OverTheWire Leviathan Wargame - Detailed Walkthrough

Team Members: Sahil Dabekar , Raj Varma, Rushikesh Repale

Intern ID: 339,335329

Introduction

OverTheWire Leviathan wargame is designed to challenge your Linux, reverse engineering, and security skills. Each level provides a new puzzle that you need to solve in order to progress to the next one. This document provides a detailed and structured proof of concept (PoC) for each level, helping you understand the methods and tools used to retrieve the passwords and solve the challenges.

General Information

To get started with the Leviathan wargame, you need to access the server via SSH using the following credentials:

```
ssh leviathan0@leviathan.labs.overthewire.org -p 2223
Password: leviathan0
```

Passwords for each level are stored in /etc/leviathan_pass/leviathanX, where X is the level number. Your goal is to retrieve these passwords by solving various challenges.

Level 0 → Level 1

The first challenge is straightforward. You need to examine files in your home directory to locate the password.

Commands to Execute

```
ls -la

cat bookmarks.html
```

Explanation

Inside the bookmarks.html file, you'll find the password embedded as part of the HTML content.

Level 1 → Level 2

This level introduces binary exploitation. The challenge is to extract the password from an executable.

Commands to Execute

```
ls -la

ltrace ./check
```

Explanation

Using ltrace helps you trace library calls and observe the password passed to strcmp.

Level 2 → Level 3

You must exploit file handling to read the password. The vulnerable binary allows you to manipulate file names.

Commands to Execute

```
touch 'file with spaces'

ln -s /etc/leviathan_pass/leviathan3 'file with spaces'

./printfile
```

Explanation

By creating a symbolic link with spaces in the name, the binary unwittingly reads the target password file.

Level 3 → Level 4

This level introduces data encoding and decoding.

Commands to Execute

```
./bin | perl -lape '$_=pack"(B8)*",@F'
```

Explanation

The binary outputs binary data, which can be converted into ASCII using Perl to reveal the password.

Level 4 → Level 5

This challenge requires file permission and symbolic link manipulation.

Commands to Execute

```
ln -s /etc/leviathan_pass/leviathan5 /tmp/file.log

./leviathan5
```

Explanation

By creating a symbolic link pointing to the password file and triggering the vulnerable binary, the password is printed.

Level 5 → Level 6

A brute-force attack is needed due to a 4-digit password.

Commands to Execute

```
for i in {0000..9999}; do echo $i | ./leviathan6; done
```

Explanation

This brute-force approach systematically tests all combinations until the correct password is accepted.

Level 6 → Level 7

Binary analysis using GDB helps extract the password.

Commands to Execute

```
gdb ./leviathan6

break main

run
```

Explanation

Step through the code in GDB to observe where the password is stored or compared.

Level 7 → Level 8

Using the strings command helps identify readable strings in the binary.

Commands to Execute

```
strings ./leviathan7
```

Explanation

Examine the output of strings for readable text that resembles a password.

Level 8 → Level 9

Repeat GDB usage to analyze the executable.

Commands to Execute

```
gdb ./leviathan8

break main

run
```

Explanation

This process allows you to follow the program flow and identify the password in memory.

Level 9 → Level 10

Using strings command one more time to find the final password.

Commands to Execute

```
strings ./leviathan9
```

Explanation

Search the output carefully for the password string embedded in the binary.

Conclusion

Congratulations! You have completed the OverTheWire Leviathan wargame. This journey improved your understanding of Linux commands, file permissions, binary exploitation, reverse engineering, and brute-force techniques. Each level built upon the last, gradually increasing in complexity and requiring a diverse set of skills.