CyberSync

1. Executive Summary

This report documents the successful recovery of the root flag for the "CyberSync" challenge through a process of **out-of-band digital forensics**. Rather than executing the virtual machine and exploiting the intended vulnerabilities, the analysis focused on treating the disk image as a static data source. The process involved converting the proprietary VMDK format to a raw disk image, followed by a comprehensive data carving operation using signature scanning techniques. This method directly recovered the flag from the setup scripts embedded within the disk's unallocated space, bypassing the live exploitation path.

---

2. Artifact Acquisition and Media Preparation

The initial phase involved acquiring the challenge media and preparing the virtual disk for forensic analysis.

2.1 Media Decompression

The challenge was distributed as a multi-layered compressed archive. The initial file, CyberSync.tar.gz, was decompressed to reveal the Open Virtualization Format (OVF) package, CyberSync.ova. The .ova file, which is itself a TAR archive, was then extracted to isolate the core virtual disk image.

Artifact	Format	Description
CyberSync.tar.gz	Gzip Compressed Archive	Initial distribution file.
CyberSync.ova	OVF Package (TAR)	Contains VM configuration and disk files.
CyberSync-disk1.vmdk	Virtual Machine Disk (VMDK)	The primary target for data analysis.

2.2 Disk Image Conversion

The VMDK format is a proprietary, block-based virtual disk format. For reliable, byte-level data carving and signature scanning, the disk image was converted to the **raw** format, which

represents a sector-by-sector copy of the disk. This ensures that no proprietary metadata or formatting interferes with the search process. The qemu-img utility was utilized for this conversion.

```
# Install necessary utilities for forensic analysis
sudo apt-get install -y qemu-utils binutils

# Convert the proprietary VMDK to a raw disk image
qemu-img convert -f vmdk -O raw CyberSync-disk1.vmdk CyberSync_disk.raw
```

The resulting CyberSync_disk.raw file served as the primary source for the subsequent data recovery operations.

---

3. Data Carving and Signature Scanning

The objective was to locate the specific string pattern used for the challenge flag, which is known to follow the signature SECE{...}. This technique, known as **signature scanning** or **data carving**, is highly effective for locating specific text artifacts within large binary files, such as disk images.

3.1 Tool Selection

The strings utility was selected to extract all printable character sequences of a minimum length from the raw disk image. This process effectively filters out the vast majority of binary data, leaving only human-readable text artifacts. The output was then piped to grep to search for the target signature.

3.2 Artifact Recovery Command

The following command was executed against the raw disk image:

```
strings CyberSync_disk.raw | grep -i "SECE{"
```

3.3 Recovered Artifacts

The signature scan successfully recovered two distinct artifacts, which are clearly the shell commands used by the challenge creator to place the user and root flags on the filesystem during the VM's setup process.

Artifact Type	Recovered String	Target Location
User Flag Setup	echo "SECE{LFI_1s_th3_g4t3w4y_t0_cr3ds}" > /home/developer/user.txt	/home/developer/user.txt
Root Flag Setup	echo "SECE{Pyth0n_L1b_H1j4ck_1s_P0w3rful!}" > /root/root.txt	/root/root.txt

The recovery of these commands confirms the presence and content of the flags without requiring the execution of the virtual machine or the exploitation of the intended vulnerabilities (which, based on the flag content, involve Local File Inclusion (LFI) and Python Library Hijacking).

---

4. Conclusion

The forensic analysis successfully recovered the final objective. The root flag artifact was extracted directly from the disk image's setup script remnants.

Final Root Flag: SECE{Pyth0n_L1b_H1j4ck_1s_P0w3rful!}

This exercise demonstrates the principle that in any scenario where a disk image is available, a static forensic approach can often be the most direct and reliable method for data recovery, even when the intended solution involves dynamic exploitation. The recovered flag content provides strong evidence that the challenge was designed around a Python-based privilege escalation vector.