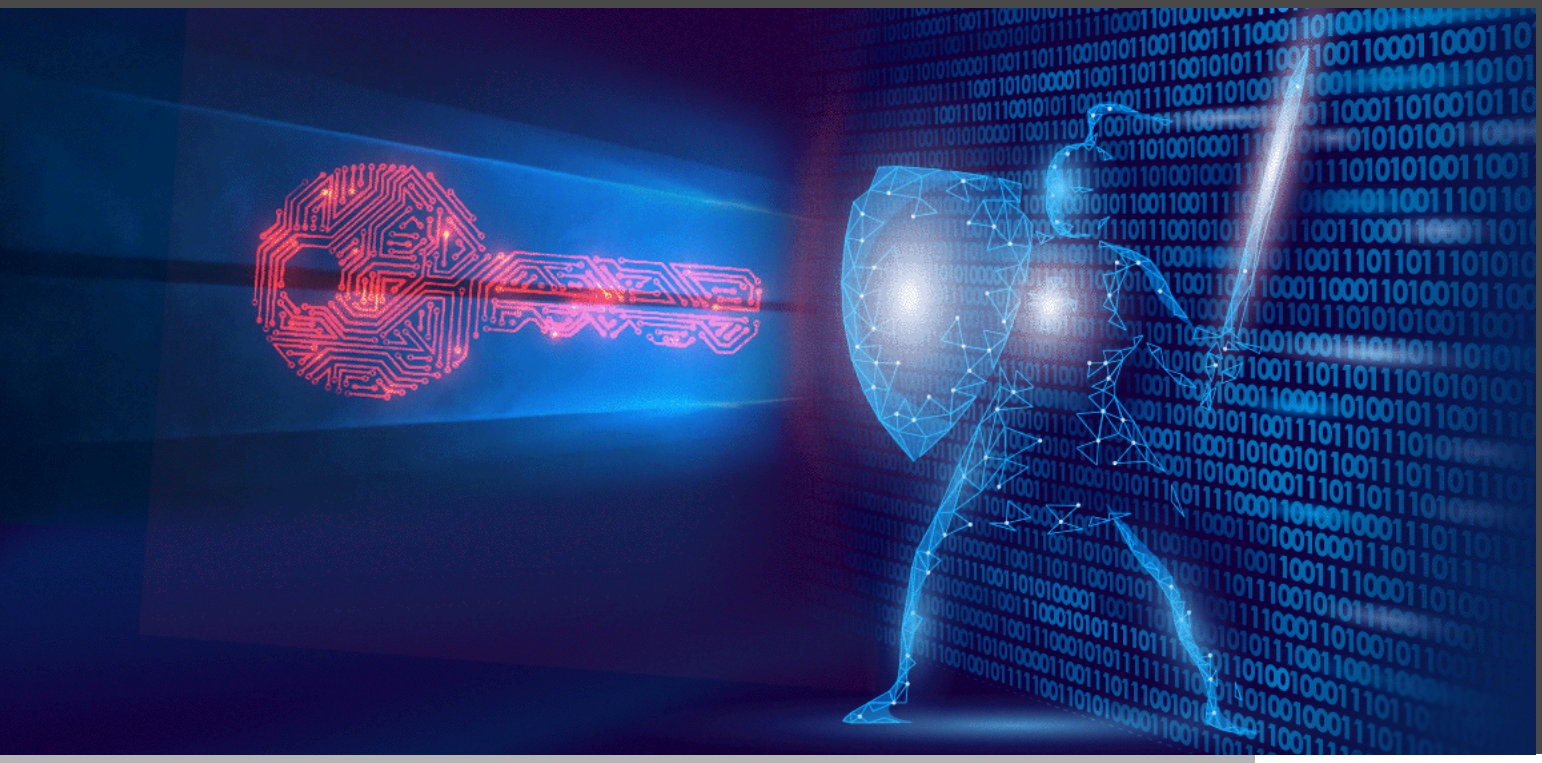
Devolutions

Protecting RDP Passwords from Mimikatz Using Remote Credential Guard



THE FULL CREDENTIALS HAVE ALWAYS BEEN SENT AS PART OF THE RDP NETWORK LEVEL AUTHENTICATION PROTOCOL

There was a bit of a stir recently when someone noticed that the password used to connect with RDP remained in memory and therefore could be grabbed by the popular mimikatz tool. The reality is that this "discovery" wasn't really a secret, because the full credentials have always been sent as part of the RDP Network Level Authentication (NLA) protocol. The only reason why it didn't get so much attention until now is that nobody bothered to look in the right place.

The Quest for Better Security

The best way to mitigate against RDP credential grabbing is to use RDP Remote Credential Guard (RCG), but this feature had so far been restricted to the built-in Windows RDP client (mstsc.exe). That was before we found a way to add Remote Credential Guard (RCG) support in Remote Desktop Manager for embedded RDP sessions on Windows, after which we shared the undocumented way to do it with our competitors. As if it wasn't enough, we went beyond the limitations of the original RDP client to support explicit credentials in RCG-enabled RDP sessions, meaning you can log in under a different user, not just the current one.

I'm Sold! How Does This Work?

Feeling lost? You are not alone. Let's begin by covering how to use Remote Credential Guard in Remote Desktop Manager. The complete technical explanation would require an entire blog post of its own. For now, the only thing you need to know is that an RCG-enabled RDP session does not send the full credentials to the RDP server, which is why mimikatz cannot grab them in memory.

Enabling RCG in the RDP Server

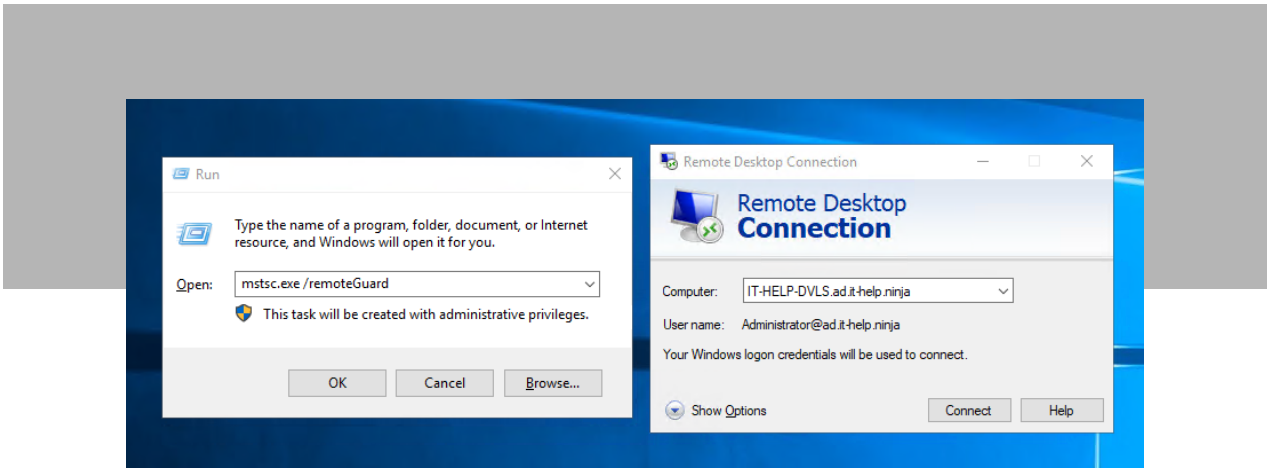
Remote Credential Guard only works between recent versions of Windows joined to the same domain. It should be supported on Windows Server 2016 and later, and on Windows 10 1607 and later. Since Kerberos is also required, make sure to use the machine FQDN when connecting to avoid an NTLM fallback. Last but not least, the target RDP server needs the "DisableRestrictedAdmin" registry key set to zero:

```
$Params = @{
    Path = «HKLM:\SYSTEM\CurrentControlSet\Control\Lsa»;
    Name = «DisableRestrictedAdmin»;
    PropertyType = «DWORD»;
    Value = 0;
}
New-ItemProperty @Params -Force
```

If you forget to set this registry key and attempt connecting with Remote Credential Guard, Winlogon will show the following error: "Account restrictions are preventing this user from signing in. For example: blank passwords are not allowed, sign-in times are limited, or a policy restriction has been enforced."

Testing RCG with mstsc

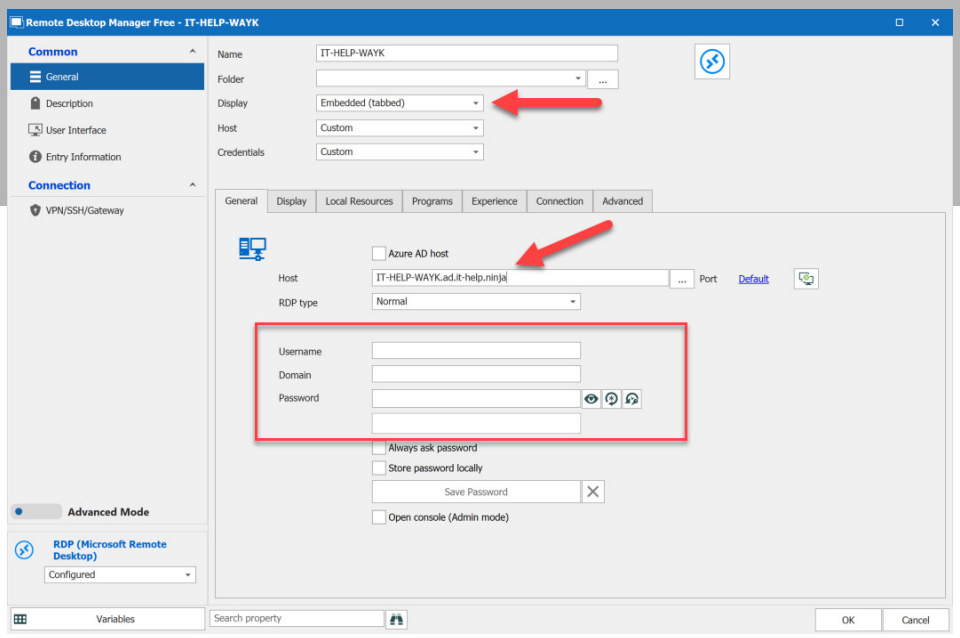
Let's make our first Remote Credential Guard connection with mstsc to make sure that it works, after which we'll do it in Remote Desktop Manager. From a domain-joined machine, launch mstsc with the "/remoteGuard" parameter, enter the hostname of the RDP server and then click Connect:



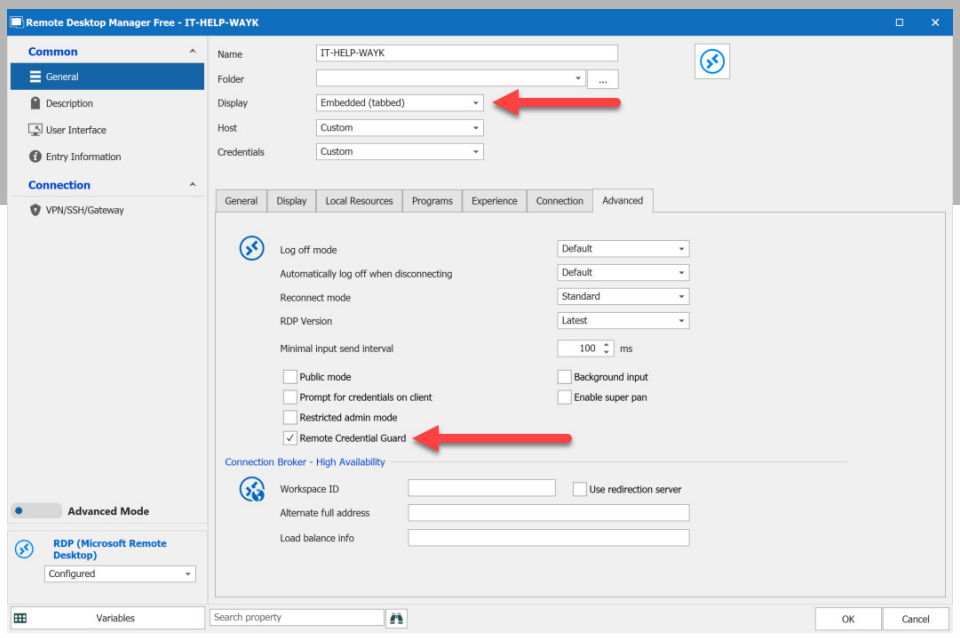
If everything worked, it should connect without prompting for the credentials, which feels like magic the first time you try it. In a way, Remote Credential Guard is a form of single sign-on (SSO) for RDP, even if Microsoft never marketed it this way. This is also the only way you can use RCG with mstsc: the interface restricts you to the current user, and you cannot manually enter credentials for a different user.

Enabling RCG in Remote Desktop Manager

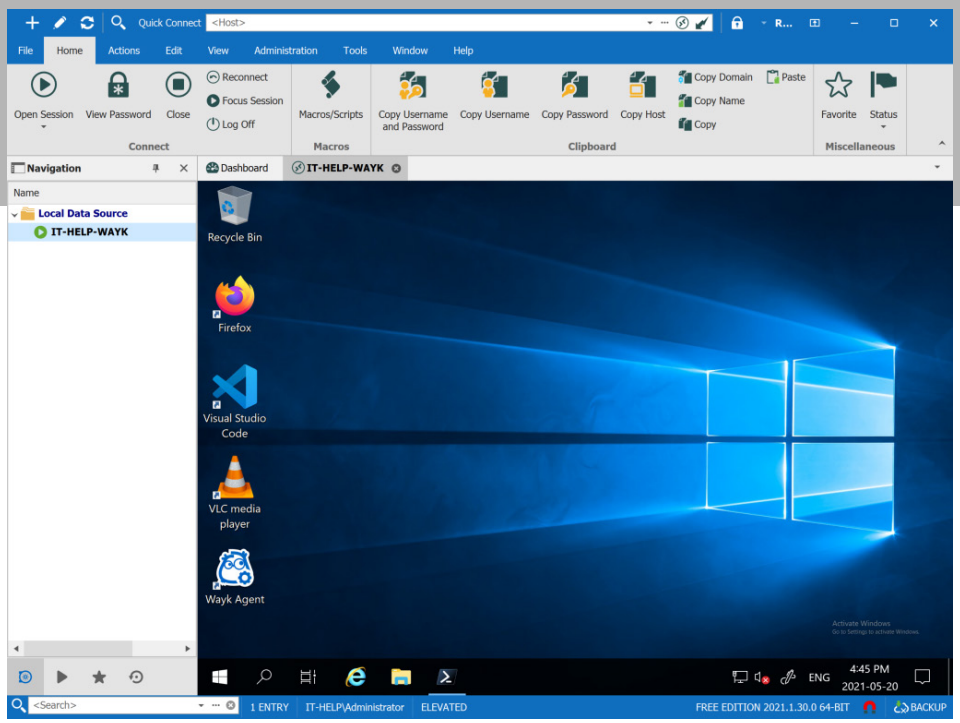
In Remote Desktop Manager, create an embedded RDP session entry. Enter the same RDP server hostname, but leave the credentials empty for now.



In the Advanced tab, enable the Remote Credential Guard option. This particular option was previously restricted to the external display mode, so if you can't see it, make sure that you are running RDM 2021.1.29 or later:



Save the new connection entry, then launch it. If it connects without prompting for credentials, it worked!



You can now edit the RDP connection entry to add your own credentials for a different user, and try connecting again. It should work the same way as the default mode which uses the current user to connect, but with the added security of Remote Credential Guard!

Closing Thoughts

While Remote Credential Guard is a good way to avoid exposing the full credentials to the RDP servers you connect to, it is a security feature currently restricted to Windows. Unfortunately, the underlying protocol that makes Remote Credential Guard possible is extremely difficult to port to other platforms, making its potential usage limited. We hope that Microsoft realizes the importance of interoperability and decides to revamp the protocol to make it easier to implement in third-party clients. Until then, stay tuned for updates, as we will keep looking for ways to help you improve security in RDP connections.