



IT SECURITY AND PRIVACY

**ESCUELA COLOMBIANA DE INGENIERIA
JULIO GARAVITO**

**IT SECURITY AND PRIVACY
GRUPO 1L**

**LINUX PRACTICE
LAB 2**

**SUBMITTED BY:
JUAN PABLO FERNANDEZ GONZALES
MARIA VALENTINA TORRES MONSALVE**

**SUBMITTED TO:
Ing. DANIEL ESTEBAN VELA LOPEZ**

BOGOTÁ D.C.

**DATE:
03/02/2025**

Table of Content

Level 0	5
Objective	5
Methodology	5
Results: ZjLjTmM6FvvyRnrb2rfNWOZOTa6ip5If	7
Level 0 → Level 1	7
Objective	7
Methodology	7
Level 1 → Level 2	9
Objective	9
Methodology	9
Level 2 → Level 3	10
Objective	10
Methodology	10
Level 3 → Level 4	11
Objective	11
Methodology	11
Level 4 → Level 5	13
Objective	13
Methodology	13
Level 5 → Level 6	14
Objective	14
Methodology	14
Level 6 → Level 7	15
Objective	15
Methodology	15
Level 7 → Level 8	17
Objective	17
Methodology	18
Level 8 → Level 9	19
Objective	19
Methodology	19

Level 9 → Level 10	20
Objective	20
Methodology	20
Level 10 → Level 11	21
Objective	21
Methodology	21
Additional Notes.....	22
Level 11 → Level 12.....	22
Objective	22
Methodology	22
Level 12 → Level 13	23
Objective	23
Methodology	24
Level 13 → Level 14	32
Objective	32
Methodology	32
Level 14 → Level 15	35
Objective	35
Methodology	36
Level 15 → Level 16	37
Objective	37
Methodology	37
Level 16 → Level 17	38
Objective	38
Methodology	38
Level 17 → Level 18	40
Objective	40
Methodology	40
Level 18 → Level 19	41
Objective	41
Methodology	41
Level 19 → Level 20	43

Objective	43
Methodology	43
Level 20 → Level 21	44
Objective	44
Methodology	44
Level 21 → Level 22	46
Objective	46
Methodology	46
Level 22 → Level 23	47
Objective	47
Methodology	47
Level 23 → Level 24	48
Objective	48
Methodology	49
Level 24 → Level 25	51
Objective	51
Methodology	51
Level 25 → Level 26	52
Objective	52
Methodology	53
Level 26 → Level 27	55
Objective	55
Methodology	56
Level 27 → Level 28	57
Objective	57
Methodology	57
Level 28 → Level 29	58
Objective	58
Methodology	58
Level 29 → Level 30	60
Objective	60
Methodology	60

Level 30 → Level 31	62
Objective	62
Methodology	62
Level 31 → Level 32	64
Objective	64
Methodology	64
Result: 3O9RfhqyAlVBEZpVb6LYStshZoqoSx5K.....	66
Level 32 → Level 33	66
Objective	66
Methodology	66
Level 33 → Level 34	67
Objective	67
Methodology	67
Bibliography	68

Level 0

Objective

In this exercise the objective is to log in to the game using SSH, we must enter with the user bandit0 and the password bandit0 and the host bandit.labs.overthewire.org through port 2220.

Methodology

The first thing we will do is enter the computer terminal, since we will be using it to interact with the Bandit server. The command used to establish the connection to the Bandit server is '*ssh bandit0@bandit.labs.overthewire.org -p 2220*'

- **ssh:** Secure Shell, is a network protocol with which we can establish a remote connection from the command line. This connection is secure since the data that is sent cannot be seen by third parties since all traffic sent and received is encrypted.
 - **bandit0@bandit.labs.overthewire.org:** This part of the command tells us which server is going to make the ssh connection to, the first part **bandit0** is the user, @ separates the user and the host, the remaining **bandit.labs.overthewire.org** is the name of the server host for the Bandit challenge.
 - **-p 2220:** -p is the tag that is used to specify the port of the server with which you want to establish the connection since by default the SSH protocol connects on 22, after this label we enter the port number that to be able to start the challenge will be **2220**.

After executing the command, it asks us if we are sure to establish the connection, after accepting we are asked for the bandit0 password . We will finally have full access to the server.

For your convenience we have installed a few useful tools which you can find in the following locations:

```
* gef (https://github.com/hugsy/gef) in /opt/gef/  
* pwndbg (https://github.com/pwndbg/pwndbg) in /opt/pwndbg/  
* gdbinit (https://github.com/gdbinit/Gdbinit) in /opt/gdbinit/  
* pwntools (https://github.com/Gallopsled/pwntools)  
* radare2 (http://www.radare.org/)
```

--[More information]--

For more information regarding individual wargames, visit
<http://www.overthewire.org/wargames/>

For support, questions or comments, contact us on discord or IRC.

Enjoy your stay!

bandit0@bandit:~\$

Results: ZjLjTmM6FvvyRnrb2rfNWOZOTa6ip5If

After entering the command and password was accepted. The user will already have access to the Bandit server with the user bandit0, they will give us some tips to navigate and perform actions on it.

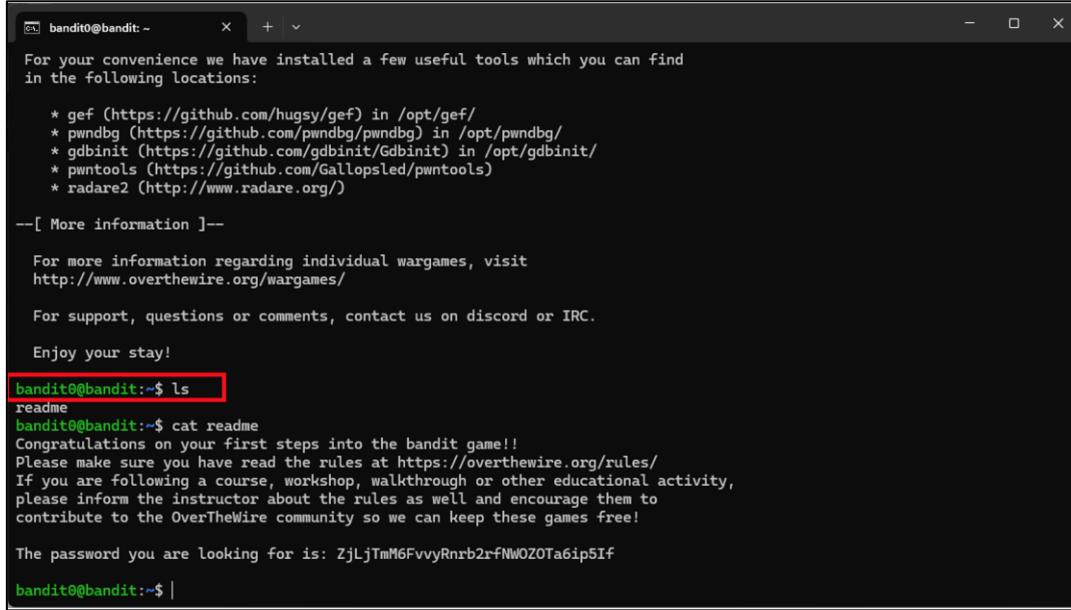
Level 0 → Level 1

Objective

The objective of this level will be to be able to access the file called '**readme**' located in the home directory and obtain the password for the next level that as in the previous level we will connect by ssh on port 2220 with the user bandit1, so it is very important to get the password to be able to progress in the game.

Methodology

The first thing we'll do is we'll list the contents of the directory we're located in to find the **readme** file using '**ls**'



```

bandit0@bandit:~ x + v
For your convenience we have installed a few useful tools which you can find
in the following locations:
* gef (https://github.com/hugsy/gef) in /opt/gef/
* pwndbg (https://github.com/pwndbg/pwndbg) in /opt/pwndbg/
* gdbinit (https://github.com/Gdbinit/Gdbinit) in /opt/gdbinit/
* pwntools (https://github.com/Gallopsled/pwntools)
* radare2 (http://www.radare.org/)

--[ More information ]--

For more information regarding individual wargames, visit
http://www.overthewire.org/wargames/

For support, questions or comments, contact us on discord or IRC.

Enjoy your stay!

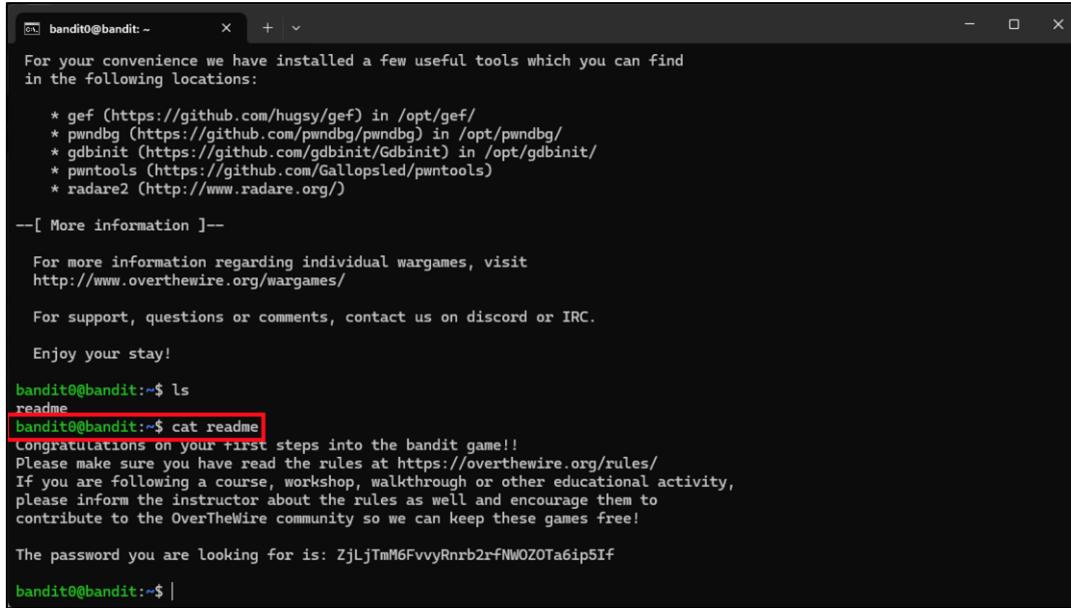
bandit0@bandit:~$ ls
readme
bandit0@bandit:~$ cat readme
Congratulations on your first steps into the bandit game!!
Please make sure you have read the rules at https://overthewire.org/rules/
If you are following a course, workshop, walkthrough or other educational activity,
please inform the instructor about the rules as well and encourage them to
contribute to the OverTheWire community so we can keep these games free!

The password you are looking for is: ZjLjTmM6FvvyRnrb2rfNW0ZOTa6ip5If
bandit0@bandit:~$ |

```

- **ls:** This command allows us to list all the files or directories in the current directory by default.

After having found the ***readme file***, the next step will be to read the contents of it in order to obtain the password, for this we will use the ***command 'cat readme'***



```

bandit0@bandit:~ x + v
For your convenience we have installed a few useful tools which you can find
in the following locations:
* gef (https://github.com/hugsy/gef) in /opt/gef/
* pwndbg (https://github.com/pwndbg/pwndbg) in /opt/pwndbg/
* gdbinit (https://github.com/Gdbinit/Gdbinit) in /opt/gdbinit/
* pwntools (https://github.com/Gallopsled/pwntools)
* radare2 (http://www.radare.org/)

--[ More information ]--

For more information regarding individual wargames, visit
http://www.overthewire.org/wargames/

For support, questions or comments, contact us on discord or IRC.

Enjoy your stay!

bandit0@bandit:~$ ls
readme
bandit0@bandit:~$ cat readme
Congratulations on your first steps into the bandit game!!
Please make sure you have read the rules at https://overthewire.org/rules/
If you are following a course, workshop, walkthrough or other educational activity,
please inform the instructor about the rules as well and encourage them to
contribute to the OverTheWire community so we can keep these games free!

The password you are looking for is: ZjLjTmM6FvvyRnrb2rfNW0ZOTa6ip5If
bandit0@bandit:~$ |

```

- **Cat Readme:** Cat is a command that allows us to read the contents of files and issue them on the command line, its basic structure is 'cat <file_name>'

Results: 263JGPfgU6LtdEvgfWU1XP5yac29mFx

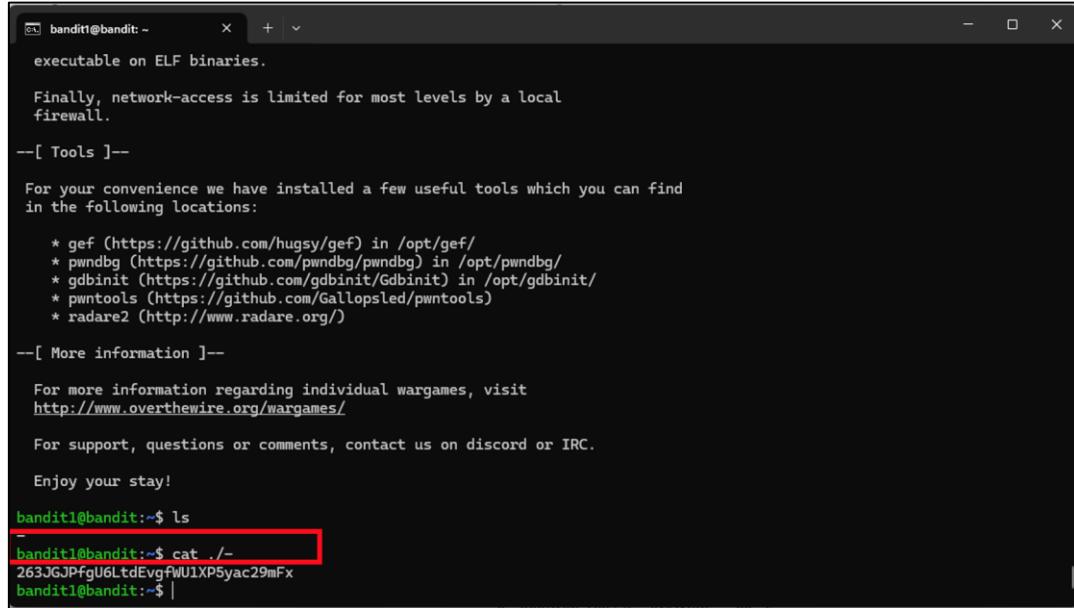
Level 1 → Level 2

Objective

The objective of this level will be to be able to access the file called '-' located in the home directory and obtain the password for the next level to which we will connect by ssh on port 2220 with the user bandit2.

Methodology

After having found the *-file*, the next step will be to read the contents of it in order to obtain the password, for this we will use the **command 'cat ./'**



```
bandit1@bandit:~      x | + | v
executable on ELF binaries.

Finally, network-access is limited for most levels by a local
firewall.

--[ Tools ]--

For your convenience we have installed a few useful tools which you can find
in the following locations:

* gef (https://github.com/hugsy/gef) in /opt/gef/
* pwndbg (https://github.com/pwndbg/pwndbg) in /opt/pwndbg/
* gdbinit (https://github.com/gdbinit/Gdbinit) in /opt/gdbinit/
* pwntools (https://github.com/Gallopsled/pwntools)
* radare2 (http://www.radare.org/)

--[ More information ]--

For more information regarding individual wargames, visit
http://www.overthewire.org/wargames/

For support, questions or comments, contact us on discord or IRC.

Enjoy your stay!

bandit1@bandit:~$ ls
bandit1@bandit:~$ cat ./
263JGPfgU6LtdEvgfWU1XP5yac29mFx
bandit1@bandit:~$ |
```

- **cat ./-**: 'cat' is a command that allows us to read the contents of files and display them in the command line. In this case, we use "**cat ./<file_name>**" because '.' specifies the current directory. Without it, entering "**cat -**" would be interpreted as an option rather than a filename. If we enter this command without specifying a file, it will prompt for user input and print whatever we type to the console instead of displaying the file's contents.

Results: MNk8KNH3Usiio41PRUEoDFPqfxLPISmx

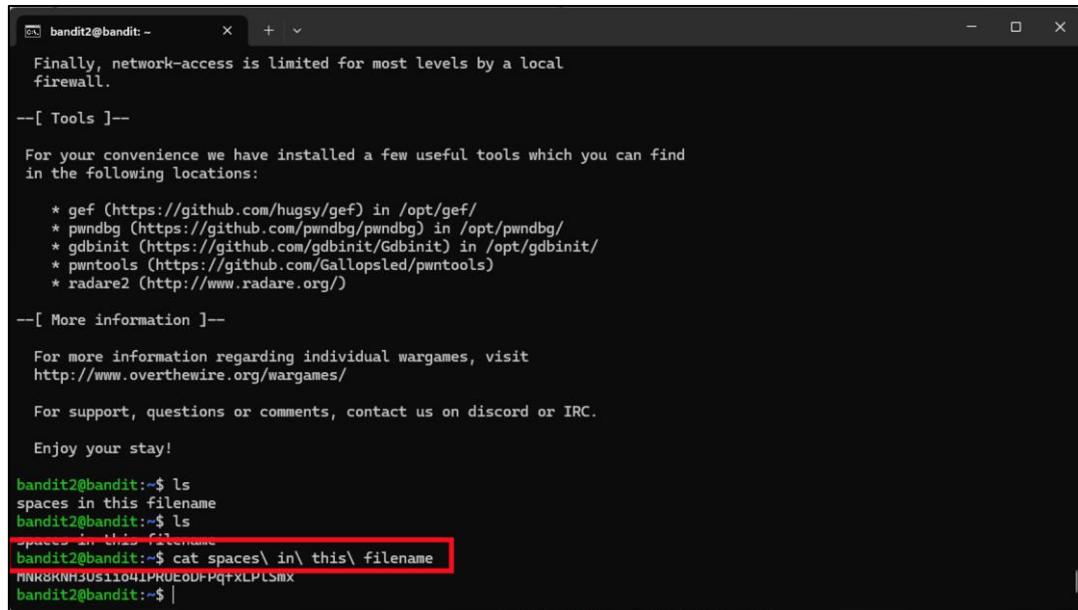
Level 2 → Level 3

Objective

The objective of this level will be to be able to access the file called '*spaces in this filename*' located in the home directory and obtain the password for the next level to which we will connect by ssh on port 2220 with the user bandit3.

Methodology

After having found the *-file*, the next step will be to read the contents of it in order to obtain the password, for this we will use the *command 'cat spaces| in| this\ filename'*



```
bandit2@bandit:~$ ls
Finally, network-access is limited for most levels by a local
firewall.

--[ Tools ]--

For your convenience we have installed a few useful tools which you can find
in the following locations:

* gef (https://github.com/hugsy/gef) in /opt/gef/
* pwndbg (https://github.com/pwndbg/pwndbg) in /opt/pwndbg/
* gdbinit (https://github.com/gdbinit/Gdbinit) in /opt/gdbinit/
* pwntools (https://github.com/Gallopsled/pwntools)
* radare2 (http://www.radare.org/)

--[ More information ]--

For more information regarding individual wargames, visit
http://www.overthewire.org/wargames

For support, questions or comments, contact us on discord or IRC.

Enjoy your stay!

bandit2@bandit:~$ ls
spaces in this filename
bandit2@bandit:~$ ls
spaces in this filename
bandit2@bandit:~$ cat spaces\ in\ this\ filename
MN8KRMH3US11041PNUEDF-PqfxLPLSmx
bandit2@bandit:~$ |
```

- **cat spaces| in| this| filename :** 'cat' is a command that allows us to read the contents of files and display them in the command line. In this case, we use "*spaces| in| this| filename*" because "\ se utiliza para leer y mostrar el contenido de un archivo cuyo nombre contiene espacios. En la línea de comandos, los espacios se interpretan como separadores de argumentos, por lo que es necesario escaparlos con una barra invertida, comillas simples o dobles. Por ejemplo, "*spaces in this filename*" o '*spaces in this filename*'.

Results: 2WmrDFRmJIq3IPxneAaMGhap0pFhF3NJ

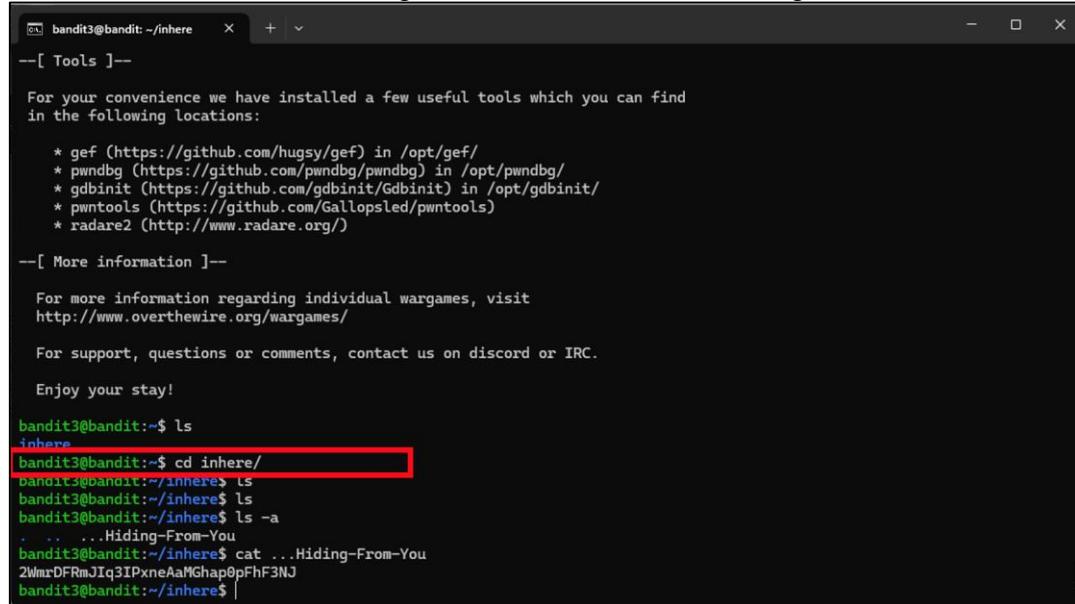
Level 3 → Level 4

Objective

The objective of this level will be to be able to access a hidden file located in the **inhere** directory and obtain the password for the next level to which we will connect by ssh on port 2220 with the user bandit4.

Methodology

To view the files, we will navigate to the "inhere" folder using the cd command.



```
bandit3@bandit:~/inhere
--[ Tools ]--
For your convenience we have installed a few useful tools which you can find
in the following locations:
* gef (https://github.com/hugsy/gef) in /opt/gef/
* pwndbg (https://github.com/pwndbg/pwndbg) in /opt/pwndbg/
* gdbinit (https://github.com/gdbinit/Gdbinit) in /opt/gdbinit/
* pwntools (https://github.com/Gallopsled/pwntools)
* radare2 (http://www.radare.org/)

--[ More information ]--
For more information regarding individual wargames, visit
http://www.overthewire.org/wargames/

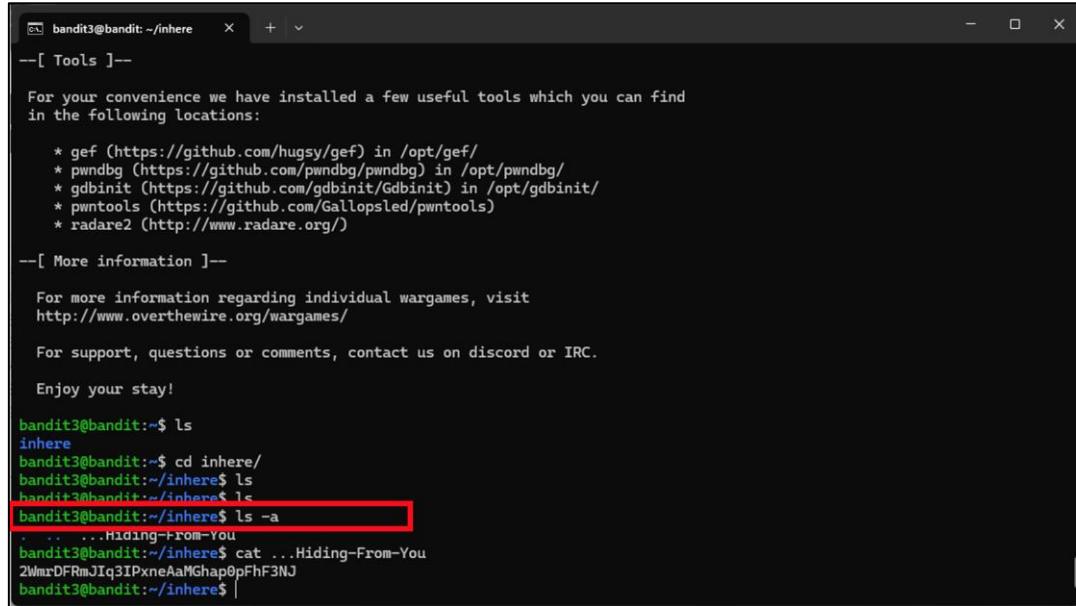
For support, questions or comments, contact us on discord or IRC.

Enjoy your stay!

bandit3@bandit:~$ ls
inhere
bandit3@bandit:~$ cd inhere/
bandit3@bandit:~/inhere$ ls
bandit3@bandit:~/inhere$ ls
bandit3@bandit:~/inhere$ ls -a
. ... ...Hiding-From-You
bandit3@bandit:~/inhere$ cat ...Hiding-From-You
2WmrDFrmIg3IPxneAaMGhap0pFhF3NJ
bandit3@bandit:~/inhere$ |
```

- **cd inhere/:** With the ‘cd’ command, we can navigate between different directories in the system. In this case, we entered the name of the directory we wanted to go to.

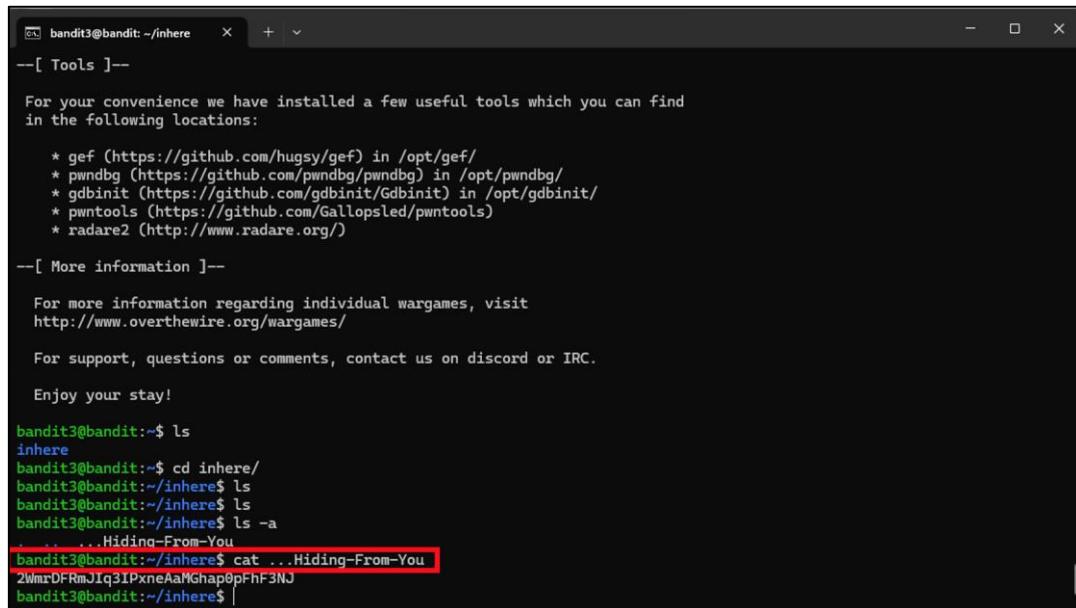
The first thing we'll do is we'll list the contents of the directory we're located in to find the file using '**ls -a**'



```
--[ Tools ]--  
  
For your convenience we have installed a few useful tools which you can find  
in the following locations:  
  
* gef (https://github.com/hugsy/gef) in /opt/gef/  
* pwndbg (https://github.com/pwndbg/pwndbg) in /opt/pwndbg/  
* gdbinit (https://github.com/gdbinit/Gdbinit) in /opt/gdbinit/  
* pwntools (https://github.com/Gallopsled/pwntools)  
* radare2 (http://www.radare.org/)  
  
--[ More information ]--  
  
For more information regarding individual wargames, visit  
http://www.overthewire.org/wargames/  
  
For support, questions or comments, contact us on discord or IRC.  
  
Enjoy your stay!  
  
bandit3@bandit:~$ ls  
inhere  
bandit3@bandit:~$ cd inhere/  
bandit3@bandit:~/inhere$ ls  
bandit3@bandit:~/inhere$ ls  
bandit3@bandit:~/inhere$ ls -a  
... ... ...Hiding-From-You  
bandit3@bandit:~/inhere$ cat ...Hiding-From-You  
2WmrDFRmJIq3IPxneAaMGhap0pFhF3NJ  
bandit3@bandit:~/inhere$ |
```

- **ls -a:** This command allows us to list all the files or directories in the current directory by default, including hidden files thanks to the **-a** flag.

After having found the file, the next step will be to read the contents of it in order to obtain the password, for this we will use the command '**cat ...Hiding-From-You**'



```
--[ Tools ]--  
  
For your convenience we have installed a few useful tools which you can find  
in the following locations:  
  
* gef (https://github.com/hugsy/gef) in /opt/gef/  
* pwndbg (https://github.com/pwndbg/pwndbg) in /opt/pwndbg/  
* gdbinit (https://github.com/gdbinit/Gdbinit) in /opt/gdbinit/  
* pwntools (https://github.com/Gallopsled/pwntools)  
* radare2 (http://www.radare.org/)  
  
--[ More information ]--  
  
For more information regarding individual wargames, visit  
http://www.overthewire.org/wargames/  
  
For support, questions or comments, contact us on discord or IRC.  
  
Enjoy your stay!  
  
bandit3@bandit:~$ ls  
inhere  
bandit3@bandit:~$ cd inhere/  
bandit3@bandit:~/inhere$ ls  
bandit3@bandit:~/inhere$ ls  
bandit3@bandit:~/inhere$ ls -a  
... ... ...Hiding-From-You  
bandit3@bandit:~/inhere$ cat ...Hiding-From-You  
2WmrDFRmJIq3IPxneAaMGhap0pFhF3NJ  
bandit3@bandit:~/inhere$ |
```

Results: 4oQYVPkxZOOEOO5pTW81FB8j8lxXGUQw

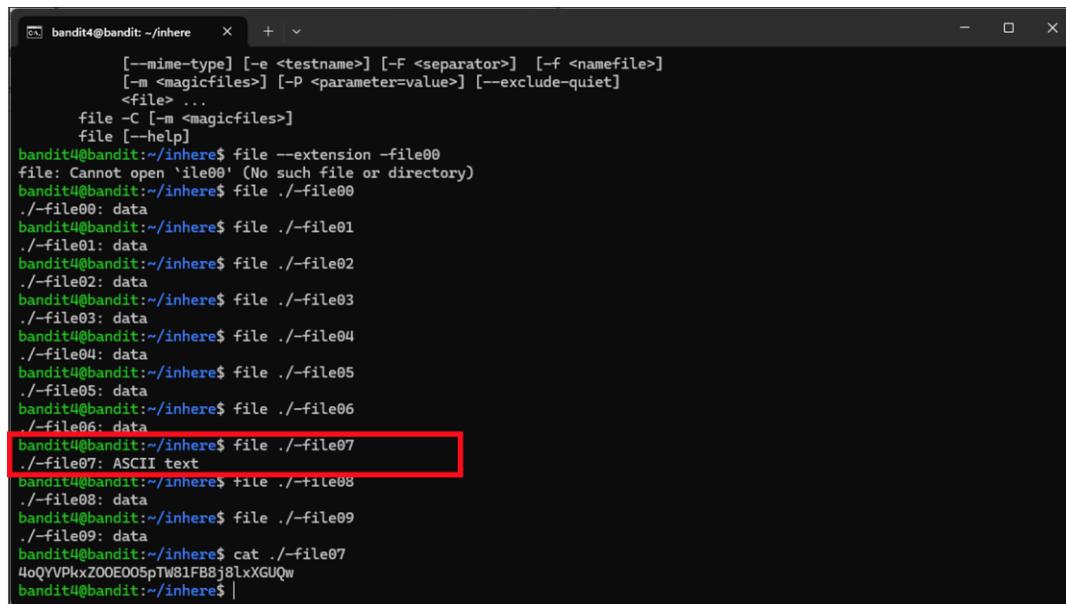
Level 4 → Level 5

Objective

The objective of this level is to find, among the multiple files in the **inhere** directory, the only one that is human-readable and obtain the password for the next level, to which we will connect by SSH on port 2220 with the user **bandit5**.

Methodology

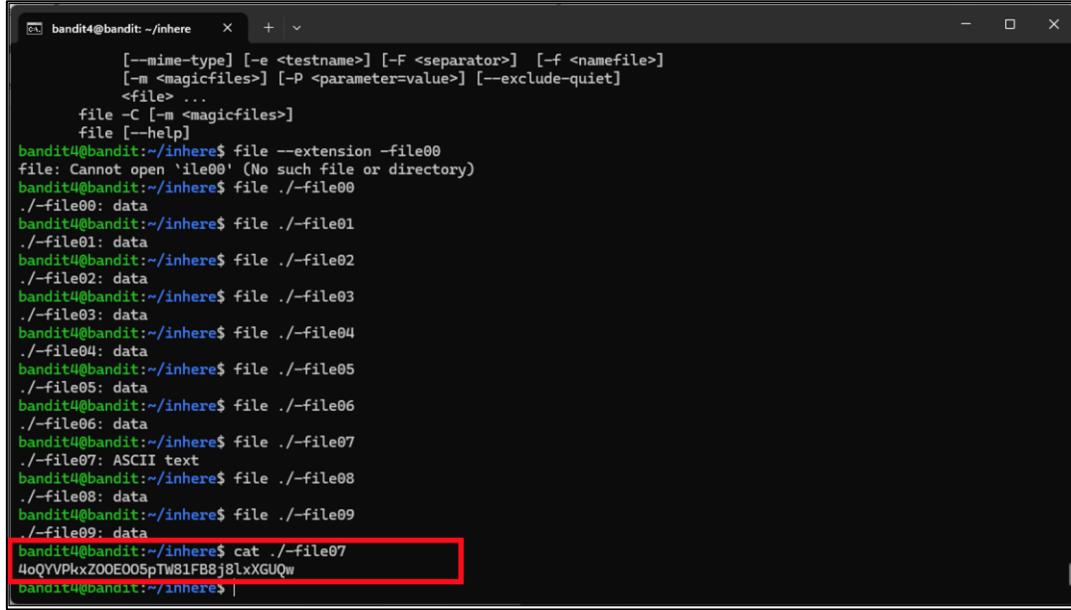
Once inside the **inhere** directory, we will look for a file with an extension that we can read and obtain the password, using the command **file ./<filename>**.



```
bandit4@bandit:~/inhere$ file --extension -file00
file: Cannot open 'file00' (No such file or directory)
bandit4@bandit:~/inhere$ file ./-file00
./-file00: data
bandit4@bandit:~/inhere$ file ./-file01
./-file01: data
bandit4@bandit:~/inhere$ file ./-file02
./-file02: data
bandit4@bandit:~/inhere$ file ./-file03
./-file03: data
bandit4@bandit:~/inhere$ file ./-file04
./-file04: data
bandit4@bandit:~/inhere$ file ./-file05
./-file05: data
bandit4@bandit:~/inhere$ file ./-file06
./-file06: data
bandit4@bandit:~/inhere$ file ./-file07
./-file07: ASCII text
bandit4@bandit:~/inhere$ file ./-file08
./-file08: data
bandit4@bandit:~/inhere$ file ./-file09
./-file09: data
bandit4@bandit:~/inhere$ cat ./-file07
u0QVVPkxZ0OE005ptW81FB8j8lxXGUQw
bandit4@bandit:~/inhere$ |
```

- **file ./<filename>**: The **file** command analyzes a file and displays its type, such as text, binary, image, or executable. The **./** prefix specifies that the file is in the current directory.

After identifying the file that is human-readable, we use the **cat** command. Just like with **file**, we use **./** to prevent **-** from being interpreted as an option and instead treat it as part of the filename, **-file07**.



```

bandit4@bandit:~/inhere$ file --extension -file00
file: Cannot open 'file00' (No such file or directory)
bandit4@bandit:~/inhere$ file ./file00
./file00: data
bandit4@bandit:~/inhere$ file ./file01
./file01: data
bandit4@bandit:~/inhere$ file ./file02
./file02: data
bandit4@bandit:~/inhere$ file ./file03
./file03: data
bandit4@bandit:~/inhere$ file ./file04
./file04: data
bandit4@bandit:~/inhere$ file ./file05
./file05: data
bandit4@bandit:~/inhere$ file ./file06
./file06: data
bandit4@bandit:~/inhere$ file ./file07
./file07: ASCII text
bandit4@bandit:~/inhere$ file ./file08
./file08: data
bandit4@bandit:~/inhere$ file ./file09
./file09: data
bandit4@bandit:~/inhere$ cat ./file07
4oQyVPkxZ00EO05pTW81FB8j8lxXGUQw
bandit4@bandit:~/inhere$ 

```

Results: HWasnPhtq9AVKe0dmk45nxy20cvUa6EG

Level 5 → Level 6

Objective

The objective of this level is to find, among the multiple files in the **inhere** directory, one that is human-readable, has a size of 1033 bytes, and is not executable. Inside, we will find the password for the next level, to which we will connect via SSH on port 2220 with the user bandit6.

Methodology

We will use the command ***find -size 1033c*** first to filter out files that do not meet this criterion, and after doing so, it returns only one result.



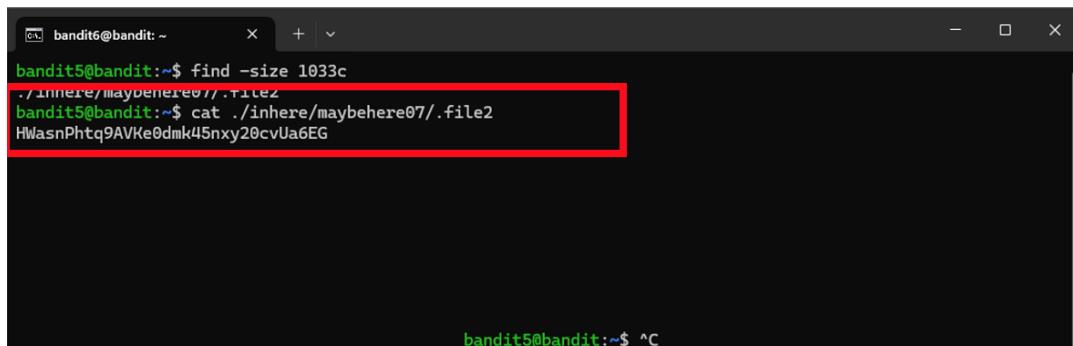
```

bandit5@bandit:~$ find -size 1033c
./inhere/maybehere07/.file2
bandit5@bandit:~$ cat ./inhere/maybehere07/.file2
HWasnPhtq9AVKe0dmk45nxy20cvUa6EG
bandit5@bandit:~$ ^C

```

- **find:** It is a command that allows searching for files and directories in the system based on different criteria.
- **-size 1033c:** Specifies that files with a size of exactly 1033 bytes should be searched for. **1033** is the size of the file in bytes, and **c** indicates that the unit of measurement is in bytes.

After identifying the only file that meets the size criteria, we will access it using the command `cat ./inhere/maybehere07/.file2`, where we will find the password.



```
bandit5@bandit:~$ find -size 1033c
./inhere/maybehere07/.file2
bandit5@bandit:~$ cat ./inhere/maybehere07/.file2
HWasnPhtq9AVk0dmlk45nxy20cvUa6EG
```

Results: morbNTDkSW6jIIUc0ymOdMaLnOlFVAaj

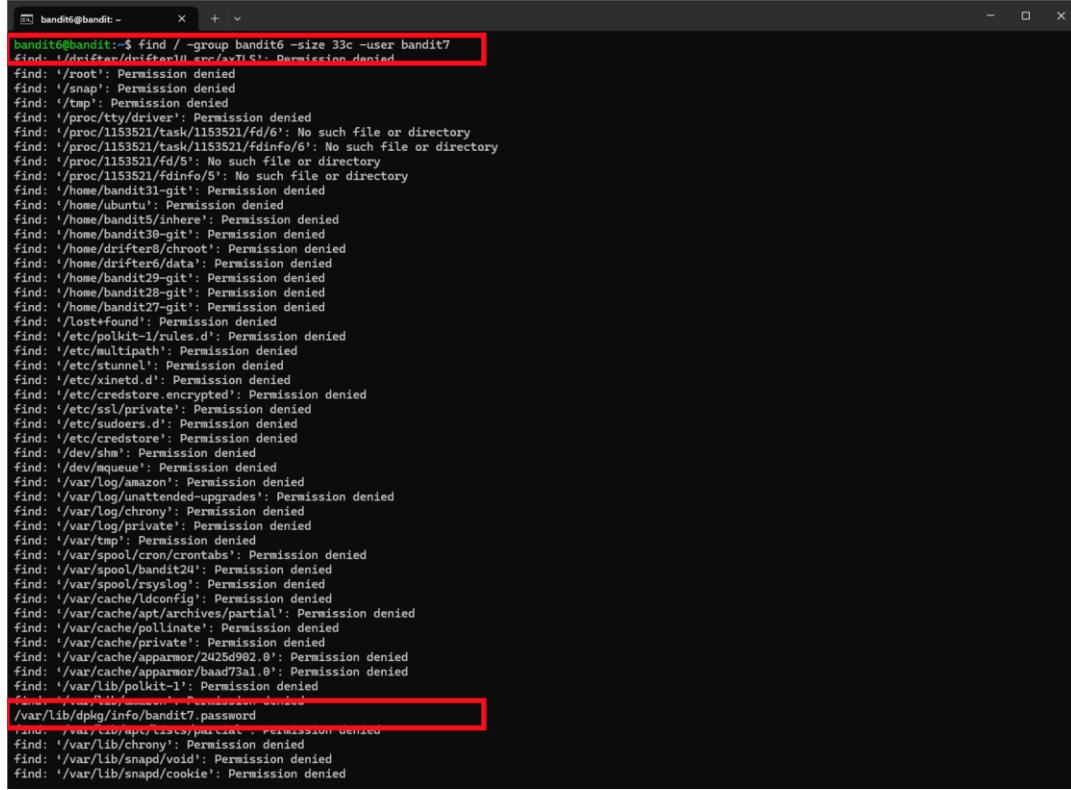
Level 6 → Level 7

Objective

The objective of this level is to find, among the multiple files somewhere on the server, one that is owned by user **bandit7**, owned by group **bandit6**, and has a size of 33 bytes. Inside, we will find the password for the next level, to which we will connect via SSH on port 2220 with user **bandit7**.

Methodology

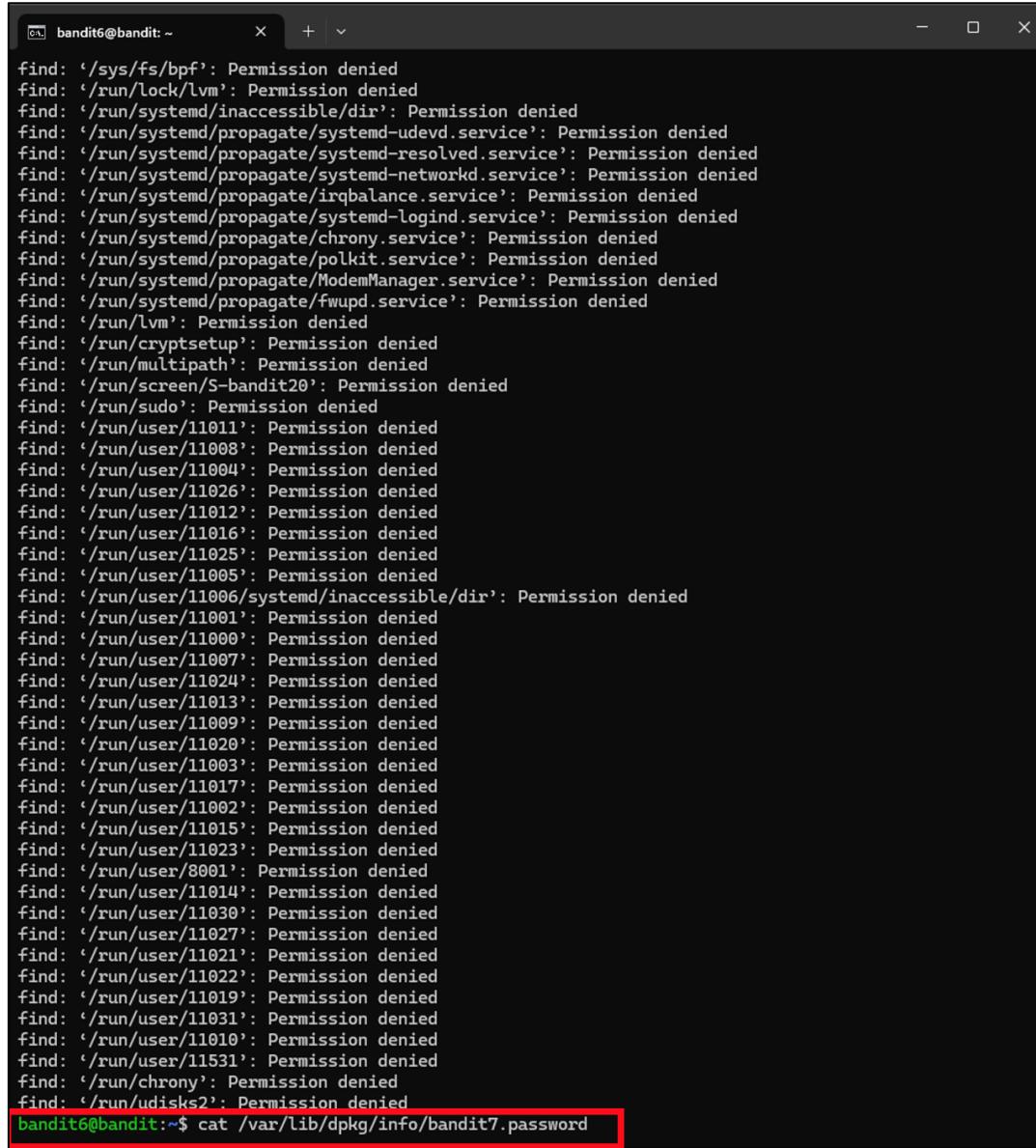
We will use the command `find / -group bandit6 -size 33c -user bandit7` to filter the files that meet the criteria of the tags. We will then identify the one file that is the only one not showing a "Permission Denied" message, which is `/var/lib/dpkg/info/bandit7.password`.



```
bandit6@bandit: ~
$ find / -group bandit6 -size 33c -user bandit7
find: '/root': Permission denied
find: '/snap': Permission denied
find: '/tmp': Permission denied
find: '/proc/tty/driver': Permission denied
find: '/proc/1153521/cashk/1153521/fd/6': No such file or directory
find: '/proc/1153521/fd/5': No such file or directory
find: '/proc/1153521/fdinfo/5': No such file or directory
find: '/home/bandit31-git': Permission denied
find: '/home/ubuntu': Permission denied
find: '/home/bandit5/inhere': Permission denied
find: '/home/bandit30-git': Permission denied
find: '/home/drifter8/chroot': Permission denied
find: '/home/drifter6/data': Permission denied
find: '/home/bandit29-git': Permission denied
find: '/home/bandit28-git': Permission denied
find: '/home/bandit27-git': Permission denied
find: '/lost+found': Permission denied
find: '/etc/polkit-1/rules.d': Permission denied
find: '/etc/multipath': Permission denied
find: '/etc/stunnel': Permission denied
find: '/etc/xinetd.d': Permission denied
find: '/etc/credstore.encrypted': Permission denied
find: '/etc/ssl/private': Permission denied
find: '/etc/sudoers.d': Permission denied
find: '/etc/credstore': Permission denied
find: '/dev/shm': Permission denied
find: '/dev/queue': Permission denied
find: '/var/log/amazon': Permission denied
find: '/var/log/unattended-upgrades': Permission denied
find: '/var/log/chrony': Permission denied
find: '/var/log/private': Permission denied
find: '/var/tmp': Permission denied
find: '/var/spool/cron/crontabs': Permission denied
find: '/var/spool/bandit24': Permission denied
find: '/var/spool/rsyslog': Permission denied
find: '/var/cache/ldconfig': Permission denied
find: '/var/cache/apt/archives/partial': Permission denied
find: '/var/cache/pollinate': Permission denied
find: '/var/cache/private': Permission denied
find: '/var/cache/apparmor/2425d992.0': Permission denied
find: '/var/cache/apparmor/baad73a1.0': Permission denied
find: '/var/lib/polkit-1': Permission denied
...
/var/lib/dpkg/info/bandit7.password
find: '/var/lib/apt/lists/partial': Permission denied
find: '/var/lib/chrony': Permission denied
find: '/var/lib/snap/void': Permission denied
find: '/var/lib/snapd/cookie': Permission denied
```

- **find /:** Starts the search from the root of the file system (the / directory). It will search throughout the entire file system.
- **-group bandit6:** Filters files that belong to the bandit6 group. It will only display files that have this group as their owner.
- **-size 33c:** Searches for files that are exactly 33 bytes in size. The c indicates that the unit of measurement is in bytes.
- **-user bandit7:** Filters files that belong to the bandit7 user. It will only display files that have this user as their owner.

After identifying the only file that meets the size criteria, we will access it using the command `cat /var/lib/dpkg/info/bandit7.password`, where we will find the password.



```
bandit6@bandit:~$ find: '/sys/fs/bpf': Permission denied
find: '/run/lock/lvm': Permission denied
find: '/run/systemd/inaccessible/dir': Permission denied
find: '/run/systemd/propagate/systemd-udevd.service': Permission denied
find: '/run/systemd/propagate/systemd-resolved.service': Permission denied
find: '/run/systemd/propagate/systemd-networkd.service': Permission denied
find: '/run/systemd/propagate/irqbalance.service': Permission denied
find: '/run/systemd/propagate/systemd-logind.service': Permission denied
find: '/run/systemd/propagate/chrony.service': Permission denied
find: '/run/systemd/propagate/polkit.service': Permission denied
find: '/run/systemd/propagate/ModemManager.service': Permission denied
find: '/run/systemd/propagate/fwupd.service': Permission denied
find: '/run/lvm': Permission denied
find: '/run/cryptsetup': Permission denied
find: '/run/multipath': Permission denied
find: '/run/screen/S-bandit20': Permission denied
find: '/run/sudo': Permission denied
find: '/run/user/11011': Permission denied
find: '/run/user/11008': Permission denied
find: '/run/user/11004': Permission denied
find: '/run/user/11026': Permission denied
find: '/run/user/11012': Permission denied
find: '/run/user/11016': Permission denied
find: '/run/user/11025': Permission denied
find: '/run/user/11005': Permission denied
find: '/run/user/11006/systemd/inaccessible/dir': Permission denied
find: '/run/user/11001': Permission denied
find: '/run/user/11000': Permission denied
find: '/run/user/11007': Permission denied
find: '/run/user/11024': Permission denied
find: '/run/user/11013': Permission denied
find: '/run/user/11009': Permission denied
find: '/run/user/11020': Permission denied
find: '/run/user/11003': Permission denied
find: '/run/user/11017': Permission denied
find: '/run/user/11002': Permission denied
find: '/run/user/11015': Permission denied
find: '/run/user/11023': Permission denied
find: '/run/user/8001': Permission denied
find: '/run/user/11014': Permission denied
find: '/run/user/11030': Permission denied
find: '/run/user/11027': Permission denied
find: '/run/user/11021': Permission denied
find: '/run/user/11022': Permission denied
find: '/run/user/11019': Permission denied
find: '/run/user/11031': Permission denied
find: '/run/user/11010': Permission denied
find: '/run/user/11531': Permission denied
find: '/run/chrony': Permission denied
find: '/run/udisks2': Permission denied
bandit6@bandit:~$ cat /var/lib/dpkg/info/bandit7.password
```

Results: dfwvzFQi4mU0wfNbFOe9RoWskMLg7eEc

Level 7 → Level 8

Objective

The objective of this level is to find the word millionth that appears before the password, and then connect via SSH on port 2220 with the user bandit8.

Methodology

We will use the command ***grep -w millionth data.txt*** to search for all the lines where the word **millionth** appears in the **data.txt** file. When we do this, it will return only one line, and immediately after it, we will find the password.

```
bandit7@bandit: ~
directory is regularly wiped.
Please play nice:

* don't leave orphan processes running
* don't leave exploit-files laying around
* don't annoy other players
* don't post passwords or spoilers
* again, DONT POST SPOILERS!
    This includes writeups of your solution on your blog or website!

--[ Tips ]--

This machine has a 64bit processor and many security-features enabled
by default, although ASLR has been switched off. The following
compiler flags might be interesting:

-m32          compile for 32bit
-fno-stack-protector  disable ProPolice
-Wl,-z,norelo  disable relro

In addition, the execstack tool can be used to flag the stack as
executable on ELF binaries.

Finally, network-access is limited for most levels by a local
firewall.

--[ Tools ]--

For your convenience we have installed a few useful tools which you can find
in the following locations:

* gef (https://github.com/hugsy/gef) in /opt/gef/
* pwndbg (https://github.com/pwndbg/pwndbg) in /opt/pwndbg/
* gdbinit (https://github.com/gdbinit/Gdbinit) in /opt/gdbinit/
* pwntools (https://github.com/Gallopsled/pwntools)
* radare2 (http://www.radare.org/)

--[ More information ]--

For more information regarding individual wargames, visit
http://www.overthewire.org/wargames/

For support, questions or comments, contact us on discord or IRC.

Enjoy your stay!

bandit7@bandit:~$ grep -w millionth
^C
bandit7@bandit:~$ grep -w millionth data.txt
millionth  dfwvzFQi4mU0wfNbFOe9RoWskMLg7eEc
bandit7@bandit:~$
```

- **grep:** It is a command used to search for text within a file.
- **-w:** Specifies that **grep** should search for exact matches of the word **millionth**, not partial matches.
- **millionth:** The word we are searching for.

- **data.txt:** The file where the search will be performed.

Results: 4CKMh1JI91bUIZZPXDqGanal4xvAg0JM

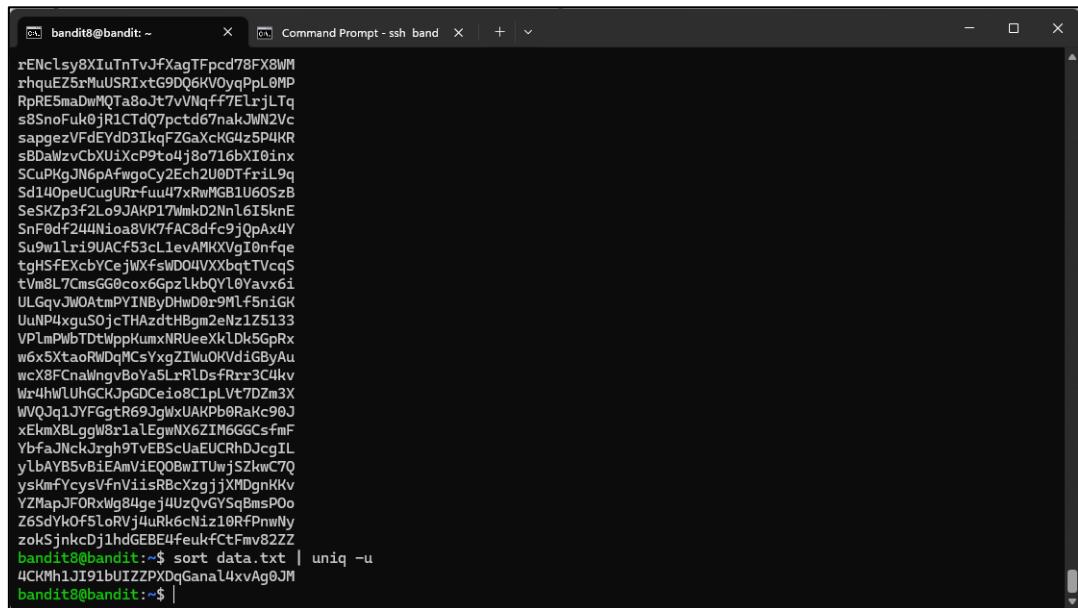
Level 8 → Level 9

Objective

The objective of this level is to find the only line of text that occurs only once, which will be the password that we will use to connect via SSH on port 2220 with the user bandit9.

Methodology

We will be using the command ***sort data.txt | uniq -u*** since we first need to sort the lines and then search for the unique one that does not repeat. After running this command, we will get the password in the console.



```

bandit8@bandit: ~      x  Command Prompt - ssh bandit  x  +  -
rENclsy8XiuTnTvJfxAgTFpcd78FX8wM
rhquE5zMuUSR1xtG9DQ6KV0yqPlOMP
RpRE5maDwMOTa8o7vNqff7ElrjlTq
s8SnoFuk0jR1CTdQ7ptcd67nakJW2Vc
sapgezVFdEYdD31kqFGaXckG4z5P4KR
sBDaWzvCbXUiXcP9t0uj8o716bXI0inx
ScUpKgJN6pAfwoCyz2ech2U0DTfr:iL9q
Sd14OpeUCugURi-fuu74xRwMGB1U60SzB
SeSKZp3f2L09JAKP17wmkD2NnL615knE
SnF0df244NioaBV7fAC8dfc9jOpAx4Y
Su9w1lri9UACf53cl1evAMKXVgI0nfqe
tgHSfExcbYCejWXfsWD04VXXbqtTTVcqS
tVm8LTcmsGG0cox6GpzlkQYl0Yaxv6i
ULGqvJW0AtmPYINByDhwD0r9Mf5niGK
UuNP4xguSOjcThAzdtHBgm2eNz1Z5133
VPImPlubTdtWppkumxNRUeeXhLdh5GrRx
w6x5XtaoRwDqMCsYxgZIw0Kvd1GbAyAu
wcX8FCnaWngvBoYa5LrRldsfRri3C4kv
Wr4hWLUhGCKjpDCeio8C1plVt7Dzm3X
WWQ3qlJYFggtR69JgWxUAKPb0RaKc90J
xEkmXBLLggW8r1alEgwNX6ZIM6GGCsfmF
YbfaJnckJrgh9TveBSzuaEUChrhdJcgIL
ylbAYB5vb1AmviEQOBwITUwjsZkwC7Q
ysKmfycysVfnViisR8cXzgjixMDgnhKv
YZMapJFORxWg84gej4UzQvGYSqBmsP0o
Z65dYkOf5f1oRvj4uRk6cNiZ10RFPhwNy
zok5jnkcDj1hdGEBE4feulkfCtFmvb2ZZ
bandit8@bandit:~$ sort data.txt | uniq -u
4CKMh1JI91bUIZZPXDqGanal4xvAg0JM
bandit8@bandit:~$ |

```

- **sort data.txt:** Sorts the data.txt file.
- **uniq -u:** Displays only the lines that occur once in the file.

Results: FGUW5iLJVrxX9kMYMmlN4MgbpfMiqey

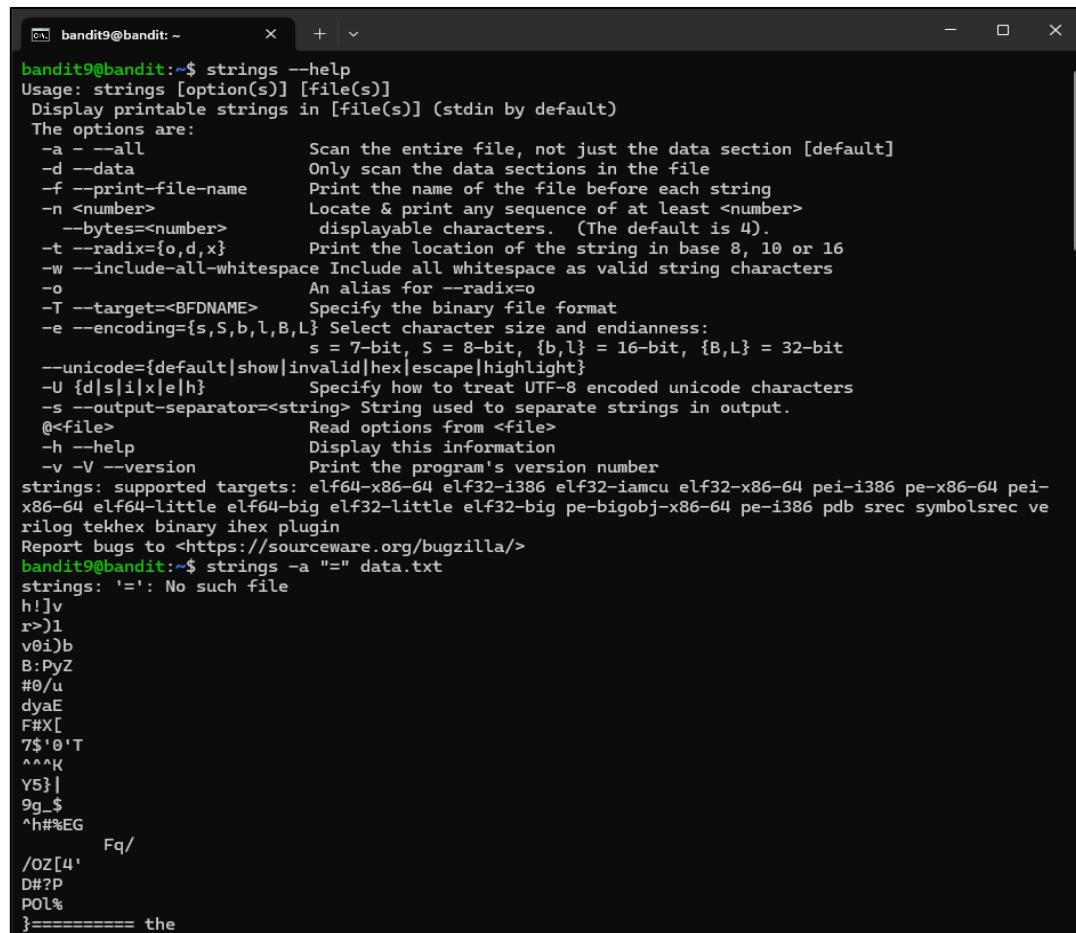
Level 9 → Level 10

Objective

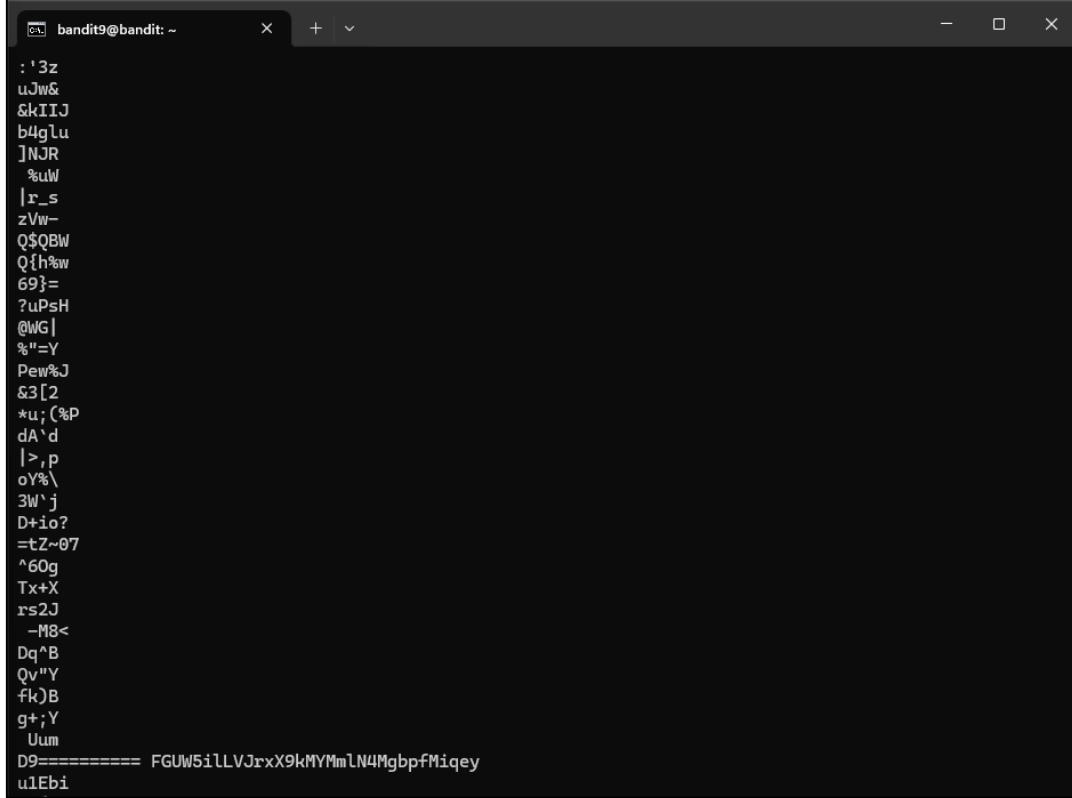
The objective of this level is to find the password that comes after a long string of '=', which we will use to connect via SSH on port 2220 with the user bandit10.

Methodology

With the command **strings -a "=" data.txt**, only readable text strings will appear, not binary ones. After this, it was just a matter of searching through the results to find the password.



```
bandit9@bandit:~$ strings --help
Usage: strings [option(s)] [file(s)]
Display printable strings in [file(s)] (stdin by default)
The options are:
-a - --all          Scan the entire file, not just the data section [default]
-d --data           Only scan the data sections in the file
-f --print-file-name Print the name of the file before each string
-n <number>        Locate & print any sequence of at least <number>
--bytes=<number>   displayable characters. (The default is 4).
-t --radix={o,d,x}  Print the location of the string in base 8, 10 or 16
-w --include-all-whitespace Include all whitespace as valid string characters
-o                 An alias for --radix=o
-T --target=<BFDNAME> Specify the binary file format
-e --encoding={s,S,b,l,B,L} Select character size and endianness:
                     s = 7-bit, S = 8-bit, {b,l} = 16-bit, {B,L} = 32-bit
--unicode={default|show|invalid|hex|escape|highlight}
-U {d|s|i|x|e|h}   Specify how to treat UTF-8 encoded unicode characters
-s --output-separator=<string> String used to separate strings in output.
@<file>           Read options from <file>
-h --help           Display this information
-v -V --version    Print the program's version number
strings: supported targets: elf64-x86-64 elf32-i386 elf32-i386-ia32 elf32-x86-64 pei-i386 pei-x86-64
          elf64-little elf64-big elf32-little elf32-big pe-bigobj-x86-64 pe-i386 pdb srec symbolsrec ve
          rilog tekhex binary ihex plugin
Report bugs to <https://sourceware.org/bugzilla/>
bandit9@bandit:~$ strings -a "=" data.txt
strings: '=': No such file
h!]v
r>)1
v0i)b
B:PyZ
#0/u
dyAE
F#X[
7$'0`T
^^^K
Y5}|_
9g_$
'h##%EG
      Fq/
/oZ[4'
D#?P
POL%
}===== the
```



```
: '3z
uJw&
&KIJ
b4glu
JNJR
%uW
|r_s
zVw-
Q$QBW
Q{h%w
69}=
?Upsh
@WG|
%*=Y
Pew%J
&3[2
*u;(%P
dA`d
|>,p
oY%\'
3W'j
D+io?
=tZ~07
^60g
Tx+X
rs2J
-M8<
Dq^B
Qv"Y
fk)B
g+;Y
Uum
D9===== FGUW5iLLVJrxX9kMYMmlN4MgbpfMiqey
u1Ebi
```

- **strings data.txt:** This command is used to extract and display printable text strings from the data.txt file.
- **-a:** This option tells the strings command to search through the entire file.
- **"=":** We are searching for the equal sign character "=".

Results: dtR173fZKb0RRsDFSGsg2RWnpNVj3qRr

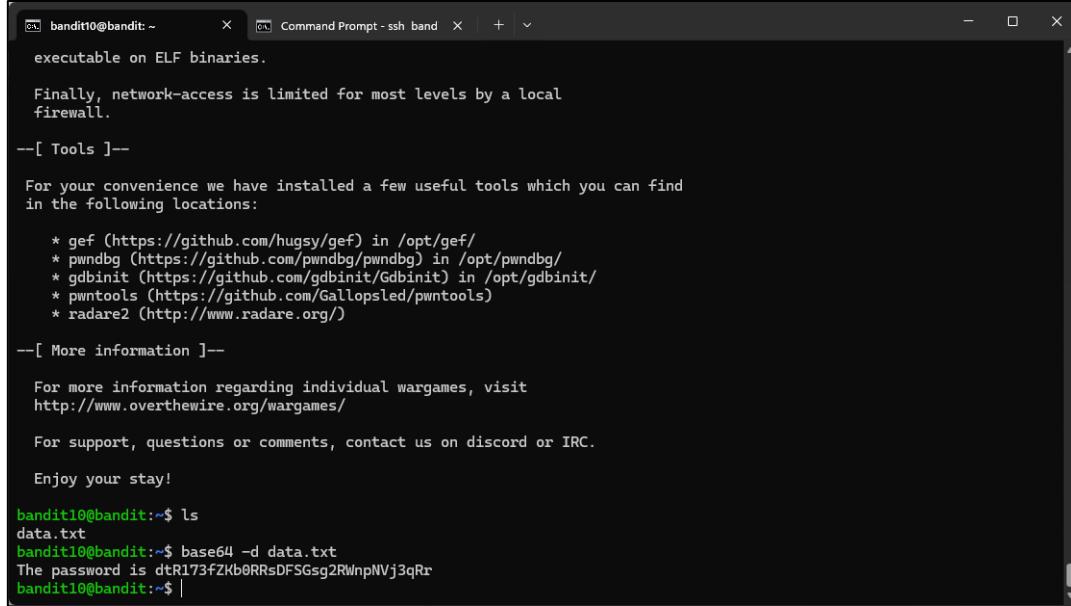
Level 10 → Level 11

Objective

The objective of this level is to find the password for the next level, which is stored in the data.txt file and contains base64 encoded data. We will use this password to connect via SSH on port 2220 with the user bandit11.

Methodology

With the command base64 -d data.txt, decode the content of the file to find the password.



```

bandit10@bandit:~ x Command Prompt - ssh bandit x + x
executable on ELF binaries.

Finally, network-access is limited for most levels by a local
firewall.

--[ Tools ]--

For your convenience we have installed a few useful tools which you can find
in the following locations:

* gef (https://github.com/hugsy/gef) in /opt/gef/
* pwndbg (https://github.com/pwndbg/pwndbg) in /opt/pwndbg/
* gdbinit (https://github.com/gdbinit/Gdbinit) in /opt/gdbinit/
* pwntools (https://github.com/Gallopsled/pwntools)
* radare2 (http://www.radare.org/)

--[ More information ]--

For more information regarding individual wargames, visit
http://www.overthewire.org/wargames/

For support, questions or comments, contact us on discord or IRC.

Enjoy your stay!

bandit10@bandit:~$ ls
data.txt
bandit10@bandit:~$ base64 -d data.txt
The password is dtR173fZKb0RRsDFSGsg2RwNpNVj3qRr
bandit10@bandit:~$ |

```

- **base64:** This command is used to encode or decode data in base64 format.
- **-d:** This option tells the base64 command to decode the data.
- **data.txt:** This is the file that contains the base64 encoded data that we want to decode.

Additional Notes

Base64 encoding is a common method used to represent binary data in an ASCII string format. The base64 command in Linux provides a straightforward way to encode and decode such data.

Results: 7x16WNeHli5YkIhWsfFIqoognUTyj9Q4

Level 11 → Level 12

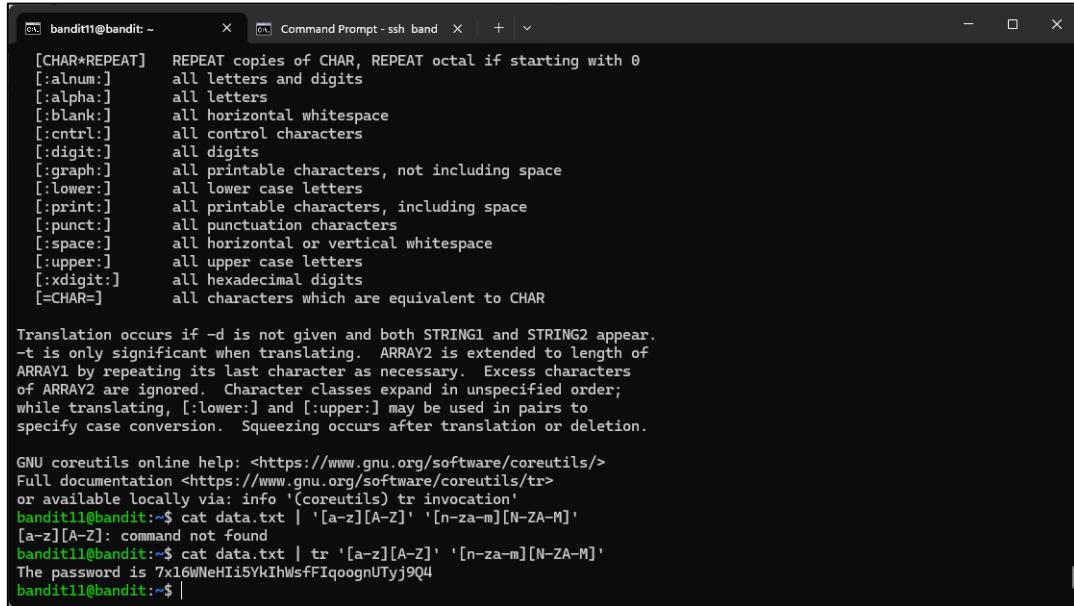
Objective

The objective of this level is to find the password for the next level, which is stored in the file data.txt, where all lowercase (a-z) and uppercase (A-Z) letters have been rotated by 13 positions. We will use this password to connect via SSH on port 2220 with the user bandit12.

Methodology

The ROT13 cipher rotates each letter by 13 positions, so the user employs the tr

command to decode it. The command used is ‘cat data.txt | tr 'A-Za-z' 'N-ZA-Mn-za-m’’



```
[CHAR*REPEAT]  REPEAT copies of CHAR, REPEAT octal if starting with 0
[:alnum:]      all letters and digits
[:alpha:]      all letters
[:blank:]     all horizontal whitespace
[:cntrl:]      all control characters
[:digit:]      all digits
[:graph:]     all printable characters, not including space
[:lower:]      all lower case letters
[:print:]      all printable characters, including space
[:punct:]      all punctuation characters
[:space:]     all horizontal or vertical whitespace
[:upper:]      all upper case letters
[:xdigit:]    all hexadecimal digits
[=CHAR=]       all characters which are equivalent to CHAR

Translation occurs if -d is not given and both STRING1 and STRING2 appear.
-t is only significant when translating. ARRAY2 is extended to length of
ARRAY1 by repeating its last character as necessary. Excess characters
of ARRAY2 are ignored. Character classes expand in unspecified order;
while translating, [:lower:] and [:upper:] may be used in pairs to
specify case conversion. Squeezing occurs after translation or deletion.

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Full documentation <https://www.gnu.org/software/coreutils/tr>
or available locally via: info '(coreutils) tr invocation'
bandit11@bandit:~$ cat data.txt | '[a-z][A-Z]' '[n-za-m][N-ZA-M]'
[a-z][A-Z]: command not found
bandit11@bandit:~$ cat data.txt | tr '[a-z][A-Z]' '[n-za-m][N-ZA-M]'
The password is 7x16WNehIi5YkIhWsfFIqoognUTyj9Q4
bandit11@bandit:~$ |
```

- cat data.txt: Outputs the content of the file.
- tr 'A-Za-z' 'N-ZA-Mn-za-m': The tr (translate) command is used to replace characters based on a specified transformation. In this case, it shifts letters by 13 positions:
 - 'A-Za-z': The input character set includes all uppercase and lowercase letters.
 - 'N-ZA-Mn-za-m': The output character set is the rotated equivalent, with letters shifted by 13 positions. This effectively decodes the ROT13 encoded text, revealing the password.

Results: FO5dwFsc0cbaIiH0h8J2eUks2vdTDwAn

Level 12 → Level 13

Objective

The password for the next level is in **data.txt**, a hexadecimal dump of a repeatedly compressed file. We will need to use different steps to reverse the compression and obtain the original content to connect via SSH to the next level using the password from the file.

Methodology

The user begins by creating a temporary directory using the mktemp command ‘mktemp -d’

```
bandit12@bandit:~$ mktemp -d
/tmp/tmp.Dsd1ByZ3at
bandit12@bandit:~$
```

mktemp -d: This command creates a new temporary directory with a randomly generated name, which is used to avoid conflicts with other files or directories.

The **data.txt** file located in the root directory will be copied to the temporary folder we created earlier with the following command **cp data.txt /tmp/tmp.Dsd1ByZ3at**

```
bandit12@bandit:~$ mktemp -d
/tmp/tmp.Dsd1ByZ3at
bandit12@bandit:~$ ls
data.txt
bandit12@bandit:~$ cp data.txt /tmp/tmp.Dsd1ByZ3at
bandit12@bandit:~$ cd /tmp/tmp.Dsd1ByZ3at
bandit12@bandit:/tmp/tmp.Dsd1ByZ3at$ ls
data.txt
bandit12@bandit:/tmp/tmp.Dsd1ByZ3at$
```

- **cp:** This command is used to copy files or directories.
- **data.txt:** This is the file being copied.
- **/tmp/tmp.Dsd1ByZ3at:** This is the destination directory, which is a temporary folder created earlier using the mktemp -d command. The randomly generated directory name ensures uniqueness and avoids conflicts with other files or processes.

With the command ***cat data.txt | xxd -r > data***, we will convert the plain text file back to binary format from the hexdump.

```
bandit12@bandit:~$ mktemp -d  
/tmp/tmp.Dsd1ByZ3at  
bandit12@bandit:~$ ls  
data.txt  
bandit12@bandit:~$ cp data.txt /tmp/tmp.Dsd1ByZ3at  
bandit12@bandit:~$ cd /tmp/tmp.Dsd1ByZ3at  
bandit12@bandit:~/tmp/tmp.Dsd1ByZ3at$ ls  
data.txt  
bandit12@bandit:~/tmp/tmp.Dsd1ByZ3at$ ls  
data.txt  
bandit12@bandit:~/tmp/tmp.Dsd1ByZ3at$ cat data | xxd -r > data  
bandit12@bandit:~/tmp/tmp.Dsd1ByZ3at$ ls  
data.data.txt  
bandit12@bandit:~/tmp/tmp.Dsd1ByZ3at$ ||
```

- `xxd -r`: Converts the hexdump back into its original binary format.

We use the `file` command at each step to identify the file type: **file data**. We discover that the file is a compressed archive, so we proceed to decompress it using the appropriate commands. This involves renaming files when needed and using the `gzip`, `bzip2`, and `tar` commands.

❖ Decompressing gzip

```
0x00001b0 : 9d53 379f 4851 52a3 4189 f426 9d6c 898a .S7.MR.A...&..1.
0x00001c0 : 9d53 379f 4851 52a3 4189 f426 9d6c 898a .S7.MR.A...&..1.
0x00001d0 : 47f4 5d5b 599d 567a 72b8 898e 9d6b 92a2 .O...X.Vz...Vc.
0x00001e0 : 2569 612c 5364 867d 24a2 888e 9d6b 92a4 N.a.Sd...Sd...
0x00001f0 : 13c7 b602 1ae3 1400 4796 437b fefb 9b43 .....G.Cp...g...C
0x0000200 : a4cb 882a 4aa8 4b81 heft 1c14 6777 8a34 ..#2.K...g...C
0x0000210 : 0867 e5b1 1dfe bbe8 8823 6d1c 416a 28d0 .g.....#mAj(.R)...
0x0000220 : 4460 1684 2652 297d 8788 4a38 e1f9 .B...R).....
0x0000230 : 3622 433a 2652 297d 8788 4a38 e1f9 5f 160b (N.AgB.
0x0000240 : 421f 4a99 97ef 2ec9 83e2 c28f fc5d c914 B.J.....]...
0x0000250 : e1a2 423a 9ecb 1b15 923e 0200 00 .BCw..Vv...>...
bandit12@bandit:~/tmp/[tmp,_IPLNt6TpSS file data.txt
data.txt ASCII text
bandit12@bandit:~/tmp/[tmp,_IPLNt6TpSS file data.txt
data.txt ASCII text
bandit12@bandit:~/tmp/[tmp,_IPLNt6TpSS head data.txt
0x0000000: 1fb8 0888 dcd7 eb66 0203 6461 7461 322e .....f..data2.
0x0000010: 6269 6e00 013e 021c fd42 5a6b 3931 415b bin.>..,B2h91AY
0x0000020: 2653 5924 8382 c108 0017 7fff df3f f4a7 85Y. .....
0x0000030: 0000 0000 0000 0000 0000 0000 0000 0000 .....-.
0x0000040: faff dfbb 97aa 0fff f0ff edff bdbd bdb5 .....0...V
0x0000050: 0400 0034 0000 0000 0000 0000 0000 0000 .....C
0x0000060: 4686 4341 0488 068d 1a69 a008 0068 d1a3 F.CA.....h...
0x0000070: 196e 1193 0433 5193 d46c 5183 4464 93a3 .....30...0.FF.4
0x0000080: 0000 0320 0000 0003 246d 0346 8683 d21a .....8M.F...
0x0000090: 0000 0000 0000 0000 0000 0000 0000 0000 .....0.....
0x00000a0: 0000 0000 0000 0000 0000 0000 0000 0000 .....d4...0....
bandit12@bandit:~/tmp/[tmp,_IPLNt6TpSS head data.txt | xxd -r >data
bandit12@bandit:~/tmp/[tmp,_IPLNt6TpSS 1s
data.txt
bandit12@bandit:~/tmp/[tmp,_IPLNt6TpSS file data
data: gzip compressed data, was "data2.bin", last modified: Thu Sep 19 07:08:15 2024, max compression, from Unix, original size modulo 2^32 57
```

- **mv data data2.gz:** The mv command renames the file from data to data2.gz. The new extension .gz indicates that the file is compressed using

gzip.

```
00000060: 4686 4341 a680 068d 1a69 a0d8 0068 0068 dia@ F.CA.....1...h..  
00000070: 1986 1193 000 5129 d4c4 5183 460d 9a34 .....30...Q.FF.4  
00000080: 8686 0884 5409 0189 a683 47d5 0198 0068 .....44...O.....  
00000090: 9834 c188 8343 0198 a648 69a8 0626 4686 .....C..01..&F.  
000000a0: 8340 0310 d340 3469 a688 0068 0068 8d0d .....04..01.h....  
000000b0: 8686 0884 5409 0189 a683 47d5 0198 0068 .....44..01..&F.  
000000c0: 9a68 4891 8101 3284 012a c469 51a8 1cae ..hh..2...N..0..  
000000e0: 532f 0884 d4d0 50d8 4088 e227 2921 4c8e S/...1.N...1)1  
000000f0: 0886 0884 e5d8 0035 f885 47fc 115a 008c .....L..5..O..Z.  
00000100: 8686 0884 5409 0189 a683 47d5 0198 0068 .....44..01..&F.  
00000110: 2346 3470 8686 01dd d539 a776 f084 #/t1..<.N.70.#  
00000120: :22f 744a b617 a393 3c06 4998 376f dc23 /t1..<.N.70.#  
00000130: 45b1 f723 0887 640b 3534 a289 4197 a7c6 E..#..054.JA..  
00000140: 08bc 7447 d4d0 4a51 d4d3 e288 1899 0882 ..to..30..01..#  
00000150: 8686 0884 5409 0189 a683 47d5 0198 0068 .....44..01..&F.  
00000160: 81a3 7774 1336 8322 cd59 e2b5 9f51 8099 a..1..6..Y..0..  
00000170: :308 2a9d 0330 68f1 f9f6 7db6 930 0d9a ..w..0..)....  
00000180: d47c 0912 1221 0026 9768 6085 9522 91f1 ..l..1..8..0..  
00000190: 8686 0884 5409 0189 a683 47d5 0198 0068 .....44..01..&F.  
000001a0: 852f f646 9792 3898 b983 36c4 c247 ceb1 /f..R..56..0..  
000001b0: 8a53 3797 4831 5283 41e9 f926 9d8c 28f4 S...1..1..1....  
000001c0: 744a 4394 6551 c5bc a76c d085 d986 d222 $...1..1..1....  
000001d0: 8686 0884 5409 0189 a683 47d5 0198 0068 .....44..01..&F.  
000001e0: 2589 412c 5364 8076 2482 8088 9b69 0284 N.a.Sd.1S...  
000001f0: 13c7 0e8b 1aa3 1a48 4796 4370 efcc 9b43 .....G.Cp..C  
00000200: a4c0 8824 a488 4681 abf7 1c14 6777 8a34 ..w.JK..K...g..4  
00000210: 8686 0884 5409 0189 a683 47d5 0198 0068 .....44..01..&F.  
00000220: :460 1049 bba3 2b52 2972 8788 4e38 e1f9 ..(R)..N0..  
00000230: 2466 0f5d 0862 2128 c094 9b48 6754 3891 &f.J0h&(N.KgB.  
00000240: 421f 0f5d 0862 2128 c094 8382 c20f fc5d c914 B.J..J.....].  
00000250: 81d7 0884 5409 0189 a683 47d5 0198 0200 .BC..Y..)....  
bandit12@bandit:~/tmp/tmp_1PLNt6Tp$ file data.txt  
data.txt: ASCII text  
bandit12@bandit:~/tmp/tmp_1PLNt6Tp$ file data.txt  
data.txt: ASCII text  
bandit12@bandit:~/tmp/tmp_1PLNt6Tp$ head data.txt  
00000000: :269 a680 0134 02c1 fd42 5a68 3931 4159 .....f..data2.  
00000001: :269 a680 0134 02c1 fd42 5a68 3931 4159 bin...>..BZn91AY  
00000002: 2653 97ca 8302 c108 0017 77ff df73 f4a7 85V.....  
00000003: 8686 0884 5409 0189 a683 47d5 0198 0068 .....44..01..&F.  
00000004: f4ff f7fa 4fff f0de edf7 b081 3b56 .....6..2..4..0..  
00000005: 0489 0834 0000 0068 a1a1 a080 0343 .....4..1..1....C  
00000006: 4686 4341 a488 0068 1a69 a0d8 0068 d1a8 F.CA.....h..  
00000007: 8686 0884 5409 0189 a683 47d5 0198 0068 .....30...Q.FF.4  
00000008: 0880 0830 0880 0883 2464 0346 8683 d21a .....44..01..&F.  
00000009: 0886 0864 3400 0189 a683 47d5 0198 0061 ..44..01..&F.  
bandit12@bandit:~/tmp/tmp_1PLNt6Tp$ cat data.txt | xxd -r >data  
data: data  
bandit12@bandit:~/tmp/tmp_1PLNt6Tp$ ls  
data  
bandit12@bandit:~/tmp/tmp_1PLNt6Tp$ file data  
data: gzip compressed data, was "data2.bin", last modified: Thu Sep 19 07:08:15 2024, max compression, from Unix, original size modulo 2^32 574  
bandit12@bandit:~/tmp/tmp_1PLNt6Tp$ mv data data2.gz  
bandit12@bandit:~/tmp/tmp_1PLNt6Tp$ ls  
data2.gz  
bandit12@bandit:~/tmp/tmp_1PLNt6Tp$ file data2  
bandit12@bandit:~/tmp/tmp_1PLNt6Tp$ gzip -d data2.gz  
bandit12@bandit:~/tmp/tmp_1PLNt6Tp$ ls  
data2
```

- gzip -d data2.gz:** The gzip -d command decompresses the data2.gz file, removing the compression and producing a decompressed file named data2.

```
000000b0: 8340 0310 d340 3469 a688 0068 0068 8d0d .....04..01.h....  
000000c0: 8686 0884 5409 0189 a683 47d5 0198 0068 .....44..01..&F.  
000000d0: 9a68 4891 8101 3284 012a c469 51a8 1cae ..hh..2...N..0..  
000000e0: 532f 0884 d4d0 50d8 4088 e227 2921 4c8e S/...1.N...1)1  
000000f0: 0886 0884 e5d8 0035 f885 47fc 115a 008c .....L..5..O..Z.  
00000100: 8686 0884 5409 0189 a683 47d5 0198 0068 .....44..01..&F.  
00000110: 2346 3470 8686 01dd d539 a776 f084 #/t1..<.N.70.#  
00000120: :22f 744a b617 a393 3c06 4998 376f dc23 /t1..<.N.70.#  
00000130: 45b1 f723 0887 640b 3534 a289 4197 a7c6 E..#..054.JA..  
00000140: 08bc 7447 d4d0 4a51 d4d3 e288 1899 0882 ..to..30..01..#  
00000150: 8686 0884 5409 0189 a683 47d5 0198 0068 .....44..01..&F.  
00000160: 81a3 7774 1336 8322 cd59 e2b5 9f51 8099 a..1..6..Y..0..  
00000170: :308 2a9d 0330 68f1 f9f6 7db6 930 0d9a ..w..0..)....  
00000180: d47c 0912 1221 0026 9768 6085 9522 91f1 ..l..1..8..0..  
00000190: 8686 0884 5409 0189 a683 47d5 0198 0068 .....44..01..&F.  
000001a0: 852f f646 9792 3898 b983 36c4 c247 ceb1 /f..R..56..0..  
000001b0: 8a53 3797 4831 5283 41e9 f926 9d8c 28f4 S...1..1..1....  
000001c0: 744a 4394 6551 c5bc a76c d085 d986 d222 $...1..1..1....  
000001d0: 8686 0884 5409 0189 a683 47d5 0198 0068 .....44..01..&F.  
000001e0: 2589 412c 5364 8076 2482 8088 9b69 0284 N.a.Sd.1S...  
000001f0: 13c7 0e8b 1aa3 1a48 4796 4370 efcc 9b43 .....G.Cp..C  
00000200: a4c0 8824 a488 4681 abf7 1c14 6777 8a34 ..w.JK..K...g..4  
00000210: 8686 0884 5409 0189 a683 47d5 0198 0068 .....44..01..&F.  
00000220: :460 1049 bba3 2b52 2972 8788 4e38 e1f9 ..(R)..N0..  
00000230: 2466 0f5d 0862 2128 c094 9b48 6754 3891 &f.J0h&(N.KgB.  
00000240: 421f 0f5d 0862 2128 c094 8382 c20f fc5d c914 B.J..J.....].  
00000250: 81d7 0884 5409 0189 a683 47d5 0198 0200 .BC..Y..)....  
bandit12@bandit:~/tmp/tmp_1PLNt6Tp$ file data.txt  
data.txt: ASCII text  
bandit12@bandit:~/tmp/tmp_1PLNt6Tp$ file data.txt  
data.txt: ASCII text  
bandit12@bandit:~/tmp/tmp_1PLNt6Tp$ head data.txt  
00000000: :269 a680 0134 02c1 fd42 5a68 3931 4159 .....f..data2.  
00000001: :269 a680 0134 02c1 fd42 5a68 3931 4159 bin...>..BZn91AY  
00000002: 2653 97ca 8302 c108 0017 77ff df73 f4a7 85V.....  
00000003: 8686 0884 5409 0189 a683 47d5 0198 0068 .....44..01..&F.  
00000004: f4ff f7fa 4fff f0de edf7 b081 3b56 .....6..2..4..0..  
00000005: 0489 0834 0000 0068 a1a1 a080 0343 .....4..1..1....C  
00000006: 4686 4341 a488 0068 1a69 a0d8 0068 d1a8 F.CA.....h..  
00000007: 8686 0884 5409 0189 a683 47d5 0198 0068 .....30...Q.FF.4  
00000008: 0880 0830 0880 0883 2464 0346 8683 d21a .....44..01..&F.  
00000009: 0886 0864 3400 0189 a683 47d5 0198 0061 ..44..01..&F.  
bandit12@bandit:~/tmp/tmp_1PLNt6Tp$ cat data.txt | xxd -r >data  
data: data  
bandit12@bandit:~/tmp/tmp_1PLNt6Tp$ ls  
data  
bandit12@bandit:~/tmp/tmp_1PLNt6Tp$ file data  
data: gzip compressed data, was "data2.bin", last modified: Thu Sep 19 07:08:15 2024, max compression, from Unix, original size modulo 2^32 574  
bandit12@bandit:~/tmp/tmp_1PLNt6Tp$ mv data data2.gz  
bandit12@bandit:~/tmp/tmp_1PLNt6Tp$ ls  
data2.gz  
bandit12@bandit:~/tmp/tmp_1PLNt6Tp$ file data2  
bandit12@bandit:~/tmp/tmp_1PLNt6Tp$ gzip -d data2.gz  
bandit12@bandit:~/tmp/tmp_1PLNt6Tp$ ls  
data2
```

- file data2:** The file command checks the type of the decompressed file data2 to determine what further processing is needed.

❖ Decompressing bzip2

- **mv data2 data3.bz**: The file is renamed from data2 to data3.bz to reflect that it is compressed using bzip2.

```

00000000: 532f 7e84 4408 5db8 4e88 c127 2921 4c0e 5/.J..N..!1L
00000000: b6ed 884c 4408 5db8 4e88 c127 2921 4c0e 5/.J..N..!1L
00000000: b6ed 884c 4408 5db8 4e88 c127 2921 4c0e 5/.J..N..!1L
00000000: 0000 0000 0000 0000 0000 0000 0000 0000 ..0..0..Z.
00000000: c33d 6734 0121 5762 56bc dfbf aef9 b6a7 ..q..!Wb*..
00000000: 2346 1d7b 0e0b 1214 01d0 0000 0000 0000 ..{..B..9.v..
00000000: 0000 0000 0000 0000 0000 0000 0000 0000 ..}..{..A..0.B
00000000: 45b1 5273 0d8f 4408 3534 d299 4395 a7c6 E..E..d.54.)A..
00000000: 0000 0000 0000 0000 0000 0000 0000 0000 ..0..0..#.
00000000: dedf 74af 4408 51d1 0add 0286 1870 .....A..J.
00000000: 9fcf 0e81 4408 51d1 0add 0286 1870 .....A..J.
00000000: 61a3 77fd 1336 8322 0909 02b6 9000 8000 A..)A..Y..Q.
00000000: 0000 0000 0000 0000 0000 0000 0000 0000 ..A..)A..Y..Q.
00000000: dddc 891a 1221 9292 07ea 0b95 9232 91f1 ..I...A..n..*.
00000000: 7bd3 8b4a 4719 0f37 0c36 0f61 02ae de95 ..G.0.7.6.a...
00000000: 0000 0000 0000 0000 0000 0000 0000 0000 ..F..-S..S6.G.
00000000: 0000 0000 0000 0000 0000 0000 0000 0000 ..S..P..A..A..11.
00000000: 24ee +394 453d cb5c 94c d985 0986 d222 S..e..V..A..*.
00000000: 4774 45d6 5897 5679 7268 8956 095c 63c0 G..X.V2.....c.
00000000: 2589 612a 5364 867d 2482 8888 9b66 82ba N..Sd.J.S.....
00000000: 1363 0000 0000 0000 0000 0000 0000 0000 ..G..G.Cp..C
00000000: 0000 0000 0000 0000 0000 0000 0000 0000 ..+X..K..A..4.
00000000: 8867 +56b 1df6 0b98 8923 dd1e 416a 28d0 0.....#.A(j.
00000000: c468 0bb3 6bb3 2e52 297d 8788 4e39 e1f9 .....R)).N.0.
00000000: 2646 8f5d 3962 2628 294e 9848 6754 3891 8f..J0b&..N.Kqt8.
00000000: 0000 0000 0000 0000 0000 0000 0000 0000 ..B.J..(.....).
00000000: 0014 432a 9ecb 8459 1b15 9236 0280 00 80*.Y..>.....
00000000: bandit12@bandit:/tmp/tmp_1PLNt0TpD$ file data.txt
data.txt ASCII text
data.txt ASCII text
bandit12@bandit:/tmp/tmp_1PLNt0TpD$ file data.txt
00000000: f0ff 0888 dfdd eb6b 0283 6463 7461 322e .....f..data2.
00000000: 6269 0e00 813e 02c1 fd42 5a63 3000 4159 bin...,.BZ9H1AY
00000000: 0000 0000 0000 0000 0000 0000 0000 0000 ..0..0..Z.
00000000: fc9f 0afe 7273 cffe f5ff ffd1 br72 5bf6 ..I..0..0..Z.
00000000: faff fdff 77aa 5fff f6ff fdd1 edff 0b97 3b54 ..o.....;V.
00000000: 0000 0000 0000 0000 0000 0000 0000 0000 ..4..-i..C
00000000: 0400 0034 0000 0000 0000 0000 0000 0000 F.CA..-i..C
00000000: 0000 0000 0000 0000 0000 0000 0000 0000 ..30..0.FF.4
00000000: 0000 0326 0688 0000 2464 0346 8683 021a ....8&F...
00000000: 8686 8064 3400 0189 4e63 01d9 0109 001a ..d4...0.....
bandit12@bandit:/tmp/tmp_1PLNt0TpD$ cat data.txt | xz -r > data
bandit12@bandit:/tmp/tmp_1PLNt0TpD$ gzip -d data.txt
data.txt
bandit12@bandit:/tmp/tmp_1PLNt0TpD$ file data
data: gzip compressed data, was "data2.bin", last modified: Thu Sep 19 07:08:15 2024, max compression, from Unix, original size modulo 2^32 6
bandit12@bandit:/tmp/tmp_1PLNt0TpD$ mv data2.gz data2.gz
bandit12@bandit:/tmp/tmp_1PLNt0TpD$ ls
data.gz
bandit12@bandit:/tmp/tmp_1PLNt0TpD$ gzip -d data2.gz
bandit12@bandit:/tmp/tmp_1PLNt0TpD$ mv data2 data3.gz
bandit12@bandit:/tmp/tmp_1PLNt0TpD$ ls
data.gz
bandit12@bandit:/tmp/tmp_1PLNt0TpD$ 

```

- **bzip2 -d data3.bz**: The bzip2 -d command decompresses the data3.bz file, producing a decompressed file named data3.

- **file data3**: The file command checks the type of the decompressed file data3 to identify the next processing step.

- ❖ Decompressing another gzip file
 - **mv data3 data4.gz**: The file is renamed from data3 to data3.gz to reflect that it is compressed using gzip.

```

00000000: 1f8b 0805 0fcd 0b0c 0295 0445 74d1 327e .....f..data2.
00000010: 2d09 8a00 8152 82c1 8a0d 3031 4289 bin.>...B2H91AY
00000020: 2d53 59ca 83b2 c108 0017 7fff bf7e 5bfe .....-l.
00000030: fc9f fefc f2f3 cffe f5ff fffd bf7e 5bfe .....-l.
00000040: faff dfb8 97aa 6fff f0de edf7 b081 3b56 .....o.....-l.
00000050: 0400 9834 0000 0000 0069 a1a1 a000 0300 .....C
00000060: 4486 4311 a090 0000 1a67 a0b0 0069 0300 .....C
00000070: 1986 1193 9433 5193 dc4c 5183 4466 9a34 F.CA.....-l..h.
00000080: 0000 0320 0680 0003 264d 0346 8683 d21a ....30...Q.FF.4
00000090: 0680 0864 3408 0189 a083 47d5 0198 001e ...04....O....
bandit12@bandit:/tmp/tmp.1PLNt6TpSS$ cat data.txt | xxd -r >data
bandit12@bandit:/tmp/tmp.1PLNt6TpSS$ ls
data1.data.txt
bandit12@bandit:/tmp/tmp.1PLNt6TpSS$ file data1
data1: gzip compressed data, was "data1.bin", last modified: Thu Sep 19 07:08:15 2024, max compression, from Unix, original size modulo 2^32 574
bandit12@bandit:/tmp/tmp.1PLNt6TpSS$ mv data1 data2.gz
bandit12@bandit:/tmp/tmp.1PLNt6TpSS$ ls
data2.gz
data1.data.txt
bandit12@bandit:/tmp/tmp.1PLNt6TpSS$ gzip -d data2.gz
bandit12@bandit:/tmp/tmp.1PLNt6TpSS$ ls
data2.data.txt
bandit12@bandit:/tmp/tmp.1PLNt6TpSS$ file data2
data2: bzip2 compressed data, block size 9000
bandit12@bandit:/tmp/tmp.1PLNt6TpSS$ mv data2 data3.gz
bandit12@bandit:/tmp/tmp.1PLNt6TpSS$ ls
data3.gz
data1.data.txt
bandit12@bandit:/tmp/tmp.1PLNt6TpSS$ gzip -d data3.gz
Command 'gzip' not found, did you mean:
  command 'bzip2' from deb bzip2 (1.8.8-5.1build0.1)
  command 'gzip' from deb gzip (1.12~ubuntu1)
Try: apt install <deb name>
bandit12@bandit:/tmp/tmp.1PLNt6TpSS$ bzip2 -d data3.bz
bzip2: Can't open input file data3.bz: No such file or directory.
bandit12@bandit:/tmp/tmp.1PLNt6TpSS$ ls
data3.gz
data1.data.txt
bandit12@bandit:/tmp/tmp.1PLNt6TpSS$ mv data3 data4.gz
bandit12@bandit:/tmp/tmp.1PLNt6TpSS$ ls
data4.gz
data1.data.txt
bandit12@bandit:/tmp/tmp.1PLNt6TpSS$ ls

```

- **gzip -d data4.gz:** The gzip -d command decompresses the data4.gz file, producing a decompressed file named data4.

```

00000000: fc9f fefc f2f3 cffe f5ff fffd bf7e 5bfe .....-l.
00000040: faff dfb8 97aa 6fff f0de edf7 b081 3b56 .....o.....-l.
00000050: 0400 9834 0000 0000 0069 a1a1 a000 0300 .....C
00000060: 4486 4311 a090 0000 1a67 a0b0 0069 0300 .....C
00000070: 1986 1193 9433 5193 dc4c 5183 4466 9a34 F.CA.....-l..h.
00000080: 0000 0320 0680 0003 264d 0346 8683 d21a ....30...Q.FF.4
00000090: 0680 0864 3408 0189 a083 47d5 0198 001e ...04....O....
bandit12@bandit:/tmp/tmp.1PLNt6TpSS$ cat data.txt | xxd -r >data
bandit12@bandit:/tmp/tmp.1PLNt6TpSS$ ls
data1.data.txt
bandit12@bandit:/tmp/tmp.1PLNt6TpSS$ file data1
data1: gzip compressed data, was "data1.bin", last modified: Thu Sep 19 07:08:15 2024, max compression, from Unix, original size modulo 2^32 574
bandit12@bandit:/tmp/tmp.1PLNt6TpSS$ mv data1 data2.gz
bandit12@bandit:/tmp/tmp.1PLNt6TpSS$ ls
data2.gz
data1.data.txt
bandit12@bandit:/tmp/tmp.1PLNt6TpSS$ gzip -d data2.gz
bandit12@bandit:/tmp/tmp.1PLNt6TpSS$ ls
data2.data.txt
bandit12@bandit:/tmp/tmp.1PLNt6TpSS$ file data2
data2: bzip2 compressed data, block size 9000
bandit12@bandit:/tmp/tmp.1PLNt6TpSS$ mv data2 data3.gz
bandit12@bandit:/tmp/tmp.1PLNt6TpSS$ ls
data3.gz
data1.data.txt
bandit12@bandit:/tmp/tmp.1PLNt6TpSS$ gzip -d data3.gz
Command 'gzip' not found, did you mean:
  command 'bzip2' from deb gzip (1.12~ubuntu1)
  command 'gzip' from deb gzip (1.8.8-5.1build0.1)
Try: apt install <deb name>
bandit12@bandit:/tmp/tmp.1PLNt6TpSS$ bzip2 -d data3.bz
bzip2: Can't open input file data3.bz: No such file or directory.
bandit12@bandit:/tmp/tmp.1PLNt6TpSS$ ls
data3.gz
data1.data.txt
bandit12@bandit:/tmp/tmp.1PLNt6TpSS$ mv data3 data4.gz
bandit12@bandit:/tmp/tmp.1PLNt6TpSS$ ls
data4.gz
data1.data.txt
bandit12@bandit:/tmp/tmp.1PLNt6TpSS$ ls

```

- **file data4:** The file command is used to inspect the type of the file data4.

```

00000059: 0400 0934 0000 0000 0069 a1a1 a0d3 0934 ..4...1....C
00000060: a468 4343 068d 00d8 1a69 a0b8 0668 d1a0 F.CA.....h...
00000061: 3030 0000 0000 a033 5193 0000 0000 92a0 .....30.....F.4
00000062: 0000 0000 0000 0000 24a0 0344 0663 0000 .....AMF....
00000063: 0850 0864 0289 a083 47d5 0198 0000 0000 .....,d.....
bandit112@bandit:~/tmp/tmp_1PLNt6tpd$ cat data.txt | xxd -r >data
bandit112@bandit:~/tmp/tmp_1PLNt6tpd$ ls
data1.gz
data2.gz
bandit112@bandit:~/tmp/tmp_1PLNt6tpd$ gzip -d data2.gz
bandit112@bandit:~/tmp/tmp_1PLNt6tpd$ ls
data2_data1.gz
data2_data1.bz
bandit112@bandit:~/tmp/tmp_1PLNt6tpd$ file data2
data2: bzip2 compressed data, block size = 980k
bandit112@bandit:~/tmp/tmp_1PLNt6tpd$ mv data2 data3.gz
bandit112@bandit:~/tmp/tmp_1PLNt6tpd$ ls
data3.gz
data3.bz
bandit112@bandit:~/tmp/tmp_1PLNt6tpd$ gzip2 -d data3.gz
Command "gzip2" not found, did you mean:
  command "bzip2" from deb bzip2 (1.0-8.5-1ubuntu0.1)
  command "gzip" from deb gzip (1.12-1ubuntu1)
Try: apt install bzip2
bandit112@bandit:~/tmp/tmp_1PLNt6tpd$ gzip2 -d data3.bz
Command "gzip2" not found, did you mean:
  command "bzip2" from deb bzip2 (1.0-8.5-1ubuntu0.1)
  command "gzip" from deb gzip (1.12-1ubuntu1)
Try: apt install gzip or name
bandit112@bandit:~/tmp/tmp_1PLNt6tpd$ bzip2 -d data3.bz
bzip2: Can't open input file data3.bz: No such file or directory.
bandit112@bandit:~/tmp/tmp_1PLNt6tpd$ ls
data1_data2.gz
bandit112@bandit:~/tmp/tmp_1PLNt6tpd$ ls
data3.gz
bandit112@bandit:~/tmp/tmp_1PLNt6tpd$ file data3
bandit112@bandit:~/tmp/tmp_1PLNt6tpd$ file data3 (no such file or directory)
bandit112@bandit:~/tmp/tmp_1PLNt6tpd$ bzip2 -d data3.bz
bzip2: Can't open input file data3.bz: No such file or directory.
bandit112@bandit:~/tmp/tmp_1PLNt6tpd$ mv data3 data3.bz
mv: cannot stat 'data3': No such file or directory
bandit112@bandit:~/tmp/tmp_1PLNt6tpd$ gzip -d data3.bz
bandit112@bandit:~/tmp/tmp_1PLNt6tpd$ file data3
data3: gzip compressed data, was "data3-bin", last modified: Thu Sep 19 07:08:15 2024, max compression, from Unix, original size modulo 2^32 2048B
bandit112@bandit:~/tmp/tmp_1PLNt6tpd$ mv data3 data4.gz
data4.gz
bandit112@bandit:~/tmp/tmp_1PLNt6tpd$ gzip -d data4.gz
bandit112@bandit:~/tmp/tmp_1PLNt6tpd$ ls
data4_data5.gz
bandit112@bandit:~/tmp/tmp_1PLNt6tpd$ file data4
data4: POSIX tar archive (GNU)
bandit112@bandit:~/tmp/tmp_1PLNt6tpd$ 

```

❖ Extracting a tar archive

- **mv data4 data5.tar:** The file is renamed from data4 to data5.tar, reflecting that it is a tar archive.
 - **tar -xf data5.tar:** The tar -xf command extracts the contents of the tar archive into the current directory. The extracted file is named data5.bin.
 - **file data5.bin:** The file command checks the type of the extracted data5.bin file.

```
data1: gzip compressed data, was "data2.bin", last modified: Thu Sep 19 07:08:15 2024, max compression, from Unix, original size modulo 2^32 574
bandit12@bandit1:~/tmp/.IPLNt6Tpds$ mv data2 data2.gz
bandit12@bandit1:~/tmp/.IPLNt6Tpds$ ls
bandit12@bandit1:~/tmp/.IPLNt6Tpds$ gzip -d data2.gz
bandit12@bandit1:~/tmp/.IPLNt6Tpds$ ls
bandit12@bandit1:~/tmp/.IPLNt6Tpds$ mv data2 data3.gz
bandit12@bandit1:~/tmp/.IPLNt6Tpds$ ls
data3.gz: data.txt
bandit12@bandit1:~/tmp/.IPLNt6Tpds$ gzip -d data3.gz
Command 'gzip2' not found, did you mean:
  command 'bzip2' from deb bzip2 [1.0.8-5.1ubuntu0.1]
  command 'gzip' from deb gzip [1.12~ubuntu1]
Try: apt install <deb name>
bandit12@bandit1:~/tmp/.IPLNt6Tpds$ gzip -d data3.gz
bandit12@bandit1:~/tmp/.IPLNt6Tpds$ ls
data3.gz: data.txt
bandit12@bandit1:~/tmp/.IPLNt6Tpds$ ls
data3.gz: data.txt
bandit12@bandit1:~/tmp/.IPLNt6Tpds$ file data3
data3: cannot open "data3" (No such file or directory)
bandit12@bandit1:~/tmp/.IPLNt6Tpds$ rm data3.gz -d -r -z
bandit12@bandit1:~/tmp/.IPLNt6Tpds$ file data3.br: No such file or directory.
bandit12@bandit1:~/tmp/.IPLNt6Tpds$ mv data3.gz data3.br
bandit12@bandit1:~/tmp/.IPLNt6Tpds$ rm data3.br -d -r -z
bandit12@bandit1:~/tmp/.IPLNt6Tpds$ file data3
data3: gzip compressed data, was "data4.bin", last modified: Thu Sep 19 07:08:15 2024, max compression, from Unix, original size modulo 2^32 20488
bandit12@bandit1:~/tmp/.IPLNt6Tpds$ mv data3 data4.gz
bandit12@bandit1:~/tmp/.IPLNt6Tpds$ ls
data4.gz: data.txt
bandit12@bandit1:~/tmp/.IPLNt6Tpds$ gzip -d data4.gz
bandit12@bandit1:~/tmp/.IPLNt6Tpds$ ls
data4: data.txt
bandit12@bandit1:~/tmp/.IPLNt6Tpds$ file data4
data4: POSIX tar archive (GNU)
bandit12@bandit1:~/tmp/.IPLNt6Tpds$ mv data4 data5.tar
bandit12@bandit1:~/tmp/.IPLNt6Tpds$ tar -xf data5.tar
bandit12@bandit1:~/tmp/.IPLNt6Tpds$ rm data5.tar
Command 'tar' failed: did you mean:
  command 'file' from deb file (1:5.45-2)
  command 'field' from deb rhoel0f (7.2-3build3)
Try: apt install <deb name>
bandit12@bandit1:~/tmp/.IPLNt6Tpds$ file data5.bin
data5.bin: POSIX tar archive (GNU)
bandit12@bandit1:~/tmp/.IPLNt6Tpds$
```

We continue decompressing the file following the same steps until we reach the password.

```
bandit12@bandit:~/tmp$ cat data9
The password is F05dwFsc0cbaiiH0h832eUks2vdTDwAnU
bandit12@bandit:~/tmp$
```

Results: MU4VWeTyJk8ROof1qqmcBPaLh7IDCPvS

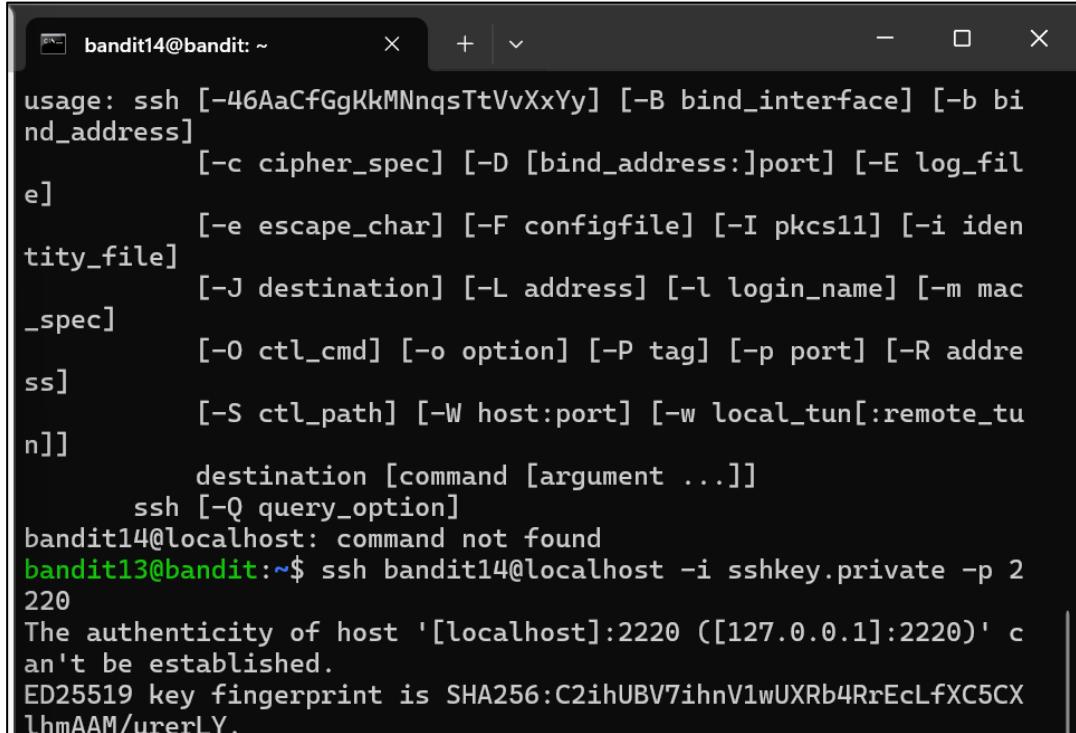
Level 13 → Level 14

Objective

The objective of this level is to find the SSH private key stored in /etc/bandit_pass/bandit14, which can only be read by the user bandit14. With this key, we will be able to connect via SSH on port 2220 using the user bandit14.

Methodology

We will use the command `ssh bandit14@localhost -i sshkey.private -p 2220` to connect to the **bandit14** user from the Bandit server.



```
bandit14@bandit: ~      + | - X
usage: ssh [-46AaCfGgKkMNnqsTtVvXxYy] [-B bind_interface] [-b bind_address]
           [-c cipher_spec] [-D [bind_address:]port] [-E log_file]
           [-e escape_char] [-F configfile] [-I pkcs11] [-i identity_file]
           [-J destination] [-L address] [-l login_name] [-m mac_spec]
           [-O ctl_cmd] [-o option] [-P tag] [-p port] [-R address]
           [-S ctl_path] [-W host:port] [-w local_tun[:remote_tun]]
           destination [command [argument ...]]
           ssh [-Q query_option]
bandit14@localhost: command not found
bandit13@bandit:~$ ssh bandit14@localhost -i sshkey.private -p 2220
The authenticity of host '[localhost]:2220 ([127.0.0.1]:2220)' can't be established.
ED25519 key fingerprint is SHA256:C2ihUBV7ihnV1wUXRb4RrEcLfXC5CX
| lhmAAM/urerLY.
```

- **ssh bandit14@localhost:** With this command, we start an SSH session with the **bandit14** user, and **localhost** indicates that we are connecting to the same IP address to which we are currently connected.
- **-i sshkey.private:** Specifies the private SSH key to be used for authentication. The key is provided in the `sshkey.private` file.
- **-p 2220:** Specifies port 2220, which is used throughout the Bandit game for SSH connections.

Use the **whoami** command to verify that you are logged in as `bandit14` in order to obtain the password. This should return **bandit14**, confirming that you are logged in with the correct user.

```
bandit14@bandit: ~      + | - | X

In addition, the execstack tool can be used to flag the stack
as
executable on ELF binaries.

Finally, network-access is limited for most levels by a local
firewall.

--[ Tools ]--

For your convenience we have installed a few useful tools which
you can find
in the following locations:

* gef (https://github.com/hugsy/gef) in /opt/gef/
* pwndbg (https://github.com/pwndbg/pwndbg) in /opt/pwndbg/
* gdbinit (https://github.com/gdbinit/Gdbinit) in /opt/gdbin
it/
* pwntools (https://github.com/Gallopsled/pwntools)
* radare2 (http://www.radare.org/)

--[ More information ]--

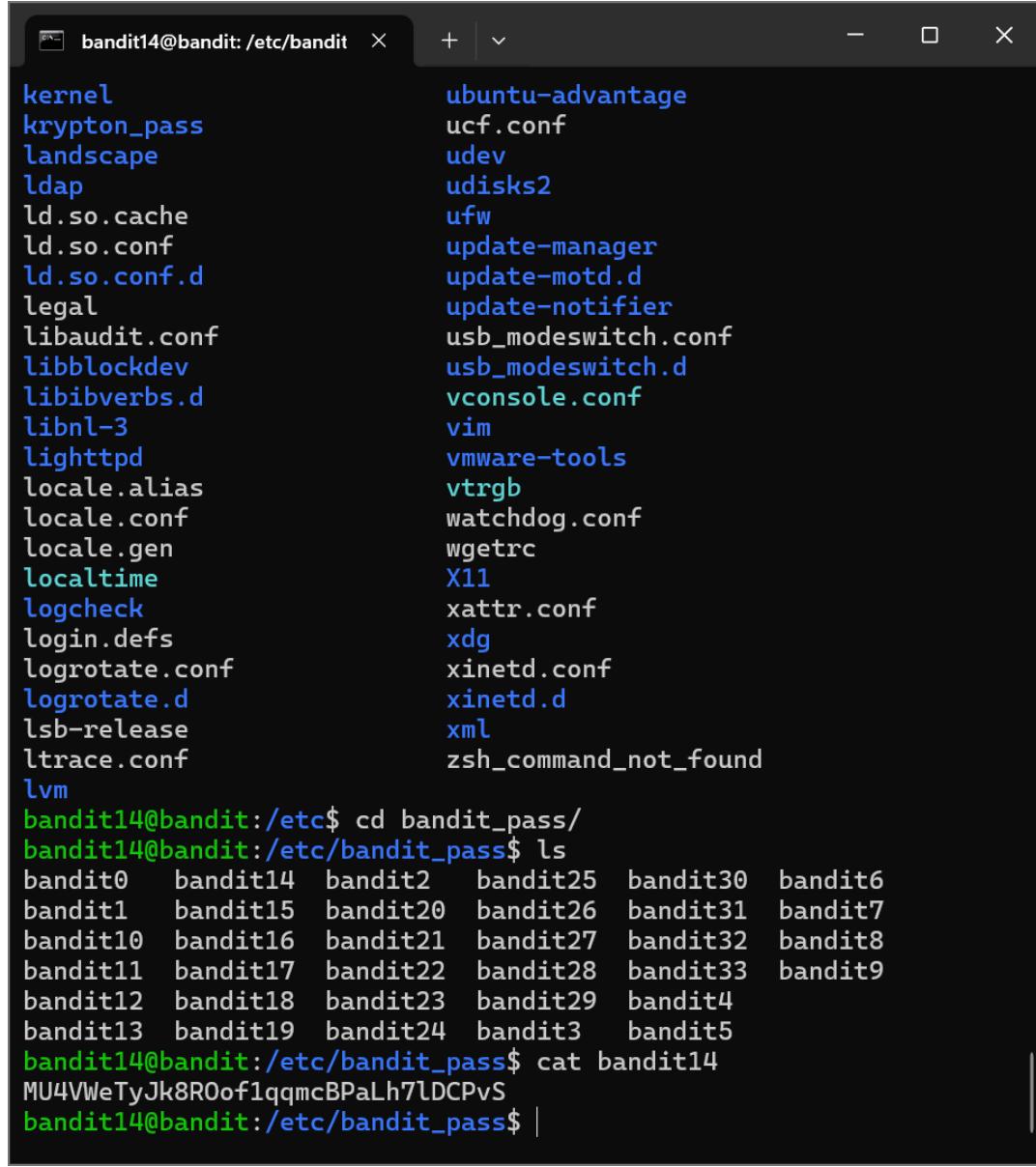
For more information regarding individual wargames, visit
http://www.overthewire.org/wargames/

For support, questions or comments, contact us on discord or I
RC.

Enjoy your stay!

bandit14@bandit:~$ whoami
bandit14
bandit14@bandit:~$ ls
bandit14@bandit:~$ |
```

From the root directory, we will locate the folder containing the password files and access the file for bandit14.



```
bandit14@bandit: /etc/bandit  x  + | - | X
kernel
krypton_pass
landscape
ldap
ld.so.cache
ld.so.conf
ld.so.conf.d
legal
libaudit.conf
libblockdev
libibverbs.d
libnl-3
lighttpd
locale.alias
locale.conf
locale.gen
localtime
logcheck
login.defs
logrotate.conf
logrotate.d
lsb-release
ltrace.conf
lvm
bandit14@bandit:/etc$ cd bandit_pass/
bandit14@bandit:/etc/bandit_pass$ ls
bandit0  bandit14  bandit2  bandit25  bandit30  bandit6
bandit1  bandit15  bandit20  bandit26  bandit31  bandit7
bandit10 bandit16  bandit21  bandit27  bandit32  bandit8
bandit11 bandit17  bandit22  bandit28  bandit33  bandit9
bandit12 bandit18  bandit23  bandit29  bandit4
bandit13 bandit19  bandit24  bandit3  bandit5
bandit14@bandit:/etc/bandit_pass$ cat bandit14
MU4VWeTyJk8R0of1qqmcBPaLh7lDCPvS
bandit14@bandit:/etc/bandit_pass$ |
```

Results: 8xCjnmgoKbGLhHFAZlGE5Tmu4M2tKJQo

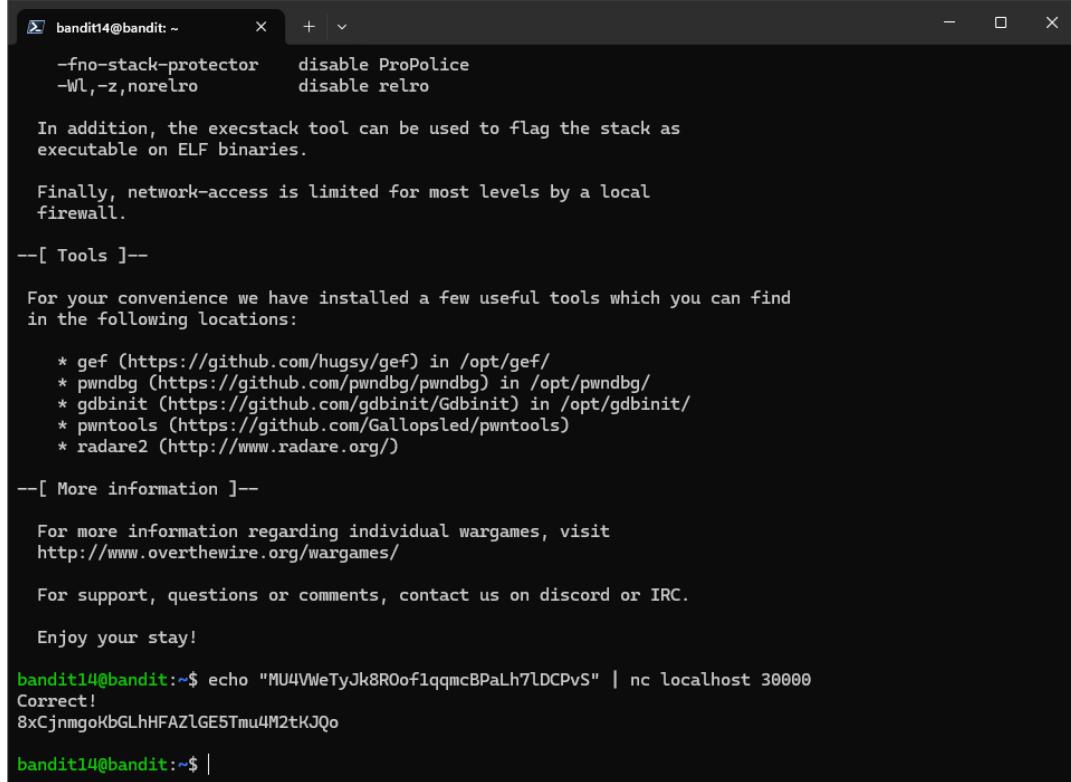
Level 14 → Level 15

Objective

The objective of this level is to send the password from the previous level through port 30000 in order to obtain the password that will allow us to connect via SSH on port 2220 with the user **bandit15**.

Methodology

We will use the command echo "MU4VWeTyJk8ROof1qqmcBPaLh7lDCPvS" | nc localhost 30000 to send the previous password to the server.



```

bandit14@bandit: ~      + - x
-fno-stack-protector    disable ProPolice
-WL,-z,noexecro        disable relro

In addition, the execstack tool can be used to flag the stack as
executable on ELF binaries.

Finally, network-access is limited for most levels by a local
firewall.

--[ Tools ]--

For your convenience we have installed a few useful tools which you can find
in the following locations:

* gef (https://github.com/hugsy/gef) in /opt/gef/
* pwndbg (https://github.com/pwndbg/pwndbg) in /opt/pwndbg/
* gdbinit (https://github.com/gdbinit/Gdbinit) in /opt/gdbinit/
* pwntools (https://github.com/Gallopsled/pwntools)
* radare2 (http://www.radare.org/)

--[ More information ]--

For more information regarding individual wargames, visit
http://www.overthewire.org/wargames/

For support, questions or comments, contact us on discord or IRC.

Enjoy your stay!

bandit14@bandit:~$ echo "MU4VWeTyJk8ROof1qqmcBPaLh7lDCPvS" | nc localhost 30000
Correct!
8xCjnmgoKbGLhHFAZlGE5Tmu4M2tKJQo

bandit14@bandit:~$ |

```

- ***echo "MU4VWeTyJk8ROof1qqmcBPaLh7lDCPvS"***: Prints the text to the standard output.
- **/**: Takes the output of the echo command and sends it as input to the next command (nc)
- ***nc localhost 30000***: (Netcat) is a tool for sending and receiving data over a network. In this case, the connection will be made to the same machine where the command is being executed, and it will connect to the specified port on the server.

Results: 8xCjnmgoKbGLhHFAZlGE5Tmu4M2tKJQo

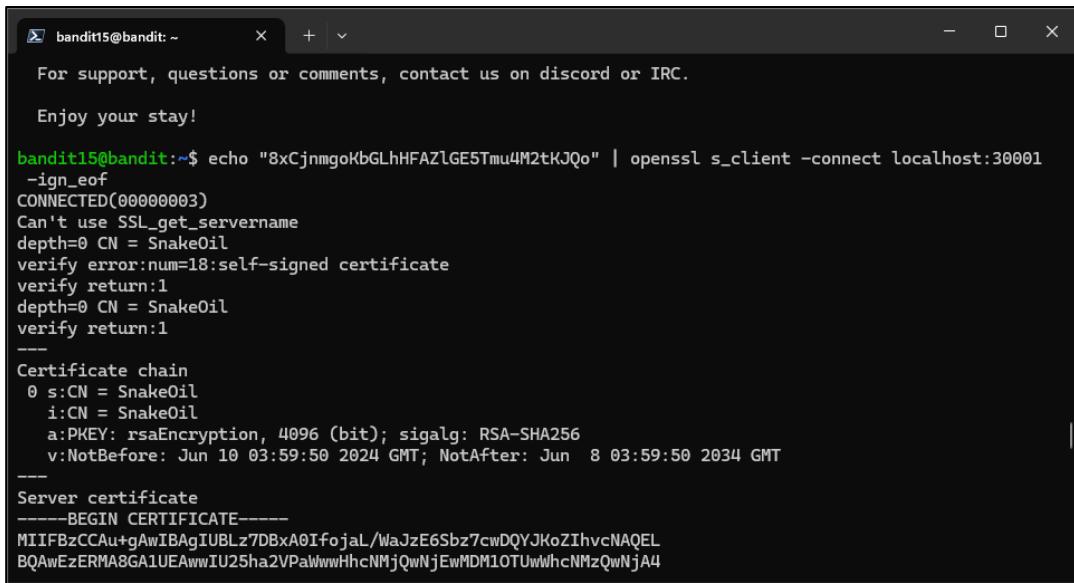
Level 15 → Level 16

Objective

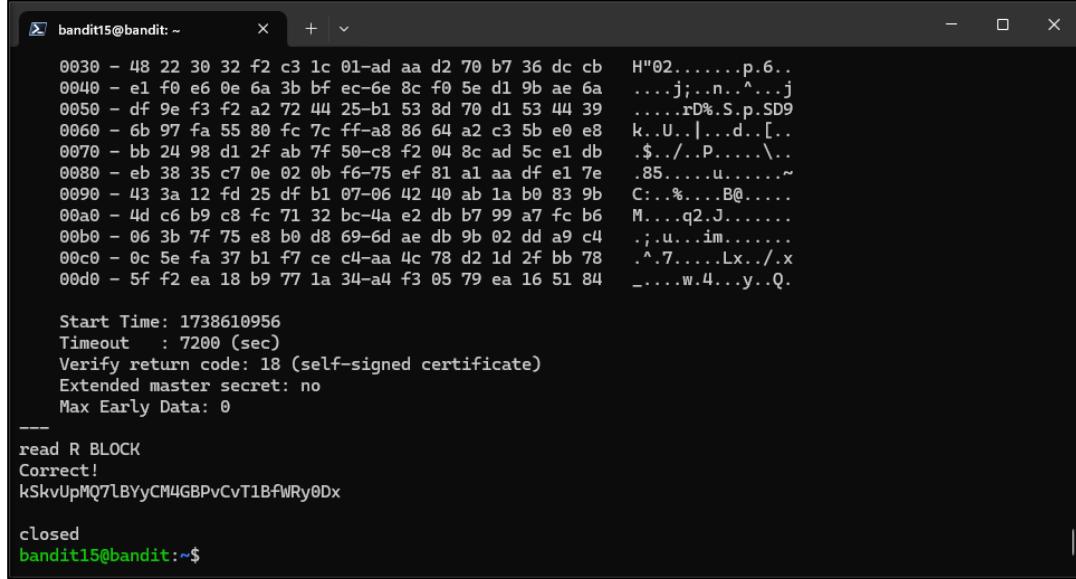
The objective of this level is to send the password from the previous level encrypted in SSL/TLS through port 30001, in order to obtain the password for the next level and use it to connect via SSH on port 2220 with the user bandit12.

Methodology

For this exercise, we will use part of the previous command to send the password obtained in the previous level to the server, but after the |, the command we will use is openssl s_client -connect localhost:30001.



```
bandit15@bandit:~$ echo "8xCjnmgoKbGLhHFAZlGE5Tmu4M2tKJQo" | openssl s_client -connect localhost:30001
-ign_eof
CONNECTED(00000003)
Can't use SSL_get_servername
depth=0 CN = SnakeOil
verify error:num=18:self-signed certificate
verify return:1
depth=0 CN = SnakeOil
verify return:1
-----
Certificate chain
0 s:CN = SnakeOil
    i:CN = SnakeOil
        a:PKEY: rsaEncryption, 4096 (bit); sigalg: RSA-SHA256
        v:NotBefore: Jun 10 03:59:50 2024 GMT; NotAfter: Jun  8 03:59:50 2034 GMT
-----
Server certificate
-----BEGIN CERTIFICATE-----
MIIFBzCCAu+gAwIBAgIUBLz7DBxA0Ifojal/WaJzE6Sbz7cwDQYJKoZIhvcNAQEL
BQAwEzERMA8GA1UEAwIU25ha2VPaWwwHhcNMjQwNjEwMDM1OTUwWhcNMzQwNjA4
```



```

bandit15@bandit: ~
0030 - 48 22 30 32 f2 c3 1c 01-ad aa d2 70 b7 36 dc cb H"02.....p.6..
0040 - e1 f0 e6 0e 6a 3b bf ec-6e 8c f0 5e d1 9b ae 6a ....j;..n..^....j
0050 - df 9e f3 f2 a2 72 44 25-b1 53 8d 70 d1 53 44 39 .....rD%.S.p.SD9
0060 - 6b 97 fa 55 80 fc 7c ff-a8 86 64 a2 c3 5b e0 e8 k..U..|...d..[..
0070 - bb 24 98 d1 2f ab 7f 50-c8 f2 04 8c ad 5c e1 db $.../..P....\..
0080 - eb 38 35 c7 0e 02 0b f6-75 ef 81 a1 aa df e1 7e .85.....u.....~
0090 - 43 3a 12 fd 25 df b1 07-06 42 40 ab 1a b0 83 9b C:..%....B@.....
00a0 - 4d c6 b9 c8 fc 71 32 bc-4a e2 db b7 99 a7 fc b6 M....q2.J.....
00b0 - 06 3b 7f 75 e8 b0 d8 69-6d ae db 9b 02 dd a9 c4 .;.u...im.....
00c0 - 0c 5e fa 37 b1 f7 ce c4-aa 4c 78 d2 1d 2f bb 78 .^....Lx.../..x
00d0 - 5f f2 ea 18 b9 77 1a 34-a4 f3 05 79 ea 16 51 84 .....w.4...y..Q.

Start Time: 1738610956
Timeout : 7200 (sec)
Verify return code: 18 (self-signed certificate)
Extended master secret: no
Max Early Data: 0
-----
read R BLOCK
Correct!
kSkvUpMQ7lBYyCM4GBPvCvT1BfWRy0Dx

closed
bandit15@bandit: ~$
```

- openssl: A cryptography and security command.
- s_client: An OpenSSL subcommand that acts as a TLS/SSL client to test secure connections.
- -connect localhost:30001: Specifies the server and port to connect to.

Results: kSkvUpMQ7lBYyCM4GBPvCvT1BfWRy0Dx

Level 16 → Level 17

Objective

The objective of this level is to obtain the password for the next level by sending the current level's password to the server through one of the ports ranging from 31000 to 32000 in order to connect via SSH on port 2220 with the user bandit17.

Methodology

To complete level 16 of Bandit, we first scan the port range using nmap to identify which ones are open. We find that five ports are active, including 31518 and 31790, which use SSL. We decide to work with port 31790 and execute a command similar to the one from level 14: echo "password" | openssl s_client -connect localhost:31790 -ign_eof.

When we run the command, we obtain the server certificate. We copy it and create a directory where we store this information in a file, granting it read and write permissions. Then, we use this information to establish the SSH connection to the next level. Finally, we access the password using the command cat /etc/bandit_pass/bandit17.

```
bandit16@bandit:~$ nmap -p 31000-32000 -sV localhost
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-26 02:39 UTC
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00017s latency).
Not shown: 996 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
31006/tcp open  echo
31518/tcp open  echo
31691/tcp open  echo
31799/tcp open  ssl/unknown
31960/tcp open  echo
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-Port31798-TCP:V=7.94SVN|T=SSL|I=7D=8/26%time=66CBEEFFP=x86_64-pc-linu
SF:-gnu%%(GenericLines,32,"Wrong!%x20Please%x20enter%x20the\x20correct\x2
SF:@current%x20password.\n")%r(GetRequest,32,"Wrong!%x20Please%x20enter\x
SF:@the\x20correct\x20current\x20password.\n")%r(HTTPOptions,32,"Wrong!\n
SF:@28PLease%x20enter%x20the\x20correct\x20current\x20password.\n")%r(RTS
SF:PRequest,32,"Wrong!%x20Please%x20enter%x20the\x20correct\x20current\x20
SF:@password.\n")%rHelp,32,"Wrong!%x20Please%x20enter%x20the\x20correct\x
SF:@current%x20password.\n")%r(FourOhFourRequest,32,"Wrong!%x20Please%x2
SF:@enter%x20the\x20correct\x20current\x20password.\n")%r(LPDString,32,"W
SF:rong!%x20Please%x20enter%x20the\x20correct\x20current\x20password.\n")
SF:@SIPOptions,32,"Wrong!%x20Please%x20enter%x20the\x20correct\x20current\x20
SF:t\x20password.\n");
```

Service detection performed. Please report any incorrect results at <https://nmap.org/submit/>.

Nmap done: 1 IP address (1 host up) scanned in 133.23 seconds

```
bandit16@bandit:~$ echo kSkvUpMQ7lBYyCM4GBPvCvT1BfwRy0Dx | openssl s_client -connect localhost:31790 -ign_eof
CONNECTED(00000003)
Can't use SSL_get_servername
depth=0 CN = SnakeOil
verify error:num=18:self-signed certificate
verify return:1
depth=0 CN = SnakeOil
verify return:1
---
Certificate chain
0 s:CN = SnakeOil
i:CN = SnakeOil
a:KEY: rsaEncryption, 4096 (bit); sigalg: RSA-SHA256
v:NotBefore: Jun 10 03:59:50 2024 GMT; NotAfter: Jun  8 03:59:50 2034 GMT
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIFBzCCAUwIBAgIUBLz7DBxA0Ifojal/WaJzE65bz7cwDQYJk0ZIhvcNAQEL
BQAwEzERMA8GA1UEAwIU25ha2VPaWwHhcNMjQwNjEwMDM1OTUwWhcNMzQwNjA4
MDM1OTUwlujATMRewDyDVQQDahTbmFrZU9pbDBCAiIwDQYJKoZhvcNAQEBBQAD
ggIPADCCAg0CggIBANI+P50Xm9Bj21F1Ps0qbqZRB5XmSZZJYam7EIJ16Fxed+f
jXAv4d/FVqiEM4BuSNsNMbMx2GqqlAfN33h+RMtjRoMbByBsZsC63MLfXCk4p+
09gtGP7BS6Iy5Xdmfy/YPhvA3JDEScdLDmd6Lsbdwhv93Q8M6POVO9sv4HuS4+/
jEj+rNhE+Bjz/wubyg7GL71BP1WPZpQnRE4OzoSrt5+bZVLvdDUWUwinB0flaGRk
GmIOr5EU0Ud7HpyoIQuiNlePGfpHRKnmXTTEZEoxeWWAaM1VhPGqfxB/Pncat+
vAJX71B0b3khnmfV0Scsg/YAUR94wSELy+UEWjaELVUntJ5heRdiTch1VQ++w
wnnjNbepaw6shopbyUF3XXfhIb4NvwLWpvokFXVtcVjl0ujF0snVvpE+MRT0wacy
tHjtZs7Ao7GYx0Dz6H8AdBLKJW67uOn37aM1260ADFM5+2vEAbNSF+f6i15mzB
18cY64Za6oUbjGK7BArDx56Brc3WFyuB1GWAfHeuB948BcsfhXY7daf5jjzPmgz
mq1zdRthQB31M0M21i6vuTkheAvKFFF+1LH4M9SnE54NSF2hj9Nhgq9V08fhYc
x0W6qu+S8HUhVF+V23yTuNgz4Q+UoGs4sHSDeSIBFqNv1nnpUntNgcR2L5PAgMB
AAGjUzBRNB0GA1UdqQWBTPo8kfze4p9EgxNuYk7+xDGftAYZafBgNVHSMEGDAW
gbTPo8kfze4p9EgxNuYk7+xDGftAYzAPBgnVHRMBAf8EBETADAQH/Ma0GCSqS1b3
DQEBCwUA4ICAQAKHomtmcGyilnhzile97Mq2+Sul5qgVWwf/x/KYOxv2T8ZmcR
Ae9xFhZT4jsAOUDK10Xx9aZgDGJHJLNEVTe9zlw10NFfNxExBxQgp7hhmDBWdtj6d
taqEW/Jp06X+08BtNtYK9NzsvDg2YRcvOHConemjvvEL7tQK0m+GVyfQlyg6jnrx
egH+abucTxbabFcWSE+vK0uJYmqcbxB4WNk29v4jV5hn7/DN4x1jfko+nREw60a
```

```
bandit16@bandit:/tmp/bandit17$ ssh -i sshkey.private bandit17@localhost -p 2220
The authenticity of host '[localhost]:2220 ([127.0.0.1]:2220)' can't be established.
ED25519 key fingerprint is SHA256:C2ihUBV7ihnVlwUXRb4RrEcLFXC5CXlhmAAM/urerLY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Could not create directory '/home/bandit16/.ssh' (Permission denied).
Failed to add the host to the list of known hosts (/home/bandit16/.ssh/known_hosts).
```



This is an OverTheWire game server.
More information on <http://www.overthewire.org/wargames>

```
!!! You are trying to log into this SSH server with a password on port 2220 from localhost.
!!! Connecting from localhost is blocked to conserve resources.
!!! Please log out and log in again.
```



Results: EReVavePLFHtFlFsjn3hyzMlvSuSACRD

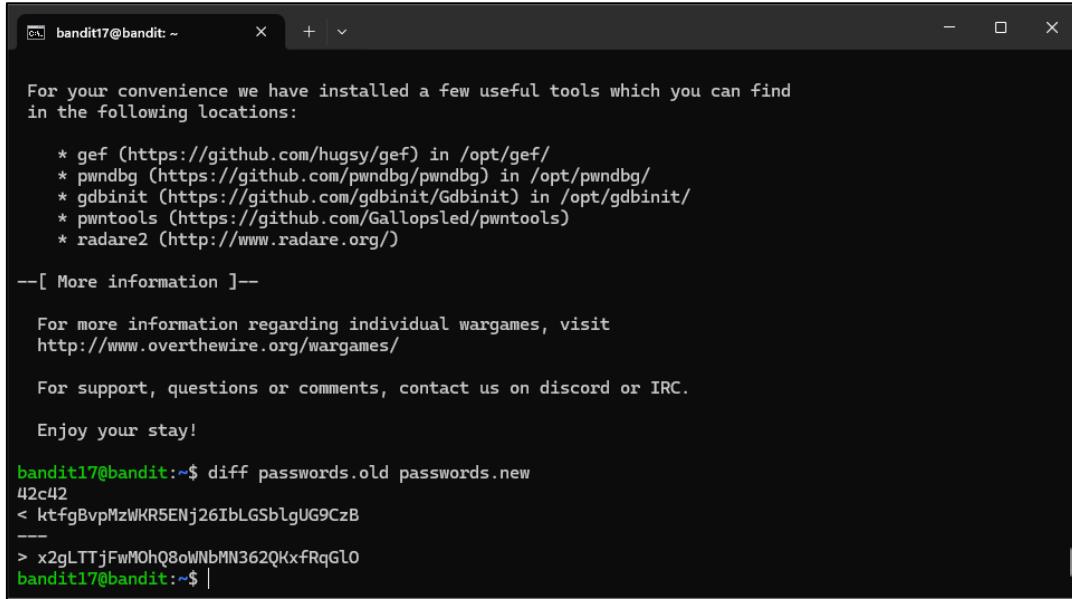
Level 17 → Level 18

Objective

The objective of this level is to find the only line that was changed between the two files (password.old and password.new), which will be the password to connect via SSH on port 2220 with the user bandit18.

Methodology

We will need to compare the two files using the command *diff passwords.new passwords.old*.



For your convenience we have installed a few useful tools which you can find in the following locations:

- * gef (<https://github.com/hugsy/gef>) in /opt/gef/
- * pwndbg (<https://github.com/pwndbg/pwndbg>) in /opt/pwndbg/
- * gdbinit (<https://github.com/gdbinit/Gdbinit>) in /opt/gdbinit/
- * pwntools (<https://github.com/Gallopsled/pwntools>)
- * radare2 (<http://www.radare.org/>)

--[More information]--

For more information regarding individual wargames, visit
<http://www.overthewire.org/wargames/>

For support, questions or comments, contact us on discord or IRC.

Enjoy your stay!

```
bandit17@bandit:~$ diff passwords.old passwords.new
42c42
< ktfgBvpMzWKR5ENj26IbLGSblgUG9CzB
---
> x2gLTTjFwMOhQ8oWNbMN362QKxfRqGIO
bandit17@bandit:~$ |
```

- diff: This command shows the differences between two files line by line.
- passwords.new: This is the first file in the comparison.
- passwords.old: This is the second file in the comparison.

Results: x2gLTTjFwMOhQ8oWNbMN362QKxfRqGIO

Level 18 → Level 19

Objective

The objective of this level is to retrieve the password for the next level, which is stored in a file named "readme" located in the home directory. However, when attempting to log in normally via SSH, the session is automatically closed due to a modification in the .bash file.

Methodology

When logging in as we have done in previous levels, after entering the password, a message appears informing us that the connection has been closed. We can solve this by connecting to the server using the command: `ssh bandit18@bandit.labs.overthewire.org -p 2220 "cat ~/readme"`, which will give us the password for the next level.

```
Command Prompt + - X
--[ Tools ]--
For your convenience we have installed a few useful tools which you can find
in the following locations:
* gef (https://github.com/hugsy/gef) in /opt/gef/
* pwndbg (https://github.com/pwndbg/pwndbg) in /opt/pwndbg/
* gdbinit (https://github.com/gdbinit/Gdbinit) in /opt/gdbinit/
* pwntools (https://github.com/Gallopsled/pwntools)
* radare2 (http://www.radare.org/)
--[ More information ]--
For more information regarding individual wargames, visit
http://www.overthewire.org/wargames/
For support, questions or comments, contact us on discord or IRC.
Enjoy your stay!
Byebye !
Connection to bandit.labs.overthewire.org closed.

C:\Users\Redes>
```

- **ssh:** It is the command to establish a secure connection to a remote server via SSH.
 - **bandit18@bandit.labs.overthewire.org:** Specifies the user and the server you are connecting to. In this case, the user is bandit18 and the server is bandit.labs.overthewire.org.
 - **-p 2220:** Indicates that the connection should be made through port 2220.
 - **"cat ~/readme":** It is the command that will be executed on the server after the connection is established. cat is used to display the content of the readme file located in the user's home directory (represented by ~).

Results: cGWpMaKXVwDUNgPAVJbWYuGHVn9zl3j8

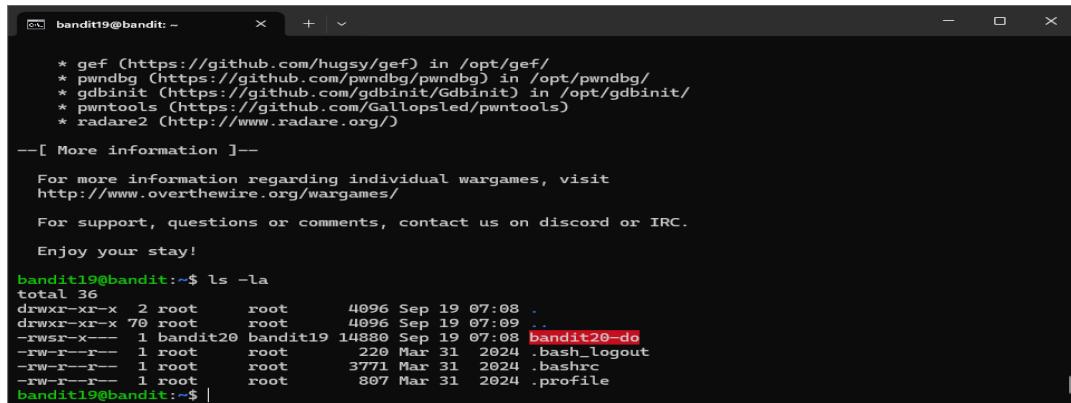
Level 19 → Level 20

Objective

The objective of this level is to find a way to correctly execute the setuid binary to gain access to the password for the next level. This binary has special configurations that allow certain operations to be executed with elevated privileges, which will enable you to access the password stored in the `/etc/bandit_pass` file.

Methodology

First, we need to list the files and directories, showing additional information, including hidden files. Then, with the id command, we display the information about the current user and the groups the user belongs to. Once we have our file `bandit20-do` highlighted previously, we execute this file in the current directory and pass the id as an argument to that file. This way, we will be able to read the password in the specific directory.



The screenshot shows a terminal window with the following content:

```
* gef (https://github.com/hugsy/gef) in /opt/gef/
* pwndbg (https://github.com/pwndbg/pwndbg) in /opt/pwndbg/
* gdbinit (https://github.com/gdbinit/Gdbinit) in /opt/gdbinit/
* pwntools (https://github.com/Gallopsled/pwntools)
* radare2 (http://www.radare.org/)

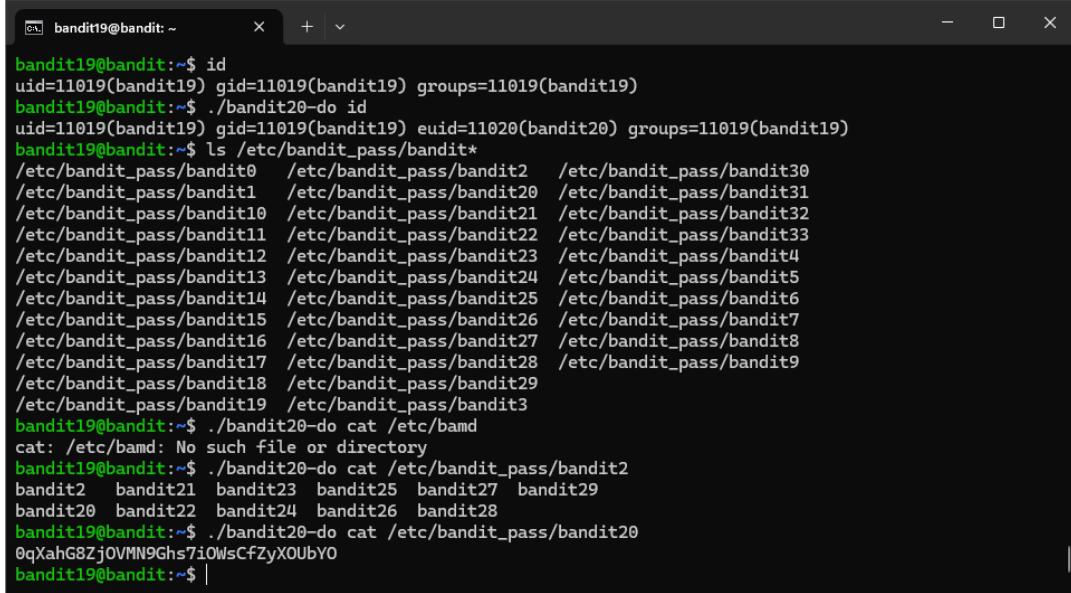
--[ More information ]--

For more information regarding individual wargames, visit
http://www.overthewire.org/wargames/

For support, questions or comments, contact us on discord or IRC.

Enjoy your stay!

bandit19@bandit:~$ ls -la
total 36
drwxr-xr-x  2 root      root      4096 Sep 19 07:08 .
drwxr-xr-x 70 root      root      4096 Sep 19 07:09 ..
-rwsr-x---  1 bandit20 bandit19 14880 Sep 19 07:08 bandit20-do
-rw-r--r--  1 root      root      220 Mar 31 2024 .bash_logout
-rw-r--r--  1 root      root     3771 Mar 31 2024 .bashrc
-rw-r--r--  1 root      root      807 Mar 31 2024 .profile
bandit19@bandit:~$ |
```



```

bandit19@bandit:~$ id
uid=11019(bandit19) gid=11019(bandit19) groups=11019(bandit19)
bandit19@bandit:~$ ./bandit20-do id
uid=11019(bandit19) gid=11019(bandit19) euid=11020(bandit20) groups=11019(bandit19)
bandit19@bandit:~$ ls /etc/bandit_pass/bandit*
/etc/bandit_pass/bandit0  /etc/bandit_pass/bandit2  /etc/bandit_pass/bandit30
/etc/bandit_pass/bandit1  /etc/bandit_pass/bandit20  /etc/bandit_pass/bandit31
/etc/bandit_pass/bandit10 /etc/bandit_pass/bandit21  /etc/bandit_pass/bandit32
/etc/bandit_pass/bandit11 /etc/bandit_pass/bandit22  /etc/bandit_pass/bandit33
/etc/bandit_pass/bandit12 /etc/bandit_pass/bandit23  /etc/bandit_pass/bandit4
/etc/bandit_pass/bandit13 /etc/bandit_pass/bandit24  /etc/bandit_pass/bandit5
/etc/bandit_pass/bandit14 /etc/bandit_pass/bandit25  /etc/bandit_pass/bandit6
/etc/bandit_pass/bandit15 /etc/bandit_pass/bandit26  /etc/bandit_pass/bandit7
/etc/bandit_pass/bandit16 /etc/bandit_pass/bandit27  /etc/bandit_pass/bandit8
/etc/bandit_pass/bandit17 /etc/bandit_pass/bandit28  /etc/bandit_pass/bandit9
/etc/bandit_pass/bandit18 /etc/bandit_pass/bandit29
/etc/bandit_pass/bandit19 /etc/bandit_pass/bandit3
bandit19@bandit:~$ ./bandit20-do cat /etc/bamd
cat: /etc/bamd: No such file or directory
bandit19@bandit:~$ ./bandit20-do cat /etc/bandit_pass/bandit2
bandit2  bandit21  bandit23  bandit25  bandit27  bandit29
bandit20  bandit22  bandit24  bandit26  bandit28
bandit19@bandit:~$ ./bandit20-do cat /etc/bandit_pass/bandit20
0qXahG8ZjOVMN9Ghs7iOWsCfZyXOUbYO
bandit19@bandit:~$ |

```

- **ls -al:** Displays the file permissions, where the s bit indicates that the binary runs with the permissions of bandit20.
- **./bandit20-do id:** Displays the user and group IDs, showing that the binary runs with the permissions of bandit20.
- **./bandit20-do cat /etc/bandit_pass/bandit20:** Runs cat through the setuid binary to read the restricted password file.

Results: 0qXahG8ZjOVMN9Ghs7iOWsCfZyXOUbYO

Level 20 → Level 21

Objective

The objective of this level is to use a setuid binary in the home directory that connects to localhost on a port specified as a command-line argument. The binary reads a line of text from the connection and compares it with the password from the previous level (bandit20). If the password is correct, the binary will transmit the password for the next level (bandit21).

Methodology

We will use the command **echo password | nc -l 61337 &** to start the local network server and listen on a specific port, simulating the network service to which the binary will connect. Next, we will ensure that the setuid binary can establish the connection by using the command to view all running processes, confirming that the

service is active on port 61337. The setuid binary is used to connect to the local network service and retrieve the password for the next level. By executing **./suconnect 61337**, the binary connects to port 61337, sends the current password, and if correct, displays the password for the next level.

```
bandit20@bandit:~ * radare2 (http://www.radare.org/)

---[ More information ]---

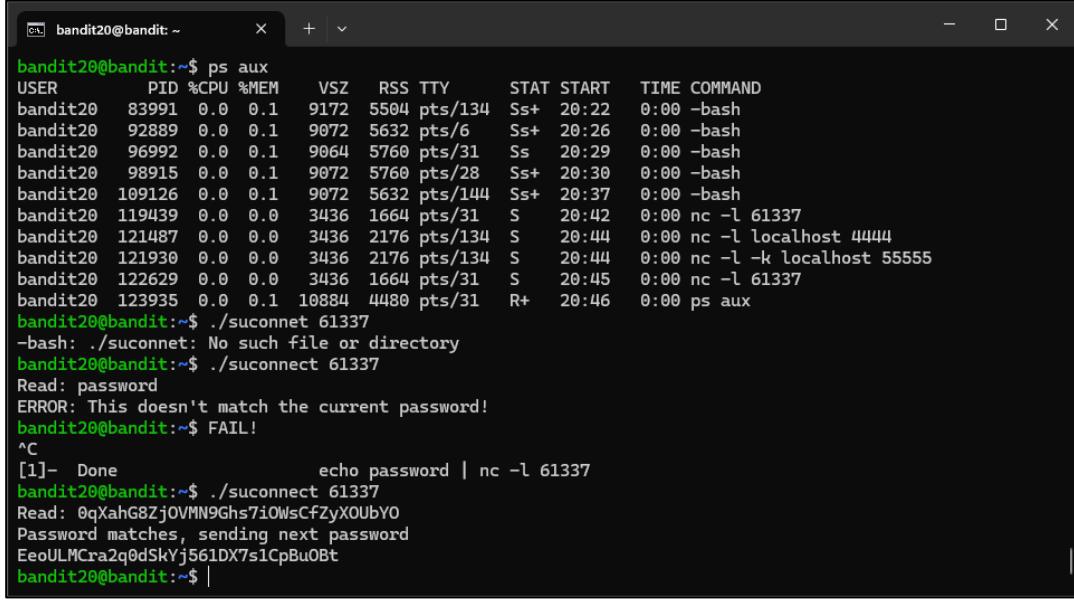
For more information regarding individual wargames, visit
http://www.overthewire.org/wargames/

For support, questions or comments, contact us on discord or IRC.

Enjoy your stay!

bandit20@bandit:~$ echo password | nc -l 61337 &
[1] 119439
bandit20@bandit:~$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START  TIME COMMAND
bandit20  83991  0.0  0.1   9172  5504 pts/134 Ss+ 20:22  0:00 -bash
bandit20  92889  0.0  0.1   9072  5632 pts/6  Ss+ 20:26  0:00 -bash
bandit20  96992  0.0  0.1   9064  5760 pts/31  Ss 20:29  0:00 -bash
bandit20  98915  0.0  0.1   9072  5760 pts/28  Ss+ 20:30  0:00 -bash
bandit20 109126  0.0  0.1   9072  5632 pts/144 Ss+ 20:37  0:00 -bash
bandit20 119439  0.0  0.0   3436  1664 pts/31   S 20:42  0:00 nc -l 61337
bandit20 120734  0.0  0.1  10884  4480 pts/31  R+ 20:43  0:00 ps aux
bandit20@bandit:~$ echo 0qXahG8Zj0VMN9Ghs7i0WsCfZyX0UbYO | nc -l 61337 &
[2] 122629
bandit20@bandit:~$ |
```

```
[1] 119439
bandit20@bandit:~$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START  TIME COMMAND
bandit20  83991  0.0  0.1   9172  5504 pts/134 Ss+ 20:22  0:00 -bash
bandit20  92889  0.0  0.1   9072  5632 pts/6  Ss+ 20:26  0:00 -bash
bandit20  96992  0.0  0.1   9064  5760 pts/31  Ss 20:29  0:00 -bash
bandit20  98915  0.0  0.1   9072  5760 pts/28  Ss+ 20:30  0:00 -bash
bandit20 109126  0.0  0.1   9072  5632 pts/144 Ss+ 20:37  0:00 -bash
bandit20 119439  0.0  0.0   3436  1664 pts/31   S 20:42  0:00 nc -l 61337
bandit20 120734  0.0  0.1  10884  4480 pts/31  R+ 20:43  0:00 ps aux
bandit20@bandit:~$ echo 0qXahG8Zj0VMN9Ghs7i0WsCfZyX0UbYO | nc -l 61337 &
[2] 122629
bandit20@bandit:~$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START  TIME COMMAND
bandit20  83991  0.0  0.1   9172  5504 pts/134 Ss+ 20:22  0:00 -bash
bandit20  92889  0.0  0.1   9072  5632 pts/6  Ss+ 20:26  0:00 -bash
bandit20  96992  0.0  0.1   9064  5760 pts/31  Ss 20:29  0:00 -bash
bandit20  98915  0.0  0.1   9072  5760 pts/28  Ss+ 20:30  0:00 -bash
bandit20 109126  0.0  0.1   9072  5632 pts/144 Ss+ 20:37  0:00 -bash
bandit20 119439  0.0  0.0   3436  1664 pts/31   S 20:42  0:00 nc -l 61337
bandit20 121487  0.0  0.0   3436  2176 pts/134  S 20:44  0:00 nc -l localhost 4444
bandit20 121930  0.0  0.0   3436  2176 pts/134  S 20:44  0:00 nc -l -k localhost 55555
bandit20 122629  0.0  0.0   3436  1664 pts/31   S 20:45  0:00 nc -l 61337
bandit20 123935  0.0  0.1  10884  4480 pts/31  R+ 20:46  0:00 ps aux
bandit20@bandit:~$ |
```



```

bandit20@bandit:~$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START  TIME COMMAND
bandit20  83991  0.0  0.1   9172  5504 pts/134  Ss+  20:22  0:00 -bash
bandit20  92889  0.0  0.1   9072  5632 pts/6   Ss+  20:26  0:00 -bash
bandit20  96992  0.0  0.1   9064  5760 pts/31  Ss  20:29  0:00 -bash
bandit20  98915  0.0  0.1   9072  5760 pts/28  Ss+  20:30  0:00 -bash
bandit20 109126  0.0  0.1   9072  5632 pts/144 Ss+  20:37  0:00 -bash
bandit20 119439  0.0  0.0   3436  1664 pts/31   S  20:42  0:00 nc -l 61337
bandit20 121487  0.0  0.0   3436  2176 pts/134   S  20:44  0:00 nc -l localhost 4444
bandit20 121930  0.0  0.0   3436  2176 pts/134   S  20:44  0:00 nc -l -k localhost 55555
bandit20 122629  0.0  0.0   3436  1664 pts/31   S  20:45  0:00 nc -l 61337
bandit20 123935  0.0  0.1  10884  4480 pts/31  R+  20:46  0:00 ps aux
bandit20@bandit:~$ ./suconnect 61337
-bash: ./suconnect: No such file or directory
bandit20@bandit:~$ ./suconnect 61337
Read: password
ERROR: This doesn't match the current password!
bandit20@bandit:~$ FAIL!
^C
[1]- Done                      echo password | nc -l 61337
bandit20@bandit:~$ ./suconnect 61337
Read: 0qXahG8Zj0VMN9Ghs7i0WsCfZyX0Ub0
Password matches, sending next password
EeoULMCra2q0dSkYj561DX7s1CpBuOBt
bandit20@bandit:~$ |

```

- **nc -l 61337:** Passes the output to the nc (netcat) command, which starts a network listener on port 61337.
- **&:** Runs the network listener in the background, allowing the terminal to be used for other commands.
- **ps aux:** Displays all running processes.
- **./suconnect 61337:** Executes the setuid binary suconnect with the port number 61337 as an argument.

Results: EeoULMCra2q0dSkYj561DX7s1CpBuOBt

Level 21 → Level 22

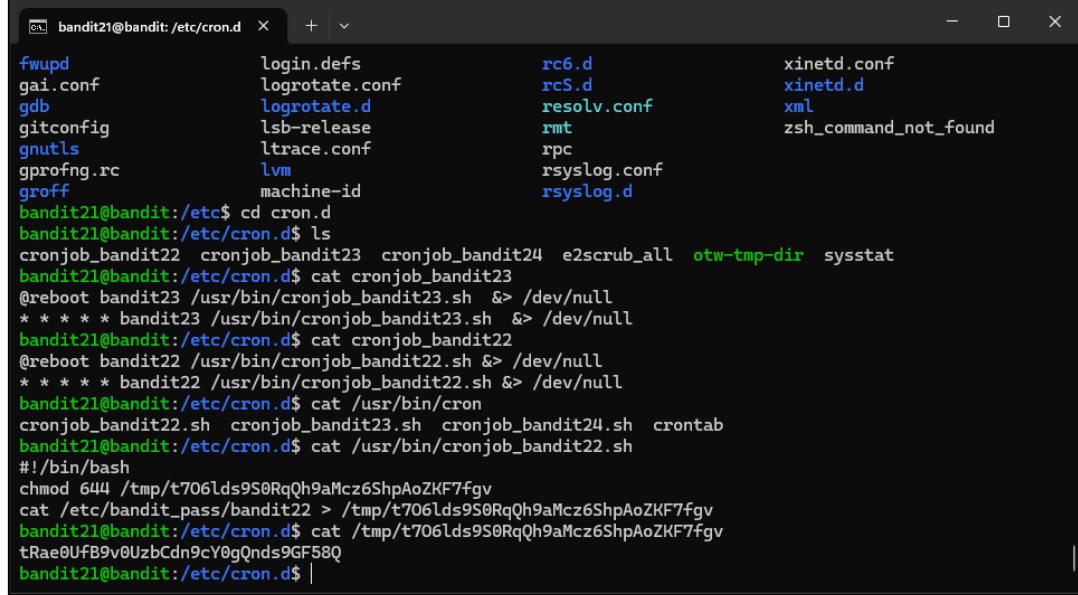
Objective

The objective of this level is to identify the command that is being executed automatically at regular intervals by cron, the time-based job scheduler. To achieve this, it is necessary to examine the configuration in the **/etc/cron.d/** directory and find the specific command that is being executed.

Methodology

You need to identify a cron job that runs automatically on the system. Start by listing and accessing the **/etc/cron.d/** directory, where cron jobs are configured. Then, inspect the **cronjob_bandit** file, which defines a specific cron job that

executes a script located at **/usr/bin/cronjob_bandit23.sh**. By reviewing the contents of this script, you can see that it generates an MD5 hash based on a specific string. This hash is the key, and it is stored in **/tmp/**



```

bandit21@bandit: /etc/cron.d ~ + - x
fwupd          login.defs          rc6.d           xinetd.conf
gai.conf       logrotate.conf     rcS.d           xinetd.d
gdb           logrotate.d        resolv.conf    xml
gitconfig      lsb-release       rmt
gnutls         ltrace.conf      rpc
gprofng.rc     lvm             rsyslog.conf
groff         machine-id      rsyslog.d
bandit21@bandit:/etc$ cd cron.d
bandit21@bandit:/etc/cron.d$ ls
cronjob_bandit22 cronjob_bandit23 cronjob_bandit24 e2scrub_all otw-tmp-dir sysstat
bandit21@bandit:/etc/cron.d$ cat cronjob_bandit23
@reboot bandit23 /usr/bin/cronjob_bandit23.sh &> /dev/null
* * * * * bandit23 /usr/bin/cronjob_bandit23.sh &> /dev/null
bandit21@bandit:/etc/cron.d$ cat cronjob_bandit22
@reboot bandit22 /usr/bin/cronjob_bandit22.sh &> /dev/null
* * * * * bandit22 /usr/bin/cronjob_bandit22.sh &> /dev/null
bandit21@bandit:/etc/cron.d$ cat /usr/bin/cron
cronjob_bandit22.sh cronjob_bandit23.sh cronjob_bandit24.sh crontab
bandit21@bandit:/etc/cron.d$ cat /usr/bin/cronjob_bandit22.sh
#!/bin/bash
chmod 644 /tmp/t706lds9S0RqQh9aMcz6ShpAoZKF7fgv
cat /etc/bandit_pass/bandit22 > /tmp/t706lds9S0RqQh9aMcz6ShpAoZKF7fgv
bandit21@bandit:/etc/cron.d$ cat /tmp/t706lds9S0RqQh9aMcz6ShpAoZKF7fgv
tRae0UfB9v0UzbCdn9cY0gQnds9GF58Q
bandit21@bandit:/etc/cron.d$ |

```

Results: tRae0UfB9v0UzbCdn9cY0gQnds9GF58Q

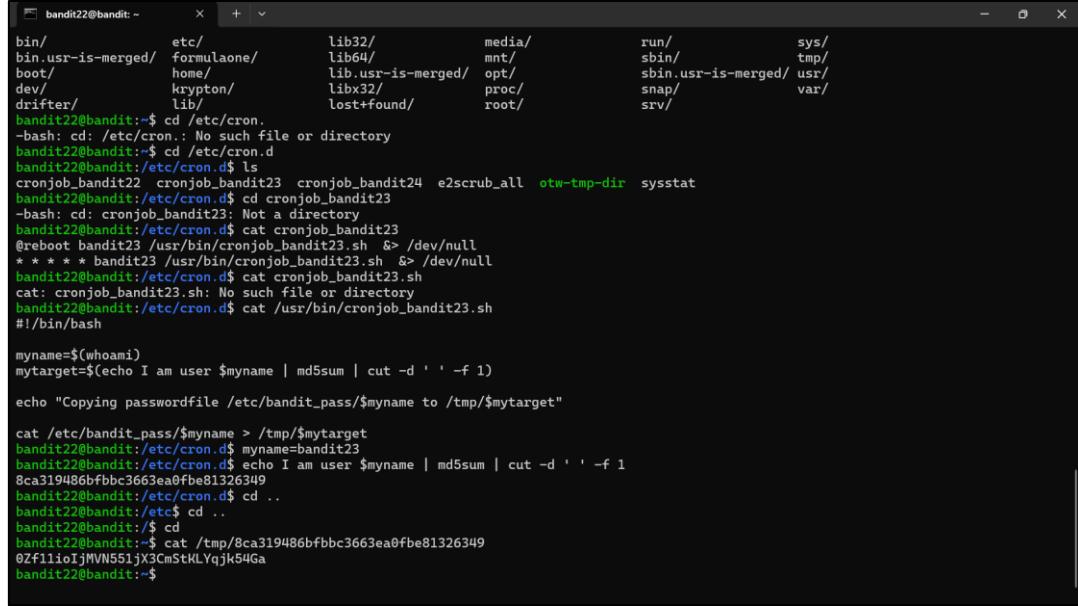
Level 22 → Level 23

Objective

The objective of this level is to find the password for the next level, this by analyzing the shell that is running at regular intervals. This means that we need to examine the cron job configuration and associated script.

Methodology

To find the password we are going to go the directory that contain the shell and by using the command *cat cronjob_bandit23.sh*



```

bandit22@bandit: ~
bin/      etc/      lib32/      media/      run/      sys/
bin usr-is-merged/ formulaone/ lib64/      mnt/      sbin/      tmp/
boot/     home/     lib usr-is-merged/ opt/      sbin usr-is-merged/ usr/
dev/      krypton/  libx32/     proc/      snap/      var/
drifter/   lib/     lost+found/ root/      srv/
bandit22@bandit:~$ cd /etc/cron.
-bash: cd: /etc/cron.: No such file or directory
bandit22@bandit:~$ cd /etc/cron.d
bandit22@bandit:/etc/cron.d$ ls
cronjob_bandit22  cronjob_bandit23  cronjob_bandit24  e2scrub_all  otw-tmp-dir  sysstat
bandit22@bandit:/etc/cron.d$ cd cronjob_bandit23
-bash: cd: cronjob_bandit23: Not a directory
bandit22@bandit:/etc/cron.d$ cat cronjob_bandit23
@reboot bandit23 /usr/bin/cronjob_bandit23.sh &> /dev/null
* * * * * bandit23 /usr/bin/cronjob_bandit23.sh &> /dev/null
bandit22@bandit:/etc/cron.d$ cat cronjob_bandit23.sh
cat: cronjob_bandit23.sh: No such file or directory
bandit22@bandit:/etc/cron.d$ cat /usr/bin/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
bandit22@bandit:/etc/cron.d$ myname=bandit23
bandit22@bandit:/etc/cron.d$ echo I am user $myname | md5sum | cut -d ' ' -f 1
8ca319486bfbb3663ea0fbe81326349
bandit22@bandit:/etc/cron.d$ cd ..
bandit22@bandit:/etc$ cd ..
bandit22@bandit:/etc$ cd ..
bandit22@bandit:/etc$ cat /tmp/8ca319486bfbb3663ea0fbe81326349
0Zf11ioIjMVN551jX3CmStKLYqjk54Ga
bandit22@bandit:~$
```

As we can see in the image above, we first saw how the shell works. With the command that the shell give to us we put it with an asignation of *myname=bandit23* (*that is the current user*). Then we shaw that the code that display was a file, so we do a cat command and the password was there.

mytarget=\$(echo I am user \$myname | md5sum | cut -d ' ' -f 1): Creates an MD5 hash of the string "I am user \$myname" and extracts the hash using cut. This hash is assigned to the variable mytarget.

Result: **0Zf11ioIjMVN551jX3CmStKLYqjk54Ga**

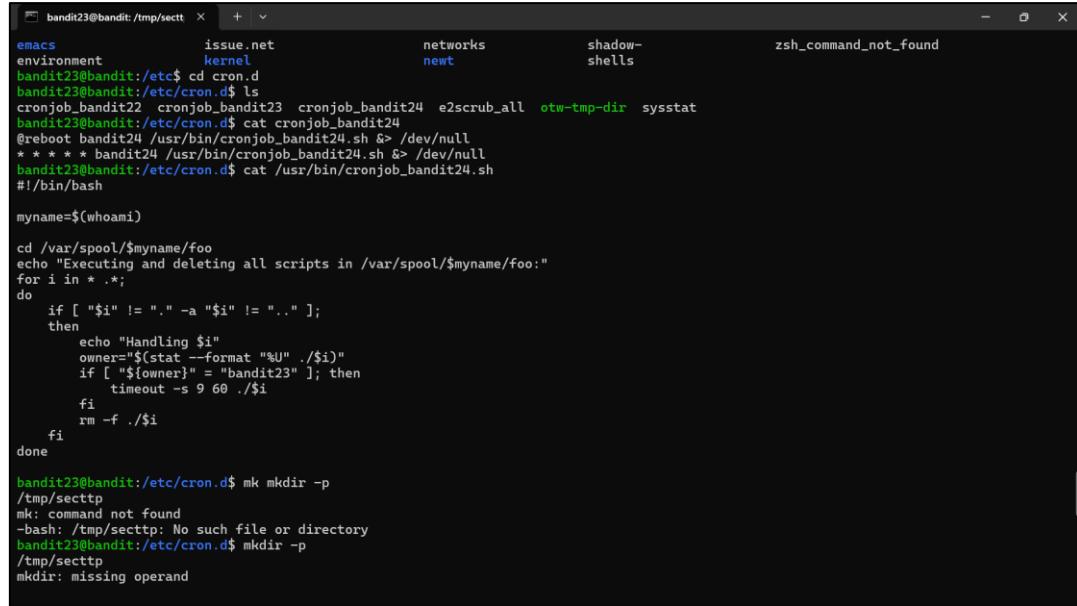
Level 23 → Level 24

Objective

The objective of this level is to find the password for the next level, this by analyzing the shell that is running at regular intervals. This means that we need to examine the cron job configuration and associated script. Also, we need to understand how cron jobs execute scripts and how them mange things.

Methodology

To find the password we are going to go the directory that contain the shell and by using the comand `cat /etc/cron.d/cronjob_bandit24.sh`. In order to saw the content of the bandit 23 cron job.



```

bandit23@bandit: /tmp/seccett ~ + - zsh_command_not_found
environment      issue.net      networks      shadow-shells
environment      kernel       newt
bandit23@bandit: /etc/cron.d$ cd cron.d
bandit23@bandit: /etc/cron.d$ ls
cronjob_bandit22 cronjob_bandit23 cronjob_bandit24 e2scrub_all otw-tmp-dir sysstat
bandit23@bandit: /etc/cron.d$ cat cronjob_bandit24
@reboot bandit24 /usr/bin/cronjob_bandit24.sh >> /dev/null
* * * * * bandit24 /usr/bin/cronjob_bandit24.sh &> /dev/null
bandit23@bandit: /etc/cron.d$ cat /usr/bin/cronjob_bandit24.sh
#!/bin/bash

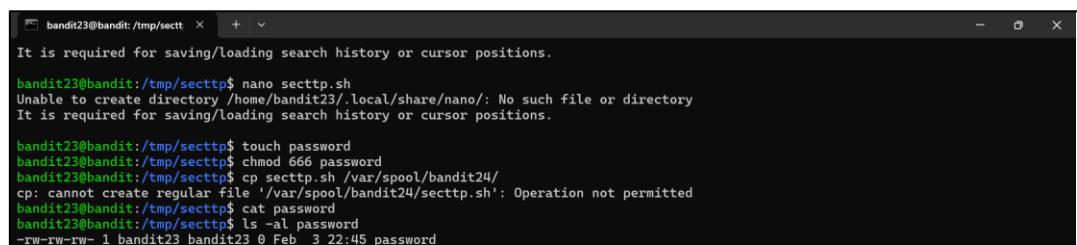
myname=$(whoami)

cd /var/spool/$myname/foo
echo "Executing and deleting all scripts in /var/spool/$myname/foo:"
for i in *.*;
do
    if [ "$i" != "." -a "$i" != ".." ];
    then
        echo "Handling $i"
        owner=$(stat --format "%U" ./i)
        if [ "$owner" = "bandit23" ]; then
            timeout -s 90 ./i
        fi
        rm -f ./i
    fi
done

bandit23@bandit: /etc/cron.d$ mk mkdir -p
/tmp/seccett
mk: command not found
-bash: /tmp/seccett: No such file or directory
bandit23@bandit: /etc/cron.d$ mkdir -p
/tmp/seccett
mkdir: missing operand

```

Then we have to make some configuration for copying the file that has the password in which we don't have access to into other in which we do have access to. So we follow the following steps: To create a shell script that will be executed by the cron job, follow these steps ‘`mkdir -p /tmp/seccett`’, ‘`cd /tmp/seccett`’, ‘`touch secttp.sh`’, ‘`chmod 777 secttp.sh`’ and ‘`vim secttp.sh`’.



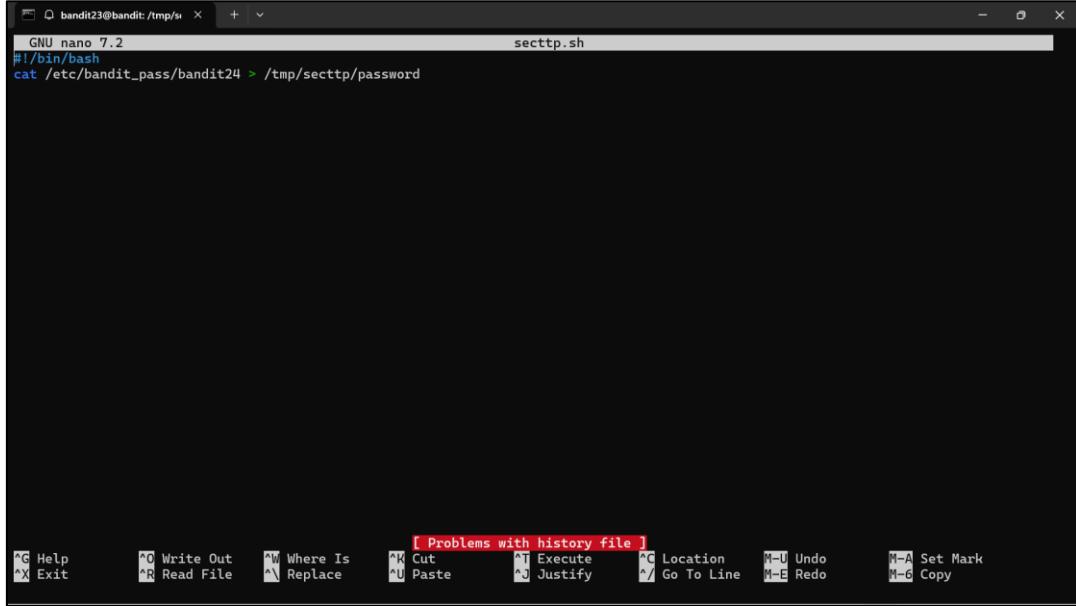
```

bandit23@bandit: /tmp/seccett ~ + - zsh_command_not_found
It is required for saving/loading search history or cursor positions.

bandit23@bandit: /tmp/seccett$ nano secttp.sh
Unable to create directory /home/bandit23/.local/share/nano/: No such file or directory
It is required for saving/loading search history or cursor positions.

bandit23@bandit: /tmp/seccett$ touch password
bandit23@bandit: /tmp/seccett$ chmod 666 password
bandit23@bandit: /tmp/seccett$ cp secttp.sh /var/spool/bandit24/
cp: cannot create regular file '/var/spool/bandit24/secttp.sh': Operation not permitted
bandit23@bandit: /tmp/seccett$ cat password
bandit23@bandit: /tmp/seccett$ ls -al password
-rw-rw-rw- 1 bandit23 bandit23 0 Feb 3 22:45 password

```



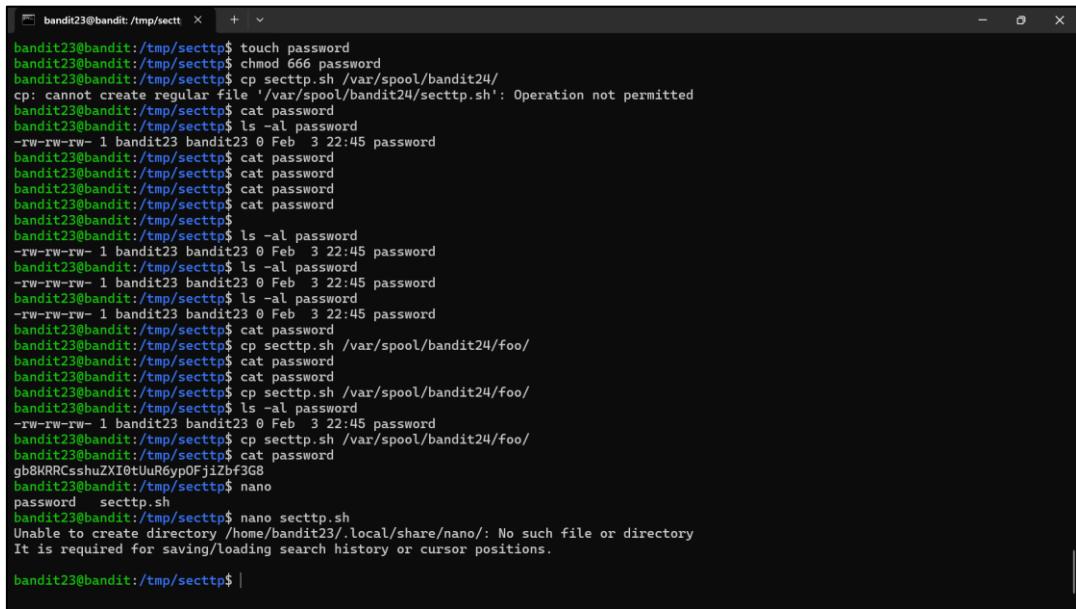
```
GNU nano 7.2          secttp.sh
#!/bin/bash
cat /etc/bandit_pass/bandit24 > /tmp/secttp/password

[ Problems with history file ]
^G Help      ^O Write Out   ^W Where Is      ^K Cut        ^T Execute    ^C Location    M-U Undo
^X Exit      ^R Read File   ^N Replace      ^U Paste     ^J Justify    ^L Go To Line  M-E Redo
                                         M-A Set Mark  M-C Copy
```

This will take the file and paste the info that has on it in the file that we choose.

cp secttp.sh /var/spool/bandit24/foo: Copies secttp.sh to /var/spool/bandit24/ with the name foo.

touch password: Creates an empty file named password in the /tmp/secttp directory.



```
bandit23@bandit:/tmp/secttp$ touch password
bandit23@bandit:/tmp/secttp$ chmod 666 password
bandit23@bandit:/tmp/secttp$ cp secttp.sh /var/spool/bandit24/
cp: cannot create regular file '/var/spool/bandit24/secttp.sh': Operation not permitted
bandit23@bandit:/tmp/secttp$ cat password
bandit23@bandit:/tmp/secttp$ ls -al password
-rw-rw-rw- 1 bandit23 0 Feb  3 22:45 password
bandit23@bandit:/tmp/secttp$ cat password
bandit23@bandit:/tmp/secttp$ cat password
bandit23@bandit:/tmp/secttp$ cat password
bandit23@bandit:/tmp/secttp$ cat password
bandit23@bandit:/tmp/secttp$ ls -al password
-rw-rw-rw- 1 bandit23 0 Feb  3 22:45 password
bandit23@bandit:/tmp/secttp$ ls -al password
-rw-rw-rw- 1 bandit23 0 Feb  3 22:45 password
bandit23@bandit:/tmp/secttp$ ls -al password
-rw-rw-rw- 1 bandit23 0 Feb  3 22:45 password
bandit23@bandit:/tmp/secttp$ cat password
bandit23@bandit:/tmp/secttp$ cp secttp.sh /var/spool/bandit24/foo/
bandit23@bandit:/tmp/secttp$ cat password
bandit23@bandit:/tmp/secttp$ cp secttp.sh /var/spool/bandit24/foo/
bandit23@bandit:/tmp/secttp$ ls -al password
-rw-rw-rw- 1 bandit23 0 Feb  3 22:45 password
bandit23@bandit:/tmp/secttp$ cp secttp.sh /var/spool/bandit24/foo/
bandit23@bandit:/tmp/secttp$ cat password
gb8KRRCsahuZXI0tU6yp0Fji2bf3G8
bandit23@bandit:/tmp/secttp$ nano
password secttp.sh
bandit23@bandit:/tmp/secttp$ nano secttp.sh
Unable to create directory /home/bandit23/.local/share/nano/: No such file or directory
It is required for saving/loading search history or cursor positions.

bandit23@bandit:/tmp/secttp$ |
```

Copying the Script to the Cron Job Directory Move the script to the directory where the cron job will pick it up. The following command is used ‘cp secttp.sh /var/spool/bandit24/foo’

After the cron job executes the script, check the contents of the password file ‘cat password’ and at the end the file will find the password.

Result: **gb8KRRCCsshuZXI0tUuR6ypOFjiZbf3G8**

Level 24 → Level 25

Objective

The objective of this level is to find the password by providing the password of the bandit 24 and a secret numeric 4-digit pin code. This only by going to all the 10000 combinations by brute-forcing

Methodology

We are going to cerate a directory in which the script and the other documents will be stored with the following commands: ‘mkdir /tmp/pass24’ and ‘cd /tmp/pass24’

```
bandit24@bandit:~$ cd /tmp/pass24
bandit24@bandit:/tmp/pass24$ ls
bruteforce.sh  pins.txt
bandit24@bandit:/tmp/pass24$ nano bruteforce.sh
Unable to create directory /home/bandit24/.local/share/nano/: No such file or directory
It is required for saving/loading search history or cursor positions.

bandit24@bandit:/tmp/pass24$ chmod +x bruteforce.sh
bandit24@bandit:/tmp/pass24$ ./bruteforce.sh
bandit24@bandit:/tmp/pass24$
```

Then we use bruteforce in order to identify the correct code. This will drop it in the archive pins.txt

```
bandit24@bandit: /tmp/pass24
GNU nano 7.2
#!/bin/bash
for i in {000..9999}
do
    echo "gb8KRRCCsshuZXI0tUuR6ypOFjiZbf3G8 $i" >> pins.txt
done
```

This is the shell that we will execute

```
bandit24@bandit:/tmp/pass24$ chmod +x bruteforce.sh
bandit24@bandit:/tmp/pass24$ ./bruteforce.sh
bandit24@bandit:/tmp/pass24$ nc localhost 30002 < pins.txt
```

Then we execute the shell

nc localhost 30002 < pins.txt: Sends the contents of pins.txt to the daemon on port 30002. The daemon will process the file and return the password for bandit25 if the correct PIN is included.

```
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Correct!
The password of user bandit25 is iCi86ttT4KSNe1armKiwbQNmB3YJP3q4
```

Result: **iCi86ttT4KSNe1armKiwbQNmB3YJP3q4**

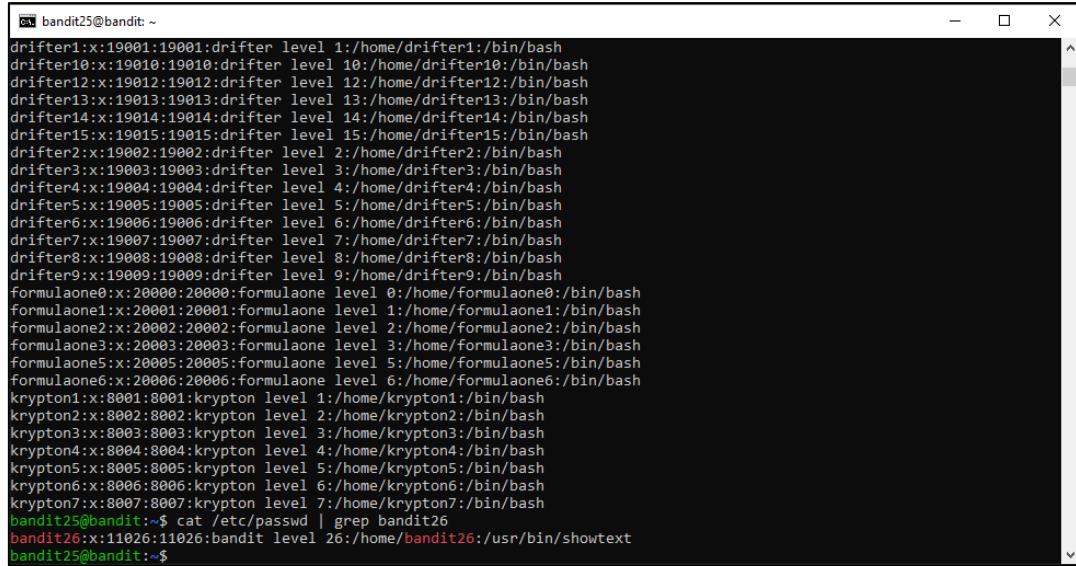
Level 25 → Level 26

Objective

The objective of this level is to find the password by Logging in to bandit26 from bandit25 should be fairly easy... The shell for user bandit26 is not /bin/bash, but something else. Find out what it is, how it works and how to break out of it.

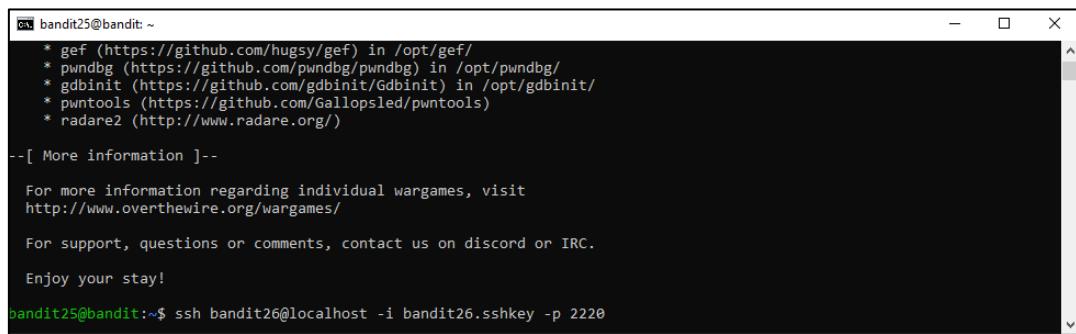
Methodology

We are going to identify the shell for bandit26. In order to determine the shell used by using the following command: `cat /etc/passwd | grep bandit26`. By the usage of this command we will have the correct bash for bandit26.



```
bandit25@bandit:~
drifter1:x:19001:19001:drifter level 1:/home/drifter1:/bin/bash
drifter10:x:19010:19010:drifter level 10:/home/drifter10:/bin/bash
drifter12:x:19012:19012:drifter level 12:/home/drifter12:/bin/bash
drifter13:x:19013:19013:drifter level 13:/home/drifter13:/bin/bash
drifter14:x:19014:19014:drifter level 14:/home/drifter14:/bin/bash
drifter15:x:19015:19015:drifter level 15:/home/drifter15:/bin/bash
drifter2:x:19002:19002:drifter level 2:/home/drifter2:/bin/bash
drifter3:x:19003:19003:drifter level 3:/home/drifter3:/bin/bash
drifter4:x:19004:19004:drifter level 4:/home/drifter4:/bin/bash
drifter5:x:19005:19005:drifter level 5:/home/drifter5:/bin/bash
drifter6:x:19006:19006:drifter level 6:/home/drifter6:/bin/bash
drifter7:x:19007:19007:drifter level 7:/home/drifter7:/bin/bash
drifter8:x:19008:19008:drifter level 8:/home/drifter8:/bin/bash
drifter9:x:19009:19009:drifter level 9:/home/drifter9:/bin/bash
formulaone0:x:20000:20000:formulaone level 0:/home/formulaone0:/bin/bash
formulaone1:x:20001:20001:formulaone level 1:/home/formulaone1:/bin/bash
formulaone2:x:20002:20002:formulaone level 2:/home/formulaone2:/bin/bash
formulaone3:x:20003:20003:formulaone level 3:/home/formulaone3:/bin/bash
formulaone5:x:20005:20005:formulaone level 5:/home/formulaone5:/bin/bash
formulaone6:x:20006:20006:formulaone level 6:/home/formulaone6:/bin/bash
krypton1:x:8001:8001:krypton level 1:/home/krypton1:/bin/bash
krypton2:x:8002:8002:krypton level 2:/home/krypton2:/bin/bash
krypton3:x:8003:8003:krypton level 3:/home/krypton3:/bin/bash
krypton4:x:8004:8004:krypton level 4:/home/krypton4:/bin/bash
krypton5:x:8005:8005:krypton level 5:/home/krypton5:/bin/bash
krypton6:x:8006:8006:krypton level 6:/home/krypton6:/bin/bash
krypton7:x:8007:8007:krypton level 7:/home/krypton7:/bin/bash
bandit25@bandit:~$ cat /etc/passwd | grep bandit26
bandit26:x:11026:11026:bandit level 26:/home/bandit26:/usr/bin/showtext
bandit25@bandit:~$
```

Then we are going to try to log in into bandit26 in the bash of bandit25. This by using the following command. This command will use the key that is contained in the file `bandit26.sshkey`



```
bandit25@bandit:~
* gef (https://github.com/hugsy/gef) in /opt/gef/
* pwndbg (https://github.com/pwndbg/pwndbg) in /opt/pwndbg/
* gdbinit (https://github.com/Gdbinit/Gdbinit) in /opt/gdbinit/
* pwntools (https://github.com/Gallopsled/pwntools)
* radare2 (http://www.radare.org/)

--[ More information ]--

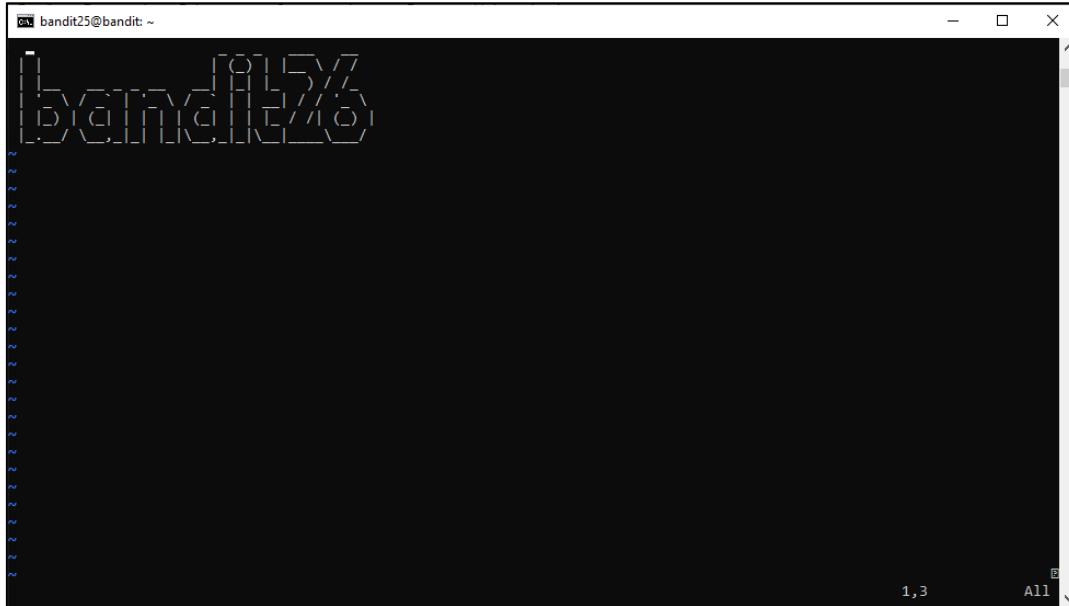
For more information regarding individual wargames, visit
http://www.overthewire.org/wargames

For support, questions or comments, contact us on discord or IRC.

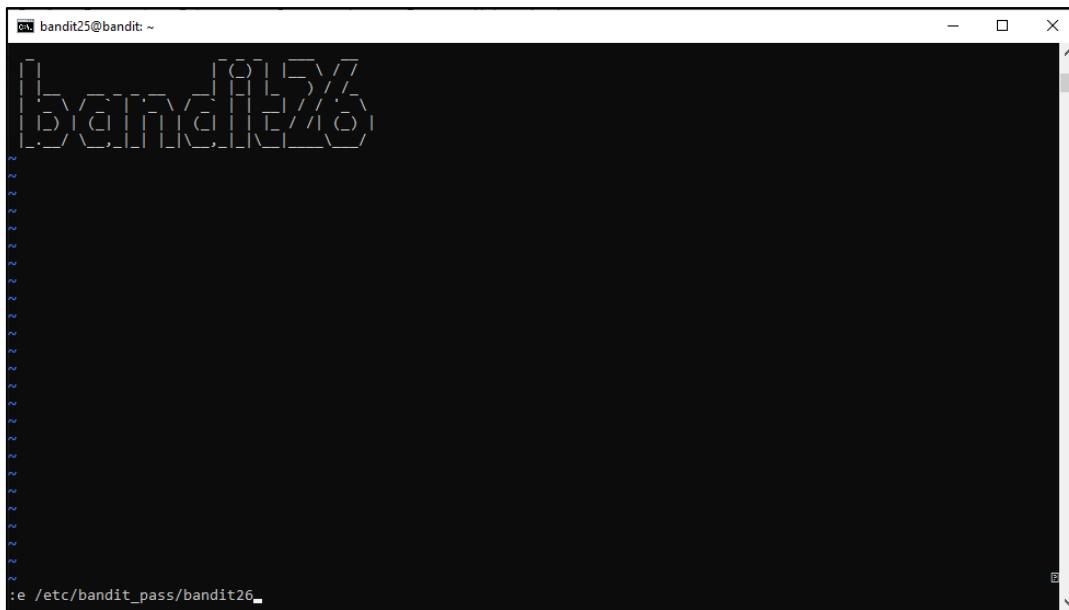
Enjoy your stay!

bandit25@bandit:~$ ssh bandit26@localhost -i bandit26.sshkey -p 2220
```

In this part we have to take advantage of a type of bug that allows us to enter in via at the bash of bandit26 and then we just use a cat command.



A terminal window titled "bandit25@bandit: ~". The screen displays a large amount of text, likely a password dictionary or a log file, consisting of various symbols and characters. At the bottom right of the terminal, there are two status indicators: "1,3" and "All".



A terminal window titled "bandit25@bandit: ~". The screen displays a large amount of text, likely a password dictionary or a log file, consisting of various symbols and characters. At the bottom left of the terminal, there is a command: ":e /etc/bandit_pass/bandit26".

Result: s0773xxkk0MXfdqOfPRVr9L3jJBUOgCZ

Level 26 → Level 27

Objective

The objective of this level is to obtain a shell and use it to quickly retrieve the password for the next level, bandit27.

Methodology

For this level, we continue to take advantage of the "bug" in `ResultsmoreResults`, which allows us to execute commands in `ResultsviResults`, and through the `pass` directory, we obtain the password for bandit27.

Result: upsNCc7vzaRDx6oZC6GiR6ERwe1MowGB

Level 27 → Level 28

Objective

The objective of this level is to access the git repository at ssh://bandit27-git@localhost/home/bandit27-git/repo via port 2220. The password for the user bandit27-git is the same as the one for the user bandit27.

Methodology

Once cloned, you need to navigate through the repository and look for the `Results/readmeResults` file, as this file generally contains additional information, and in this specific case, the password for the next level.

```
bandit27@bandit: /tmp/repo
Cloning into 'repo'...
The authenticity of host '[localhost]:2220 ([127.0.0.1]:2220)' can't be established.
ED25519 key fingerprint is SHA256:C2ihUBV7ihnV1wUXRb4RrEcLfxC5CXlhmAAM/urerLY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Could not create directory '/home/bandit27/.ssh' (Permission denied).
Failed to add the host to the list of known hosts (/home/bandit27/.ssh/known_hosts).

      [ ] _ \ / - . - \ / - [ ] - [ ]
      [ ] | [ ] | [ ] | [ ] | [ ] |
      [ ] _ / \ _ , _ \ _ | \ _ , _ \ _ | \ _ |

This is an OverTheWire game server.
More information on http://www.overthewire.org/wargames

bandit27-git@localhost's password:
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), 285 bytes | 285.00 KiB/s, done.
bandit27@bandit:/tmp/repo$ LS
LS: command not found
bandit27@bandit:/tmp/repo$ ls
README repo
bandit27@bandit:/tmp/repo$ cat README
The password to the next level is: Yz9IpL0sBcCeuG7m9uQFt8ZNpS4HZRcN
bandit27@bandit:/tmp/repo$
```

Result: Yz9IpL0sBcCeuG7m9uQFt8ZNpS4HZRcN

Level 28 → Level 29

Objective

The objective of this level is to access the git repository at ssh://bandit28-git@localhost/home/bandit28-git/repo via port 2220. The password for the user bandit28-git is the same as the one for the user bandit28.

Methodology

First, we create a folder in /tmp to clone the repository and check the file inside. Since that file doesn't provide the key, we use git log to view previous commits and switch to the branch that made the last commit. Finally, we open the readme file, where we find the credentials for the next level.

```
bandit28@bandit:/tmp/repo2/repo
bandit28@bandit:/tmp/repo2/repo$ git logs
git: 'logs' is not a git command. See 'git --help'.

The most similar command is
    log
bandit28@bandit:/tmp/repo2/repo$ git log
commit 817e303aa6c2b207ea043c7bbabb757dc4ea73 (HEAD -> master, origin/master, origin/HEAD)
Author: Morla Porla <morla@overthewire.org>
Date:   Thu Sep 19 07:08:39 2024 +0000

    fix info leak

commit 3621de89d8eac9d3b64302bfb2dc67e9a566dec
Author: Morla Porla <morla@overthewire.org>
Date:   Thu Sep 19 07:08:39 2024 +0000

    add missing data

commit 0622b73250502618babac3d174724bb303c32182
Author: Ben Dover <noone@overthewire.org>
Date:   Thu Sep 19 07:08:39 2024 +0000

    initial commit of README.md
bandit28@bandit:/tmp/repo2/repo$ git show 0622b73250502618babac3d174724bb303c32182
commit 0622b73250502618babac3d174724bb303c32182
Author: Ben Dover <noone@overthewire.org>
Date:   Thu Sep 19 07:08:39 2024 +0000

    initial commit of README.md
```

In this part we use the command show using as a parameter the code of the commit.

```
bandit28@bandit:/tmp/repo2/repo
--- /dev/null
+++ b/README.md
@@ -0,0 +1,8 @@
## Bandit Notes
Some notes for level29 of bandit.
+
## credentials
+
+- username: bandit29
+- password: <TBD>
+
bandit28@bandit:/tmp/repo2/repo$ git show 3621de89d8eac9d3b64302bfb2dc67e9a566dec
commit 3621de89d8eac9d3b64302bfb2dc67e9a566dec
Author: Morla Porla <morla@overthewire.org>
Date:   Thu Sep 19 07:08:39 2024 +0000

    add missing data

diff --git a/README.md b/README.md
index 7ba2d2f..d4e3b74 100644
--- a/README.md
+++ b/README.md
@@ -4,5 +4,5 @@ Some notes for level29 of bandit.
## credentials

- username: bandit29
-- password: <TBD>
+- password: 4pT1t5DENaYuqnqvadYs1oE4QLCdjmJ7

bandit28@bandit:/tmp/repo2/repo$
```

And just by trying with two commits we found the password.

Result: 4pT1t5DENaYuqnqvadYs1oE4QLCdjmJ7

Level 29 → Level 30

Objective

The objective of this level is to clone the git repository at ssh://bandit29-git@localhost/home/bandit29-git/repo via port 2220. The password for the user bandit29-git is the same as the one for the user bandit29. Once cloned, find the password for the next level.

Methodology

For this level, we follow the same steps as the previous one until we reach the `readme.txt`, which says "no password in production." After that, we check the branches and switch to the `dev` branch, where we find the key for this level.

```
bandit29@bandit: /tmp/repo3/repo
Connection closed by 127.0.0.1 port 22
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
bandit29@bandit:/tmp/repo3$ git clone ssh://bandit29-git@localhost:2220/home/bandit29-git/repo
Cloning into 'repo'...
Connection closed by 127.0.0.1 port 2220
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
bandit29@bandit:/tmp/repo3$ git clone ssh://bandit29-git@localhost:2220/home/bandit29-git/repo
Cloning into 'repo'...
The authenticity of host '[localhost]:2220 ([127.0.0.1]:2220)' can't be established.
ED25519 key fingerprint is SHA256:C2ihUBV7ihihV1wUXRb4RrEcLXC5CX1hmAM/urerLY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes[[B
Please type 'yes', 'no' or the fingerprint: yes
Could not create directory '/home/bandit29/.ssh' (Permission denied).
Failed to add the host to the list of known hosts (/home/bandit29/.ssh/known_hosts).

[|_ \ /--.-\ .-.\ /_ [(_)]_]
 [(_)] [(_)] [(_)] [(_)] [(_)] [(_)]
 [ _] \ _/ _/ _/ \ _/ _/ _/ \ _/
```

```
bandit29@bandit: /tmp/repo3/repo

bandit29@localhost's password:
remote: Enumerating objects: 16, done.
remote: Counting objects: 100% (16/16), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 16 (delta 2), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (16/16), done.
Resolving deltas: 100% (2/2), done.
bandit29@bandit:/tmp/repo3/repo$ cd repo/
bandit29@bandit:/tmp/repo3/repo$ git branch
* master
bandit29@bandit:/tmp/repo3/repo$ git branch -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/dev
  remotes/origin/master
  remotes/origin/sploits-dev
bandit29@bandit:/tmp/repo3/repo$ git checkout dev
branch 'dev' set up to track 'origin/dev'.
Switched to a new branch 'dev'
bandit29@bandit:/tmp/repo3/repo$ cat README.md
# Bandit Notes
Some notes for bandit30 of bandit.

## credentials

- username: bandit30
- password: qp30ex3VLz5MDG1n91YowTv4Q8l7CDZL

bandit29@bandit:/tmp/repo3/repo$
```

Results: qp30ex3VLz5MDG1n91YowTv4Q8l7CDZL

Level 30 → Level 31

Objective

The objective of this level is to find the password by cloning the repository using the URL provided in the support material of the website. Then we must move in the repo to find clues of the password.

Methodology

First, a temporary directory was created in /tmp to organize the workspace, and the repository was cloned using the provided SSH URL. Then, the cloned repository was navigated to, and the commit history was examined using the git log command to look for relevant information. After identifying a relevant commit, the commit's content was inspected using git show along with the commit hash. Next, the tags in the repository were listed using git tag, and the "secret" tag was examined with git show secret, where the password for the next level was finally found.

```
bandit30@bandit:~/tmp/repo4
Enjoy your stay!
bandit30@bandit:~$ mkdir /tmp/repo4
bandit30@bandit:~$ cd /tmp/repo4
bandit30@bandit:/tmp/repo4$ git clone ssh://bandit30-git@localhost:2220/home/bandit30-git/repo
Cloning into 'repo'...
The authenticity of host '[localhost]:2220 ([127.0.0.1]:2220)' can't be established.
ED25519 key fingerprint is SHA256:C2ihuBV71hnV1wUXRb4RnEclfxC5CXlhmAAM/urerLY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? wyes
Please type 'yes', 'no' or the fingerprint: yes
Could not create directory '/home/bandit30/.ssh' (Permission denied).
Failed to add the host to the list of known hosts (/home/bandit30/.ssh/known_hosts).

      _\ _ \ /_ - - \ / _ \ / [ ( ) ] _ \
     [ _ ) | ( _ | [ ] | | ( _ | | [ ] |
    [ _ - / \ _ , _ | _ | \ _ , _ | _ | \ _ |

This is an OverTheWire game server.
More information on http://www.overthewire.org/wargames

bandit30-git@localhost's password:
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.
bandit30@bandit:/tmp/repo4$
```

```
bandit30@bandit:/tmp/repo4/repo
| |) |(| | | | |(| | | |
|_|/_\_,_|_|_|_|_\_,_|_|_\_|

This is an OverTheWire game server.
More information on http://www.overthewire.org/wargames

bandit30-git@localhost's password:
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.
bandit30@bandit:/tmp/repo4$ ls
repo
bandit30@bandit:/tmp/repo4$ cd repo/
bandit30@bandit:/tmp/repo4/repo$ ls
README.md
bandit30@bandit:/tmp/repo4/repo$ cat README.md
just an empty file... muahaha
bandit30@bandit:/tmp/repo4/repo$ git log
commit acfc3c67816fc778c4ae5893299451ca6d65a78 (HEAD -> master, origin/master, origin/HEAD)
Author: Ben Dover <noone@overthewire.org>
Date:   Thu Sep 19 07:08:44 2024 +0000

    initial commit of README.md
bandit30@bandit:/tmp/repo4/repo$ git show acfc3c67816fc778c4ae5893299451ca6d65a78
commit acfc3c67816fc778c4ae5893299451ca6d65a78 (HEAD -> master, origin/master, origin/HEAD)
Author: Ben Dover <noone@overthewire.org>
Date:   Thu Sep 19 07:08:44 2024 +0000

initial commit of README.md
bandit30@bandit:/tmp/repo4/repo$ git show acfc3c67816fc778c4ae5893299451ca6d65a78
commit acfc3c67816fc778c4ae5893299451ca6d65a78 (HEAD -> master, origin/master, origin/HEAD)
Author: Ben Dover <noone@overthewire.org>
Date:   Thu Sep 19 07:08:44 2024 +0000

initial commit of README.md

diff --git a/README.md b/README.md
new file mode 100644
index 000000..029ba42
--- /dev/null
+++ b/README.md
@@ -0,0 +1 @@
+just an empty file... muahaha
bandit30@bandit:/tmp/repo4/repo$ git branch -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
bandit30@bandit:/tmp/repo4/repo$ git tag
secret
bandit30@bandit:/tmp/repo4/repo$ git show secret
fb552xb7DrYfAvQYQGeqsbhVyJqhnb
bandit30@bandit:/tmp/repo4/repo$
```

- **mkdir -p /tmp/b30:** Creates a temporary directory named b30 in /tmp.
- **cd /tmp/b30:** Navigates into the b30 directory.
- **git clone ssh://bandit30-git@localhost/home/bandit30-git/repo:** Clones the Git repository from the provided SSH URL.
- **cd repo:** Changes into the cloned repository's directory.
- **git log:** Displays the commit history of the repository, including commit messages, authors, and commit hashes.
- **git show numberCommit:** Displays the content and changes associated with a specific commit identified by its hash.
- **git tag:** Lists all tags in the repository, which may point to significant commits or information.

- **git show secret:** Displays the content associated with the "secret" tag, revealing important information.

Result: fb5S2xb7bRyFmAvQYQGEqsbhVyJqhnDy

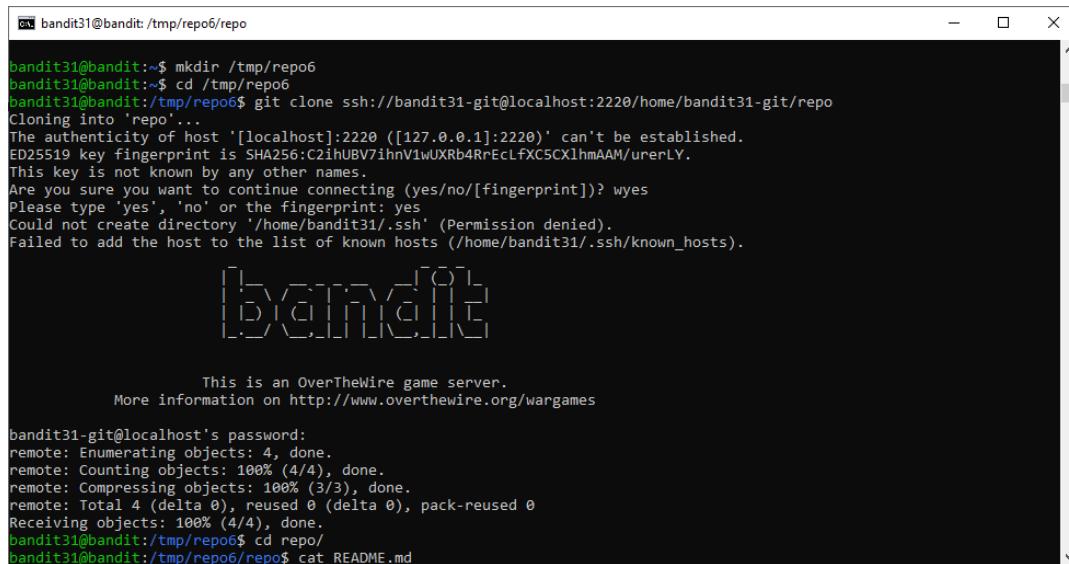
Level 31 → Level 32

Objective

The objective of this level is to find the password by Logging in to bandit26 from bandit25 should be fairly easy... The shell for user bandit26 is not /bin/bash, but something else. Find out what it is, how it works and how to break out of it.

Methodology

To complete this task, first, a temporary directory is created and navigated to using **mkdir -p /tmp/b31** and **cd /tmp/b31**. The Git repository is then cloned using **git clone ssh://bandit31-git@localhost/home/bandit31-git/repo**. After cloning, the README file is read with **cat README.md**, which provides instructions. Following these instructions, a key.txt file is created with the content "May I come in?" using **echo "May I come in?" > key.txt**. Next, the **.gitignore** file is edited with **vim .gitignore** to stop ignoring .txt files. The changes are staged with **git add key.txt** and committed with **git commit -m "requiredMessage"**. Finally, the changes are pushed to the remote repository using **git push origin master**, successfully revealing the password for the next level.



```

bandit31@bandit:~/tmp/repo6$ mkdir /tmp/repo6
bandit31@bandit:~/tmp$ cd /tmp/repo6
bandit31@bandit:/tmp/repo6$ git clone ssh://bandit31-git@localhost:2220/home/bandit31-git/repo
Cloning into 'repo'...
The authenticity of host '[localhost]:2220 ([127.0.0.1]:2220)' can't be established.
ED25519 key fingerprint is SHA256:C2ihUBV7ihnViwUXRb4RrEcLfxC5CXlhmAAM/urerLY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? wyes
Please type 'yes', 'no' or the fingerprint: yes
Could not create directory '/home/bandit31/.ssh' (Permission denied).
Failed to add the host to the list of known hosts (/home/bandit31/.ssh/known_hosts).

          _/\_ 
         [ ] [ ] 
        [ ] [ ] 
       [ ] [ ] 
      [ ] [ ] 
     [ ] [ ] 
    [ ] [ ] 
   [ ] [ ] 
  [ ] [ ] 
 [ ] [ ] 
[ ] [ ] 

This is an OverTheWire game server.
More information on http://www.overthewire.org/wargames

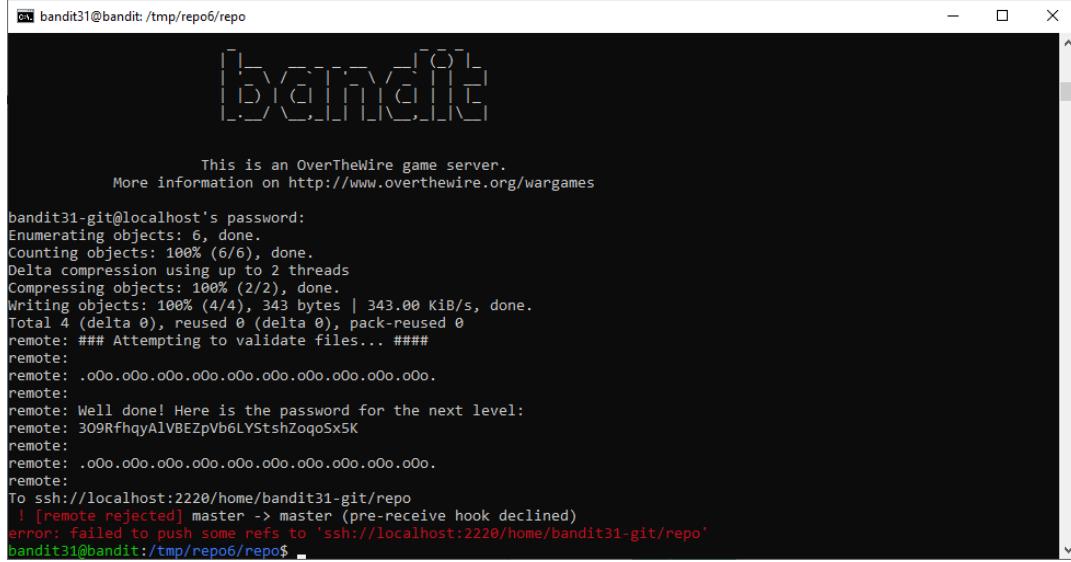
bandit31-git@localhost's password:
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.
bandit31@bandit:/tmp/repo6$ cd repo/
bandit31@bandit:/tmp/repo6/repo$ cat README.md

```

```
bandit31@bandit: /tmp/repo6/repo$ ls
README.md
bandit31@bandit: /tmp/repo6/repo$ ls -la
total 20
drwxrwxr-x 3 bandit31 bandit31 4096 Feb 4 04:23 .
drwxrwxr-x 3 bandit31 bandit31 4096 Feb 4 04:23 ..
drwxrwxr-x 8 bandit31 bandit31 4096 Feb 4 04:23 .git
-rw-rw-r-- 1 bandit31 bandit31 6 Feb 4 04:23 .gitignore
-rw-rw-r-- 1 bandit31 bandit31 147 Feb 4 04:23 README.md
bandit31@bandit: /tmp/repo6/repo$ echo "May I come in? >" key.txt
May I come in? > key.txt
bandit31@bandit: /tmp/repo6/repo$ ls
README.md
bandit31@bandit: /tmp/repo6/repo$ echo "May I come in?" > key.txt
bandit31@bandit: /tmp/repo6/repo$ ls
key.txt README.md
bandit31@bandit: /tmp/repo6/repo$ git add key.txt
The following paths are ignored by one of your .gitignore files:
key.txt
hint: Use -f if you really want to add them.
hint: Turn this message off by running
hint: "git config advice.addIgnoredFile false"
bandit31@bandit: /tmp/repo6/repo$ nano .gitignore
Unable to create directory /home/bandit31/.local/share/nano/: No such file or directory
It is required for saving/loading search history or cursor positions.

Unable to create directory /home/bandit31/.local/share/nano/: No such file or directory
It is required for saving/loading search history or cursor positions.

bandit31@bandit: /tmp/repo6/repo$ git add key.txt
```



```
[bandit31@bandit: /tmp/repo6/repo]
[!] [!] [!] [!] [!] [!]
[!] [!] [!] [!] [!] [!]

This is an OverTheWire game server.
More information on http://www.overthewire.org/wargames

bandit31-git@localhost's password:
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 343 bytes | 343.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
remote: ### Attempting to validate files... ####
remote:
remote: .000.000.000.000.000.000.000.000.
remote:
remote: Well done! Here is the password for the next level:
remote: 3O9RfhqyAlVBEZpVb6LYStshZoqoSx5K
remote:
remote: .000.000.000.000.000.000.000.000.
remote:
remote: To ssh://localhost:2220/home/bandit31-git/repo
remote: ! [remote rejected] master -> master (pre-receive hook declined)
error: failed to push some refs to 'ssh://localhost:2220/home/bandit31-git/repo'
bandit31@bandit:/tmp/repo6/repo$
```

Result: 3O9RfhqyAlVBEZpVb6LYStshZoqoSx5K

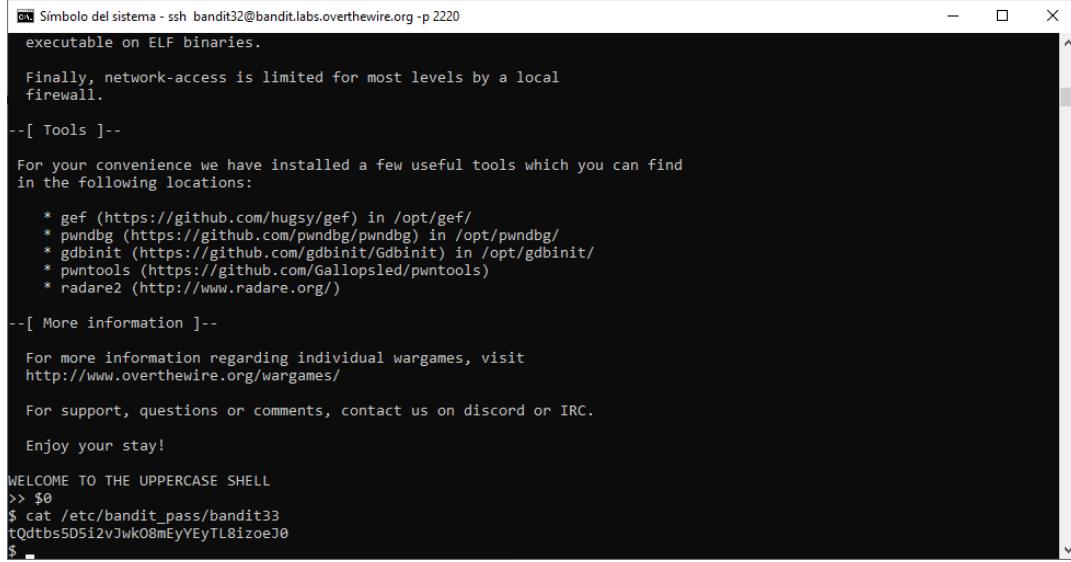
Level 32 → Level 33

Objective

The objective of this level is to find the password by implementing new commands and learn how does other types of bash works.

Methodology

We are going to only use the command \$0: In Bash, \$0 can be used to print the name of the shell or script. In this context, it reveals information about the shell being used.



```
Símbolo del sistema - ssh bandit32@bandit.labs.overthewire.org -p 2220
executable on ELF binaries.

Finally, network-access is limited for most levels by a local
firewall.

--[ Tools ]--

For your convenience we have installed a few useful tools which you can find
in the following locations:

* gef (https://github.com/hugsy/gef) in /opt/gef/
* pwndbg (https://github.com/pwndbg/pwndbg) in /opt/pwndbg/
* gdbinit (https://github.com/gdbinit/Gdbinit) in /opt/gdbinit/
* pwntools (https://github.com/Gallopsled/pwntools)
* radare2 (http://www.radare.org/)

--[ More information ]-

For more information regarding individual wargames, visit
http://www.overthewire.org/wargames/

For support, questions or comments, contact us on discord or IRC.

Enjoy your stay!

WELCOME TO THE UPPERCASE SHELL
>> $0
$ cat /etc/bandit_pass/bandit33
tQdtbs5D5i2vJwkO8mEyYEyTL8izoeJ0
$
```

This give us the actual password.

Result: tQdtbs5D5i2vJwkO8mEyYEyTL8izoeJ0

Level 33 → Level 34

Objective

The objective of this level is to find the password.

Methodology

In this level we only have to put the command cat for read the README file and there was the congratulations for finishing the entire bandit program.

```
bandit33@bandit: ~
For your convenience we have installed a few useful tools which you can find
in the following locations:
* gef (https://github.com/hugsy/gef) in /opt/gef/
* pwndbg (https://github.com/pwndbg/pwndbg) in /opt/pwndbg/
* gdbinit (https://github.com/gdbinit/Gdbinit) in /opt/gdbinit/
* pwntools (https://github.com/Gallopsled/pwntools)
* radare2 (http://www.radare.org/)

--[ More information ]--

For more information regarding individual wargames, visit
http://www.overthewire.org/wargames/

For support, questions or comments, contact us on discord or IRC.

Enjoy your stay!

bandit33@bandit:~$ ls
README.txt
bandit33@bandit:~$ cat README.txt
Congratulations on solving the last level of this game!

At this moment, there are no more levels to play in this game. However, we are constantly working
on new levels and will most likely expand this game with more levels soon.
Keep an eye out for an announcement on our usual communication channels!
In the meantime, you could play some of our other wargames.

If you have an idea for an awesome new level, please let us know!
bandit33@bandit:~$
```

Bibliography

Canonical. (s/f-a). *Ubuntu Manpage: cat - concatenate files and print on the standard output.* Ubuntu.com. Recuperado el 4 de febrero de 2025, de <https://manpages.ubuntu.com/manpages/noble/man1/cat.1.html>

Canonical. (s/f-b). *Ubuntu Manpage: du - estimate file space usage.* Ubuntu.com. Recuperado el 4 de febrero de 2025, de <https://manpages.ubuntu.com/manpages/noble/man1/du.1.html>

Canonical. (s/f-c). *Ubuntu Manpage: find - search for files in a directory hierarchy.* Ubuntu.com. Recuperado el 4 de febrero de 2025, de <https://manpages.ubuntu.com/manpages/noble/man1/find.1.html>

Canonical. (s/f-d). *Ubuntu Manpage: ls - list directory contents.* Ubuntu.com. Recuperado

el 4 de febrero de 2025, de

<https://manpages.ubuntu.com/manpages/noble/man1/ls.1.html>

Canonical. (s/f-e). *Ubuntu Manpage: This manual page is part of the POSIX Programmer's*

Manual. The Linux implementation of this interface. Ubuntu.com. Recuperado el 4

de febrero de 2025, de

<https://manpages.ubuntu.com/manpages/noble/man1/cd.1posix.html>

Serna, I. (2021, marzo 24). *Tutorial del comando Grep – Cómo buscar un archivo en Linux*

y Unix con búsqueda recursiva. freecodecamp.org.

<https://www.freecodecamp.org/espanol/news/grep-command-tutorial-how-to-search-for-a-file-in-linux-and-unix-with-recursive-find/>