🏹 LianYu – TryHackMe CTF Write-Up

Author: hyena
Platform: TryHackMe
Difficulty: Medium
Theme: Arrowverse (Season 1 Inspired)

📌 Overview

LianYu is an Arrowverse-themed Capture The Flag (CTF) machine inspired by Arrow (Season 1). The box does not require prior knowledge of the series, but cleverly uses its lore to guide exploitation paths.

The machine focuses on:

- Enumeration and reconnaissance
- Web discovery and directory bruteforcing
- File analysis and forensics
- Encoding and decoding techniques
- Steganography
- Privilege escalation via misconfigured sudo permissions

🧭 Methodology

I followed a structured penetration testing methodology throughout this challenge:

1. **Initial Enumeration** – Port scanning and service identification
2. **Web Discovery** – Directory bruteforcing and content analysis
3. **Credential Discovery** – Token extraction and decoding
4. **File Analysis** – Forensic examination and header repair
5. **Initial Access** – FTP and SSH exploitation
6. **Privilege Escalation** – Sudo misconfiguration abuse
7. **Post-Exploitation** – Flag capture and documentation

🔍 Step 1: Initial Enumeration

🔹 Port Scanning

I began by running an Nmap scan to identify open ports and services on the target machine:

```bash
nmap -sC -sV -oN nmap.txt 10.48.186.81
```

Key Findings:

The scan revealed several open ports:

- **Port 21** – FTP (vsFTPd 3.0.2)
- **Port 22** – SSH (OpenSSH)
- **Port 80** – HTTP (Apache Web Server)
- **Port 111** – RPC Portmapper
- Additional ports: **58127**

The presence of both FTP and HTTP services immediately suggested multiple potential attack vectors. The web server running on port 80 became my primary focus for initial enumeration.

🌐 Step 2: Web Enumeration

🔹 Manual Browsing

I navigated to the web server at `http://10.48.186.81` and was greeted with a landing page containing Arrowverse-themed imagery and text. The page appeared to be static with minimal interactive elements, prompting me to enumerate hidden directories.

🔹 Directory Bruteforcing

Using Gobuster with a common wordlist, I searched for hidden directories:

```bash
gobuster dir -u http://10.48.186.81 \
-w /usr/share/wordlists/dirb/common.txt \
-x php,txt,html
```

Interesting Discovery:

The scan revealed:

- `/island` – A subdirectory with additional content
- `/island/2100` – A deeper path with story-based hints

🔹 Exploring /island/2100

Visiting `http://10.48.186.81/island/2100` revealed narrative content referencing:

- **Lian Yu** – The island from Arrow
- **Vigilante** – A key character reference
- **Arrowverse lore** – Contextual hints embedded in the page source

The page contained cryptic messages suggesting further enumeration was necessary. I decided to perform a more targeted directory scan on this subdirectory.

🧠 Step 3: Hidden Token Discovery

🔹 Extended Directory Enumeration

I ran another Gobuster scan specifically targeting the `/island/2100` directory with a larger wordlist and custom file extensions:

```bash
gobuster dir -u http://10.48.186.81/island/2100 \
-w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt \
-x ticket
```

The `.ticket` extension was chosen based on hints in the page content referencing a "token" needed to access "Queen's Gambit."

🎯 Critical Finding:

```
/green_arrow.ticket
```

🔹 Token Content

Accessing `http://10.48.186.81/island/2100/green_arrow.ticket` revealed:

```
This is just a token to get into Queen's Gambit (Ship)
RTy8yhBQdscX
```

The string `RTy8yhBQdscX` appeared to be encoded credentials.

🔐 Step 4: Decoding Credentials

🔹 Identifying the Encoding

The string `RTy8yhBQdscX` had characteristics of Base58 encoding:

- No ambiguous characters (0, O, I, l)
- Fixed length
- Alphanumeric composition

🔹 Base58 Decoding

I used CyberChef and command-line tools to decode the string:

```bash
echo "RTy8yhBQdscX" | base58 -d
```

Result: `!#th3h00d`

This appeared to be a password. Combined with the "vigilante" reference found earlier, I had potential FTP credentials.

📂 Step 5: FTP Access

🔹 FTP Login

With the decoded password, I attempted to connect to the FTP service:

```bash
ftp 10.48.186.81
```

Credentials:

- Username: `vigilante`
- Password: `!#th3h00d`

✅ **Login successful!**

🔹 FTP Directory Listing

After connecting, I explored the FTP structure:

```bash
ftp> ls
ftp> cd ..
ftp> ls -la
```

Directories Found:

- `slade/`
- `vigilante/`

Inside the `vigilante/` directory, I discovered three image files:

- `Leave_me_alone.png`
- `Queen's_Gambit.png`
- `aa.jpg`

🔹 Downloading Files

I downloaded all files for local analysis:

```bash
ftp> binary
ftp> mget *
```

🖼️ Step 6: File Analysis & Forensics

🔹 Initial File Examination

I attempted to open the downloaded images, but `Leave_me_alone.png` failed to display. This suggested file corruption or manipulation.

🔹 Checking File Signatures

I inspected the file headers using `xxd`:

```bash
xxd -l 16 Leave_me_alone.png
```

Output:

```
00000000: 5546 5247 0d0a 1a0a 0000 000d 4948 4452  UFRG........IHDR
```

❌ Invalid PNG Header Detected

The file began with `UFRG` instead of the correct PNG magic bytes.

🔹 Understanding PNG File Structure

A valid PNG file must start with the following 8-byte signature:

```
89 50 4E 47 0D 0A 1A 0A
```

Breaking down the signature:

- `89` – Non-ASCII character to detect text file corruption
- `50 4E 47` – "PNG" in ASCII
- `0D 0A` – DOS line ending (CRLF)
- `1A` – DOS end-of-file character
- `0A` – Unix line ending (LF)

🔹 Repairing the PNG Header

I used the `printf` and `dd` commands to overwrite the corrupted header:

```bash
printf '\x89\x50\x4E\x47\x0D\x0A\x1A\x0A' | \
dd of=Leave_me_alone.png bs=1 seek=0 count=8 conv=notrunc
```

Verification:

```bash
file Leave_me_alone.png
```

Output:

```
Leave_me_alone.png: PNG image data, 1920 x 1080, 8-bit/color RGB
```

✔️ **Success!** The image now opened correctly.

🔹 **Analyzing the Repaired Image**

Opening the repaired PNG revealed embedded text containing a password:

```
password: M3tahuman
```

This password would be crucial for the next phase of exploitation.

🕵️ Step 7: Steganography

🔹 **Extracting Hidden Data**

With the password `M3tahuman`, I suspected steganography in the other image files. I used `steghide` to extract hidden content from `aa.jpg`:

```bash
steghide extract -sf aa.jpg
```

Passphrase: `M3tahuman`

Extracted File: `ss.zip`

🔹 **Analyzing the ZIP Archive**

I unzipped the archive:

```bash
unzip ss.zip
```

Contents:

- `passwd.txt`
- `shado`

🔹 **Examining the Files**

passwd.txt:

```
This is your visa to Land on Lian_Yu # Just for Fun ***
a small Note about it
Having spent years on the island, Oliver learned how to be resourceful and
set booby traps all over the island in the common event he ran into dangerous
people. The island is also home to many animals, including pheasants,
wild pigs and wolves.
```

shado file:

```
M3tahuman
```

The `shado` file contained what appeared to be a password. Combined with the username hint "slade" (from the FTP directory structure), I had potential SSH credentials.

🔑 Step 8: SSH Access

🔹 **SSH Login**

I attempted to connect via SSH using the discovered credentials:

```bash
ssh slade@10.48.186.81
```

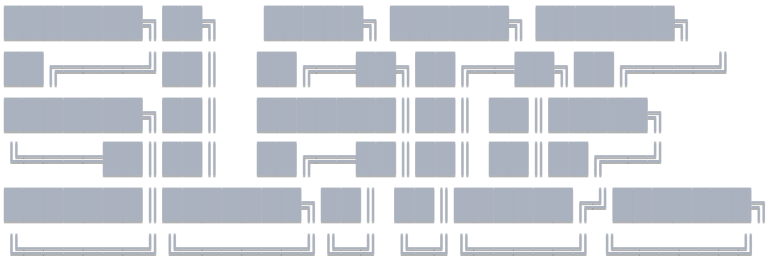
Credentials:

- Username: `slade`
- Password: `M3tahuman`

✅ Login successful!

Upon successful authentication, I was greeted with a custom ASCII art banner:

```
Way To SSH...
Loading.....Done..
Connecting To Lian_Yu  Happy Hacking
```



👤 Step 9: User Enumeration

🔹 Initial Exploration

After gaining shell access, I explored the user's home directory:

```
bash

ls -la
```

Files Found:

- `user.txt` – First flag
- `.Important` – Hidden file with escalation hints

🔹 User Flag

```
bash

cat user.txt
```

Flag: `THM{P30P7E_K33P_53CRET5__C0MPUT3R5_D0N'T}`

🔹 Analyzing .Important

```
bash

cat .Important
```

Content:

Slade Wilson was 16 years old when he enlisted in the United States Army, having lied about his age. After serving a stint in Korea, he was later assigned to Camp Washington where he had been promoted to the rank of major. In the early 1960s, he met Captain Adeline Kane, who was tasked with training young soldiers in new fighting techniques in anticipation of brewing troubles taking place in Vietnam. Kane was amazed at how skilled Slade was and how quickly he adapted to modern conventions of warfare. She immediately fell in love with him and realized that he was without a doubt the most able-bodied combatant that she had ever encountered. She offered to privately train Slade in guerrilla warfare. In less than a year, Slade mastered every fighting form presented to him and was soon promoted to the rank of lieutenant colonel. Six months later, Adeline and he were married and she became pregnant with their first child. The war in Vietnam began to escalate and Slade was shipped overseas. In the war, his unit massacred a village, an event which sickened him. He was also rescued by SAS member Wintergreen, to whom he would later return the favor.

Chosen for a secret experiment, the Army imbued him with enhanced physical powers in an attempt to create metahuman soldiers for the U.S. military. Deathstroke became a mercenary soon after the experiment when he defied orders and rescued his friend Wintergreen, who had been sent on a suicide mission by a commanding officer with a grudge.[7] However, Slade kept this career secret from his family, even though his wife was an expert military combat instructor.

A criminal named the Jackal took his younger son Joseph Wilson hostage to force Slade to divulge the name of a client who had hired him as an assassin. Slade refused, claiming it was against his personal honor code. He attacked and killed the kidnappers at the rendezvous. Unfortunately, Joseph's throat was slashed by one of the criminals before Slade could prevent it, destroying Joseph's vocal cords and rendering him mute.

After taking Joseph to the hospital, Adeline was enraged at his endangerment of her son and tried to kill Slade by shooting him, but only managed to destroy his right eye. Afterward, his confidence in his physical abilities was such that he made no secret of his impaired vision, marked by his mask which has a black, featureless half covering his lost right eye. Without his mask, Slade wears an eyepatch to cover his eye.

At the bottom of this lengthy backstory was a critical hint:

What you are Looking for is in the secret directory :)

🚀 Step 10: Privilege Escalation

🔹 Sudo Permission Check

I checked what commands the user could run with sudo privileges:

bash

```
sudo -l
```

Output:

```
Matching Defaults entries for slade on LianYu:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User slade may run the following commands on LianYu:
    (root) PASSWD: /usr/bin/pkexec
```

🚨 Critical Misconfiguration Detected

The user `slade` could execute `/usr/bin/pkexec` as root with password authentication. This binary allows policy-based privilege escalation and can be used to spawn a root shell.

👑 Step 11: Root Access

🔹 Exploiting pkexec

I leveraged the sudo permission to execute a root shell:

```bash
sudo /usr/bin/pkexec /bin/bash
```

After entering the password `M3tahuman`, I obtained root access:

```bash
whoami
```

Output: `root`

🔹 Root Flag

I navigated to the root directory and captured the final flag:

```bash
cd /root
ls -la
cat root.txt
```

Flag:

`THM{MY_W0RD_I5_MY_B0ND_IF_I_ACC3PT_YOUR_CONTRACT_THEN_IT_WILL_BE_COMPL3TED_OR_I'L`

L_BE_D34D}

🎉 Mission Accomplished!

The final message displayed:

```
Mission accomplished

You are injected me with Mirakuru:) --> Now slade Will become DEATHSTROKE.

THM{MY_W0RD_I5_MY_B0ND_IF_I_ACC3PT_YOUR_CONTRACT_THEN_IT_WILL_BE_COMPL3TED_OR_I'LL_
                                                --DEATHSTROKE

Let me know your comments about this machine :)
I will be available @twitter @User6825
```

🏁 Captured Flags

User Flag

```
THM{P30P7E_K33P_53CRET5__C0MPUT3R5_D0N'T}
```

Root Flag

```
THM{MY_W0RD_I5_MY_B0ND_IF_I_ACC3PT_YOUR_CONTRACT_THEN_IT_WILL_BE_COMPL3TED_OR_I'LL_
```

⚠️ Challenges Faced

Throughout this CTF, I encountered several technical challenges that required problem-solving and research:

1. **Understanding File Magic Bytes** – Recognizing corrupted file signatures and knowing the correct PNG header structure
2. **Fixing Corrupted PNG Headers** – Using hexadecimal manipulation tools to repair file headers
3. **Identifying Base58 Encoding** – Distinguishing between various encoding schemes
4. **Recognizing pkexec as Privilege Escalation Vector** – Understanding PolicyKit exploitation
5. **Avoiding Red-Herring Services** – Not getting distracted by RPC and other non-exploitable services

6. **Steganography Password Discovery** – Understanding the chain of clues leading to hidden data

---

📚 What I Learned

This CTF provided valuable hands-on experience with several cybersecurity concepts:

Technical Skills

- **File Signature Analysis** – Understanding magic bytes and file format specifications
- **PNG Header Structure** – Learning the specific byte sequence required for valid PNG files
- **Encoding vs Encryption** – Distinguishing between Base58, Base64, and other encoding schemes
- **Steganography Basics** – Using tools like steghide for data extraction
- **Linux Privilege Escalation** – Exploiting sudo misconfigurations
- **Enumeration Methodology** – The importance of thorough and systematic reconnaissance

Security Concepts

- **Defense in Depth** – Multiple layers of obfuscation can protect sensitive data
- **Least Privilege Principle** – Overly permissive sudo configurations create security risks
- **Information Disclosure** – Even non-sensitive files can contain valuable reconnaissance information
- **Password Reuse** – Using the same password across multiple contexts is a vulnerability

---

🔐 Security Recommendations

Based on the vulnerabilities discovered in this CTF, I recommend the following security controls:

1. Restrict Sudo Permissions

- Implement the principle of least privilege
- Audit sudo configurations regularly
- Avoid granting access to privilege escalation binaries like pkexec without strong justification
- Use sudo logs for monitoring and alerting

2. Avoid Password Reuse

- Implement unique passwords for different services and accounts

- Use password managers to maintain password complexity
- Enforce password rotation policies
- Consider implementing multi-factor authentication

3. Disable Unnecessary Services

- Remove or disable unused network services (RPC, SMB, etc.)
- Reduce the attack surface by minimizing exposed ports
- Implement host-based firewalls
- Regular security audits of running services

4. Validate File Uploads

- Implement file type validation beyond extension checking
- Verify file magic bytes match declared file types
- Scan uploaded files for malicious content
- Restrict file upload locations and permissions

5. Monitor for Steganographic Content

- Implement data loss prevention (DLP) solutions
- Monitor for unusual file patterns
- Educate users about information hiding techniques
- Consider steganography detection tools in security stack

6. Implement Security Hardening

- Follow CIS benchmarks for system configuration
- Regular security patching and updates
- Implement intrusion detection systems
- Conduct regular penetration testing and vulnerability assessments

---

🧠 Final Thoughts

LianYu was an exceptionally well-crafted CTF challenge that balanced entertainment with education. The machine successfully incorporated:

Strengths

- **Creative Storytelling** 🎨 – The Arrowverse theme provided engaging context
- **Educational Value** 📘 – Multiple realistic exploitation techniques
- **Progressive Difficulty** – Logical escalation from enumeration to privilege escalation
- **Attention to Detail** – Subtle hints throughout the narrative

Learning Outcomes

This CTF reinforced several critical cybersecurity principles:

- The importance of systematic enumeration
- File forensics and metadata analysis
- Multiple encoding/obfuscation techniques
- Real-world privilege escalation scenarios
- The value of patience and attention to detail

Community Appreciation

Special thanks to the creator (@User6825 on Twitter) for designing this engaging and educational challenge. The integration of popular culture with technical security concepts made this an enjoyable learning experience.

Rating: ⭐⭐⭐⭐⭐ (5/5)

A perfect blend of entertainment and technical depth, suitable for intermediate security practitioners looking to strengthen their penetration testing skills.

📞 Contact

TryHackMe: hyena
Completed: January 22, 2026

This write-up is intended for educational purposes and authorized security testing only. Always obtain proper authorization before conducting security assessments.