HUNT EVIL

You Can't Protect What You Don't Know About

Conducted by Amir Hossein Sharifi Sadr

Version 1

https://t.me/learning_net

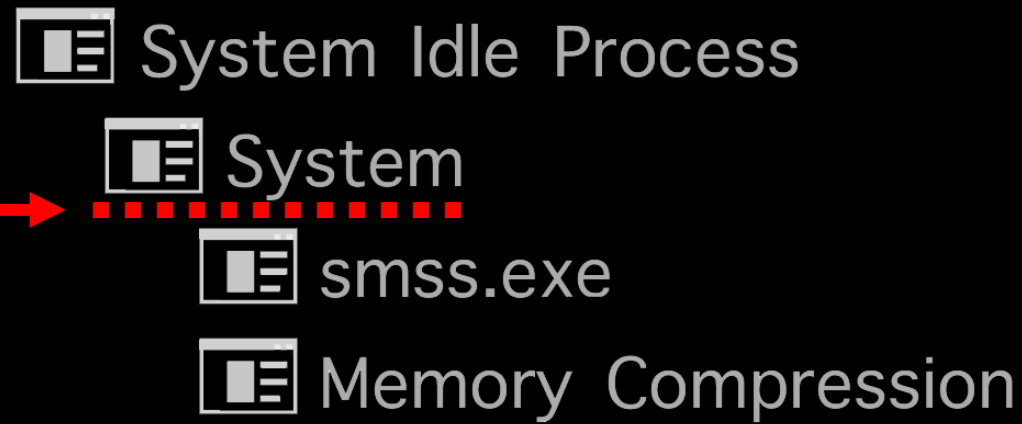
Find Evil Know Normal

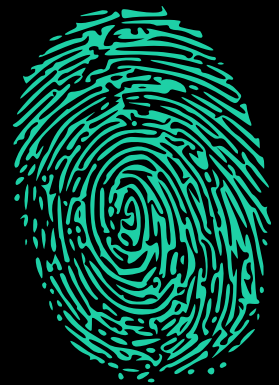
Knowing what's normal on a Windows host helps cut through the noise to quickly locate potential malware. Use the information below as a reference to know what's normal in Windows and to focus your attention on the outliers

Process listing from Windows 10 Enterprise

System

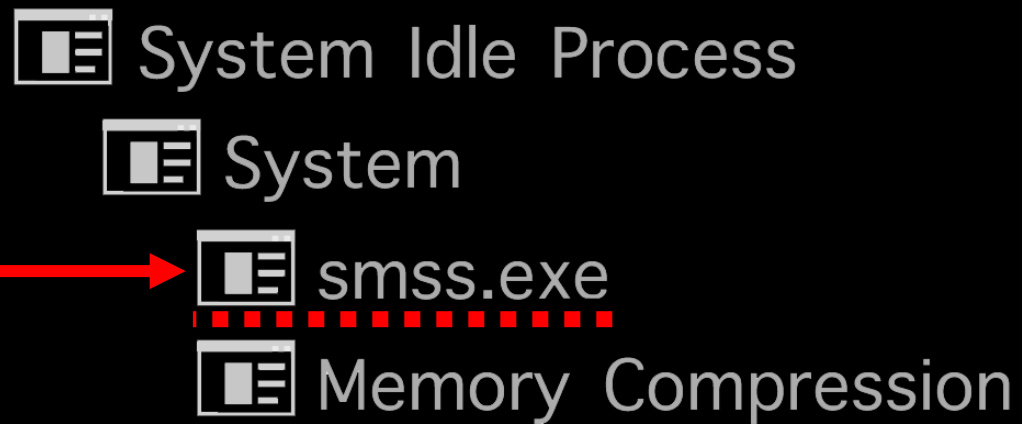
Image Path: N/A for **system.exe** – Not generated from an executable image
Parent Process: None
Number of Instances: One
User Account: Local System
Start Time: At boot time
Description: The **System** process is responsible for most kernel-mode threads. Modules run under **System** are primarily drivers (.sys files), but also include several important DLLs as well as the kernel executable, **ntoskrnl.exe**.





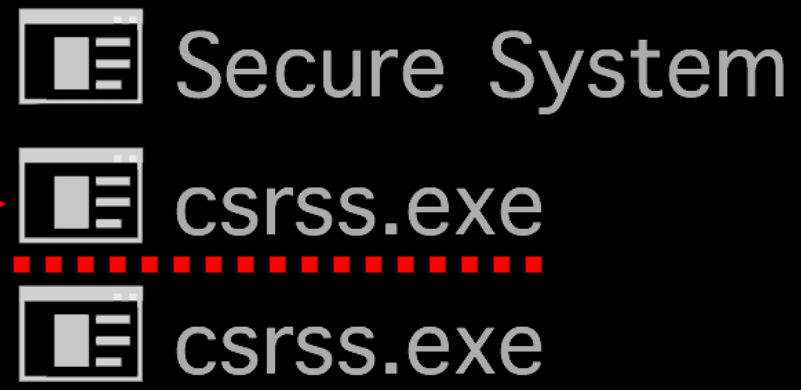
smss.exe

Image Path: **%SystemRoot%\System32\smss.exe**
Parent Process: System
Number of Instances: One master instance and another child instance per session. Children exit after creating their session.
User Account: Local System
Start Time: Within seconds of boot time for the master instance
Description: The Session Manager process is responsible for creating new sessions. The first instance creates a child instance for each new session.Once the child instance initializes the new session by starting the Windows subsystem (**csrss.exe**) and **wininit.exe** for Session 0 or **winlogon.exe** for Session 1 and higher, the child instance exits



csrss.exe

Image Path: **%SystemRoot%\System32\csrss.exe**
Parent Process: Created by an instance of **smss.exe** that exits, so analysis tools usually do not provide the parent process name.
Number of Instances: Two or more
User Account: Local System
Start Time: Within seconds of boot time for the first two instances (for Session 0 and 1). Start times for additional instances occur as new sessions are created, although often only Sessions 0 and 1 are created.
Description: The Client/Server Run-Time Subsystem is the user-mode process for the Windows subsystem. Its duties include managing processes and threads, importing many of the DLLs that provide the Windows API, and facilitating shutdown of the GUI during system shutdown. An instance of **csrss.exe** will run for each session. Session 0 is for services and Session 1 for the local console session. Additional sessions are created through the use of Remote Desktop and/or Fast User Switching. Each new session results in a new instance of **csrss.exe**.

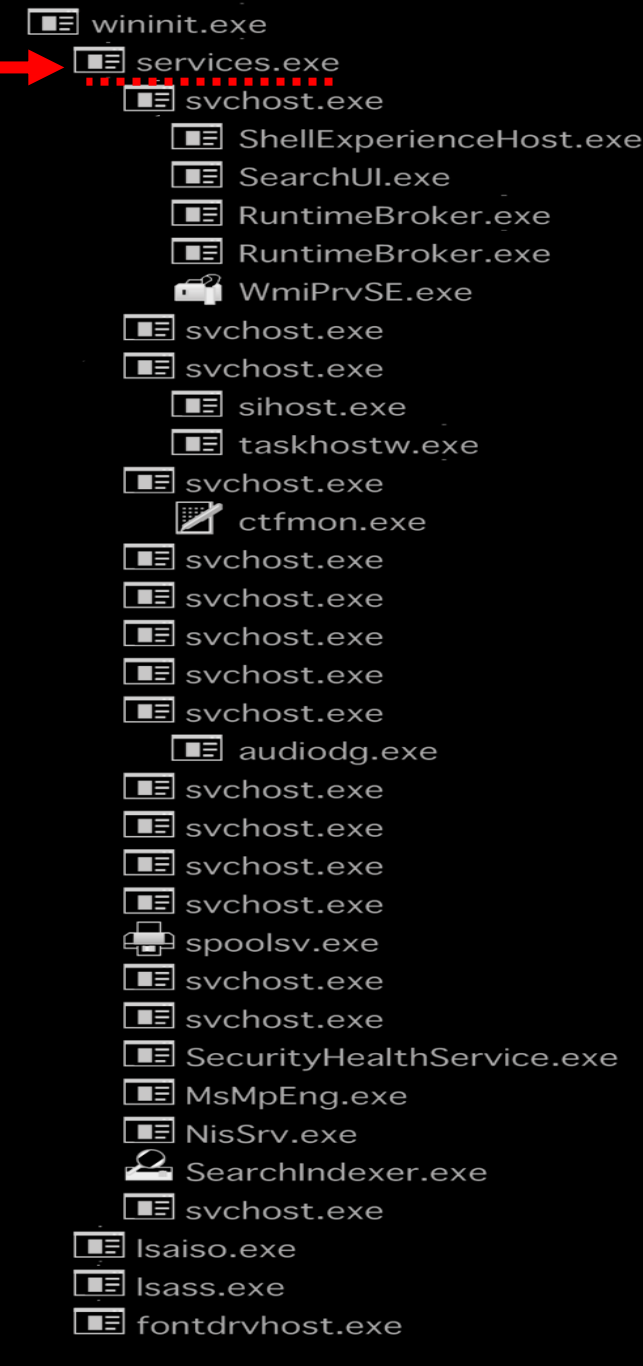


wininit.exe

Image Path: **%SystemRoot%\System32\wininit.exe**
Parent Process: Created by an instance of **smss.exe** that exits, so tools usually do not provide the parent process name.
Number of Instances: One
User Account: Local System
Start Time: Within seconds of boot time
Description: Wininit.exe starts key background processes within Session 0. It starts the Service Control Manager (**services.exe**), the Local Security Authority process (**lsass.exe**), and **lsaiso.exe** for systems with Credential Guard enabled. Note that prior to Windows 10, the Local Session Manager process (**lsm.exe**) was also started by wininit.exe. As of Windows 10, that functionality has moved to a service DLL (lsm.dll) hosted by **svchost.exe**

services.exe

Image Path: **%SystemRoot%\System32\services.exe**
Parent Process: **wininit.exe**
Number of Instances: One
User Account: Local System
Start Time: Within seconds of boot time
Description: Implements the Unified Background Process Manager (UBPM), which is responsible for background
activities such as services and scheduled tasks. **Services.exe** also implements the Service Control Manager (SCM),which specifically handles the loading of services and device drivers marked for auto-start. In addition, once a user has successfully logged on interactively, the SCM (**services.exe**) considers the boot successful and sets the Last Known Good control set (**HKLM\SYSTEM\Select\LastKnownGood**) to the value of the CurrentControlSet.



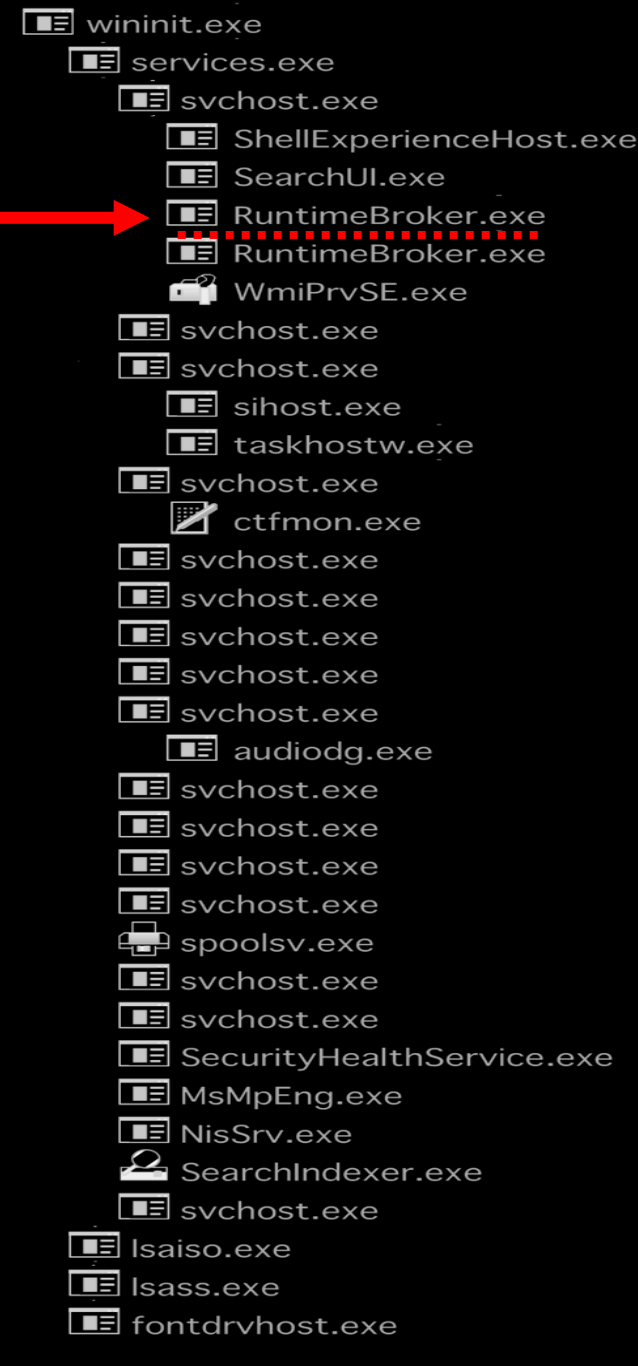
svchost.exe

Image Path: **%SystemRoot%\system32\svchost.exe**
Parent Process: **services.exe** (most often)
Number of Instances: Many (generally at least 10)
User Account: Varies depending on **svchost** instance, though it typically will be Local System, Network Service, or Local Service accounts. Windows 10 also has some instances running as logged-on users.
Start Time: Typically within seconds of boot time. However, services can be started after boot (e.g., at logon), which results in new instances of **svchost.exe** after boot time.
Description: Generic host process for Windows services. It is used for running service DLLs. Windows will run multiple instances of **svchost.exe**, each using a unique "**-k**" parameter for grouping similar services.Typical "**-k**" parameters include DcomLaunch, RPCSS, LocalServiceNetworkRestricted, LocalServiceNoNetwork,LocalServiceAndNoImpersonation, netsvcs, NetworkService, and more. Malware authors often take advantage of the ubiquitous nature of **svchost.exe** and use it either to host a malicious DLL as a service, or run a malicious process named **svchost.exe** or similar spelling. Beginning in Windows 10 version 1703, Microsoft changed the default grouping of similar services if the system has more than 3.5 GB of RAM. In such cases, most services will run under their own instance of **svchost.exe**. On systems with more than 3.5 GB RAM, expect to see more than 50 instances of **svchost.exe** (the screenshot in the poster is a Windows 10 VM with 3 GB RAM).

wininit.exe
services.exe
svchost.exe
ShellExperienceHost.exe
SearchUI.exe
RuntimeBroker.exe
RuntimeBroker.exe
WmiPrvSE.exe
svchost.exe
svchost.exe
sihost.exe
taskhostw.exe
svchost.exe
ctfmon.exe
svchost.exe
svchost.exe
svchost.exe
svchost.exe
svchost.exe
audiodg.exe
svchost.exe
svchost.exe
svchost.exe
svchost.exe
spoolsv.exe
svchost.exe
svchost.exe
SecurityHealthService.exe
MsMpEng.exe
NisSrv.exe
SearchIndexer.exe
svchost.exe
lsaiso.exe
lsass.exe
fontdrvhost.exe

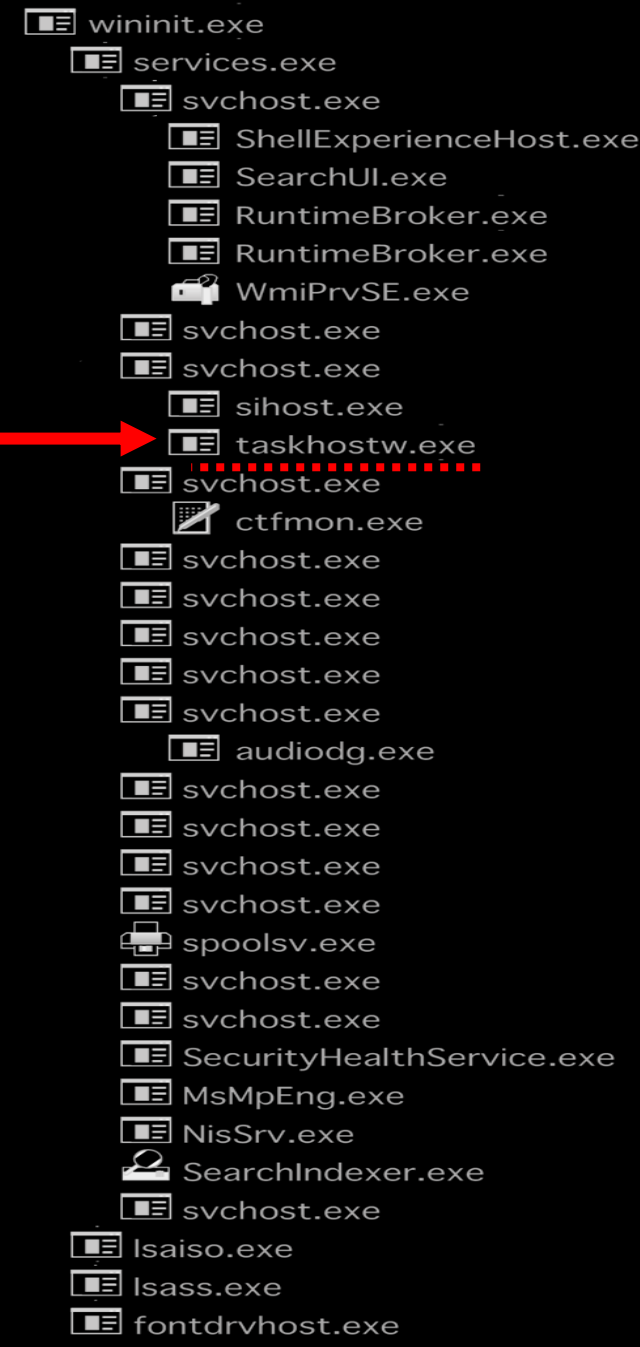
RuntimeBroker.exe

Image Path: **%SystemRoot%\System32\RuntimeBroker.exe**
Parent Process: **svchost.exe**
Number of Instances: One or more
User Account: Typically the logged-on user(s)
Start Time: Start times vary greatly
Description: RuntimeBroker.exe acts as a proxy between the constrained
Universal Windows Platform (UWP) apps (formerly called Metro apps) and the full Windows API. UWP apps have limited capability to interface with hardwareand the file system. Broker processes such as RuntimeBroker.exe are therefore used to provide the necessary level of access for UWP apps. Generally, there will be one **RuntimeBroker.exe** for each UWP app. For example, starting **Calculator.exe** will cause a corresponding **RuntimeBroker.exe** process to initiate



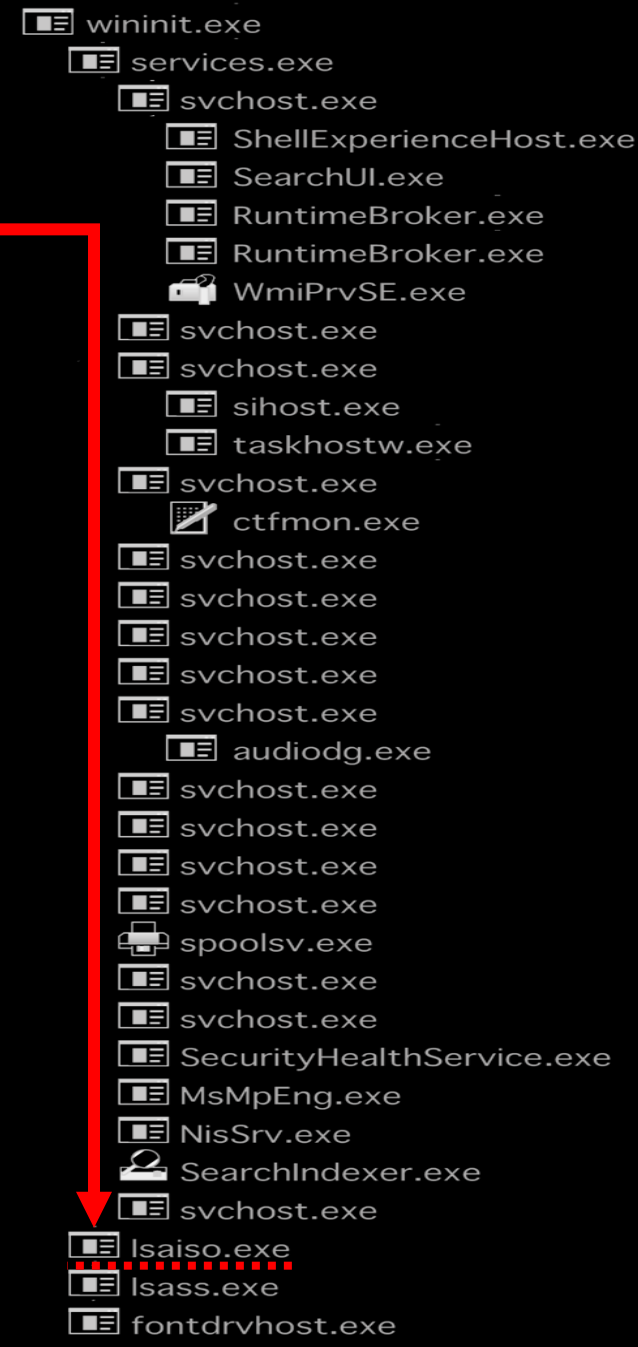
taskhostw.exe

Image Path: **%SystemRoot%\System32\taskhostw.exe**
Parent Process: **svchost.exe**
Number of Instances: One or more
User Account: Multiple taskhostw.exe processes are normal. One or more may be owned by logged-on users and/or by local service accounts.
Start Time: Start times vary greatly
Description: The generic host process for Windows Tasks. Upon initialization, taskhostw.exe runs a continuous loop listening for trigger events. Example trigger events that can initiate a task include a defined schedule, user logon, system startup, idle CPU time, a Windows log event, workstation lock, or workstation unlock.There are more than 160 tasks preconfigured on a default installation ofWindows 10 Enterprise (though many are disabled). All executable files (DLLs &EXEs) used by the default Windows 10 scheduled tasks are signed by Microsoft



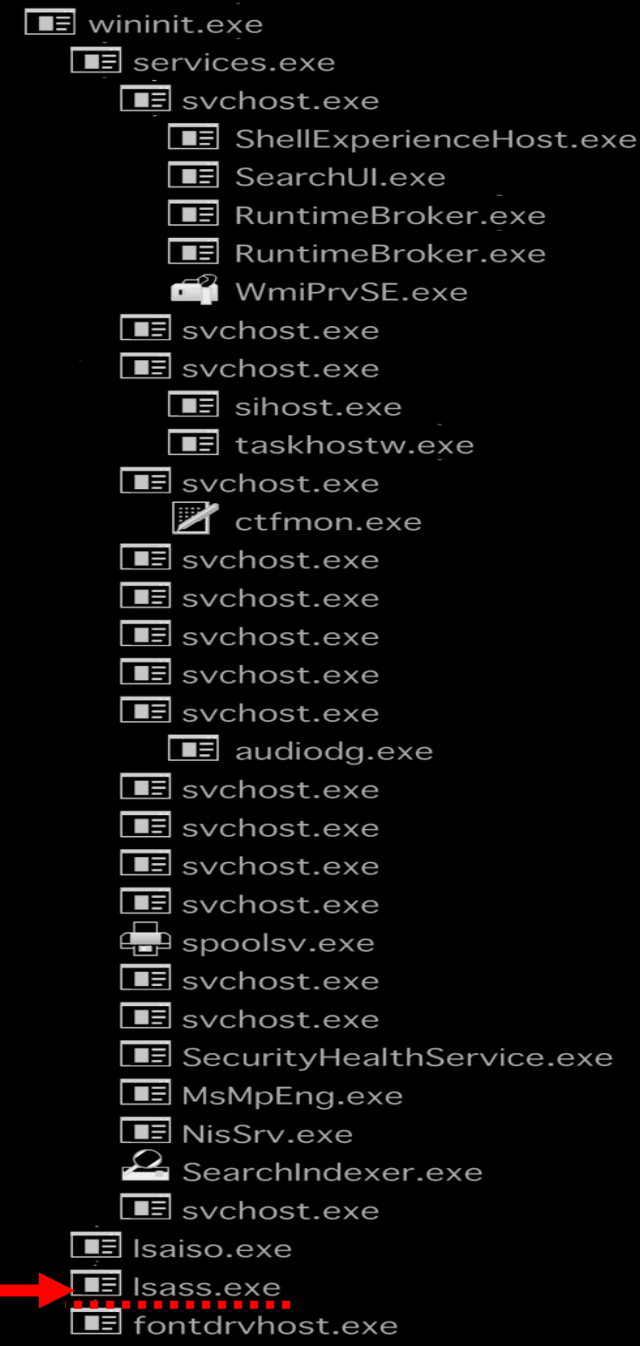
lsaiso.exe

Image Path: **%SystemRoot%\System32\lsaiso.exe**
Parent Process: **wininit.exe**
Number of Instances: Zero or one
User Account: Local System
Start Time: Within seconds of boot time
Description: When Credential Guard is enabled, the functionality of **lsass.exe** is split between two processes – itself and **lsaiso.exe**. Most of the functionality stays within **lsass.exe**, but the important role of safely storing account credentials moves to **lsaiso.exe**. It provides safe storage by running in a context that is isolated from other processes through hardware virtualization technology. When remote authentication is required, **lsass.exe** proxies the requests using an RPC channel with **lsaiso.exe** in order to authenticate the user to the remote service. Note that if Credential Guard is not enabled, **lsaiso.exe** should not be running on the system.



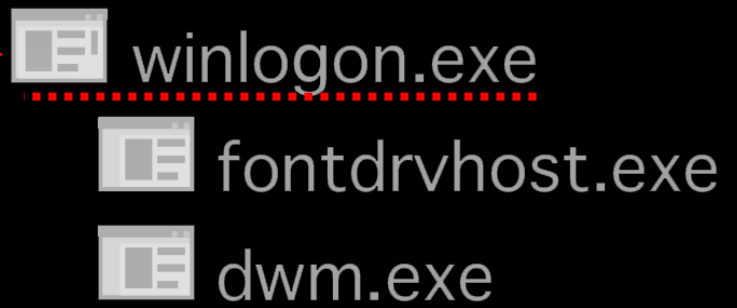
lsass.exe

Image Path: **%SystemRoot%\System32\lsass.exe**
Parent Process: **wininit.exe**
Number of Instances: One
User Account: Local System
Start Time: Within seconds of boot time
Description: The Local Security Authentication Subsystem Service process is responsible for authenticating users by calling an appropriate authentication package specified in **HKLM\SYSTEM\CurrentControlSet\Control\Lsa**. Typically, this will be Kerberos for domain accounts or MSV1_0 for local accounts. In addition to authenticating users, **lsass.exe** is also responsible for implementing the local security policy (such as password policies and audit policies) and for writing events to the security event log. Only one instance of this process should occur and it should rarely have child processes (EFS is a known exception)



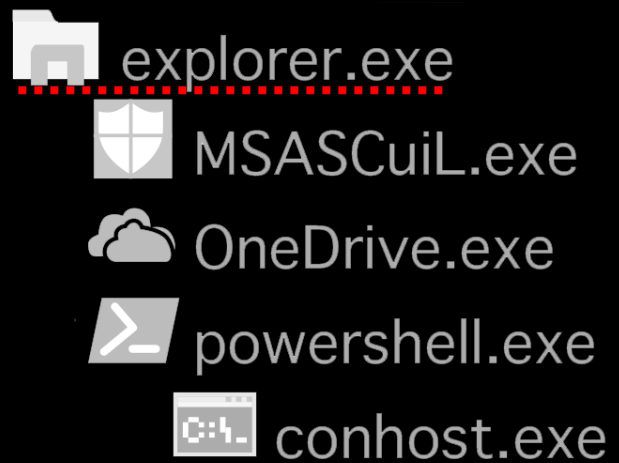
winlogon.exe

Image Path: **%SystemRoot%\System32\winlogon.exe**
Parent Process: Created by an instance of **smss.exe** that exits, so analysis tools usually do not provide the parent process name.
Number of Instances: One or more
User Account: Local System
Start Time: Within seconds of boot time for the first instance (for Session 1).Start times for additional instances occur as new sessions are created, typically through Remote Desktop or Fast User Switching logons.
Description: Winlogon handles interactive user logons and logoffs. It launches**LogonUI.exe**, which uses a credential provider to gather credentials fromZthe user, and then passes the credentials to **lsass.exe** for validation. Once the user is authenticated, Winlogon loads the user's **NTUSER.DAT** into **HKCU** and starts the user's shell (usually **explorer.exe**) via **userinit.exe**



explorer.exe

Image Path: **%SystemRoot%\explorer.exe**
Parent Process: Created by an instance of **userinit.exe** that exits, so analysis tools usually do not provide the parent process name.
Number of Instances: One or more per interactively logged-on user
User Account: <logged-on user(s)>
Start Time: First instance starts when the owner's interactive logon begins
Description: At its core, Explorer provides users access to files. Functionally, though, it is both a file browser via Windows Explorer (though still **explorer.exe**) and a user interface providing features such as the user's Desktop, the Start Menu, the Taskbar, the Control Panel, and application launching via file extension associations and shortcut files. **Explorer.exe** is the default user interface specified in the Registry value **HKLM\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\Winlogon\Shell**, though Windows can alternatively function with another interface such as **cmd.exe** or **powershell.exe**. Notice that the legitimate **explorer.exe** resides in the **%SystemRoot%** directory rather than **%SystemRoot%\System32**. Multiple instances per user can occur, such as when the option "Launch folder windows in a separate process" is enabled



Hunt Evil Lateral Movement

During incident response and threat hunting, it is critical to understand how attackers move around your network. Lateral movement is an inescapable requirement for attackers to stealthily move from system to system and accomplish their objectives. Every adversary, including the most skilled, will use some form of lateral movement technique described here during a breach. Understanding lateral movement tools and techniques allows responders to hunt more efficiently, quickly perform incident response scoping, and better anticipate future attacker activity. Tools and techniques to hunt the artifacts described below are detailed in the SANS DFIR course FOR508: Advanced Digital Forensics, Incident Response, and Threat Hunting

Lateral Movement

MITRE | ATT&CK®

The adversary is trying to move through your environment. Lateral Movement consists of techniques that adversaries use to enter and control remote systems on a network. Following through on their primary objective often requires exploring the network to find their target and subsequently gaining access to it. Reaching their objective often involves pivoting through multiple systems and accounts to gain. Adversaries might install their own remote access tools to accomplish Lateral Movement or use legitimate credentials with native network and operating system tools, which may be stealthier. ID: TA0008

ID	Name
T1021.001	Remote Desktop Protocol
T1021.002	SMB/Windows Admin Shares
T1021.003	Distributed Component Object Model
T1021.004	SSH
T1021.005	VNC
T1021.006	Windows Remote Management

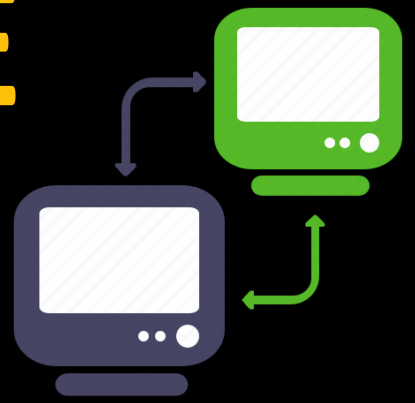
ID		Name	Description
T1210		Exploitation of Remote Services	Adversaries may exploit remote services to gain unauthorized access to internal systems once inside of a network. Exploitation of a software vulnerability occurs when an adversary takes advantage of a programming error in a program, service, or within the operating system software or kernel itself to execute adversary-controlled code. A common goal for post-compromise exploitation of remote services is for lateral movement to enable access to a remote system.
T1534		Internal Spearphishing	Adversaries may use internal spearphishing to gain access to additional information or exploit other users within the same organization after they already have access to accounts or systems within the environment. Internal spearphishing is multi-staged attack where an email account is owned either by controlling the user's device with previously installed malware or by compromising the account credentials of the user. Adversaries attempt to take advantage of a trusted internal account to increase the likelihood of tricking the target into falling for the phish attempt.
T1570		Lateral Tool Transfer	Adversaries may transfer tools or other files between systems in a compromised environment. Files may be copied from one system to another to stage adversary tools or other files over the course of an operation. Adversaries may copy files laterally between internal victim systems to support lateral movement using inherent file sharing protocols such as file sharing over SMB to connected network shares or with authenticated connections with SMB/Windows Admin Shares or Remote Desktop Protocol. Files can also be copied over on Mac and Linux with native tools like scp, rsync, and sftp.
T1563		Remote Service Session Hijacking	Adversaries may take control of preexisting sessions with remote services to move laterally in an environment. Users may use valid credentials to log into a service specifically designed to accept remote connections, such as telnet, SSH, and RDP. When a user logs into a service, a session will be established that will allow them to maintain a continuous interaction with that service.
T1563 .	001	SSH Hijacking	Adversaries may hijack a legitimate user's SSH session to move laterally within an environment. Secure Shell (SSH) is a standard means of remote access on Linux and macOS systems. It allows a user to connect to another system via an encrypted tunnel, commonly authenticating through a password, certificate or the use of an asymmetric encryption key pair.
	002	RDP Hijacking	Adversaries may hijack a legitimate user’s remote desktop session to move laterally within an environment. Remote desktop is a common feature in operating systems. It allows a user to log into an interactive session with a system desktop graphical user interface on a remote system. Microsoft refers to its implementation of the Remote Desktop Protocol (RDP) as Remote Desktop Services (RDS).
T1021		Remote Services	Adversaries may use Valid Accounts to log into a service specifically designed to accept remote connections, such as telnet, SSH, and VNC. The adversary may then perform actions as the logged-on user.
T1021.	00.1	Remote Desktop Protocol	Adversaries may use Valid Accounts to log into a computer using the Remote Desktop Protocol (RDP). The adversary may then perform actions as the logged-on user.
	00.2	SMB/Windows Admin Shares	Adversaries may use Valid Accounts to interact with a remote network share using Server Message Block (SMB). The adversary may then perform actions as the logged-on user.
	00.3	Distributed Component Object Model	Adversaries may use Valid Accounts to interact with remote machines by taking advantage of Distributed Component Object Model (DCOM). The adversary may then perform actions as the logged-on user.
	00.4	SSH	Adversaries may use Valid Accounts to log into remote machines using Secure Shell (SSH). The adversary may then perform actions as the logged-on user.
	00.5	VNC	Adversaries may use Valid Accounts to remotely control machines using Virtual Network Computing (VNC). The adversary may then perform actions as the logged-on user.
	00.6	Windows Remote Management	Adversaries may use Valid Accounts to interact with remote systems using Windows Remote Management (WinRM). The adversary may then perform actions as the logged-on user.

https://t.me/learninghets

ID		Name	Description
T1091		Replication Through Removable Media	Adversaries may move onto systems, possibly those on disconnected or air-gapped networks, by copying malware to removable media and taking advantage of Autorun features when the media is inserted into a system and executes. In the case of Lateral Movement, this may occur through modification of executable files stored on removable media or by copying malware and renaming it to look like a legitimate file to trick users into executing it on a separate system. In the case of Initial Access, this may occur through manual manipulation of the media, modification of systems used to initially format the media, or modification to the media's firmware itself.
T1072		Software Deployment Tools	Adversaries may gain access to and use third-party software suites installed within an enterprise network, such as administration, monitoring, and deployment systems, to move laterally through the network. Third-party applications and software deployment systems may be in use in the network environment for administration purposes (e.g., SCCM, HBSS, Altiris, etc.).
T1080		Taint Shared Content	Adversaries may deliver payloads to remote systems by adding content to shared storage locations, such as network drives or internal code repositories. Content stored on network drives or in other shared locations may be tainted by adding malicious programs, scripts, or exploit code to otherwise valid files. Once a user opens the shared tainted content, the malicious portion can be executed to run the adversary's code on a remote system. Adversaries may use tainted shared content to move laterally.
T1550		Use Alternate Authentication Material	Adversaries may use alternate authentication material, such as password hashes, Kerberos tickets, and application access tokens, in order to move laterally within an environment and bypass normal system access controls.
T1550.	00.1	Application Access Token	Adversaries may use stolen application access tokens to bypass the typical authentication process and access restricted accounts, information, or services on remote systems. These tokens are typically stolen from users and used in lieu of login credentials.
	00.2	Pass the Hash	Adversaries may "pass the hash" using stolen password hashes to move laterally within an environment, bypassing normal system access controls. Pass the hash (PtH) is a method of authenticating as a user without having access to the user's cleartext password. This method bypasses standard authentication steps that require a cleartext password, moving directly into the portion of the authentication that uses the password hash.
	00.3	Pass the Ticket	Adversaries may "pass the ticket" using stolen Kerberos tickets to move laterally within an environment, bypassing normal system access controls. Pass the ticket (PtT) is a method of authenticating to a system using Kerberos tickets without having access to an account's password. Kerberos authentication can be used as the first step to lateral movement to a remote system.
	00.4	Web Session Cookie	Adversaries can use stolen session cookies to authenticate to web applications and services. This technique bypasses some multi-factor authentication protocols since the session is already authenticated.

REMOTE Access

REMOTE Desktop



Remote Services: Remote Desktop Protocol

MITRE | ATT&CK®

Remote Desktop Protocol

Adversaries may use Valid Accounts to log into a computer using the Remote Desktop Protocol (RDP). The adversary may then perform actions as the logged-on user.

Remote desktop is a common feature in operating systems. It allows a user to log into an interactive session with a system desktop graphical user interface on a remote system. Microsoft refers to its implementation of the Remote Desktop Protocol (RDP) as Remote Desktop Services (RDS).[1]

Adversaries may connect to a remote system over RDP/RDS to expand access if the service is enabled and allows access to accounts with known credentials. Adversaries will likely use Credential Access techniques to acquire credentials to use with RDP. Adversaries may also use RDP in conjunction with the Accessibility Features technique for Persistence.[2]

MITRE | ATT&CK®

Remote Desktop Protocol

ID: T1021.001
Sub-technique of: T1021
Tactic: Lateral Movement
Platforms: Windows
System Requirements: RDP service enabled, account in the Remote Desktop Users group
Permissions Required: Remote Desktop Users, User
Data Sources: Logon Session: Logon Session Creation, Network Traffic: Network Connection Creation, Network Traffic: Network Traffic Flow, Process: Process Creation
CAPEC ID: CAPEC-555
Contributors: Matthew Demaske, Adaptforward
Version: 1.0
Created: 11 February 2020
Last Modified: 25 February 2020

MITRE | ATT&CK®

Remote Desktop Protocol

Procedure Examples

ID	Name	Description
G0006	APT1	The APT1 group is known to have used RDP during operations.[3]
G0022	APT3	APT3 enables the Remote Desktop Protocol for persistence.[4] APT3 has also interacted with compromised systems to browse and copy files through RDP sessions.[5]
G0087	APT39	APT39 has been seen using RDP for lateral movement and persistence, in some cases employing the rdpwinst tool for mangement of multiple sessions.[6][7]
G0096	APT41	APT41 used RDP for lateral movement.[8][9]
G0001	Axiom	The Axiom group is known to have used RDP during operations.[10]
G0108	Blue Mockingbird	Blue Mockingbird has used Remote Desktop to log on to servers interactively and manually copy files to remote hosts.[11]
S0030	Carbanak	Carbanak enables concurrent Remote Desktop Protocol (RDP) sessions.[12]
G0114	Chimera	Chimera has used RDP to access targeted systems.[13]
G0080	Cobalt Group	Cobalt Group has used Remote Desktop Protocol to conduct lateral movement.[14]
S0154	Cobalt Strike	Cobalt Strike can start a VNC-based remote desktop server and tunnel the connection through the already established C2 channel.[15]
S0334	DarkComet	DarkComet can open an active screen of the victim’s machine and take control of the mouse and keyboard.[16]
G0074	Dragonfly 2.0	Dragonfly 2.0 moved laterally via RDP.[17][18]
G0051	FIN10	FIN10 has used RDP to move laterally to systems in the victim environment.[19]
G0037	FIN6	FIN6 used RDP to move laterally in victim networks.[20][21]
G0061	FIN8	FIN8 has used RDP for Lateral Movement.[22]
G0117	Fox Kitten	Fox Kitten has used RDP to log in and move laterally in the target environment.[23][24]
S0434	Imminent Monitor	Imminent Monitor has a module for performing remote desktop access.[25]
S0283	jRAT	jRAT can support RDP control.[26]
S0250	Koadic	Koadic can enable remote desktop on the victim's machine.[27]

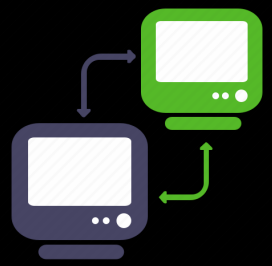
ID	Name	Description
G0032	Lazarus Group	Lazarus Group malware SierraCharlie uses RDP for propagation.[28][29]
G0065	Leviathan	Leviathan has targeted RDP credentials and used it to move through the victim environment.[30]
G0045	menuPass	menuPass has used RDP connections to move across the victim network.[31][32]
S0385	njRAT	njRAT has a module for performing remote desktop access.[33]
G0049	OilRig	OilRig has used Remote Desktop Protocol for lateral movement. The group has also used tunneling tools to tunnel RDP into the environment.[34][35][9]
G0040	Patchwork	Patchwork attempted to use RDP to move laterally.[36]
S0192	Pupy	Pupy can enable/disable RDP connection and can start a remote desktop session using a browser web socket client.[37]
S0583	Pysa	Pysa has laterally moved using RDP connections.[38]
S0262	QuasarRAT	QuasarRAT has a module for performing remote desktop access.[39][40]
S0379	Revenge RAT	Revenge RAT has a plugin to perform RDP access.[41]
S0461	SDBbot	SDBbot has the ability to use RDP to connect to victim's machines.[42]
S0382	ServHelper	ServHelper has commands for adding a remote desktop user and sending RDP traffic to the attacker through a reverse SSH tunnel.[43]
G0086	Stolen Pencil	Stolen Pencil utilized RDP for direct remote point-and-click access.[45]
G0088	TEMP.Veles	TEMP.Veles utilized RDP throughout an operation.[46]
G0102	Wizard Spider	Wizard Spider has used RDP for lateral movement.[47][48][49]
S0350	zwShell	zwShell has used RDP for lateral movement.[50]
S0412	ZxShell	ZxShell has remote desktop functionality.[51]

https://t.me/le

(SOURCE)

Microsoft-Windows–TerminalServices-RDPClient%4Operational.evtx
- **1024**
- Destination Host Name
- **1102**
- Destination IP Address

security.evtx
- **4648 – Logon specifying alternate credentials - if NLA enabled on destination**
- **Current logged-on User Name**
- **Alternate User Name**
- **Destination Host Name/IP**
- **Process Name**



REMOTE EVENT ACCESS LOGS

(DESTINATION)

Security Event Log –security.evtx
- **4624 Logon Type 10**
- **Source IP/Logon User Name**
- **4778/4779**
- **IP Address of Source/Source System Name**
- **Logon User Name**

Microsoft-Windows-TerminalServices-RemoteConnectio Manager%4Operational.evtx
- **1149**
- **Source IP/Logon User Name**
- **Blank user name may indicate use of Sticky Keys**

Microsoft-WindowsRemoteDesktopServicesRdpCoreTS%4Operational.evtx
- **131 – Connection Attempts**
- **Source IP**
- **98 – Successful Connections**

Microsoft-Windows-Terminal-ServicesLocalSessionManager%4Operational.evtx
- **21, 22, 25**
- **Source IP/Logon User Name**
- **41**
- **Logon User Name**

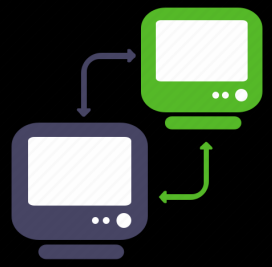
(SOURCE)

Remote desktop destinations are tracked per-user
NTUSER\Software\Microsoft\Terminal Server Client\Servers
ShimCache – SYSTEM
mstsc.exe Remote
Desktop Client
BAM/DAM – SYSTEM – Last
Time Executed
mstsc.exe Remote
Desktop Client
AmCache.hve – First Time
Executed
mstsc.ex

UserAssist – NTUSER.DAT
mstsc.exe Remote
Desktop Client execution
Last Time Executed
Number of Times Executed
RecentApps – NTUSER.DAT
mstsc.exe Remote
Desktop Client execution
Last Time Executed
Number of Times Executed
RecentItems subkey tracks
connection destinations and times

(DESTINATION)

ShimCache – SYSTEM
rdpclip.exe
tstheme.exe
AmCache.hve – First Time Executed
rdpclip.exe
tstheme.exe

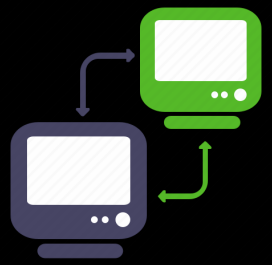


REMOTE ACCESS

REGISTRY

(SOURCE)

Jumplists – C:\Users\<Username>\AppData\Roaming\Microsft\Windows\Recent\AutomaticDestinations\
{MSTSC-APPID}-automaticDestinations-ms
Tracks remote desktop connection destination and times
Prefetch – C:\Windows\Prefetch\
mstsc.exe-{hash}.pf
Bitmap Cache – C:\USERS\<USERNAME>\AppData\Local\Microsoft\TerminalServer Client\Cache
bcache##.bmc
cache####.bin



REMOTE ACCESS

FILE SYSTEM

(DESTINATION)

Prefetch – C:\Windows\Prefetch\
rdpclip.exe-{hash}.pf
tstheme.exe-{hash}.pf

MITRE | ATT&CK® | Mitigations

Remote Desktop Protocol

ID	Mitigation	Description
M1047	Audit	Audit the Remote Desktop Users group membership regularly. Remove unnecessary accounts and groups from Remote Desktop Users groups.
M1042	Disable or Remove Feature or Program	Disable the RDP service if it is unnecessary.
M1035	Limit Access to Resource Over Network	Use remote desktop gateways.
M1032	Multi-factor Authentication	Use multi-factor authentication for remote logins.[52]
M1030	Network Segmentation	Do not leave RDP accessible from the internet. Enable firewall rules to block RDP traffic between network security zones within a network.
M1028	Operating System Configuration	Change GPOs to define shorter timeouts sessions and maximum amount of time any single session can be active. Change GPOs to specify the maximum amount of time that a disconnected session stays active on the RD session host server.[53]
M1026	Privileged Account Management	Consider removing the local Administrators group from the list of groups allowed to log in through RDP.
M1018	User Account Management	Limit remote user permissions if remote access is necessary.

Detection

Use of RDP may be legitimate, depending on the network environment and how it is used. Other factors, such as access patterns and activity that occurs after a remote login, may indicate suspicious or malicious behavior with RDP. Monitor for user accounts logged into systems they would not normally access or access patterns to multiple systems over a relatively short period of time.

Map Network Shares (net.exe) to C$ or Admin$

Remote Services: SMB/Windows Admin Shares

Adversaries may use Valid Accounts to interact with a remote network share using Server Message Block (SMB). The adversary may then perform actions as the logged-on user.

SMB is a file, printer, and serial port sharing protocol for Windows machines on the same network or domain. Adversaries may use SMB to interact with file shares, allowing them to move laterally throughout a network. Linux and macOS implementations of SMB typically use Samba. Windows systems have hidden network shares that are accessible only to administrators and provide the ability for remote file copy and other administrative functions. Example network shares include C$, ADMIN$, and IPC$. Adversaries may use this technique in conjunction with administrator-level Valid Accounts to remotely access a networked system over SMB,[1] to interact with systems using remote procedure calls (RPCs),[2] transfer files, and run transferred binaries through remote Execution. Example execution techniques that rely on authenticated sessions over SMB/RPC are Scheduled Task/Job, Service Execution, and Windows Management Instrumentation. Adversaries can also use NTLM hashes to access administrator shares on systems with Pass the Hash and certain configuration and patch levels.[3]

MITRE | ATT&CK®

SMB/Windows Admin Shares

ID: T1021.002
Sub-technique of: T1021
Tactic: Lateral Movement
Platforms: Windows
System Requirements: SMB enabled; Host/network firewalls not blocking SMB ports between source and destination; Use of domain account in administrator group on remote system or default system admin account.
Permissions Required: Administrator, User
Data Sources: Command: Command Execution, Logon Session: Logon Session Creation, Network Share: Network Share Access, Network Traffic: Network Connection Creation, Network Traffic: Network Traffic Flow
CAPEC ID: CAPEC-561
Version: 1.0
Created: 11 February 2020
Last Modified: 23 March 2020

MITRE | ATT&CK®

SMB/Windows Admin Shares

Procedure Examples

ID	Name	Description
S0504	Anchor	Anchor can support windows execution via SMB shares.[4]
G0022	APT3	APT3 will copy files over to Windows Admin Shares (like ADMIN$) as part of lateral movement.[5]
G0050	APT32	APT32 used Net to use Windows' hidden network shares to copy their tools to remote machines for execution.[6]
G0087	APT39	APT39 has used SMB for lateral movement.[7]
G0096	APT41	APT41 has transferred implant files using Windows Admin Shares.[8]
S0089	BlackEnergy	BlackEnergy has run a plug-in on a victim to spread through the local network by using PsExec and accessing admin shares.[9]
G0108	Blue Mockingbird	Blue Mockingbird has used Windows Explorer to manually copy malicious files to remote hosts over SMB.[10]
G0114	Chimera	Chimera has used Windows admin shares to move laterally.[11][12]
S0154	Cobalt Strike	Cobalt Strike can use Window admin shares (C$ and ADMIN$) for lateral movement.[13]
S0575	Conti	Conti can spread via SMB and encrypts files on different hosts, potentially compromising an entire network.[14][15]
G0009	Deep Panda	Deep Panda uses net.exe to connect to network shares using net use commands with compromised credentials.[16]
S0038	Duqu	Adversaries can instruct Duqu to spread laterally by copying itself to shares it has enumerated and for which it has obtained legitimate credentials (via keylogging or other means). The remote host is then infected by using the compromised credentials to schedule a task on remote machines that executes the malware.[17]
S0367	Emotet	Emotet leverages the Admin$ share for lateral movement once the local admin password has been brute forced. [18]
G0061	FIN8	FIN8 has attempted to map to C$ on enumerated hosts to test the scope of their current credentials/context.[19]
G0117	Fox Kitten	Fox Kitten has used valid accounts to access SMB shares.[20]
G0004	Ke3chang	Ke3chang actors have been known to copy files to the network shares of other computers to move laterally.[21][22]
S0236	Kwampirs	Kwampirs copies itself over network shares to move laterally [illegible] network.[23]

ID	Name	Description
G0032	Lazarus Group	Lazarus Group malware SierraAlfa accesses the ADMIN$ share via SMB to conduct lateral movement.[24][25]
S0532	Lucifer	Lucifer can infect victims by brute forcing SMB.[26]
S0039	Net	Lateral movement can be done with Net through net use commands to connect to the on remote systems.[27]
S0056	Net Crawler	Net Crawler uses Windows admin shares to establish authenticated sessions to remote systems over SMB as part of lateral movement.[28]
S0368	NotPetya	NotPetya can use PsExec, which interacts with the ADMIN$ network share to execute commands on remote systems.[29][30][31]
S0365	Olympic Destroyer	Olympic Destroyer uses PsExec to interact with the ADMIN$ network share to execute commands on remote systems.[32][31]
G0116	Operation Wocao	Operation Wocao has used Impacket's smbexec.py as well as accessing the C$ and IPC$ shares to move laterally.[33]
G0071	Orangeworm	Orangeworm has copied its backdoor across open network shares, including ADMIN$, C$WINDOWS, D$WINDOWS, and E$WINDOWS.[23]
S0029	PsExec	PsExec, a tool that has been used by adversaries, writes programs to the ADMIN$ network share to execute commands on remote systems.[31]
S0019	Regin	The Regin malware platform can use Windows admin shares to move laterally.[34]
S0446	Ryuk	Ryuk has used the C$ network share for lateral movement.[35]
S0140	Shamoon	Shamoon accesses network share(s), enables share access to the target device, copies an executable payload to the target system, and uses a Scheduled Task/Job to execute the malware.[36]
S0236	Kwampirs	Kwampirs copies itself over network shares to move laterally on a victim network.[23]
G0032	Lazarus Group	Lazarus Group malware SierraAlfa accesses the ADMIN$ share via SMB to conduct lateral movement.[24][25]
S0532	Lucifer	Lucifer can infect victims by brute forcing SMB.[26]
S0039	Net	Lateral movement can be done with Net through net use commands to connect to the on remote systems.[27]
S0056	Net Crawler	Net Crawler uses Windows admin shares to establish authenticated sessions to remote systems over SMB as part of lateral movement.[28]

https://t.me/learningnets

ID	Name	Description
S0368	NotPetya	NotPetya can use PsExec, which interacts with the ADMIN$ network share to execute commands on remote systems.[29][30][31]
S0365	Olympic Destroyer	Olympic Destroyer uses PsExec to interact with the ADMIN$ network share to execute commands on remote systems.[32][31]
G0116	Operation Wocao	Operation Wocao has used Impacket's smbexec.py as well as accessing the C$ and IPC$ shares to move laterally.[33]
G0071	Orangeworm	Orangeworm has copied its backdoor across open network shares, including ADMIN$, C$WINDOWS, D$WINDOWS, and E$WINDOWS.[23]
S0029	PsExec	PsExec, a tool that has been used by adversaries, writes programs to the ADMIN$ network share to execute commands on remote systems.[31]
S0019	Regin	The Regin malware platform can use Windows admin shares to move laterally.[34]
S0446	Ryuk	Ryuk has used the C$ network share for lateral movement.[35]
S0140	Shamoon	Shamoon accesses network share(s), enables share access to the target device, copies an executable payload to the target system, and uses a Scheduled Task/Job to execute the malware.[36]
G0028	Threat Group-1314	Threat Group-1314 actors mapped network drives using net use.[37]
G0010	Turla	Turla used net use commands to connect to lateral systems within a network.[38]
G0102	Wizard Spider	Wizard Spider has used SMB to drop Cobalt Strike Beacon on a domain controller for lateral movement.[39][40]
S0350	zwShell	zwShell has been copied over network shares to move laterally.[41]

(SOURCE)

security.evtx
- 4648 – Logon specifying
  - alternate credentials
  - Current logged-on User Name
  - Alternate User Name
  - Destination Host Name/IP
  - Process Name

Microsoft-WindowsSmbClient%4Security.evtx
- 31001 – Failed logon to destination
  - Destination Host Name
  - User Name for failed logon
  - Reason code for failed destination logon (e.g. bad password)

(DESTINATION)

Security Event Log –security.evtx

4624 Logon Type 3
- Source IP/Logon User Name

4672-
- Logon User Name
- Logon by user with administrative rights
- Requirement for accessing default shares such as C$ and ADMIN$

4776
- NTLM if authenticating to Local System
- Source Host Name/Logon User Name

4768 – TGT Granted
- Source Host Name/Logon UserName
- Available only on domain controller

4769 – Service Ticket Granted if
- authenticating to Domain Controller
- Destination Host Name/Logon UserName
- Source IP
- Available only on domain controller

5140
- Share Access

5145
- Auditing of shared files – NOISY!

Map Network Shares (net.exe) to C$ or Admin$

EVENT LOGS

(SOURCE)

MountPoints2 – Remotely mapped shares
NTUSER\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2
Shellbags – USRCLASS.DAT
Remote folders accessed inside an interactive session via Explorer by attackers
ShimCache – SYSTEM
net.exe
net1.exe
BAM/DAM – NTUSER.DAT – Last Time Executed
net.exe
net1.exe
AmCache.hve – First Time Executed
net.exe
net1.exe

(DESTINATION)

Empty

Map Network Shares (net.exe) to C$ or Admin$

REGISTRY

(SOURCE)

Prefetch – C:\Windows\Prefetch\
net.exe-{hash}.pf
net1.exe-{hash}.pf
User Profile Artifacts
Review shortcut files and jumplists for remote files accessed by attackers, if they had interactive access (RDP)

```
net use z: \\host\c$ /user:domain\username <password>
```

Map Network Shares (net.exe) to C$ or Admin$

FILE SYSTEM

(DESTINATION)

File Creation
Attacker's files (malware) copied to destination system
Look for Modified Time beforeCreation Time
Creation Time is time of file copy

MITRE | ATT&CK® | Mitigations

SMB/Windows Admin Shares

ID	Mitigation	Description
M1037	Filter Network Traffic	Consider using the host firewall to restrict file sharing communications such as SMB. [42]
M1035	Limit Access to Resource Over Network	Consider disabling Windows administrative shares.
M1027	Password Policies	Do not reuse local administrator account passwords across systems. Ensure password complexity and uniqueness such that the passwords cannot be cracked or guessed.
M1026	Privileged Account Management	Deny remote use of local admin credentials to log into systems. Do not allow domain user accounts to be in the local Administrators group multiple systems.

Detection

Ensure that proper logging of accounts used to log into systems is turned on and centrally collected. Windows logging is able to collect success/failure for accounts that may be used to move laterally and can be collected using tools such as Windows Event Forwarding. [43][44] Monitor remote login events and associated SMB activity for file transfers and remote process execution. Monitor the actions of remote users who connect to administrative shares. Monitor for use of tools and commands to connect to remote shares, such as Net, on the command-line interface and Discovery techniques that could be used to find remotely accessible systems.[45]

REMOTE EXECUTION

PsExec

MITRE | ATT&CK®

System Services: Service Execution PsExec

PsExec is a free Microsoft tool that can be used to execute a program on another computer. It is used by IT administrators and attackers. [1] [2]

ID: S0029
Type: TOOL
Platforms: Windows
Version: 1.1
Created: 31 May 2017
Last Modified: 20 March 2020

MITRE | ATT&CK®

System Services: Service Execution | PsExec

Procedure Examples

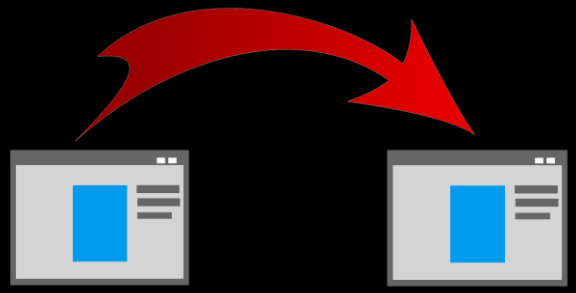
MITRE | ATT&CK®

System Services: Service Execution | PsExec

Domain	ID		Name	Use
Enterprise	**T1570**		**Lateral Tool Transfer**	**PsExec can be used to download or upload a file over a network share.[3]**
Enterprise	T1021	.002	Remote Services: SMB/Windows Admin Shares	PsExec, a tool that has been used by adversaries, writes programs to the ADMIN$ network share to execute commands on remote systems.[3]
Enterprise	T1569	.002	System Services: Service Execution	Microsoft Sysinternals PsExec is a popular administration tool that can be used to execute binaries on remote systems using a temporary Windows service.[1]

(SOURCE)

ecurity.evtx
4648 – Logon specifying alternate credentials
Current logged-on User Name
Alternate User Name
Destination Host Name/IP
Process Name



EVENT LOGS

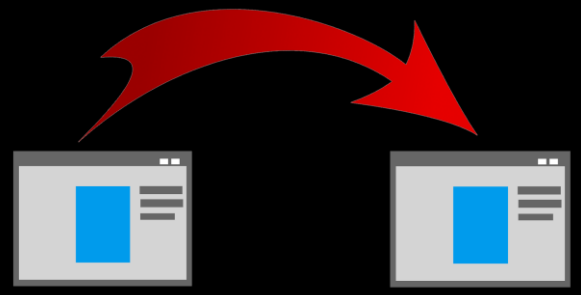
PsExec

(DESTINATION)

security.evtx
4648 Logon specifying alternate credentials
Connecting User Name
Process Name
4624 Logon Type 3 (and Type 2 if “-u” Alternate Credentials are used)
Source IP/Logon User Name
4672
Logon User Name
Logon by a user with administrative rights
Requirement for access default shares such as C$ and ADMIN$
5140 – Share Access
ADMIN$ share used by PsExec
system.evtx
7045
Service Instal

(SOURCE)

NTUSER.DAT
Software\SysInternals\PsExec\EulaAccepted
ShimCache – SYSTEM
psexec.exe
BAM/DAM – SYSTEM – Last Time Executed
psexec.exe
AmCache.hve – First Time Executed
psexec.exe



REGISTRY

PsExec

(DESTINATION)

New service creation
configured in SYSTEM\CurrentControlSet\Services\PSEXESVC
“-r” option can allow
attacker to rename service
ShimCache – SYSTEM
psexesvc.exe
AmCache.hve
First Time Executed
psexesvc.exe

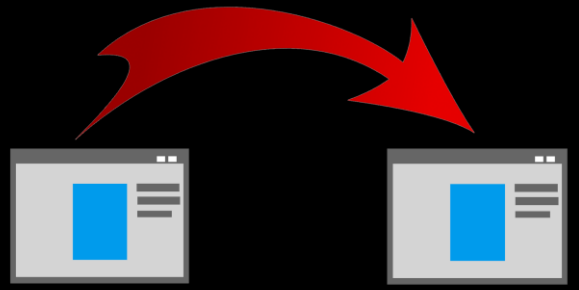
(SOURCE)

Prefetch – C:\Windows\Prefetch\
psexec.exe-{hash}.pf
Possible references to other files accessed by psexec.exe, such as executables copied to target system with the “-c” option

File Creation
psexec.exe file downloaded and created on local host as the file is not native to Windows

```
psexec.exe \\host -accepteula -d -c c:\temp\evil.exe
```

FILE SYSTEM



PsExec

(DESTINATION)

Prefetch – C:\Windows\Prefetch\
psexesvc.exe-{hash}.pf
evil.exe-{hash}.pf

File Creation
User profile directory structure created unless “-e” option used

psexesvc.exe will be placed in ADMIN$ (\Windows) by default, as well as other executables (evil.exe) pushed by PsExec

Scheduled Tasks

MITRE | ATT&CK®

Adversaries may abuse task scheduling functionality to facilitate initial or recurring execution of malicious code. Utilities exist within all major operating systems to schedule programs or scripts to be executed at a specified date and time. A task can also be scheduled on a remote system, provided the proper authentication is met (ex: RPC and file and printer sharing in Windows environments). Scheduling a task on a remote system typically requires being a member of an admin or otherwise privileged group on the remote system.[1]

Adversaries may use task scheduling to execute programs at system startup or on a scheduled basis for persistence. These mechanisms can also be abused to run a process under the context of a specified account (such as one with elevated permissions/privileges).

MITRE | ATT&CK®

ID	Name
T1053.001	At (Linux)
T1053.002	At (Windows)
T1053.003	Cron
T1053.004	Launchd
T1053.005	Scheduled Task
T1053.006	Systemd Timers
T1053.007	Container Orchestration Job

ID: T1053
Sub-techniques: T1053.001, T1053.002, T1053.003, T1053.004, T1053.005, T1053.006, T1053.007
Tactics: Execution, Persistence, Privilege Escalation
Platforms: Containers, Linux, Windows, macOS
Permissions Required: Administrator, SYSTEM, User
Effective Permissions: Administrator, SYSTEM, User
Data Sources: Command: Command Execution, Container: Container Creation, File: File Creation, File: File Modification, Process: Process Creation, Scheduled Job: Scheduled Job Creation
Supports Remote: Yes
CAPEC ID: CAPEC-557
Contributors: Alain Homewood, Insomnia Security; Leo Loobeek, @leoloobeek; Prashant Verma, Paladion; Travis Smith, Tripwire
Version: 2.1
Created: 31 May 2017
Last Modified: 20 April 2021

MITRE | ATT&CK®

Scheduled Task/Job: At (Windows)

Adversaries may abuse the at.exe utility to perform task scheduling for initial or recurring execution of malicious code. The at utility exists as an executable within Windows for scheduling tasks at a specified time and date. Using at requires that the Task Scheduler service be running, and the user to be logged on as a member of the local Administrators group.

An adversary may use at.exe in Windows environments to execute programs at system startup or on a scheduled basis for persistence. at can also be abused to conduct remote Execution as part of Lateral Movement and or to run a process under the context of a specified account (such as SYSTEM).

Note: The at.exe command line utility has been deprecated in current versions of Windows in favor of schtasks.

MITRE | ATT&CK®

Scheduled Task/Job: At (Windows)

ID: T1053.002
Sub-technique of: T1053
Tactics: Execution, Persistence, Privilege Escalation
Platforms: Windows
Permissions Required: Administrator
Data Sources: Command: Command Execution, File: File Modification, Process: Process Creation, Scheduled Job: Scheduled Job Creation
Supports Remote: Yes
Version: 1.0
Created: 27 November 2019
Last Modified: 24 March 2020

MITRE | ATT&CK®

Scheduled Task/Job: At (Windows)

Procedure Examples

MITRE | ATT&CK®

Scheduled Task/Job: At (Windows)

ID	Name	Description
G0026	APT18	APT18 actors used the native at Windows task scheduler tool to use scheduled tasks for execution on a victim network.[1]
S0110	at	at can be used to schedule a task on a system.[2]
G0060	BRONZE BUTLER	BRONZE BUTLER has used at to register a scheduled task to execute malware during lateral movement.[3]
S0488	CrackMapExec	CrackMapExec can set a scheduled task on the target system to execute commands remotely using at.[4]
S0233	MURKYTOP	MURKYTOP has the capability to schedule remote AT jobs.[5]
G0027	Threat Group-3390	Threat Group-3390 actors use at to schedule tasks to run self-extracting RAR archives, which install HTTPBrowser or PlugX on other victims on a network.[6]

MITRE | ATT&CK® | Mitigations

Scheduled Task/Job: At (Windows)

ID	Mitigation	Description
M1047	Audit	Toolkits like the PowerSploit framework contain PowerUp modules that can be used to explore systems for permission weaknesses in scheduled tasks that could be used to escalate privileges. [7] Windows operating system also creates a registry key specifically associated with the creation of a scheduled task on the destination host at: Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tree\At1. [8]
M1028	Operating System Configuration	Configure settings for scheduled tasks to force tasks to run under the context of the authenticated account instead of allowing them to run as SYSTEM. The associated Registry key is located at HKLM\SYSTEM\CurrentControlSet\Control\Lsa\SubmitControl. The setting can be configured through GPO: Computer Configuration > [Policies] > Windows Settings > Security Settings > Local Policies > Security Options: Domain Controller: Allow server operators to schedule tasks, set to disabled. [9]
M1026	Privileged Account Management	Configure the Increase Scheduling Priority option to only allow the Administrators group the rights to schedule a priority process. This can be configured through GPO: Computer Configuration > [Policies] > Windows Settings > Security Settings > Local Policies > User Rights Assignment: Increase scheduling priority. [10]
M1018	User Account Management	Limit privileges of user accounts and remediate Privilege Escalation vectors so only authorized administrators can create scheduled tasks on remote systems.

Detection

Monitor process execution from the svchost.exe in Windows 10 and the Windows Task Scheduler taskeng.exe for older versions of Windows. [11] If scheduled tasks are not used for persistence, then the adversary is likely to remove the task when the action is complete. Monitor Windows Task Scheduler stores in %systemroot%\System32\Tasks for change entries related to scheduled tasks that do not correlate with known software, patch cycles, etc.

Configure event logging for scheduled task creation and changes by enabling the "Microsoft-Windows-TaskScheduler/Operational" setting within the event logging service. [12] Several events will then be logged on scheduled task activity, including: [13][14]

- Event ID 106 on Windows 7, Server 2008 R2 - Scheduled task registered
- Event ID 140 on Windows 7, Server 2008 R2 / 4702 on Windows 10, Server 2016 - Scheduled task updated
- Event ID 141 on Windows 7, Server 2008 R2 / 4699 on Windows 10, Server 2016 - Scheduled task deleted
- Event ID 4698 on Windows 10, Server 2016 - Scheduled task created
- Event ID 4700 on Windows 10, Server 2016 - Scheduled task enabled
- Event ID 4701 on Windows 10, Server 2016 - Scheduled task disabled

Tools such as Sysinternals Autoruns may also be used to detect system changes that could be attempts at persistence, including listing current scheduled tasks. [15]

Remote access tools with built-in features may interact directly with the Windows API to perform these functions outside of typical system utilities. Tasks may also be created through Windows system management tools such as Windows Management Instrumentation and PowerShell, so additional logging may need to be configured to gather the appropriate data.

MITRE | ATT&CK®

Scheduled Task/Job: Scheduled Task

Adversaries may abuse the Windows Task Scheduler to perform task scheduling for initial or recurring execution of malicious code. There are multiple ways to access the Task Scheduler in Windows. The schtasks can be run directly on the command line, or the Task Scheduler can be opened through the GUI within the Administrator Tools section of the Control Panel. In some cases, adversaries have used a .NET wrapper for the Windows Task Scheduler, and alternatively, adversaries have used the Windows netapi32 library to create a scheduled task.

The deprecated at utility could also be abused by adversaries (ex: At (Windows)), though at.exe can not access tasks created with schtasks or the Control Panel.

An adversary may use Windows Task Scheduler to execute programs at system startup or on a scheduled basis for persistence. The Windows Task Scheduler can also be abused to conduct remote Execution as part of Lateral Movement and or to run a process under the context of a specified account (such as SYSTEM).

MITRE | ATT&CK®

Scheduled Task/Job: Scheduled Task

ID: T1053.005
Sub-technique of: T1053
Tactics: Execution, Persistence, Privilege Escalation
Platforms: Windows
Permissions Required: Administrator
Data Sources: Command: Command Execution, File: File Modification, Process: Process Creation, Scheduled Job: Scheduled Job Creation
Supports Remote: Yes
Version: 1.0
Created: 27 November 2019
Last Modified: 30 December 2020

MITRE | ATT&CK®

Scheduled Task/Job: Scheduled Task

Procedure Examples

ID	Name	Description
S0331	Agent Tesla	Agent Tesla has achieved persistence via scheduled tasks.[1]
S0504	Anchor	Anchor can create a scheduled task for persistence.[2]
S0584	AppleJeus	AppleJeus has created a scheduled SYSTEM task that runs when a user logs in.[3]
G0099	APT-C-36	APT-C-36 has used a macro function to set scheduled tasks, disguised as those used by Google.[4]
G0016	APT29	APT29 used scheduler and schtasks to create new tasks on remote hosts as part of lateral movement.[5] They have manipulated scheduled tasks by updating an existing legitimate task to execute their tools and then returned the scheduled task to its original configuration.[6] APT29 also created a scheduled task to maintain SUNSPOT persistence when the host booted during the 2020 SolarWinds intrusion.[7] They previously used named and hijacked scheduled tasks to also establish persistence.[8]
G0022	APT3	An APT3 downloader creates persistence by creating the following scheduled task: schtasks /create /tn "mysc" /tr C:\Users\Public\test.exe /sc ONLOGON /ru "System".[9]
G0050	APT32	APT32 has used scheduled tasks to persist on victim systems.[10][11][12][13]
G0064	APT33	APT33 has created a scheduled task to execute a .vbe file multiple times a day.[14]
G0087	APT39	APT39 has created scheduled tasks for persistence. [15][16][17]
G0096	APT41	APT41 used a compromised account to create a scheduled task on a system.[18][19]
S0438	Attor	Attor's installer plugin can schedule a new task that loads the dispatcher on boot/logon.[20]
S0414	BabyShark	BabyShark has used scheduled tasks to maintain persistence.[19]
S0475	BackConfig	BackConfig has the ability to use scheduled tasks to repeatedly execute malicious payloads on a compromised host.[21]
S0128	BADNEWS	BADNEWS creates a scheduled task to establish by executing a malicious payload every subsequent minute.[22]

ID	Name	Description
S0534	Bazar	Bazar can create a scheduled task for persistence.[23][24]
G0108	Blue Mockingbird	Blue Mockingbird has used Windows Scheduled Tasks to establish persistence on local and remote hosts.[25]
S0360	BONDUPDATER	BONDUPDATER persists using a scheduled task that executes every minute.[26]
G0060	BRONZE BUTLER	BRONZE BUTLER has used schtasks to register a scheduled task to execute malware during lateral movement.[27]
S0335	Carbon	Carbon creates several tasks for later execution to continue persistence on the victim's machine.[28]
G0114	Chimera	Chimera has used scheduled tasks to invoke Cobalt Strike including through batch script schtasks /create /ru "SYSTEM" /tn "update" /tr "cmd /c c:\windows\temp\update.bat" /sc once /f /st and to maintain persistence.[29][30]
G0080	Cobalt Group	Cobalt Group has created Windows tasks to establish persistence.[31]
S0126	ComRAT	ComRAT has used a scheduled task to launch its PowerShell loader.[32][33]
S0050	CosmicDuke	CosmicDuke uses scheduled tasks typically named "Watchmon Service" for persistence.[34]
S0046	CozyCar	One persistence mechanism used by CozyCar is to register itself as a scheduled task.[35]
S0538	Crutch	Crutch has the ability to persist using scheduled tasks.[36]
S0527	CSPY Downloader	CSPY Downloader can use the schtasks utility to bypass UAC.[37]
G0074	Dragonfly 2.0	Dragonfly 2.0 used scheduled tasks to automatically log out of created accounts every 8 hours as well as to execute malicious files.[38][39]
S0038	Duqu	Adversaries can instruct Duqu to spread laterally by copying itself to shares it has enumerated and for which it has obtained legitimate credentials (via keylogging or other means). The remote host is then infected by using the compromised credentials to schedule a task on remote machines that executes the malware.[40]

ID	Name	Description
S0024	Dyre	Dyre has the ability to achieve persistence by adding a new task in the task scheduler to run every minute.[41]
S0367	Emotet	Emotet has maintained persistence through a scheduled task. [42]
S0363	Empire	Empire has modules to interact with the Windows task scheduler.[43]
S0396	EvilBunny	EvilBunny has executed commands via scheduled tasks.[44]
G0051	FIN10	FIN10 has established persistence by using S4U tasks as well as the Scheduled Task option in PowerShell Empire.[45][43]
G0037	FIN6	FIN6 has used scheduled tasks to establish persistence for various malware it uses, including downloaders known as HARDTACK and SHIPBREAD and FrameworkPOS.[46]
G0046	FIN7	FIN7 malware has created scheduled tasks to establish persistence.[47][48][49][50]
G0061	FIN8	FIN8 has used scheduled tasks to maintain RDP backdoors.[51]
G0117	Fox Kitten	Fox Kitten has used Scheduled Tasks for persistence and to load and execute a reverse proxy binary.[52][53]
G0101	Frankenstein	Frankenstein has established persistence through a scheduled task using the command: /Create /F /SC DAILY /ST 09:00 /TN WinUpdate /TR , named "WinUpdate".[54]
G0093	GALLIUM	GALLIUM established persistence for PoisonIvy by created a scheduled task.[55]
G0047	Gamaredon Group	Gamaredon Group has created a scheduled task to launch an executable every 10 minutes.[56]
S0168	Gazer	Gazer can establish persistence by creating a scheduled task.[57][58]
S0588	GoldMax	GoldMax has used scheduled tasks to maintain persistence.[59]
S0477	Goopy	Goopy has the ability to maintain persistence by creating scheduled tasks set to run every hour.[12]
S0237	GravityRAT	GravityRAT creates a scheduled task to ensure it is re-executed everyday.[60]

ID	Name	Description
S0417	GRIFFON	GRIFFON has used sctasks for persistence. [61]
S0170	Helminth	Helminth has used a scheduled task for persistence.[62]
G0126	Higaisa	Higaisa dropped and added officeupdate.exe to scheduled tasks.[63][64]
S0431	HotCroissant	HotCroissant has attempted to install a scheduled task named "Java Maintenance64" on startup to establish persistence.[65]
S0483	IcedID	IcedID has created a scheduled task that executes every hour to establish persistence.[66]
S0260	InvisiMole	InvisiMole has used scheduled tasks named MSST and \Microsoft\Windows\Autochk\Scheduled to establish persistence.[67]
S0581	IronNetInjector	IronNetInjector has used a task XML file named mssch.xml to run an IronPython script when a user logs in or when specific system events are created.[68]
S0189	ISMInjector	ISMInjector creates scheduled tasks to establish persistence.[69]
S0044	JHUHUGIT	JHUHUGIT has registered itself as a scheduled task to run each time the current user logs in.[70][71]
S0532	Lucifer	Lucifer has established persistence by creating the following scheduled task schtasks /create /sc minute /mo 1 /tn QQMusic ^ /tr C:Users\%USERPROFILE%\Downloads\spread.exe /F.[72]
S0409	Machete	The different components of Machete are executed by Windows Task Scheduler.[73][74]
G0095	Machete	Machete has created scheduled tasks to maintain Machete's persistence.[75]
S0167	Matryoshka	Matryoshka can establish persistence by adding a Scheduled Task named "Microsoft Boost Kernel Optimization".[76][77]

mingnets

ID	Name	Description
S0449	Maze	Maze has created scheduled tasks using name variants such as "Windows Update Security", "Windows Update Security Patches", and "Google Chrome Security Update", to launch Maze at a specific time.[78]
S0500	MCMD	MCMD can use scheduled tasks for persistence.[79]
G0045	menuPass	menuPass has used a script (atexec.py) to execute a command on a target machine via Task Scheduler.[80]
G0021	Molerats	Molerats has created scheduled tasks to persistently run VBScripts.[81]
G0069	MuddyWater	MuddyWater has used scheduled tasks to establish persistence.[82]
G0129	Mustang Panda	Mustang Panda has created a scheduled task to execute additional malicious software, as well as maintain persistence.[83][84][85]
S0198	NETWIRE	NETWIRE can create a scheduled task to establish persistence.[86]
S0368	NotPetya	NotPetya creates a task to reboot the system one hour after infection.[87]
G0049	OilRig	OilRig has created scheduled tasks that run a VBScript to execute a payload on victim machines.[88][89][90]
S0439	Okrum	Okrum's installer can attempt to achieve persistence by creating a scheduled task.[91]
S0264	OopsIE	OopsIE creates a scheduled task to run itself every three minutes.[88][92]
G0116	Operation Wocao	Operation Wocao has used scheduled tasks to execute malicious PowerShell code on remote systems.[93]
G0040	Patchwork	A Patchwork file stealer can run a TaskScheduler DLL to add persistence.[94]
S0194	PowerSploit	PowerSploit's New-UserPersistenceOption Persistence argument can be used to establish via a Scheduled Task/Job.[95][96]
S0223	POWERSTATS	POWERSTATS has established persistence through a scheduled task using the command "C:\Windows\system32\schtasks.exe" /Create /F /SC DAILY /ST 12:00 /TN MicrosoftEdge /TR "c:\Windows\system32\wscript.exe C:\Windows\temp\Windows.vbe".[97]

ID	Name	Description
S0184	POWRUNER	POWRUNER persists through a scheduled task that executes it every minute.[98]
S0147	Pteranodon	Pteranodon schedules tasks to invoke its components in order to establish persistence.[99]
S0269	QUADAGENT	QUADAGENT creates a scheduled task to maintain persistence on the victim's machine.[89]
S0262	QuasarRAT	QuasarRAT contains a .NET wrapper DLL for creating and managing scheduled tasks for maintaining persistence upon reboot.[100]
S0458	Ramsay	Ramsay can schedule tasks via the Windows COM API to maintain persistence.[101]
G0075	Rancor	Rancor launched a scheduled task to gain persistence using the schtasks /create /sc command.[102]
S0375	Remexi	Remexi utilizes scheduled tasks as a persistence mechanism.[103]
S0166	RemoteCMD	RemoteCMD can execute commands remotely by creating a new schedule task on the remote system[104]
S0379	Revenge RAT	Revenge RAT schedules tasks to run malicious scripts at different intervals.[105]
S0148	RTM	RTM tries to add a scheduled task to establish persistence.[106][107]
S0446	Ryuk	Ryuk can remotely create a scheduled task to execute itself on a system.[108]
S0111	schtasks	schtasks is used to schedule tasks on a Windows system to run at a specific date and time.[109]

https://t.me/learn

ID	Name	Description
S0382	ServHelper	ServHelper contains modules that will use schtasks to carry out malicious operations.[110]
S0140	Shamoon	Shamoon copies an executable payload to the target system by using SMB/Windows Admin Shares and then scheduling an unnamed task to execute the malware.[111][112]
S0546	SharpStage	SharpStage has a persistence component to write a scheduled task for the payload.[113]
S0589	Sibot	Sibot has been executed via a scheduled task.[59]
G0091	Silence	Silence has used scheduled tasks to stage its operation.[114]
S0226	Smoke Loader	Smoke Loader launches a scheduled task.[115]
S0516	SoreFang	SoreFang can gain persistence through use of scheduled tasks.[116]
S0390	SQLRat	SQLRat has created scheduled tasks in %appdata%\Roaming\Microsoft\Templates\.[50]
G0038	Stealth Falcon	Stealth Falcon malware creates a scheduled task entitled "IE Web Cache" to execute a malicious file hourly.[117]
G0088	TEMP.Veles	TEMP.Veles has used scheduled task XML triggers.[118]
S0266	TrickBot	TrickBot creates a scheduled task on the system that provides persistence.[119][120][121]
S0476	Valak	Valak has used scheduled tasks to execute additional payloads and to gain persistence on a compromised host.[122][123][124]
G0102	Wizard Spider	Wizard Spider has used scheduled tasks establish persistence for TrickBot and other malware.[125][126][127][128]
S0248	yty	yty establishes persistence by creating a scheduled task with the command SchTasks /Create /SC DAILY /TN BigData /TR " + path_file + "/ST 09:30".[129]
S0251	Zebrocy	Zebrocy has a command to create a scheduled task for persistence.[130]
S0350	zwShell	zwShell has used SchTasks for execution.[131]

MITRE | ATT&CK® | Mitigations

Scheduled Task/Job: Scheduled Task

ID	Mitigation	Description
M1047	Audit	Toolkits like the PowerSploit framework contain PowerUp modules that can be used to explore systems for permission weaknesses in scheduled tasks that could be used to escalate privileges. [7] Windows operating system also creates a registry key specifically associated with the creation of a scheduled task on the destination host at: Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tree\At1. [8]
M1028	Operating System Configuration	Configure settings for scheduled tasks to force tasks to run under the context of the authenticated account instead of allowing them to run as SYSTEM. The associated Registry key is located at HKLM\SYSTEM\CurrentControlSet\Control\Lsa\SubmitControl. The setting can be configured through GPO: Computer Configuration > [Policies] > Windows Settings > Security Settings > Local Policies > Security Options: Domain Controller: Allow server operators to schedule tasks, set to disabled. [9]
M1026	Privileged Account Management	Configure the Increase Scheduling Priority option to only allow the Administrators group the rights to schedule a priority process. This can be configured through GPO: Computer Configuration > [Policies] > Windows Settings > Security Settings > Local Policies > User Rights Assignment: Increase scheduling priority. [10]
M1018	User Account Management	Limit privileges of user accounts and remediate Privilege Escalation vectors so only authorized administrators can create scheduled tasks on remote systems.

Detection

Monitor process execution from the svchost.exe in Windows 10 and the Windows Task Scheduler taskeng.exe for older versions of Windows. [11] If scheduled tasks are not used for persistence, then the adversary is likely to remove the task when the action is complete. Monitor Windows Task Scheduler stores in %systemroot%\System32\Tasks for change entries related to scheduled tasks that do not correlate with known software, patch cycles, etc.

Configure event logging for scheduled task creation and changes by enabling the "Microsoft-Windows-TaskScheduler/Operational" setting within the event logging service. [12] Several events will then be logged on scheduled task activity, including: [13][14]

- Event ID 140 on Windows 7, Server 2008 R2 / 4702 on Windows 10, Server 2016 - Scheduled task updated
- Event ID 141 on Windows 7, Server 2008 R2 / 4699 on Windows 10, Server 2016 - Scheduled task deleted
- Event ID 4698 on Windows 10, Server 2016 - Scheduled task created
- Event ID 4700 on Windows 10, Server 2016 - Scheduled task enabled
- Event ID 4701 on Windows 10, Server 2016 - Scheduled task disabled

Tools such as Sysinternals Autoruns may also be used to detect system changes that could be attempts at persistence, including listing current scheduled tasks. [15]

Remote access tools with built-in features may interact directly with the Windows API to perform these functions outside of typical system utilities. Tasks may also be created through Windows system management tools such as Windows Management Instrumentation and PowerShell, so additional logging may need to be configured to gather the appropriate data.

(SOURCE)

security.evtx
4648 – Logon specifying alternate credentials
- Current logged-on User Name
- Alternate User Name
- Destination Host Name/IP
- Process Name

(DESTINATION)

security.evtx
4624 Logon Type 3
- Source IP/Logon User Name

4672
- Logon User Name
- Logon by a user with administrative rights

Requirement for accessing default shares such as C$ and ADMIN$

4698 – Scheduled task created
4702 – Scheduled task updated
4699 – Scheduled task deleted
4700/4701 – Scheduled task
- enabled/disabled

Microsoft-Windows-Task Scheduler%4Operational.evtx
- 106 – Scheduled task created
- 140 – Scheduled task updated
- 141 – Scheduled task deleted
- 200/201 – Scheduled task executed/completed

EVENT LOGS

Scheduled Tasks

(SOURCE)

ShimCache – SYSTEM
- at.exe
- schtasks.exe

BAM/DAM – SYSTEM – LastTime Executed
- at.exe
- schtasks.exe

AmCache.hve -First Time Executed
- at.exe
- schtasks.exe

(DESTINATION)

SOFTWARE
Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tasks
Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tree\
ShimCache – SYSTEM
- evil.exe

AmCache.hve –First Time Executed
- evil.exe

REGISTRY

Scheduled Tasks

(SOURCE)

Prefetch – C:\Windows\Prefetch\
at.exe-{hash}.pf
schtasks.exe-{hash}.pf

```
at \\host 13:00 "c:\temp\evil.exe"
schtasks /CREATE /TN taskname /TR c:\temp\evil.exe
/SC once /RU “SYSTEM” /ST 13:00 /S host /U username
```

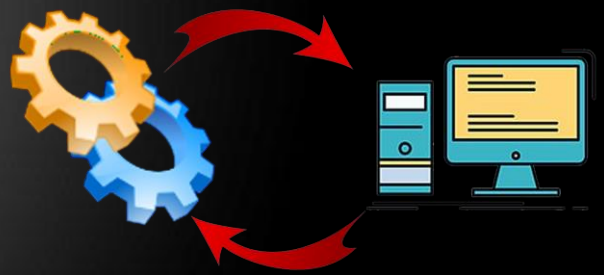
(DESTINATION)

File Creation
evil.exe
Job files created in
C:\Windows\Tasks
XML task files created in
C:\Windows\System32\Tasks
Author tag under "RegistrationInfo"can identify:
Source system name
Creator username
Prefetch – C:\Windows\Prefetch\
evil.exe-{hash}.pf

FILE SYSTEM

Scheduled Tasks

Services



MITRE | ATT&CK®

Windows Service

Adversaries may create or modify Windows services to repeatedly execute malicious payloads as part of persistence. When Windows boots up, it starts programs or applications called services that perform background system functions.[1] Windows service configuration information, including the file path to the service's executable or recovery programs/commands, is stored in the Windows Registry. Service configurations can be modified using utilities such as sc.exe and Reg.

Adversaries may install a new service or modify an existing service by using system utilities to interact with services, by directly modifying the Registry, or by using custom tools to interact with the Windows API. Adversaries may configure services to execute at startup in order to persist on a system.

An adversary may also incorporate Masquerading by using a service name from a related operating system or benign software, or by modifying existing services to make detection analysis more challenging. Modifying existing services may interrupt their functionality or may enable services that are disabled or otherwise not commonly used.

Services may be created with administrator privileges but are executed under SYSTEM privileges, so an adversary may also use a service to escalate privileges from administrator to SYSTEM. Adversaries may also directly start services through Service Execution.

MITRE | ATT&CK®

Windows Service

ID	Name
T1543.001	Launch Agent
T1543.002	Systemd Service
T1543.003	Windows Service
T1543.004	Launch Daemon

ID: T1543.003
Sub-technique of: T1543
Tactics: Persistence, Privilege Escalation
Platforms: Windows
Effective Permissions: Administrator, SYSTEM
Data Sources: Command: Command Execution, Process: OS API Execution, Process: Process Creation, Service: Service Creation, Service: Service Modification, Windows Registry: Windows Registry Key Creation, Windows Registry: Windows Registry Key Modification
CAPEC ID: CAPEC-478, CAPEC-550, CAPEC-551
Contributors: Matthew Demaske, Adaptforward; Pedro Harrison; Travis Smith, Tripwire
Version: 1.1
Created: 17 January 2020
Last Modified: 16 September 2020

MITRE | ATT&CK®

Windows Service

Procedure Examples

ID	Name	Description
S0504	Anchor	Anchor can establish persistence by creating a service.[2]
S0584	AppleJeus	AppleJeus can install itself as a service.[3]
G0073	APT19	An APT19 Port 22 malware variant registers itself as a service.[4]
G0022	APT3	APT3 has a tool that creates a new service for persistence.[5]
G0050	APT32	APT32 modified Windows Services to ensure PowerShell scripts were loaded on the system. APT32 also creates a Windows service to establish persistence.[6][7][8]
G0096	APT41	APT41 modified legitimate Windows services to install malware backdoors.[9] APT41 created the StorSyncSvc service to provide persistence for Cobalt Strike.[10]
S0438	Attor	Attor's dispatcher can establish persistence by registering a new service.[11]
S0347	AuditCred	AuditCred is installed as a new service on the system.[12]
S0239	Bankshot	Bankshot can terminate a specific process by its process id.[13][14]
S0127	BBSRAT	BBSRAT can modify service configurations.[15]
S0570	BitPaymer	BitPaymer has attempted to install itself as a service to maintain persistence.[16]
S0089	BlackEnergy	One variant of BlackEnergy creates a new service using either a hard-coded or randomly generated name.[17]
G0108	Blue Mockingbird	Blue Mockingbird has made their XMRIG payloads persistent as a Windows Service.[18]
S0204	Briba	Briba installs a service pointing to a malicious DLL dropped to disk.[19]
G0008	Carbanak	Carbanak malware installs itself as a service to provide persistence and SYSTEM privileges.[20]
S0335	Carbon	Carbon establishes persistence by creating a service and naming it based off the operating system version running on the current machine.[21]
S0261	Catchamas	Catchamas adds a new service named NetAdapter to establish persistence.[22]

ID	Name	Description
G0080	Cobalt Group	Cobalt Group has created new services to establish persistence.[23]
S0154	Cobalt Strike	Cobalt Strike can install a new service.[24]
S0050	CosmicDuke	CosmicDuke uses Windows services typically named "javamtsup" for persistence.[25]
S0046	CozyCar	One persistence mechanism used by CozyCar is to register itself as a Windows service.[26]
G0105	DarkVishnya	DarkVishnya created new services for shellcode loaders distribution.[27]
S0567	Dtrack	Dtrack can add a service called WBService to establish persistence.[28]
S0038	Duqu	Duqu creates a new service that loads a malicious driver when the system starts. When Duqu is active, the operating system believes that the driver is legitimate, as it has been signed with a valid private key.[29]
S0024	Dyre	Dyre registers itself as a service by adding several Registry keys.[30]
S0081	Elise	Elise configures itself as a service.[31]
S0082	Emissary	Emissary is capable of configuring itself as a service.[32]
S0367	Emotet	Emotet has been observed creating new services to maintain persistence. [33][34]
S0363	Empire	Empire can utilize built-in modules to modify service binaries and restore them to their original state.[35]

https://t.me/learn

ID	Name	Description
S0343	Exaramel for Windows	The Exaramel for Windows dropper creates and starts a Windows service named wsmprovav with the description "Windows Check AV."[36]
S0181	FALLCHILL	FALLCHILL has been installed as a Windows service.[3]
G0046	FIN7	FIN7 created new Windows services and added them to the startup directories for persistence.[37]
S0182	FinFisher	FinFisher creates a new Windows service with the malicious executable for persistence.[38][39]
S0032	gh0st RAT	gh0st RAT can create a new service to establish persistence.[40][41]
S0493	GoldenSpy	GoldenSpy has established persistence by running in the background as an autostart service.[42]
S0342	GreyEnergy	GreyEnergy chooses a service, drops a DLL file, and writes it to that serviceDLL Registry key.[43]
S0071	hcdLoader	hcdLoader installs itself as a service for persistence.[44][45]
G0072	Honeybee	Honeybee has batch files that modify the system service COMSysApp to load a malicious DLL.[46]
S0203	Hydraq	Hydraq creates new services to establish persistence.[47][48][49]
S0259	InnaputRAT	Some InnaputRAT variants create a new Windows service to establish persistence.[50]
S0260	InvisiMole	InvisiMole can register a Windows service named CsPower as part of its execution chain, and a Windows service named clr_optimization_v2.0.51527_X86 to achieve persistence.[51]
S0044	JHUHUGIT	JHUHUGIT has registered itself as a service to establish persistence.[52]
S0265	Kazuar	Kazuar can install itself as a new service.[53]
G0004	Ke3chang	Ke3chang backdoor RoyalDNS established persistence through adding a service called Nwsapagent.[54]
S0387	KeyBoy	KeyBoy installs a service pointing to a malicious DLL dropped to disk.[55]
G0094	Kimsuky	Kimsuky has created new services for persistence.[56][57]

ID	Name	Description
S0236	Kwampirs	Kwampirs creates a new service named WmiApSrvEx to establish persistence.[58]
G0032	Lazarus Group	Several Lazarus Group malware families install themselves as new services on victims.[59][60]
S0451	LoudMiner	LoudMiner can automatically launch a Linux virtual machine as a service at startup if the AutoStart option is enabled in the VBoxVmService configuration file.[61]
S0149	MoonWind	MoonWind installs itself as a new service with automatic startup to establish persistence. The service checks every 60 seconds to determine if the malware is running; if not, it will spawn a new instance.[62]
S0205	Naid	Naid creates a new service to establish.[63]
S0210	Nerex	Nerex creates a Registry subkey that registers a new service.[64]
S0118	Nidiran	Nidiran can create a new service named msamger (Microsoft Security Accounts Manager).[65]
S0439	Okrum	To establish persistence, Okrum can install itself as a new service named NtmSsvc.[66]
S0501	PipeMon	PipeMon can establish persistence by registering a malicious DLL as an alternative Print Processor which is loaded when the print spooler service starts.[67]
S0013	PlugX	PlugX can be added as a service to establish persistence. PlugX also has a module to change service configurations as well as start, control, and delete services.[68][69][70][71][72]
S0012	PoisonIvy	PoisonIvy creates a Registry subkey that registers a new service. PoisonIvy also creates a Registry entry modifying the Logical Disk Manager service to point to a malicious DLL dropped to disk.[73]

https://t.me/learningnets

ID	Name	Description
S0169	RawPOS	RawPOS installs itself as a service to maintain persistence.[78][79][80]
S0495	RDAT	RDAT has created a service when it is installed on the victim machine.[81]
S0172	Reaver	Reaver installs itself as a new service.[82]
S0074	Sakula	Some Sakula samples install themselves as services for persistence by calling WinExec with the net start argument.[83]
S0345	Seasalt	Seasalt is capable of installing itself as a service.[84]
S0140	Shamoon	Shamoon creates a new service named "ntssrv" to execute the payload. Newer versions create the "MaintenaceSrv" and "hdv_725x" services.[85][86]
S0444	ShimRat	ShimRat has installed a Windows service to maintain persistence on victim machines.[87]
S0533	SLOTHFULMEDIA	SLOTHFULMEDIA has created a service on victim machines named "TaskFrame" to establish persistence.[88]
S0142	StreamEx	StreamEx establishes persistence by installing a new service pointing to its DLL and setting the service to auto-start.[89]
S0491	StrongPity	StrongPity has created new services and modified existing services for persistence.[90]
S0164	TDTESS	If running as administrator, TDTESS installs itself as a new service named bmwappushservice to establish persistence.[91]
S0560	TEARDROP	TEARDROP ran as a Windows service from the c:\windows\syswow64 folder.[92][93]
G0027	Threat Group-3390	A Threat Group-3390 tool can create a new service, naming it after the config information, to gain persistence.[94]
S0004	TinyZBot	TinyZBot can install as a Windows service for persistence.[95]
S0266	TrickBot	TrickBot establishes persistence by creating an autostart service that allows it to run whenever the machine boots.[96]
G0081	Tropic Trooper	Tropic Trooper has installed a service pointing to a malicious DLL dropped to disk.[97]

ID	Name	Description
S0263	TYPEFRAME	TYPEFRAME variants can add malicious DLL modules as new services.TYPEFRAME can also delete services from the victim’s machine.[98]
S0386	Ursnif	Ursnif has registered itself as a system service in the Registry for automatic execution at system startup.[99]
S0180	Volgmer	Volgmer installs a copy of itself in a randomly selected service, then overwrites the ServiceDLL entry in the service's Registry entry. Some Volgmer variants also install .dll files as services with names generated by a list of hard-coded strings.[100][101][102]
S0366	WannaCry	WannaCry creates the service "mssecsvc2.0" with the display name "Microsoft Security Center (2.0) Service."[103][104]
S0206	Wiarp	Wiarp creates a backdoor through which remote attackers can create a service.[105]
S0176	Wingbird	Wingbird uses services.exe to register a new autostart service named "Audit Service" using a copy of the local lsass.exe file.[106][107]
S0141	Winnti for Windows	Winnti for Windows sets its DLL file as a new service in the Registry to establish persistence.[108]
G0102	Wizard Spider	Wizard Spider has installed TrickBot as a service named ControlServiceA in order to establish persistence.[109]
S0230	ZeroT	ZeroT can add a new service to ensure PlugX persists on the system when delivered as another payload onto the system.[72]
S0086	ZLib	ZLib creates Registry keys to allow itself to run as various services.[110]
S0350	zwShell	zwShell has established persistence by adding itself as a new service.[111]
S0412	ZxShell	ZxShell can create a new service using the service parser function ProcessScCommand.[112]

MITRE | ATT&CK® | Mitigations

Windows Service

ID	Mitigation	Description
M1047	Audit	Use auditing tools capable of detecting privilege and service abuse opportunities on systems within an enterprise and correct them.
M1018	User Account Management	Limit privileges of user accounts and groups so that only authorized administrators can interact with service changes and service configurations.

Detection

Monitor processes and command-line arguments for actions that could create or modify services. Command-line invocation of tools capable of adding or modifying services may be unusual, depending on how systems are typically used in a particular environment. Services may also be modified through Windows system management tools such as Windows Management Instrumentation and PowerShell, so additional logging may need to be configured to gather the appropriate data. Remote access tools with built-in features may also interact directly with the Windows API to perform these functions outside of typical system utilities. Collect service utility execution and service binary path arguments used for analysis. Service binary paths may even be changed to execute commands or scripts.

- Look for changes to service Registry entries that do not correlate with known software, patch cycles, etc. Service information is stored in the Registry at HKLM\SYSTEM\CurrentControlSet\Services. Changes to the binary path and the service startup type changed from manual or disabled to automatic, if it does not typically do so, may be suspicious. Tools such as Sysinternals Autoruns may also be used to detect system service changes that could be attempts at persistence.[113]
- Creation of new services may generate an alterable event (ex: Event ID 4697 and/or 7045 [114][115]). New, benign services may be created during installation of new software.
- Suspicious program execution through services may show up as outlier processes that have not been seen before when compared against historical data. Look for abnormal process call trees from known services and for execution of other commands that could relate to Discovery or other adversary techniques. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as network connections made for Command and Control, learning details about the environment through Discovery, and Lateral Movement.

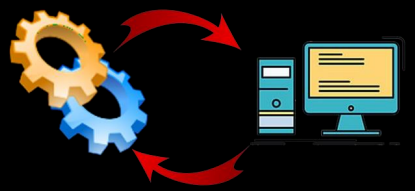
(SOURCE)

(DESTINATION)

security.evtx
4624 Logon Type 3
Source IP/Logon User Name
4697
Security records service install,if enabled
Enabling non-default Security events such as ID 4697 are particularly useful if only the Security logs are forwarded to a centralized log server

system.evtx
7034 – Service crashed unexpectedly
7035 – Service sent a Start/Stop control
7036 – Service started or stopped
7040 – Start type changed (Boot | On Request | Disabled)
7045 – A service was installed on the system

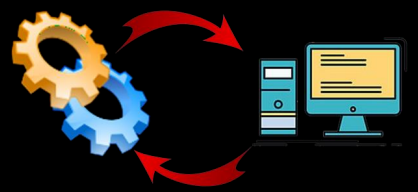
EVENT LOGS



Services

(SOURCE)

ShimCache – SYSTEM
sc.exe
BAM/DAM – SYSTEM – Last Time Executed
sc.exe
AmCache.hve – First Time Executed
sc.exe



REGISTRY

Services

(DESTINATION)

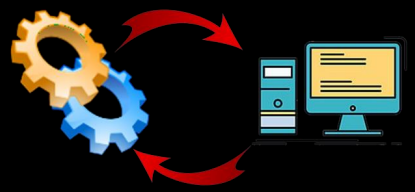
SYSTEM
\CurrentControlSet\Services\
New service creation
ShimCache – SYSTEM
evil.exe
ShimCache records existence of malicious service executable, unless implemented as a service DLL
AmCache.hve –First Time Executed
evil.exe

(SOURCE)

Prefetch – C:\Windows\Prefetch\
sc.exe-{hash}.pf

sc \\host create servicename binpath= “c:\temp\evil.exe”
sc \\host start servicename

FILE SYSTEM



Services

(DESTINATION)

File Creation
evil.exe or evil.dll malicious service executable or service DLL
Prefetch – C:\Windows\Prefetch\
evil.exe-{hash}.pf

WMI
WMIC

MITRE | ATT&CK®

Windows Management Instrumentation

Windows Management Instrumentation

Adversaries may abuse Windows Management Instrumentation (WMI) to achieve execution. WMI is a Windows administration feature that provides a uniform environment for local and remote access to Windows system components. It relies on the WMI service for local and remote access and the server message block (SMB) [1] and Remote Procedure Call Service (RPCS) [2] for remote access. RPCS operates over port 135. [3]

An adversary can use WMI to interact with local and remote systems and use it as a means to perform many tactic functions, such as gathering information for Discovery and remote Execution of files as part of Lateral Movement

MITRE | ATT&CK®

Windows Management Instrumentation

ID: T1047
Sub-techniques: No sub-techniques
Tactic: Execution
Platforms: Windows
System Requirements: WMI service, winmgmt, running; Host/network firewalls allowing SMB and WMI ports from source to destination; SMB authentication.
Permissions Required: Administrator, User
Data Sources: Command: Command Execution, Network Traffic: Network Connection Creation, Process: Process Creation
Supports Remote: Yes
Version: 1.1
Created: 31 May 2017
Last Modified: 13 May 2020

MITRE | ATT&CK®

Windows Management Instrumentation

Procedure Examples

ID	Name	Description
S0331	Agent Tesla	Agent Tesla has used wmi queries to gather information from the system.[6]
G0016	APT29	APT29 used WMI to steal credentials and execute backdoors at a future time.[7] They have also used WMI for the remote execution of files for lateral movement.[8][9]
G0050	APT32	APT32 used WMI to deploy their tools on remote machines and to gather information about the Outlook process.[10]
G0096	APT41	APT41 used WMI in several ways, including for execution of commands via WMIEXEC as well as for persistence via PowerSploit.[11]
S0373	Astaroth	Astaroth uses WMIC to execute payloads. [12]
S0534	Bazar	Bazar can execute a WMI query to gather information about the installed antivirus engine.[13][14]
S0089	BlackEnergy	A BlackEnergy 2 plug-in uses WMI to gather victim host details.[15]
G0108	Blue Mockingbird	Blue Mockingbird has used wmic.exe to set environment variables.[16]
G0114	Chimera	Chimera has used WMIC to execute remote commands.[17][18]
S0154	Cobalt Strike	Cobalt Strike can use WMI to deliver a payload to a remote host.[19]
S0488	CrackMapExec	CrackMapExec can execute remote commands using Windows Management Instrumentation.[20]
G0009	Deep Panda	The Deep Panda group is known to utilize WMI for lateral movement.[21]
S0062	DustySky	The DustySky dropper uses Windows Management Instrumentation to extract information about the operating system and whether an anti-virus is active.[22]
S0367	Emotet	Emotet has used WMI to execute powershell.exe.[23]
S0363	Empire	Empire can use WMI to deliver a payload to a remote host.[24]
S0396	EvilBunny	EvilBunny has used WMI to gather information about the system.[25]
S0568	EVILNUM	EVILNUM has used the Windows Management Instrumentation (WMI) tool to enumerate infected machines.[26]

ID	Name	Description
S0267	FELIXROOT	FELIXROOT uses WMI to query the Windows Registry.[27]
G0037	FIN6	FIN6 has used WMI to automate the remote execution of PowerShell scripts.[28]
G0061	FIN8	FIN8's malicious spearphishing payloads use WMI to launch malware and spawn cmd.exe execution. FIN8 has also used WMIC during and post compromise cleanup activities.[29][30]
S0381	FlawedAmmyy	FlawedAmmyy leverages WMI to enumerate anti-virus on the victim.[31]
G0101	Frankenstein	Frankenstein has used WMI queries to check if various security applications were running, as well as the operating system version.[32]
G0093	GALLIUM	GALLIUM used WMI for execution to assist in lateral movement as well as for installing tools across multiple assets.[33]
S0237	GravityRAT	GravityRAT collects various information via WMI requests, including CPU information in the Win32_Processor entry (Processor ID, Name, Manufacturer and the clock speed).[34]
S0151	HALFBAKED	HALFBAKED can use WMI queries to gather system information.[35]
S0376	HOPLIGHT	HOPLIGHT has used WMI to recompile the Managed Object Format (MOF) files in the WMI repository.[36]
S0483	IcedID	IcedID has used WMI to execute binaries.[37]
S0357	Impacket	Impacket's wmiexec module can be used to execute commands through WMI.[38]

https://t.me/learningnets

ID	Name	Description
S0283	jRAT	jRAT uses WMIC to identify anti-virus products installed on the victim’s machine and to obtain firewall details.[39]
S0265	Kazuar	Kazuar obtains a list of running processes through WMI querying.[40]
S0250	Koadic	Koadic can use WMI to execute commands.[41]
S0156	KOMPROGO	KOMPROGO is capable of running WMI queries.[42]
G0032	Lazarus Group	Lazarus Group malware SierraAlfa uses the Windows Management Instrumentation Command-line application wmic to start itself on a target system during lateral movement.[43][44]
G0065	Leviathan	Leviathan has used WMI for execution.[45]
S0532	Lucifer	Lucifer can use WMI to log into remote machines for propagation.[46]
S0449	Maze	Maze has used WMI to attempt to delete the shadow volumes on a machine, and to connect a virtual machine to the network domain of the victim organization's network.[47][48]
G0045	menuPass	menuPass has used a modified version of pentesting script wmiexec.vbs, which logs into a remote machine using WMI.[49][50][51]
S0339	Micropsia	Micropsia searches for anti-virus software and firewall products installed on the victim’s machine using WMI.[52][53]
S0553	MoleNet	MoleNet can perform WMI commands on the system.[54]
S0256	Mosquito	Mosquito's installer uses WMI to search for antivirus display names.[55]
G0069	MuddyWater	MuddyWater has used malware that leveraged WMI for execution and querying host information.[56][57][58]
G0129	Mustang Panda	Mustang Panda has executed PowerShell scripts via WMI.[59][60]
S0457	Netwalker	Netwalker can use WMI to delete Shadow Volumes.[61]

ID	Name	Description
S0368	NotPetya	NotPetya can use wmic to help propagate itself across a network.[62][63]
S0340	Octopus	Octopus uses wmic.exe for local discovery information.[64]
G0049	OilRig	OilRig has used WMI for execution.[65]
S0365	Olympic Destroyer	Olympic Destroyer uses WMI to help propagate itself across a network.[66]
S0264	OopsIE	OopsIE uses WMI to perform discovery techniques.[67]
G0116	Operation Wocao	Operation Wocao has used WMI to execute commands.[68]
S0378	PoshC2	PoshC2 has a number of modules that use WMI to execute tasks.[69]
S0194	PowerSploit	PowerSploit's Invoke-WmiCommand CodeExecution module uses WMI to execute and retrieve the output from a PowerShell payload.[70][71]
S0223	POWERSTATS	POWERSTATS can use WMI queries to retrieve data from compromised hosts.[72][57]
S0184	POWRUNER	POWRUNER may use WMI when collecting information about a victim.[73]
S0241	RATANKBA	RATANKBA uses WMI to perform process monitoring.[74][75]
S0375	Remexi	Remexi executes received commands with wmic.exe (for WMI commands). [76]

https://t.me/lear

ID	Name	Description
S0496	REvil	REvil can use WMI to monitor for and kill specific processes listed in its configuration file.[77][78]
S0270	RogueRobin	RogueRobin uses various WMI queries to check if the sample is running in a sandbox.[79][80]
S0546	SharpStage	SharpStage can use WMI for execution.[54][81]
S0589	Sibot	Sibot has used WMI to discover network connections and configurations. Sibot has also used the Win32_Process class to execute a malicious DLL.[82]
G0038	Stealth Falcon	Stealth Falcon malware gathers system information via Windows Management Instrumentation (WMI).[83]
S0380	StoneDrill	StoneDrill has used the WMI command-line (WMIC) utility to run tasks.[84]
S0559	SUNBURST	SUNBURST used the WMI query Select * From Win32_SystemDriver to retrieve a driver listing.[85]
G0027	Threat Group-3390	A Threat Group-3390 tool can use WMI to execute a binary.[86]
S0386	Ursnif	Ursnif droppers have used WMI classes to execute PowerShell commands.[87]
S0476	Valak	Valak can use wmic process call create in a scheduled task to launch plugins and for execution.[88]
S0366	WannaCry	WannaCry utilizes wmic to delete shadow copies.[89][90][91]
G0112	Windshift	Windshift has used WMI to collect information about target machines.[92]
G0102	Wizard Spider	Wizard Spider has used WMI and LDAP queries for network discovery and to move laterally.[93][94][95][96]
S0251	Zebrocy	One variant of Zebrocy uses WMI queries to gather information.[97]

(SOURCE)

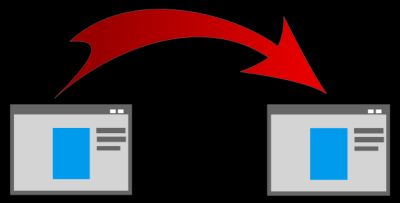
security.evtx
4648 – Logon specifying alternate credentials
- Current logged-on User Name
- Alternate User Name
- Destination Host Name/IP
- Process Name

(DESTINATION)

security.evtx
4624 Logon Type 3
- Source IP/Logon User Name

4672
- Logon User Name
- Logon by an a user with administrative rights

EVENT LOGS



WMI
WMIC

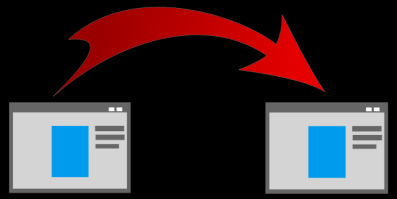
(SOURCE)

ShimCache – SYSTEM
wmic.exe
BAM/DAM – SYSTEM – Last Time Executed
wmic.exe
AmCache.hve – First Time Executed
wmic.exe

(DESTINATION)

ShimCache – SYSTEM
scrcons.exe
mofcomp.exe
wmiprvse.exe
evil.exe
AmCache.hve –First Time Executed
scrcons.exe
mofcomp.exe
wmiprvse.exe
evil.exe

REGISTRY



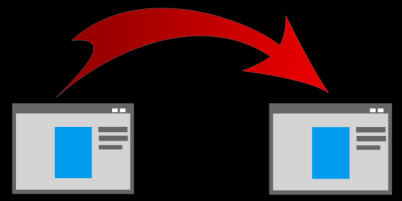
WMI
WMIC

(SOURCE)

Prefetch – C:\Windows\Prefetch\
wmic.exe-{hash}.pf

```
wmic /node:host process call create "C:\temp\evil.exe"
Invoke-WmiMethod –Computer host –Class Win32_Process –Name create –
Argument “c:\temp\evil.exe"
```

FILE SYSTEM



WMI WMIC

(DESTINATION)

ShimCache – SYSTEM
- scrcons.exe
- mofcomp.exe
- wmiprvse.exe
- evil.exe

AmCache.hve –First Time Executed
- scrcons.exe
- mofcomp.exe
- wmiprvse.exe
- evil.exe

MITRE | ATT&CK®

Windows Management Instrumentation

Mitigations

ID	Mitigation	Description
M1026	Privileged Account Management	Prevent credential overlap across systems of administrator and privileged accounts. [5]
M1018	User Account Management	By default, only administrators are allowed to connect remotely using WMI. Restrict other users who are allowed to connect, or disallow all users to connect remotely to WMI.

Detection

Monitor network traffic for WMI connections; the use of WMI in environments that do not typically use WMI may be suspect. Perform process monitoring to capture command-line arguments of "wmic" and detect commands that are used to perform remote behavior. [5]

PowerShell Remoting

MITRE | ATT&CK®

PowerShell Profile

ID	Name
T1546.001	Change Default File Association
T1546.002	Screensaver
T1546.003	Windows Management Instrumentation Event Subscription
T1546.004	Unix Shell Configuration Modification
T1546.005	Trap
T1546.006	LC_LOAD_DYLIB Addition
T1546.007	Netsh Helper DLL
T1546.008	Accessibility Features
T1546.009	AppCert DLLs
T1546.010	AppInit DLLs
T1546.011	Application Shimming
T1546.012	Image File Execution Options Injection
T1546.013	PowerShell Profile
T1546.014	Emond
T1546.015	Component Object Model Hijacking

Adversaries may gain persistence and elevate privileges by executing malicious content triggered by PowerShell profiles. A PowerShell profile (profile.ps1) is a script that runs when PowerShell starts and can be used as a logon script to customize user environments.

PowerShell supports several profiles depending on the user or host program. For example, there can be different profiles for PowerShell host programs such as the PowerShell console, PowerShell ISE or Visual Studio Code. An administrator can also configure a profile that applies to all users and host programs on the local computer. [1]

Adversaries may modify these profiles to include arbitrary commands, functions, modules, and/or PowerShell drives to gain persistence. Every time a user opens a PowerShell session the modified script will be executed unless the -NoProfile flag is used when it is launched. [2]

An adversary may also be able to escalate privileges if a script in a PowerShell profile is loaded and executed by an account with higher privileges, such as a domain administrator. [3]

MITRE | ATT&CK®

PowerShell Profile

Procedure Examples

MITRE | ATT&CK®

PowerShell Profile

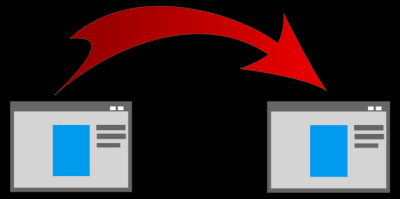
ID	Name	Description
G0010	Turla	Turla has used PowerShell profiles to maintain persistence on an infected machine.[2]

(SOURCE)

security.evtx
4648 – Logon specifying alternate credentials
Current logged-on User Name
Alternate User Name
Destination Host Name/IP
Process Name
Microsoft-WindowsWinRM%4Operational.evtx 6 – WSMan Session initialize
Session created
Destination Host Name or IP
Current logged-on User Name

8, 15, 16, 33 – WSMan Session deinitialization
Closing of WSMan session
Current logged-on User Name
Microsoft-WindowsPowerShell%4Operational.evtx
40961, 40962
Records the local initiation of powershell.exe and associated user account
8193 & 8194
Session created
8197 - Connect
Session close

EVENT LOGS



PowerShell Remoting

(DESTINATION)

security.evtx
4624 Logon Type 3
Source IP/Logon User Name
4672
Logon User Name
Logon by an a user with administrative rights
Microsoft-WindowsPowerShell%4Operational.evtx
4103, 4104 – Script Block logging
Logs suspicious scripts by default in PS v5
Logs all scripts if configured
53504 Records the authenticating user

Windows PowerShell.evtx
400/403 "ServerRemoteHost“ indicates start/end of Remoting session
800 Includes partial script code Microsoft WindowsWinRM%4Operational.evtx
91 Session creation
168 Records the authenticating user

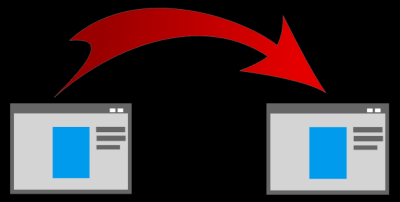
(SOURCE)

ShimCache – SYSTEM
powershell.exe
BAM/DAM – SYSTEM –Last Time Executed
powershell.exe
AmCache.hve – First Time Executed
powershell.exe

(DESTINATION)

ShimCache – SYSTEM
wsmprovhost.exe
evil.exe
SOFTWARE
Microsoft\PowerShell\1\ShellIds\Microsoft.PowerShell\ExecutionPolicy
Attacker may change
execution policy to a less restrictive setting, such as "bypass"
AmCache.hve – First Time Executed
wsmprovhost.exe
evil.ex

REGISTRY



PowerShell Remoting

(SOURCE)

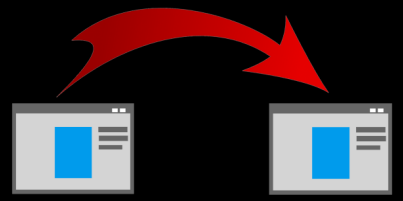
Prefetch – C:\Windows\Prefetch\
powershell.exe-{hash}.pf
PowerShell scripts (.ps1 files) that run within 10 seconds of powershell.exe launching will be tracked in powershell.exe prefetch file
Command history
C:\USERS\<USERNAME>\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline\ConsoleHost_history.txt
With PS v5+, a history file with previous 4096 commands is maintained per user

```
Enter-PSSession –ComputerName host
Invoke-Command –ComputerName host –ScriptBlock {Start-Process c:\temp\evil.exe}
```

(DESTINATION)

File Creation
evil.exe
With Enter-PSSession, a user profile directory may be created
Prefetch – C:\Windows\Prefetch\
evil.exe-{hash].pf
wsmprovhost.exe-{hash].pf

FILE SYSTEM



PowerShell Remoting

MITRE | ATT&CK® | Mitigations

PowerShell Profile

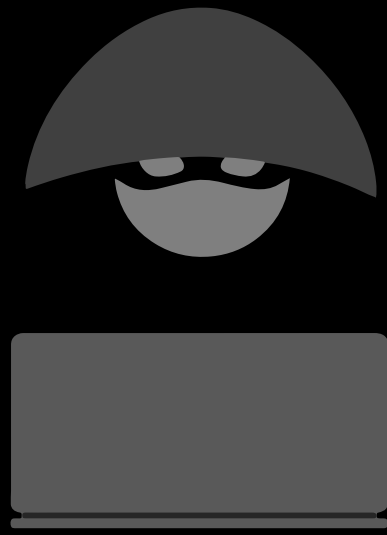
ID	Mitigation	Description
M1045	Code Signing	Enforce execution of only signed PowerShell scripts. Sign profiles to avoid them from being modified.
M1022	Restrict File and Directory Permissions	Making PowerShell profiles immutable and only changeable by certain administrators will limit the ability for adversaries to easily create user level persistence.
M1054	Software Configuration	Avoid PowerShell profiles if not needed. Use the -No Profile flag with when executing PowerShell scripts remotely to prevent local profiles and scripts from being executed.

Detection

Locations where profile.ps1 can be stored should be monitored for new profiles or modifications. [4] Example profile locations include:

- $PsHome\Profile.ps1
- $PsHome\Microsoft.{{HostProgram}}_profile.ps1
- $Home\My Documents\PowerShell\Profile.ps1
- $Home\My Documents\PowerShell\Microsoft.{{HostProgram}}_profile.ps1

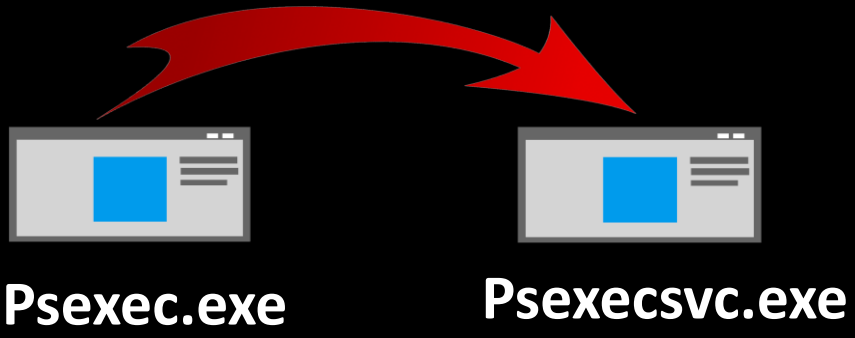
Monitor abnormal PowerShell commands, unusual loading of PowerShell drives or modules, and/or execution of unknown programs.



WMIWMIC

PowerShell

PsExec



Hunt Evil by SYSMON

This “Windows Sysmon” is intended to help you understand where Microsoft’s FREE Sysinternals Sysmon agent can supplement and enhance your Windows Logging, NOT replace it. Sysmon can provide more information than standard default Windows logs provide. Sysmon is great to collect data you need for Incident Response, malware labs, high security situations, your own personal systems, or just improve the existing log data you are collecting

This is an example to help understand where the overlap between Windows logs and what Sysmon covers.

SYSMON

Event ID 1: Process creation

Event ID 2: A process changed a file creation time

Event ID 3: Network connection

Event ID 4: Sysmon service state changed

Event ID 5: Process terminated

Event ID 6: Driver loaded

Event ID 7: Image loaded

Event ID 8: CreateRemoteThread

Event ID 9: RawAccessRead

Event ID 10: ProcessAccess

Event ID 11: FileCreate

Event ID 13: RegistryEvent (Value Set)

Event ID 14: RegistryEvent (Key and Value Rename)

Event ID 15: FileCreateStreamHash

Event ID 16: ServiceConfigurationChange

Event ID 17: PipeEvent (Pipe Created)

Event ID 18: PipeEvent (Pipe Connected)

Event ID 19: WmiEvent (WmiEventFilter activity detected)

Event ID 20: WmiEvent (WmiEventConsumer activity detected)

Event ID 21: WmiEvent (WmiEventConsumerToFilter activity detected)

Event ID 22: DNSEvent (DNS query)

Event ID 23: FileDelete (File Delete archived)

Event ID 24: ClipboardChange (New content in the clipboard)

Event ID 25: ProcessTampering (Process image change)

Event ID 26: FileDeleteDetected (File Delete logged)

Event ID 255: Error

Sysmon ID	windowsID	Event Type	Note	Valuable Additional data	Production Recommendation
1	Security - 4688	Process Creation	Noisy (3)	Hash of the process/file	Use Windows
2	Security - 4657	Process Changed a file creation time	///////////////	//	Use Windows
3	Security - 5156	Network Connection	Noisy (3)	Provides some name resolution of IP	Use Windows
4	(1)	Sysmon Service State Change	///////////////	//	Collect
5	Security - 4689	Process Terminated	Noisy (3)	//	Use Windows
6	System 6, 219, 7026	Driver Loaded	///////////////	//	Collect
7	n/a	Image Loaded	Noisy (3)	Most malware is NOT Signed, so is .NET	Collect Signed False only
8	n/a	Create Remote Thread	///////////////	//	Optional
9	n/a	Raw File Access Read	///////////////	//	Optional
10	//////////////////////////////	Process Access	Noisy (3)	//	Optional
11	4663	File Create	Noisy (3)	//	Include only (2)
12	4657	Registry Create and Delete	Noisy (3)	//	Include only (2)
13	4657	Registry Value Set	Noisy (3)	//	Include only (2)
14	4657	Registry Key and Value Rename	///////////////	//	Collect
15	n/a	File Create Stream Hash	///////////////	//	Collect
16	(1)	Sysmon Config Change	///////////////	//	Collect
17	n/a	Pipe Event Created	///////////////	//	Collect

https://t.me/learningnets

Sysmon ID	windowsID	Event Type	note	Valuable Additional data	Production Recommendation
18	n/a	Pipe Event Connected	///////////////	//	Use Windows (4)
19	5861, 5858, 5859	WMI EventFilter activity	///////////////	//	Use Windows (4)
20	5861, 5858, 5859	WMI EventConsumer activity	///////////////	//	Use Windows (4)
21	5861, 5858, 5859	WMI EventConsumerToFilter activity	///////////////	//	Use Windows (4)
22	1016, 3008, 3010, 3020	DNS Query	Noisy (3)	Process that made the DNS Query	Use Windows (4)
225	n/a	Sysmon error	///////////////	//	Collect

1	**Refer to the Windows Advanced Logging Cheat Sheet on how to monitor Service with built-in Windows logging**
2	It is recommended to use Windows logging as the primary source, enable a policy of what you want to audit and apply to all systems
3	These events are incredibly noisy and will effect how long you can keep a history in the local logs - These will take a lot of time to filter down
4	Sysmon is optional, but pick only one

Windows Security Log

User Account Changes			
4720	Created	4722	Enabled
4723	User Changed Own Password		
4724	Privileged User Changed this User's Password		
4725	Disabled	4726	Deleted
4738	Changed	4740	Locked Out
4767	Unlocked	4781	Name Change

Domain Controller Authentication Events		
4768	A Kerberos authentication ticket (TGT) was requested	
4771	Kerberos pre authentication failed	See Kerberos Failure Codes
4820	A Kerberos TGT was denied because the device does not meet the access control restrictions	

Logon Session Events		
4624	Successful logon	Correlate by
4647	User initiated logoff Logon ID	
4625	Logon failure (See Logon Failure Codes)	
4778	Remote desktop session reconnected	
4779	Remote desktop session disconnected	
4800	Workstation locked	
4801	Workstation unlocked	
4802	Screen saver invoked	
4803	Screen saver dismissed	

Logon Types	
2	Interactive
3	Network (i.e. mapped drive)
4	Batch (i.e. schedule task)
5	Service (service startup)
7	Unlock (i.e. unnattended workstation with password protected screen saver)
8	Network Cleartext (Most often indicates a logon to IIS with "basic authentication")
10	Remote Desktop
11	Logon with cached credentials

Logon Failure Codes	
0xC0000064	User name does not exist
0xC000006A	User name is correct but the password is wrong
0xC0000234	User is currently locked out
0xC0000072	Account is currently disabled
0xC000006F	User tried to logon outside his day of week or time of day restrictions
0xC0000070	Workstation restriction
0xC00000193	Account expiration
0xC0000071	Expired password
0xC0000133	Clocks between DC and other computer too far out of sync
0xC0000224	User is required to change password at next logon
0xC0000225	Evidently a bug in Windows and not a risk
0xC000015b	The user has not been granted the requested logon type (aka logon right) at this machine

Kerberos Failure Codes	
0x6	Bad user name
0x7	New computer account?
0x9	Administrator should reset password
0xC	Workstation restriction
0x12	Account disabled, expired, locked out, logon hours restriction
0x17	The user's password has expired
0x18	Bad password
0x20	Frequently logged by computer accounts
0x25	Workstation's clock too far out of sync with the DC's

Windows ID	Status Message
106	Scheduled Task Registered
1149	Remote Desktop Auth Succeeded
1102	Audit Logs Clearedz
4104	Powershell Scriptblock contents
4105	Powershell Scriptblock start
4106	Powershell Scriptblock stop
4624	Network Logons
4648	Logon Using Explicit Credentials

Windows ID	Status Message
4688	New Process
4689	Process Exit
4698	Scheduled Task Created
4699	Scheduled Task Deleted
4702	Scheduled Task Updated
7035	Service Start / Stop Control
7036	Service Running / Stopped
7045	Service Installation

Windows ID	Status Message
4688	Process Execution
4663	File monitoring
4100	PowerShell logs
5861	WMI
5140/5145	Share connection
4657	Windows Registry
5140/5145	Net Shares
5156	Firewall Logs
4624	Authentication logs
4689	Process Term
4616	The system time was changed.
4725	A user account was disabled
4720	A user account was created
4726	A user account was deleted
5025	The Windows Firewall Service has been stopped
4740	A user account was locked out

Windows ID	Status Message
1100	The event logging service has shut down
1101	Audit events have been dropped by the transport
1102	The audit log was cleared
1104	The security Log is now full
1105	Event log automatic backup
1108	The event logging service encountered an error
4608	Windows is starting up
4609	Windows is shutting down
4610	An authentication package has been loaded by the Local Security Authority
4611	A trusted logon process has been registered with the Local Security Authority
4612	Internal resources allocated for the queuing of audit messages have been exhausted, leading to the loss of some audits.
4614	A notification package has been loaded by the Security Account Manager
4615	Invalid use of LPC port
4616	The system time was changed.
4618	A monitored security event pattern has occurred
4621	Administrator recovered system from CrashOnAuditFail
4622	A security package has been loaded by the Local Security Authority.
4624	An account was successfully logged on
4625	An account failed to log on
4626	User/Device claims information

Windows ID	Status Message
4627	Group membership information.
4634	An account was logged off
4646	IKE DoS-prevention mode started
4647	User initiated logoff
4648	A logon was attempted using explicit credentials
4649	A replay attack was detected
4650	An IPsec Main Mode security association was established
4651	An IPsec Main Mode security association was established
4652	An IPsec Main Mode negotiation failed
4653	An IPsec Main Mode negotiation failed
4654	An IPsec Quick Mode negotiation failed
4655	An IPsec Main Mode security association ended
4656	A handle to an object was requested
4657	A registry value was modified
4658	The handle to an object was closed
4659	A handle to an object was requested with intent to delete
4660	An object was deleted
4661	A handle to an object was requested
4662	An operation was performed on an object
4663	An attempt was made to access an object
4664	An attempt was made to create a hard link
4665	An attempt was made to create an application client context.

Windows ID	Status Message
4666	An application attempted an operation
4667	An application client context was deleted
4668	An application was initialized
4670	Permissions on an object were changed
4671	An application attempted to access a blocked ordinal through the TBS
4672	Special privileges assigned to new logon
4673	A privileged service was called
4674	An operation was attempted on a privileged object
4675	SIDs were filtered
4688	A new process has been created
4689	A process has exited
4690	An attempt was made to duplicate a handle to an object
4691	Indirect access to an object was requested
4692	Backup of data protection master key was attempted
4693	Recovery of data protection master key was attempted
4694	Protection of auditable protected data was attempted
4695	Unprotection of auditable protected data was attempted
4696	A primary token was assigned to process
4697	A service was installed in the system
4698	- A scheduled task was created
4699	A scheduled task was deleted

Windows ID	Status Message
4700	A scheduled task was enabled
4701	A scheduled task was disabled
4702	A scheduled task was updated
4703	A token right was adjusted
4704	A user right was assigned
4705	A user right was removed
4706	A new trust was created to a domain
4707	A trust to a domain was removed
4709	IPsec Services was started
4710	IPsec Services was disabled
4711	PAStore Engine (1%)
4712	IPsec Services encountered a potentially serious failure
4713	Kerberos policy was changed
4714	Encrypted data recovery policy was changed
4715	The audit policy (SACL) on an object was changed
4716	Trusted domain information was modified
4717	System security access was granted to an account
4718	System security access was removed from an account
4719	System audit policy was changed
4720	A user account was created
4722	A user account was enabled
4723	An attempt was made to change an account's password
4724	An attempt was made to reset an accounts password
4725	A user account was disabled
4726	A user account was deleted
4727	A security-enabled global group was created
4728	A member was added to a security-enabled global group
4729	A member was removed from a securityenabled global group
4730	A security-enabled global group was deleted
4731	A security-enabled local group was created

Windows ID	Status Message
4732	A member was added to a security-enabled local group
4733	A member was removed from a securityenabled local group
4734	A security-enabled local group was deleted
4735	A security-enabled local group was changed
4737	A security-enabled global group was changed
4738	A user account was changed
4739	Domain Policy was changed
4740	A user account was locked out
4741	A computer account was created
4742	A computer account was changed
4743	A computer account was deleted
4744	A security-disabled local group was created
4745	A security-disabled local group was changed
4746	A member was added to a security-disabled local group
4747	A member was removed from a securitydisabled local group
4748	A security-disabled local group was deleted
4749	A security-disabled global group wascreated
4750	A security-disabled global group was changed
4751	A member was added to a security-disabled global group
4752	A member was removed from a securitydisabled global group
4753	A security-disabled global group was deleted
4754	A security-enabled universal group was created
4755	A security-enabled universal group was changed
4756	A member was added to a security-enabled universal group
4757	A member was removed from a securityenabled universal group
4758	A security-enabled universal group was deleted
4759	A security-disabled universal group was created
4760	A security-disabled universal group was changed
4761	A member was added to a security-disabled universal group

Windows ID	Status Message
4762	A member was removed from a securitydisabled universal group
4763	A security-disabled universal group was deleted
4764	A groups type was changed
4765	SID History was added to an account
4766	An attempt to add SID History to an account failed
4767	A user account was unlocked
4768	A Kerberos authentication ticket (TGT) was requested
4769	A Kerberos service ticket was requested
4770	A Kerberos service ticket was renewed
4771	Kerberos pre-authentication failed
4772	A Kerberos authentication ticket request failed
4773	A Kerberos service ticket request failed
4774	An account was mapped for logon
4775	An account could not be mapped for logon
4776	The domain controller attempted to validate the credentials for an account
4777	The domain controller failed to validate the credentials for an account
4778	A session was reconnected to a Window Station
4779	A session was disconnected from a Window Station
4780	The ACL was set on accounts which are members of administrators groups
4781	The name of an account was changed
4782	The password hash an account was accessed
4783	A basic application group was created
4784	A basic application group was changed
4785	A member was added to a basic application group
4786	A member was removed from a basic application group
4787	A non-member was added to a basic application group
4788	A non-member was removed from a basic application group..
4789	A basic application group was deleted
4790	An LDAP query group was created
4791	A basic application group was changed
4800	The workstation was locked

Windows ID	Status Message
4801	The workstation was unlocked
4802	The screen saver was invoked
4803	The screen saver was dismissed
4816	RPC detected an integrity violation while decrypting an incoming message
4817	Auditing settings on object were changed.
4818	Proposed Central Access Policy does not grant the same access permissions as the current Central Access Policy
4819	Central Access Policies on the machine have been changed
4820	A Kerberos Ticket-granting-ticket (TGT) was denied because the device does not meet the access control restrictions
4821	A Kerberos service ticket was denied because the user, device, or both does not meet the access control restrictions
4822	NTLM authentication failed because the account was a member of the Protected User group
4823	NTLM authentication failed because access control restrictions are required
4824	Kerberos preauthentication by using DES or RC4 failed because the account was a member of the Protected User group
4825	A user was denied the access to Remote Desktop. By default, users are allowed to connect only if they are members of the Remote Desktop Users group or Administrators group
4826	Boot Configuration Data loaded
4830	SID History was removed from an account
4864	A namespace collision was detected
4865	A trusted forest information entry was added
4866	A trusted forest information entry was removed
4867	A trusted forest information entry was modified
4868	The certificate manager denied a pending certificate request
4869	Certificate Services received a resubmitted certificate request
4870	Certificate Services revoked a certificate
4871	Certificate Services received a request to publish the certificate revocation list (CRL)

Windows ID	Status Message
4872	Certificate Services published the certificate revocation list (CRL)
4873	A certificate request extension changed
4874	One or more certificate request attributes changed.
4875	Certificate Services received a request to shut down
4876	Certificate Services backup started
4877	Certificate Services backup completed
4878	Certificate Services restore started
4879	Certificate Services restore completed
4880	Certificate Services started
4881	Certificate Services stopped
4882	The security permissions for Certificate Services changed
4883	Certificate Services retrieved an archived key
4884	Certificate Services imported a certificate into its database
4885	The audit filter for Certificate Services changed
4886	Certificate Services received a certificate request
4887	Certificate Services approved a certificate request and issued a certificate
4888	Certificate Services denied a certificate request
4889	Certificate Services set the status of a certificate request to pending
4890	The certificate manager settings for Certificate Services changed.
4891	A configuration entry changed in Certificate Services
4892	A property of Certificate Services changed
4893	Certificate Services archived a key
4894	Certificate Services imported and archived a key
4895	Certificate Services published the CA certificate to Active Directory Domain Services
4896	One or more rows have been deleted from the certificate database
4897	Role separation enabled
4898	Certificate Services loaded a template
4899	A Certificate Services template was updated
4900	Certificate Services template security was updated
4905	An attempt was made to unregister a security event source

Windows ID	Status Message
4906	The CrashOnAuditFail value has changed
4907	Auditing settings on object were changed
4908	Special Groups Logon table modified
4909	The local policy settings for the TBS were changed
4910	The group policy settings for the TBS were changed
4911	Resource attributes of the object were changed
4952	Parts of a rule have been ignored because its minor version number was not recognized by Windows Firewall
4953	A rule has been ignored by Windows Firewall because it could not parse the rule
4954	Windows Firewall Group Policy settings has changed. The new settings have been applied
4956	Windows Firewall has changed the active profile
4957	Windows Firewall did not apply the following rule
4958	Windows Firewall did not apply the following rule because the rule referred to items not configured on this computer
4960	IPsec dropped an inbound packet that failed an integrity check
4961	IPsec dropped an inbound packet that failed a replay check
4962	IPsec dropped an inbound packet that failed a replay check
4963	IPsec dropped an inbound clear text packet that should have been secured
4964	Special groups have been assigned to a new logon
4965	IPsec received a packet from a remote computer with an incorrect Security Parameter Index (SPI).
4976	During Main Mode negotiation, IPsec received an invalid negotiation packet.
4977	During Quick Mode negotiation, IPsec received an invalid negotiation packet.
4978	During Extended Mode negotiation, IPsec received an invalid negotiation packet.
4979	IPsec Main Mode and Extended Mode security associations were established.
4980	IPsec Main Mode and Extended Mode security associations were established

Windows ID	Status Message
4981	IPsec Main Mode and Extended Mode security associations were established
4982	IPsec Main Mode and Extended Mode security associations were established
4983	An IPsec Extended Mode negotiation failed
4984	An IPsec Extended Mode negotiation failed
4985	The state of a transaction has changed
5024	The Windows Firewall Service has started successfully
5025	The Windows Firewall Service has been stopped
5027	The Windows Firewall Service was unable to retrieve the security policy from the local storage
5028	The Windows Firewall Service was unable to parse the new security policy.
5029	The Windows Firewall Service failed to initialize the driver
5030	The Windows Firewall Service failed to start
5031	The Windows Firewall Service blocked an application from accepting incoming connections on the network.
5032	Windows Firewall was unable to notify the user that it blocked an application from accepting incoming connections on the network
5033	The Windows Firewall Driver has started successfully
5034	The Windows Firewall Driver has been stopped
5035	The Windows Firewall Driver failed to start
5037	The Windows Firewall Driver detected critical runtime error. Terminating
5038	Code integrity determined that the image hash of a file is not valid
5039	A registry key was virtualized.
5040	A change has been made to IPsec settings.An Authentication Set was added.
5041	A change has been made to IPsec settings.An Authentication Set was modified
5042	A change has been made to IPsec settings.An Authentication Set was deleted
5043	A change has been made to IPsec settings.A Connection Security Rule was added
5044	A change has been made to IPsec settings.A Connection Security Rule was modified
5045	A change has been made to IPsec settings.A Connection Security Rule was deleted
5046	A change has been made to IPsec settings.A Crypto Set was added
5047	A change has been made to IPsec settings.A Crypto Set was modified
5048	A change has been made to IPsec settings.A Crypto Set was deleted
5049	An IPsec Security Association was deleted
5050	An attempt to programmatically disable theWindows Firewall using a call to INetFwProfile.FirewallEnabled(FALSE
5051	A file was virtualized
5056	A cryptographic self test was performed
5057	A cryptographic primitive operation failed
5058	Key file operation
5059	Key migration operation
5060	Verification operation failed
5061	Cryptographic operation
5062	A kernel-mode cryptographic self test was performed
5063	A cryptographic provider operation was attempted
5064	A cryptographic context operation was attempted
5065	A cryptographic context modification was attempted
5066	A cryptographic function operation was attempted
5067	A cryptographic function modification was attempted
5068	A cryptographic function provider operation was attempted
5069	A cryptographic function property operation was attempted
5070	A cryptographic function property operation was attempted
5071	Key access denied by Microsoft key distribution service
5120	OCSP Responder Service Started
5121	OCSP Responder Service Stopped
5122	A configuration entry changed in the OCSP Responder Service
5123	A configuration entry changed in the OCSP Responder Service
5124	A security setting was updated on OCSP Responder Service
5125	A request was submitted to OCSP Responder Service
5126	Signing Certificate was automatically updated by the OCSP Responder Service
5127	The OCSP Revocation Provider successfully updated the revocation information
5136	A directory service object was modified
5137	A directory service object was created
5138	A directory service object was undeleted
5139	A directory service object was moved
5140	A network share object was accessed
5141	A directory service object was deleted
5142	A network share object was added.
5143	A network share object was modified
5144	A network share object was deleted.
5145	A network share object was checked to see whether client can be granted desired access
5146	The Windows Filtering Platform has blocked a packet
5147	A more restrictive Windows Filtering Platform filter has blocked a packet
5148	The Windows Filtering Platform has detected a DoS attack and entered a defensive mode
5149	The DoS attack has subsided and normal processing is being resumed.
5150	The Windows Filtering Platform has blocked a packet.
5151	A more restrictive Windows Filtering Platform filter has blocked a packet.
5152	The Windows Filtering Platform blocked a packet
5153	A more restrictive Windows Filtering Platform filter has blocked a packet
5154	The Windows Filtering Platform has permitted an application or service to listen on a port for incoming connections

https://linuxlearnmore.com

Windows ID	Status Message
5155	The Windows Filtering Platform has blocked an application or service from listening on a port for incoming connections
5156	The Windows Filtering Platform has allowed a connection
5157	The Windows Filtering Platform has blocked a connection
5158	The Windows Filtering Platform has permitted a bind to a local port
5159	The Windows Filtering Platform has blocked a bind to a local port
5168	Spn check for SMB/SMB2 fails.
5169	A directory service object was modified
5170	A directory service object was modified during a background cleanup task
5376	Credential Manager credentials were backed up
5377	Credential Manager credentials were restored from a backup
5378	The requested credentials delegation was disallowed by policy
5379	Credential Manager credentials were read
5380	Vault Find Credential
5381	Vault credentials were read
5382	Vault credentials were read
5440	The following callout was present when the Windows Filtering Platform Base Filtering Engine started
5441	The following filter was present when the Windows Filtering Platform Base Filtering Engine started
5442	The following provider was present when the Windows Filtering Platform Base Filtering Engine started
5443	The following provider context was present when the Windows Filtering Platform Base Filtering Engine started
5444	The following sub-layer was present when the Windows Filtering Platform Base Filtering Engine started
5446	A Windows Filtering Platform callout has been changed
5447	A Windows Filtering Platform filter has been changed

Windows ID	Status Message
5448	A Windows Filtering Platform provider has been changed
5449	A Windows Filtering Platform provider context has been changed
5450	A Windows Filtering Platform sub-layer has been changed
5451	An IPsec Quick Mode security association was established
5452	An IPsec Quick Mode security association ended
5453	An IPsec negotiation with a remote computer failed because the IKE and AuthIP IPsec Keying Modules (IKEEXT) service is not started
5456	PAStore Engine applied Active Directory storage IPsec policy on the computer
5457	PAStore Engine failed to apply Active Directory storage IPsec policy on the computer
5458	PAStore Engine applied locally cached copy of Active Directory storage IPsec policy on the computer
5459	PAStore Engine failed to apply locally cached copy of Active Directory storage IPsec policy on the computer
5460	PAStore Engine applied local registry storage IPsec policy on the computer
5461	PAStore Engine failed to apply local registry storage IPsec policy on the computer
5462	PAStore Engine failed to apply some rules of the active IPsec policy on the computer
5463	PAStore Engine polled for changes to the active IPsec policy and detected no changes
5464	PAStore Engine polled for changes to the active IPsec policy, detected changes, and applied them to IPsec Services
5465	PAStore Engine received a control for forced reloading of IPsec policy and processed the control successfully
5466	PAStore Engine polled for changes to the Active Directory IPsec policy, determined that Active Directory cannot be reached, and will use the cached copy of the Active Directory IPsec policy instead
5467	PAStore Engine polled for changes to the Active Directory IPsec policy, determined that Active Directory can be reached, and found no changes to the policy

Windows ID	Status Message
5468	PAStore Engine polled for changes to the Active Directory IPsec policy, determined that Active Directory can be reached, found changes to the policy, and applied those changes
5471	PAStore Engine loaded local storage IPsec policy on the computer
5472	PAStore Engine failed to load local storage IPsec policy on the computer
5473	PAStore Engine loaded directory storage IPsec policy on the computer
5474	PAStore Engine failed to load directory storage IPsec policy on the computer
5477	PAStore Engine failed to add quick mode filter
5478	IPsec Services has started successfully
5479	IPsec Services has been shut down successfully
5480	IPsec Services failed to get the complete list of network interfaces on the computer
5483	IPsec Services failed to initialize RPC server. IPsec Services could not be started
5484	IPsec Services has experienced a critical failure and has been shut down
5485	IPsec Services failed to process some IPsec filters on a plug-and-play event for network interfaces
5632	A request was made to authenticate to a wireless network
5633	A request was made to authenticate to a wired network
5712	A Remote Procedure Call (RPC) was attempted
5888	An object in the COM+ Catalog wasmodified
5889	An object was deleted from the COM+Catalog
5890	An object was added to the COM+ Catalog
6144	Security policy in the group policy objects has been applied successfully
6145	One or more errors occured while processing security policy in the group policy objects
6272	Network Policy Server granted access to a user

Windows ID	Status Message
6273	Network Policy Server denied access to a user
6274	Network Policy Server discarded the request for a user
6275	Network Policy Server discarded the accounting request for a user
6276	Network Policy Server quarantined a user
6277	Network Policy Server granted access to a user but put it on probation because the host did not meet the defined health policy
6278	Network Policy Server granted full access to a user because the host met the defined health policy
6279	Network Policy Server locked the user account due to repeated failed authentication attempts
6280	Network Policy Server unlocked the user account
6281	Code Integrity determined that the page hashes of an image file are not valid...
6400	BranchCache: Received an incorrectly formatted response while discovering availability of content.
6401	BranchCache: Received invalid data from a peer. Data discarded.
6402	BranchCache: The message to the hosted cache offering it data is incorrectly formatted.
6403	BranchCache: The hosted cache sent an incorrectly formatted response to the client's message to offer it data.
6404	BranchCache: Hosted cache could not be authenticated using the provisioned SSL certificate.
6405	BranchCache: %2 instance(s) of event id %1 occurred.
6406	%1 registered to Windows Firewall to control filtering for the following:
6407	1%
6408	Registered product %1 failed and Windows Firewall is now controlling the filtering for %2.
6409	BranchCache: A service connection point object could not be parsed

Windows ID	Status Message
6410	Code integrity determined that a file does not meet the security requirements to load into a process. This could be due to the use of shared sections or other issues
6416	A new external device was recognized by the system.
6417	The FIPS mode crypto selftests succeeded
6418	The FIPS mode crypto selftests failed
6419	A request was made to disable a device
6420	A device was disabled
6421	A request was made to enable a device
6422	A device was enabled
6423	The installation of this device is forbidden by system policy
6424	The installation of this device was allowed,after having previously been forbidden by policy
8191	Highest System-Defined Audit Message Value

Status Codes

List of HTTP status codes

When accessing a web server or application, every HTTP request that is received by a server is responded to with an HTTP status code. HTTP status codes are three-digit codes, and are grouped into five different classes. The class of a status code can be quickly identified by its first digit:with more details.

1xx–Informational

This class of status code indicates a provisional response, consisting only of the Status-Line and optional headers, and is terminated by an empty line

100–Continue
101–Switching Protocols
102–Processing

2xx–Successful

This class of status code indicates that the client's request was successfully received, understood, and accepted.

200–OK
201–Created
202–Accepted
203–Non-Authoritative Information
204–No Content
205–Reset Content
206–Partial Content
207–Multi-Status

3xx–Redirection

This class of status code indicates that further action needs to be taken by the user agent in order tofulfillthe request.

300–Multiple Choices
301–Moved Permanently
302–Found
303–See Other
304–Not Modified
305–Use Proxy
307–Temporary Redirect

4xx–Client Error

The 4xx class of status code is intended for cases in which the client seems to have erred.

400–Bad Request
401–Unauthorised
402–Payment Required
403–Forbidden
404–Not Found
405–Method Not Allowed
406–Not Acceptable
407–Proxy Authentication Required
408–Request Timeout
409–Conflict
410–Gone
411–Length Required
412–Precondition Failed
413–Request Entity Too Large
414–Request URI Too Long
415–Unsupported Media Type
416–Requested Range Not Satisfiable
417–Expectation Failed
422–UnprocessableEntity
423–Locked
424–Failed Dependency
425–Unordered Collection
426–Upgrade Required

5xx–Server Error

Response status codes beginning with the digit "5"indicate cases in which the server is aware that it has erred or is incapable of performing the request

500–Internal Server Error
501–Not Implemented
502–Bad Gateway
503–Service Unavailable
504–Gateway Timeout
505–HTTP Version Not Supported
506–Variant Also Negotiates
507–Insufficient Storage
510–Not Extended

Key	Description
white	HTTP version 1.0
Blue	HTTP version 1.1
Aqua	Extension RFC 2295
Green	ExtensionRFC 2518
Yellow	Extension RFC 2774
Orange	Extension RFC 2817
Purple	Extension RFC 3648
Red	Extension RFC 4918

DNS Response Code

The following table explains the DNS return codes that can be returned when doing a DNS query and may appear in your logs. Each return code has its own purpose in the DNS infrastructure. Typically, you'll see **NOERROR (RCODE:0) when doing most of your successful browsing**, all of the other return codes are consider errors.

DNS Return Message	DNS Response Code	Function
NOERROR	RCODE:0	DNS Query completed successfully
FORMERR	RCODE:1	DNS Query Format Error
SERVFAIL	RCODE:2	Server failed to complete the DNS request
NXDOMAIN	RCODE:3	Domain name does not exist.
NOTIMP	RCODE:4	Function not implemented
REFUSED	RCODE:5	The server refused to answer for the query
YXDOMAIN	RCODE:6	Name that should not exist, does exist
XRRSET	RCODE:7	RRset that should not exist, does exist
NOTAUTH	RCODE:8	Server not authoritative for the zone
NOTZONE	RCODE:9	Name not in zone

Domain Name System (DNS)

Type	Description	Function
A	Address Record	Returns a 32-bit IPv4 address, most commonly used to map hostnames to an IP address of the host,but it is also used for DNSBLs, storing subnet masks in RFC 1101, etc.
CNAME	Canonical Name Record	Alias of one name to another: the DNS lookup will continue by retrying the lookup with the new name.
MX	Mail Exchange Record	Maps a domain name to a list of message transfer agents for that domain
AAAA	IPv6 Address Record	Returns a 128-bit IPv6 address, most commonly used to map hostnames to an IP address of the host.
TXT	Text Record	Originally for arbitrary human-readable text in a DNS record. Since the early 1990s, however, this record more often carriesmachine-readable data, such as specified by RFC 1464, opportunistic encryption, Sender Policy Framework, DKIM, DMARC, DNS-SD, etc.
PTR	Pointer Record	Pointer to a canonical name. Unlike a CNAME, DNS processing stops and just the name is returned. The most common use is for implementing reverse DNS lookups, but other uses include such things as DNS-SD.
SRV	Service locator	Generalized service location record, used for newer protocols instead of creating protocol-specific records such as MX.
SPF	Sender Policy Framework	SPF(99) (from RFC 4408) was specified as part of the Sender Policy Framework protocol as an alternative to storing SPF data in TXT records, using the same format. It was later found that the majority of SPF deployments lack proper support for this record type, and support for it was discontinued in RFC 7208.
NS	Name Server record	Delegates a DNS zone to use the given authoritative name servers
SOA	Start of [a zone of] Authority Record	Specifies authoritative information about a DNS zone, including the primary name server, the email of the domain administrator, the domain serial number, and several timers relating to refreshing the zone.

IIS Events and Error Messages

Internet Information Services (IIS, formerly Internet Information Server) is an extensible web server software created by Microsoft for use with the Windows NT family.[2] IIS supports HTTP, HTTP/2, HTTPS, FTP, FTPS, SMTP and NNTP. It has been an integral part of the Windows NT family since Windows NT 4.0, though it may be absent from some editions (e.g. Windows XP Home edition), and is not active by default.

W3C

Windows ID	Status Message IIS
Date (date)	the date on which the request occurred.
Time (time)	the time, in Coordinated Universal Time (UTC), at which the request occurred.
Client IP Address (c-ip)	the IP address of the client that made the request.
User Name (cs-username)	the name of the authenticated user who accessed your server. Anonymous users are indicated by a hyphen.
Service Name (s-sitename)	the site instance number that fulfilled the request.
Server Name (s-computername)	the name of the server on which the log file entry was generated.
Server IP Address (s-ip)	the IP address of the server on which the log file entry was generated.
Server Port (s-port)	the server port number that is configured for the service.
Method (cs-method)	the requested action, for example, a GET method.
URI Stem (cs-uri-stem)	the Universal Resource Identifier, or target, of the action.
URI Query (cs-uri-query)	the query, if any, that the client was trying to perform. A Universal Resource Identifier (URI) query is necessary only for dynamic pages.
Protocol Status (sc-status)	the HTTP or FTP status code.
Protocol Sub-status (sc-substatus)	the HTTP or FTP substatus code.
Win32 Status (sc-win32-status)	the Windows status code.
Bytes Sent (sc-bytes)	the number of bytes that the server sent.
Bytes Received (cs-bytes)	the number of bytes that the server received.
Time Taken (time-taken)	the length of time that the action took in milliseconds.
Protocol Version (cs-version)	the protocol version that the client used.
Host (cs-host)	the host name, if any.
User Agent (cs(UserAgent))	the browser type that the client used.
Cookie (cs(Cookie))	the content of the cookie sent or received, if any.
Referrer (cs(Referrer))	the site that the user last visited. This site provided a link to the current site.

%SystemDrive%\inetpub\logs\LogFiles

Schedule: to create new log file that is based on one of the following values:
Hourly: a new log file is created each hour.
Daily: a new log file is created each day.
Weekly: a new log file is created each week.
Monthly: a new log file is created each month

•**Maximum file size (in bytes)**: to create a log file when the file reaches a certain size (in bytes).The minimum file size is 1048576 bytes. If this attribute is set to a value less than 1048576 bytes, the default value is implicitly assumed as 1048576 bytes.
•**Do not create a new log file**: there is a single log file that continues to grow as information is logged.

Reference

Reference:

SANS Information Security White Papers

https://www.sans.org/white-papers/?focus-area=digital-forensics&msc=dfir-lp

https://attack.mitre.org/techniques/T1021/

https://attack.mitre.org/tactics/TA0008/

https://www.ultimatewindowssecurity.com/securitylog/quickref/Default.aspx

Thank you !