Proof Of Concept: OverTheWire — Bandit

Intern ID : 195
Intern Name : Prathamesh Arun Kamble
Target: bandit.labs.overthewire.org (SSH) — default port 2220

Connection / Starter credentials
Host: bandit.labs.overthewire.org
Port: 2220
Username: bandit0
Password: bandit0
SSH command:
ssh bandit0@bandit.labs.overthewire.org -p 2220

Level 0 → 1

Tools Used: ssh, ls, cat
Objective: Connect to Bandit server and read the password for next level.
Commands Used:
ssh bandit0@bandit.labs.overthewire.org -p 2220
ls
cat readme
Steps:
1. SSH into the server using username bandit0, password bandit0.
2. Use ls to list files and find readme.
3. Use cat readme to read password for bandit1.
Learning: Basic SSH login and reading files.

Level 1 → 2

Tools Used: cat, ./ path prefix
Objective: Read a file with unusual name (-).
Commands Used:
ls
cat ./-
Steps:
1. Use ls to find file named -.
2. Use cat ./- to avoid interpretation as an option.
Learning: Handle filenames starting with -.

Level 2 → 3

Tools Used: ls, cat
Objective: Read a file with spaces in its name.
Commands Used:
ls
cat "spaces in this filename"
Steps:
1. Use ls to find file with spaces.
2. Quote or escape the filename to read with cat.
Learning: Handle filenames with spaces.

Level 3 → 4

Tools Used: ls, cat
Objective: Read hidden file in a directory.

Commands Used:
ls -a inhere
cat inhere/.hidden
Steps:
1. Change into inhere directory.
2. Use ls -a to reveal hidden files.
3. Found .hidden and read it.
Learning: Use ls -a to reveal hidden files.

Level 4 → 5

Tools Used: file, cat
Objective: Find human-readable file among binary files.
Commands Used:
cd inhere
file ./-file*
cat ./-file07
Steps:
1. Change into inhere.
2. Use file to check each file.
3. Find the one that is ASCII text and read it.
Learning: Identify text files among binaries.

Level 5 → 6

Tools Used: find, cat
Objective: Find file with specific properties.
Commands Used:

find . -type f -size 1033c ! -executable
cat ./maybehere07/.file2
Steps:
1. Use find with conditions: size 1033 bytes, not executable.
2. Locate file and read with cat.
Learning: Use find with size and permission filters.

Level 6 → 7

Tools Used: find, cat
Objective: Locate a file owned by user bandit7 and group bandit6.
Commands Used:
find / -user bandit7 -group bandit6 -size 33c 2>/dev/null
cat /var/lib/dpkg/info/bandit7.password
Steps:
1. Use find with owner, group, size filters.
2. Redirect errors to /dev/null.
3. Read found file for password.
Learning: Advanced use of find with ownership filters.

Level 7 → 8

Tools Used: grep, cat
Objective: Find password stored next to keyword millionth.
Commands Used:

grep millionth data.txt
Steps:
1. Use grep to search for the keyword.
2. The line contains the password.
Learning: Search for keywords with grep.

Level 8 → 9

Tools Used: sort, uniq
Objective: Find unique line in a file.
Commands Used:
sort data.txt | uniq -u
Steps:
1. Sort the file to group duplicates.
2. Use uniq -u to print the unique line.
Learning: Use sort + uniq to detect unique entries.

Level 9 → 10

Tools Used: strings, grep
Objective: Find human-readable password in binary file.
Commands Used:
strings data.txt | grep ==
Steps:
1. Use strings to extract readable text.
2. Filter with grep for ==.
Learning: Extract printable strings from binaries.

Level 10 → 11

Tools Used: base64
Objective: Decode base64 encoded text.
Commands Used:
base64 -d data.txt
Steps:
1. Detect file is base64.
2. Decode with base64 -d.
Learning: Work with base64 encoded files.

Level 11 → 12

Tools Used: tr
Objective: Decode ROT13 encoded text.
Commands Used:
cat data.txt | tr 'A-Za-z' 'N-ZA-Mn-za-m'
Steps:
1. Apply ROT13 translation.
2. Output is the password.
Learning: Use tr for character substitution.

Level 12 → 13

Tools Used: file, xxd, tar, gzip, bzip2
Objective: Extract multiple layers of encoded/compressed files.
Commands Used:

xxd -r data.txt > data
file data
mv data data.gz
gzip -d data.gz
Steps:
1. Reverse hex with xxd -r.
2. Repeatedly check file type and decompress (gzip, bzip2, tar).
3. Eventually extract password file.
Learning: Iterative decoding and decompression.

Level 13 → 14

Tools Used: ssh key
Objective: Use private SSH key to login as bandit14.
Commands Used:
ssh -i sshkey.private bandit14@localhost -p 2220
Steps:
1. Use provided key.
2. SSH into localhost as bandit14.
Learning: Authentication with SSH private keys.

Level 14 → Level 15

Tools Used: nc
Objective: Submit password to localhost port 30000.
Commands Used:
cat /etc/bandit_pass/bandit14 | nc localhost 30000
Steps:

1. Retrieve password.
2. Pipe into netcat.
Learning: Simple TCP client-server interaction.

Level 15 → 16

Tools Used: openssl
Objective: Connect over SSL to localhost port 30001.
Commands Used:
openssl s_client -connect localhost:30001
Steps:
1. Connect with openssl.
2. Provide password, receive next password.
Learning: Use openssl s_client for SSL.

Level 16 → Level 17

Tools Used: nmap, ssh
Objective: Scan for open ports and connect with SSH key.
Commands Used:
nmap -p31000-32000 localhost
openssl s_client -connect localhost:31790
Steps:
1. Scan range with nmap.
2. Identify open SSL service.
3. Connect, get SSH key for bandit17.
Learning: Port scanning, SSL connection, key extraction.

Level 17 → 18

Tools Used: diff
Objective: Find difference between two files**.**
Commands Used:
diff passwords.new passwords.old
Steps:
1. Compare both files.
2. Find differing line with password.
Learning: Use diff to compare files.

Level 18 → 19

Tools Used: ssh
Objective: Escape forced command shell.
Commands Used:
ssh bandit18@localhost -p 2220 cat readme
Steps:
1. Pass command directly to ssh.
2. Get password.
Learning: Pass commands via ssh directly.

Level 19 → 20

Tools Used: setuid binary
Objective: Execute binary with setuid to read password.
Commands Used:

./bandit20-do cat /etc/bandit_pass/bandit20
Steps:
1. Run helper binary with command.
2. Get password.
Learning: Use setuid programs to access files.

Level 20 → 21

Tools Used: nc
Objective: Use networking to send data to server.
Commands Used:
echo "password" | nc -l -p 1234 &
./suconnect 1234
Steps:
1. Start listener with password.
2. Run suconnect to fetch.
Learning: Netcat for local communication.

Level 21 → 22

Tools Used: cron, cat
Objective: Read password via cron jobs.
Commands Used:
cat /etc/cron.d/*
cat /usr/bin/cronjob_bandit22.sh
cat /tmp/bandit22/password

Steps:
1. Inspect cron jobs.

2. Find script storing password.
3. Read temporary file.
Learning: Analyze cron jobs.

Level 22 → 23

Tools Used: cron, cat
Objective: Read cron job output for bandit23**.**
Commands Used:
cat /etc/cron.d/*
cat /usr/bin/cronjob_bandit23.sh
cat /tmp/*bandit23/*
Steps:
1. Inspect cron scripts.
2. Password is written to a file in /tmp.
Learning: Use cron analysis again.

Level 23 → Level 24

Tools Used: cron, cat, md5sum
Objective: Generate hashed filename to retrieve password.
Commands Used:
echo I_am_user_bandit23 | md5sum
cat /tmp/<hash>
Steps:
1. Hash username.
2. Use hash as filename.
3. Retrieve password.

Learning: Hash-based file naming**.**

Level 24 → Level 25

Tools Used: bash script, nc
Objective: Brute-force PIN code.
Commands Used:
```
for i in {0000..9999}; do
  echo "$pass $i" | nc localhost 30002
done
```
Steps:
1. Automate brute force.
2. Correct PIN gives password**.**
Learning: Automate brute force with scripts.

Level 25 → 26

Tools Used: ssh, screen
Objective: Exploit running screen session**.**
Commands Used:
```
ps aux
cat /etc/bandit_pass/bandit26
```
Steps:
1. Inspect screen process.
2. Recover password**.**
Learning: Analyze screen sessions.

Level 26 → 27

Tools Used: ssh, setuid
Objective: Use setuid shell**.**
Commands Used:
./bandit27-do cat /etc/bandit_pass/bandit27
Steps:
1. Use helper binary to read password.
Learning: Reuse setuid binary.

Level 27 → 28

Tools Used: git
Objective: Clone repo and read password.
Commands Used:
git clone
ssh://bandit27-git@localhost:2220/home/bandit27-git/repo
cat repo/README
Steps:
1. Clone git repo.
2. Inspect README.
Learning: Use git to retrieve files.

Level 28 → 29

Tools Used: git log
Objective: Inspect commit history**.**
Commands Used:
git log
git show <commit>

Steps:
1. Look at past commits.
2. Find password.
Learning: Inspect git history**.**

Level 29 → 30

Tools Used: git tag
Objective: Inspect git tags**.**
Commands Used:
git tag
git show secret
Steps:
1. List tags.
2. Inspect tag content.
Learning: Use git tags.

Level 30 → 31

Tools Used: git branch
Objective: Inspect git branches.
Commands Used:
git branch -a
git checkout dev
cat README.md
Steps:
1. Switch branch.
2. Read password.

Learning: Use git branches.

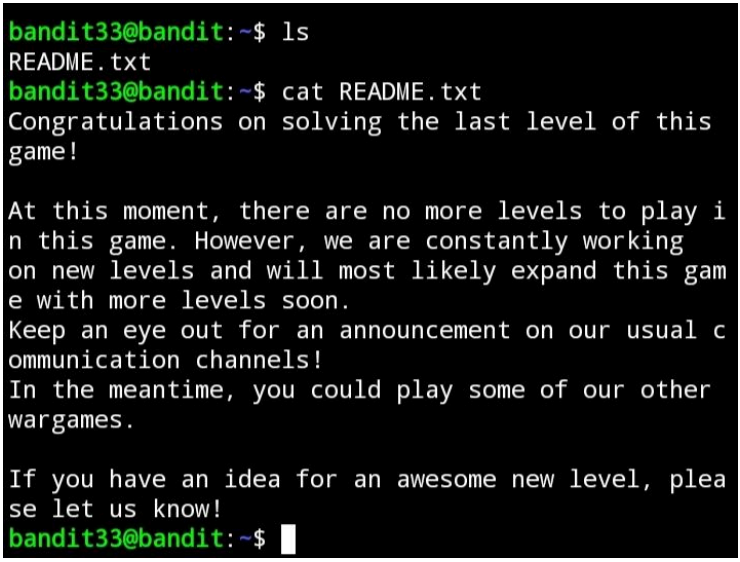
Level 31 → Level 32

Tools Used: git
Objective: Use git to inspect .gitignore.
Commands Used:
cat .gitignore
cat .gitignore_hidden_file
Steps:
1. Inspect ignored files.
2. Read hidden password file.
Learning: Use .gitignore to find ignored files.

Level 32 → 33

Tools Used: bash tricks
Objective: Escape shell environment**.**
Commands Used:
$0
cat /etc/bandit_pass/bandit33
Steps:
1. Invoke shell using $0.
2. Read password.
Learning: Escape restricted shell**.**

Level 33 → ✅


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 i
n this game. However, we are constantly working
on new levels and will most likely expand this gam
e with more levels soon.
Keep an eye out for an announcement on our usual c
ommunication channels!
In the meantime, you could play some of our other
wargames.
If you have an idea for an awesome new level, plea
se let us know!
bandit33@bandit:~$

_____ THANK YOU _____