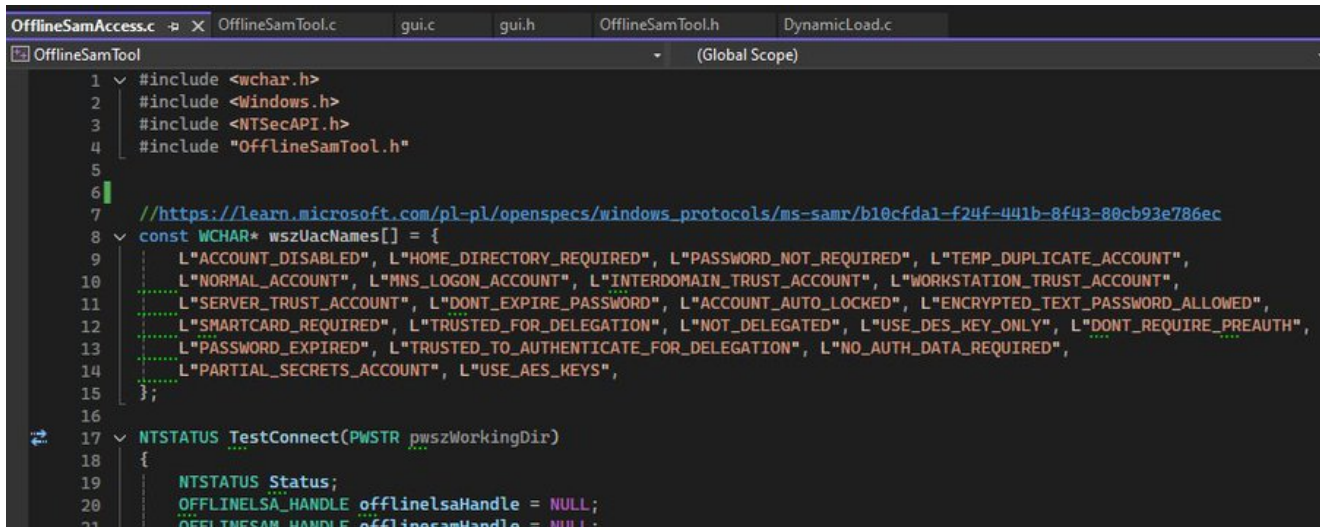


(5) X

 x.com/0gtweet/article/1824019000528891927



```
OfflineSamAccess.c  OfflineSamTool.c  gui.c  gui.h  OfflineSamTool.h  DynamicLoad.c
OfflineSamTool (Global Scope)
1  #include <wchar.h>
2  #include <Windows.h>
3  #include <NTSecAPI.h>
4  #include "OfflineSamTool.h"
5
6
7  //https://learn.microsoft.com/pl-pl/openspecs/windows_protocols/ms-samr/b10cfdal-f24f-441b-8f43-80cb93e786ec
8  const WCHAR* wszUacNames[] = {
9      L"ACCOUNT_DISABLED", L"HOME_DIRECTORY_REQUIRED", L"PASSWORD_NOT_REQUIRED", L"TEMP_DUPLICATE_ACCOUNT",
10     L"NORMAL_ACCOUNT", L"MNS_LOGON_ACCOUNT", L"INTERDOMAIN_TRUST_ACCOUNT", L"WORKSTATION_TRUST_ACCOUNT",
11     L"SERVER_TRUST_ACCOUNT", L"DONT_EXPIRE_PASSWORD", L"ACCOUNT_AUTO_LOCKED", L"ENCRYPTED_TEXT_PASSWORD_ALLOWED",
12     L"SMARTCARD_REQUIRED", L"TRUSTED_FOR_DELEGATION", L"NOT_DELEGATED", L"USE_DES_KEY_ONLY", L"DONT_REQUIRE_PREAUTH",
13     L"PASSWORD_EXPIRED", L"TRUSTED_TO_AUTHENTICATE_FOR_DELEGATION", L"NO_AUTH_DATA_REQUIRED",
14     L"PARTIAL_SECRETS_ACCOUNT", L"USE_AES_KEYS",
15 };
16
17 NTSTATUS TestConnect(PWSTR pwszWorkingDir)
18 {
19     NTSTATUS Status;
20     OFFLINELSA_HANDLE offlinesaHandle = NULL;
21     OFFLINESAM_HANDLE offlinesamHandle = NULL;
```

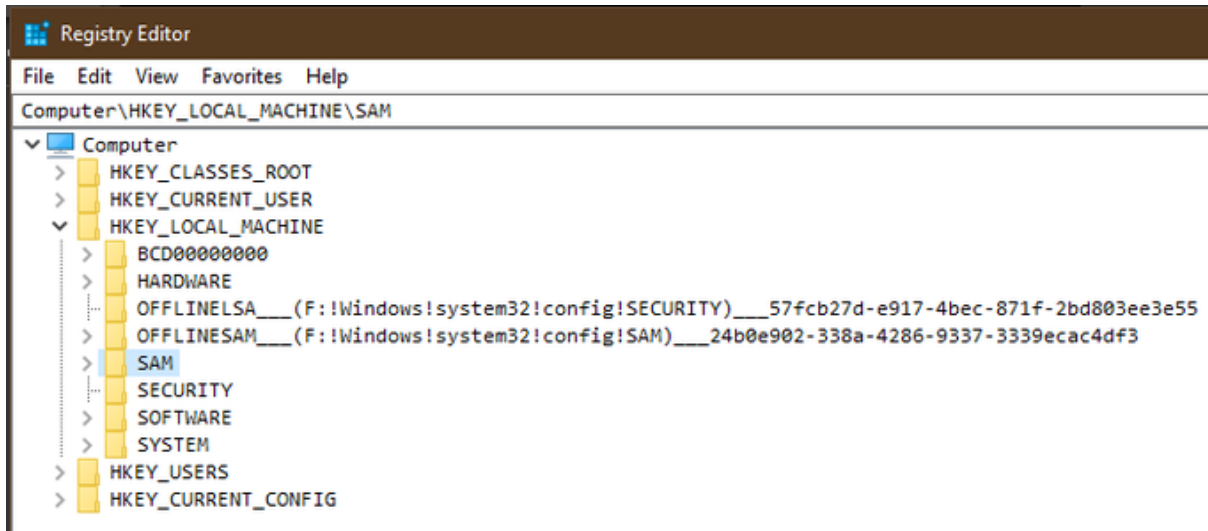
Offline SAM Editing



ilil
39K

The crucial role of the SAM database makes it an obvious target for manipulations, especially offline ones. It's why different reversing approaches happened since '90s, not always following changes in the database format. Starting from Windows 10, the situation changed a bit, as each system is fitted with special DLLs allowing any application to call dedicated methods focused on SAM editions. The difference between old and new approaches is significant: no more reversing, no more bits and bytes guessing, just politely asking the DLL to do the magic on its own. Loading DLLs from the system where SAM exists makes them perfectly match the database format.

The way how these DLLs are working is very interesting on its own: instead of file-level manipulations, both DLLs load the SAM database under a temporary registry key (OFFLINESAM for SAM and OFFLINELSA for SECURITY keys), each of which contains additionally the path and GUID to make them unique. On the other hand, it also means that you cannot load the same SAM database twice at the same time and that is not unloaded automatically even if the editing process terminates. Of course, manual closing works as expected, but sometimes it may be easy to miss it.



Offline keys loaded to registry

After loading keys, Windows accesses them exactly the same way real (online) SAM/LSA databases are accessed. It's yet another reason why it just works.

The disadvantage of relying on DLLs is that you are contained into methods exported. If something (such as allowed logon hours management, SID change, or a group rename) is not implemented, it will not work, even if theoretically should be possible when the SAM is manipulated manually.

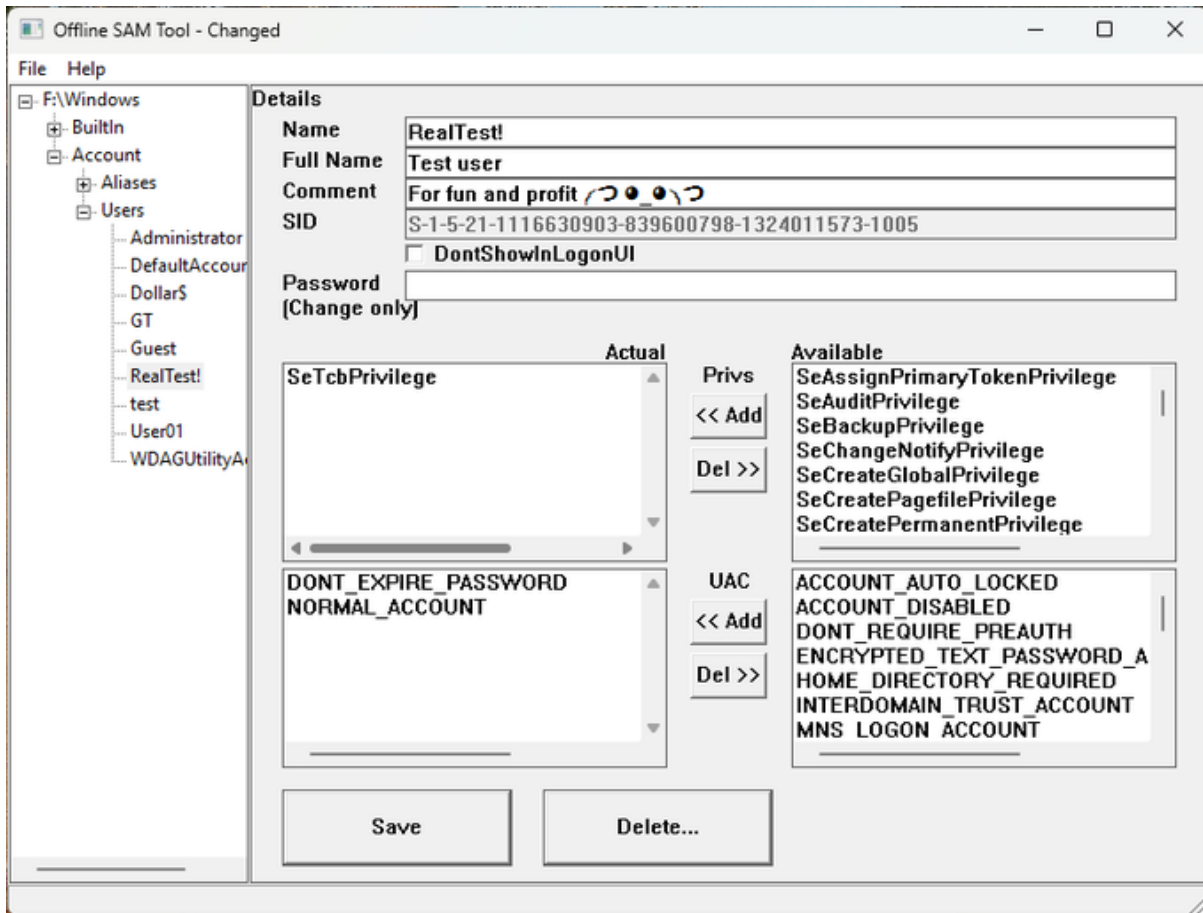
Even the "limited" approach seems to be interesting enough, as it allows to:

- add and remove groups and accounts,
- change comments, descriptions, full names and names,
- set users passwords,
- change all UAC bits,
- change privileges of groups and accounts,
- change group membership.

The only thing to do is to call DLLs, which may be not easy, but should be doable if you analyze my two small opensource tools published at

<https://github.com/gtworek/PSBits/tree/master/OfflineSAM>

And what is even more exciting, the new all-in-one tool is coming soon.



OSET (Offline SAM Editing Tool) screenshot

You can also see it in a "it's working on my machine" version on YT:

<https://youtu.be/6ay5s7fGhrk>

-

