

OFFENSIVE SECURITY

Credential Access & Dumping

Lateral Movement

WinRM for Lateral Movement

WinRS for Lateral Movement

WMI for Lateral Movement

RDP Hijacking for Lateral Movement with tscon

Shared Webroot

Lateral Movement via DCOM

WMI + MSI Lateral Movement

Lateral Movement via Service Configuration Manager

Lateral Movement via SMB Relaying

WMI + NewScheduledTaskAction Lateral Movement

WMI + PowerShell Desired State Configuration Lateral Movement

Simple TCP Relaying with NetCat

Empire Shells with NetNLTMv2 Relaying

Lateral Movement with Psexec

From Beacon to Interactive RDP Session

SSH Tunnelling / Port Forwarding

Lateral Movement via WMI Event Subscription

Lateral Movement via DLL Hijacking

Lateral Movement over headless RDP with SharpRDP

Man-in-the-Browser via Chrome Extension

ShadowMove: Lateral Movement by Duplicating Existing Sockets

Persistence

Exfiltration

REVERSING, FORENSICS & MISC

Internals


Cloud

Neo4j

Dump Virtual Box Memory

AES Encryption Using Crypto++ .lib in Visual Studio C++

Reversing Password Checking Routine

Powered by GitBook

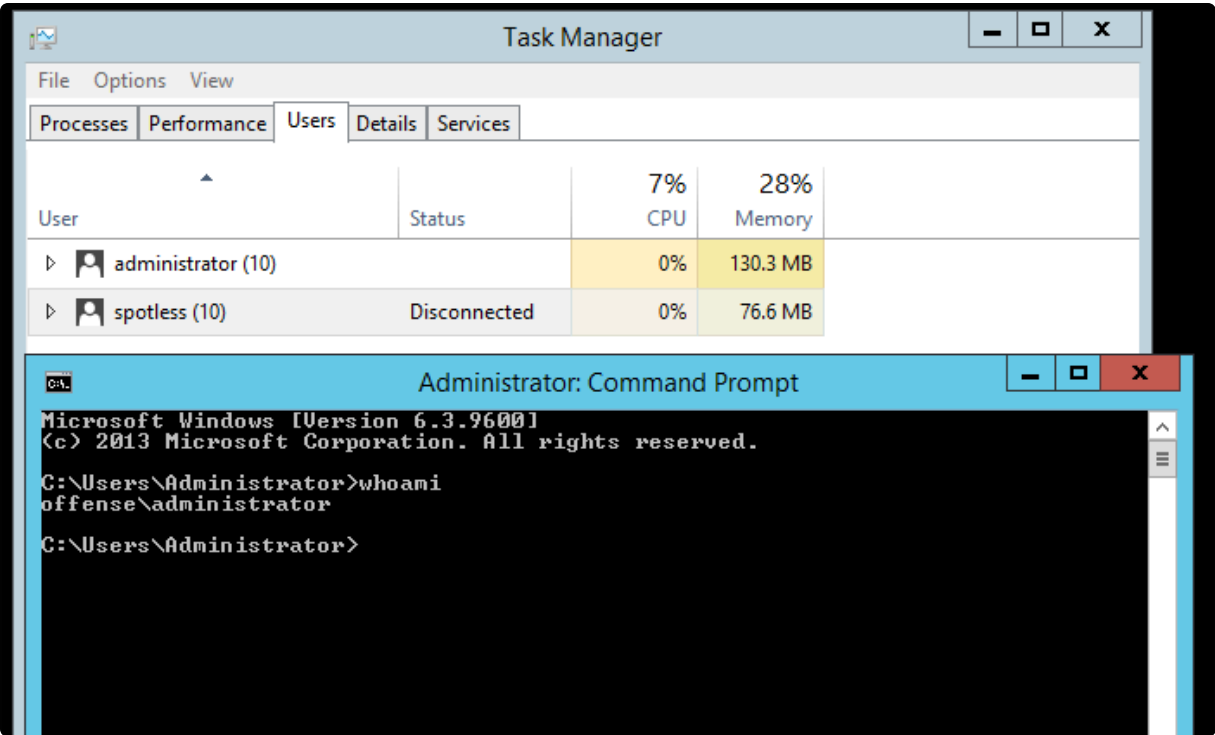
RDP Hijacking for Lateral Movement with tscon

This lab explores a technique that allows a SYSTEM account to move laterally through the network using RDP without the need for credentials.

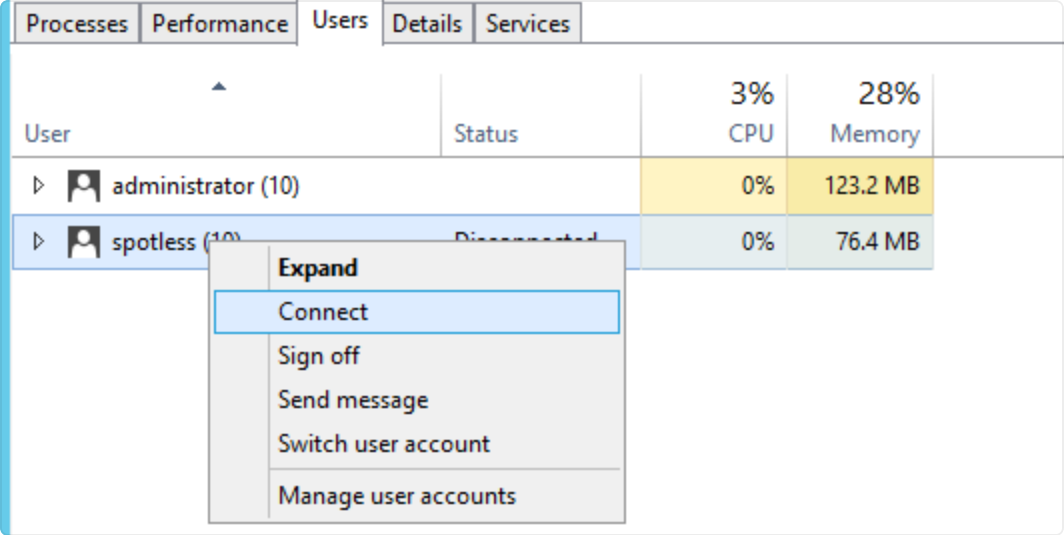
Execution

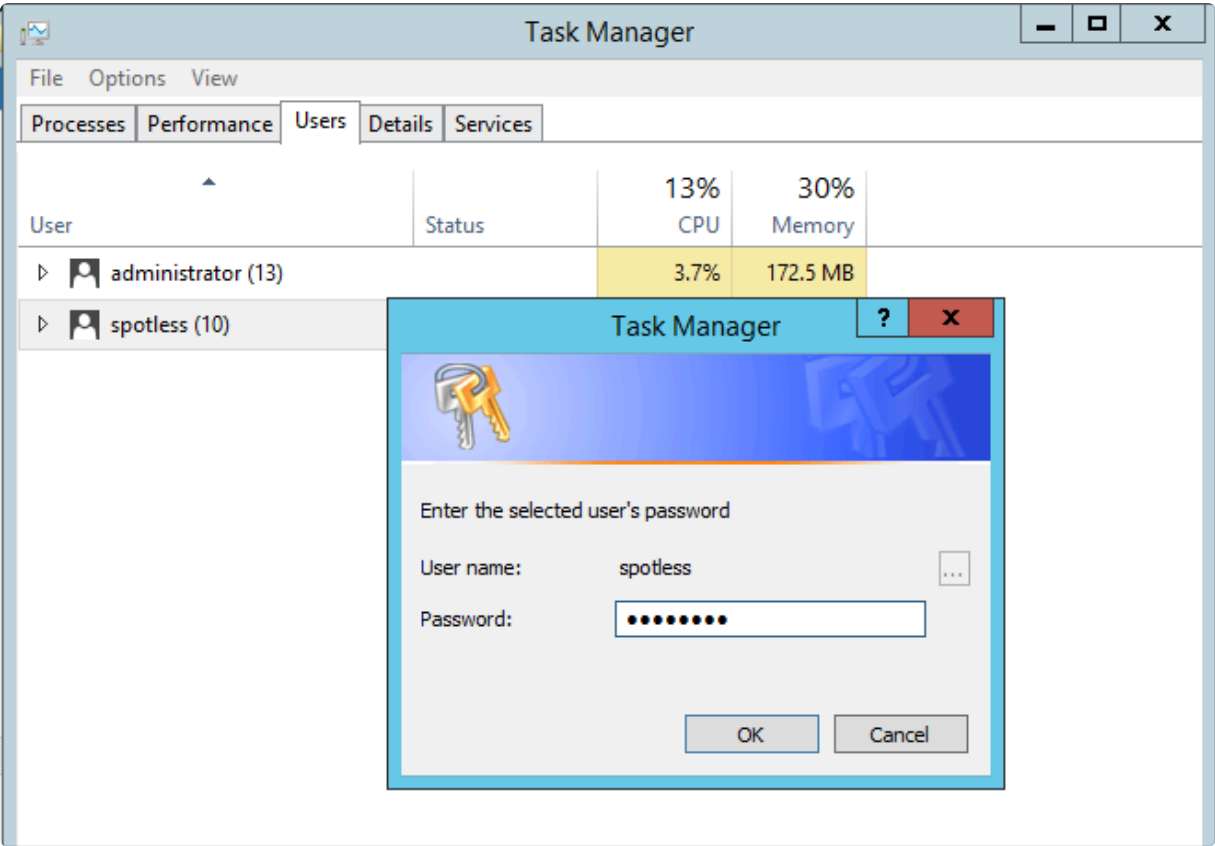
It is possible by design to switch from one user's desktop session to another through the Task Manager (one of the ways).

Below shows that there are two users on the system and currently the administrator session is in active:

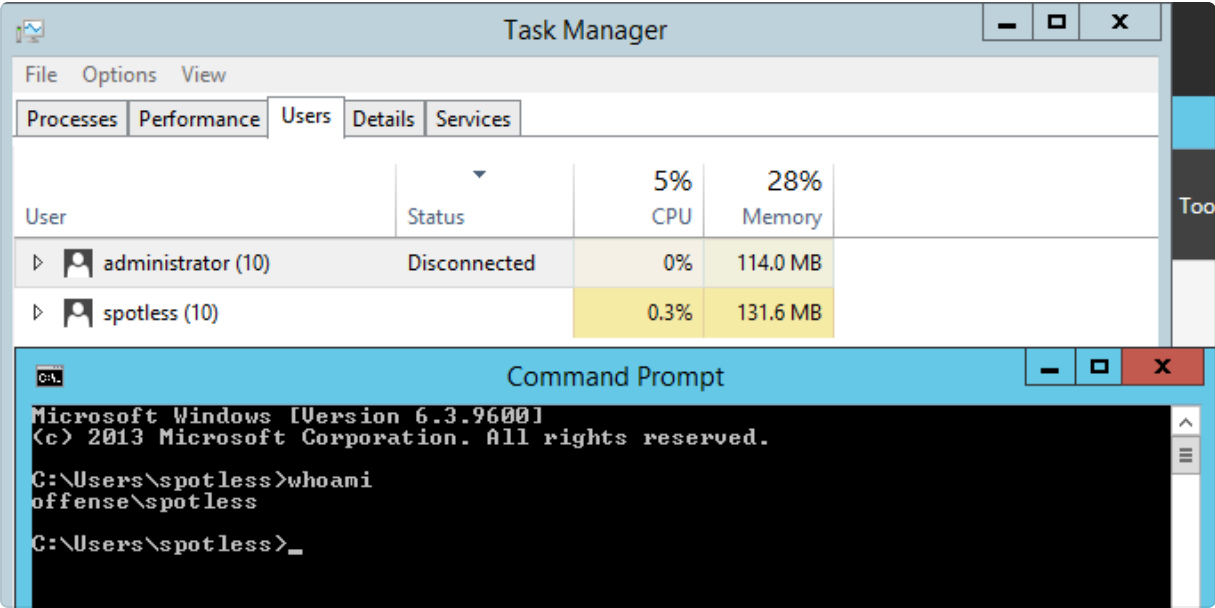


Let's switch to the `spotless` session - this requires knowing the user's password, which for this exercise is known, so lets enter it:



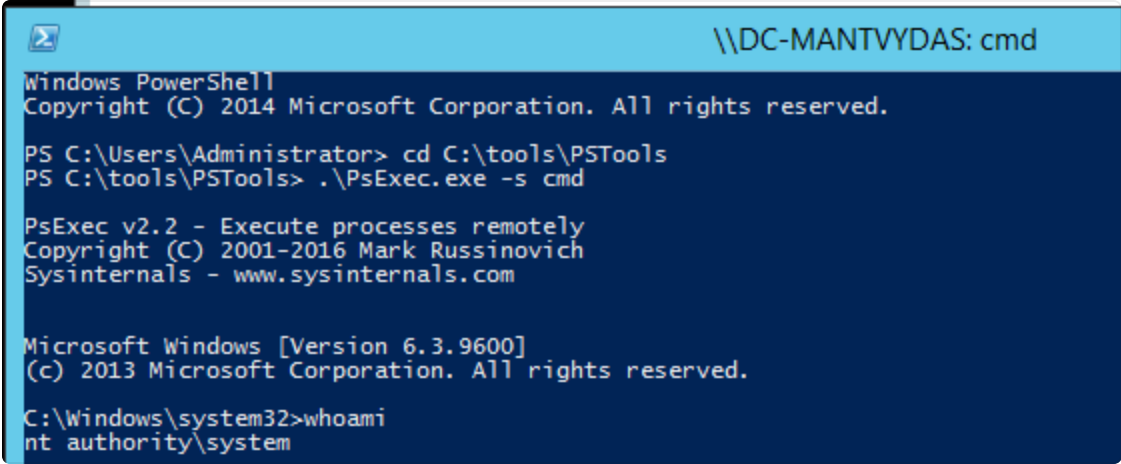


We are now reconnected to the `spotless` session:

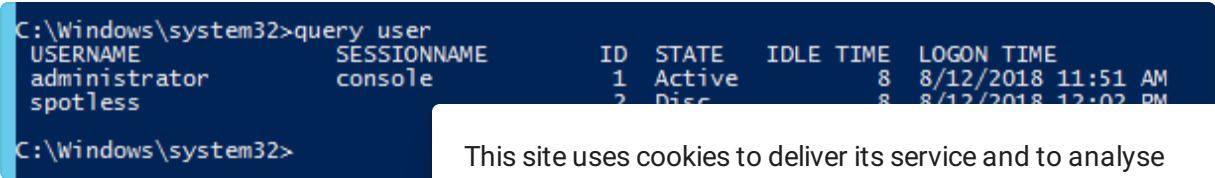


Now this is where it gets interesting. It is possible to reconnect to a users session without knowing their password if you have `SYSTEM` level privileges on the system. Let's elevate to `SYSTEM` using psexec (privilege escalation exploits, service creation or any other technique will also do):

```
psexec -s cmd
```



Enumerate available sessions on the host with `query user`:



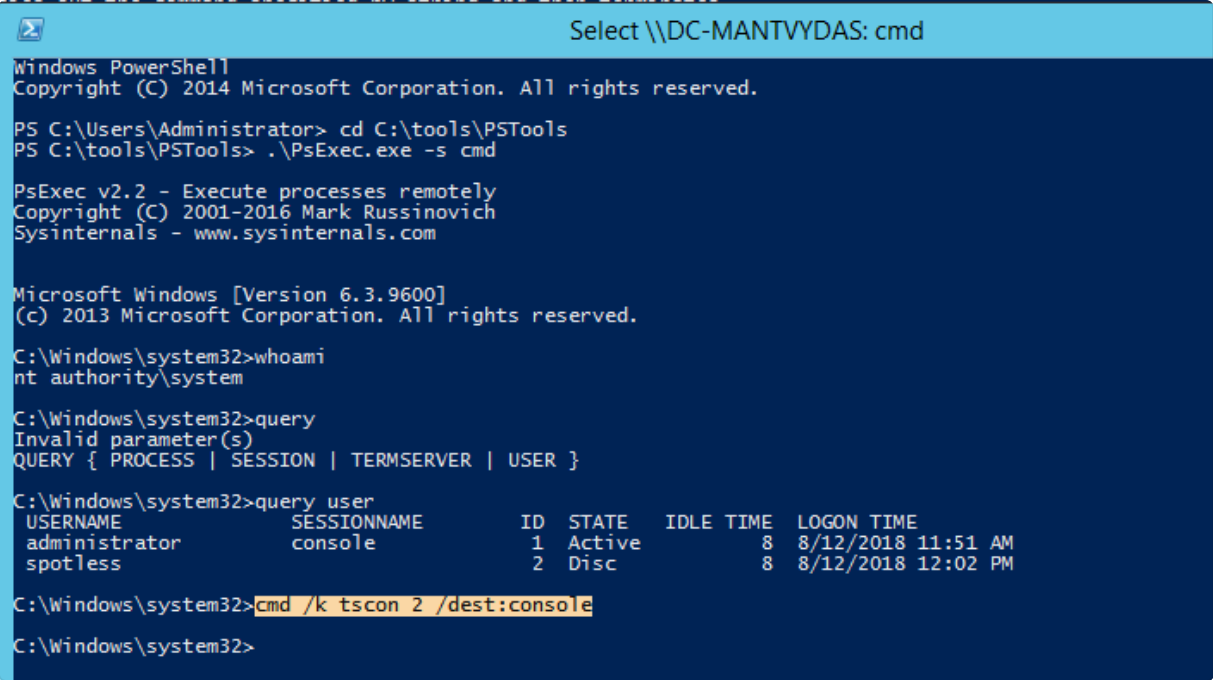
Switch to the `spotless` session
the native windows binary `tscon`

This site uses cookies to deliver its service and to analyse traffic. By browsing this site, you accept the [privacy policy](#).

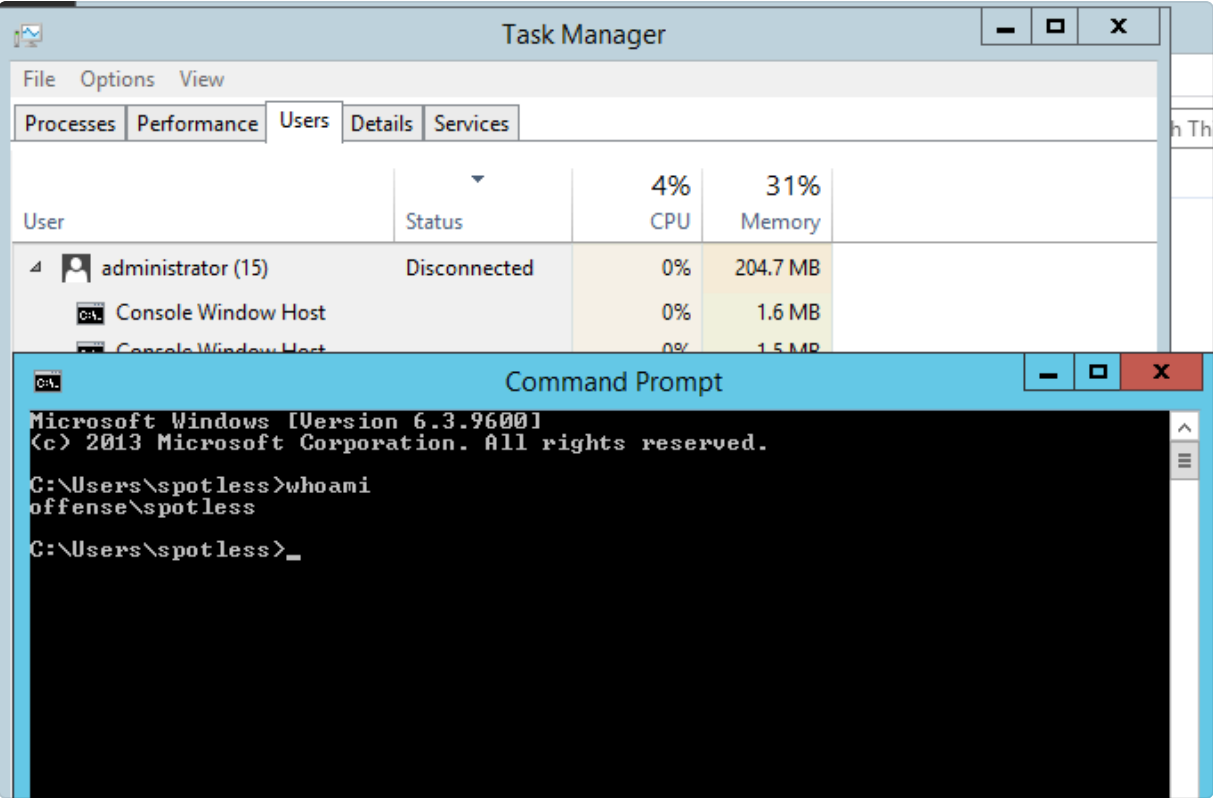
X

sessions by specifying which session ID (2 in this case for the spotless session) should be connected to which session (console in this case, where the active administator session originates from):

```
cmd /k tscon 2 /dest:console
```



Immediately after that, we are presented with the desktop session for spotless :



Observations

Looking at the logs, tscon.exe being executed as a SYSTEM user is something you may want to investigate further to make sure this is not a lateral movement attempt:

Time...	task	event_data.LogonType	event_data.CommandLine	user_name	event_id	process_id	event_data.TargetLogonId	event_data.SessionName	event_data.SubjectLogonId	event_data.LogonId
August 12th 2018, 12:17:49.802	Other Logon/Logoff Events	-	-	-	4778	572	-	Console	-	0x00000000
August 12th 2018, 12:17:49.803	Process Create (Local Process Create)	-	tscon 2 /dest:console	SYSTEM	1	1,872	-	-	-	-
August 12th 2018, 12:17:49.809	Other Logon/Logoff Events	-	-	-	4779	572	-	Console	-	0x00000000
August 12th 2018, 12:17:49.860	Process Create (Local Process Create)	-	cmd /k tscon 2 /dest:console	SYSTEM	1	1,872	-	-	-	-

Also, note how event_data.LogonID and event_ids 4778 (logon) and 4779 (logoff) events can be used to figure out which desktop sessions got disconnected/reconnected:

event_id4779

host_namedc-mantvydas

keywordsAudit Success

levelInformation

log_nameSecurity

message

Subject:

Account Name: adm

Account Domain: OFF

Logon ID: 0x2

Session:

Session Name: Con

Additional Information:

Client Name: Link

Client Address: LOC

This event is generated when a user

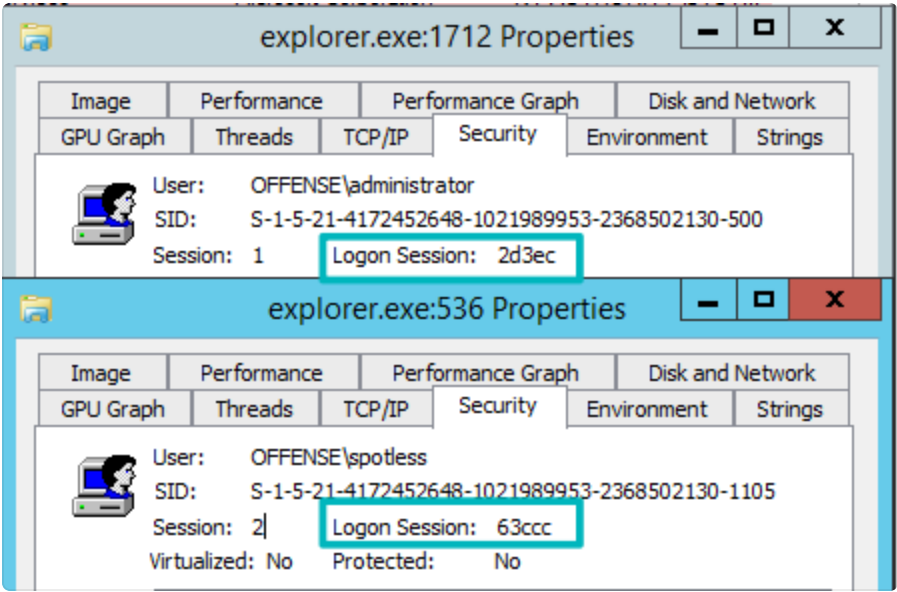
This site uses cookies to deliver its service and to analyse traffic. By browsing this site, you accept the [privacy policy](#).

Administrator session disconnected


# event_id	4,778
t host.name	dc-mantvydas
t keywords	Audit Success
t level	Information
t log_name	Security
t message	<div><div>A session was reconnected to a Window Station.</div><div><div>Subject:</div><div>Account Name:spotless</div><div>Account Domain:OFFENSE</div><div>Logon ID:0x63CCC</div></div><div><div>Session:</div><div>Session Name:Console</div></div><div><div>Additional Information:</div><div>Client Name:Unknown</div><div>Client Address:LOCAL</div></div></div> <div>This event is generated when a user reconnects to an existing Terminal Services session, or when a user switches to an existing desktop using Fast User Switching.</div>

Spotless session reconnected (hijacked)


Just reinforcing the above - note the usernames and logon session IDs:




References

 Vol de session RDP
Blog de Gentil Kiwi

Passwordless RDP Session Hijacking Feature All Windows versions

 Windows Security Log Event ID 4778 - A session was reconnected to a Window Station

 tscon
docsmsft

<

Previous

WMI for Lateral Movement

Next

Shared Webroot

>

Last updated 6 years ago