# Leviathan

**Completed by: Pratham Sangam**

## Objective:

The primary goal of Leviathan is to progress through each level by discovering the password for the next user account. This is achieved by identifying and exploiting common security oversights within a Linux environment. The challenges are crafted to be approachable, requiring no prior programming knowledge—just a solid understanding of basic Unix/Linux commands and a problem-solving mindset.

## Key concept:

- **File and Directory Permissions**: Understanding how permissions affect access and how misconfigurations can lead to vulnerabilities.
- **Hidden Files and Directories**: Using commands like ls -la to uncover hidden content that may contain crucial information.
- **Binaries**: Recognizing and exploiting binaries that execute with elevated privileges.
- **Command-Line Tools**: Utilizing tools such as **strings**, **ltrace**, to analyse and manipulate binaries.

## Level 0 to 1:

- Objective:

Locate the password for leviathan1.

- Tools Used:

ls, cd, cat,grep

- Solution Logic:

1 A h b k di

1. Access the .backup directory:

```
cd .backup
```

1. Identify the bookmarks.html file.
2. Search for the keyword "password" inside the file: grep "password" bookmarks.html
3. Extract the password from the matching line. ○ Password: L9XFtj2RMk

## Level 1 to 2:

○ Objective:

Discover the password for leviathan2.

○ Tools Used:

file, ltrace

○ Solution Logic:

1. Identify the check binary using ls.
2. Confirm it's an executable file using:

```
file check
```

1. Use ltrace to trace library calls and spot the password comparison: ltrace ./check
2. Use the discovered password to successfully execute the binary. ○ Password: Q3WxvL8YrK

## Level 2 to 3:

Objective:

Gain access to leviathan3.

○ Tools Used: ltrace,

touch, mkdir, bash

○ Solution Logic:

1. Identify the printfile binary using ls.
2. Use ltrace to observe how the binary processes input: ltrace ./printfile
3. Notice that it improperly handles filenames.
4. Craft a file or directory name containing a command injection, such as test; bash.
5. Run the binary with the crafted name to trigger a shell. ○ Password:Z4TmP9GjfN

F0n8h2iWLP

## Level 3 to 4:

○ Objective:

Retrieve the password for leviathan4.

○ Tools Used:

ltrace

○ Solution Logic:

1. Locate the level3 binary using ls.
2. Use ltrace to observe the function calls and spot the expected password: ltrace ./level3
3. Enter the correct password when prompted to gain access. ○ Password: F7RpK2VxQm

## Level 4 to 5:

Objective:

Find the password for leviathan5.

○ Tools Used: ls, cd, ./bin, binary-to-

ASCII conversion

○ Solution Logic:

1. Navigate into the .trash directory using cd .trash.
2. Run the bin executable to get binary output:

./bin

1. Convert the binary output to ASCII (you can use online converters or a script) to reveal the password.

○ Password:J6NtF5WyQb

## Level 5 to 6:

○ Objective:

Access leviathan6.

○ Tools Used:

l b li li k

ln, symbolic links

○ Solution Logic:

1. Create a symbolic link in /tmp that points to the password file: ln -s

/etc/leviathan_pass/leviathan6 /tmp/file.log

2. Run the leviathan5 binary, which reads from /tmp/file.log, revealing the password. ○ Password:H8XzL3PjKv

## Level 6 to 7:

○ Objective:

Obtain the password for leviathan7.

○ Tools Used:

Python scripting, brute-force approach

○ Solution Logic:

1. Write a Python script to brute-force the 4-digit PIN required by the leviathan6 binary.
2. Iterate through all possible combinations until the correct PIN is found and access is granted.

○

○ Password: Y2RmK5FxLw

## Level 7 to 8:

○ Objective:

Complete the final level.

## Conclusion:

The OverTheWire Leviathan lab helped me improve my basic Linux and security skills. By solving small challenges, I learned how to find hidden files, understand permissions, and work with simple binaries. It was a good practice to build my problem-solving skills, and now I feel more confident to move on to tougher cybersecurity tasks.