
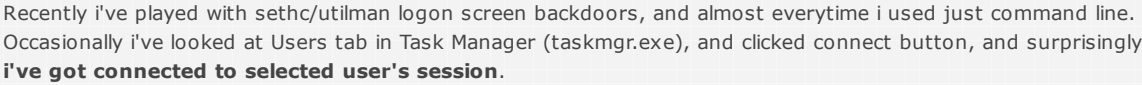


Friday, March 17, 2017

\* This post periodically updated, all updates in the end of the post.

Hey there,

Blogpost in 20 seconds: Fun with sethc backdoored host :) somewhere in the internet:

A small, square portrait of a man with short dark hair, wearing dark sunglasses and a light-colored shirt. He is looking slightly to the right. The background is a blurred outdoor setting with trees.

Opportunities welcome.  
Follow @nopernik

[View my complete profile](#)

## #HOWTO #Linux: Scripting

ScriptingIntro. There are a lot of programming languages, for simplicity, we will group them:Low ....

## #HOWTO #Linux: Input, Output, Redirection & PIPEs

Input, Output, redirections and PIPEs. Every program in Linux can get, process and finally output ....

## #HOWTO #Linux: File-System

Filesystem Structure: Each file has its own permissions for read, write and execute access for every ....

## #HOWTO #Linux: Basics

## SSHPry v2 - Spy & Control

### SSH Connected client's TTY

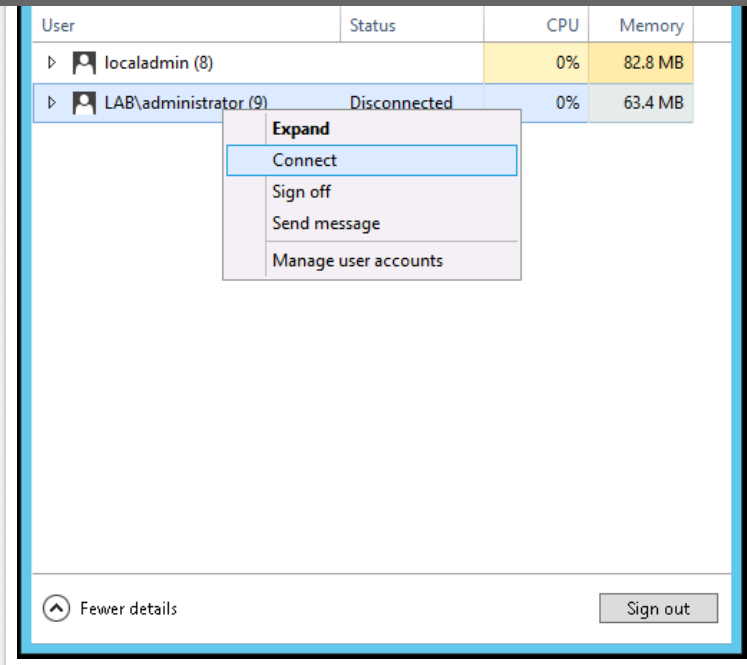
- What if we'll have a tool that can show us a terminal of active SSH connection? and... maybe... ....

~# history

- ▶ 2022 (4)
- ▼ 2017 (6)

Ce site utilise des cookies provenant de Google pour fournir ses services et analyser le trafic. Votre adresse IP et votre user-agent, ainsi que des statistiques relatives aux performances et à la sécurité, sont transmis à Google afin d'assurer un service de qualité, de générer des statistiques d'utilisation, et de détecter et de résoudre les problèmes d'abus.

EN SAVOIR PLUS OK!



- ▶ 2016 (12)
- ▶ 2015 (15)
- ▶ 2014 (25)

## ~# Contact Me

Name

Email \*

Message \*

Send

When i checked it again with local admin rights, it **failed** by asking user's password. Why and how that happened? Let's dig deeper.

Related to Microsoft documentation:

[https://technet.microsoft.com/en-us/library/cc770988\(v=ws.11\).aspx](https://technet.microsoft.com/en-us/library/cc770988(v=ws.11).aspx)

[https://technet.microsoft.com/en-us/library/cc731007\(v=ws.11\).aspx](https://technet.microsoft.com/en-us/library/cc731007(v=ws.11).aspx)

we can see couple important remarks:

## Remarks

- You must have **Full Control access permission** or **Connect special access permission** to connect to another session.
- The **/dest:<SessionName>** parameter allows you to connect the session of another user to a different session.
- If you do not specify a password in the **<Password>** parameter, and the target session belongs to a user other than the current one, **tscon** fails (**not really**).

I've got it! Sticky Keys (cmd backdoor) at windows login screen runs with NT AUTHORITY/SYSTEM and have Full Control access permission, and can connect to EVERY user session without asking for a password.

So we've got a session hijacking here. The most funny thing is that the legit user isn't asked for logout, by using this technique the user just will be **kicked out of the session without any notification**.

## Attack Vector Details:

A privileged user, which can gain command execution with NT AUTHORITY/SYSTEM rights can hijack any currently logged in user's session, without any knowledge about his credentials.

Terminal Services session can be either in connected or disconnected state.

**This is high-risk vulnerability which allows any local admin to hijack a session and get access to:**

Ce site utilise des cookies provenant de Google pour fournir ses services et analyser le trafic. Votre adresse IP et votre user-agent, ainsi que des statistiques relatives aux performances et à la sécurité, sont transmis à Google afin d'assurer un service de qualité, de générer des statistiques d'utilisation, et de détecter et de résoudre les problèmes d'abus.

EN SAVOIR PLUS OK!

## Example scenario:

Some bank employee have access to billing system, and it's credentials to login.

One day, he come to work, logging in to the billing system and start to work. At lunch time he **will lock his workstation**, and out to lunch.

Then, system administrator gets to employee's workstation, and logs in with his administrator's account.

According to the bank's policy, administrator's account should not have access to the billing system, but with couple of **built-in commands** in windows, this system administrator will hijack employee's desktop which he leaved locked. From now, sysadmin can perform malicious actions in billing system as billing employee account.

There are huge amount of scenarios like this.

Furthermore, an attacker doesn't need to use tools like metasploit, incognito, mimikatz etc, which is commonly used for user's token manipulation and impersonating logged in users. **Everything is done with built-in commands**. Every admin **can** impersonate any logged in user either locally with physical access or remotely via Remote Desktops (see PoC).

## Tested on:

Windows 2016 (Confirmed by Kevin Beaumont @GossiTheDog)

Windows 2012 R2

Windows 2008

Windows 10

Windows 7

## We can talk about endless amount of examples.

It can be done remotely, as shown in Proof of Concepts.

An attacker can hijack active or disconnected session remotely via remote desktops.

**I use this technique about three weeks** in my on-going penetration tests on daily basis. It in very simple way helps me to get access to sensitive information like emails, opened documents, clear-text passwords that administrators write down in notepad (not intended for saving, but for temporally writing it somewhere), **opened RDP sessions to another external domains** (think cloud), or another applications that make use of different login credentials.

Someone can say, if you admin, you can dump server's memory and parse it. That's correct, but you don't need it any more. Just two simple commands and you are in. The most incredible thing, is that I don't need to know the credentials of hijacked user, it is pure passwordless hijacking.

A successful **attack heavily related on time** and gathered information. If you need to dump a memory, to get your sensitive info, you're in problem. That means that you've tried all quick-wins that you know.

In example of hijacking user (active or disconnected) while he is working now remotely on some sensitive server that i have no access to, and haven't even knew about it, this technique allows me to compromise that server in **less than a minute**. Everything is real and from my own experience.

Furthermore, as I understand it is very hard to catch if this attack happen. Kevin Beaumont @GossiTheDog make an alert on tscon.exe usage, with Microsoft OMS.

I had a conversation about this finding with Benjamin Delpy @gentilkiwi author of mimikatz:

*"That is normal Windows API, that's the design flow, they use it. As mentioned earlier, if you admin, you can do everything. But here is the point. Why and HOW you become admin? If some unprivileged user becomes admin using some kind of local privilege escalation - that's the problem and not the design flow we are talking about. You can do everything, even patch terminal services the way that it will accept your token and allow shadowing mode, without user's knowledge."*, he said.

Ce site utilise des cookies provenant de Google pour fournir ses services et analyser le trafic. Votre adresse IP et votre user-agent, ainsi que des statistiques relatives aux performances et à la sécurité, sont transmis à Google afin d'assurer un service de qualité, de générer des statistiques d'utilisation, et de détecter et de résoudre les problèmes d'abus.

EN SAVOIR PLUS OK!

But we need is not to know any SYSTEM command line.

Easiest method with psexec, but requires psexec.exe to be there:

```
psexec -s \\localhost cmd
```

Another method is to create a service that will connect selected session to ours.

1. Get all sessions information:

```
C:\Windows\system32>query user
USERNAME          SESSIONNAME      ID  STATE  IDLE TIME  LOGON TIME
administrator      1  Disc    1  3/12/2017 3:07 PM
>localadmin        rdp-tcp#55       2  Active .  3/12/2017 3:10 PM
C:\Windows\system32>
```

2. Create service which will hijack user's session:

```
C:\Windows\system32>sc create sesshijack binpath= "cmd.exe /k tscon 1 /dest:rdp-tcp#55"
[SC] CreateService SUCCESS
```

3. Start service:

```
net setart sesshijack
```

Right after that your session will be replaced with target session.

Proof of Concept video:

**Windows Server 2016 Demo (new):**

<https://youtu.be/bbTfN5geSKw>

**Windows 7 via Task Manager:**

<https://youtu.be/oPk5off3yUg>

**Windows 7 via command line:**

<https://youtu.be/VytjV2kPwSg>

**Windows 2012 R2 via service creation:**

<https://youtu.be/OgsoIoWmhWw>

Update: @gentilkiwi has found that before in 2011, so that is a feature and not zero-day:

<http://blog.gentilkiwi.com/securite/vol-de-session-rdp>

Update: If you still think that this don't have high attack value, read a great writeup by Kevin Beaumont about this feature:

<https://medium.com/@networksecurity/rdp-hijacking-how-to-hijack-rds-and-remoteapp-sessions-transparently-to-move-through-an-da2a1e73a5f6>

Update: RedSnarf has now support in RDP Hijacking [https://www.youtube.com/watch?v=VrF8uXK\\_ePY](https://www.youtube.com/watch?v=VrF8uXK_ePY)

Follow @nopernik



Автор: nopernik на 1:57:00AM



43 comments:

Ce site utilise des cookies provenant de Google pour fournir ses services et analyser le trafic. Votre adresse IP et votre user-agent, ainsi que des statistiques relatives aux performances et à la sécurité, sont transmis à Google afin d'assurer un service de qualité, de générer des statistiques d'utilisation, et de détecter et de résoudre les problèmes d'abus.

EN SAVOIR PLUS OK!

thanks shai

Reply



**Unknown** March 20, 2017 at 4:52 PM

This comment has been removed by the author.

Reply



**Unknown** March 20, 2017 at 5:20 PM

Just stop using this M\$ shit :)

Reply

**Anonymous** March 20, 2017 at 8:03 PM

What about M\$ protection ? released M\$ security vulnerability updates ?

Reply

**Anonymous** March 20, 2017 at 9:54 PM

This \*bug\* is due to a call to WTSQueryUserToken, which gives you a token handle that you can then pass into CreateProcessAsUser. You have the SE\_TCB\_NAME privilege set, hence why you need to do it as SYSTEM. I released code to exploit this in 2010. sjl

Reply

▼ Replies



**nopernik** March 21, 2017 at 12:31 AM

where can i see the exploit code?

**Anonymous** March 21, 2017 at 6:35 PM

first of all, I'm not the same anonymous :)

Hi Alexander, thanks a lot for sharing your findings. Even if I knew @gentilkiwi's work I admit I missed his post about this issue.

Doing a bit of googling on "SE\_TCB\_NAME" I found this link:

[http://forums.codeguru.com/showthread.php?159961-How-\\_programmatically\\_-grant-privilege-SE\\_TCB\\_NAME](http://forums.codeguru.com/showthread.php?159961-How-_programmatically_-grant-privilege-SE_TCB_NAME)

Alex Fedotov in 2001 wrote:

Re: How \_programmatically\_ grant privilege SE\_TCB\_NAME

[...]You should never grant the SE\_TCB\_NAME privilege to any real user account, even administrator's account. It's too dangerous. If you need to call LogonUser and CreateProcessAsUser, \*\*\*do it in a service that runs in the LocalSystem logon session\*\*\*.[...]

At the time, soon.exe and srvany.exe (<https://www.microsoft.com/resources/documentation/windowsnt/4/server/reskit/en-us/reskt4u4/rku4list.msp?mfr=true>) were commonly used to do such things (i.e. bypass user specific ACL). Just login as local admin, soon.exe in order to spawn a cmd as NT/SYSTEM and do... what you had to do. At least I finally found WHY this was working.

There is also an old MS KB article showing "How To Manage User Privileges Programmatically in Windows

Ce site utilise des cookies provenant de Google pour fournir ses services et analyser le trafic. Votre adresse IP et votre user-agent, ainsi que des statistiques relatives aux performances et à la sécurité, sont transmis à Google afin d'assurer un service de qualité, de générer des statistiques d'utilisation, et de détecter et de résoudre les problèmes d'abus.

EN SAVOIR PLUS OK!



**nopernik** March 21, 2017 at 8:29 PM

Why you so anonymous? :) thanks for your comment!

Reply



**Carlos** March 21, 2017 at 2:10 AM

If you are a local machine admin you are by definition a "god" on that machine. Similar to Linux root. Windows UAC notwithstanding, there is no such thing as a higher privileged account. This is a failure in understanding by the poster. A "domain account" is not "higher" either. Why would it be? Local machine admin is god on that machine.

Reply

▼ Replies



**nopernik** March 21, 2017 at 2:12 AM

on that machine, not on the domain.



**Carlos** March 21, 2017 at 1:55 PM

As the blog post suggests, if a domain admin is \*also\* logged on to the machine (first off, why is he on your machine?), then yeah, he now has an active session on \*that\* machine and you can take over his account because you are the god on that machine just like Linux root. This just doesn't sound like vulnerability but a general misunderstanding on the part of the poster.



**nopernik** March 21, 2017 at 2:03 PM

think about domain post exploitation.

1. an attacker have hash of local admin
2. an attacker executes command on some fileserver with system privilege (adding sethc backdoor for example)
3. connects via rdp and hijacks session of domain admin

There can be endless amount of scenarios.

On other hand, you are talking about linux root. How hard it will be to hijack some ssh linux session? But you are the "god" in that machine?

In case of windows, it's done with one command now.

**Anonymous** March 21, 2017 at 6:52 PM

@Carlos this is certainly not a misunderstanding on the part of the poster. This effectively gives anyone with a local admin on 1 machine in a domain, the possibility to easily become domain admin. Good practice is to log out fully, but in reality it can be forgotten or just not always done. This is a very major security flaw



**Carlos** March 21, 2017 at 6:53 PM

But it is all based on the premise that there is a domain admin currently logged into the workstation. When and why in the world would that happen? And you're telling me root on Linux can't spy on another Linux session on the machine using Screen or otherwise do a whole host of other stuff? C'mon now. The attacker in this case must already be a local admin. Why would you ever give anybody Local Admin privileges? Why would you ever log into a machine (via RDP or otherwise) where a local admin existed and then leave that session running knowing full well Local Admin is God on that machine???



**Carlos** March 21, 2017 at 7:07 PM

Ce site utilise des cookies provenant de Google pour fournir ses services et analyser le trafic. Votre adresse IP et votre user-agent, ainsi que des statistiques relatives aux performances et à la sécurité, sont transmis à Google afin d'assurer un service de qualité, de générer des statistiques d'utilisation, et de détecter et de résoudre les problèmes d'abus.

EN SAVOIR PLUS OK!



**nopernik** March 21, 2017 at 7:42 PM

Imagine scenario like this (real life scenario):

Intro:

1. We have regular domain
2. Domain Users doesn't have local admin privileges except of IT Dept.

So, the attack flow:

1. User John boot from USB/CD/Network some kind of linux/rescue\_cd
2. John backdoors it's own workstation with sticky key backdoor.
3. With next boot, John have system privileges.
4. John dumps local hashes with command "reg save hklm\sam" (legit right?)
5. John call IT Dept for remote help
6. John catch with netstat the IP address of IT Admin
7. John connects back with pass the hash technique and execute command as system
8. John connects to the IP of the Admin and hijacks it's session while admin out for lunch.
9. Game over.

Mitigations?

Full disk encryption. But if you have an enterprise of 2000 employees, it is relatively hard to implement.

It happens everywhere. You can harden the things, but almost everywhere you can do everything with built-in commands.

You are right, it's not zero-day, it's not vulnerability - it is attack vector.



**Carlos** March 21, 2017 at 8:21 PM

Maybe I'm misunderstanding but how is any of this different than John, local admin (again, God on the machine), installs keylogger on system message pump (something I can do in 10 minutes in C++). John asks IT to RDP into their machine. John has all their passwords. That's an even worse scenario because I no longer need the Domain Admin to be logged in any more (via RDP, local session, or otherwise). This is also possible in Linux (as root).



**Carlos** March 21, 2017 at 8:33 PM

I think (maybe I'm wrong) that the problem is the idea that Local Admin is below System. But, they are not. Windows is not designed this way. A real Local Admin is "root"... System is just a variant of Local Admin. Local Admin can always escalate to System otherwise they are \*not\* "Local Admin." You can create other locked down accounts via AD and Group Policy that come close to having some of the same rights, but they would not be Local Admins (such as Local Root without network access, network access without local rights, etc)... and indeed we do this all the time. Disclosure: I am a developer, with a strong interest in security, but I am not IT. I'm open-minded.



**nopernik** March 21, 2017 at 8:35 PM

Yes, you can do anything. Everything is depends on point of view and scenarios that we can mind. By the way, IMHO one-two commands is much simpler than writing a keylogger. :) again, it's an attack vector, and i know that admin can do what ever he wants.



**Carlos** March 21, 2017 at 9:42 PM

Agreed. It is definitely an attack vector. And I think this vector illustrates a weakness in using RDP to administer systems. But similar issues exist with using VNC to administer Linux or Mac boxes. It's probably even worse there. I understand that RDP is the quick and easy way for domain admins to administer Windows boxes who don't really care to use PowerShell, remote CMD, or any of the myriad of MSC Remote Management Console tools available.

**Carlos** March 21, 2017 at 9:49 PM

Ce site utilise des cookies provenant de Google pour fournir ses services et analyser le trafic. Votre adresse IP et votre user-agent, ainsi que des statistiques relatives aux performances et à la sécurité, sont transmis à Google afin d'assurer un service de qualité, de générer des statistiques d'utilisation, et de détecter et de résoudre les problèmes d'abus.

EN SAVOIR PLUS OK!



**Carlos** March 21, 2017 at 9:59 PM

:-) Also, perhaps not 2-lines of CMD commands, but a keylogger is like 5 lines of code (not just on Windows, but on any OS). The issue is getting it to run as SYSTEM. Which you can't do unless you're a \*Local Admin\*.

---

Reply

**r00tk4** March 21, 2017 at 12:38 PM

Hey mate you arrive late, this is a design flow, in windows basically the system account can impersonate each user.

You can find more info on impersonation and a tool made by us here [www.blackmath.it](http://www.blackmath.it) or a video here <https://www.youtube.com/watch?v=iI6JKRivgHU>, have fun!

Reply

▼ Replies



**nopernik** March 21, 2017 at 2:14 PM

Just for clarification. I've not invented pass the token.

In your video demo, you show some kind of external program which behave like incognito or mimikatz, and can pass the token.

I assume that the attacker is on the left side, and the client on the right side.

So, left side is never get gui session of impersonated user, on the right side you are connecting an active user (which may be legit) to another session. Pointless.

I'm talking about full GUI RDP passwordless session hijacking, that's all.

**r00tk4** March 21, 2017 at 3:45 PM

Without any external program:

```
sc create myserv binpath= "tscon 2 /dest: tcp-rdp#0 "  
sc start myserv
```

"NT AUTHORITY\SYSTEM" can impersonate each user, no zero day, no feature, simply how windows is built.

You understand that from System to one user the way is easy, you can do that because system account can open handles to each user token on the machine and our software does exactly the same. I hope you understand what we means... In any case mimikatz do something completely different...

Anyway we are opened to collaborate on this theme....

---

Reply



**Mike** March 21, 2017 at 10:03 PM

This is not really an exploit... if you have local admin you can record windows sessions or use a keylogger or pretty much anything you want. It's like giving somebody root access when it's not needed (always).

Reply



**Mike** March 21, 2017 at 10:06 PM

This is not really an exploit... if you have local admin you can record windows sessions or use a keylogger or pretty much anything you want. It's like giving somebody root access when it's not needed (always).

Reply

**Anonymous** March 22, 2017 at 4:24 AM

Can this technique be used remotely? I don't see a remote parameter for tscon.



Ce site utilise des cookies provenant de Google pour fournir ses services et analyser le trafic. Votre adresse IP et votre user-agent, ainsi que des statistiques relatives aux performances et à la sécurité, sont transmis à Google afin d'assurer un service de qualité, de générer des statistiques d'utilisation, et de détecter et de résoudre les problèmes d'abus.

EN SAVOIR PLUS OK!

If you still think this is a security issue, let me give you another "0 day" for your next blogpost: on Linux, you may use a live CD in order to become root, and then if you're root you can "su" any user without knowing his password. You're welcome.

Ps: for your "a local user may become a local administrator using a live USB/CD/whatever", let me introduce you to a new security concept: Bitlocker + TPM. Can be enabled through GPO and is enabled in lots of large companies.

Reply

▼ Replies



**nopernik** March 22, 2017 at 11:09AM

You're right.



**Carlos** March 22, 2017 at 2:51PM

You don't even need bitlocker if you use SSD's like Samsung's EVO. The data is automatically encrypted \*by default\* and the factory encryption key (obviously accessible at first) can in turn be encrypted using simple, classic Class 0 BIOS password (the kind that "protected" old HDD's- but never really did). I don't know, but I think many SSD's do this as a matter of course- not just Samsung. And there are more Enterprise-managed options (Opal, for instance) on the same.

---

Reply



**Carlos** March 22, 2017 at 8:14PM

This comment has been removed by the author.

Reply

▼ Replies



**nopernik** March 22, 2017 at 8:19PM

Carlos, because of your comments count, I see that you are very interested in this "feature" :) so would you like to continue conversation via email? :) nopernik at gmail



**Carlos** March 22, 2017 at 8:25PM

Sure. I highly respect your work. Sorry for the deletes. Just wanted to clarify my post (below). Feel free to respond to the thread.

---

Reply



**Carlos** March 22, 2017 at 8:22PM

BTW, couple of food-for-thought things:

1) I wonder if Microsoft's remote tools like remote MSC or remote PowerShell sessions can be hijacked locally by a Local Admin? I don't think either establish local "user sessions" but I could be wrong.

2) I would think remote CMD (WinRM) would suffer similarly to RDP since I think that gives you the option of loading the domain user's \*local\* profile environment on that machine (C:\Users\...Desktop etc) and the profile would be created on the fly if it doesn't exist (just like RDP does).

3) How about VNC sessions? Can they be hijacked by Local Admin? I would think yes. Things like VNC and TeamViewer are especially problematic because they install privileged System Windows Services with console interactive rights (the login screen) as opposed to RDP, which does not do that (although it \*appears\* to do it

Ce site utilise des cookies provenant de Google pour fournir ses services et analyser le trafic. Votre adresse IP et votre user-agent, ainsi que des statistiques relatives aux performances et à la sécurité, sont transmis à Google afin d'assurer un service de qualité, de générer des statistiques d'utilisation, et de détecter et de résoudre les problèmes d'abus.

EN SAVOIR PLUS OK!

this does not works under Windows Server 2016 Datacenter, running a full RDP server.

I watched your video and instructions and did exactly the same psexec command. The outcome is that task manager or prompt is opened, but inside the user session. It does not "pop out" on my session just like your video, so it is useless. And yes, I am a local admin and domain admin, so that is not a priviledge problem.

But I was able to reproduce under my another Server 2012R2 RDP server.

Reply

▼ Replies



**nopernik** March 24, 2017 at 3:31AM

Specially for you <https://youtu.be/bbTfN5geSKw>

---

Reply

**Luiz** March 23, 2017 at 9:23PM

this does not works under Windows Server 2016 Datacenter, running a full RDP server.

I watched your video and instructions and did exactly the same psexec command. The outcome is that task manager or prompt is opened, but inside the user session. It does not "pop out" on my session just like your video, so it is useless. And yes, I am a local admin and domain admin, so that is not a priviledge problem.

But I was able to reproduce under my another Server 2012R2 RDP server.

Reply

**Anonymous** March 28, 2017 at 3:03PM

Unfortunately, you're not a right. it's also working on w2016. I have already tested on .. below

OS Name: Microsoft Windows Server 2016 Datacenter

OS Version: 10.0.14393 N/A Build 14393

Reply

**Anonymous** July 31, 2018 at 6:52PM

the service won't start even when i am running as NT System. any advice how to fix this in windows 10?

Reply



**Unknown** August 16, 2018 at 9:13AM

everytime i use SERVICE i can't get it to start despite the SERVICE exist, tried this on win7 and win10 have an idea what might be wrong?

Reply



**Unknown** August 16, 2018 at 9:14AM

I have tried this on win7 and win10 using the SERVICE option, but the SERVICE refuse to start anytime i tried it. have a clue what might be wrong? even starting SERVICE from NT AUTHORITY via cmd doesn't help

Reply

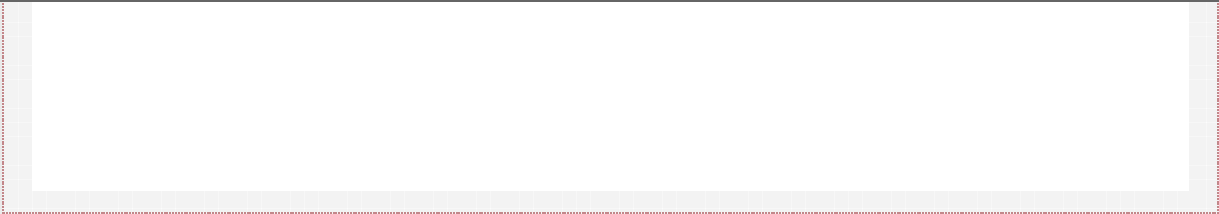
**BloggingGuru** October 9, 2018 at 5:32PM

ohh...Thanks for sharing this amazing and informative article.

Reply

Ce site utilise des cookies provenant de Google pour fournir ses services et analyser le trafic. Votre adresse IP et votre user-agent, ainsi que des statistiques relatives aux performances et à la sécurité, sont transmis à Google afin d'assurer un service de qualité, de générer des statistiques d'utilisation, et de détecter et de résoudre les problèmes d'abus.

EN SAVOIR PLUS OK !



Newer Post

Home

Older Post

Subscribe to: Post Comments (Atom)

Powered by [Blogger](#).