Restricted Admin mode for RDP

blog.ahasayen.com/restricted-admin-mode-for-rdp

Ammar Hasayen

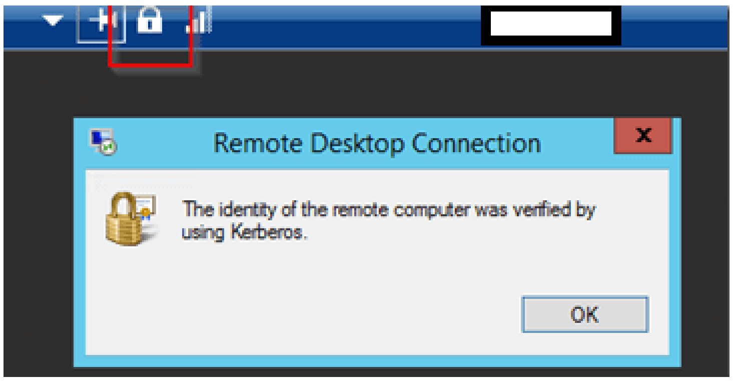
With Windows 8.1 and Windows Server 2012 R2, new security features were introduced. One of those security features is the *Restricted Admin mode for RDP*as I personally use RDP to logon to my servers and perform a lot of administrative tasks. This new security feature is introduced to mitigate the risk of pass the hash attacks.

When you connect to a remote computer using RDP, your credentials are stored on the remote computer that you RDP into. Usually you are using a powerful account to connect to remote servers, and having your credentials stored on all these computers is a security threat indeed.

Imagine that you are connecting to a Remote Desktop Server with your admin credentials using RDP, With so many other users using that server, the possibility for a malware infecting that box is high. Remote desktop servers are very tempting destination for attackers, as many users are logged on at once on such device.

with *Restricted Admin mode for RDP*, when you connect to a remote computer using the command, **mstsc.exe /RestrictedAdmin**, you will be authenticated to the remote computer, but your credentials will not be stored on that remote computer, as they would have been in the past. This means that if a malware or even a malicious user is active on that remote server, your credentials will not be available on that remote desktop server for the malware to attack.

When connecting to a remote computer using RDP and specifying the **/RestrictedAdmin** switch, the experience looks like this:



[redacted]

Remote Desktop Connection

The identity of the remote computer was verified by using Kerberos.

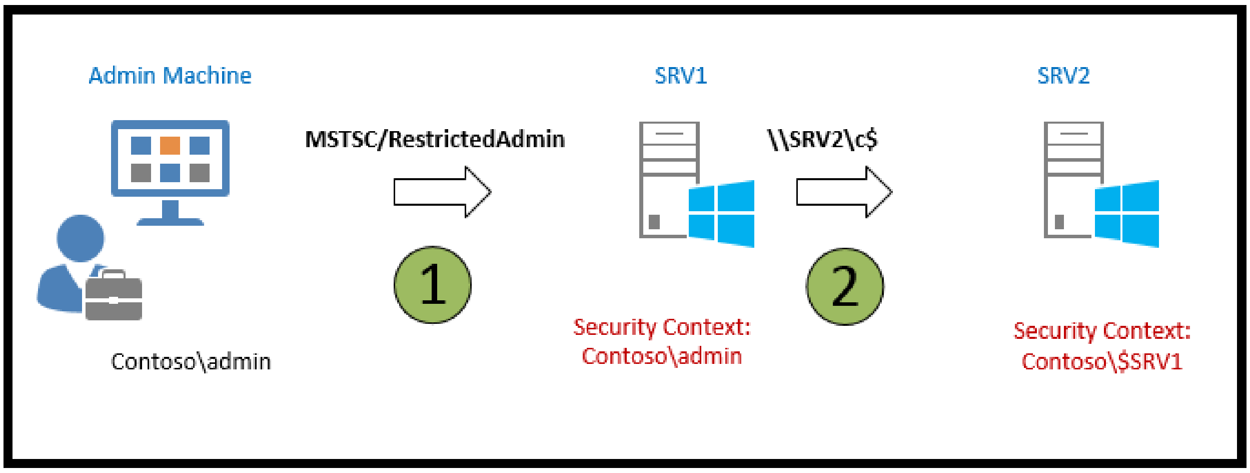
OK

Things to watch out when using this feature

When you connect to a remote computer using this feature, your identity is preserved on that remote server. Say for example that you are connecting from your machine to a server called (SRV1), any activity that you are doing during that remote desktop session on SR1, is performed using your identity. If you tried to access any network resource from that remote server (SRV1), then the identity that is being used is the computer account **$SRV1**, and not your identity. This is because your identity is not stored on SRV1 server, and it cannot be used to jump or connect to a second network resource from there.

Microsoft documentation mentions this *“Restricted mode may limit access to resources located on other servers or networks beyond the target computer because credentials are not delegated*.”

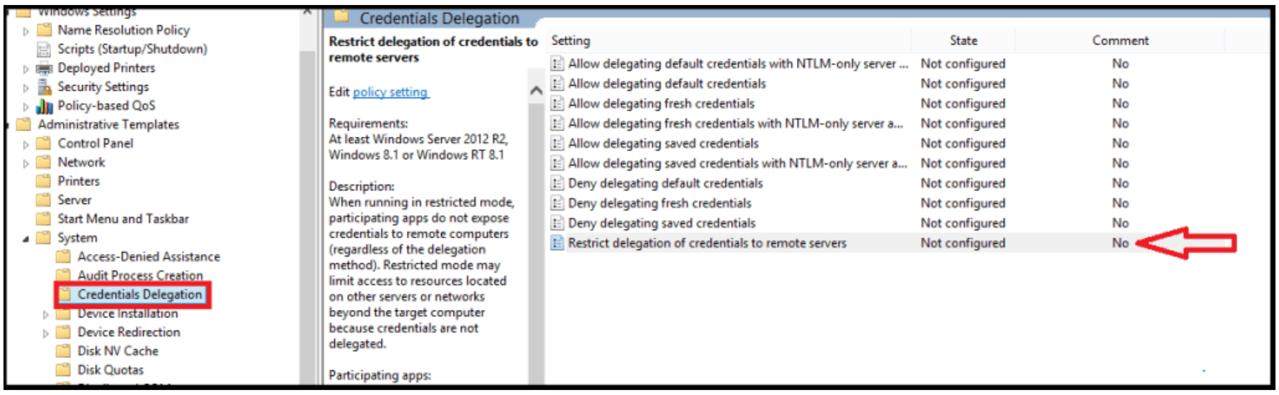
So if I connect to SRV1 from my machine, and then I tried to access the admin share on SRV2 from that remote desktop session, then the connection will happen using **$SRV1** computer account and not mine.



GPO Settings

There is a tricky GPO to control and enforce this new feature. The tricky part that this GPO setting should be applied to the machines initiating the remote desktop session using **/RestrcitedAdmin** feature, and not on the target RDP server. For example, if I had Windows 8.1 clients all over my network, it would be a good idea to force this setting on my help-desk workstations, so that when they RDP to client systems, they would be forced to use *Restricted Admin mode for RDP*.

GPO setting is located under the **Administrative Templates under Computer Configuration > System > Credential Delegation > Restrict delegation of credentials to remote servers.**



Restrict delegation of credentials to remote servers

Requirements:
At least Windows Server 2012 R2, Windows 8.1 or Windows RT 8.1

Description:
When running in restricted mode, participating apps do not expose credentials to remote computers (regardless of the delegation method). Restricted mode may limit access to resources located on other servers or networks beyond the target computer because credentials are not delegated.

Participating apps:

Setting	State	Comment
Allow delegating default credentials with NTLM-only server ...	Not configured	No
Allow delegating default credentials	Not configured	No
Allow delegating fresh credentials	Not configured	No
Allow delegating fresh credentials with NTLM-only server a...	Not configured	No
Allow delegating saved credentials	Not configured	No
Allow delegating saved credentials with NTLM-only server a...	Not configured	No
Deny delegating default credentials	Not configured	No
Deny delegating fresh credentials	Not configured	No
Deny delegating saved credentials	Not configured	No
Restrict delegation of credentials to remote servers	Not configured	No

Limitations

Restricted Admin mode for RDP only applies to administrators, so it cannot be used when you log on to a remote computer with non admin account. Also the destination server should support the *Restricted Admin mode for RDP.*

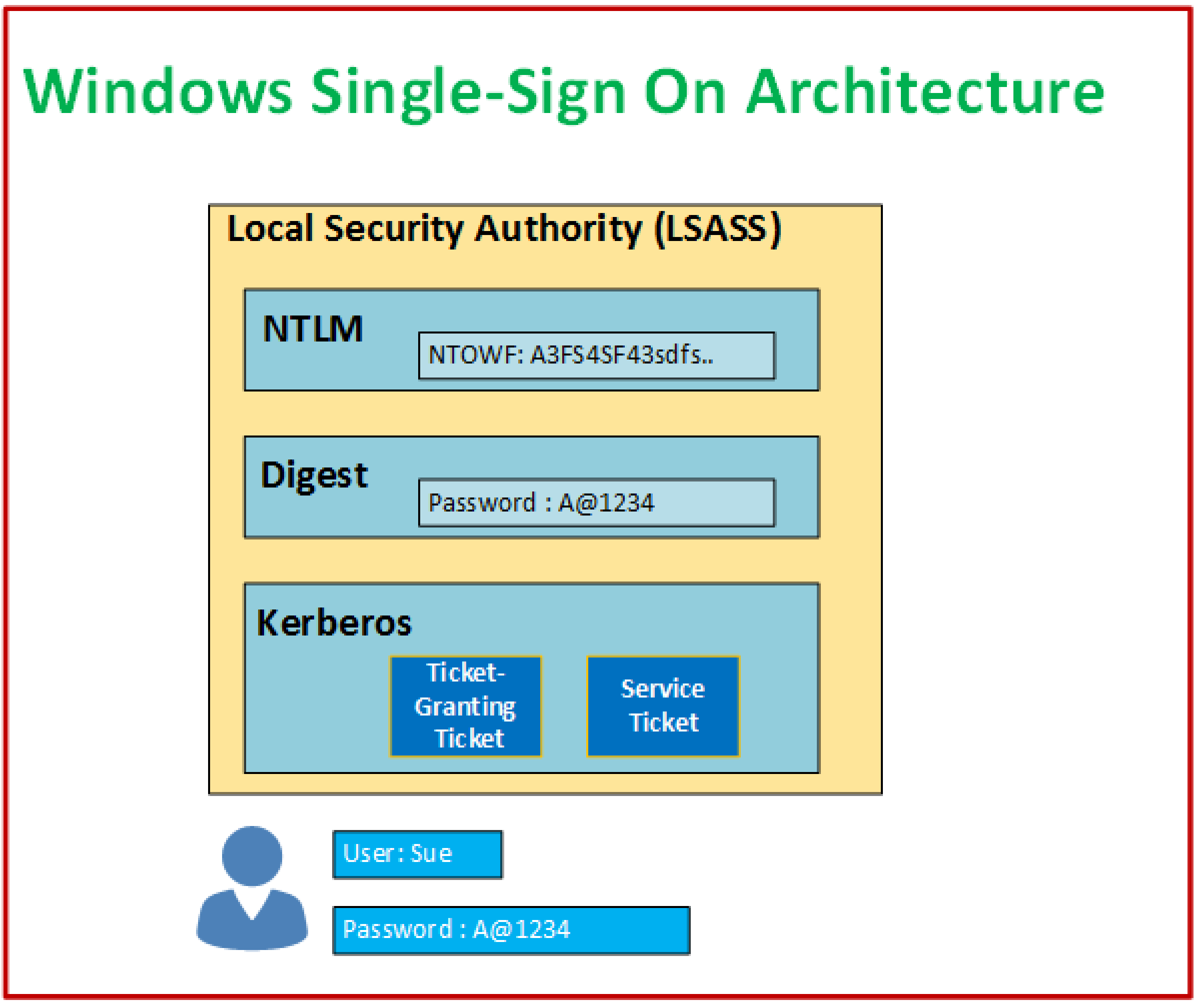
Furthermore, the remote server cannot delegate your credentials to a second network resource. This can become a problem with some implementations like remote apps.

Security Trade-Off

There is a big argument on the internet about how vulnerable this feature can be to pass the hash attacks. To explain my point of view, I will talk about how interactive logon works and how network logon works.

Interactive Logon

- John inputs his credentials to the machine by entering his username and password.
- The machine checks if the credentials are right by contacting a domain controller using (Kerberos by default, or NTLM when kerberos is not available).
- If the domain controller approves that identity, the user is authorized to access the machine and a Single-Sing On (SSO) data is stored on that machine. This can be a *Ticket Granting Ticket* **TGT** or NTLM hash of the user password. SSO data is stored in memory, and is required to ensure *Single Sign On* experience for John, so he can access network resources without the need to type his credentials over and over.



Network Domain Logon

- John logs on to his machine using interactive logon and has his SSO data is stored in memory as shown the previous figure.
- When John wants to access a network resources like a remote file share using network domain logon, an SSO token derivative (a Kerberos TGS ticket or a challenge encrypted with the NTLM hash) is used to prove the user's identity to the target machine.
- The target machine uses the domain controller to validate the authenticity of the SSO derivative, and to receive authorization data for the user. It's important to note that the SSO token itself does not leave the user's machine and specifically, it is not sent to the target machine.

Which one is better?

- **From John's machine perspective**, *network logon* is better because when he access a network share, he can do that using *network logon*. His actual SSO data is not sent to the target server, and thus *network logon* reduces the user's exposure to pass-the-hash attack.

- **From the remote server perspective**, allowing *network logon* means that an attacker that has access to user hash, can use *network logon* to access it. On the other hand, if that server does not allow network logon, then pass-the-hash attack is not possible. In other words, a server that does not allow network logon, is not vulnerable to pass-the-hash attack.

How normal RDP connection works (without /RestrictedAdmin)?

Prior to Windows 8.1, the only way to connect and authenticate to a remote computer using RDP was with the **Remote Interactive Logon** Process:

- John enters his credentials to the RDP client.
- RDP client performs *network logon* to the target server to authorize the John.
- Once John is authorized, the RDP client securely relays the credentials to the target machine over a secure channel.
- The target server uses there credentials to perform an *interactive logon* on behalf of John.

Note: the remote server should gain access to the actual credentials to allow remote desktop connection.

How RestrictedAdmin RDP connection works ?

Using this mode with administrative credentials, RDP will try to interactively logon to the remote server without sending credentials. *Restricted Admin mode for RDP* does not at any point send plain text or other re-usable forms of credentials to remote computers.

This means that if an attacker has only the hash of the password, he can access a remote computer using *Restricted Admin mode for RDP* as now the actual credentials are not a requirement to establish the connection. While without using *Restricted Admin mode for RDP*, knowing the actual credentials is a must.

In other words, *network authentication* is used heavily when using *Restricted Admin mode for RDP*, which means that either NTLM or Kerbeors will work by default.

Previously, if you know the admin hash, you can pass-the-hash with psexec tool and take over the remote system if SMB/RPC (ports 445,135,139,,) were exposed. But because many administrators already block these ports leaving only RDP inbound connection allowed, now the attacker can pass-the-hash using the RDP protocol.

Blog Post Notification