**CVE-2020-0787**

**WINDOWS BACKGROUND INTELLIGENT TRANSFER SERVICE**

**ELEVATION OF PRIVILEGE VULNERABILITY**

**Subject name:** HOD401 – Offensive Security and Ethical Hacking

**Student name:** Ngo Phuong Ha

**Student ID:** SE140479

**Date of submission:** 10/06/2021

1. **Overview**

An elevation of privilege vulnerability exists when the Windows Background Intelligent Transfer Service (BITS) improperly handles **symbolic links**, which are “virtual” files or folders which reference a physical file or folder located elsewhere, and are an important feature built into many operating systems, including Linux and Windows. An attacker who successfully exploited this vulnerability could overwrite a targeted file leading to an elevated status.

To exploit this vulnerability, an attacker would first have to log on to the system. An attacker could then run a specially crafted application that could exploit the vulnerability and take control of an affected system.

The security update addresses the vulnerability by correcting how Windows BITS handles symbolic links.

For more information of this vulnerability and how BITS classes work, please refer to the link:

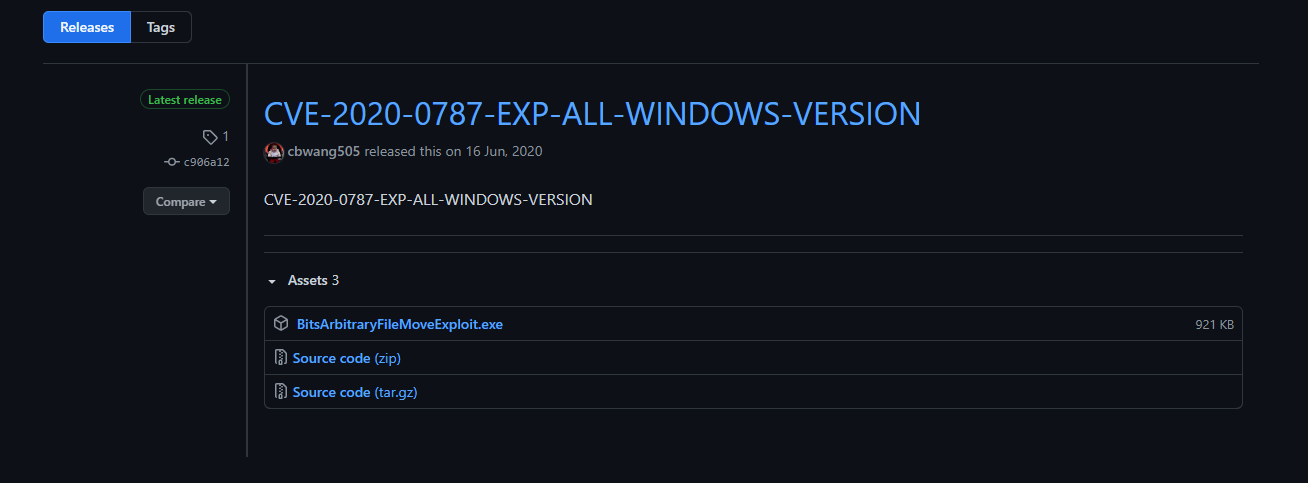
<https://itm4n.github.io/cve-2020-0787-windows-bits-eop/>

**Affected systems:** Windows 7/8/10, Windows Server 2016,…

To check for detailed versions, please refer to the link: <https://msrc.microsoft.com/update-guide/en-US/vulnerability/CVE-2020-0787>

1. **Implementation**
2. **Exploit with executable file**

**Step 1:** Download the executable file at: <https://github.com/cbwang505/CVE-2020-0787-EXP-ALL-WINDOWS-VERSION/releases> Save it somewhere in your machine.

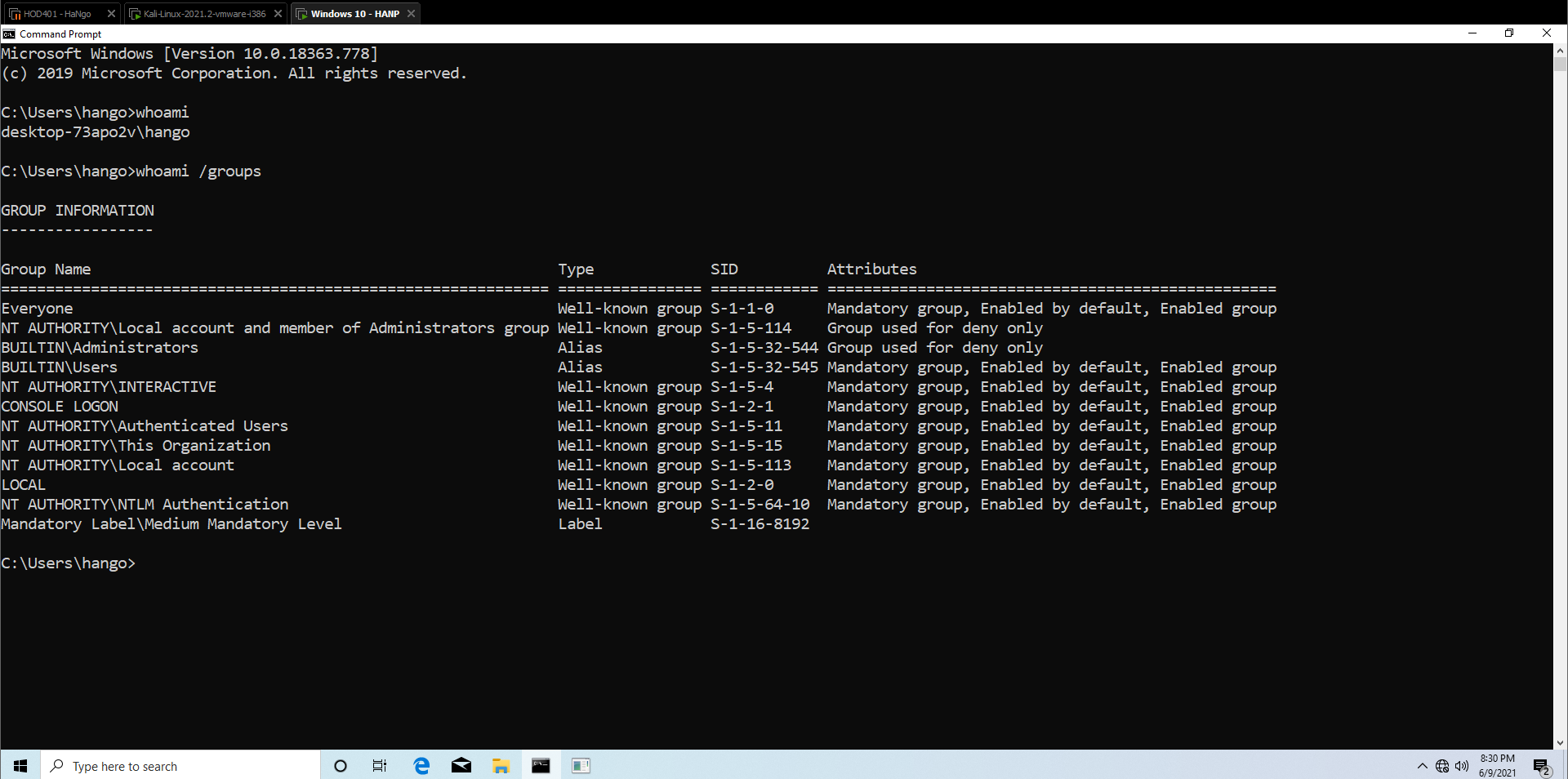


**Step 2:** Login Windows machine as low-privileged user. Check it with the command:

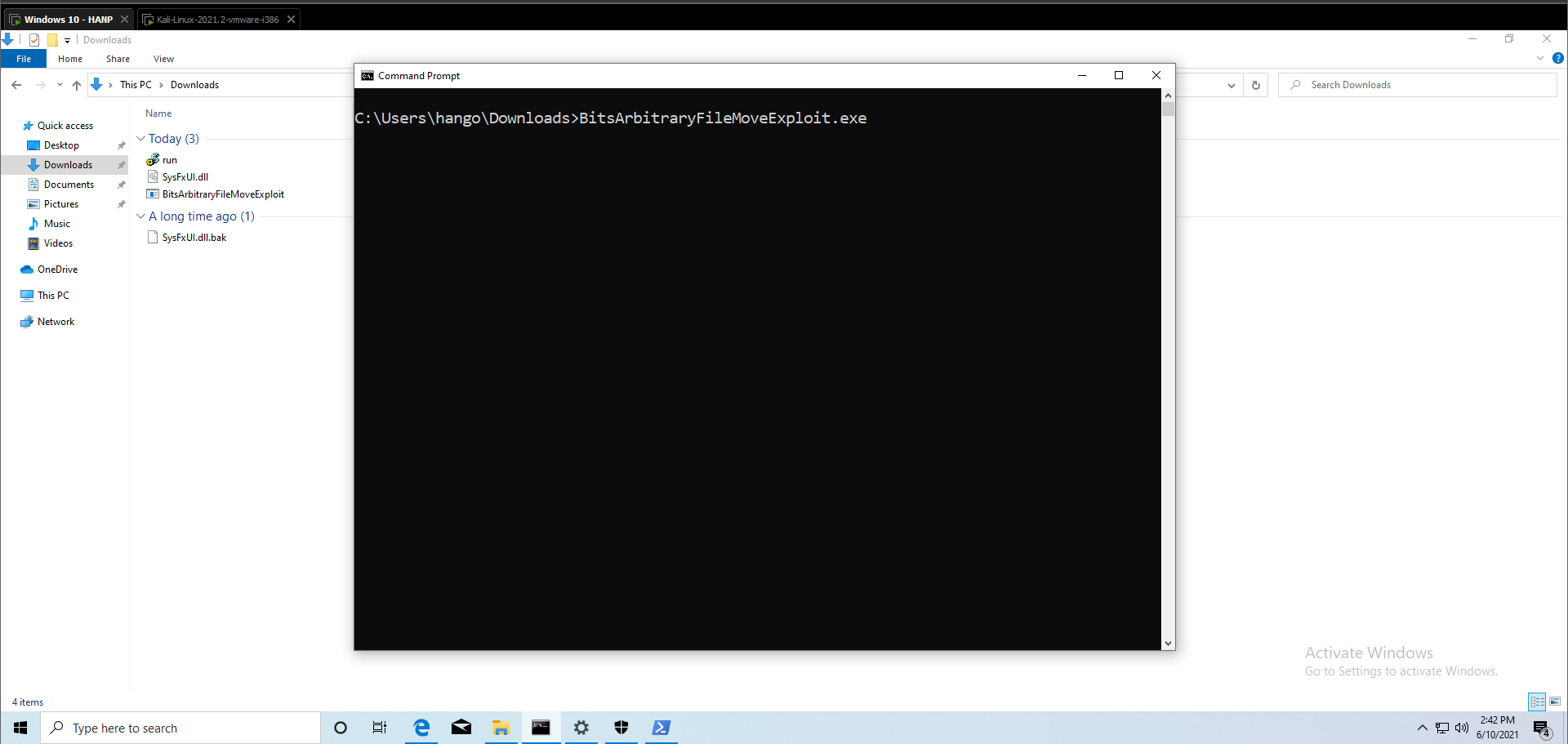
*whoami*

*whoami /groups*

It is observed that group names containing Administrator are used for deny only, which means the current user is a normal one.

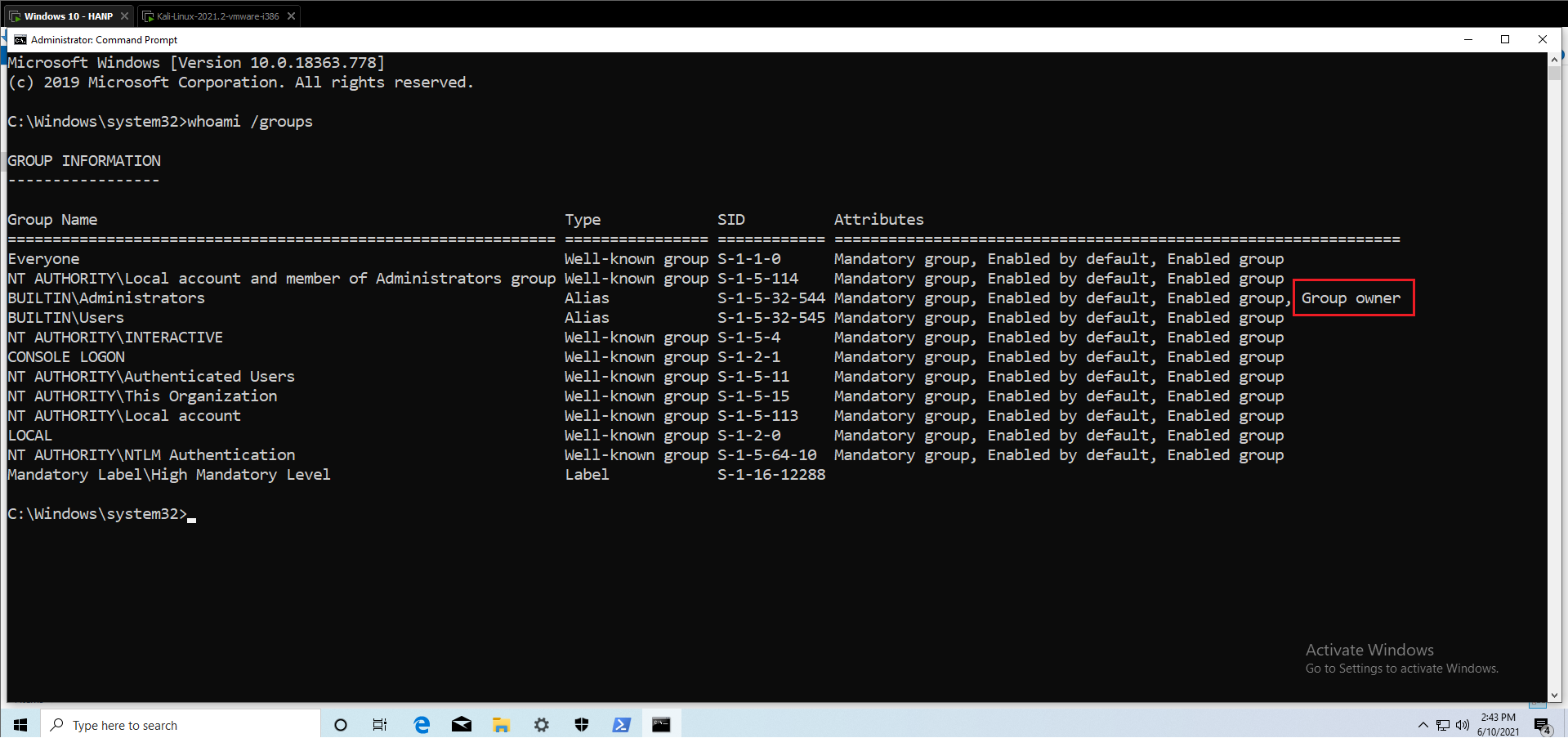


**Step 3:** Navigate to the directory containing this file. Click *Shift + right-click* to open *cmd.exe* in that directory. Then execute this file by the command: *BitsArbitraryFileMoveExploit.exe*



**Step 4:** Wait a few minutes until it finishes. Then, a **new** Command Line pops up. This time, to check if you have successfully escalated or not, type the command:

*whoami /groups*

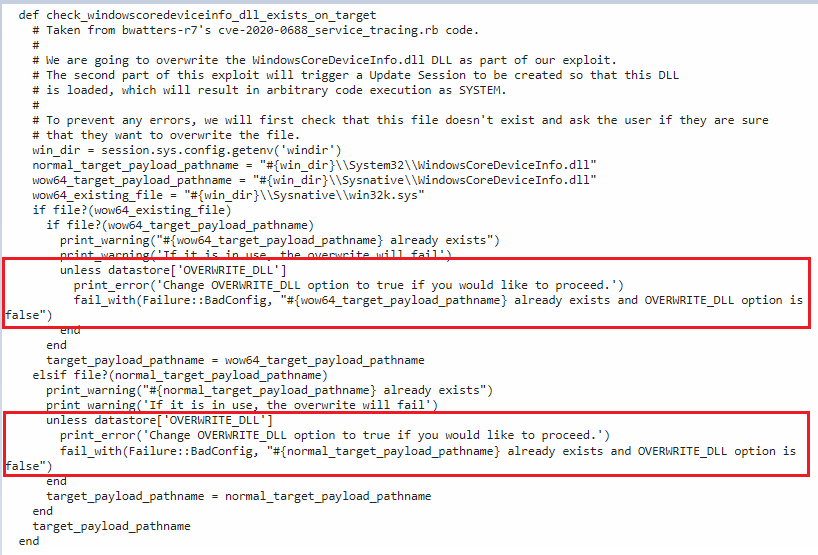


Now, the attribute description is different. The current user is *Group owner* of *BUILTIN\Administrators.*

1. **Exploit with Metasploit module**

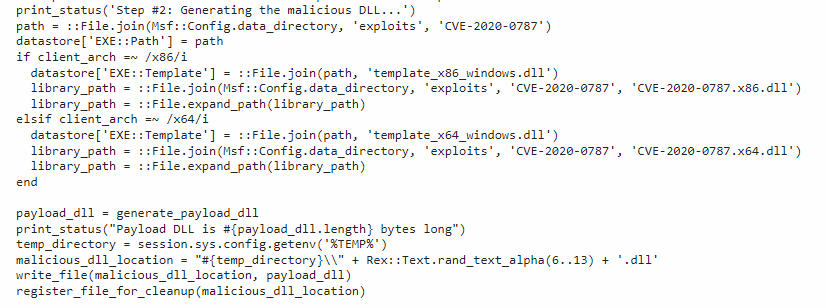
Metasploit’s *exploit/windows/local/cve\_2020\_0787\_bits\_arbitrary\_file\_move* module is written to overwrite *C:\Windows\System32\WindowsCoreDeviceInfo.dll* with a malicious DLL containing the attacker's payload.

The following screenshot is a piece of code written in the module, which aims to check if *WindowsCoreDeviceInfo.dll* exists on the target machine and if DLL overwriting is possible . Here, the overwrite operation only succeeds if the *OVERWRITE\_DLL* option is set to true AND the target payload pathname does not exist.



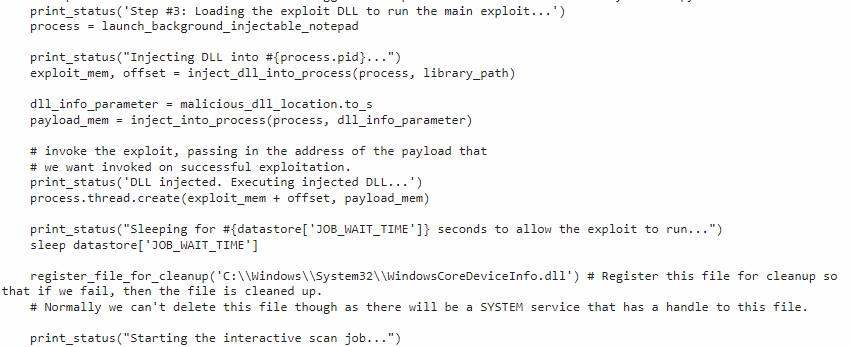
To achieve code execution as the SYSTEM user, the **Update Session Orchestrator** service (Windows OS component that orchestrates the sequence of downloading and installing various update types from *Windows Update*.) is then started, which will result in the malicious *WindowsCoreDeviceInfo.dll* being run with SYSTEM privileges due to a DLL hijacking issue within the *Update Session Orchestrator Service*.

After environment checking, the module generates the malicious DLL and insert it in a temp location, as indicated in the following screenshot:



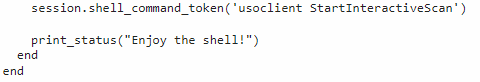
This module works well with both x64-based and x32-based Windows machine.

Next, it loads the main DLL to trigger the exploitation and conduct the arbitrary file copy:



Note that as the bottom lines of code indicate, if the exploitation fails, the malicious DLL file would be removed.

Finally, it executes USO service to trigger the payload:

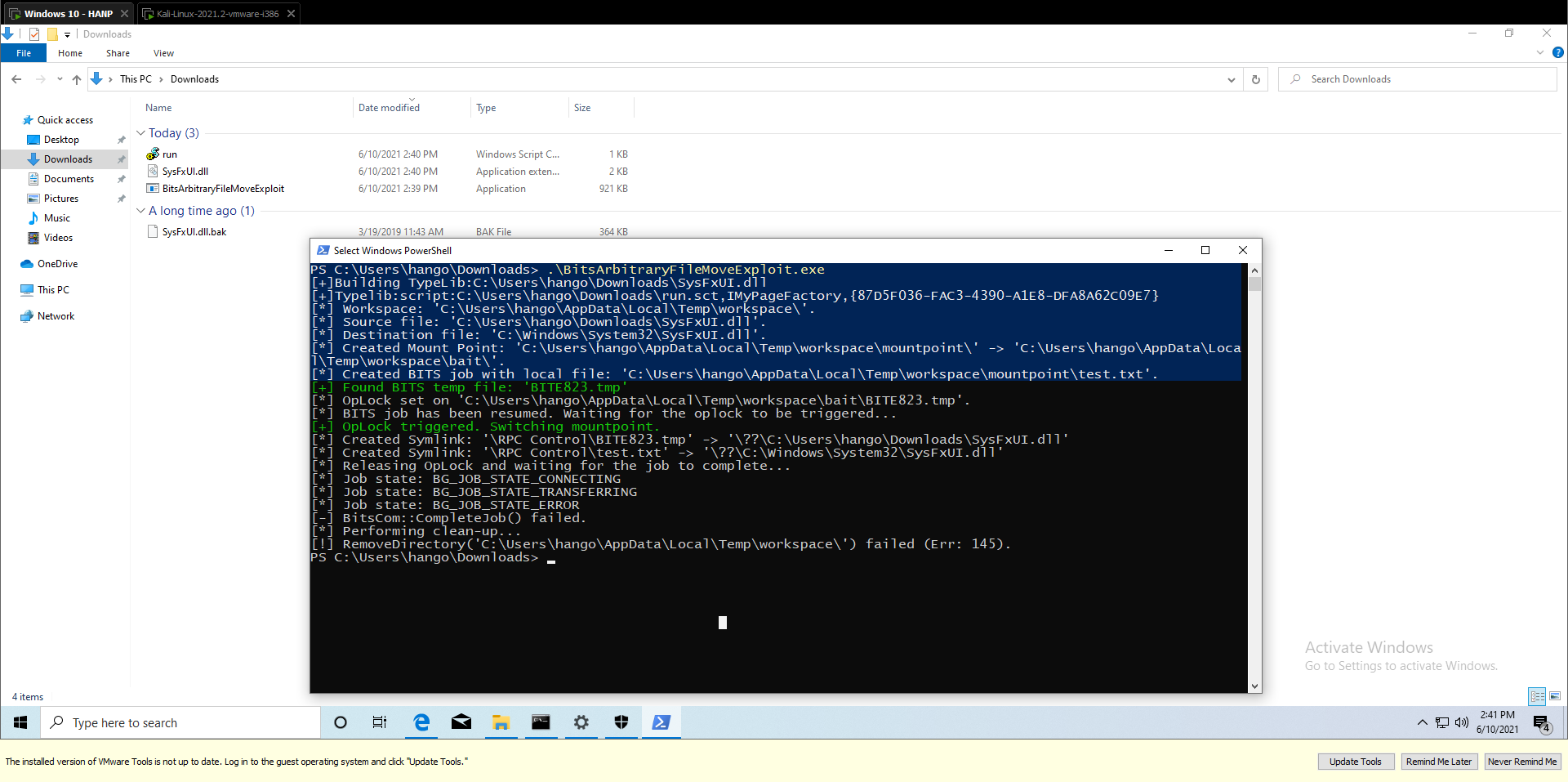


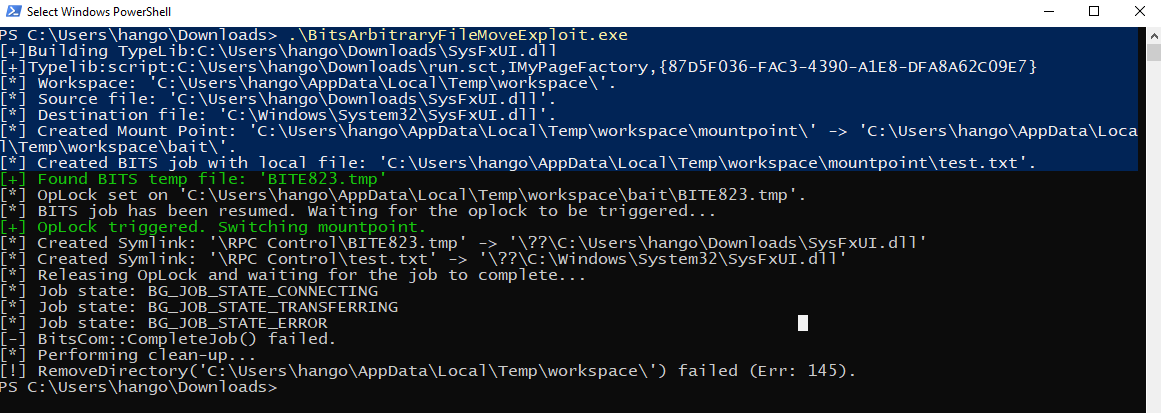
For more information of this exploitation module, please refer to the link:

<https://packetstormsecurity.com/files/158056/Background-Intelligent-Transfer-Service-Privilege-Escalation.html>

1. **Solution**

This issue has been fixed in the following updates. Running the malicious file in an up-to-date machine would return the result like this:





As analyzed above, if the exploitation fails, the temp directory storing the malicious file would be removed.