cat /etc/bandit\_pass/bandit22 > /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv

The password seems to bet inside the tmp file so i ran the command

Cat /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv

And i found the password as WdDozAdTM2z9DiFEQ2mGlwngMfj4EZff

**LEVEL 22 -> LEVEL 23**

A program is running automatically at regular intervals from **cron**, the time-based job scheduler. Look in **/etc/cron.d/** for the configuration and see what command is being executed.

**NOTE:** Looking at shell scripts written by other people is a very useful skill. The script for this level is intentionally made easy to read. If you are having problems understanding what it does, try executing it to see the debug information it prints.

Just like the previous level there is a cron job configuration on the "/etc/cron.d" folder. When I read the file I saw that it is another bash script being run at every minute.

cronjob\_bandit15\_root cronjob\_bandit23 e2scrub\_all

cronjob\_bandit17\_root cronjob\_bandit24 otw-tmp-dir

cronjob\_bandit22 cronjob\_bandit25\_root sysstat

I then "cd" into the "/usr/bin/" folder "cat" the file to see what it does.

cat cronjob\_bandit23.sh

#!/bin/bash

myname=$(whoami)

mytarget=$(echo I am user $myname | md5sum | cut -d ' ' -f 1)

echo "Copying passwordfile /etc/bandit\_pass/$myname to /tmp/$mytarget"

cat /etc/bandit\_pass/$myname > /tmp/$mytarget

So i tried the following command:

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

And i got the hash as 8ca319486bfbbc3663ea0fbe81326349

So i cat /tmp/8ca319486bfbbc3663ea0fbe81326349

QYw0Y2aiA672PsMmh9puTQuhoz8SyR2G = Password

## 