Initial Approach

Since direct access to the VM was unavailable, the first step was to locate and analyze the **OVA file** provided by the challenge.

1. Extracting the OVA

The .ova file was extracted using 7z, which revealed a .vmdk virtual disk image.

```
7z x b2r.ova
```

Disk & Filesystem Analysis

2. Identifying Partitions

Standard disk utilities like fdisk were not available in the environment, so the extracted **VMDK** was analyzed directly. The partition was identified as an **ext4 filesystem**.

3. Mounting the Filesystem

The ext4 partition was mounted locally to inspect its contents:

```
mount -o loop disk.vmdk /mnt/vm
```

This gave full read access to the VM's filesystem.

Enumeration Phase

4. User Enumeration

Inside /home, two user accounts were discovered:

- devops
- auditor

Both users had .bash_history files available, which is often a goldmine during offline analysis.

5. Bash History Review

Reviewing the bash histories revealed previously executed commands, offering insight into:

- Internal scripts
- Maintenance utilities
- Potential misconfigurations

This hinted at custom tooling being used for system checks.

Privilege Escalation Discovery

6. Finding SUID Binaries

A search for **SUID binaries** revealed an interesting file:

/usr/local/bin/syscheck

Key observations:

- Owned by **root**
- Has the **setuid bit** enabled
- Custom binary (not a standard system utility)

This immediately suggested a possible privilege escalation vector.

Binary Analysis

7. Analyzing syscheck

Static analysis of the syscheck binary showed that:

- It calls system() internally
- It attempts to execute a script named **check.sh**
- The script path is **not hardcoded with an absolute path**

This is a classic vulnerability:

Executing external commands without an absolute path in a SUID binary allows PATH hijacking.

Even though the live exploit path isn't required here (since we have filesystem access), this confirms a **broken trust chain** between root and user-controlled execution context.

Flag Retrieval

8. Accessing Root Files

Since we already had full access to the mounted filesystem, the final step was to check root-owned files.

The flag was located at:

/root/flag.txt

Flag

SECE{ch41n_th3_m1st4k3s_n0t_c0mm4nds}