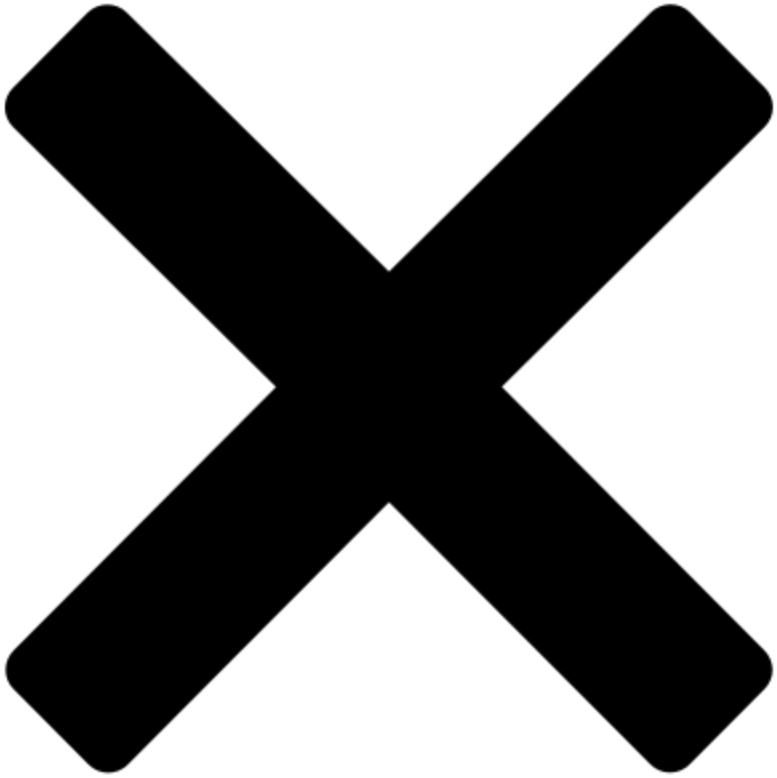
Escalating Privileges via Third-Party Windows Installers

mandiant.com/resources/blog/privileges-third-party-windows-installers



Picture this: you've finally made it past the perimeter of a highly secured organization. You're feeling pretty pleased with yourself, until you realize you only have Active Directory privileges of a newly hired intern and the thrill trickles away. However, with some crafty tricks and a bit of luck, you just might be able to climb the corporate ladder and get promoted to SYSTEM. Welcome to the high-stakes game of privilege escalation!

For red teamers, elevation of privilege attacks come in two forms: domain and local privilege escalation. Domain privilege escalation attacks focus on exploitation of Active Directory or Cloud misconfigurations and vulnerabilities. Such attacks include Kerberoasting, file share enumeration, and the notorious Zerologon (CVE-2020-1472) vulnerability. Given the prevalence of Active Directory misconfigurations in modern networks, domain privilege escalation attacks are among the first things red teamers look for after obtaining a foothold in

an internal network. But what happens when traditional domain escalations fail? As red teamers, often the only way forward is to elevate locally, and sometimes that may require taking a couple steps back and researching for these escalation vulnerabilities.

Local privilege escalation attacks focus on elevating a standard user's privileges on a system to local administrator privileges. Such attacks may include exploitation of insecure service permissions or insecure file permissions. Unlike domain privilege escalation attacks, many red teamers overlook local privilege escalation attacks. This is because domain privilege escalation paths are usually more fruitful and easier to take advantage of. However, successful exploitation on a local system could segue into more significant attacks at a domain level. This has been demonstrated with research on Active Directory Certificate Services (AD CS) misconfigurations and credential recovery of Microsoft's System Center Configuration Manager (SCCM) network access account (NAA). After all, this is the Internet, and everything is connected.

In this blog post, we will share how Mandiant's red team researches and exploits zero-day vulnerabilities in third-party Windows Installers, what software developers should do to reduce risk of exploitation, and introduce a new tool to simplify enumeration of cached Microsoft Software Installer (MSI) files: **msi_search**.

Microsoft Software Installer Background

MSI files are database files that contain data and instructions for installing and uninstalling software on Windows operating systems. Organizations commonly use MSI files for their standardized format, which makes it easy to manage the installation, maintenance, and removal of software. MSI files also provide flexibility to software developers by allowing them to execute additional code during installation, removal, or repairs through Custom Actions. These actions enable developers to customize the installation process and perform specific tasks as needed.

When installing software with MSI files, Windows caches them in the `C:\Windows\Installer` folder using randomized filenames consisting of alphanumeric characters followed by the ".msi" extension. This allows standard users to access and use the "repair" feature, which is intended to address various issues like missing files, broken shortcuts, incorrect registry entries, and other software malfunctions. During MSI repairs, several operations, such as file creation or execution, may be triggered from a `NT AUTHORITY\SYSTEM` context, even if initiated by a standard user.

Potential Abuses

The ability to initiate an operation from a `NT AUTHORITY\SYSTEM` context can present potential security risks if not properly managed. For instance, misconfigured Custom Actions running as `NT AUTHORITY\SYSTEM` can be exploited by attackers to execute local privilege escalation

attacks. One particular misconfiguration involves performing file operations in a folder where standard users have write privileges. This allows attackers to modify files used by `NT AUTHORITY\SYSTEM`, enabling them to run arbitrary code and elevate their privileges.

Microsoft's Process Monitor (ProcMon) can be used to analyze and monitor file operations on a Windows system. To filter the results and focus only on file operations from a `NT AUTHORITY\SYSTEM` process executed within a folder where standard users have write privileges, we configure the ProcMon filters listed in Figure 1. It is important to note that standard users have write privileges to the system's `C:\Windows\Temp` and `C:\ProgramData` folders and may sometimes have excessive privileges to their subfolders due to permission inheritance. Therefore, these paths should also be considered when configuring the filters.

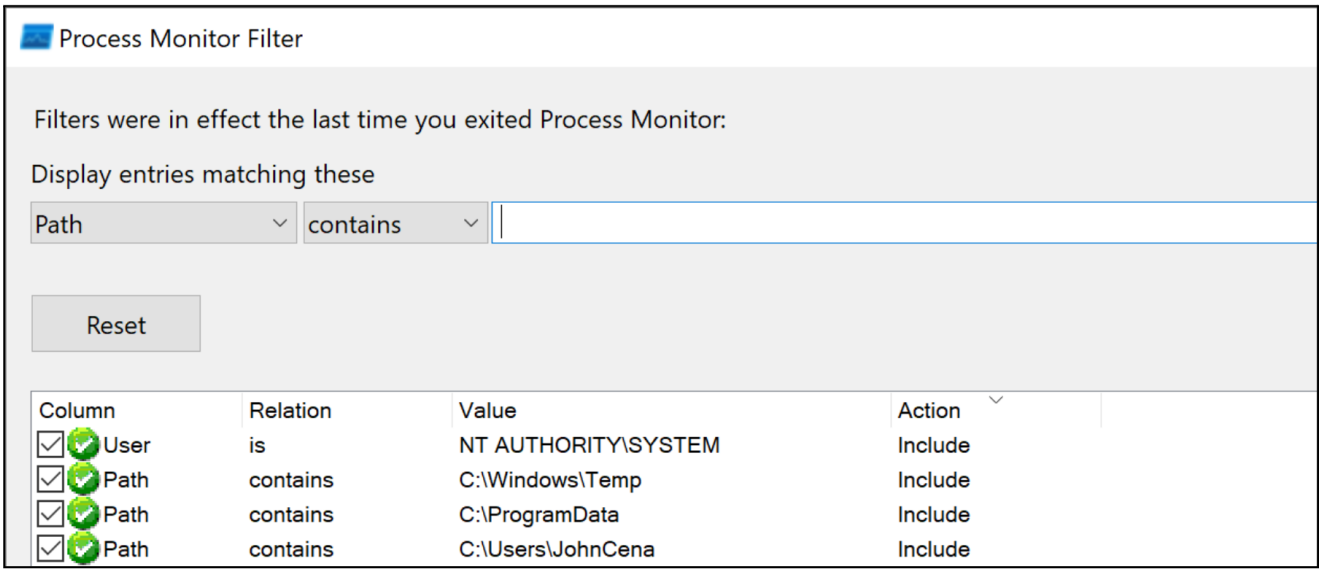


Figure 1: ProcMon filters to analyze insecure file operations from NT AUTHORITY\System
Once configured, MSI repairs can be initiated using the Windows Installer API or by running "`msiexec.exe /fa C:\Windows\Installer\[XXXXX].msi`".

Analyzing CVE-2023-26077

In this example, we will analyze the MSI installer for Atera Agent 1.8.3.6 using the aforementioned ProcMon filters. When running the MSI's repair functionality, we can see the `AgentPackageUpgradeAgent.exe` file is executed as `NT AUTHORITY\SYSTEM` from the `C:\Windows\Temp\AteraUpgradeAgentPackage` folder, as highlighted in Figure 2. Moreover, we can see the process attempts to load missing dynamic-link library (DLL) files from the same folder. Due to standard users having write permissions in `C:\Windows\Temp\AteraUpgradeAgentPackage`, which contains permissions inherited from `C:\Windows\Temp`, the MSI installer's repair functionality is susceptible to a local privilege escalation attack that can be exploited through DLL hijacking.

Figure 2: Process attempting to load missing DLLs

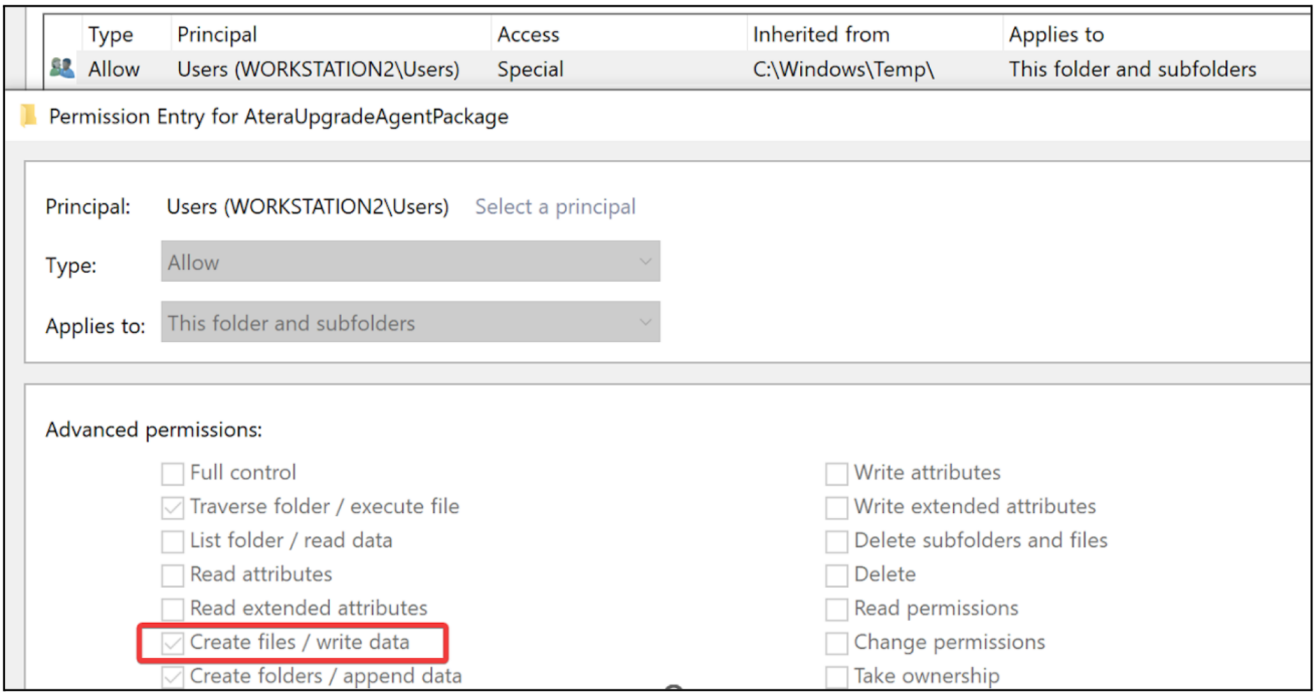


Figure 3: AteraUpgradeAgentPackage folder permissions

To exploit the vulnerability, we simply drop a payload as one of the missing DLLs into `C:\Windows\Temp\AteraUpgradeAgentPackage`, such as `CRYPTSP.dll`, and then run the MSI repair to obtain a Command Prompt as `NT AUTHORITY\SYSTEM`.

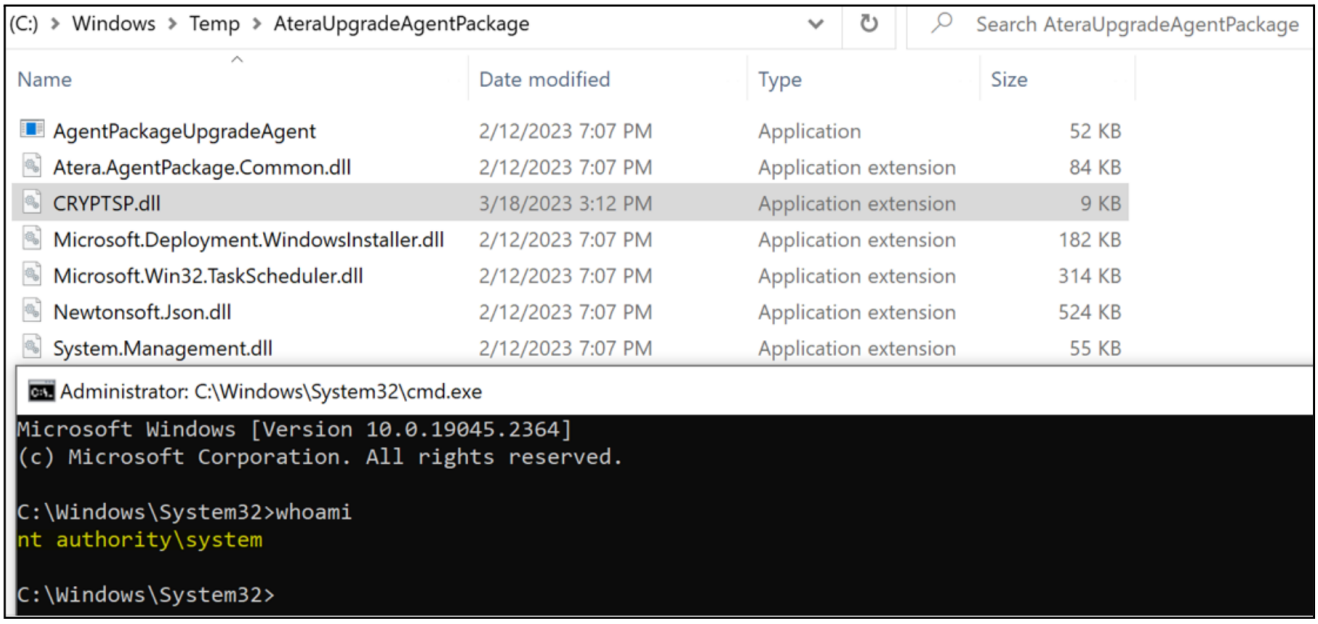


Figure 4: Local privilege escalation via DLL hijacking

Analyzing CVE-2023-26078

Another noteworthy misconfiguration involving Custom Actions is the execution of system commands that trigger the Windows Console Host (`conhost.exe`) as a child process. The `conhost.exe` process is responsible for hosting and managing console windows in the Windows operating systems. When a process runs with `conhost.exe` as its child process, it opens a command window, which, if executed with elevated privileges, can be exploited by an attacker to perform a local privilege escalation attack.

In the case of Atera's Windows installer, the repair functionality runs `net.exe` and `taskkill.exe` as `NT AUTHORITY\SYSTEM`, both of which spawn `conhost.exe` as a child process. This action causes a command window to briefly appear, which can be frozen by quickly selecting a portion of the window with the mouse, as depicted in Figure 6.

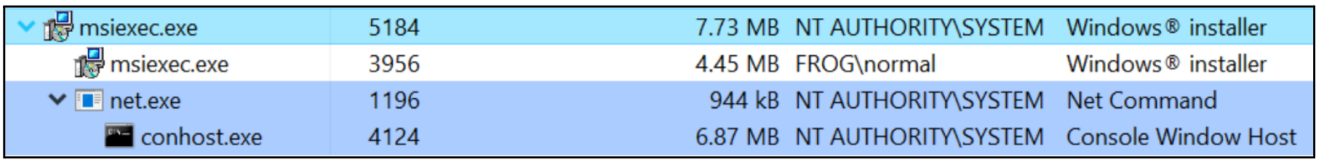


Figure 5: net.exe parent-child process relationship

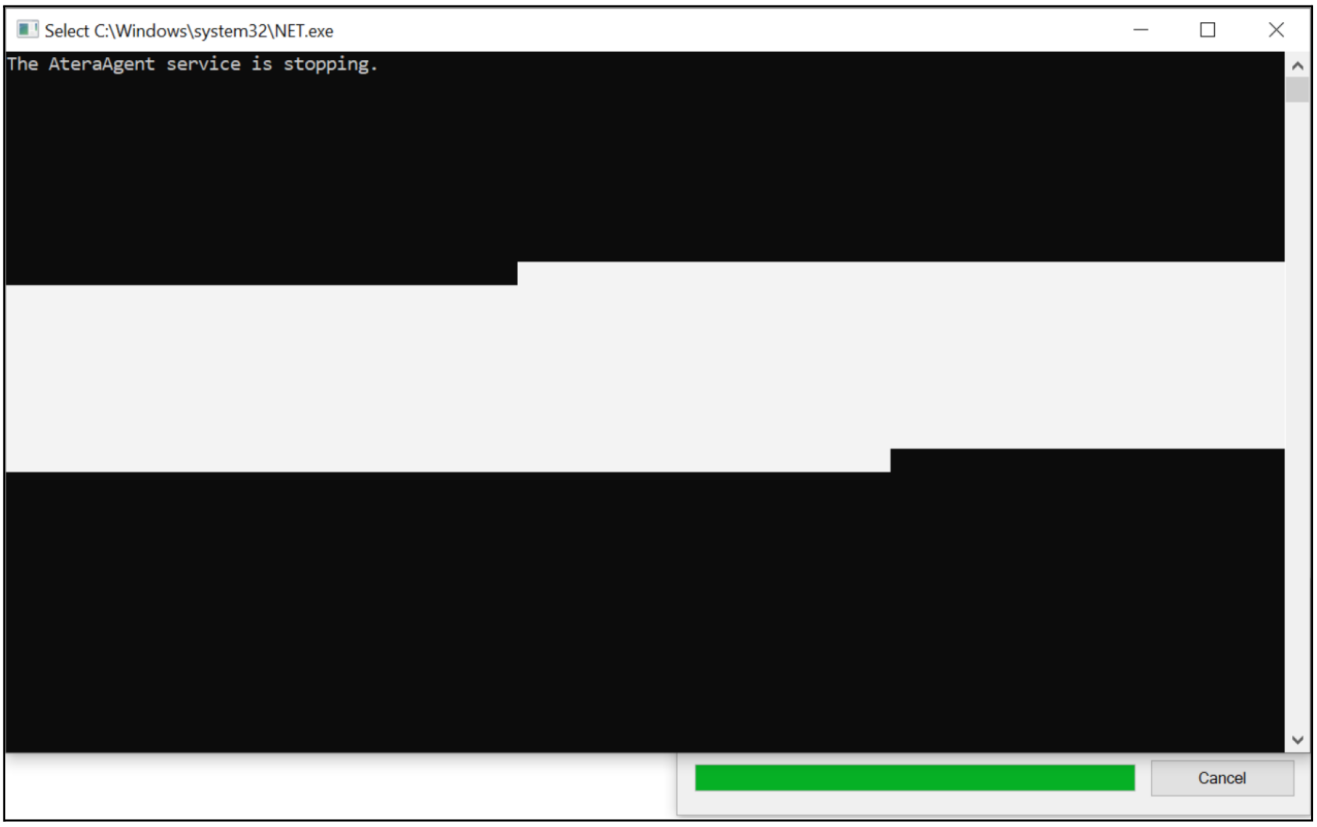


Figure 6: Pausing net.exe execution by highlighting part of the command window with mouse

By accessing the command window's "Properties", an attacker can access a couple of hyperlinks that can be utilized to open a web browser as `NT AUTHORITY\SYSTEM`. It is important to note that the `NT AUTHORITY\SYSTEM` account does not have a web browser configured to open hyperlinks by default. Therefore, an attacker is provided with the opportunity to select a web browser, as illustrated in Figure 7. Once the web browser is chosen, the remaining steps for the local privilege escalation attack involve opening the Print menu; printing the web page to PDF to launch File Explorer; and finally executing Command Prompt as `NT AUTHORITY\SYSTEM`.

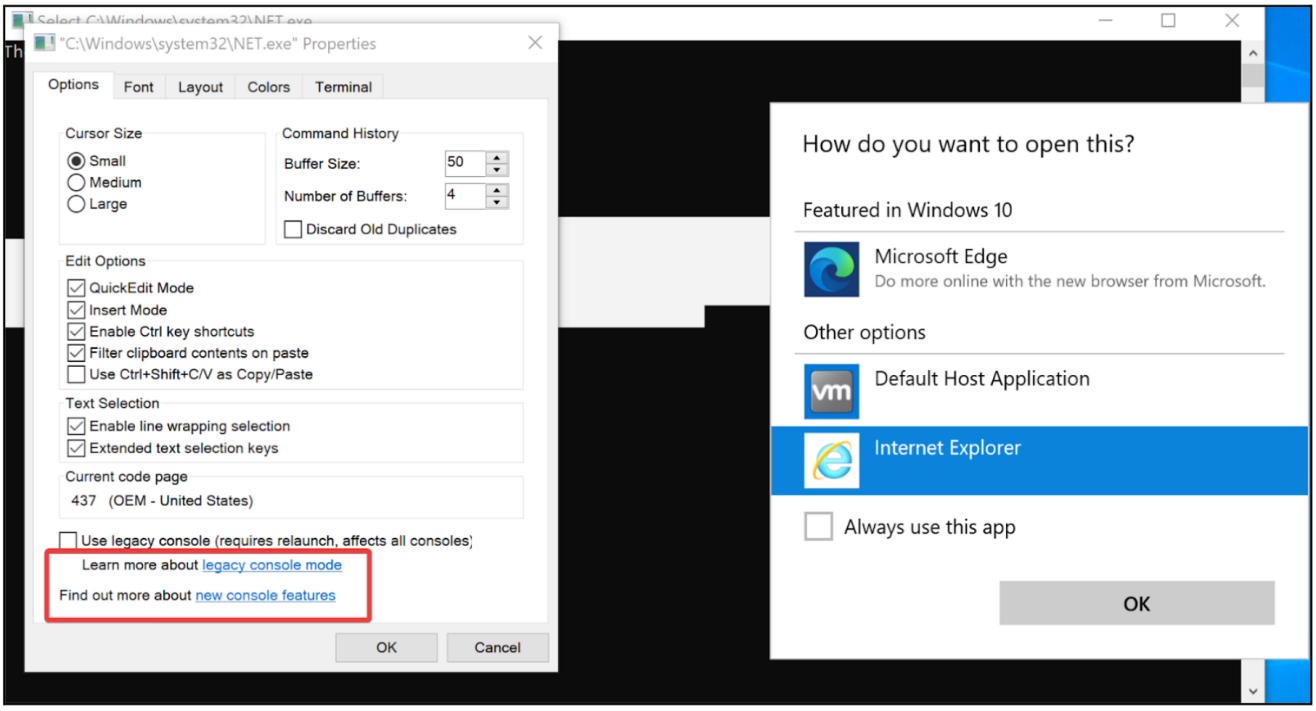


Figure 7: Opening hyperlinks with web browser of choice

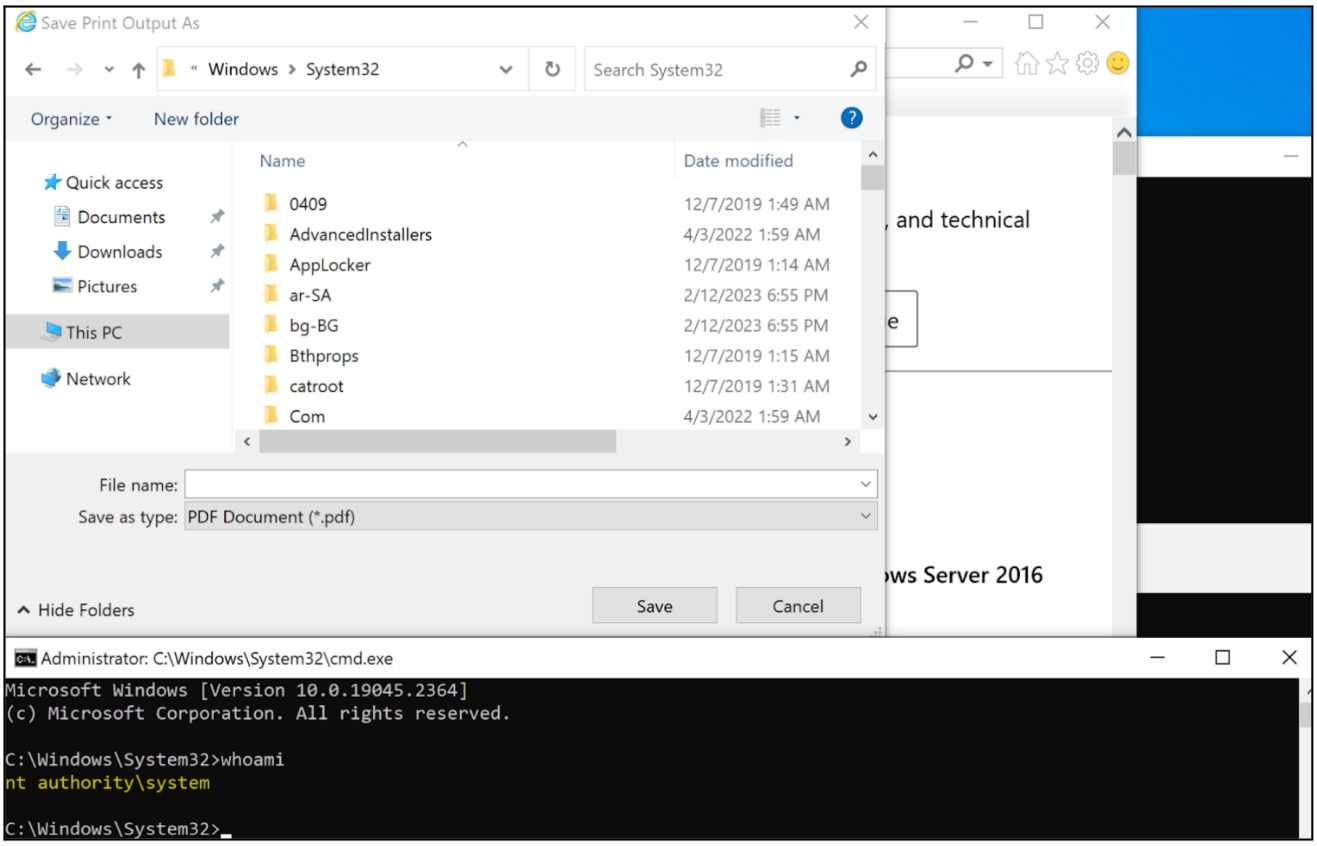


Figure 8: Opening Command Prompt through web browser

Programmatically Searching for MSI Files

Mandiant's red team have discovered numerous similar vulnerabilities in third-party Windows installers, which were subsequently leveraged during red team assessments. However, identifying which MSI files correspond to which software can be a tedious task since Windows caches MSI files with random alphanumeric characters. To simplify this task, Mandiant's red team created a Beacon Object File (BOF) and a PowerShell script found in **msi_search** to read and output relevant metadata for all MSI files cached in `C:\Windows\Installer`. Using this tool will allow red team operators and security teams to download relevant files to investigate local privilege escalation vulnerabilities through MSI repairs.

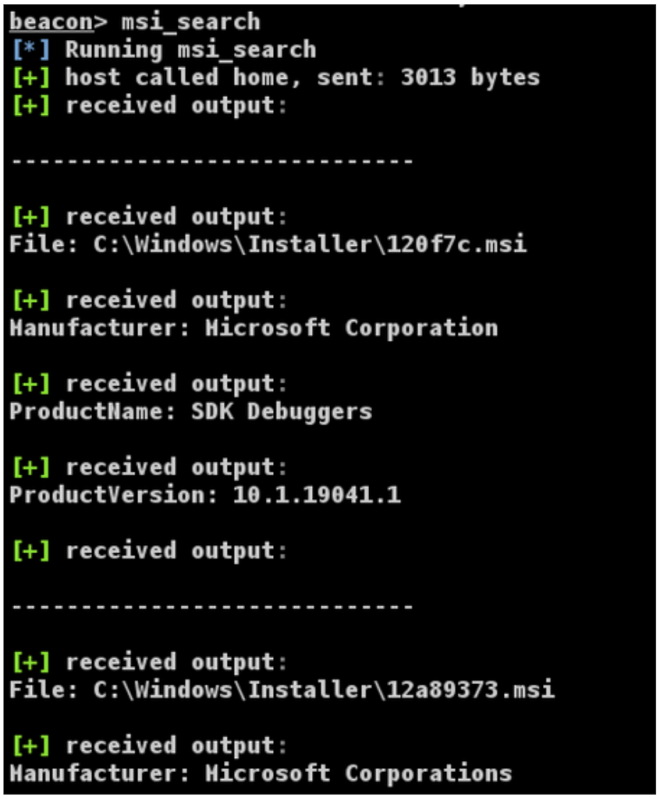


Figure 9: Excerpt of msi_search BOF

Defensive Considerations

Misconfigured Custom Actions can be trivial to identify and exploit, thereby posing significant security risks for organizations. It is essential for software developers to thoroughly review their Custom Actions to prevent attackers from hijacking `NT AUTHORITY\SYSTEM` operations triggered by MSI repairs.

When configuring Custom Actions, it is important to remember that on Windows standard users have write permissions to the following folders and may also have write permissions to their subfolders due to permission inheritance:

- `C:\Windows\Temp`
- `C:\ProgramData`
- `C:\` (ability to create folders)

- `C:\Users\XXXX\*`

Hence, software developers must bear this in mind and ensure that any privileged process using these folders are appropriately secured. Alternatively, using the `C:\Program Files` or `C:\Program Files (x86)` folders could hinder an attacker's ability to hijack execution, as these folders are protected by default with administrative privileges.

Atera fixed CVE-2023-26077 in version 1.8.3.7 by hardening the `C:\Windows\Temp\AteraUpgradeAgent` folder to disallow standard users from writing files to it, therefore blocking an attacker's ability to perform a DLL hijacking attack.

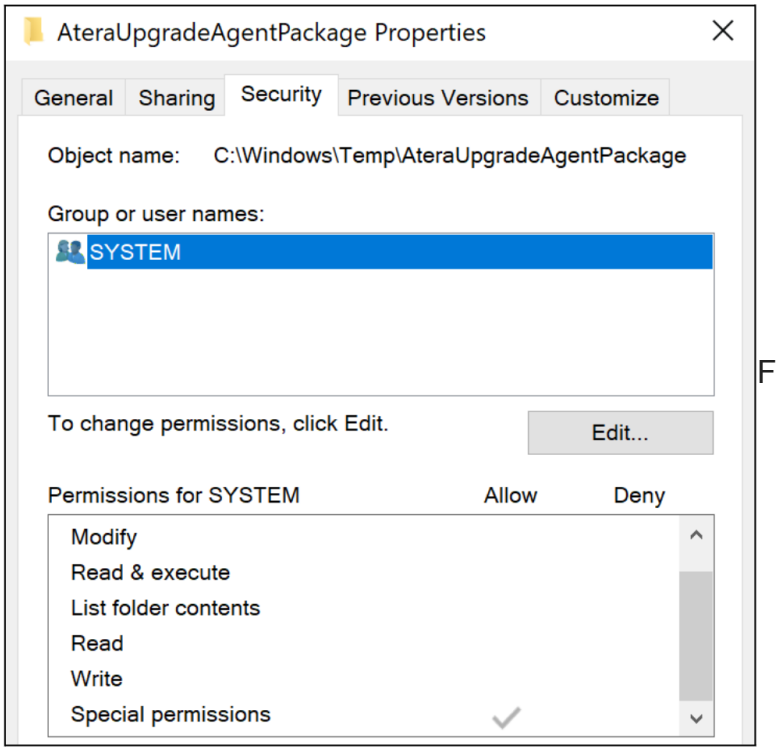


Figure 10: CVE-2023-26077 fix in Atera Agent 1.8.3.7

It is also important for software developers to exercise caution when using system commands that spawn `conhost.exe` as a child process. As demonstrated in CVE-2023-26078, executing `net.exe` as a Custom Action could temporarily open a command window, which could be exploited by attackers to elevate their privileges. To mitigate this potential security risk, software developers should use WixQuietExec to silently run system commands in the background. WixQuietExec allows commands to be executed without displaying a command window, significantly reducing the possibility of hijacking. Atera addressed CVE-2023-26078 in version 1.8.4.9.

To detect MSI repair privilege escalation attacks, incident responders and security operation teams can monitor the Application event ID 11728 originating from non-administrator users. This event ID is specifically associated with MSI repairs and provides valuable information such as the affected product, the user involved, and the date of the event.

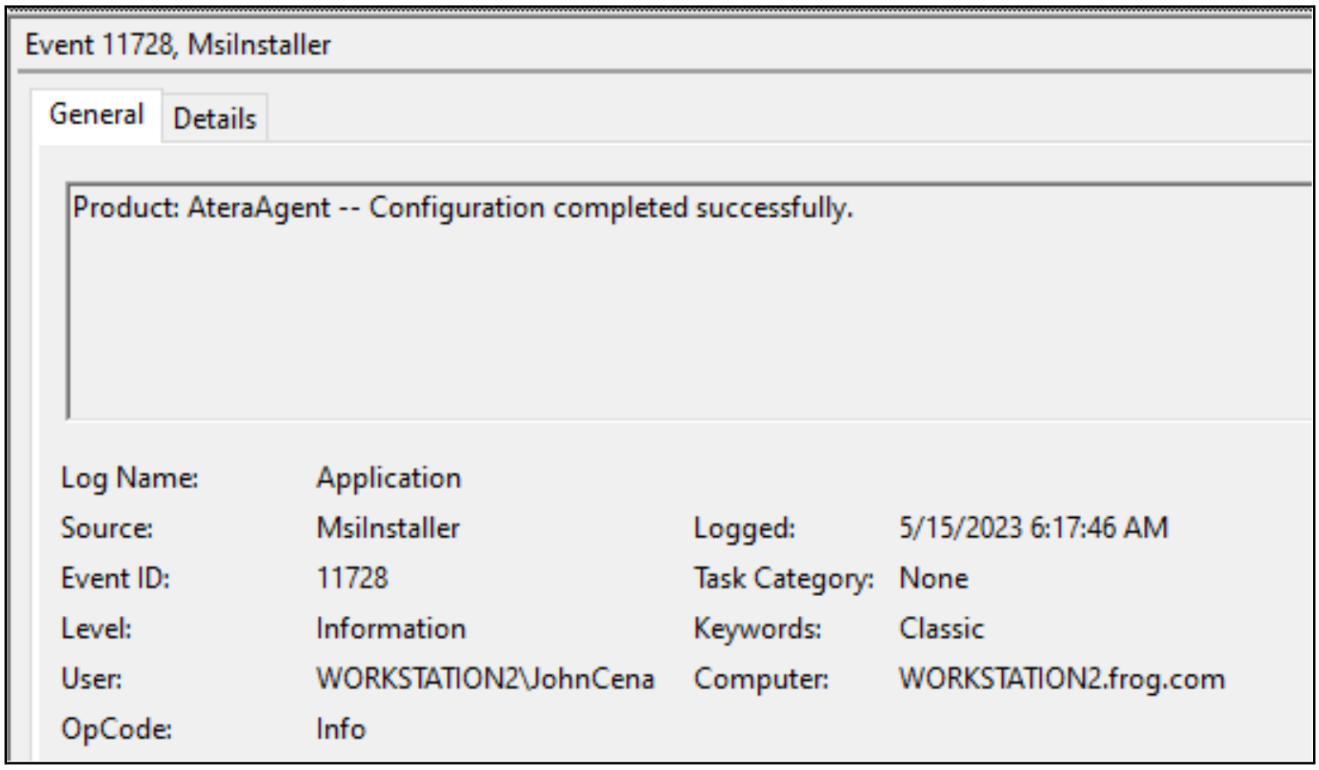


Figure 11: Event ID 11728

CVE-2023-26077 Disclosure Timeline

- **February 28, 2023** – Vulnerability reported to Atera
- **March 29, 2023** – Vulnerability confirmed by Atera
- **April 17, 2023** – Vulnerability fixed in version 1.8.3.7

CVE-2023-26078 Disclosure Timeline

- **February 28, 2023** – Vulnerability reported to Atera
- **March 29, 2023** – Vulnerability confirmed by Atera
- **June 26, 2023** – Vulnerability fixed in version 1.8.4.9