**Proof Of Concept:- Overthewire: Bandit**

**Submitted to:** Digisuraksha Parhari Foundation

**Members:** Riya Bhutal (248) ,

Gautam Poojari (251)

***Level 0 → Level 1***

Tools Used: ssh, ls, cat

Objective: Connect to the Bandit server and find the password for the next level.

Commands Used: `cat readme`

Steps:

1. SSH to bandit.labs.overthewire.org on port 2220 with username bandit0 and password bandit0

2. Used `ls` to list files and found `readme`

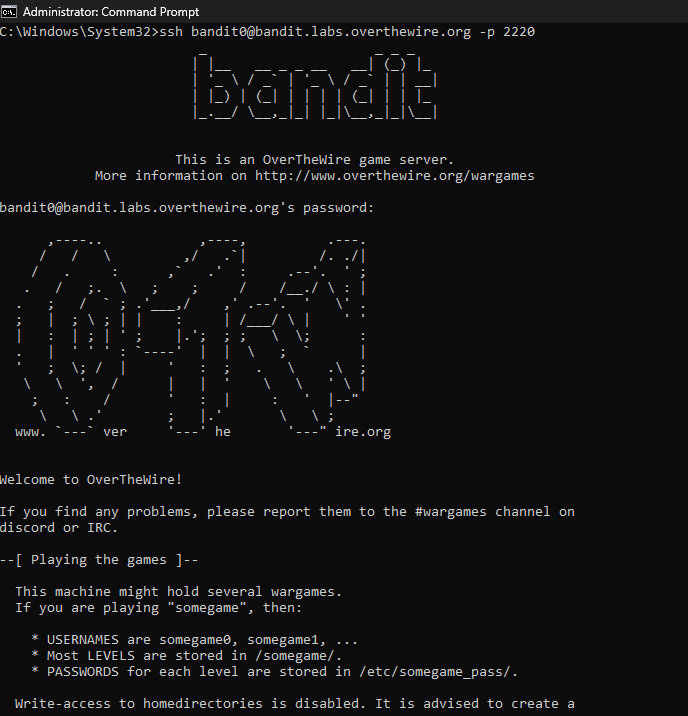
3. Used `cat readme` to read the file contents

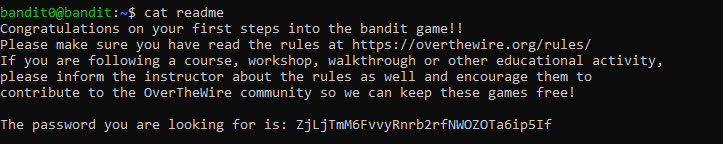
Credentials:

- Username: bandit1

- Password: ZjLjTmM6FvvyRnrb2rfNWOZOTa6ip5If

Learning: Basic SSH connection and file reading with cat command.





***Level 1 → Level 2***

Tools Used: cat

Objective: Read a file with an unusual name (hyphen `-`).

Commands Used: `cat ./-`

Steps:

1. Found file named `-` using `ls`

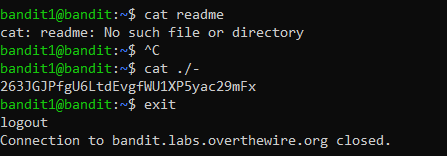
2. Used `cat ./-` to read the file (avoiding interpretation as command flag)

Credentials:

- Username: bandit2

- Password: 263JGJPfgU6LtdEvgfWU1XP5yac29mFx

Learning: Handle special character filenames using path prefixes to prevent flag interpretation.



***Level 2 → Level 3***

Tools Used: cat

Objective: Read a file with spaces in its name.

Commands Used: `cat -- "--spaces in this filename--"`

Steps:

1. Found file with spaces in name

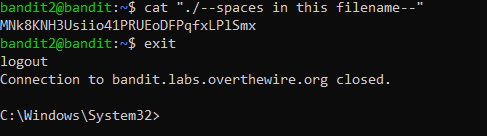
2. Used quotes to handle spaces in filename

Credentials:

- Username: bandit3

- Password: MNk8KNH3Usiio41PRUEoDFPqfxLPlSmx

Learning: Handle filenames with spaces using quotes or escape characters.



***Level 3 → Level 4***

Tools Used: ls, cd, cat

Objective: Find password in a hidden file within the `inhere` directory.

Commands Used: `ls -a`, `cat .hidden`

Steps:

1. Navigated to `inhere` directory with `cd inhere`

2. Used `ls -a` to show hidden files

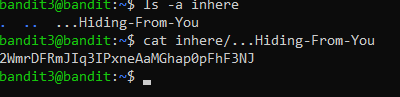
3. Found `.hidden` file and read it with `cat .hidden`

Credentials:

- Username: bandit4

- Password: 2WmrDFRmJIq3IPxneAaMGhap0pFhF3NJ

Learning: Use `ls -a` to display hidden files starting with dots.



***Level 4 → Level 5***

Tools Used: find, file, cat

Objective: Find human-readable file among many files in `inhere` directory.

Commands Used: `find . -type f -exec file {} \;`, `cat ./-file07`

Steps:

1. Used `find` with `file` command to identify file types

2. Located the ASCII text file among binary files

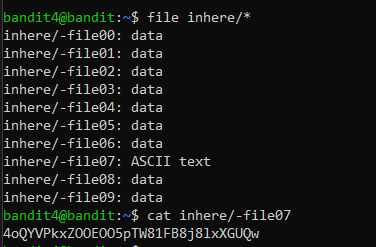
3. Read the human-readable file

Credentials:

- Username: bandit5

- Password: 4oQYVPkxZOOEOO5pTW81FB8j8lxXGUQw

Learning: Use `file` command to identify file types and find human-readable files.



***Level 5 → Level 6***

Tools Used: find

Objective: Find file with specific properties: human-readable, 1033 bytes, not executable.

Commands Used: `find . -type f -size 1033c ! -executable`

Steps:

1. Used `find` with multiple conditions to locate specific file

2. Found file in `./maybehere07/.file2`

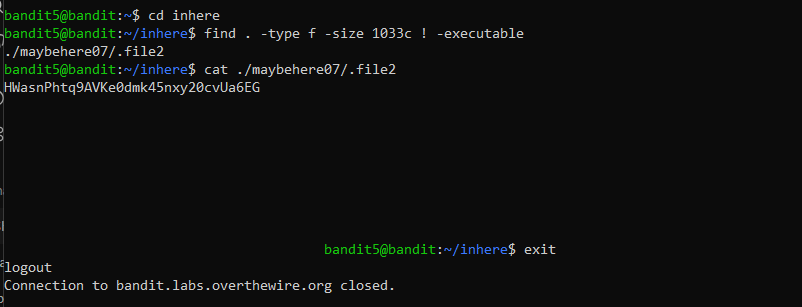
3. Read the file contents

Credentials:

- Username: bandit6

- Password: HWasnPhtq9AVKe0dmk45nxy20cvUa6EG

Learning: Combine multiple find conditions using size, type, and permission flags.



***Level 6 → Level 7***

Tools Used: find

Objective: Find file owned by bandit7:bandit6 with size 33 bytes.

Commands Used: `find / -user bandit7 -group bandit6 -size 33c 2>/dev/null`

Steps:

1. Searched entire filesystem from root with ownership and size criteria

2. Suppressed permission errors with `2>/dev/null`

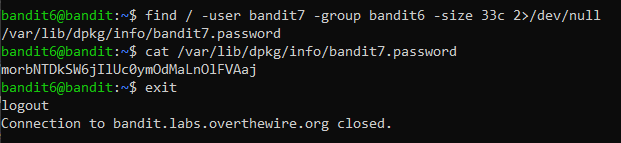
3. Found file at `/var/lib/dpkg/info/bandit7.password`

Credentials:

- Username: bandit7

- Password: morbNTDkSW6jIlUc0ymOdMaLnOlFVAaj

Learning: Search files by ownership and redirect errors to suppress permission denials.



***Level 7 → Level 8***

Tools Used: grep

Objective: Find password next to word "millionth" in data.txt.

Commands Used: `grep millionth data.txt`

Steps:

1. Used `grep` to search for specific keyword in large file

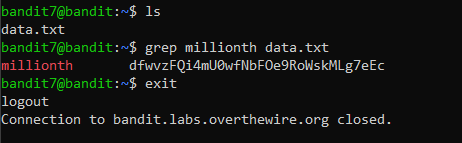
2. Found password on the same line as "millionth"

Credentials:

- Username: bandit8

- Password: dfwvzFQi4mU0wfNbFOe9RoWskMLg7eEc

Learning: Use grep to quickly search for patterns in large text files.



***Level 8 → Level 9***

Tools Used: sort, uniq

Objective: Find the only unique line in data.txt.

Commands Used: `sort data.txt | uniq -u`

Steps:

1. Sorted the file contents alphabetically

2. Used `uniq -u` to find lines that appear only once

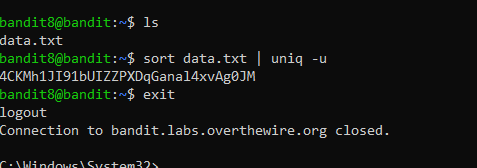
3. Found the unique password line

Credentials:

- Username: bandit9

- Password: 4CKMh1JI91bUIZZPXDqGanal4xvAg0JM

Learning: Combine sort and uniq commands to find unique entries in text files.



***Level 9 → Level 10***

Tools Used: strings

Objective: Find human-readable strings in binary data file.

Commands Used: `strings data.txt`

Steps:

1. Used `strings` to extract printable characters from binary file

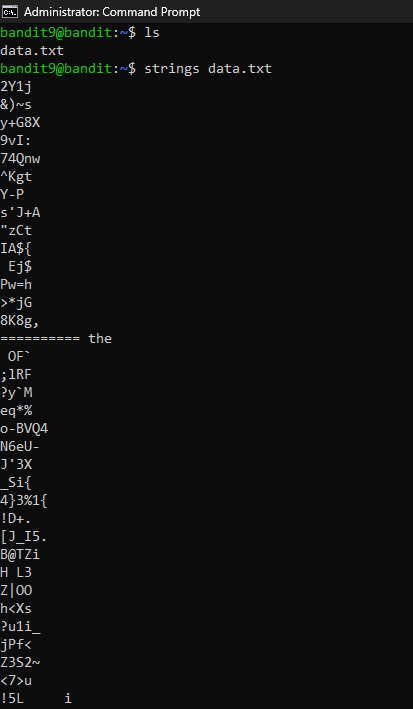
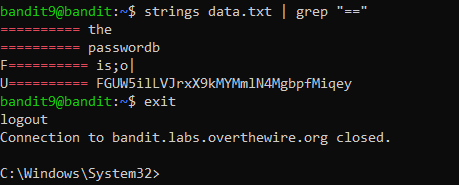
2. Located password among the readable strings

Credentials:

- Username: bandit10

- Password: FGUW5ilLVJrxX9kMYMmlN4MgbpfMiqey

Learning: Extract readable text from binary files using the strings command.



***Level 10 → Level 11***

Tools Used: base64

Objective: Decode Base64 encoded data.

Commands Used: `base64 -d data.txt`

Steps:

1. Identified Base64 encoded content

2. Used `base64 -d` to decode the content

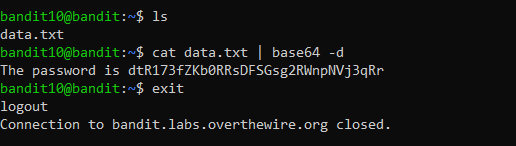
3. Retrieved the decoded password

Credentials:

- Username: bandit11

- Password: dtR173fZKb0RRsDFSGsg2RWnpNVj3qRr

Learning: Recognize and decode Base64 encoded data using base64 command.



***Level 11 → Level 12***

Tools Used: tr

Objective: Decode ROT13 cipher.

Commands Used: `cat data.txt | tr 'A-Za-z' 'N-ZA-Mn-za-m'`

Steps:

1. Identified ROT13 encoded text

2. Used `tr` to translate characters with ROT13 mapping

3. Decoded the password

Credentials:

- Username: bandit12

- Password: 7x16WNeHIi5YkIhWsfFIqoognUTyj9Q4

Learning: Use tr command for character translation and ROT13 decoding.

***Level 12 → Level 13***

Tools Used: xxd, gzip, bzip2, tar

Objective: Extract password from repeatedly compressed hex dump.

Commands Used:

- `xxd -r data.txt > data.bin`

- Multiple instances of:

1. mv <filename> <new filename>

2. gzip -d <new filename in .gz >

3. bzip2 -d <new filename in .bz2 >

4. tar -xf data.bin

Steps:

1. Reversed hex dump using `xxd -r`

2. Performed 9 extraction operations alternating between gzip, bzip2, and tar

3. Used `file` command to identify compression type at each step

Credentials:

- Username: bandit13

- Password: FO5dwFsc0cbaIiH0h8J2eUks2vdTDwAn

Learning: Handle multiple compression layers and use file command to identify formats.

***Level 13 → Level 14***

Tools Used: ssh (private key authentication)

Objective: Use SSH private key to access next level.

Commands Used: `ssh -i sshkey.private bandit14@localhost -p 2220`

ed /etc/bandit\_pass

Steps:

1. Found SSH private key file

2. Used key-based authentication instead of password

3. Accessed `/etc/bandit\_pass/bandit14` for password

Credentials:

- Username: bandit14

- Password: MU4VWeTyJk8ROof1qqmcBPaLh7lDCPvS

Learning: Use SSH private keys for authentication and understand key-based access.

***Level 14 → Level 15***

Tools Used: netcat (nc)

Objective: Connect to localhost port 30000 to retrieve password.

Commands Used: `nc localhost 30000`

Steps:

1. Used netcat to establish TCP connection to port 30000

2. Server immediately returned the password

Credentials:

- Username: bandit15

- Password: 8xCjnmgoKbGLhHFAZlGE5Tmu4M2tKJQo

Learning: Use netcat for basic network connections and port communication.

***Level 15 → Level 16***

Tools Used: openssl

Objective: Connect to port 30001 using SSL encryption.

Commands Used: `openssl s\_client -connect localhost:30001 -quiet`

Steps:

1. Established SSL connection using openssl s\_client

2. Server returned password over secure connection

Credentials:

- Username: bandit16

- Password: kSkvUpMQ7lBYyCM4GBPvCvT1BfWRy0Dx

Learning: Use openssl for secure SSL/TLS connections.

***Level 16 → Level 17***

Tools Used: nmap, openssl

Objective: Find correct SSL port among port range 31000-32000.

Commands Used:

- `nmap -p 31000-32000 localhost`

- `openssl s\_client -connect localhost:31790 -quiet`

- nano sshkey.private //paste the key here

- chmod 600 sshkey.private

- ssl -i sshkey.private bandit17@localhost -p 2220

Steps:

1. Used nmap to scan port range and identify open ports

2. Connected to SSL-enabled port 31790

3. Received SSH private key for next level

4. Set up key file with proper permissions (`chmod 600`)

Credentials:

- Username: bandit17

- SSH Key:

-----BEGIN RSA PRIVATE KEY-----

MIIEogIBAAKCAQEAvmOkuifmMg6HL2YPIOjon6iWfbp7c3jx34YkYWqUH57SUdyJ

imZzeyGC0gtZPGujUSxiJSWI/oTqexh+cAMTSMlOJf7+BrJObArnxd9Y7YT2bRPQ

Ja6Lzb558YW3FZl87ORiO+rW4LCDCNd2lUvLE/GL2GWyuKN0K5iCd5TbtJzEkQTu

DSt2mcNn4rhAL+JFr56o4T6z8WWAW18BR6yGrMq7Q/kALHYW3OekePQAzL0VUYbW

JGTi65CxbCnzc/w4+mqQyvmzpWtMAzJTzAzQxNbkR2MBGySxDLrjg0LWN6sK7wNX

x0YVztz/zbIkPjfkU1jHS+9EbVNj+D1XFOJuaQIDAQABAoIBABagpxpM1aoLWfvD

KHcj10nqcoBc4oE11aFYQwik7xfW+24pRNuDE6SFthOar69jp5RlLwD1NhPx3iBl

J9nOM8OJ0VToum43UOS8YxF8WwhXriYGnc1sskbwpXOUDc9uX4+UESzH22P29ovd

d8WErY0gPxun8pbJLmxkAtWNhpMvfe0050vk9TL5wqbu9AlbssgTcCXkMQnPw9nC

YNN6DDP2lbcBrvgT9YCNL6C+ZKufD52yOQ9qOkwFTEQpjtF4uNtJom+asvlpmS8A

vLY9r60wYSvmZhNqBUrj7lyCtXMIu1kkd4w7F77k+DjHoAXyxcUp1DGL51sOmama

+TOWWgECgYEA8JtPxP0GRJ+IQkX262jM3dEIkza8ky5moIwUqYdsx0NxHgRRhORT

8c8hAuRBb2G82so8vUHk/fur85OEfc9TncnCY2crpoqsghifKLxrLgtT+qDpfZnx

SatLdt8GfQ85yA7hnWWJ2MxF3NaeSDm75Lsm+tBbAiyc9P2jGRNtMSkCgYEAypHd

HCctNi/FwjulhttFx/rHYKhLidZDFYeiE/v45bN4yFm8x7R/b0iE7KaszX+Exdvt

SghaTdcG0Knyw1bpJVyusavPzpaJMjdJ6tcFhVAbAjm7enCIvGCSx+X3l5SiWg0A

R57hJglezIiVjv3aGwHwvlZvtszK6zV6oXFAu0ECgYAbjo46T4hyP5tJi93V5HDi

Ttiek7xRVxUl+iU7rWkGAXFpMLFteQEsRr7PJ/lemmEY5eTDAFMLy9FL2m9oQWCg

R8VdwSk8r9FGLS+9aKcV5PI/WEKlwgXinB3OhYimtiG2Cg5JCqIZFHxD6MjEGOiu

L8ktHMPvodBwNsSBULpG0QKBgBAplTfC1HOnWiMGOU3KPwYWt0O6CdTkmJOmL8Ni

blh9elyZ9FsGxsgtRBXRsqXuz7wtsQAgLHxbdLq/ZJQ7YfzOKU4ZxEnabvXnvWkU

YOdjHdSOoKvDQNWu6ucyLRAWFuISeXw9a/9p7ftpxm0TSgyvmfLF2MIAEwyzRqaM

77pBAoGAMmjmIJdjp+Ez8duyn3ieo36yrttF5NSsJLAbxFpdlc1gvtGCWW+9Cq0b

dxviW8+TFVEBl1O4f7HVm6EpTscdDxU+bCXWkfjuRb7Dy9GOtt9JPsX8MBTakzh3

vBgsyi/sN3RqRBcGU40fOoZyfAMT8s1m/uYv52O6IgeuZ/ujbjY=

-----END RSA PRIVATE KEY-----

Learning: Use nmap for port scanning and handle SSH key setup with proper permissions.

***Level 17 → Level 18***

Tools Used: diff

Objective: Find password by comparing two password files.

Commands Used: `diff passwords.old passwords.new`

Steps:

1. Compared two password files using diff

2. Identified the changed line showing new password

Credentials:

- Username: bandit18

- Password: x2gLTTjFwMOhQ8oWNbMN362QKxfRqGlO

Learning: Use diff command to compare files and identify changes.

***Level 18 → Level 19***

Tools Used: ssh (command execution)

Objective: Execute command via SSH when login shell is restricted.

Commands Used: `ssh bandit18@bandit.labs.overthewire.org -p 2220 cat readme`

Steps:

1. Bypassed restricted shell by executing command directly via SSH

2. Read readme file without interactive login

Credentials:

- Username: bandit19

- Password: cGWpMaKXVwDUNgPAVJbWYuGHVn9zl3j8

Learning: Execute commands through SSH without interactive shell access.

***Level 19 → Level 20***

Tools Used: setuid binary

Objective: Use setuid binary to read password file as different user.

Commands Used: `./bandit20-do cat /etc/bandit\_pass/bandit20`

Steps:

1. Found setuid binary that executes commands as bandit20

2. Used it to read bandit20's password file

Credentials:

- Username: bandit20

- Password: 0qXahG8ZjOVMN9Ghs7iOWsCfZyXOUbYO

Learning: Understand setuid binaries and privilege escalation concepts.

***Level 20 → Level 21***

Tools Used: netcat, setuid binary

Objective: Use network client that validates password before giving next one.

Commands Used:

- `nc -l -p 12345 < /etc/bandit\_pass/bandit20 &`

- `./suconnect 12345` (2 times)

Steps:

1. Started netcat listener serving current password

2. Used suconnect binary to connect and validate

3. Received next level password

Credentials:

- Username: bandit21

- Password: EeoULMCra2q0dSkYj561DX7s1CpBuOBt

Learning: Coordinate network services and understand client-server validation.

***Level 21 → Level 22***

Tools Used: cron analysis

Objective: Find password by examining cron job output.

Commands Used:

- `ls -l /etc/cron.d/`

- `cat /etc/cron.d/cronjob\_bandit22`

- `cat /tmp/<tempfilename>`

Steps:

1. Examined cron job configuration

2. Found script that writes password to temporary file

3. Read the password from temp file

Credentials:

- Username: bandit22

- Password: tRae0UfB9v0UzbCdn9cY0gQnds9GF58Q

Learning: Understand cron jobs and how scheduled tasks can expose information.

***Level 22 → Level 23***

Tools Used: cron analysis, md5sum

Objective: Predict filename generated by script using username.

Commands Used:

- ` ls -l /etc/cron.d/`

- ` cat /etc/cron.d/cronjob\_bandit23`

- `cat /usr/bin/cronjob\_bandit23.sh`

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

- ` cat /tmp/<generated md5hash>`

Steps:

1. Analysed cron script that generates filename from username hash

2. Computed MD5 hash for "I am user bandit23"

3. Read password from predicted filename

Credentials:

- Username: bandit23

- Password: 0Zf11ioIjMVN551jX3CmStKLYqjk54Ga

Learning: Reverse engineer script logic to predict filenames and understand hashing.

***Level 23 → Level 24***

Tools Used: script creation, cron exploitation

Objective: Create script that will be executed by cron job.

Commands Used:

- ls -l /etc/cron.d/

- cat /etc/cron.d/cronjob\_bandit24

- cat /usr/bin/cronjob\_bandit24.sh

- echo '#!/bin/bash' > getpass.sh

- echo 'cat /etc/bandit\_pass/bandit24 > /tmp/bandit24\_pass' >> getpass.sh

- chmod +x getpass.sh

- cp getpass.sh /var/spool/bandit24/foo/

- cat /tmp/bandit24\_pass

Steps:

1. Created script to dump password to accessible location

2. Placed script in directory monitored by cron job

3. Waited for execution and read password

Credentials:

- Username: bandit24

- Password: gb8KRRCsshuZXI0tUuR6ypOFjiZbf3G8

Learning: Exploit cron job execution to run custom scripts and extract information.

***Level 24 → Level 25***

Tools Used: brute force scripting, netcat

Objective: Find 4-digit pincode by brute force attack.

Commands Used:

- password=$(cat /etc/bandit\_pass/bandit24)

- for i in $(seq -w 0 9999); do echo "$password $i" | nc -q1 localhost 30002; done

result obtained:

- 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.

- Correct!

- The password of user bandit25 is iCi86ttT4KSNe1armKiwbQNmB3YJP3q4

Steps:

1. Created loop to try all 4-digit combinations (0000-9999)

2. Used netcat to submit password and pincode combinations

3. Found correct combination and received password

Credentials:

- Username: bandit25

- Password: iCi86ttT4KSNe1armKiwbQNmB3YJP3q4

Learning: Implement brute force attacks and understand the importance of rate limiting.

***Level 25 → Level 26***

Tools Used: vi/vim, shell escape

Objective: Escape restricted shell through text editor.

Commands Used:

- ls -la

- cat /etc/passwd | grep bandit26

- ssh bandit26@localhost -p 2220 -i bandit26.sshkey

- :set shell?

- :set shell=/bin/bash

- :set shell?

- :ls

- cat /etc/bandit\_pass/bandit26

Steps:

1. Connected but was dropped into restricted `more` command

2. Pressed `v` in more to enter vi editor

3. Changed shell setting and escaped to bash

4. Read password file

Credentials:

- Username: bandit26

- Password: s0773xxkk0MXfdqOfPRVr9L3jJBUOgCZ

Learning: Escape restricted environments using text editor capabilities.

***Level 26 → Level 27***

Tools Used: setuid binary

Objective: Use setuid binary to access bandit27 password.

Commands Used: `./bandit27-do cat /etc/bandit\_pass/bandit27`

Steps:

1. Found setuid binary for bandit27

2. Used it to read protected password file

Credentials:

- Username: bandit27

- Password: upsNCc7vzaRDx6oZC6GiR6ERwe1MowGB

Learning: Apply setuid binary knowledge from previous levels.

***Level 27 → Level 28***

Tools Used: git

Objective: Clone git repository to find password.

Commands Used: `mktemp -d`, `git clone ssh://bandit27-git@localhost:2220/home/bandit27-git/repo`

Steps:

1. Cloned git repository using SSH

2. Found README file containing password

Credentials:

- Username: bandit28

- Password: Yz9IpL0sBcCeuG7m9uQFt8ZNpS4HZRcN

Learning: Basic git repository cloning and file examination.

***Level 28 → Level 29***

Tools Used: git log, git show

Objective: Find password in git commit history.

Commands Used:

- `mktemp -d`

- `git clone ssh://bandit28-git@localhost:2220/home/bandit28-git/repo`

- `git log`

- `git show f257900db7c134cb5224c91013817e76d18457e0`

Steps:

1. Examined git commit history

2. Found commit that removed password from README

3. Viewed commit diff to see original password

Credentials:

- Username: bandit29

- Password: 4pT1t5DENaYuqnqvadYs1oE4QLCdjmJ7

Learning: Investigate git history to find accidentally committed sensitive information.

***Level 29 → Level 30***

Tools Used: git branch, git checkout

Objective: Find password in different git branch.

Commands Used:

mktemp -d

- git clone ssh://bandit29-git@localhost:2220/home/bandit29-git/repo

- cd repo

- ls

- cat README.md

- git log

- git checkout 57d7

- git branch

- git checkout master

- git status

- git branch -a

- git checkout dev

- git status

- ls

- cat README.md

Steps:

1. Listed all branches including remote ones

2. Switched to dev branch

3. Found password in README file on dev branch

Credentials:

- Username: bandit30

- Password: qp30ex3VLz5MDG1n91YowTv4Q8l7CDZL

Learning: Explore git branches to find information not present in main branch.

***Level 30 → Level 31***

Tools Used: git tag, git show

Objective: Find password in git tag.

Commands Used:

- mktemp -d

- git clone ssh://bandit30-git@localhost:2220/home/bandit30-git/repo

- cd repo

- ls

- cat README.md

- git log

- git branch

- git branch -a

- git tag

- git show secret

Steps:

1. Listed available git tags

2. Examined "secret" tag contents

3. Found password in tag annotation

Credentials:

- Username: bandit31

- Password: fb5S2xb7bRyFmAvQYQGEqsbhVyJqhnDy

Learning: Use git tags to mark and store additional information in repositories.

***Level 31 → Level 32***

Tools Used: git add, git commit, git push

Objective: Push specific file to remote repository despite gitignore.

Commands Used:

- mktemp -d

- git clone ssh://bandit30-git@localhost:2220/home/bandit30-git/repo

- cd repo

- cat README.md

- nano key.txt

- git add key.txt

- rm .gitignore

- git add key.txt

- git commit -m "text"

- git push

Steps:

1. Created required key.txt file with specified content

2. Removed .gitignore that was blocking the file

3. Added, committed, and pushed the file

4. Received password from remote repository

Credentials:

- Username: bandit32

- Password: 3O9RfhqyAlVBEZpVb6LYStshZoqoSx5K

Learning: Understand git workflow and how gitignore affects file tracking.

***Level 32 → Level 33***

Tools Used: shell variable exploitation

Objective: Break out of uppercase shell restriction.

Commands Used:

- `$0` (to get normal shell)

- `whoami

- `cat /etc/bandit\_pass/bandit33`

Steps:

1. Found shell that converts all input to uppercase

2. Used `$0` variable to spawn new shell

3. Read final password

Credentials:

- Username: bandit33

- Password: tQdtbs5D5i2vJwDUNgPAVJbWYuGHVn9zl3j8

Learning: Exploit shell variables to bypass input restrictions and escape constrained environments.

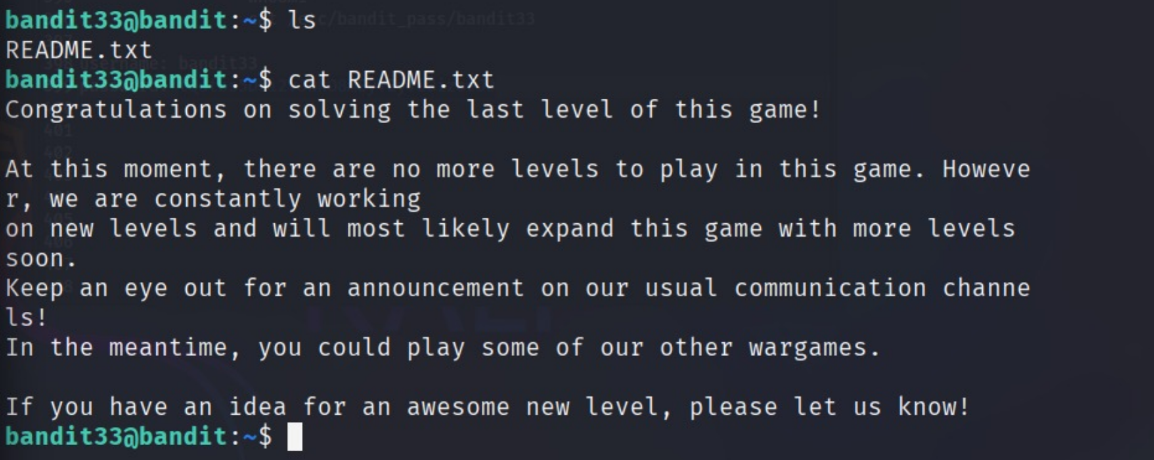
***Level 33 → Level 34***

Objective: Obtain the final message.

Commands Used:

- “ls” to list the files.

- “cat README.md” to showcase the final message.

Message: