👛👥 Token Impersonation

Intro

When we authenticate on a windows system a login session associated with our user is created and then we can execute processes on that machine. These processes have an **access token** associated with the privileges we will have when accessing different OS objects. And the access token always makes references, through the Session ID, to the *logon session* **where the credentials are**. So if we need the credential, for example, to access a network resource or do any other thing, ***it will solicit those credentials from the logon session.***

So, What is the purpose to these technique of 👛👥**Token Impersonation?** Basically, the goal is that *as a user who has local administrator privileges* and ***can copy access tokens***, we are going to **1) take the access token of another user who has an active logon session in the system**, **2) we are going to copy their access token** and finally **3) are going to use it to make requests on their behalf.**

👛👥 Token Impersonation

To that we are going to use **Metasploit,** assuming we have compromised a user's credentials, being the plain text credentials or NTLM hash, and using **Meterpreter**, because this payload has some **built-in** functions to do token impersonation. For this case, to get the **Mertrepreter** we are going to use the module **windows/smb/psexec**.

Once we've got it our **Mertrepreter** if we made a **getuid (whoami on powershell)** we will see that we are ***NT Authority System***, because the credentials that user we had compromise high privilege *in that machine* and metasploit automatically elevates the privilege.

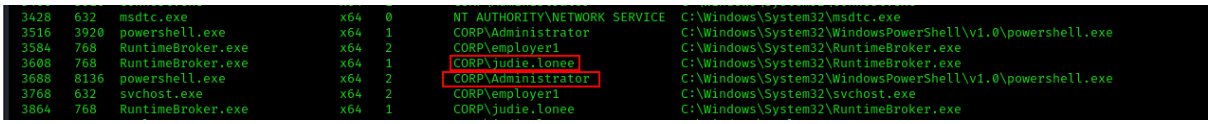
But we have privileges in that *Local Machine* but **that doesn't mean we have privileges in the domain**. For example, if we try to list **\c$** folder of the **Domain Controller** probably the result will be that

So, what can we do? Well, as we already have a Local Administration Session we can **Copy Tokens.** So let's suppose that a domain user we have connected remotely, maybe usin

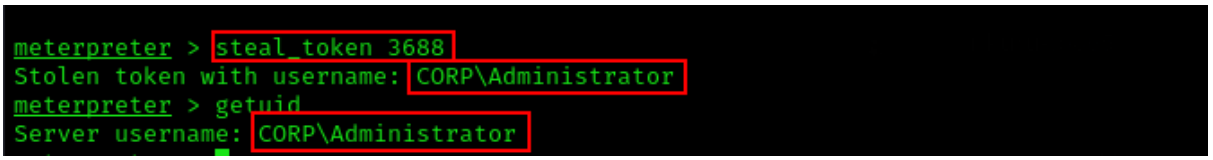
winrm, rdp, powershell or any other protocol… In short, you have an ***interactive session*** on the machine that we have compromised. Therefore, it has a **logon session**, **an access token associated** to the process that it is used for and that access token is ***referenced to the logon session of that user. We are his credentials.***

So let's suppose that we cannot crack the credentials of his hash or we use **Pass the hash** and it didn't work.

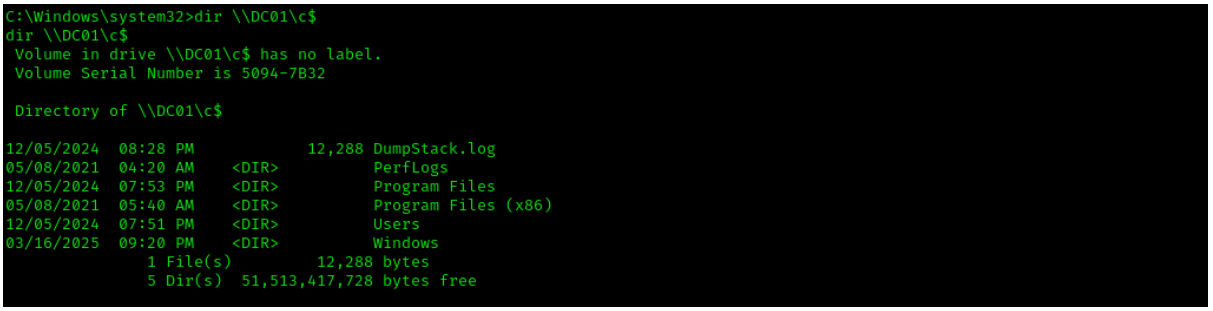
Well one thing that we can do is **find all process that are active on our machine** and **which users are executing those process** with the **Meterpreter** command **ps**



Once we have locate the process of a user that we are interested we can do the command **steal_token** or the command **migrate** followed by the **PID** and as we have ***High Local Privs*** the result will be:



Look, we have **taken the token** of the user that we are interested in, and are running a process on our machine. Now, since we have the ***access token*** that refers to this user, we can ***make domain requests on behalf of that user*****,** in this case the **Domain Administrator**. Do you remember where we tried to list the **c$** folder of the **Domain Controller**? What would happen if we tried it now that we have the **DC Admin token**? The result will be this:



We have successfully listed the **c$** folder of the **DC** because we have used the **access token** of the **Domain Admin,** because we have used the technique named **Token Impersonation**.

Note 📝: To reuse our user's token we can do it with the command **rev2sel**