



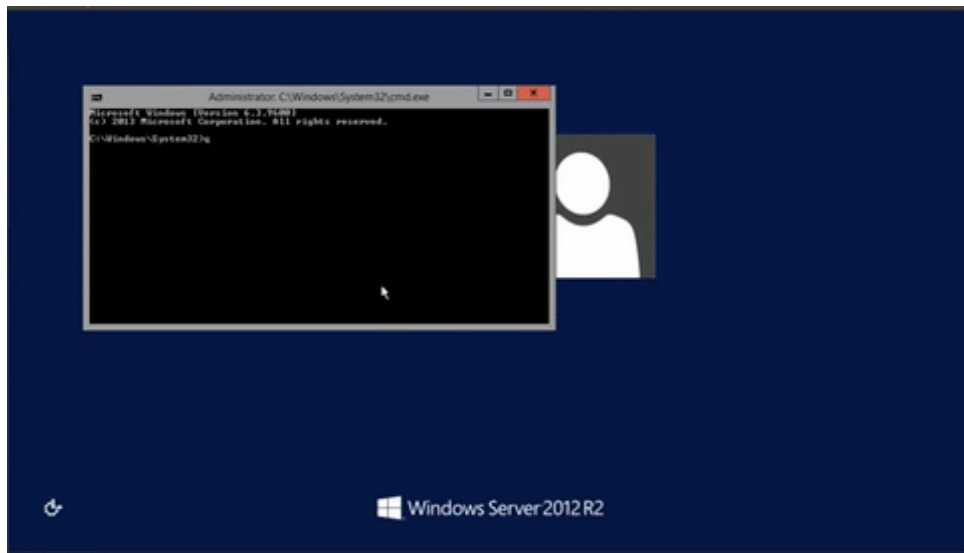
RDP hijacking — how to hijack RDS and RemoteApp sessions transparently to move through an organisation

How you can very easily use Remote Desktop Services to gain lateral movement through a network, using no external software — and how to defend against it.



Kevin Beaumont

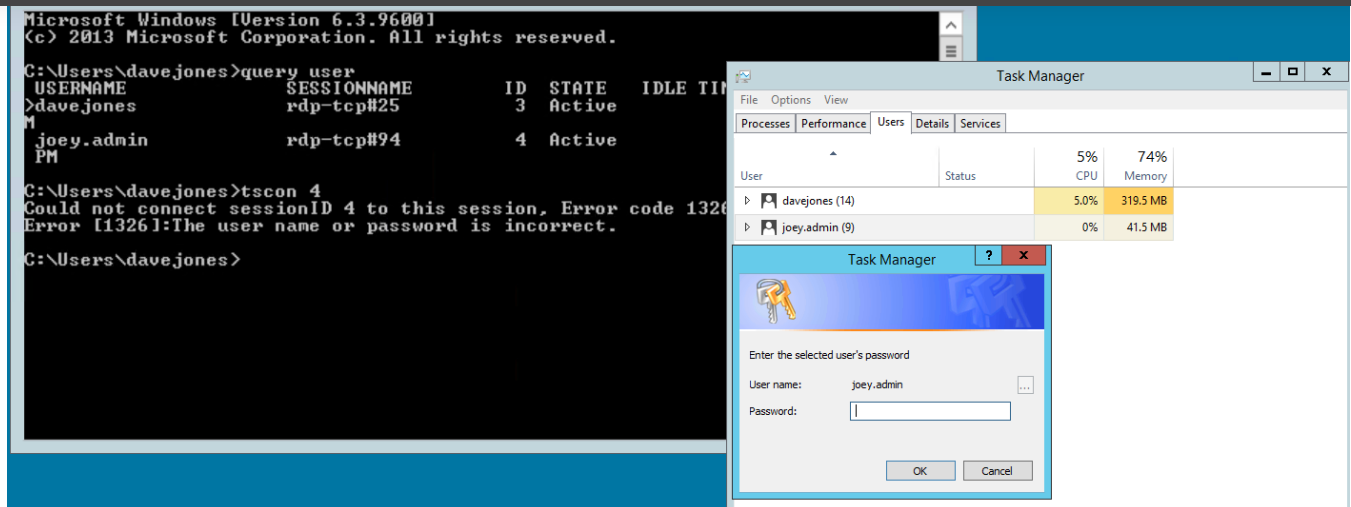
Mar 19, 2017 · 7 min read



Alexander Korznikov demonstrates using Sticky Keys and tscon to access an administrator RDP session — without even logging into the server.

Brief background on RDP session connection

If you've used Remote Desktop Services before, or Terminal Services if you're as old as me, you will know there's a feature where you connect to another user's session — *if you know their password*. Did you know you can also hijack a session without the user password? Read on.



Some tricks allow credential-less Session Hijacking

Here's the deal. As revealed by [Benjamin Delpy](#) (of Mimikatz) in 2011 and by [Alexander Korznikov](#) on Friday, if you run `tscon.exe` as the SYSTEM user, **you can connect to any session without a password**. It doesn't prompt, it just connects you to the user's desktop. I believe this is due to the way session shadowing was implemented in Microsoft Windows, and it runs throughout the years like this.

Now, you might be saying 'If you're SYSTEM, you're already root... You can already do anything'.

Yes. Yes you can. You could, for example, dump out the server memory and get user passwords. That's a long process compared to just running `tscon.exe` with a session number, and instantly get the desktop of said user — with no obvious trace, or external tools. This isn't about SYSTEM — this is about what you can do with it very quickly, and quietly. Attackers aren't interested in playing, they're interested in what they can do with techniques. This is a very valid technique.

So, you have full blown RDP session hijacking, with a single command.

Some parameters about how far this reaches

- You can connect to disconnected sessions. So if somebody logged out 3 days ago, you can just connect straight to their session and start using it.



unlocks the physical console, too.

- You can connect to ANY session — so if, for example, it's the Helpdesk, you can connect to it without any authentication. If it's a Domain Admin, you're in. Because of the above point (you can connect to disconnected sessions), this makes it an incredibly simple way to laterally move through a network.
- You can use win32k SYSTEM exploits — there are many — to gain SYSTEM permissions, and then use this feature. Meaning even as a standard user, if patches aren't applied properly you can use this. Obviously, any route to SYSTEM is valid — e.g. any method to get to a local administrator (there's a few!).
- There are no external tools. Nothing to get through application whitelisting. No executable is written to disk.
- Unless you know what to monitor (more on that later), you won't know this is happening.
- It works remotely. You can take over sessions on remote computers, even if you're not logged into that server.

Gaining SYSTEM for tscon.exe

If you're an administrator, you can use a service as Alexander demonstrates:





and your own SESSIONNAME. Your own Session will be replaced with the hijacked session. The service will run as SYSTEM by default — you're in.

Just remember to delete the service afterwards, if you're evil.

Here's an example of it in practice on a Windows Server 2012 R2 server:

<https://www.youtube.com/watch?v=OgsoIoWmhWw>

Other methods:

- You can use Scheduled Tasks to gain SYSTEM and run the command. Just schedule the command to run immediately as SYSTEM with interactive privileges.
- Use can use a variety of methods like Sticky Keys to get SYSTEM, without even needing to log in (in the future). See below.
- Exploits etc (see above).

Lateral movement

Most organisations allow Remote Desktop through their internal network, because it's 2017 and that's how Windows administration works. Also, RemoteApp uses RDP. Because of this, it's a fantastic way to move around an organisation's network — forget passwords, just surf around and abuse other people's access. You appear in the organisation logs as that user, not yourself.

How to backdoor for credential-less hijacking

Remote Desktop bruteforcing is a major problem. Anybody who has setup a honeypot recently will know within seconds you will be getting hit with failed RDP logins. First they portscan, then thousands of login attempts arrive.

It gets worse — I run RDP honeypots, and I see them regularly — when breached they get backdoored using the techniques below.

From research, over 1 in 200 scanned Remote Desktop servers online are already backdoored using these methods. This means that you can session hijack with them



RDP backdoor method one — Sticky Keys

The concept here is pretty simple — Windows supports a feature called Sticky Keys, which is an Accessibility feature built into the OS and available pre-logon (at the login screen, either via a physical console or via Remote Desktop). It runs as SYSTEM.

If you set Sethc.exe (Sticky Keys) to spawn cmd.exe, you have a backdoor you can use if you are locked out of a box — you have SYSTEM access, so you can do anything even without an account. You can do this by either replacing sethc.exe with cmd.exe — this requires a reboot, and physical access to the box — or just set the registry key using the command below.

```
REG ADD "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image  
File Execution Options\sethc.exe" /t REG_SZ /v Debugger /d  
"C:\windows\system32\cmd.exe" /f
```

Ta-da! The box is now permanently backdoored. Just Remote Desktop in and at the login screen, hit F5 a bunch of times.

Method two — Utilman

It's exactly the same as before, just trojan utilman.exe instead. At the login screen, press Windows Key+U, and you get a cmd.exe window as SYSTEM.

```
REG ADD "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image  
File Execution Options\utilman.exe" /t REG_SZ /v Debugger /d  
"C:\windows\system32\cmd.exe" /f
```

Scanning for backdoor'd RDP servers

There is a prebuilt tool here, which works wonders — just spin it up and find servers which already have a SYSTEM level backdoor exposed:

ztgrace/sticky_keys_hunter

sticky_keys_hunter - A script to test an RDP host





From online scanning, a significant amount of open RDP servers online are already backdoored.

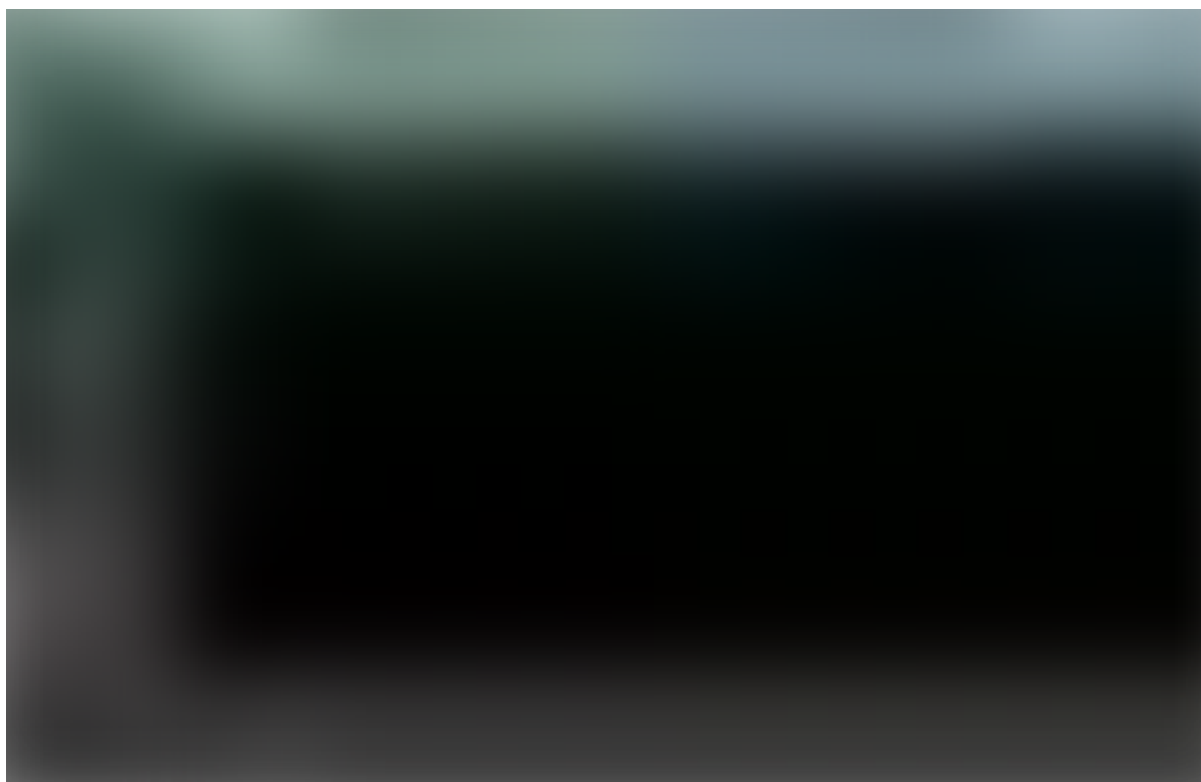
Mimikatz module

There is now a Mimikatz module for very easily doing this:

gentilkiwi/mimikatz

mimikatz - A little tool to play with Windows security

[github.com](https://github.com/gentilkiwi/mimikatz)



gentilkiwi rocking it

Mitigations

OS-I had a section about Window Server 2016 here, however after further investigation it appears to also be impacted. After testing this applies to every OS since Windows 2000, including Windows 10 and 2016.



world. I would also log off idle sessions.

Don't expose RDS/RDP to the internet — if you do, I *strongly suggest* you implement multi-factor authentication. You can use things like Microsoft RD Gateway or Azure Multi-Factor Authentication Server to get very low cost multi-factor authentication. If you're exposing RDP directly to the internet and somebody creates a local user or your domain users have easy to guess or reused credentials, things will go downhill fast. Trust me — I've seen hospitals and others be ransomware'd by RDS servers.

Monitoring

It is surprisingly very difficult to record session hijacking — there is one event log (Microsoft-Windows-TerminalServices-LocalSessionManager/Operational) which records sessions connecting — however it does not appear to differentiate between a normal user connecting and tscon.exe being used — I've been through every other event log and can't see anything which suggests this is happening. This is actually a major issue and I lobby Microsoft to add some kind of Event Log ASAP — it's a real gap.

My suggestion is you alert for other related behaviour using the Event Log and tools like Microsoft OMS, Windows Event Forwarding, Splunk etc. You're looking for SYSTEM being misused.

For example abnormal Service creation and abnormal scheduled task creation should be logged centrally, and recorded against. Additionally, you can look for Mimikatz related activity.

- k

FAQ

Q: This isn't new or a vulnerability.

A: Java applets and macros aren't new. If the technique works, it will get used. This one has flown under the radar — that doesn't mean it is not valid.

Q: If you have SYSTEM you already own the box.

[Microsoft](#)[Sysadmin](#)[Rds](#)[Hijack](#)

818 claps

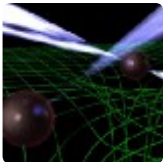


WRITTEN BY

Kevin Beaumont

"Influential UK infosec geezer" - The Register | "Cyber security expert" - BBC News | "Top 20 Influencers" - Imperva | "A bloody idiot" - my mum.

Follow



DoublePulsar

Cybersecurity from the trenches of reality, written by Kevin Beaumont. Opinions are of the author alone, not their employer.

Follow

See responses (5)

