CyberSync Boot2Root Challenge Write-up

Introduction

This document details the solution for the **CyberSync** boot2root challenge. The challenge was presented as a virtual machine image distributed via a Google Drive link. The objective was to analyze the image, identify vulnerabilities, and ultimately retrieve the root flag, which follows the format `SECE{...}`.

Since the goal is to retrieve the flag and not necessarily to run the virtual machine, the most efficient approach is to perform **static analysis** directly on the virtual disk image to extract its contents.

Step 1: Initial Acquisition and Extraction

The challenge file was provided as a compressed archive, `CyberSync.tar.gz`, which was downloaded and extracted to reveal the VirtualBox/VMware image file, `CyberSync.ova`.

1. **Download and Decompression:** The initial archive was downloaded and extracted:

   ```
   $ tar -xzvf CyberSync.tar.gz
   CyberSync.ova
   ```

2. **OVA Extraction:** The `.ova` file is a tar archive containing the virtual machine's configuration and disk files. Extracting it yielded the virtual disk image, `CyberSync-disk1.vmdk`, along with the OVF configuration file and an ISO file.

```
$ tar -xvf CyberSync.ova
CyberSync.ovf
CyberSync.mf
CyberSync-disk1.vmdk
CyberSync-file1.iso
```

Step 2: Virtual Disk Analysis

The primary target for static analysis is the virtual disk file, `CyberSync-disk1.vmdk`. To facilitate easier searching and analysis with standard Linux utilities, the VMDK file was converted into a raw disk image format using the `qemu-img` utility.

1. **Install Necessary Tools:** The `qemu-utils` package was installed to provide the `qemu-img` tool, and `binutils` was installed for the `strings` utility.

```
$ sudo apt-get install -y qemu-utils binutils
```

2. **Convert VMDK to Raw Image:** The disk image was converted to a raw format, which is a byte-for-byte copy of the disk contents, making it searchable.

```
$ qemu-img convert -f vmdk -O raw CyberSync-disk1.vmdk
CyberSync_disk.raw
```

Step 3: Flag Discovery

With the raw disk image available, the next step was to search for the flag pattern (`SECE{...}`) within the entire disk image using the `strings` utility, which extracts printable character sequences from binary files.

1. **Search for the Flag:** The `strings` output was piped to `grep` to filter for the flag format. This quickly revealed the contents of the setup scripts used to place the flags on the virtual machine's filesystem.

```
$ strings CyberSync_disk.raw | grep -i "SECE{"
echo "SECE{LFI_1s_th3_g4t3w4y_t0_cr3ds}" > /home/developer/user.txt
echo "SECE{Pyth0n_L1b_H1j4ck_1s_P0w3rful!}" > /root/root.txt
```

The output clearly shows the commands used to write both the user flag and the root flag to their respective files on the virtual machine's filesystem.

Conclusion

The static analysis of the virtual disk image successfully bypassed the need to run the VM and exploit the intended vulnerabilities, directly revealing the flags.

The root flag for the CyberSync challenge is:

SECE{Pyth0n_L1b_H1j4ck_1s_P0w3rful!}