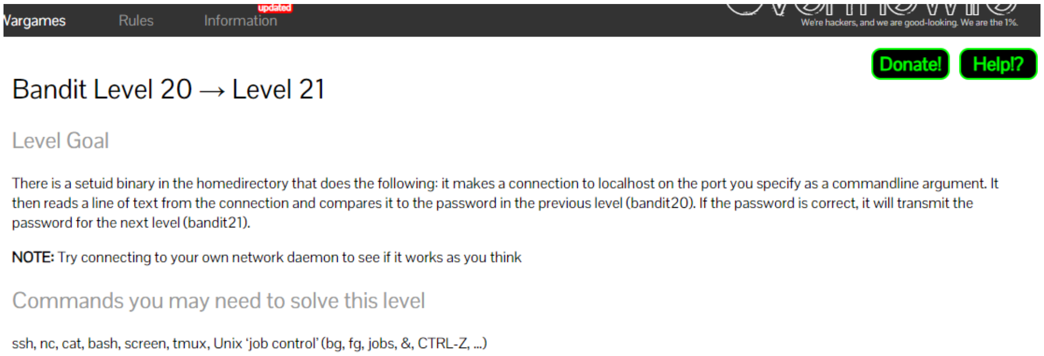
OverTheWire – Bandit:

Level 20-30

Today, I will play a war-game called **Bandit**. It has 34 levels. In this write-up I will play level 20-30.

The main objective is to access password files which will help us login into the next levels.

Level 20-21:



Bandit Level 20 → Level 21

Level Goal

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

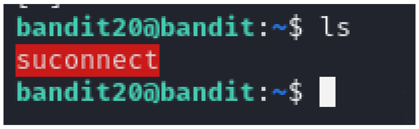
NOTE: Try connecting to your own network daemon to see if it works as you think

Commands you may need to solve this level

ssh, nc, cat, bash, screen, tmux, Unix 'job control' (bg, fg, jobs, &, CTRL-Z, ...)

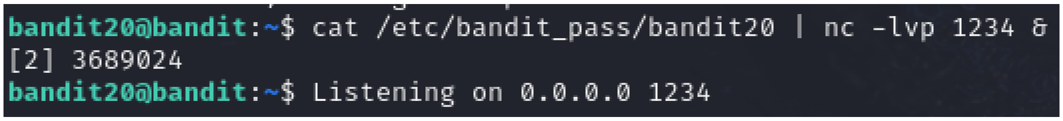
Here the hint shows us that there is a **setuid binary** in this level whose job is to make a connection to localhost on a port and it reads the password used to login as bandit20 and then send the password for the next level.

So lets use the `ls` command:



```
bandit20@bandit:~$ ls
suconnect
bandit20@bandit:~$
```

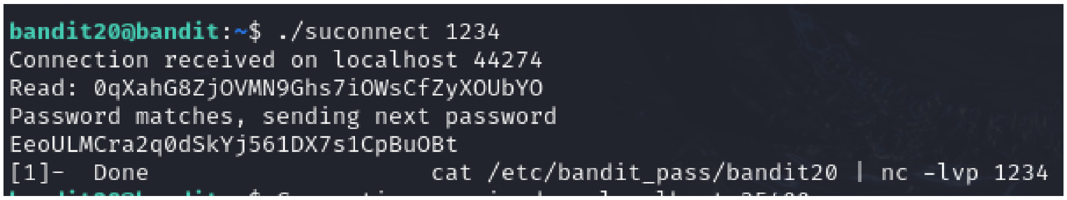
So here I need to connect it to a port, I used the netcat command. The command will read the password from the file and sends it on a network connection.



```
bandit20@bandit:~$ cat /etc/bandit_pass/bandit20 | nc -lvp 1234 &
[2] 3689024
bandit20@bandit:~$ Listening on 0.0.0.0 1234
```

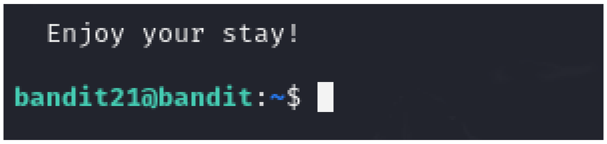
The `nc -lvp 1234` part sets up a Netcat listener on port 1234, waiting to receive the data.

Now I will use the `./suconnect` command to establish a command:



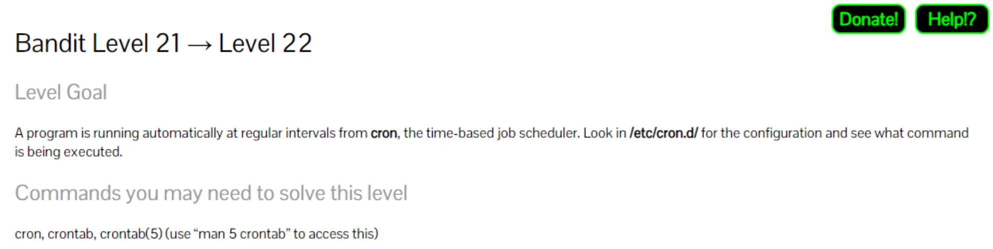
```
bandit20@bandit:~$ ./suconnect 1234
Connection received on localhost 44274
Read: 0qXahG8ZjOVMN9Ghs7iOWsCfZyXOUbYO
Password matches, sending next password
EeoULMCra2q0dSkYj561DX7s1CpBuOBt
[1]-  Done                    cat /etc/bandit_pass/bandit20 | nc -lvp 1234
```

So here we got the password of next level bandit21, so by using the ssh command we can login to bandit21 shell:



```
  Enjoy your stay!

bandit21@bandit:~$ 
```

Level 21-22:



Bandit Level 21 → Level 22

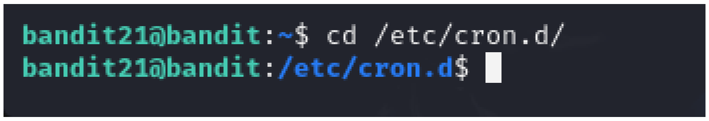
Level Goal

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

Commands you may need to solve this level

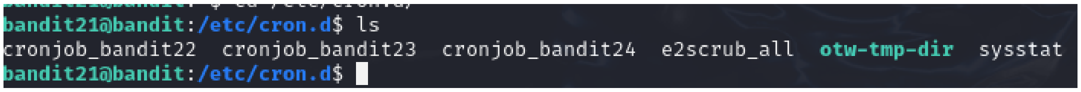
cron, crontab, crontab(5) (use “man 5 crontab” to access this)

Here we are hinted that there’s a cron script running and we have to enumerate **/etc/cron.d/** to get the password. So at first I will change the directory to **/etc/cron.d/ :**



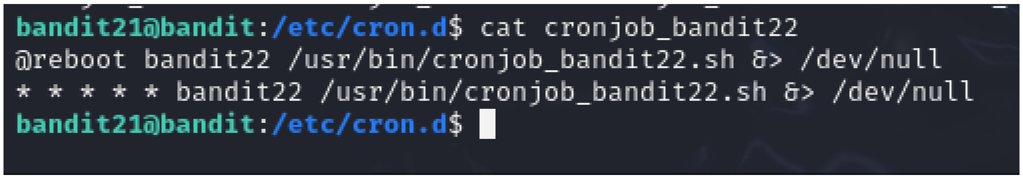
```
bandit21@bandit:~$ cd /etc/cron.d/
bandit21@bandit:/etc/cron.d$ 
```

Now we will use the ls command to list the files here:



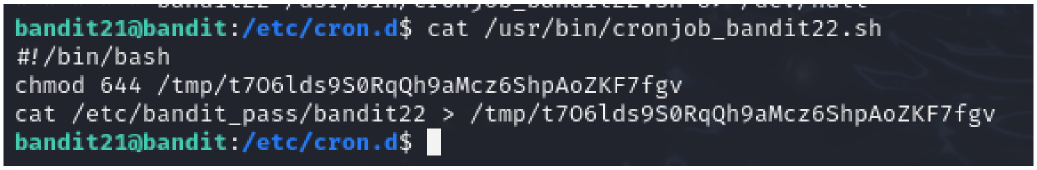
```
bandit21@bandit:/etc/cron.d$ ls
cronjob_bandit22  cronjob_bandit23  cronjob_bandit24  e2scrub_all  otw-tmp-dir  sysstat
bandit21@bandit:/etc/cron.d$ 
```

As the next level is bandit22 so I will read the file named **cronjob_bandit22** by using cat command:



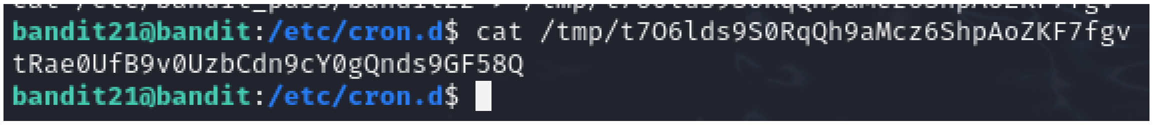
```
bandit21@bandit:/etc/cron.d$ cat cronjob_bandit22
@reboot bandit22 /usr/bin/cronjob_bandit22.sh &> /dev/null
* * * * * bandit22 /usr/bin/cronjob_bandit22.sh &> /dev/null
bandit21@bandit:/etc/cron.d$ 
```

Here it displays a script **/usr/bin/cronjob_bandit22/sh** so maybe I should read that script to find the password:



```
bandit21@bandit:/etc/cron.d$ cat /usr/bin/cronjob_bandit22.sh
#!/bin/bash
chmod 644 /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv
cat /etc/bandit_pass/bandit22 > /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv
bandit21@bandit:/etc/cron.d$ 
```

So here it shows that the password for bandit22 is in **tmp** directory so lets read this directory to read the password:

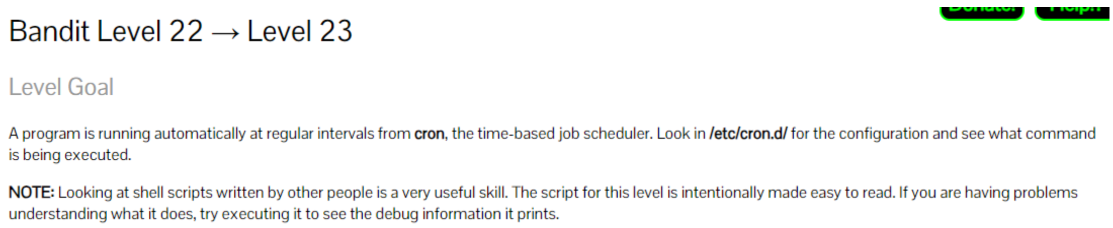


```
bandit21@bandit:/etc/cron.d$ cat /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv
tRae0UfB9v0UzbCdn9cY0gQnds9GF58Q
bandit21@bandit:/etc/cron.d$ 
```

And yes!! Here’s the password for bandit22, lets access the bandit23 shell by using the ssh command:

```
  Enjoy your stay!

bandit22@bandit:~$ 
```

Level 22-23:



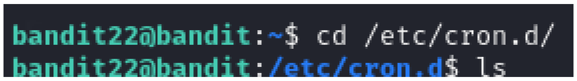
Bandit Level 22 → Level 23

Level Goal

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

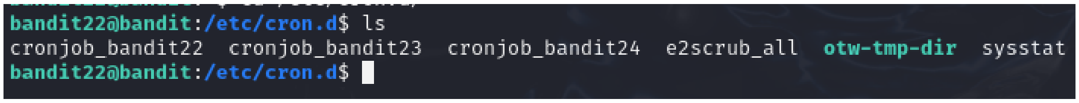
NOTE: Looking at shell scripts written by other people is a very useful skill. The script for this level is intentionally made easy to read. If you are having problems understanding what it does, try executing it to see the debug information it prints.

Here we are hinted that there’s a cron script running and we have to enumerate **/etc/cron.d/** to get the password. So at first I will change the directory to **/etc/cron.d/ :**



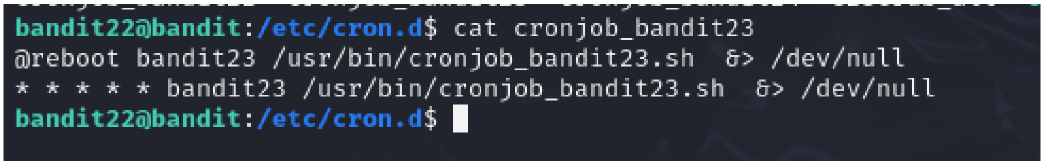
```
bandit22@bandit:~$ cd /etc/cron.d/
bandit22@bandit:/etc/cron.d$ ls
```

Now we will use the ls command to list the files here:



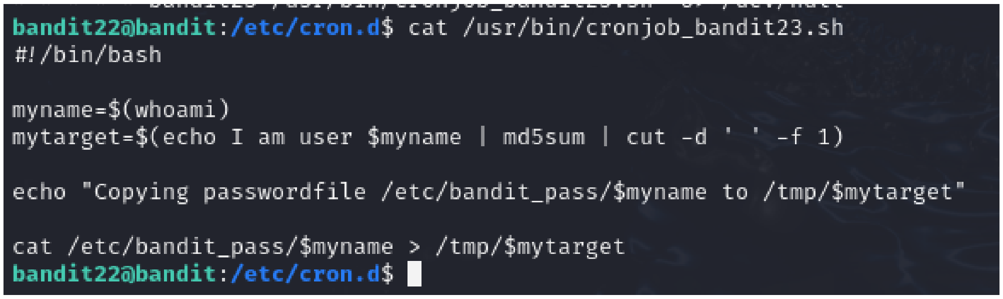
```
bandit22@bandit:/etc/cron.d$ ls
cronjob_bandit22  cronjob_bandit23  cronjob_bandit24  e2scrub_all  otw-tmp-dir  sysstat
bandit22@bandit:/etc/cron.d$ 
```

As the next level is bandit23 so I will read the file named **cronjob_bandit23** by using cat command:



```
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$ 
```

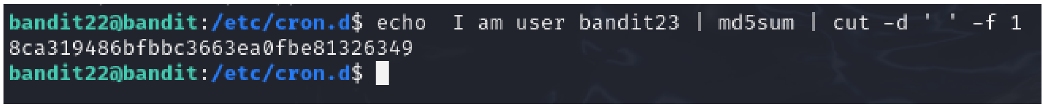
Here it displays a script **/usr/bin/cronjob_bandit23.sh** so maybe I should read that script to find the password:



```
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$ 
```

Here the script shows a variable named **myname** and it results in the command **whoami** and it returns bandit22, “my name is bandit22” and it is **md5** encrypted.

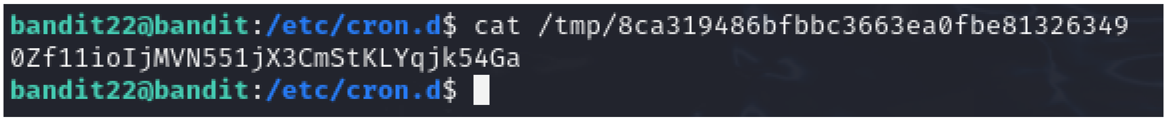
Now to get the password I will set the variable **myname** to **bandit23**:



```
bandit22@bandit:/etc/cron.d$ echo  I am user bandit23 | md5sum | cut -d ' ' -f 1
8ca319486bfbbc3663ea0fbe81326349
bandit22@bandit:/etc/cron.d$ 
```

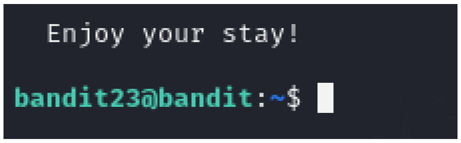
Harrum Fatima

This gave us the hash value in **tmp** directory:



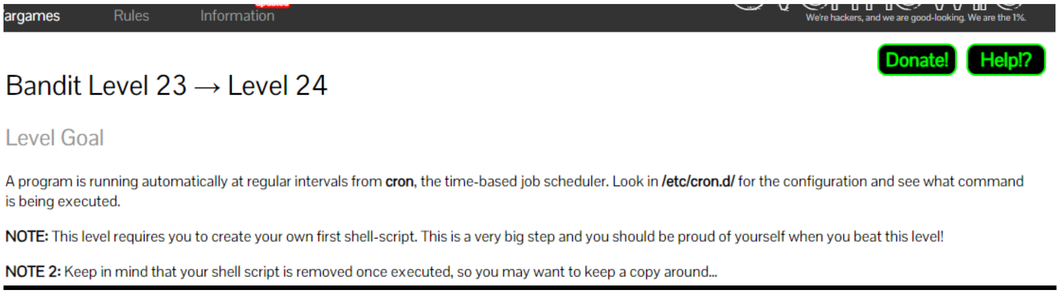
```
bandit22@bandit:/etc/cron.d$ cat /tmp/8ca319486bfbbc3663ea0fbe81326349
0Zf11ioIjMVN551jX3CmStKLYqjk54Ga
bandit22@bandit:/etc/cron.d$ 
```

Here we got the password of bandit 23 and now by ssh command I am going to access bandit23 shell:



```
  Enjoy your stay!

bandit23@bandit:~$ 
```

Level 23-24:



Bandit Level 23 → Level 24

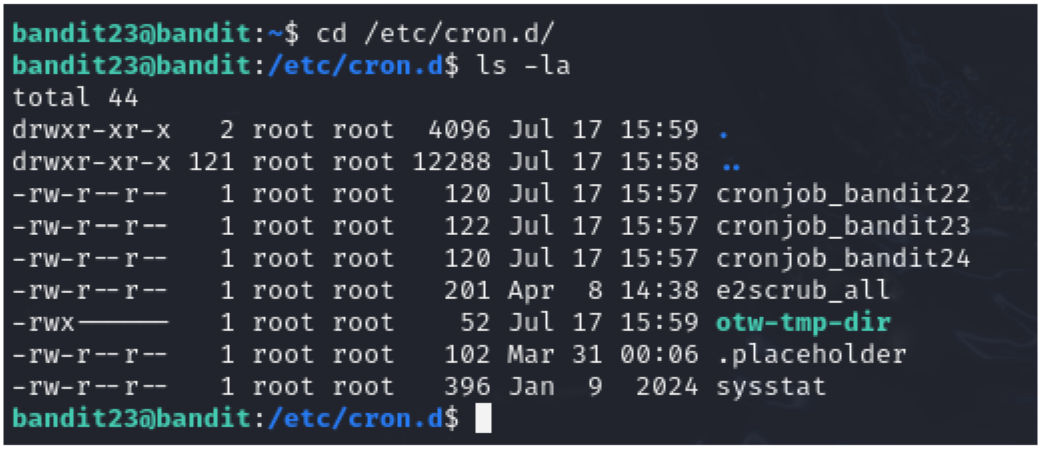
Level Goal

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

NOTE: This level requires you to create your own first shell-script. This is a very big step and you should be proud of yourself when you beat this level!

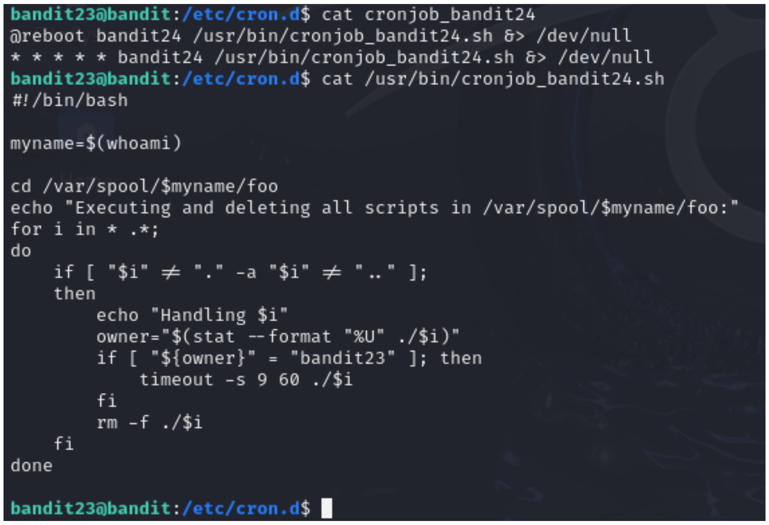
NOTE 2: Keep in mind that your shell script is removed once executed, so you may want to keep a copy around…

Here we are hinted that there is a **cron** script running and we need to enumerate **/etc/cron.d/** for the password.So I will firstly traverse in that directory and will then list the files in that directory:



```
bandit23@bandit:~$ cd /etc/cron.d/
bandit23@bandit:/etc/cron.d$ ls -la
total 44
drwxr-xr-x   2 root root  4096 Jul 17 15:59 .
drwxr-xr-x 121 root root 12288 Jul 17 15:58 ..
-rw-r--r--   1 root root   120 Jul 17 15:57 cronjob_bandit22
-rw-r--r--   1 root root   122 Jul 17 15:57 cronjob_bandit23
-rw-r--r--   1 root root   120 Jul 17 15:57 cronjob_bandit24
-rw-r--r--   1 root root   201 Apr  8 14:38 e2scrub_all
-rwx------   1 root root    52 Jul 17 15:59 otw-tmp-dir
-rw-r--r--   1 root root   102 Mar 31 00:06 .placeholder
-rw-r--r--   1 root root   396 Jan  9  2024 sysstat
bandit23@bandit:/etc/cron.d$ 
```

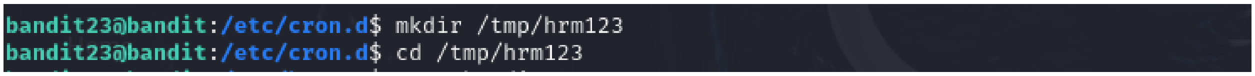
As the next level is bandit24 so we will read **cronjob_bandit24** by using cat command:



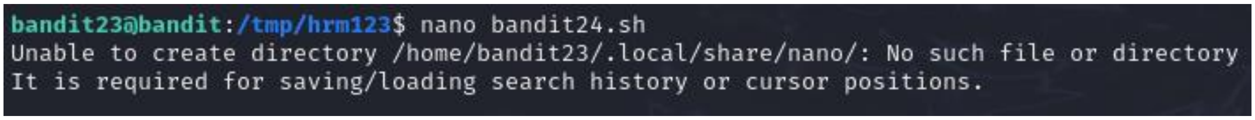
```
bandit23@bandit:/etc/cron.d$ cat cronjob_bandit24
@reboot bandit24 /usr/bin/cronjob_bandit24.sh &> /dev/null
* * * * * bandit24 /usr/bin/cronjob_bandit24.sh &> /dev/null
bandit23@bandit:/etc/cron.d$ cat /usr/bin/cronjob_bandit24.sh
#!/bin/bash

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 9 60 ./$i
        fi
        rm -f ./$i
    fi
done

bandit23@bandit:/etc/cron.d$ 
```

The script shows a variable **myname** which is an output of command **whoami**. The script will first change the names directory to **/var/spool/** and then will execute the files and after execution it will delete all the files.

Now for the password for the next directory i will create a script of my own so i can put it inside the /var/spool that will cat the password file from the /etc/bandit_pass/bandit24. Then i will save the file with the name of the next user in order to run the file as a cron job successfully.

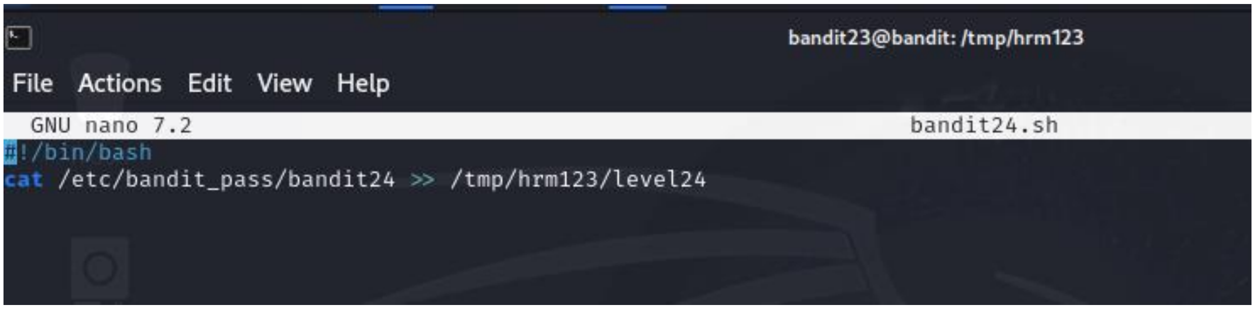


```
bandit23@bandit:/etc/cron.d$ mkdir /tmp/hrm123
bandit23@bandit:/etc/cron.d$ cd /tmp/hrm123
```



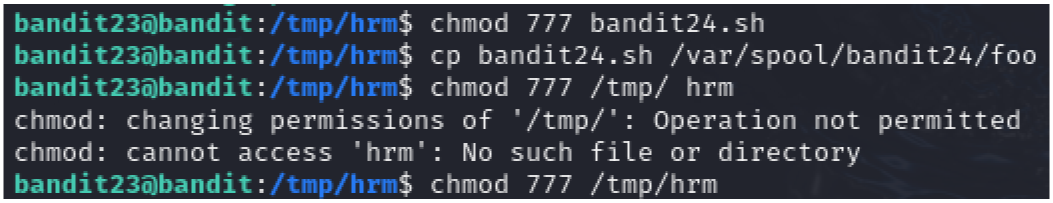
```
bandit23@bandit:/tmp/hrm123$ nano bandit24.sh
Unable to create directory /home/bandit23/.local/share/nano/: No such file or directory
It is required for saving/loading search history or cursor positions.
```

So now I had created a file using nano, i will write the script that will read the password from the /etc/bandit_pass and writes in the file inside the directory I just created.



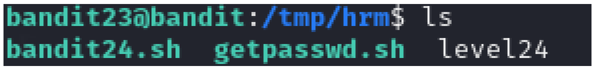
```
#!/bin/bash
cat /etc/bandit_pass/bandit24 >> /tmp/hrm123/level24
```

Now to execute this file I will give it proper read and wrote permissions by **chmod** command.



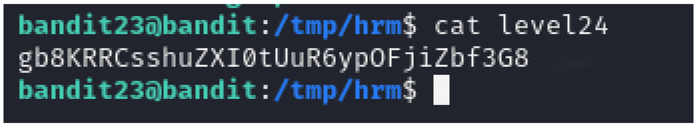
```
bandit23@bandit:/tmp/hrm$ chmod 777 bandit24.sh
bandit23@bandit:/tmp/hrm$ cp bandit24.sh /var/spool/bandit24/foo
bandit23@bandit:/tmp/hrm$ chmod 777 /tmp/ hrm
chmod: changing permissions of '/tmp/': Operation not permitted
chmod: cannot access 'hrm': No such file or directory
bandit23@bandit:/tmp/hrm$ chmod 777 /tmp/hrm
```

So now I will list the files to ensure level24 file:



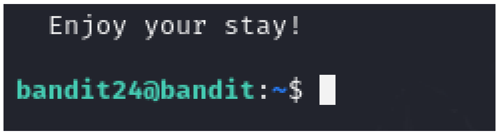
```
bandit23@bandit:/tmp/hrm$ ls
bandit24.sh  getpasswd.sh  level24
```

And now by cat command I will read that file:



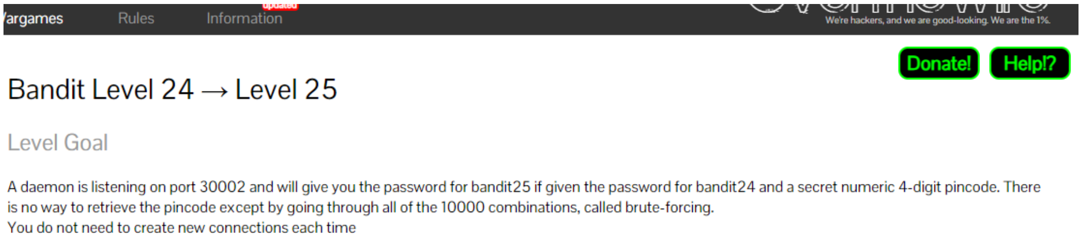
```
bandit23@bandit:/tmp/hrm$ cat level24
gb8KRRCsshuZXI0tUuR6ypOFjiZbf3G8
bandit23@bandit:/tmp/hrm$ 
```

And yes!! We got the password for level 24, now by using ssh command I can access bandit24 shell:



```
  Enjoy your stay!

bandit24@bandit:~$ 
```

Level 24-25:



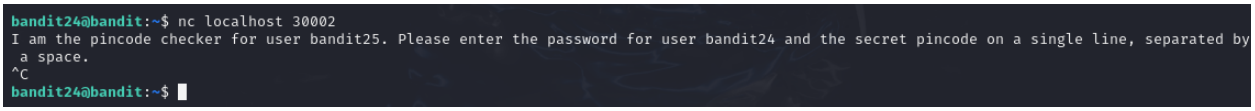
Bandit Level 24 → Level 25

Level Goal

A daemon is listening on port 30002 and will give you the password for bandit25 if given the password for bandit24 and a secret numeric 4-digit pincode. There is no way to retrieve the pincode except by going through all of the 10000 combinations, called brute-forcing.
You do not need to create new connections each time

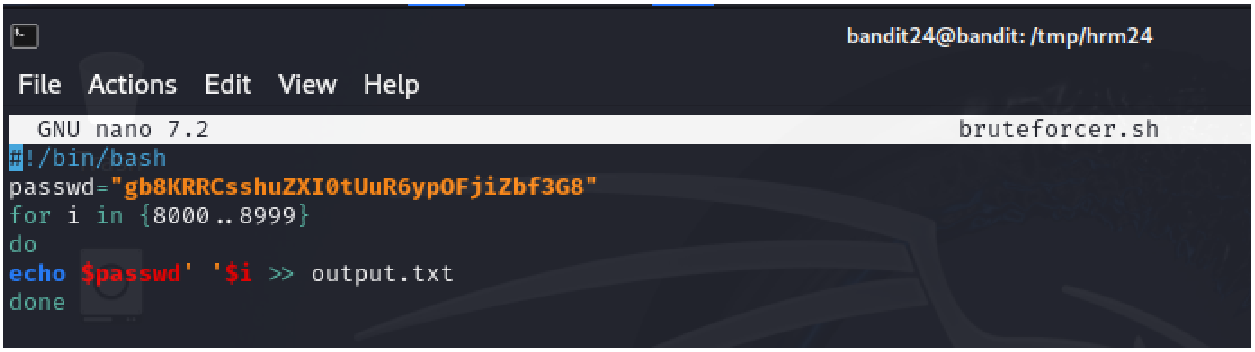
Here it shows that a background process is running and is listening at port **30002** and we will get the password for the next level, Also We will also have to provide a 4-digit secret passcode which will have to Bruteforce. Now to Bruteforce i will start by creating a dictionary.

So lets start, firstly I will connect the host to port 30002 by using netcat:



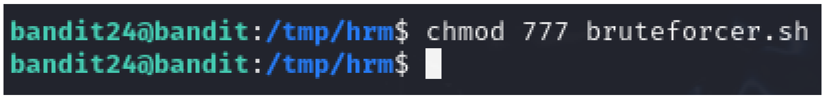
```
bandit24@bandit:~$ nc localhost 30002
I am the pincode checker for user bandit25. Please enter the password for user bandit24 and the secret pincode on a single line, separated by
 a space.
^C
bandit24@bandit:~$ 
```

But ughh, so I will leave this here and now I will create a directory and here I will make a dictionary file and here is the script:



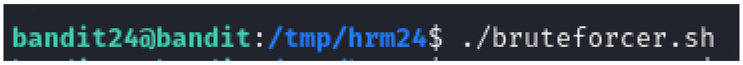
```
#!/bin/bash
passwd="gb8KRRCsshuZXI0tUuR6ypOFjiZbf3G8"
for i in {8000..8999}
do
echo $passwd' '$i >> output.txt
done
```

So now I will give this script proper permissions:



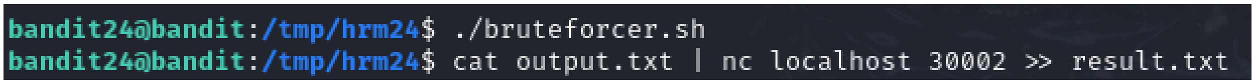
```
bandit24@bandit:/tmp/hrm$ chmod 777 bruteforcer.sh
bandit24@bandit:/tmp/hrm$ 
```

After giving permissions I will run that script:



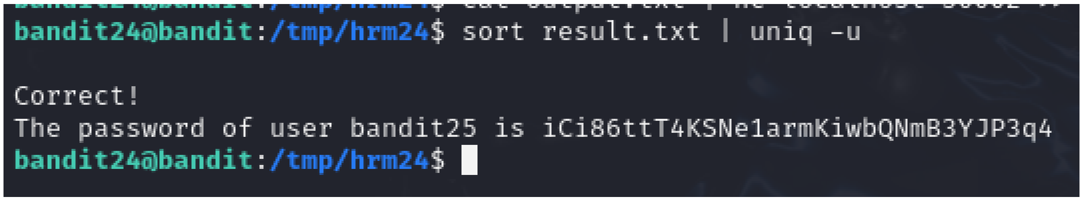
```
bandit24@bandit:/tmp/hrm24$ ./bruteforcer.sh
```

Now to apply bruteforce I will use piping which will first read the password we created in **output.txt** and will feed its output at 30002 and will feed this output into **result.txt** :



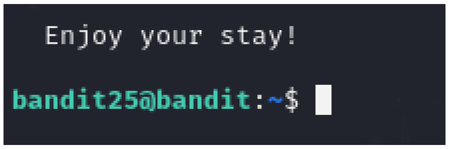
```
bandit24@bandit:/tmp/hrm24$ ./bruteforcer.sh
bandit24@bandit:/tmp/hrm24$ cat output.txt | nc localhost 30002 >> result.txt
```

Now using the sort command combined with the uniq command, I got the password:



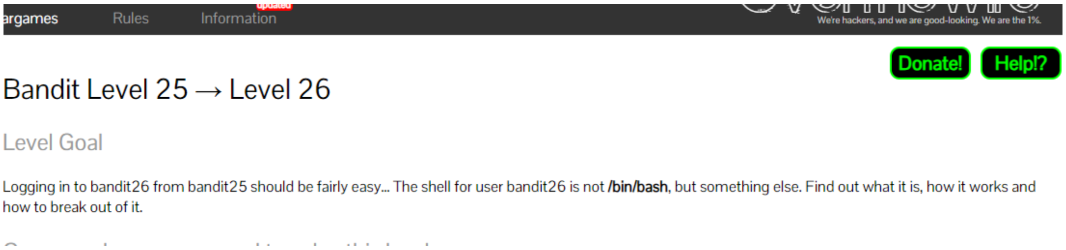
```
bandit24@bandit:/tmp/hrm24$ sort result.txt | uniq -u

Correct!
The password of user bandit25 is iCi86ttT4KSNe1armKiwbQNmB3YJP3q4
bandit24@bandit:/tmp/hrm24$ 
```

Now by using ssh we can acess bandit25 shell:



```
  Enjoy your stay!

bandit25@bandit:~$ 
```

Level 25-26:

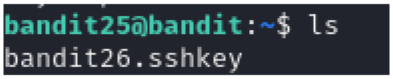


Bandit Level 25 → Level 26

Level Goal

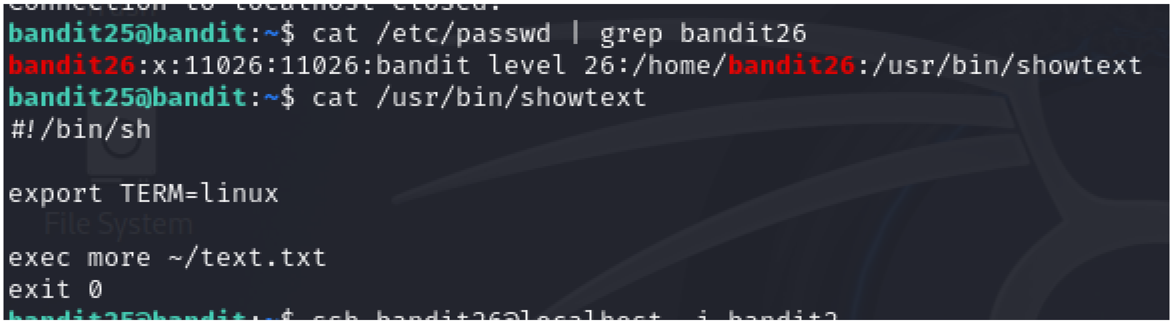
Logging in to bandit26 from bandit25 should be fairly easy… The shell for user bandit26 is not **/bin/bash**, but something else. Find out what it is, how it works and how to break out of it.

This level states that bandit26 shell is not **/bin/bash** and we have to figure it out. So lets use the ls command first to list the directories:



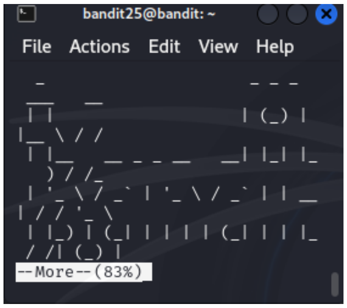
```
bandit25@bandit:~$ ls
bandit26.sshkey
```

a session was generated but it displayed a pattern as below and then the session was closed. After a bit enumeration, it hit us to check the /etc/passwd file. As this was a machine with lots of users so i used the grep command to get a refined result for the bandit26 user. It gave us a file called showtext. We read the file showtext using the cat command. It shows us that ‘**more’** is used with the text file that shows us the pattern we saw before. Now, this gave us an idea that we need to provoke the more command. To do this we will have to decrease the size of the terminal so that it can’t display that pattern.



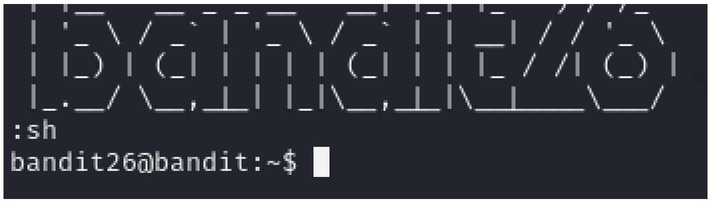
```
connection to localhost closed.
bandit25@bandit:~$ cat /etc/passwd | grep bandit26
bandit26:x:11026:11026:bandit level 26:/home/bandit26:/usr/bin/showtext
bandit25@bandit:~$ cat /usr/bin/showtext
#!/bin/sh

export TERM=linux

exec more ~/text.txt
exit 0
bandit25@bandit:~$ ssh bandit26@localhost -i bandit2
```

So I decreased the shell size and When it displayed **more** I entered the following command to invoke a shell here:



```
--More--(83%)
```

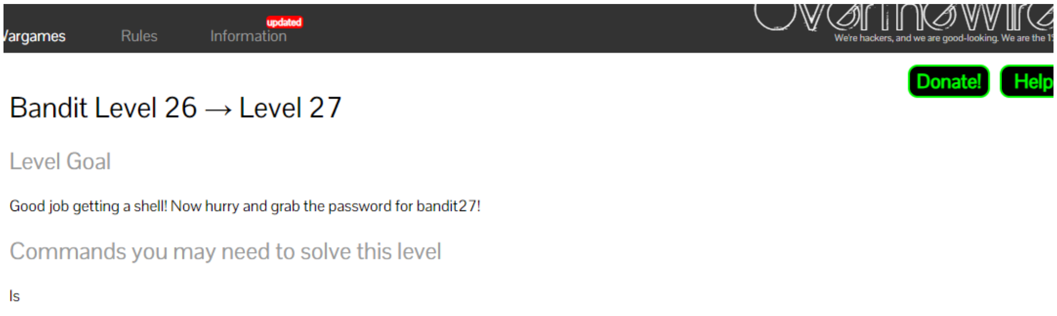
```
:set shell=/bin/bash
```



```
:sh
bandit26@bandit:~$
```

And here by entering **:ssh** I was in bandit26 shell.

Level 26-27:



Bandit Level 26 → Level 27

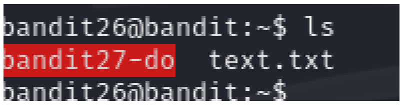
Level Goal

Good job getting a shell! Now hurry and grab the password for bandit27!

Commands you may need to solve this level

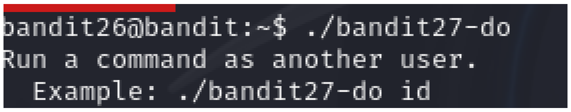
ls

In this level we are not given any hints, so lets see the files:



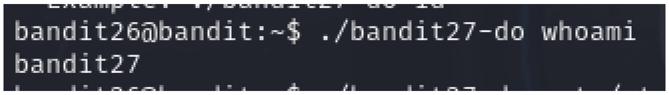
```
bandit26@bandit:~$ ls
bandit27-do  text.txt
bandit26@bandit:~$
```

Here we got a script so lets execute this script:



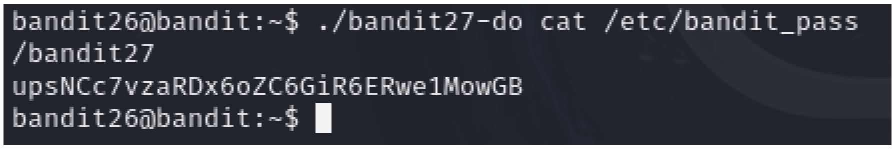
```
bandit26@bandit:~$ ./bandit27-do
Run a command as another user.
  Example: ./bandit27-do id
```

It shows that we have to run this script as a user so lets execute this as a user:



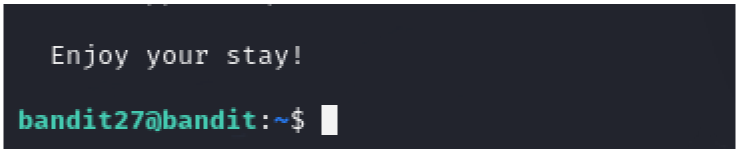
```
bandit26@bandit:~$ ./bandit27-do whoami
bandit27
```

Lets read the password file located at /etc/bandit_pass/bandit27:



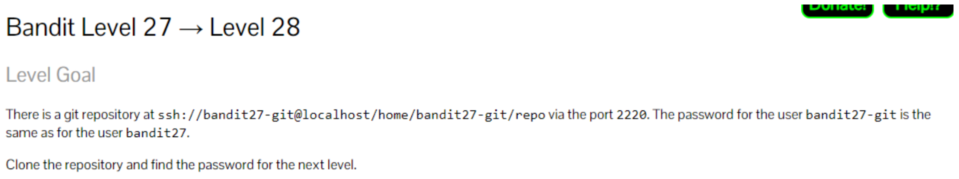
```
bandit26@bandit:~$ ./bandit27-do cat /etc/bandit_pass
/bandit27
upsNCc7vzaRDx6oZC6GiR6ERwe1MowGB
bandit26@bandit:~$ 
```

So here's the password for bandit27, lets use the ssh command to access bandit27 shell:



```
  Enjoy your stay!

bandit27@bandit:~$ 
```

Level 27-28:



Bandit Level 27 → Level 28

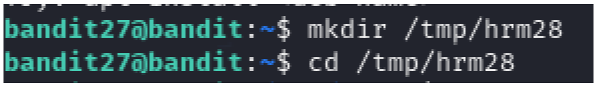
Level Goal

There is a git repository at `ssh://bandit27-git@localhost/home/bandit27-git/repo` via the port `2220`. The password for the user `bandit27-git` is the same as for the user `bandit27`.

Clone the repository and find the password for the next level.

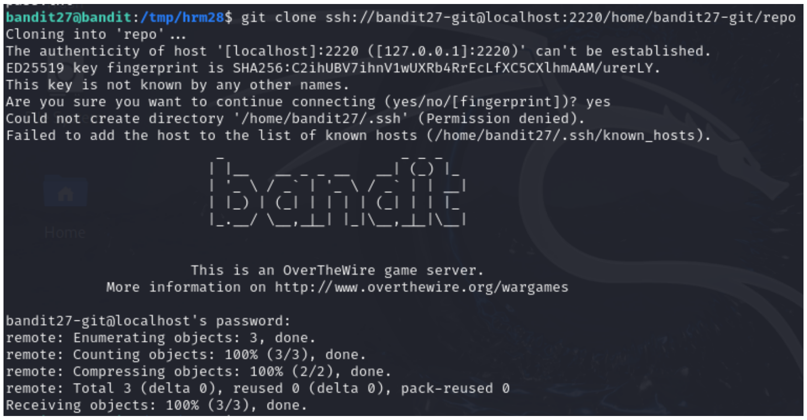
In this level there is a git repository and the password for that repository is the same password that was used to login in bandit27. We have to clone the repository.

So to clone this I made a tmp directory:



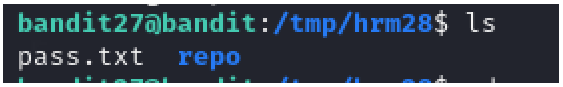
```
bandit27@bandit:~$ mkdir /tmp/hrm28
bandit27@bandit:~$ cd /tmp/hrm28
```

Now we need to have the write permission to clone a repository. So, we create a directory in the tmp directory.



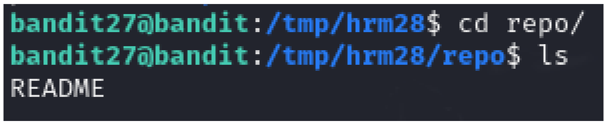
```
bandit27@bandit:/tmp/hrm28$ git clone ssh://bandit27-git@localhost:2220/home/bandit27-git/repo
Cloning into 'repo'...
The authenticity of host '[localhost]:2220 ([127.0.0.1]:2220)' can't be established.
ED25519 key fingerprint is SHA256:C2ihUBV7ihnV1wUXRb4RrEcLfXC5CXlhmAAM/urerLY.
This key is not 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), done.
```

After cloning let’s list all the file in the repo.



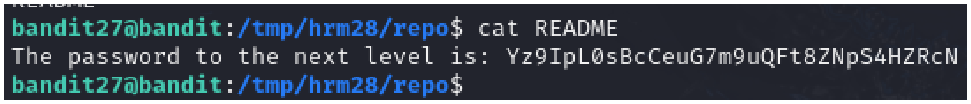
```
bandit27@bandit:/tmp/hrm28$ ls
pass.txt  repo
```

Lets change the directory to repo and see the files tjere:



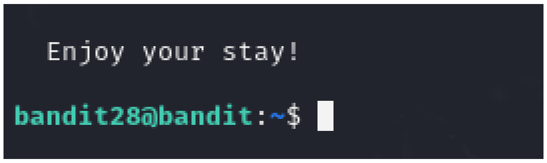
```
bandit27@bandit:/tmp/hrm28$ cd repo/
bandit27@bandit:/tmp/hrm28/repo$ ls
README
```

We got a README file and found the password for the next level.



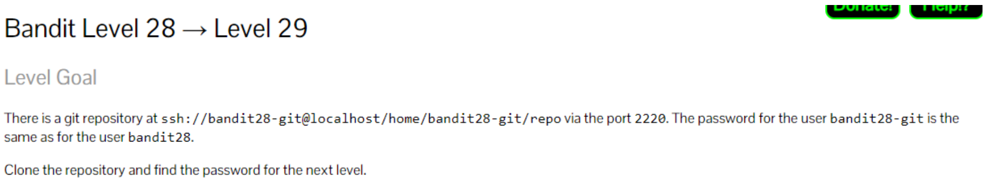
```
bandit27@bandit:/tmp/hrm28/repo$ cat README
The password to the next level is: Yz9IpL0sBcCeuG7m9uQFt8ZNpS4HZRcN
bandit27@bandit:/tmp/hrm28/repo$
```

Now by using ssh command I can access the bandit28 shell:



```
  Enjoy your stay!

bandit28@bandit:~$ 
```

Level 28-29:



Bandit Level 28 → Level 29

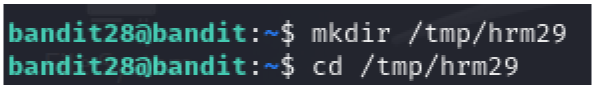
Level Goal

There is a git repository at `ssh://bandit28-git@localhost/home/bandit28-git/repo` via the port `2220`. The password for the user `bandit28-git` is the same as for the user `bandit28`.

Clone the repository and find the password for the next level.

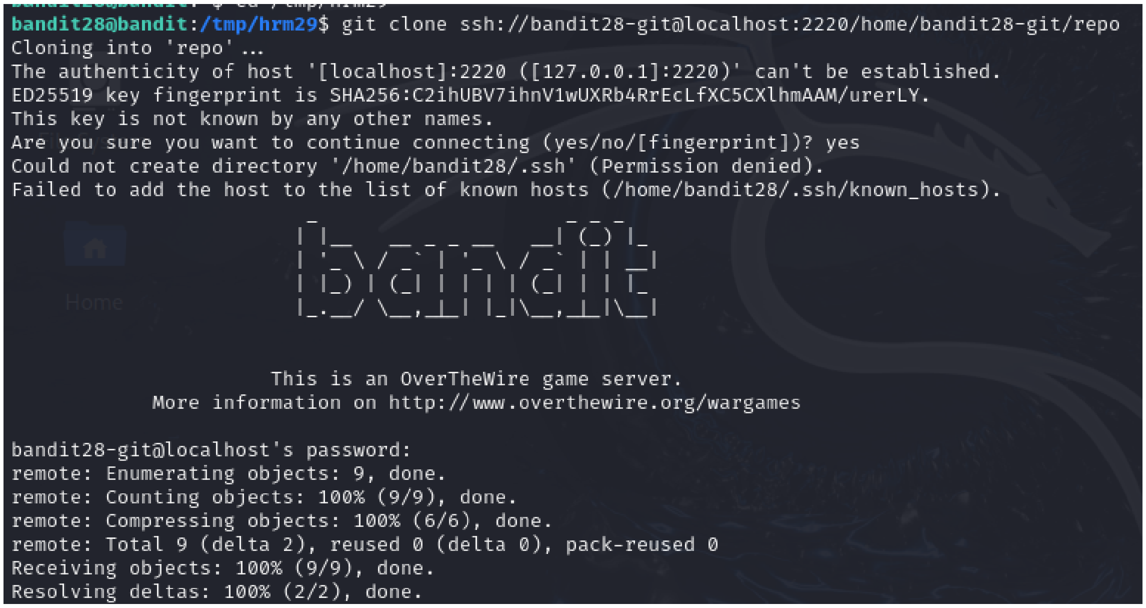
This level is pretty same to the previous level.

In this level, we are hinted that there is a git repository and the password for that repository is the same password that was used to login in as user bandit28. We are required to clone the repository. Now we need to have the write permission to clone a repository. So, we create a directory in the tmp directory.



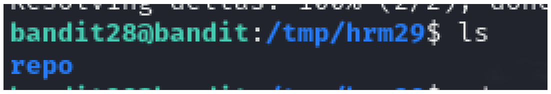
```
bandit28@bandit:~$ mkdir /tmp/hrm29
bandit28@bandit:~$ cd /tmp/hrm29
```

Now we will clone the repository inside this directory.



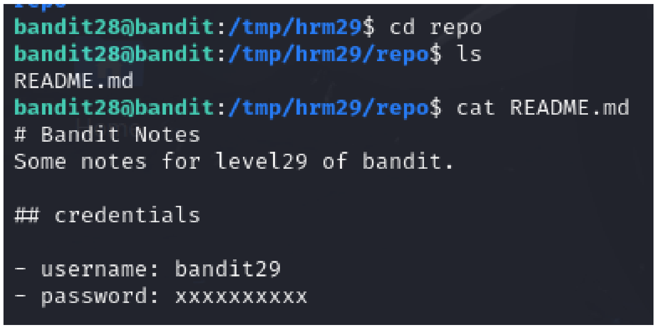
```
bandit28@bandit:/tmp/hrm29$ git clone ssh://bandit28-git@localhost:2220/home/bandit28-git/repo
Cloning into 'repo'...
The authenticity of host '[localhost]:2220 ([127.0.0.1]:2220)' can't be established.
ED25519 key fingerprint is SHA256:C2ihUBV7ihnV1wUXRb4RrEcLfXC5CXlhmAAM/urerLY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Could not create directory '/home/bandit28/.ssh' (Permission denied).
Failed to add the host to the list of known hosts (/home/bandit28/.ssh/known_hosts).
                         _                     _ _ _
                        | |__   __ _ _ __   __| (_) |_
                        | '_ \ / _` | '_ \ / _` | | __|
                        | |_) | (_| | | | | (_| | | |_
                        |_.__/ \__,_|_| |_|\__,_|_|\__|


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

bandit28-git@localhost's password:
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 9 (delta 2), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (9/9), done.
Resolving deltas: 100% (2/2), done.
```

After cloning let’s list all the file in the repo.



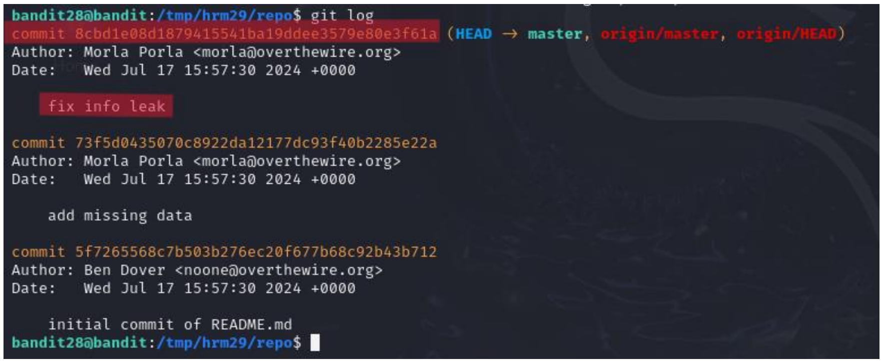
```
bandit28@bandit:/tmp/hrm29$ ls
repo
```

Lets change the directory to repo and see the files there:



```
bandit28@bandit:/tmp/hrm29$ cd repo
bandit28@bandit:/tmp/hrm29/repo$ ls
README.md
bandit28@bandit:/tmp/hrm29/repo$ cat README.md
# Bandit Notes
Some notes for level29 of bandit.

## credentials

- username: bandit29
- password: xxxxxxxxxx
```

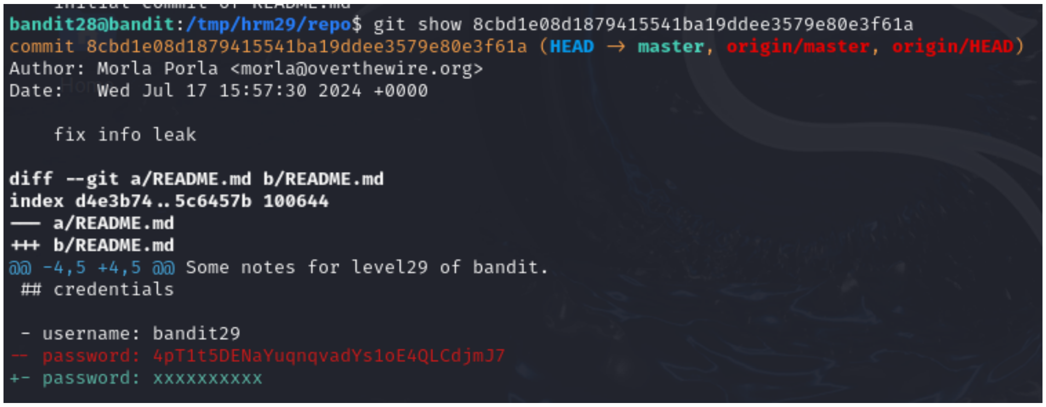
So here we got a readme.md file but the password is not shown, maybe it was removed, But whenever a change is made in git a log entry is created. So lets check that log:



```
bandit28@bandit:/tmp/hrm29/repo$ git log
commit 8cbd1e08d1879415541ba19ddee3579e80e3f61a (HEAD → master, origin/master, origin/HEAD)
Author: Morla Porla <morla@overthewire.org>
Date:   Wed Jul 17 15:57:30 2024 +0000

    fix info leak

commit 73f5d0435070c8922da12177dc93f40b2285e22a
Author: Morla Porla <morla@overthewire.org>
Date:   Wed Jul 17 15:57:30 2024 +0000

    add missing data

commit 5f7265568c7b503b276ec20f677b68c92b43b712
Author: Ben Dover <noone@overthewire.org>
Date:   Wed Jul 17 15:57:30 2024 +0000

    initial commit of README.md
bandit28@bandit:/tmp/hrm29/repo$
```

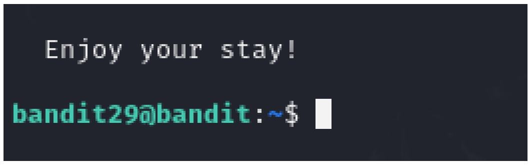
So yes here we got a commit **fix info leak** so lets check that commit:

Harrum Fatima



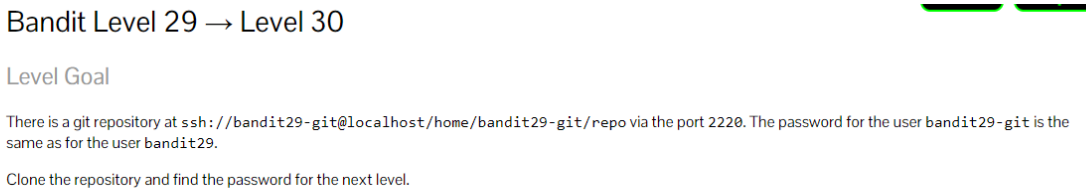
```
    Initial commit of README.md
bandit28@bandit:/tmp/hrm29/repo$ git show 8cbd1e08d1879415541ba19ddee3579e80e3f61a
commit 8cbd1e08d1879415541ba19ddee3579e80e3f61a (HEAD → master, origin/master, origin/HEAD)
Author: Morla Porla <morla@overthewire.org>
Date:   Wed Jul 17 15:57:30 2024 +0000

    fix info leak

diff --git a/README.md b/README.md
index d4e3b74..5c6457b 100644
--- a/README.md
+++ b/README.md
@@ -4,5 +4,5 @@ Some notes for level29 of bandit.
 ## credentials

 - username: bandit29
-- password: 4pT1t5DENaYuqnqvadYs1oE4QLCdjmJ7
+- password: xxxxxxxxxx
```

So yahoo!! We got the password for bandit 29, lets access the shell by using ssh command:



```
  Enjoy your stay!

bandit29@bandit:~$ 
```

Level 29-30:



Bandit Level 29 → Level 30

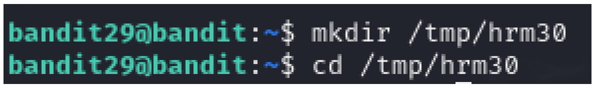
Level Goal

There is a git repository at `ssh://bandit29-git@localhost/home/bandit29-git/repo` via the port `2220`. The password for the user `bandit29-git` is the same as for the user `bandit29`.

Clone the repository and find the password for the next level.

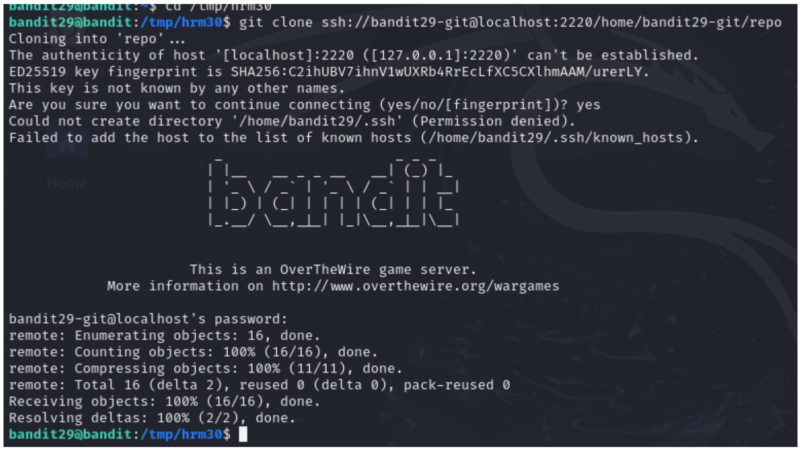
This level is pretty same to the previous levels.

In this level, we are hinted that there is a git repository and the password for that repository is the same password that was used to login in as user bandit29. We are required to clone the repository. Now we need to have the write permission to clone a repository. So, we create a directory in the tmp directory.



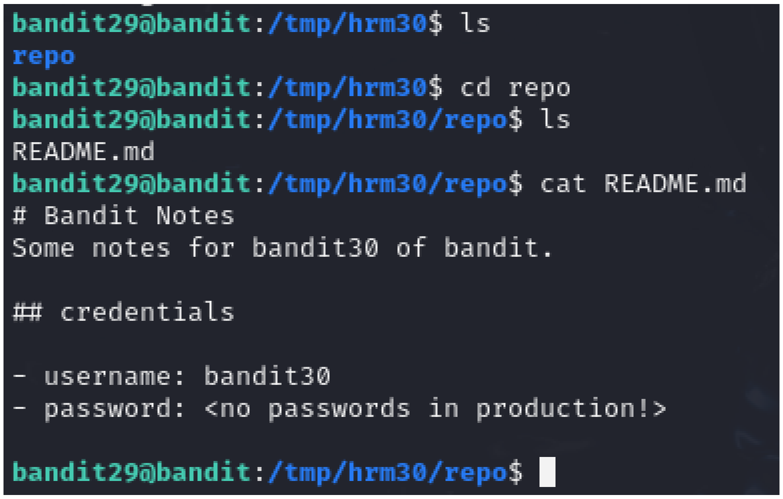
```
bandit29@bandit:~$ mkdir /tmp/hrm30
bandit29@bandit:~$ cd /tmp/hrm30
```

Now we will clone the repository inside this directory.



```
bandit29@bandit:~$ cd /tmp/hrm30
bandit29@bandit:/tmp/hrm30$ 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: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/bandit29/.ssh' (Permission denied).
Failed to add the host to the list of known hosts (/home/bandit29/.ssh/known_hosts).
                         _                     _ _ _
                        | |__   __ _ _ __   __| (_) |_
                        | '_ \ / _` | '_ \ / _` | | __|
                        | |_) | (_| | | | | (_| | | |_
                        |_.__/ \__,_|_| |_|\__,_|_|\__|


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

bandit29-git@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/hrm30$ 
```

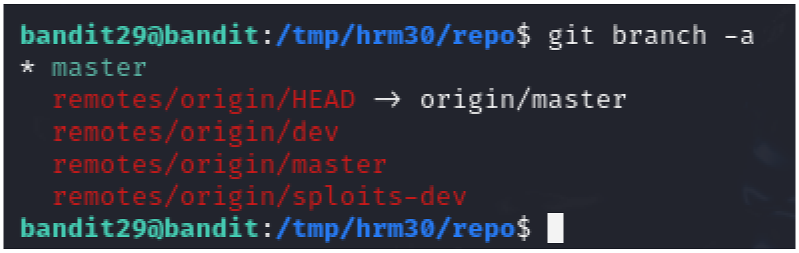
After cloning the files, lets list the files in the repo directory, Here we got a **README.md** file:



```
bandit29@bandit:/tmp/hrm30$ ls
repo
bandit29@bandit:/tmp/hrm30$ cd repo
bandit29@bandit:/tmp/hrm30/repo$ ls
README.md
bandit29@bandit:/tmp/hrm30/repo$ cat README.md
# Bandit Notes
Some notes for bandit30 of bandit.

## credentials

- username: bandit30
- password: <no passwords in production!>

bandit29@bandit:/tmp/hrm30/repo$ 
```

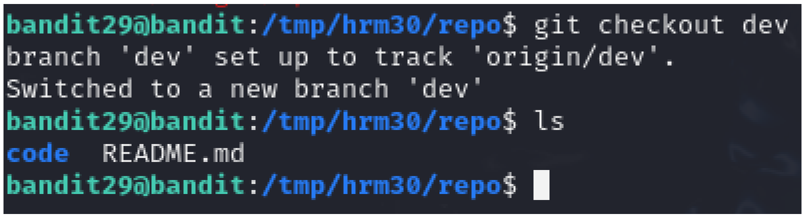
But there's no password displayed in production, hmmmm!! Lets enumerate the git.

Firstly I will list all the branches of git:



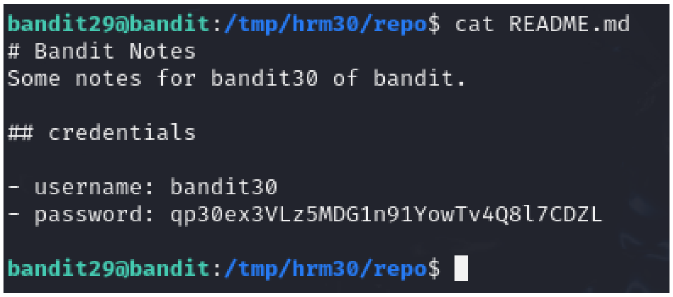
```
bandit29@bandit:/tmp/hrm30/repo$ git branch -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/dev
  remotes/origin/master
  remotes/origin/sploits-dev
bandit29@bandit:/tmp/hrm30/repo$ 
```

Here we got a **dev** branch, lets check out this branch for the password:



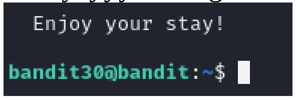
```
bandit29@bandit:/tmp/hrm30/repo$ git checkout dev
branch 'dev' set up to track 'origin/dev'.
Switched to a new branch 'dev'
bandit29@bandit:/tmp/hrm30/repo$ ls
code  README.md
bandit29@bandit:/tmp/hrm30/repo$ 
```

Here I found a **README.md** file, I hope the password is in this file, lets read it out:



```
bandit29@bandit:/tmp/hrm30/repo$ cat README.md
# Bandit Notes
Some notes for bandit30 of bandit.

## credentials

- username: bandit30
- password: qp30ex3VLz5MDG1n91YowTv4Q8l7CDZL

bandit29@bandit:/tmp/hrm30/repo$ 
```

And yayyy!! We got the password for bandit level 30, lets access bandit30 shell by using ssh command:



```
  Enjoy your stay!

bandit30@bandit:~$ 
```

Harrum Fatima

Harrum Fatima