Attack-Defense CTF Competition Writeup

Our first Attack-Defense CTF was a chaotic but educational experience. This writeup details our journey, from initial panic to creative victories and hard-earned lessons.

The Beginning: Complete Chaos

Initial Confusion

As first-timers, we were lost upon logging into our Linux box. We stared at the terminal, unsure where to start. Panic set in.I

First Breakthrough

We found a backup folder with service programs. We located running services on ports 8000-8006.

Game Changer

Using sudo -u servicename bash helped manage services. Then, an enemy attack caused more panic.

Our First Victory: The Auth Service

1 **Vulnerability Found**

The "auth" binary only checked the beginning of the key for authentication.

2 **Exploit Developed**

A script was quickly made to exploit this flaw, gaining easy points.

3 **Service Rewritten**

To be safe, the entire authentication service was rewritten from scratch.

The Great Discovery: Traffic Analysis

Pcaps Folder

We found /pcaps, containing all incoming network traffic. This was our eureka moment.

Pwnazon Vulnerability

A PHP vulnerability in "pwnazon" allowed arbitrary function execution.

Patch and Exploit

We blocked dangerous functions and used the exploit against other teams.

The Filesystem Nightmare

Banananananana Vulnerability

A Local File Inclusion bug allowed attackers to get our flag easily.

Write Access Issues

We couldn't patch services due to lack of write access in /opt/services.

Service Relocation

We stopped services, copied files to writable directories, and moved keys.

Monitoring Script

For pwnazon, a script copied the key file every minute to maintain functionality.

The Powerball Saga: A Comedy of Errors

1

Bounds Checking Bug

Negative array access in "powerball" allowed reading outside intended memory.

2

Shuffle Seed Discovery

The program used the key's first 4 bytes as a shuffle seed for the lottery.

3

Exploit Failure

Initial attempts to grab the flag from memory failed due to a reading bug.

4

Shellcode Injection

A new vulnerability allowed injecting shellcode into the test function.

5

Flags Acquired

This shellcode injection successfully retrieved flags from other teams.

The Racehorse Revelation: Hidden Menus and Backdoors Galore

Hidden Menu

A secret menu entry at "-1" led directly to a give_flag() function.

Built-in Backdoor

A hardcoded backdoor in the main function executed "cat key" to print the flag.

Buffer Overflow Cascade

Mismatched horse structures allowed overwriting function pointers.

Integer Overflow

Large stats in horse_is_awesome() caused unexpected behavior.

Bonus: Powerball Binary Hacked

We discovered someone hacked our Powerball binary. They gained shell access and created a program. This program automatically stole our flag.