OverTheWire Bandit Wargame

Name : Prateek
Intern ID : 166

The **Bandit wargame** is a series of beginner-friendly Linux challenges hosted on the **OverTheWire** platform. Its primary goal is to teach players the basics of using the **Linux command-line environment**, navigating the filesystem, handling files, and performing simple security-related tasks. . Each level introduces Linux commands and concepts that help build foundational skills for security, system administration, and Capture the Flag (CTF) challenges.

Core Commands Used

Throughout these levels, the following Linux commands and tools were essential:

- **ssh** → Connect securely to a remote machine (used to log into each level).
- **ls** → List files and directories in the current location.
- **cd** → Change directory.
- **pwd** → Print the current working directory.
- **cat** → Display the contents of a file.
- **file** → Identify the file type.
- **du** → Show file/directory disk usage (often to locate files by size).
- **find** → Search for files by name, size, permissions, etc.
- **grep** → Search for text patterns inside files.
- **sort** → Sort lines of text.
- **uniq** → Filter out or detect duplicate lines.
- **strings** → Extract human-readable strings from binary files.
- **base64** → Encode/decode base64 data.
- **tr** → Translate or replace characters (e.g., ROT13).
- **tar** → Extract or compress `.tar` archive files.
- **gzip/gunzip** → Compress/uncompress `.gz` files.
- **bzip2/bunzip2** → Compress/uncompress `.bz2` files.
- **xxd** → Create or reverse hexdumps.
- **cp** → Copy files.
- **mv** → Move/rename files.

- **mkdir** → Create a new directory.
- **mktemp -d** → Create a temporary directory with a random name.
- **less/more** → View file contents page by page.
- **head/tail** → View first or last lines of a file.
- **nc (netcat)** → Read/write data to network connections (send password to ports).
- **telnet** → Simple TCP connection tool (like nc, older).
- **ncat** → Enhanced netcat with SSL support.
- **socat** → Advanced tool to connect or forward between sockets.
- **openssl s_client** → Connect to SSL/TLS services and interact with them.
- **nmap** → Network mapper, scans for open ports and services.
- **netstat** → Show active network connections/ports (older tool).
- **ss** → Modern replacement for netstat to view sockets.
- **whoami** → Show the current logged-in username.
- **id** → Show user ID and group ID.
- **chmod** → Change file permissions.
- **touch** → Create an empty file or update timestamps.
- **echo** → Print text or pass strings into commands.
- **man** → Display manual pages (help) for commands.

Level Summaries (0 → 20)

Level 0 → 1
Login via SSH using given credentials. Learn to connect to a remote system.

Level 1 → 2
Find password stored in a file named `-` which requires './' to access.

Level 2 → 3
Password hidden in a file with spaces in its name; escape spaces with quotes or backslashes.

Level 3 → 4
Look inside a hidden file in a directory.

Level 4 → 5
Identify the human-readable file among many using 'file' command.

Level 5 → 6
Find file based on conditions: human-readable, specific size, non-executable using 'find'.

Level 6 → 7
Search for password in a file owned by user bandit7 and group bandit6.

Level 7 → 8
Password hidden inside a text file; use 'grep' to locate it.

Level 8 → 9
Sort and find unique password string using 'sort' and 'uniq'.

Level 9 → 10
Extract printable characters from binary file with 'strings'.

Level 10 → 11
Password is encoded in base64; decode using 'base64 -d'.

Level 11 → 12
ROT13 encoding used; decode using 'tr'.

Level 12 → 13
Password hidden in a repeatedly compressed file; use 'xxd', 'tar', 'gzip', 'bzip2' iteratively.

Level 13 → 14
Use provided SSH private key to login as bandit14.

Level 14 → 15
Send current password to correct localhost port (31000-32000) using 'nc' and 'openssl'.

Level 15 → 16
Submit password via SSL-enabled service using 'openssl s_client'.

Level 16 → 17
Scan ports with 'nmap', test SSL with 'openssl', retrieve private SSH key for next login.

Level 17 → 18
Use 'diff' to compare files and spot password difference.

Level 18 → 19
Escape forced command shell; use SSH with options to bypass.

Level 19 → 20
Setuid binary used to execute commands as bandit20 and read their password.

Level 20 → 21

Connect to localhost with special binary and port forwarding to retrieve next password.