Practical Cryptography for Infosec Noobs

Art by @DenUngeHerrHolm
https://denungeherrholm.smugmug.com/Other-Arts/Random-Arts/i-XstM6ZL/A

Practical Cryptography for Infosec Noobs

like me

Art by @DenUngeHerrHolm
https://denungeherrholm.smugmug.com/Other-Arts/Random-Arts/i-XstM6ZL/A

Practical Cryptography for Infosec Noobs

like me

90 slides in
20 minutes

Art by @DenUngeHerrHolm
https://denungeherrholm.smugmug.com/Other-Arts/Random-Arts/i-XstM6ZL/A

Crypto is easier than you think

Crypto still sucks...

TL;D(L?)

Thanks for coming to my TED talk...


E|CES™
EC-COUNCIL CERTIFIED ENCRYPTION SPECIALIST


EXPIRED
EC-COUNCIL CERTIFIED ENCRYPTION SPECIALIST

Complaints

- Cryptography is hard. It’s a lot of math.
- You have to be good at reverse engineering to do practical crypto
- No one talks about cryptography used in real applications and malware...

Tools of the Trade - CTF Specific

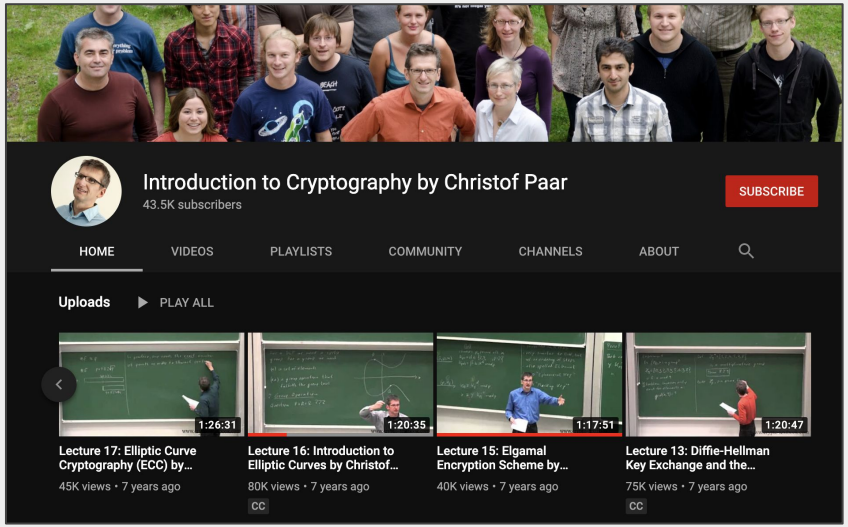
- FeatherDuster - https://github.com/nccgroup/featherduster
- RSATool - https://github.com/ius/rsatool
- RsaCtfTool - https://github.com/Ganapati/RsaCtfTool
- QuipQuip - https://quipqiup.com/
- BinaryNinja / IDA / Radare2 / Ghidra

like everything else in infosec...

and now that song is stuck in every parent’s head
… you’re welcome… :)

Best Crypto Training

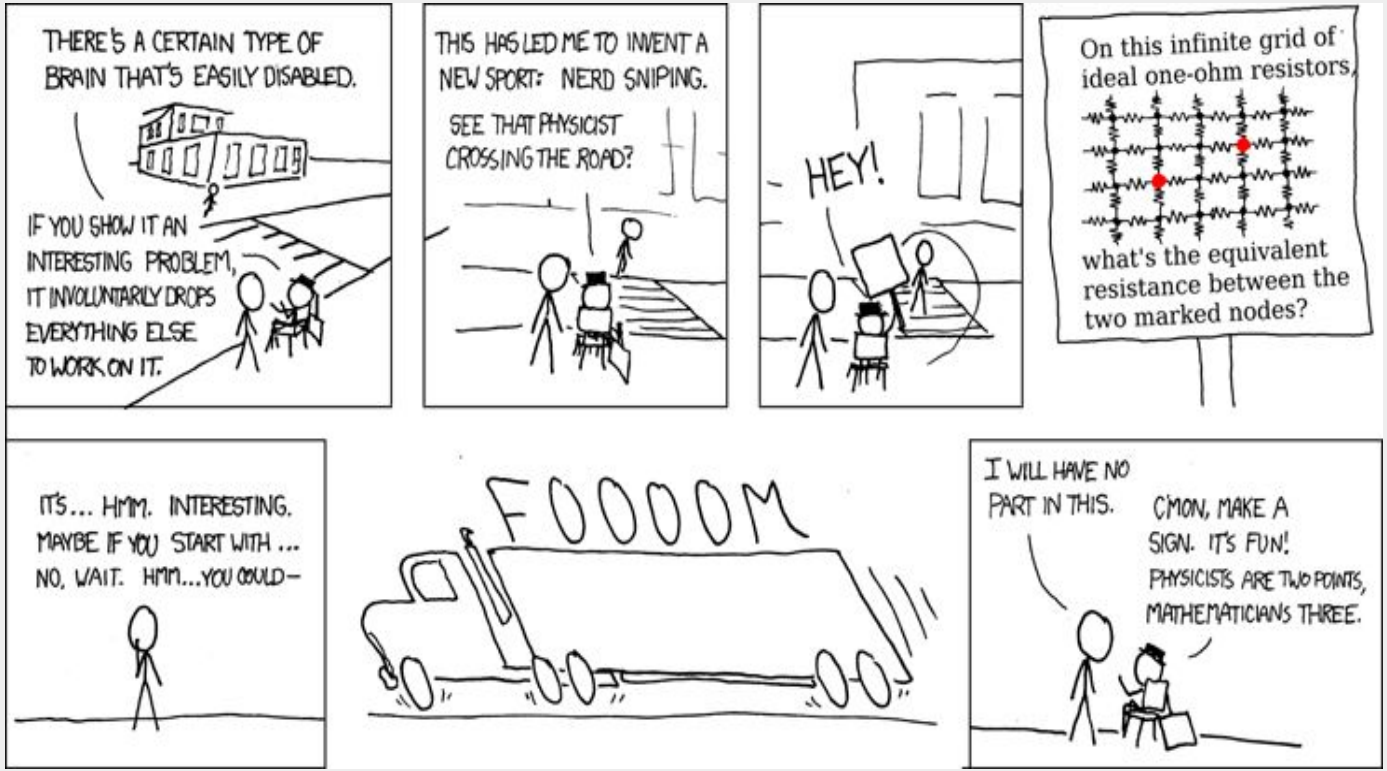
Intro to Cryptography by Christof Paar



https://www.youtube.com/channel/UC1usFRN4LCMcfIV7UjHNuQg

Nerd Sniping

https://xkcd.com/356/



Hey Rob, what's this?

CPUU.ini

```
[Credentials]
Admin=domain1\test1
Password=0C0101020000106600000A4000000D0429CEBC3CB4FF3BFC47341BD8B54DB74DA3F46FF4754AA9057B26F73EB2
5D92CA4E1292BE9EC6C44290F7497770F400E2A12C881C86A17AA3BEAED7D80E80F70119BCA6D87DB2FFC993126227EAD1E1
A9E6B81FCD354C38A4FD5AB392E7500E81E39CA8C0AF6A4EE3527D7B7D19DAAD95B2FB1847B7F85D45325EFE2111145564C8
99DC82D78CBA14E813E60BBB442B88D2106210719F9EEA90C8F048C36BF24DB7A32D64268A5462594A6E62540343F0B7B98E
60EDAF1824A16A12AA5ACCB4543BF59334CCF6539667D61528ACE6178A5FB7778901DAA6DD5CCBEA7DA12405F41C9F87CC05
952AF766298E204F4080C5B26AFC5A16422ED5312D0A7D03B4BDC7841BEA7DE0FF93C6809F4A8921ED816F33C4F3AE68F1CC
2E820F792048F
AdminTrg=domain2\test2
PasswordTrg=0C0101020000106600000A400003E45061A399EA74CA2A44E5686D0B0F002C65841669B49770E99D9AA2374
48F5417D52FF23C6027658A95EFD182F651EA92691F1996B49D80E445468417DA59B138390A495050A94FD70AD6738B5F54B
72564558D53AE0C6E174857B039D6F24B7AD5455B88E54F735B9B85CE196E6C45D9C7B14EB4CB6BCAD2289F0D58304EC2770
BC21668E919A93FB425A4FE7ABD704F7437FEC1694279DAF2F4037E235A124294438C6AAE560BF49A87F790E29DA6687D9BE
276B1C18D33398D1758B1B57CC45DE204A6407072C261972EF1FFCF55E8EE95D5E065A015E4D9B45D159862954B388BA6CC4
7386E508016108A32A489E1E52F1FE14D359ABA3C183451ED66BA721929CC7AB5D0E075D4860C7823917CB101AA585724C90
1313BC423586351E
AskCredentials=0
EncryptionMode=Unrestricted
AskSourceCredentialsOnly=0
```

CPUU.ini

What is CPUU?

Overview

During migration, Migration Manager for Exchange moves mailboxes from the source Microsoft Exchange server to the target Microsoft Exchange server. Before users can start working with the new target mailbox, their Microsoft Outlook profiles must also be updated.

The **Client Profile Updating Utility (CPUU; legacy name: EMWProf) 5.8.X** allows you to update these profiles automatically and transparently. The utility is used to update end-user Microsoft Outlook profiles settings, migrate additional features of user mailboxes and finally switch the profiles from the source to the target Exchange server once the user's mailbox is migrated and switched either manually or by the Migration Manager's Mail Agent.

CPUU supports Microsoft Outlook 2010/2013/2016/2019 and Outlook for Office 365. It can be used in conjunction only with the following products:

- Migration Manager 8.14 or later
- On Demand Migration for Email

https://support.quest.com/technical-documents/client-profile-updating-utility/5.8/administrator-guide

TIP: For details how to use CPUU with ODME, see Working with On Demand Migration for Email

What is CPUU?

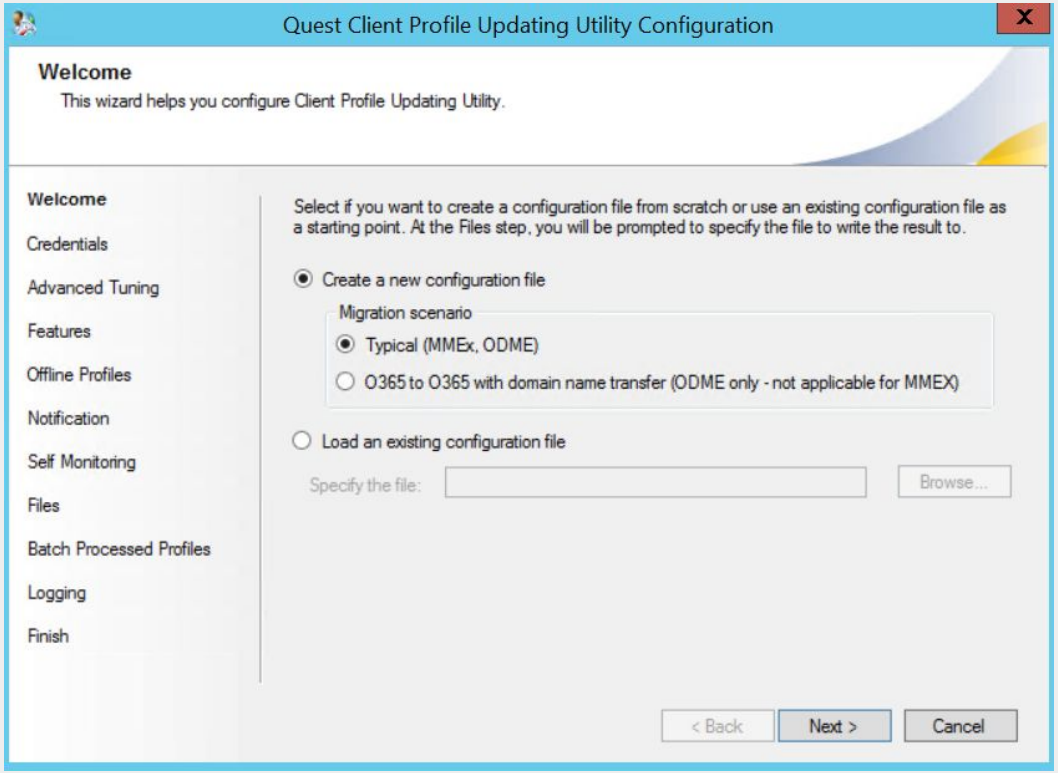
Name	Size	Modified
ClientProfileUpdatingUtility.chm	473 895	2020-03-13 21:31
ClientProfileUpdatingUtility.exe	3 956 088	2020-03-14 01:11
ClientProfileUpdatingUtilityConfiguration.exe	588 152	2020-03-14 01:11
ClientProfileUpdatingUtilityConfiguration.exe.config	335	2020-03-13 21:31
ClientProfileUpdatingUtility_x64.exe	5 686 648	2020-03-14 01:11
DlgHookHandler.dll	977 784	2020-03-14 01:11
DlgHookHandler_x64.dll	1 667 960	2020-03-14 01:11
SwitchResMB.exe	733 048	2020-03-14 01:11

What is CPUU?

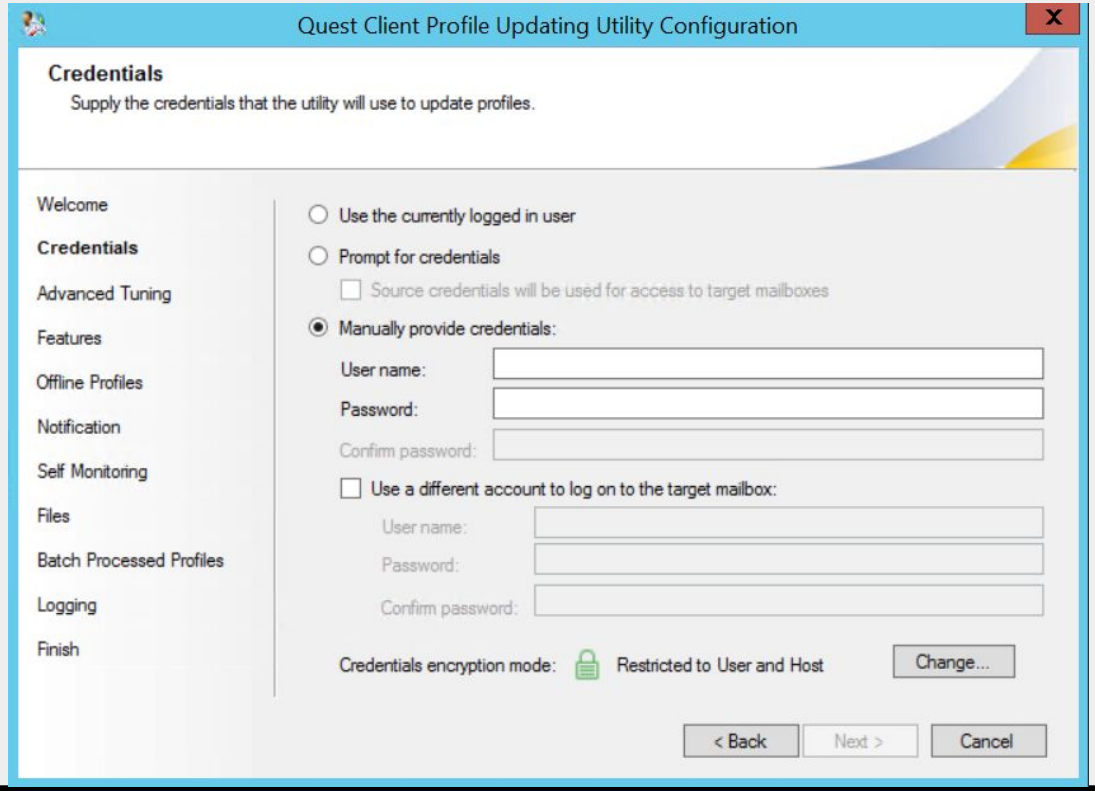
Name	Size	Modified
ClientProfileUpdatingUtility.chm	473 895	2020-03-13 21:31
ClientProfileUpdatingUtility.exe	3 956 088	2020-03-14 01:11
ClientProfileUpdatingUtilityConfiguration.exe	588 152	2020-03-14 01:11
ClientProfileUpdatingUtilityConfiguration.exe.config	335	2020-03-13 21:31
ClientProfileUpdatingUtility_x64.exe	5 686 648	2020-03-14 01:11
DlgHookHandler.dll	977 784	2020-03-14 01:11
DlgHookHandler_x64.dll	1 667 960	2020-03-14 01:11
SwitchResMB.exe	733 048	2020-03-14 01:11

C#

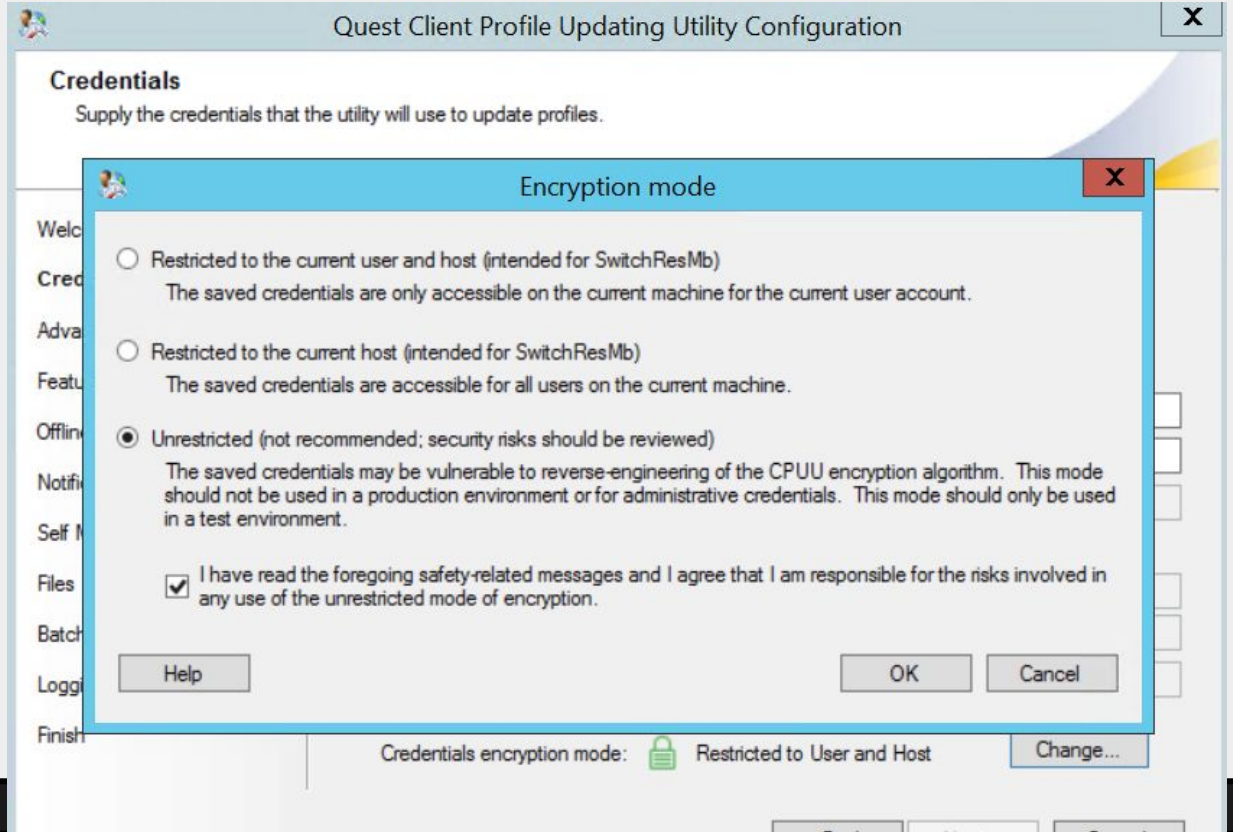
Lets run the configuration tool...



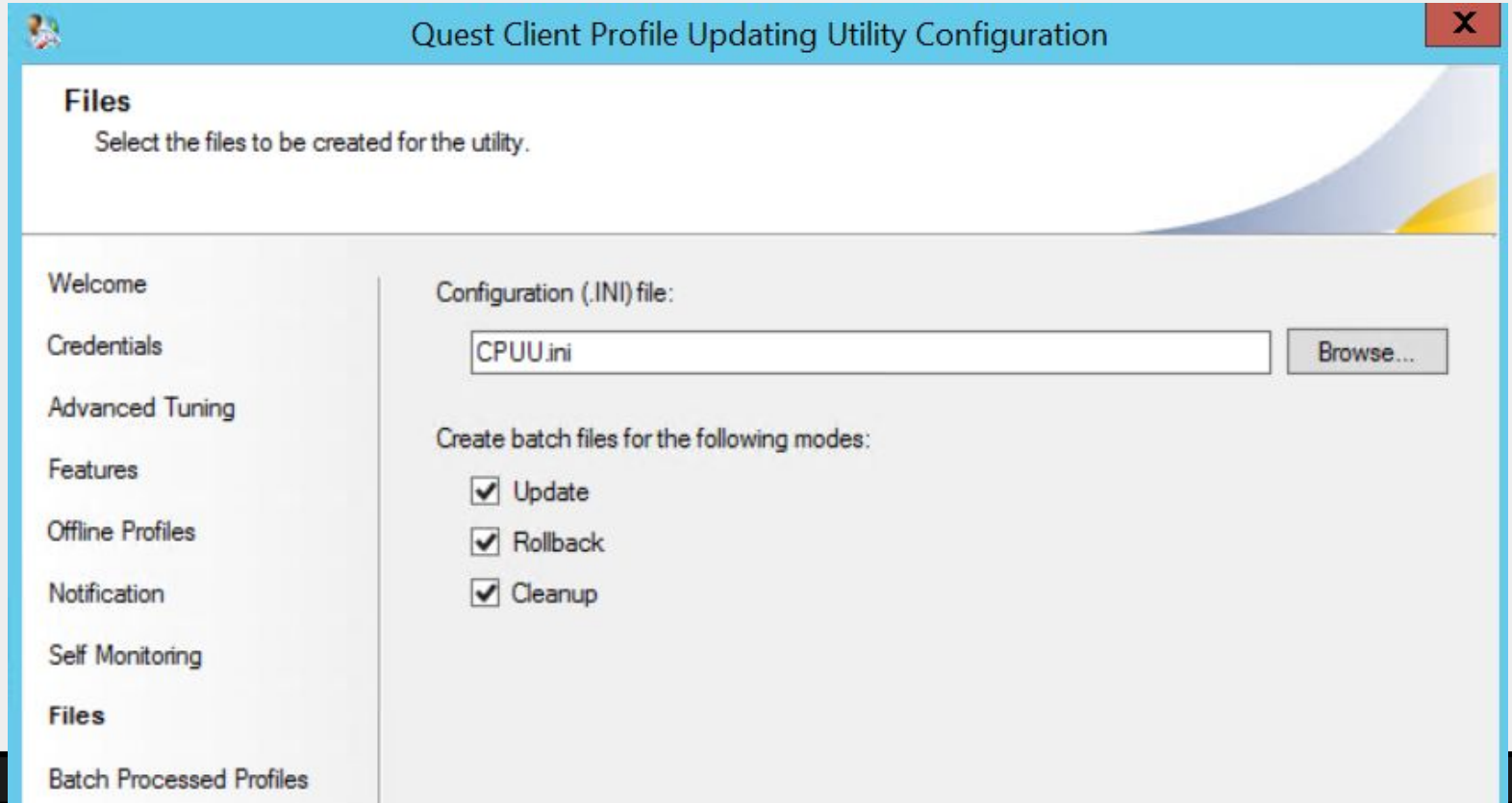
Lets run the configuration tool...



Credential Storage Options



Lets run the configuration tool...



Lets stop and appreciate good security...

1. The default is to only use the Current User session and not store credentials in the CPUU.ini file
2. If you are going to store credentials in the CPUU.ini file, the default is to have it in “Restricted to User and Host” that uses the built in Windows CryptProtectData with the user’s key.
3. It warns you and even makes you click another checkbox stating “Hey this crypto can be reversed! are you sure?!”

Rabbit Hole #1

Reverse engineering the crypto

Time to dig into the binary... (RE 4 Dummies)

```
root@kali:~/Desktop/tmp# strings ClientProfileUpdatingUtilityConfiguration.exe
  | grep -i crypt
CryptAcquireContextW
CryptReleaseContext
CryptImportKey
CryptExportKey
CryptGenKey
CryptDestroyKey
CryptEncrypt
CryptProtectData
Crypt32.dll
CryptUnprotectData
```

Time to dig into the binary... (RE 4 Dummies)

```
root@kali:~/Desktop/tmp# strings -el ClientProfileUpdatingUtilityConfiguration.exe
c"e#
'!($.%5&9':(<)=*G+N,O-S.T/V0W7a8b9d:e=g?h@jAkGvQxRzS{T}W
435363YXZX`_gfhfjiki
Microsoft Enhanced RSA and AES Cryptographic Provider
CPUU_#h_#d_#t.log
Admin
Password
AdminTrg
PasswordTrg
AskCredentials
EncryptionMode
AskSourceCredentialsOnly
Features
```

De4Dot DeObfucator

https://github.com/0xd4d/de4dot

".NET deobfuscator and unpacker."

```
PS C:\Users\user\source\repos\de4dot\Release\net35> .\de4dot.exe C:\Users\user\Desktop\ClientProfileUpdatingUtilityConfi
guration.exe

de4dot v3.1.41592.3405

Detected Unknown Obfuscator (C:\Users\user\Desktop\ClientProfileUpdatingUtilityConfiguration.exe)
Cleaning C:\Users\user\Desktop\ClientProfileUpdatingUtilityConfiguration.exe
Renaming all obfuscated symbols
Saving C:\Users\user\Desktop\ClientProfileUpdatingUtilityConfiguration-cleaned.exe

Press any key to exit...
```

De4Dot DeObfucator -> DnSpy

```
10  using System;
11  using System.Runtime.CompilerServices;
12
13  [module: SuppressIldasm]
14  [module: ConfusedBy("ConfuserEx v1.0.0")]
15
```

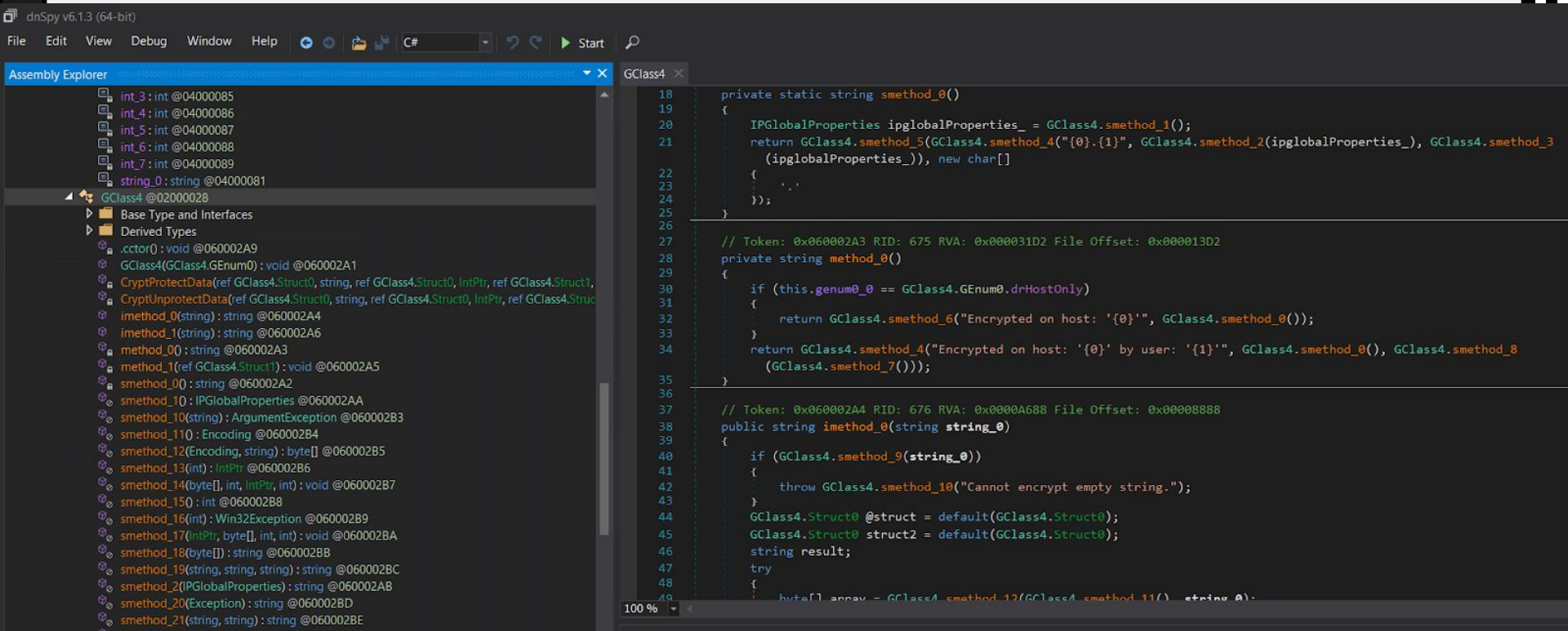
De4Dot DeObfucator + ConfuserEx Support

https://github.com/ViRb3/de4dot-cex

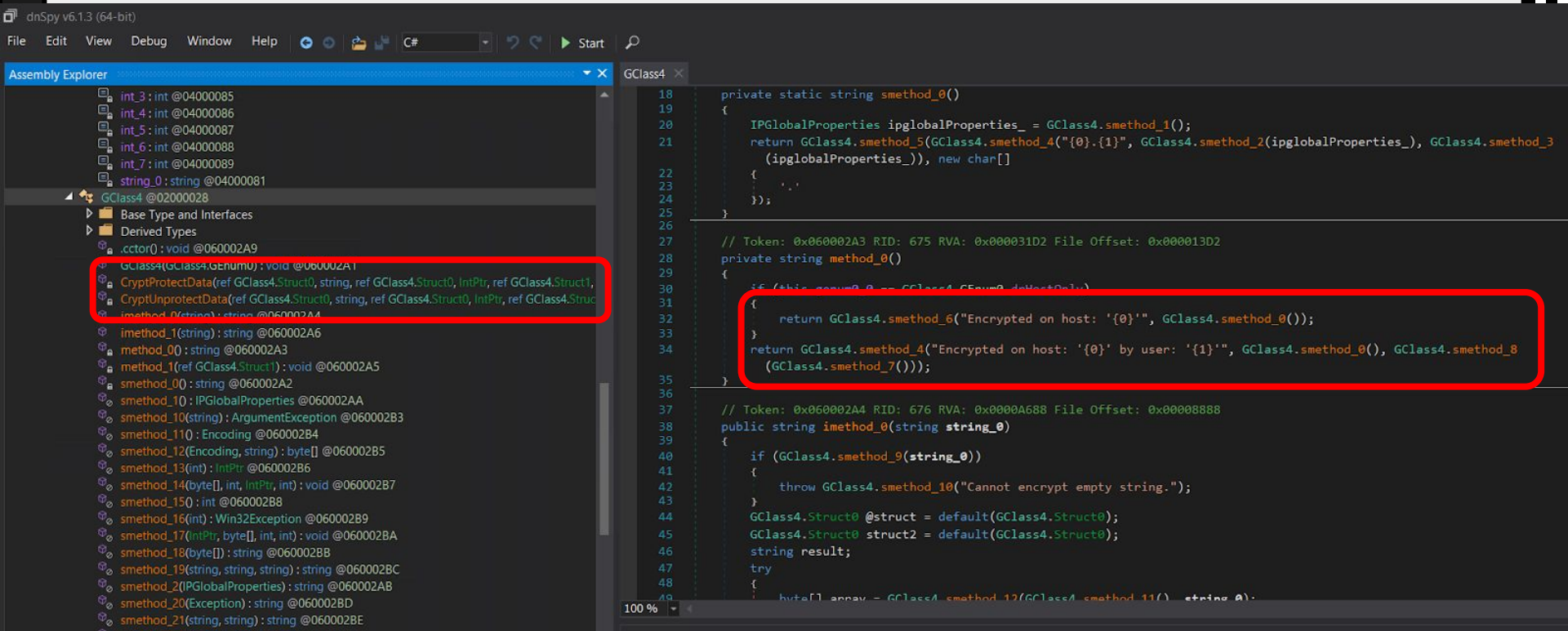
“de4dot deobfuscator with full support for vanilla ConfuserEx”

```
PS C:\Users\user\source\repos\de4dot-cex\Debug> .\de4dot.exe C:\Users\user\Desktop\ClientProfileUpdatingUtilityConfigura
tion.exe

de4dot v3.1.41592.3405 Copyright (C) 2011-2015 de4dot@gmail.com
Latest version and source code: https://github.com/0xd4d/de4dot

Detected ConfuserEx v1.0.0 (C:\Users\user\Desktop\ClientProfileUpdatingUtilityConfiguration.exe)
Cleaning C:\Users\user\Desktop\ClientProfileUpdatingUtilityConfiguration.exe
Renaming all obfuscated symbols
ERROR: Could not resolve MethodRef System.Delegate □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□`3<□,□,□>::□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□(System.Delegate,System.Delegate) (0A000055) (from ClientProfileUpdatingUtilityConfiguration.exe
 -> ClientProfileUpdatingUtilityConfiguration.exe)
Saving C:\Users\user\Desktop\ClientProfileUpdatingUtilityConfiguration-cleaned.exe
Ignored 3 warnings/errors
Use -v/-vv option or set environment variable SHOWALLMESSAGES=1 to see all messages


Press any key to exit...
```

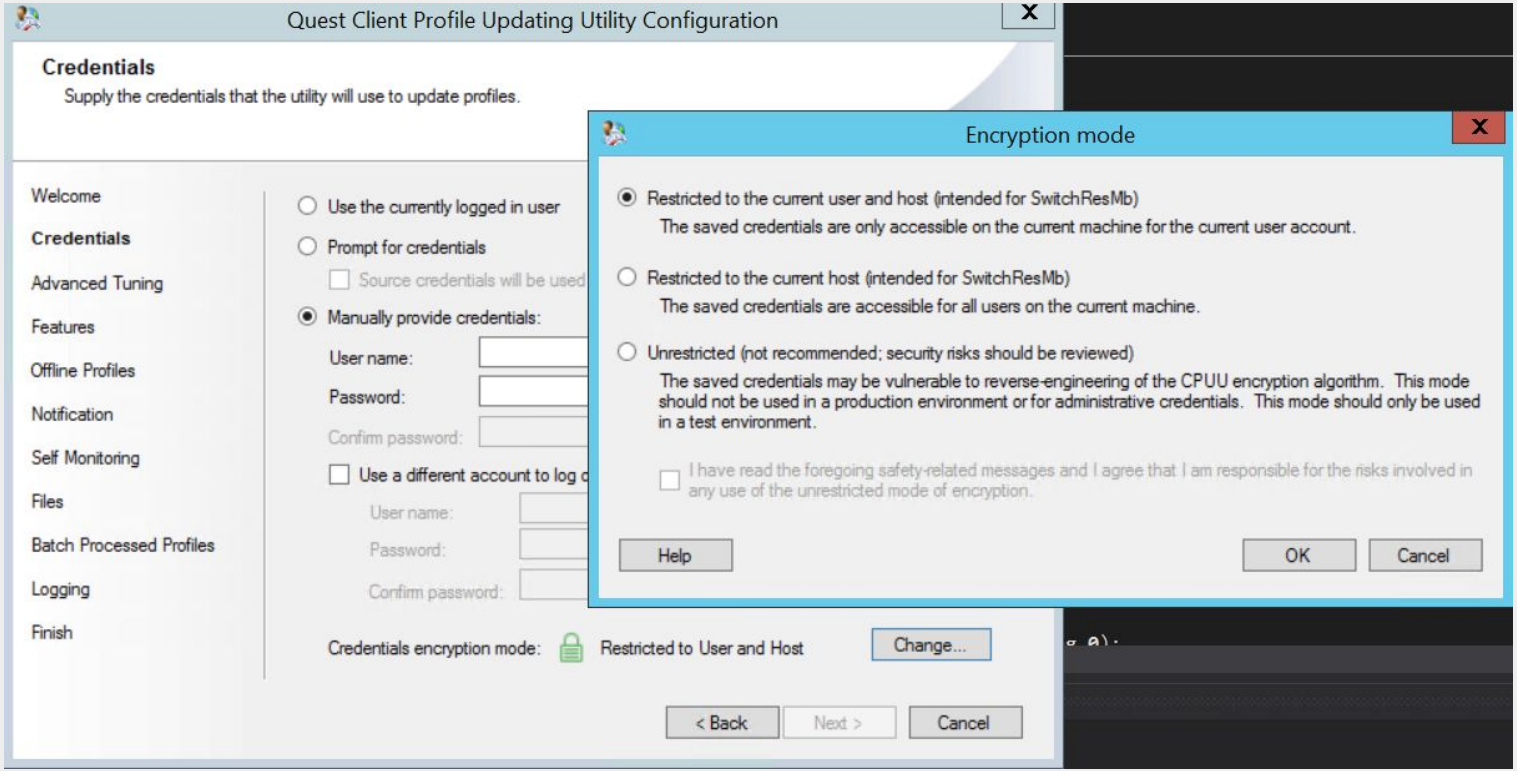
Let's play "find the crypto"



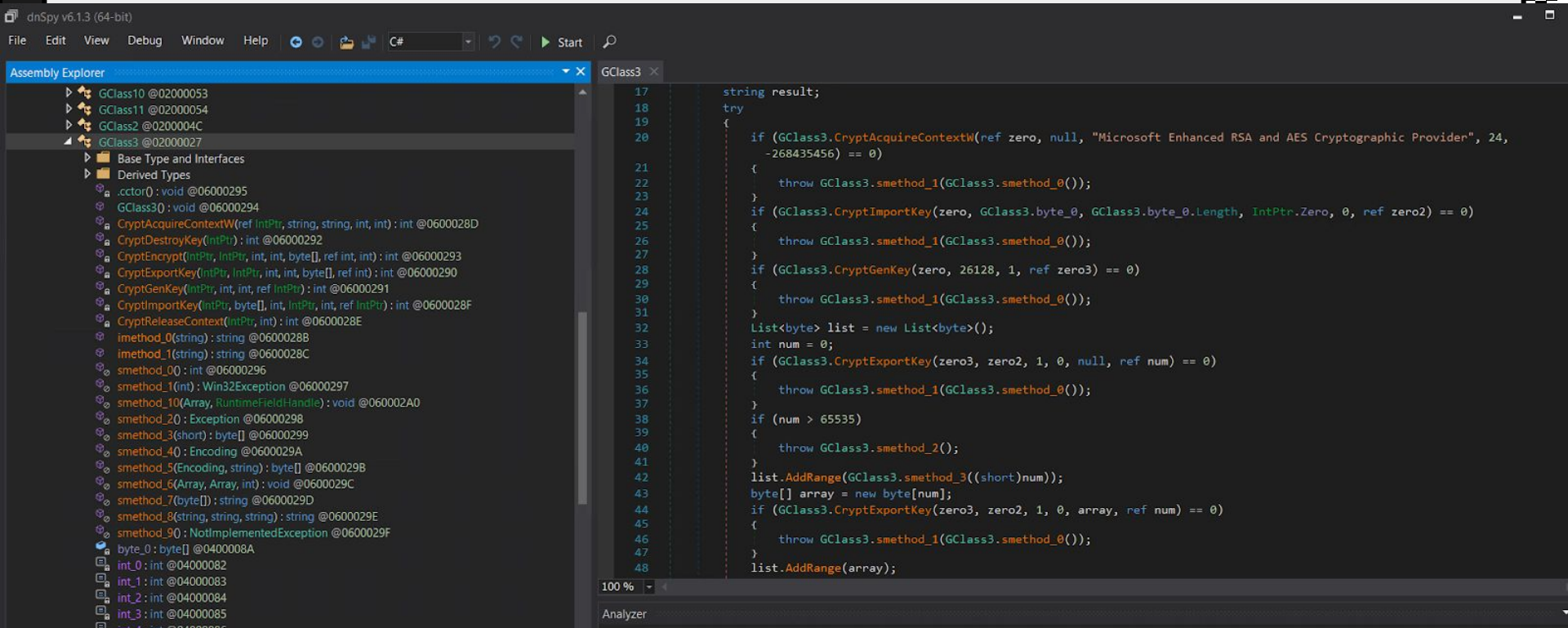
Let's play "find the crypto"



Let's play "find the crypto"



Let's play "find the crypto"



CryptEncrypt looks interesting...

```
GClass3() : void @06000294
CryptAcquireContextW(ref IntPtr, string, string, int, int) : int @0600028D
CryptDestroyKey(IntPtr) : int @06000292
CryptEncrypt(IntPtr, IntPtr, int, int, byte[], ref int, int) : int @06000293
CryptExportKey(IntPtr, IntPtr, int, int, byte[], ref int) : int @06000290
CryptGenKey(IntPtr, int, int, ref IntPtr) : int @06000291
CryptImportKey(IntPtr, byte[], int, IntPtr, int, ref IntPtr) : int @0600028F
CryptReleaseContext(IntPtr, int) : int @0600028E
```

More on this later...

The **CryptEncrypt** function encrypts data. The algorithm used to encrypt the data is designated by the key held by the CSP module and is referenced by the *hKey* parameter.

```
// Set the IV for the original key. Do not use the original key for
// encryption or decryption after doing this because the key's
// feedback register will get modified and you cannot change it.
CryptSetKeyParam(hOriginalKey, KP_IV, newIV)

while(block = NextBlock())
{
    // Create a duplicate of the original key. This causes the
    // original key's IV to be copied into the duplicate key's
    // feedback register.
    hDuplicateKey = CryptDuplicateKey(hOriginalKey)

    // Encrypt the block with the duplicate key.
    CryptEncrypt(hDuplicateKey, block)

    // Destroy the duplicate key. Its feedback register has been
    // modified by the CryptEncrypt function, so it cannot be used
    // again. It will be re-duplicated in the next iteration of the
    // loop.
    CryptDestroyKey(hDuplicateKey)
}
```

C++

```
BOOL CryptEncrypt(
  HCRYPTKEY  hKey,
  HCRYPTHASH hHash,
  BOOL       Final,
  DWORD      dwFlags,
  BYTE       *pbData,
  DWORD      *pdwDataLen,
  DWORD      dwBufLen
);
```

What the heck is CryptEncrypt?
https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-cryptdecrypt

Wait... there is no "CryptDecrypt"

```
GClass3() : void @06000294
CryptAcquireContextW(ref IntPtr, string, string, int, int) : int @0600028D
CryptDestroyKey(IntPtr) : int @06000292
CryptEncrypt(IntPtr, IntPtr, int, int, byte[], ref int, int) : int @06000293
CryptExportKey(IntPtr, IntPtr, int, int, byte[], ref int) : int @06000290
CryptGenKey(IntPtr, int, int, ref IntPtr) : int @06000291
CryptImportKey(IntPtr, byte[], int, IntPtr, int, ref IntPtr) : int @0600028F
CryptReleaseContext(IntPtr, int) : int @0600028E
```

More on this later...

Encrypt vs Decrypt

- Attacking Asymmetric encryption:
  - You need to find the Decryption functions because that is where the private key will be used. Public / Private keys etc
- Attacking Symmetric encryption:
  - You can use the Encryption **OR** Decryption functions to get at the relevant keys because they are the same in both functions

Ok, so this is RSA + AES

“Microsoft Enhanced RSA and AES CryptoGraphic Provider”

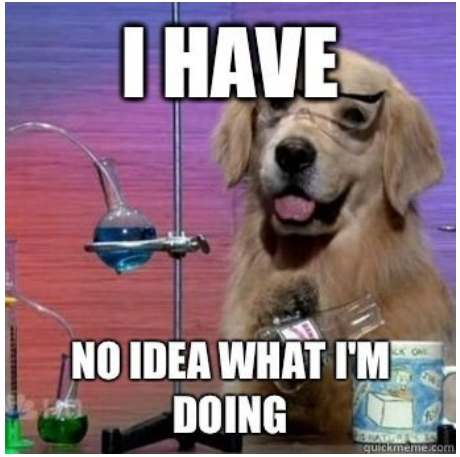
PROV_RSA_AES == 24

https://docs.microsoft.com/en-us/windows/win32/seccrypto/microsoft-aes-cryptographic-provider

```
public string imethod_0(string string_1)
{
    IntPtr zero = IntPtr.Zero;
    IntPtr zero2 = IntPtr.Zero;
    IntPtr zero3 = IntPtr.Zero;
    string result;
    try
    {
        if (GClass3.CryptAcquireContextW(ref zero, null, "Microsoft Enhanced RSA and AES Cryptographic Provider", 24,
          -268435456) == 0)
        {
            throw GClass3.smethod_1(GClass3.smethod_0());
        }
        if (GClass3.CryptImportKey(zero, GClass3.byte_0, GClass3.byte_0.Length, IntPtr.Zero, 0, ref zero2) == 0)
        {
            throw GClass3.smethod_1(GClass3.smethod_0());
        }
        if (GClass3.CryptGenKey(zero, 26128, 1, ref zero3) == 0)
        {
            throw GClass3.smethod_1(GClass3.smethod_0());
        }
        List<byte> list = new List<byte>();
        int num = 0;
        if (GClass3.CryptExportKey(zero3, zero2, 1, 0, null, ref num) == 0)
        {
            throw GClass3.smethod_1(GClass3.smethod_0());
        }
```

Ok, so this is RSA + AES

Algorithm	Base Provider key length	Strong Provider key length	AES Provider key length
RSA public key signature algorithm	512 bits	1,024 bits	1,024 bits
RSA public key exchange algorithm	512 bits	1,024 bits	1,024 bits
RC2 block encryption algorithm	40 bits	128 bits	128 bits Salt length can be set.
RC4 stream encryption algorithm	40 bits	128 bits	128 bits Salt length can be set.
DES	56 bits	56 bits	56 bits
Triple DES (2 key)	Not supported	112 bits	112 bits
Triple DES (3 key)	Not supported	168 bits	168 bits



PID: 118C - Module: advapi32.dll - Thread: Main Thread 1190 - x64dbg

File View Debug Trace Plugins Favourites Options Help Aug 16 2020

CPU | Log | Notes | Breakpoints | Memory Map | Call Stack | SEH | Script | Symbols | Source | References | Threads | Handles | Trace

```
RIP R10  00007FFD94E109E0  48:FF25 C9560700  jmp qword ptr ds:[<&CryptGenKey>]      CryptGenKey
         00007FFD94E109E7  90                nop
         00007FFD94E109E8  90                nop
         00007FFD94E109E9  90                nop
         00007FFD94E109EA  90                nop
         00007FFD94E109EB  90                nop
         00007FFD94E109EC  90                nop
         00007FFD94E109ED  90                nop
         00007FFD94E109EE  90                nop
         00007FFD94E109EF  90                nop
         00007FFD94E109F0  48:FF25 A1560700  jmp qword ptr ds:[<&CryptEncrypt>]     CryptEncrypt
         00007FFD94E109F7  90                nop
         00007FFD94E109F8  90                nop
         00007FFD94E109F9  90                nop
         00007FFD94E109FA  90                nop
         00007FFD94E109FB  90                nop
         00007FFD94E109FC  90                nop
         00007FFD94E109FD  90                nop
         00007FFD94E109FE  90                nop
         00007FFD94E109FF  90                nop
         00007FFD94E10A00  48:FF25 E1560700  jmp qword ptr ds:[<&CryptGetUserKey>]  CryptGetUserKey
         00007FFD94E10A07  90                nop
         00007FFD94E10A08  90                nop
         00007FFD94E10A09  90                nop
         00007FFD94E10A0A  90                nop
         00007FFD94E10A0B  90                nop
         00007FFD94E10A0C  90                nop
         00007FFD94E10A0D  90                nop
         00007FFD94E10A0E  90                nop
         00007FFD94E10A0F  90                nop
         00007FFD94E10A10  48:FF25 29560700  jmp qword ptr ds:[<&CryptSignHashA>]  CryptSignHashA
         00007FFD94E10A17  90                nop
         00007FFD94E10A18  90                nop
         00007FFD94E10A19  90                nop
         00007FFD94E10A1A  90                nop
         00007FFD94E10A1B  90                nop
         00007FFD94E10A1C  90                nop
         00007FFD94E10A1D  90                nop
         00007FFD94E10A1E  90                nop
         00007FFD94E10A1F  90                nop
         00007FFD94E10A20  90                nop
         00007FFD94E10A21  90                nop
         00007FFD94E10A22  90                nop
         00007FFD94E10A23  90                nop
         00007FFD94E10A24  90                nop
         00007FFD94E10A25  90                nop
         00007FFD94E10A26  90                nop
         00007FFD94E10A27  90                nop
         00007FFD94E10A28  90                nop
         00007FFD94E10A29  90                nop
         00007FFD94E10A2A  90                nop
```

Hide FPU

```
RAX  0000000000000000
RBX  0000000000000001
RCX  000000001BF2FC70   <&CPAcquireContext>
RDX  0000000000006610
RBP  0000000000E7D700
RSP  0000000000E7D6B8
RSI  0000000000E7D7A0
RDI  0000000001009360

R8   0000000000000001
R9   0000000000E7D780
R10  00007FFD94E109E0   <advapi32.CryptGenKey>
R11  0000000000E7D7A0
R12  0000000001009360
R13  000000001BF2FC70   <&CPAcquireContext>
R14  0000000000000000
R15  0000000002F46380

RIP  00007FFD94E109E0   <advapi32.CryptGenKey>

RFLAGS   0000000000000344
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 1  IF 1

LastError   00000000 (ERROR_SUCCESS)
LastStatus  00000000 (STATUS_SUCCESS)

GS 002B  FS 0053
ES 002B  DS 002B
CS 0033  SS 002B

ST(0) 098D000000000000040F x87r0 Empty invalid
ST(1) 0163000000000F90000 x87r1 Empty invalid
ST(2) 044000000000000040F x87r2 Empty invalid
ST(3) DFB80000000000000000 x87r3 Empty invalid
ST(4) 00000000000000E7E340 x87r4 Empty 5.539606812442171147
ST(5) 69270000000000000578 x87r5 Empty invalid
ST(6) 0002FFFFFFFFFFFFFFFE x87r6 Empty 1.344841257244837402
```

Default (x64 fastcall) 5 Unlocked

```
1: rcx 000000001BF2FC70 <&CPAcquireContext>
2: rdx 0000000000006610
3: r8 0000000000000001
4: r9 0000000000E7D780
5: [rsp+28] 000000005B3072A8
```

```
qword ptr ds:[00007FFD94E860B0 <advapi32.&CryptGenKey>]=<cryptsp.CryptGenKey>

.text:00007FFD94E109E0 advapi32.dll:$309E0 #2FDE0 <CryptGenKey>
```

Dump 1 | Dump 2 | Dump 3 | Dump 4 | Dump 5 | Watch 1 | [x=] Locals | Struct

```
Address          Hex                                              ASCII
00007FFD96D11000 00 46 01 00 E0 49 01 00 30 4B 01 00 C0 4B 01 00  .F..àI..0K..ÀK..
00007FFD96D11010 80 4E 01 00 A0 4E 01 00 F0 4F 01 00 C0 52 01 00  .N.. N..ðO..ÀR..
00007FFD96D11020 D0 52 01 00 C0 54 01 00 60 58 01 00 10 5A 01 00  ÐR..ÀT..`X...Z..
00007FFD96D11030 10 5D 01 00 B0 5F 01 00 10 60 01 00 B0 61 01 00  .]..°_...`..°a..
00007FFD96D11040 90 65 01 00 D0 66 01 00 90 68 01 00 A0 76 01 00  .e..Ðf...h.. v..
00007FFD96D11050 B0 76 01 00 E0 82 01 00 50 84 01 00 60 85 01 00  °v..à...P...`...
00007FFD96D11060 C0 85 01 00 50 86 01 00 80 8A 01 00 40 8C 01 00  À...P.......@...
00007FFD96D11070 10 8D 01 00 B0 8D 01 00 C0 8F 01 00 90 90 01 00  ....°...À.......
00007FFD96D11080 80 96 01 00 10 99 01 00 20 9C 01 00 E0 9E 01 00  ........ ...à...
00007FFD96D11090 60 A0 01 00 90 A0 01 00 80 A8 01 00 C0 AF 01 00  `... ... ...À...
00007FFD96D110A0 D0 B3 01 00 30 B4 01 00 D0 B6 01 00 10 BD 01 00  Ð³..0´..Ð¶...½..
00007FFD96D110B0 A0 C1 01 00 F0 C2 01 00 80 C4 01 00 C0 C5 01 00  Á..ðÂ...Ä..ÀÅ..
00007FFD96D110C0 90 C7 01 00 60 C8 01 00 F0 CA 01 00 70 CF 01 00  .Ç..`È..ðÊ..pÏ..
00007FFD96D110D0 C0 D1 01 00 D0 D2 01 00 70 D4 01 00 A0 D7 01 00  ÀÑ..ÐÒ..pÔ.. ×..
00007FFD96D110E0 E0 D7 01 00 70 D9 01 00 30 DA 01 00 90 DC 01 00  à×..pÙ..0Ú...Ü..
00007FFD96D110F0 40 DE 01 00 50 DE 01 00 60 E1 01 00 10 E2 01 00  @Þ..PÞ..`á...â..
00007FFD96D11100 50 E2 01 00 60 E9 01 00 20 ED 01 00 10 F1 01 00  Pâ..`é.. í...ñ..
00007FFD96D11110 00 F3 01 00 D0 FD 01 00 50 FF 01 00 90 01 02 00  .ó..Ðý..Pÿ......
```

```
0000000000E7D6B8  00007FFD6F87A847  return to mscorwks.00007FFD6F87A847 fro
0000000000E7D6C0  FFFFFFFF00000114
0000000000E7D6C8  000000000F99F30
0000000000E7D6D0  0000000000000000
0000000000E7D6D8  00007FFD1007CFF8
0000000000E7D6E0  000000053B3072A8
0000000000E7D6E8  0000000000016F0
0000000000E7D6F0  0000000000E7D8B8
0000000000E7D6F8  0000000000000002
0000000000E7D700  0000000000000000
0000000000E7D708  0000000000000035
0000000000E7D710  0000000002F46380
0000000000E7D718  00007FFD6F8754E3  return to mscorwks.00007FFD6F8754E3 fro
0000000000E7D720  0000000002F46380
0000000000E7D728  0000000000000000
0000000000E7D730  000000001BF2FC70
0000000000E7D738  0000000000006610
0000000000E7D740  000000002AA16BCC
0000000000E7D748  0000000000006610
0000000000E7D750  0000000000E7D8A8
```

Command: Default

Paused | INT3 breakpoint at <advapi32.CryptGenKey> (00007FFD94E109E0)! | Time Wasted Debugging: 0:02:08:02

HELP!!!

ding! ding! ding! winner!

Rohitab API Monitor - http://www.rohitab.com/apimonitor

10	4:34:34.683 PM	1	mscorwks.dll	CryptAcquireContextW (0x000000000096d3c0, NULL, "Microsoft Enhanced RSA and AES Cryptographic Provider", PROV_RSA_AES, C...
11	4:34:34.683 PM	1	mscorwks.dll	CryptImportKey (0x000000001b7eed80, 0x0000000029fe2f0, 276, NULL, 0, 0x000000000096d3d0)
12	4:34:34.683 PM	1	mscorwks.dll	CryptGenKey (0x000000001b7eed80, CALG_AES_256, 1, 0x000000000096d3c0)
13	4:34:34.683 PM	1	mscorwks.dll	CryptExportKey (0x0000000000ba5550, 0x0000000000ba4c20, SIMPLEBLOB, 0, NULL, 0x000000000096d3d0)
14	4:34:34.683 PM	1	mscorwks.dll	CryptExportKey (0x0000000000ba5550, 0x0000000000ba4c20, SIMPLEBLOB, 0, 0x0000000029fe548, 0x000000000096d3d0)
15	4:34:34.683 PM	1	mscorwks.dll	CryptEncrypt (0x0000000000ba5550, NULL, TRUE, 0, NULL, 0x000000000096d3d0, 0)
16	4:34:34.683 PM	1	mscorwks.dll	CryptEncrypt (0x0000000000ba5550, NULL, TRUE, 0, 0x0000000029fe998, 0x000000000096d3d0, 32)

Hex Buffer: 276 bytes (Post-Call)

```
0000  06 02 00 00 00 a4 00 00 52 53 41 31 00 08 00 00 01 00 01 00 ab 0f de 39 c1 60 ac 4d  ........RSA1...........9.`.M
001c  0f 83 11 59 9e ad cf cb 1c 54 76 1b ce 72 37 11 b6 8c a8 8f 9e 94 0f 18 85 44 5b 5f  ...Y.....Tv..r7..........D[_
0038  06 a2 05 cd f2 0c 47 a6 02 45 71 7d f1 1c 28 3c af 81 4d 75 26 92 9b a5 96 fe 35 51  ......G..Eq}..(<..Mu&.....5Q
0054  fc cb 14 d9 eb 8f c1 e9 68 a0 ef 42 00 4b 1f 37 43 c5 9e 03 93 cb 93 0c 25 4d 87 6e  ........h..B.K.7C.......%M.n
0070  8e 42 00 05 e5 61 a5 b3 6b 42 bf 1a dd 00 d1 63 fb 61 3c fe 23 68 3f 28 2d 2b 64 d7  .B...a..kB.....c.a<.#h?(-+d.
008c  e9 9a 15 d3 b9 5e a9 1e a3 32 b1 4e 4c 12 52 29 2f f3 33 53 b8 c3 94 ad fb 7d 1b b3  .....^...2.NL.R)/.3S.....}..
00a8  8d f7 4d ff 0f 56 33 28 cb 1b 6e cd 11 5b 53 34 a8 bb ae cd 15 c2 c8 33 bf 63 d2 8d  ..M..V3(..n..[S4.......3.c..
00c4  4c ba 7b fe 95 f2 ba 44 4f e1 ed 17 df 34 36 16 bc 75 32 ad f8 a2 39 56 e8 9b d6 9d  L.{....DO....46..u2...9V....
00e0  d4 32 96 77 9a 50 9a 98 fb d8 4b 0a 04 44 f4 1b 12 4d eb bb 34 b9 bb 2b e4 df e9 ec  .2.w.P....K..D...M..4..+....
00fc  33 58 36 24 59 a3 7a e0 b4 1b 27 d8 a7 68 36 48 69 a6 29 19 ad cc ad a1              3X6$Y.z...'..h6Hi.)....
```

Output

That looks like an RSA key to me!!

"RSA1" is for public keys.... womp womp...

RSAPUBKEY structure (wincrypt.h)

12/05/2018 • 2 minutes to read

The **RSAPUBKEY** structure contains information specific to the

```
........RSA1...........9.`.M
...Y.....Tv..r7..........D[_
.....G..Eq}..(<..Mu&.....5Q
.........h..B.K.7C.......%M.n
.B...a..kB.....c.a<.#h?(-+d.
.....^...2.NL.R)/.3S.....}..
..M..V3(..n..[S4........3.c..
L.{....DO....46..u2...9V....
.2.w.P....K..D...M..4..+....
3X6$Y.z...'..h6Hi.).....
```

Syntax

C++ Copy

```
typedef struct _RSAPUBKEY {
  DWORD magic;
  DWORD bitlen;
  DWORD pubexp;
} RSAPUBKEY;
```

Members

`magic`

Set to RSA1 (0x31415352) for public keys and to RSA2 (0x32415352) for private keys.

Note The hexadecimal values are the **ASCII** encoding of RSA1 and RSA2.

So now I have to find a “RSA2” thing...

https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/ns-wincrypt-rsapubkey

Rabbit Hole #2

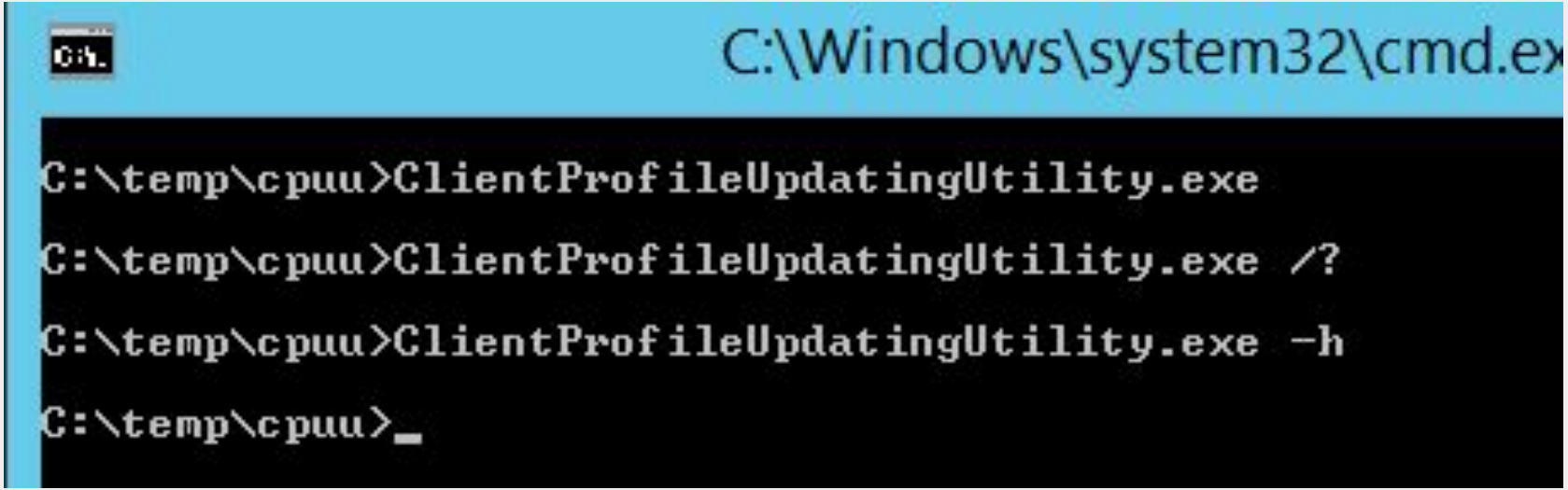
Decrypting the Passwords

What do we know so far?

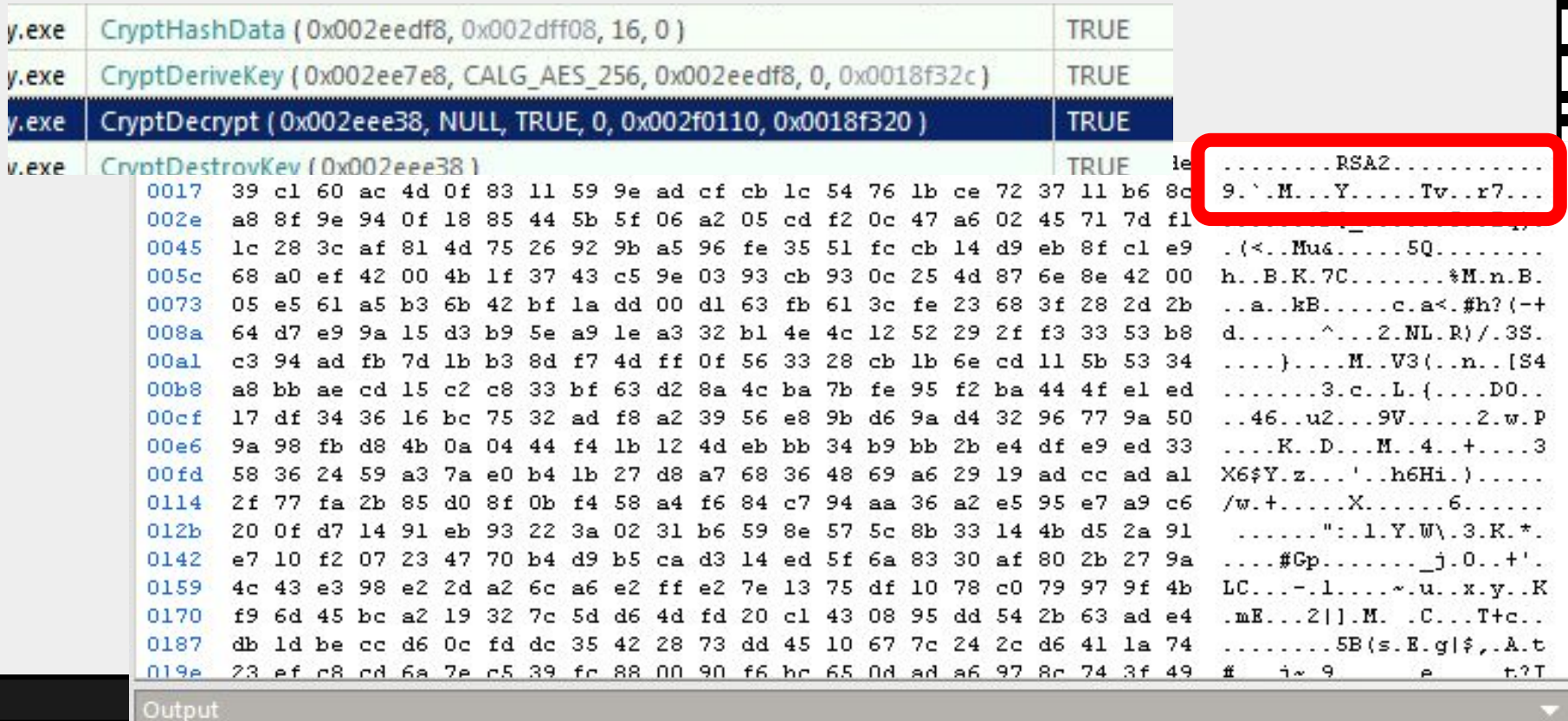
- ClientProfileUpdatingUtilityConfiguration.exe is written in C#
  - It is obfuscated w/ ConfuserEX
  - It’s uses Asymmetric cryptography, RSA + AES 256
  - It uses a public key that is built into the binary somewhere
- ClientProfileUpdatingUtility.exe has 32 and 64 bit versions
  - They are written in C++ or at least not C#
  - This is the utility that takes the CPUU.ini and actually runs the migration, so it needs to be able to do the decryption somehow...
  - Probably where we will find the private key. Fingers crossed

ClientProfileUpdatingUtility.exe

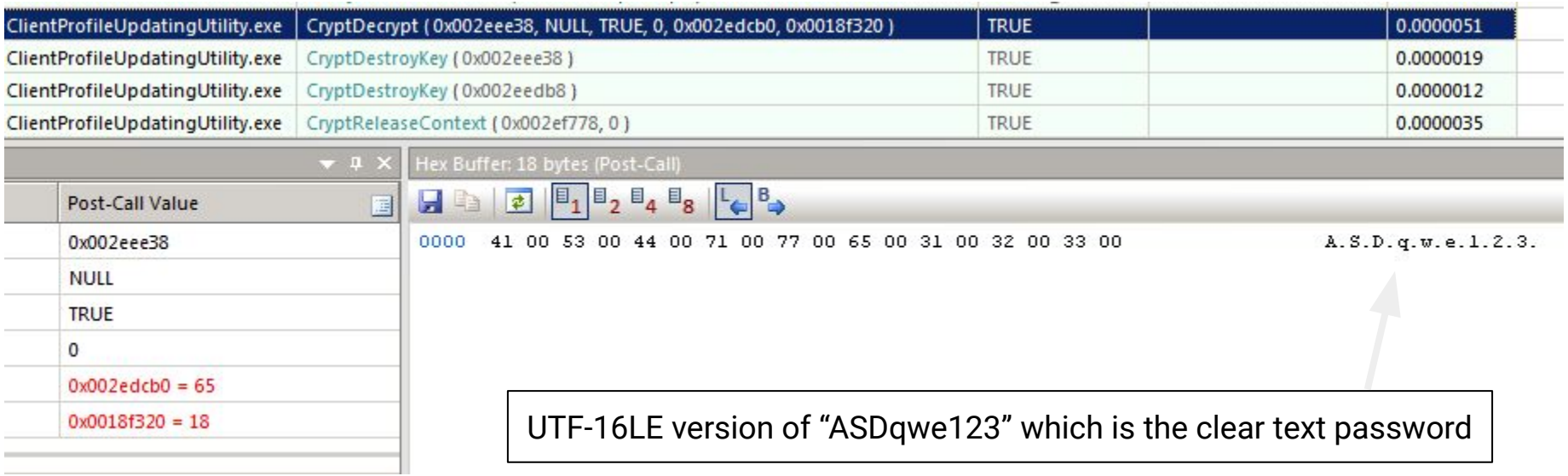
Console time... but what does this thing do?



Hey there's that RSA2 key!!!



WOOHOO!!



UTF-16LE version of “ASDqwe123” which is the clear text password

Great as a one-off, but...

- In order to pull of this decryption you need:
  - A system with Outlook installed
  - A copy of the ClientProfileUpdatingUtility.exe
  - API Monitor installed
  - Time to run the binary and dig through the API calls
- This doesn’t seem like much, but it’s not very portable

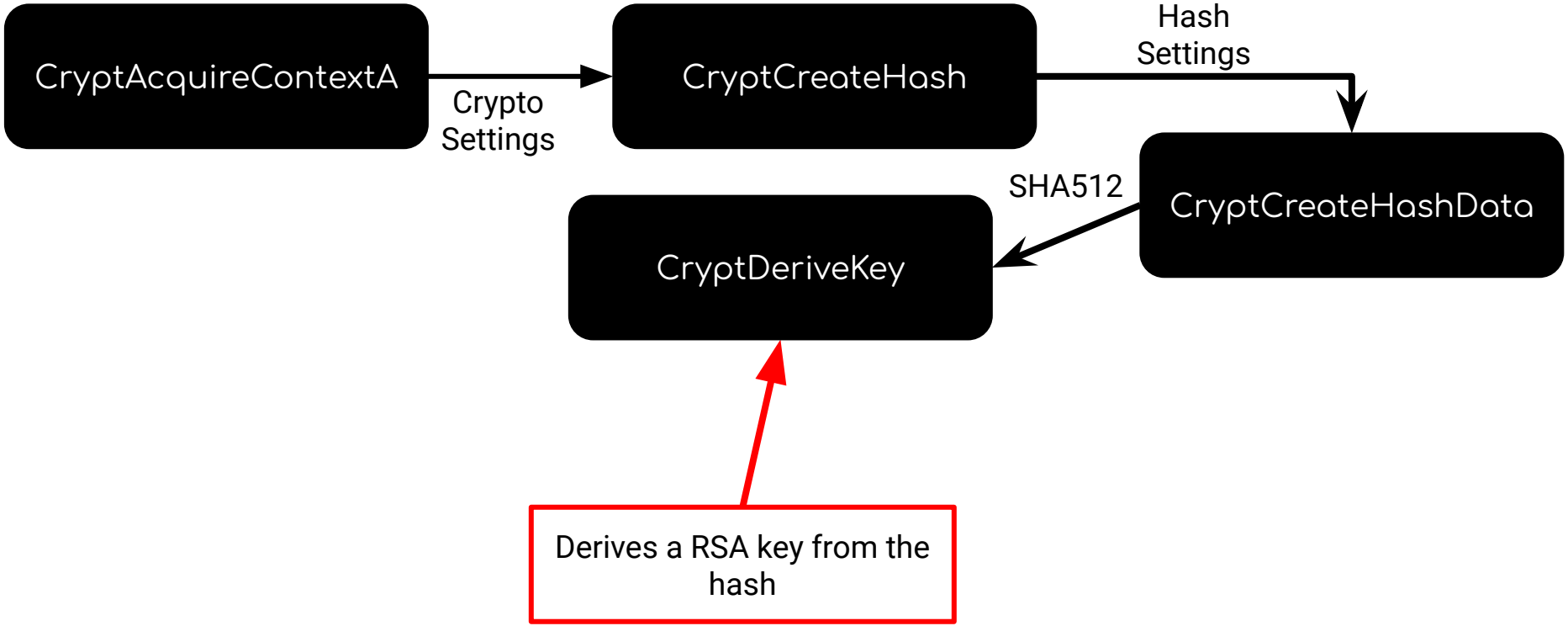
43	2:08:52.419 PM	1	msi.dll	ConvertSidToStringSidW (0x0018
44	2:08:52.436 PM	1	ClientProfileUpdatingUtility.exe	CryptAcquireContextA (0x0018f33
45	2:08:52.499 PM	1	rsaenh.dll	OpenThreadToken (GetCurre
46	2:08:52.499 PM	1	rsaenh.dll	OpenProcessToken (GetCurre
47	2:08:52.499 PM	1	rsaenh.dll	GetTokenInformation (0x0000

Parameters: CryptAcquireContextA (Advapi32.dll)

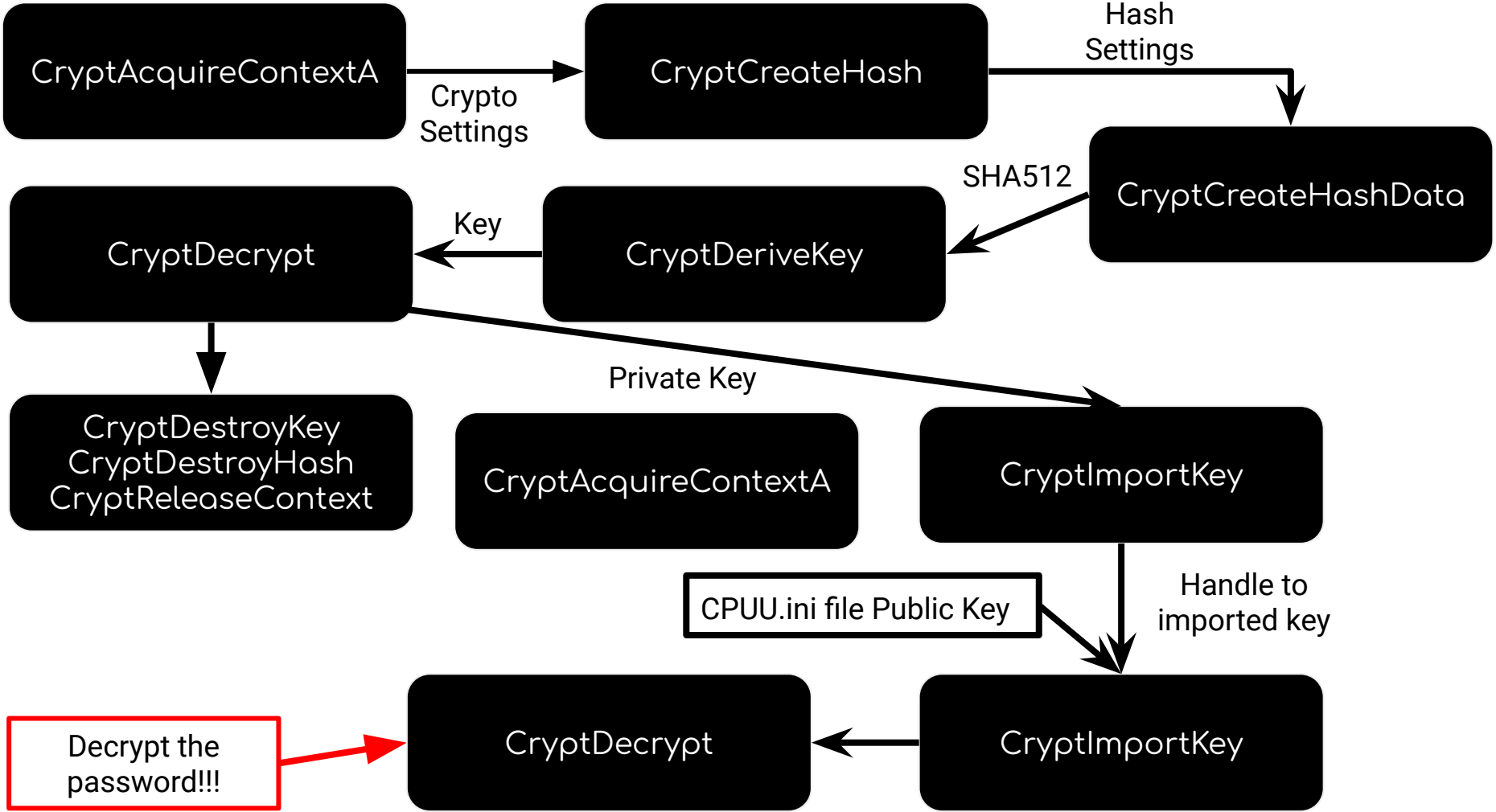
#	Type	Name	Pre-Call Value	Post-Call Value
1	HCRYPTPROV*	phProv	0x0018f330 = NULL	0x0018f330 = 0x0082e870
2	LPCTSTR	pszContainer	NULL	NULL
3	LPCTSTR	pszProvider	0x006bf538 "Microsoft Enhanced R...	0x006bf538 "Microsoft Enhanced R...
4	DWORD	dwProvType	PROV_RSA_AES	PROV_RSA_AES
5	DWORD	dwFlags	CRYPT_VERIFYCONTEXT	CRYPT_VERIFYCONTEXT

```
00
00
00
```

```
int main(int argc, char *argv[])
{
    if (argc <= 1) {
        std::cout << "This tool requires one argument. The string of bytes after Password= or PasswordTrg= \n";
        exit(EXIT_FAILURE);
    };

    std::cout << "Crypto Is Horrible...\n";
    bool final;
    HCRYPTPROV phProv = 0;
    LPCSTR szContainer = NULL;
    const char* szProvider = "Microsoft Enhanced RSA and AES Cryptographic Provider";
    DWORD dwProvType = PROV_RSA_AES;
    DWORD dwFlags = CRYPT_VERIFYCONTEXT;
```

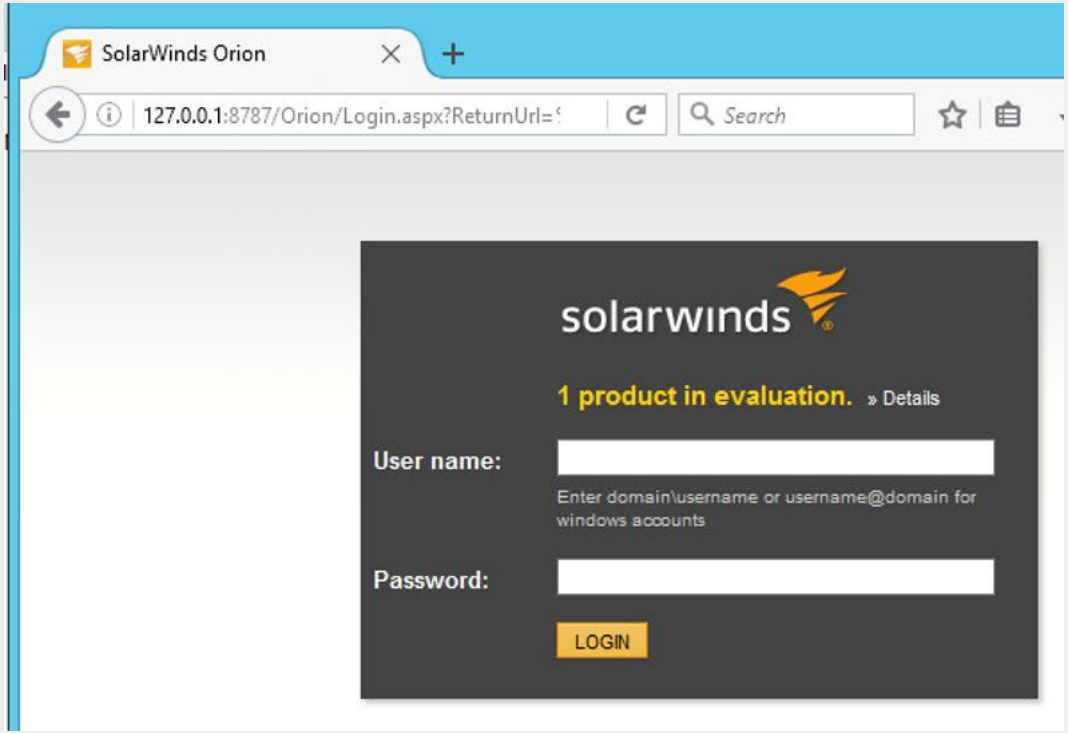


https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-cryptderivekey


CryptAcquireContextA
Crypto Settings
CryptCreateHash
Hash Settings
CryptCreateHashData
SHA512
CryptDeriveKey
Key
CryptDecrypt
CryptDestroyKey
CryptDestroyHash
CryptReleaseContext
Private Key
CryptAcquireContextA
CryptImportKey
CPUU.ini file Public Key
Handle to imported key
Decrypt the password!!!
CryptDecrypt
CryptImportKey


DecryptCPUU
https://github.com/mubix/decryptcpuu

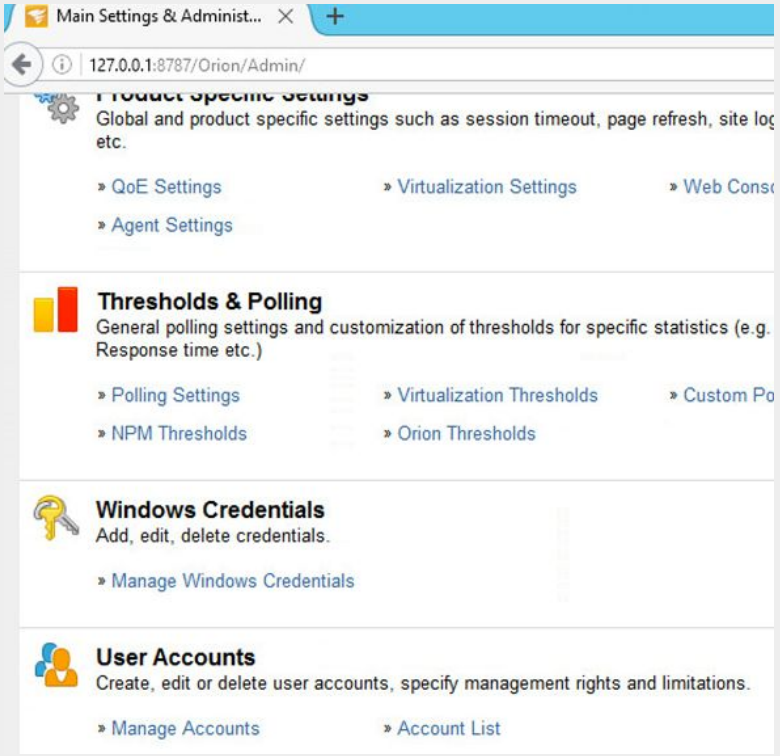
Hey Rob, What's this?



SolarWinds

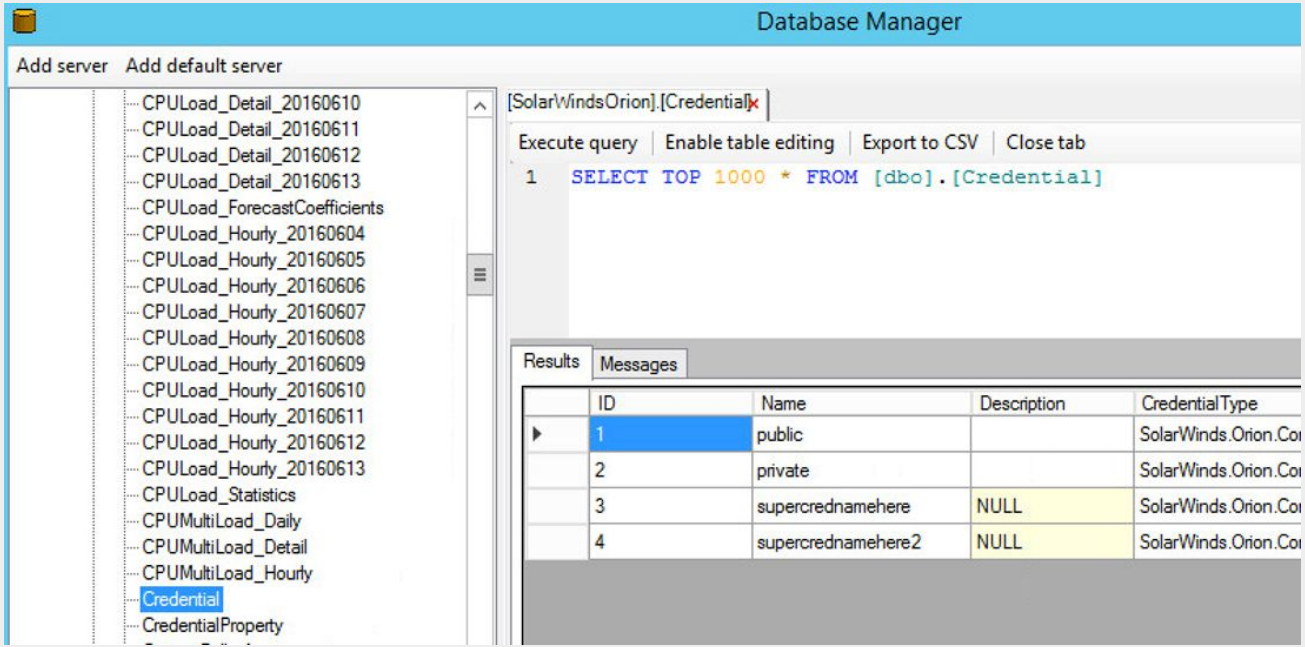
Stores all the creds...

- VMware
- SNMP
- Active Directory
- AWS/Azure
- Cisco
- Meraki
- Twitter...

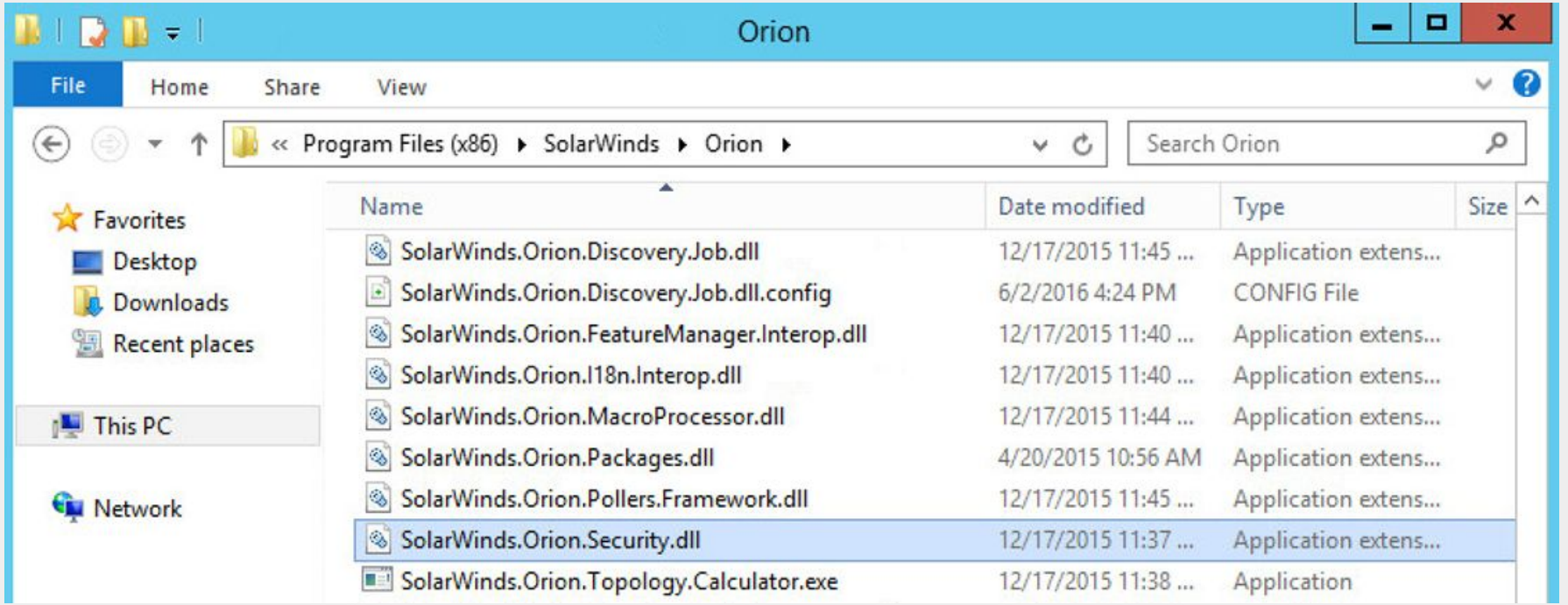


Where are the credentials?

Broken into two tables. Credential and CredentialProperty



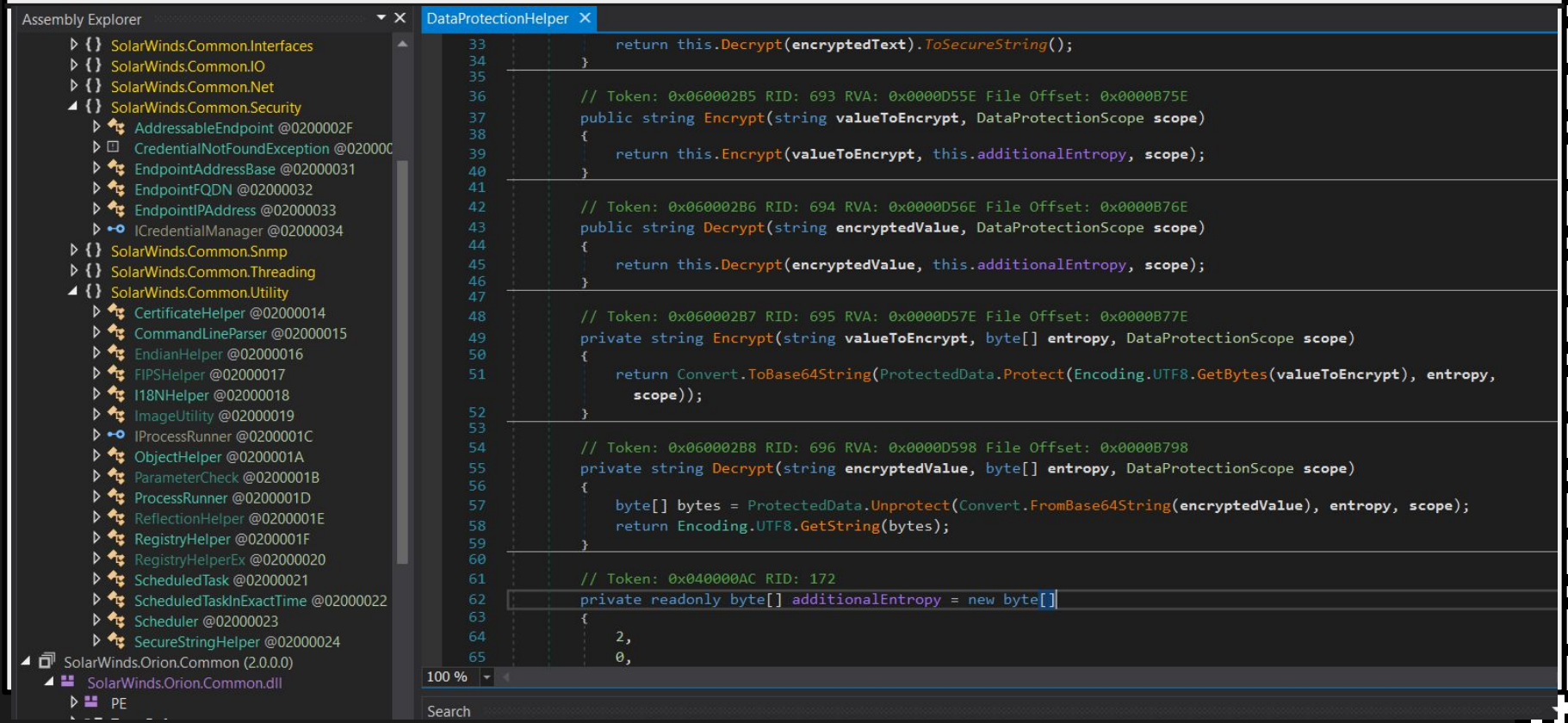
Back to DnSpy!



Password Hashing

```
157         public static string HashPassword(string password, string salt, bool caseSensitive)
158         {
159             if (password == null)
160             {
161                 throw new ArgumentException("password");
162             }
163             if (salt == null)
164             {
165                 throw new ArgumentException("salt");
166             }
167             HashAlgorithm hashAlgorithm = new SHA512CryptoServiceProvider();
168             string password2 = password;
169             if (!caseSensitive)
170             {
171                 password2 = password.ToUpperInvariant();
172             }
173             Encoding encoding = new UTF8Encoding();
174             int length = salt.ToLowerInvariant().Length;
175             byte[] bytes = encoding.GetBytes(salt.ToLowerInvariant() + ((length < 8) ? "1244352345234".Substring(0, 8
                  - length) : ""));
176             string result;
177             using (DeriveBytes deriveBytesAlgorithm = EncryptionHelper.GetDeriveBytesAlgorithm(password2, bytes))
178             {
179                 result = Convert.ToBase64String(hashAlgorithm.ComputeHash(deriveBytesAlgorithm.GetBytes
                      (EncryptionHelper.BytesUsedForHash)));
180             }
181             return result;
```

CryptUnprotect / ProtectedData.Unprotect



DecryptAes, DecryptShort, DecryptXml

```
57      // Token: 0x0600001D RID: 29 RVA: 0x00002750 File Offset: 0x00000950
58      public string Decrypt(string encryptedText)
59      {
60          if (encryptedText == null)
61          {
62              throw new ArgumentNullException();
63          }
64          if (encryptedText.StartsWith("-"))
65          {
66              return this.DecryptAes(encryptedText);
67          }
68          if (!encryptedText.StartsWith("<"))
69          {
70              return this.DecryptShort(encryptedText);
71          }
72          return CryptoHelper.DecryptXml(encryptedText);
73      }
74
75      // Token: 0x0600001E RID: 30 RVA: 0x0000278B File Offset: 0x0000098B
76      public static string GetOrionCertificateName()
77      {
78          return "SolarWinds-Orion";
79      }
80
81      // Token: 0x0600001F RID: 31 RVA: 0x00002792 File Offset: 0x00000992
82      public SecureString DecryptSecure(string encryptedText)
83      {
```

Old Password Encryption == Long Division

```
// Depending on how long the SolarWinds box may be around
// It might still store the credentials in the old format
// which can be decoded using long divion.
1 reference
static string DecodeOldPassword(string password)
{
    if (string.IsNullOrEmpty(password))
    {
        return string.Empty;
    }
    bool flag = password.StartsWith("U-");
    if (flag)
    {
        password = password.Replace("U-", "");
    }
    string value = password.Substring(0, password.IndexOf("-"));
    password = password.Substring(password.IndexOf('-') + 1);
    password = password.Replace("-", "");
    password = password.Trim();
    string text = string.Empty;
    for (int i = 1; i <= password.Length - 1; i += 2)
    {
        text = text + password[i] + password[i - 1];
    }
    if (text.Length < password.Length)
    {
        text += password[password.Length - 1];
    }
    password = text;
    Int64.TryParse(value, out long divisor);
    text = longDivision(password, divisor);
    text = longDivision(text, 1244352345234);
    text = text.Substring(1);
    password = string.Empty;
    int num = flag ? 3 : 2;
```

Math Warning

11-417578424799297-9-626069743079568576306772 4

1. 41757842479929796260697430795685763067724 / 11
    a. 37961674981754360236997664359714330061 56
2. 37961674981754360236997664359714330061 56 / 1244352345234
    a. 30507175180045713856766965 24
3. From Hex(30507175180045713856766965 24) -> 0Pqu..Eq8Vvie$

Old Password Encryption == Long Division + Uppercase

```
namespace DeCryptSolarWindsAccounts
{
    class Program
    {
        static void Main(string[] args)
        {
            string password = "107-41811132619 52-05215272239 20590-9616111489183";
            bool flag = password.StartsWith("U-");
            if (flag)
            {
```

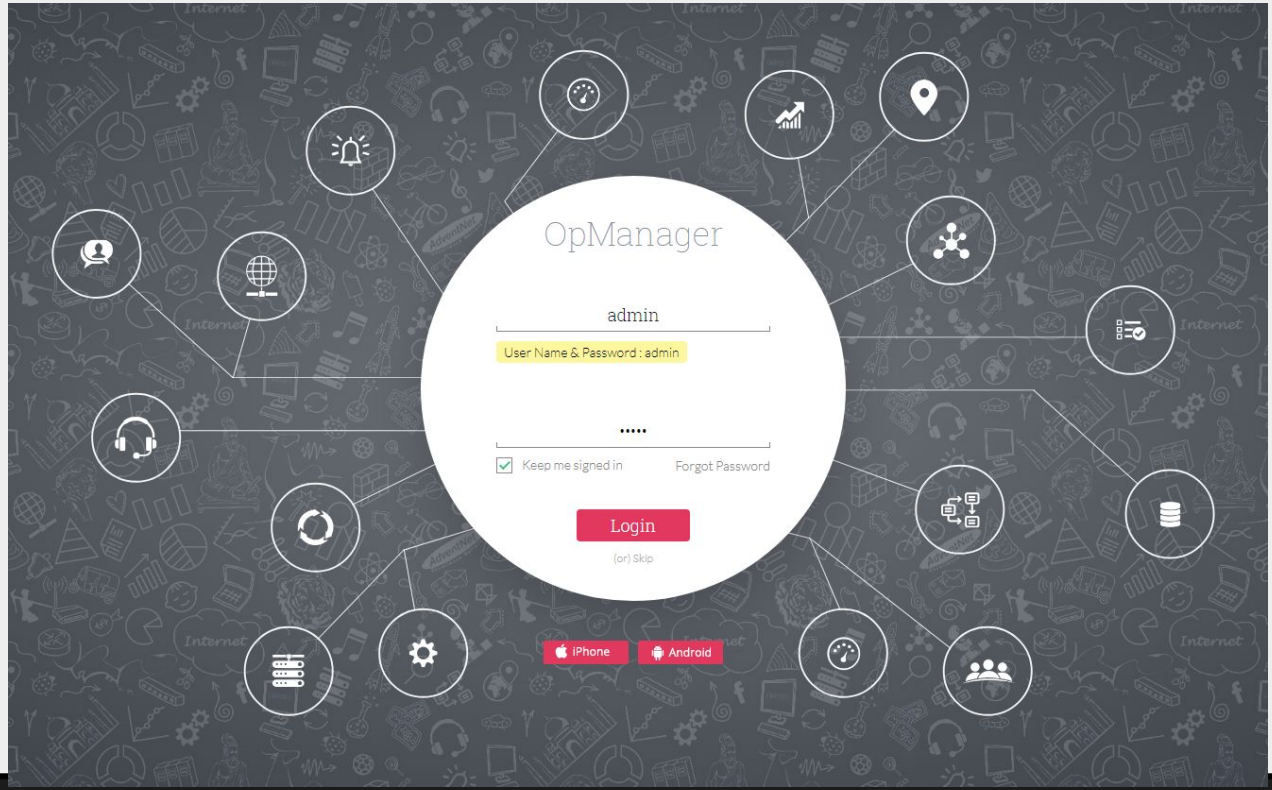
file:///c:/users/administrator/documents/visual studio 2015/Projects/DeC

```
ASDQWE123
```

SolarFlare

https://github.com/mubix/solarflare

Hey Rob, What's this?



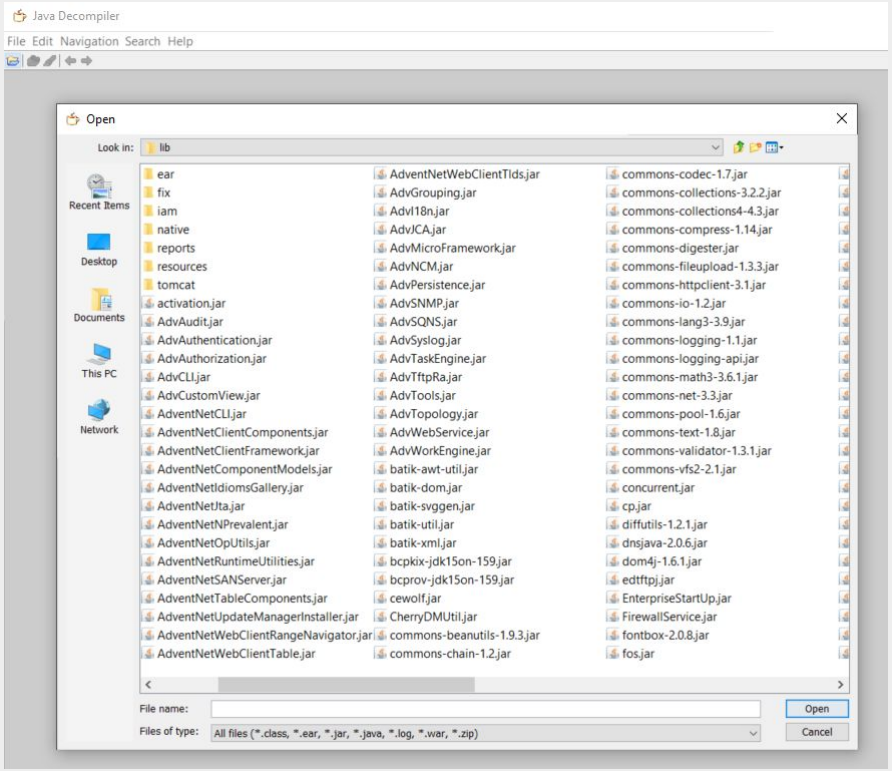
ManageEngine OpManager

OpManager

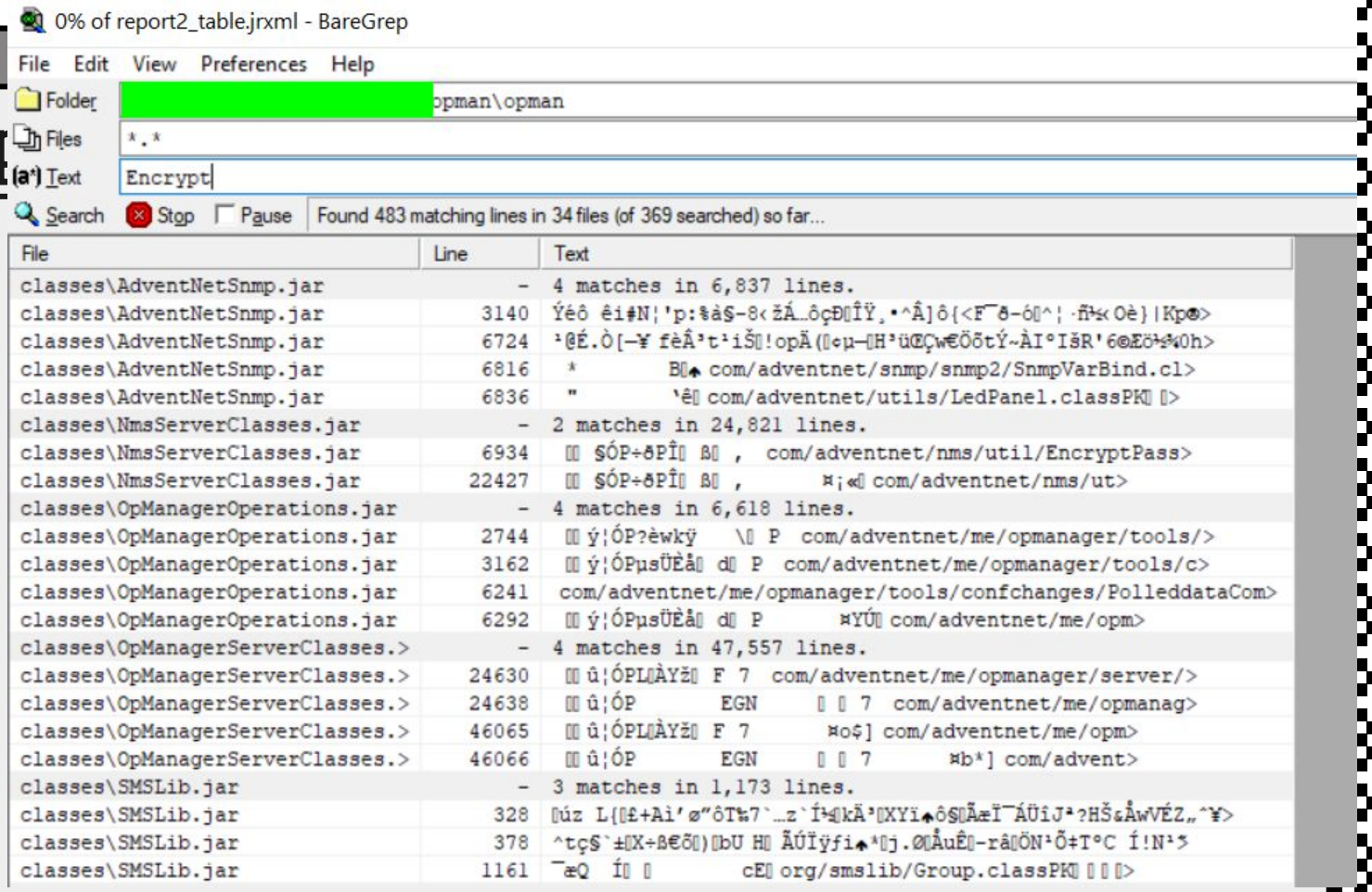
- Java (harder to grep)
- Learned about Java Debugger
    - wrapper.conf already had Java debugger configuration

```
  GNU nano 4.8                                                    wrapper.conf
wrapper.java.additional.9=-Duser.language=en
wrapper.java.additional.10=-Duser.home=../logs
wrapper.java.additional.11=-Dproduct.home=..
wrapper.java.additional.12=-DstartingAsService="true"
# Additional PermGen Space
wrapper.java.additional.13=-XX:MetaspaceSize=128m
wrapper.java.additional.14=-XX:MaxMetaspaceSize=256m
wrapper.java.additional.15=-XX:+HeapDumpOnOutOfMemoryError
#wrapper.java.additional.16=-Dsun.net.client.defaultReadTimeout=20000
#wrapper.java.additional.17=-Dsun.net.client.defaultConnectTimeout=6000


#uncomment the following to enable JPDA debugging
#wrapper.java.additional.3=-Xdebug
#wrapper.java.additional.4=-Xnoagent
#wrapper.java.additional.5=-Xrunjdwp:transport=dt_socket,address=8787,server=y,suspend=n
```

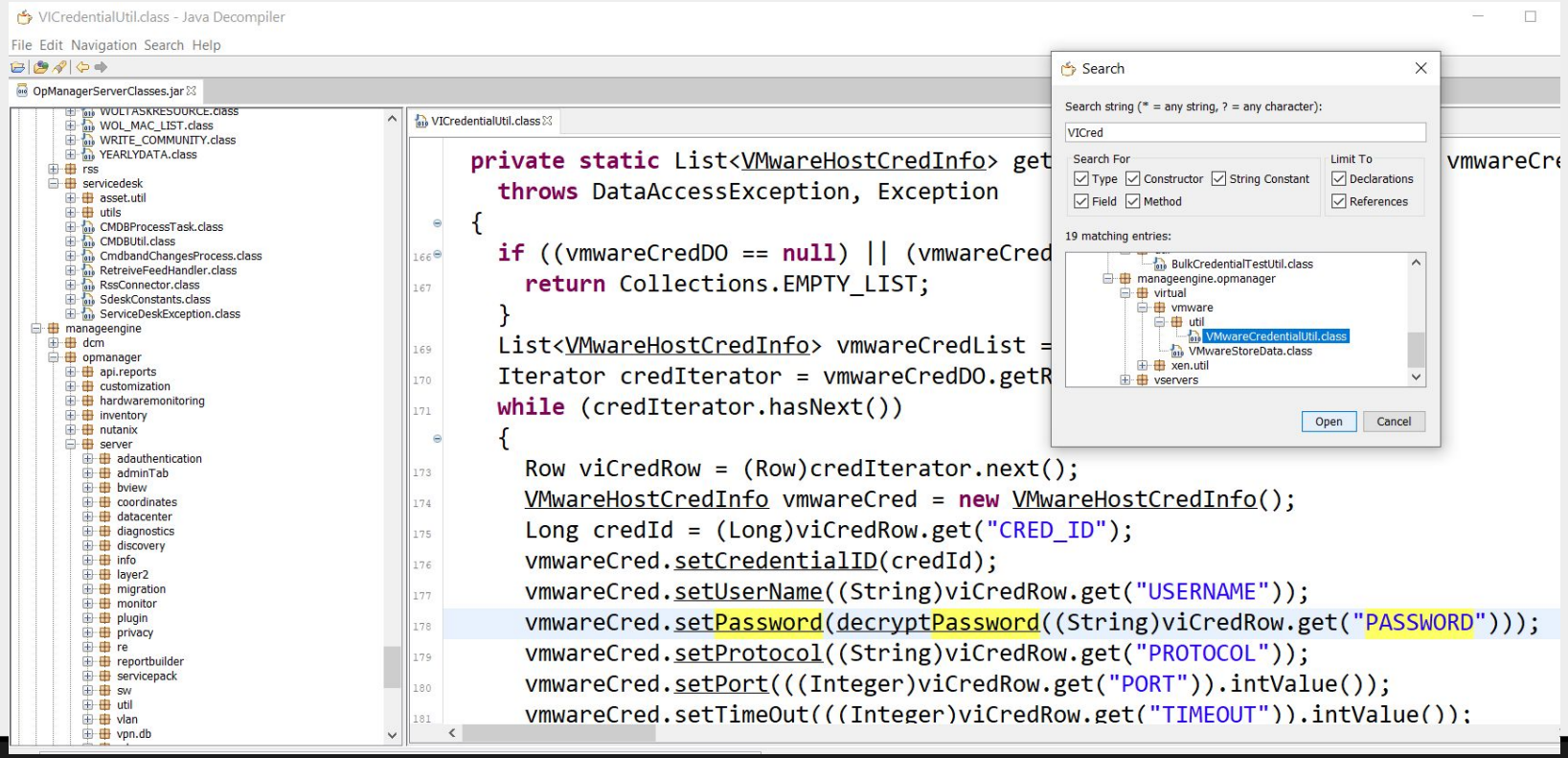
Huge Application



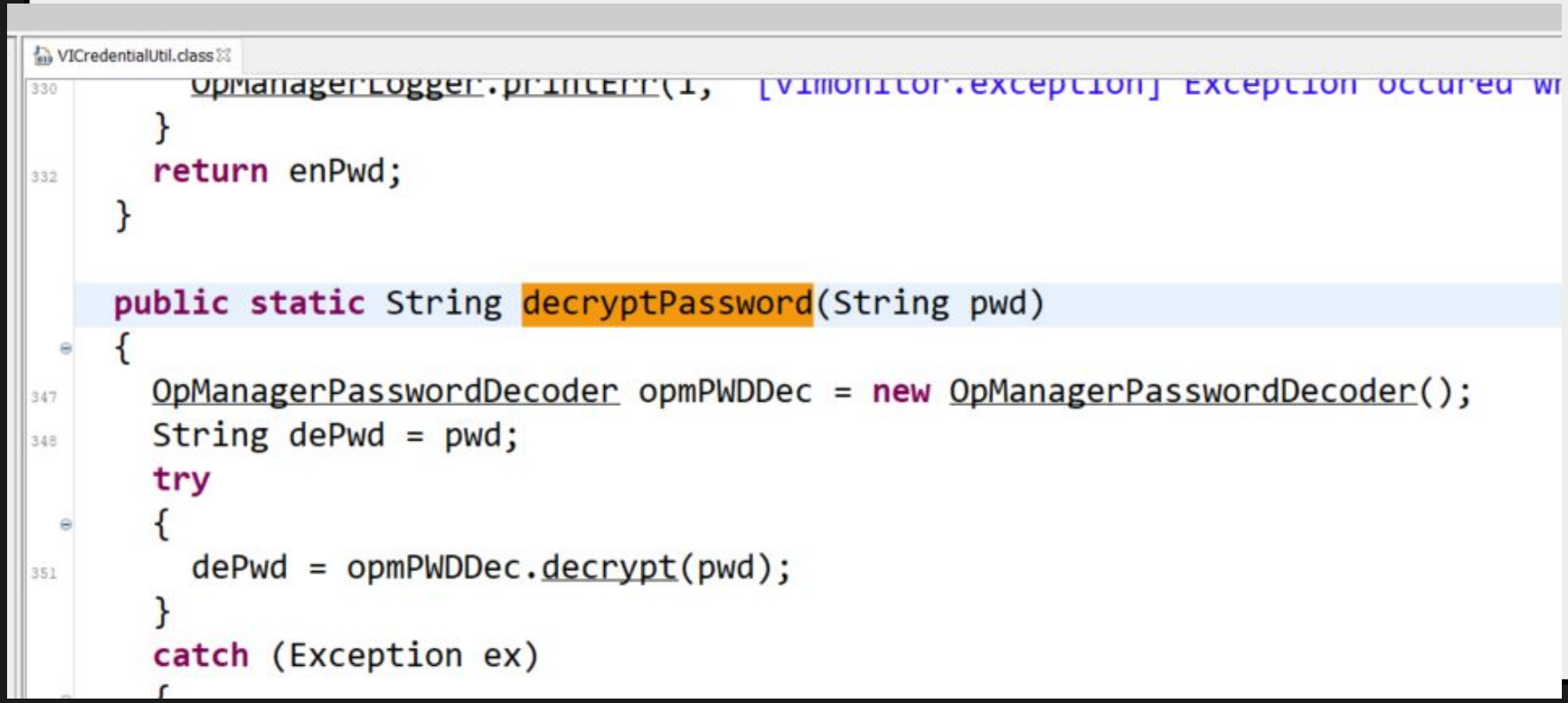
BareGrep


0% of report2_table.jrxml - BareGrep
File Edit View Preferences Help
Folder opman\opman
Files *.*
(a*) Text Encrypt
Search Stop Pause
Found 483 matching lines in 34 files (of 369 searched) so far...
File Line Text
classes\AdventNetSnmp.jar - 4 matches in 6,837 lines.
classes\AdventNetSnmp.jar 3140 Ýéô êi♯N¦'p:%à$-8‹žÁ…ôçÐ▯ÎŸ¸•^Â]ô{<F¯ð-ó▯^¦·ñ¾‹Oè}|Kp☻>
classes\AdventNetSnmp.jar 6724 ¹@É.Ò[—¥ fèÂ³t¹iŠ▯!opÄ(▯¢µ—▯H³üŒÇw€ÖõtÝ~ÀI°IšR'6☺Ëö¹⅝¼0h>
classes\AdventNetSnmp.jar 6816 * B▯♠ com/adventnet/snmp/snmp2/SnmpVarBind.cl>
classes\AdventNetSnmp.jar 6836 " 'ê▯ com/adventnet/utils/LedPanel.classPK▯ ▯>
classes\NmsServerClasses.jar - 2 matches in 24,821 lines.
classes\NmsServerClasses.jar 6934 ▯▯ §ÓP÷ðPÎ▯ ß▯ , com/adventnet/nms/util/EncryptPass>
classes\NmsServerClasses.jar 22427 ▯▯ §ÓP÷ðPÎ▯ ß▯ , ¤¡« com/adventnet/nms/ut>
classes\OpManagerOperations.jar - 4 matches in 6,618 lines.
classes\OpManagerOperations.jar 2744 ▯▯ ý¦ÓP?èwkÿ \▯ P com/adventnet/me/opmanager/tools/>
classes\OpManagerOperations.jar 3162 ▯▯ ý¦ÓPµsÜÈå▯ d▯ P com/adventnet/me/opmanager/tools/c>
classes\OpManagerOperations.jar 6241 com/adventnet/me/opmanager/tools/confchanges/PolleddataCom>
classes\OpManagerOperations.jar 6292 ▯▯ ý¦ÓPµsÜÈå▯ d▯ P ¤YÚ▯ com/adventnet/me/opm>
classes\OpManagerServerClasses.> - 4 matches in 47,557 lines.
classes\OpManagerServerClasses.> 24630 ▯▯ û¦ÓPL▯ÀYž▯ F 7 com/adventnet/me/opmanager/server/>
classes\OpManagerServerClasses.> 24638 ▯▯ û¦ÓP EGN ▯ ▯ 7 com/adventnet/me/opmanag>
classes\OpManagerServerClasses.> 46065 ▯▯ û¦ÓPL▯ÀYž▯ F 7 ¤o$] com/adventnet/me/opm>
classes\OpManagerServerClasses.> 46066 ▯▯ û¦ÓP EGN ▯ ▯ 7 ¤b*] com/advent>
classes\SMSLib.jar - 3 matches in 1,173 lines.
classes\SMSLib.jar 328 ▯úz L([▯£+Ai'ø"ôT℅7`…z`Í¾▯kÄ³▯XYï♠ô§▯ÃæÏ¯ÁÜîJ²?HŠ&ÅwVÉZ„^¥>
classes\SMSLib.jar 378 ^tç§`±▯X÷ß€õ▯)▯bU H▯ ÃÚÏÿƒi♠⁴▯j.Ø▯Åuê▯-râ▯ÖN¹Õ‡T°C Í!N¹⅗
classes\SMSLib.jar 1161 ¯æQ Í▯ ▯ cE▯ org/smslib/Group.classPK▯ ▯▯▯>
Encrypt

JD-GUI


VICredentialUtil.class - Java Decompiler
File Edit Navigation Search Help
OpManagerServerClasses.jar
VICredentialUtil.class
WOLTASKRESOURCE.class
WOL_MAC_LIST.class
WRITE_COMMUNITY.class
YEARLYDATA.class
rss
servicedesk
asset.util
utils
CMDBProcessTask.class
CMDBUtil.class
CmdbandChangesProcess.class
RetreiveFeedHandler.class
RssConnector.class
SdeskConstants.class
ServiceDeskException.class
manageengine
dcm
opmanager
api.reports
customization
hardwaremonitoring
inventory
nutanix
server
adauthentication
adminTab
bview
coordinates
datacenter
diagnostics
discovery
info
layer2
migration
monitor
plugin
privacy
re
reportbuilder
servicepack
sw
util
vlan
vpn.db
private static List<VMwareHostCredInfo> get
vmwareCre
throws DataAccessException, Exception
{
if ((vmwareCredDO == null) || (vmwareCred
return Collections.EMPTY_LIST;
}
List<VMwareHostCredInfo> vmwareCredList =
Iterator credIterator = vmwareCredDO.getR
while (credIterator.hasNext())
{
Row viCredRow = (Row)credIterator.next();
VMwareHostCredInfo vmwareCred = new VMwareHostCredInfo();
Long credId = (Long)viCredRow.get("CRED_ID");
vmwareCred.setCredentialID(credId);
vmwareCred.setUserName((String)viCredRow.get("USERNAME"));
vmwareCred.setPassword(decryptPassword((String)viCredRow.get("PASSWORD")));
vmwareCred.setProtocol((String)viCredRow.get("PROTOCOL"));
vmwareCred.setPort(((Integer)viCredRow.get("PORT")).intValue());
vmwareCred.setTimeOut(((Integer)viCredRow.get("TIMEOUT")).intValue());
Search
Search string (* = any string, ? = any character):
VICred
Search For
Type
Constructor
String Constant
Field
Method
Limit To
Declarations
References
19 matching entries:
BulkCredentialTestUtil.class
manageengine.opmanager
virtual
vmware
util
VMwareCredentialUtil.class
VMwareStoreData.class
xen.util
vservers
Open
Cancel

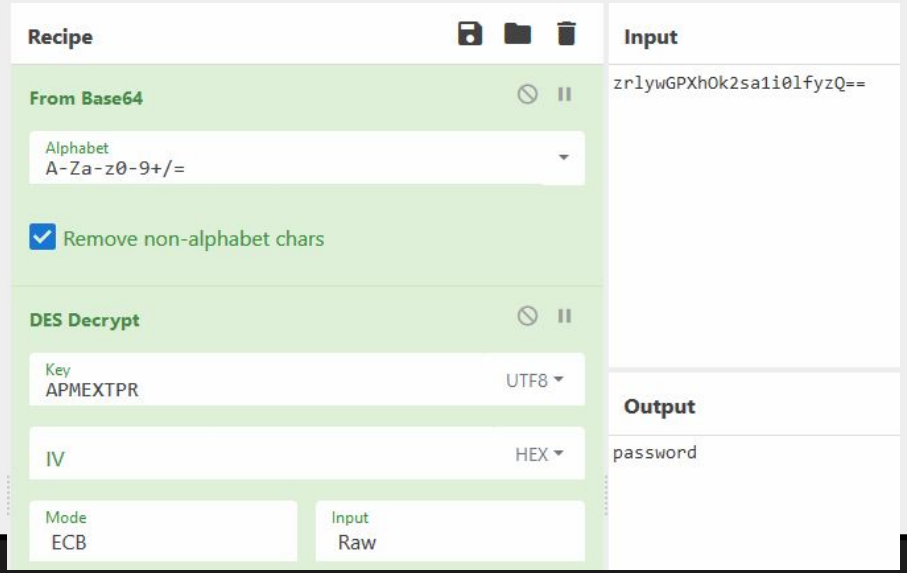
JD-GUI


VICredentialUtil.class
330
}
332
return enPwd;
}
public static String decryptPassword(String pwd)
{
347
OpManagerPasswordDecoder opmPWDDec = new OpManagerPasswordDecoder();
348
String dePwd = pwd;
try
{
351
dePwd = opmPWDDec.decrypt(pwd);
}
catch (Exception ex)

```
public static final String DES_ENCRYPTION_SCHEME = "DES";
private String encryptionKey = ITOMSecurityUtil.getInstance().getExternalAuthKey("opm", "apmConnector", false);

public String decrypt(String encryptedString)
  throws Exception
{
  return decrypt(encryptedString, this.encryptionKey);
}

public String decrypt(String encryptedString, String encryptionKey)
  throws Exception
{
  if (encryptionKey != null) {
    try
    {
      byte[] keyAsBytes = encryptionKey.getBytes("UTF8");
      this.keySpec = new DESKeySpec(keyAsBytes);
      this.keyFactory = SecretKeyFactory.getInstance("DES");
      this.cipher = Cipher.getInstance("DES");
      if ((encryptedString == null) || (encryptedString.trim().length() <= 0)) {
        throw new IllegalArgumentException("encrypted string was null or empty");
      }
      SecretKey key = this.keyFactory.generateSecret(this.keySpec);
      this.cipher.init(2, key);
      BASE64Decoder base64decoder = new BASE64Decoder();
      byte[] cleartext = base64decoder.decodeBuffer(encryptedString);
      byte[] ciphertext = this.cipher.doFinal(cleartext);
      return bytes2String(ciphertext);
    }
    catch (Exception ex)
```

JD-GUI - DES Decryption

1. Go get “opm / apmConnector” from Database
    a. Database decrypts it to this: APMEXTPRODjZ_7004PROD_AppManager
2. Take the first 8 bytes of that (**APMEXTPR**) as the DES key
3. ECP DES Decrypt



PostgreSQL PGP Symmetric Key Encryption/Decryption

- When installed on Linux (or using PostgreSQL, OpManager uses PGP encryption to encrypt ALL sensitive fields)
- For some reason this includes things like Time Zones and other ?GDPR? related items
- Default password of “Mickey”

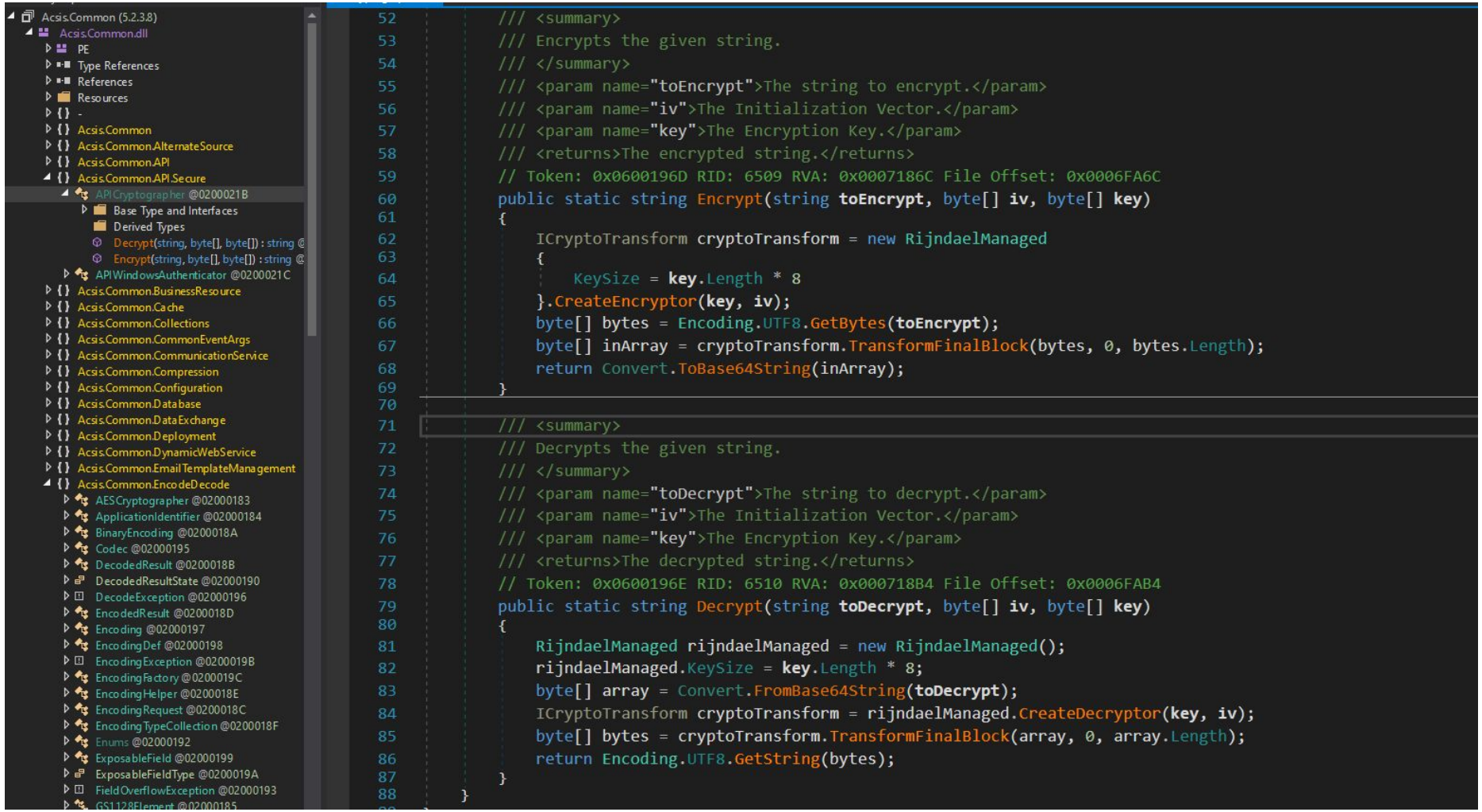
```
OpManagerDB=> select username, pgp_sym_decrypt(password::bytea, 'Mickey') from vicre
 username  |     pgp_sym_decrypt
-----------+-------------------------
 superuser | iDOqCguX/aXpAjtDgTAvTg==
(1 row)
```

Manage2Decrypt

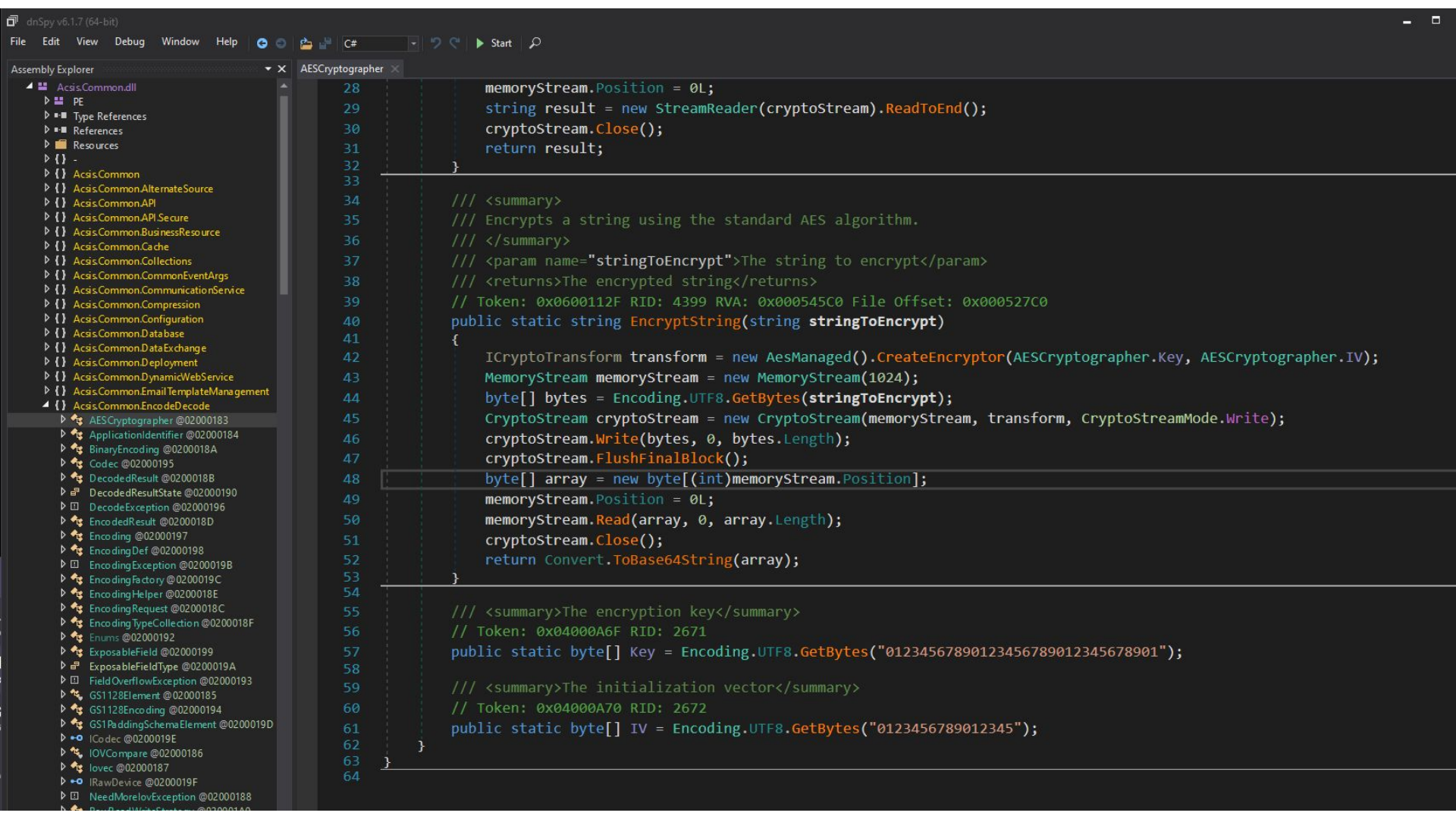
https://github.com/mubix/manage2decrypt

Hey Rob, What's this?

Acsis



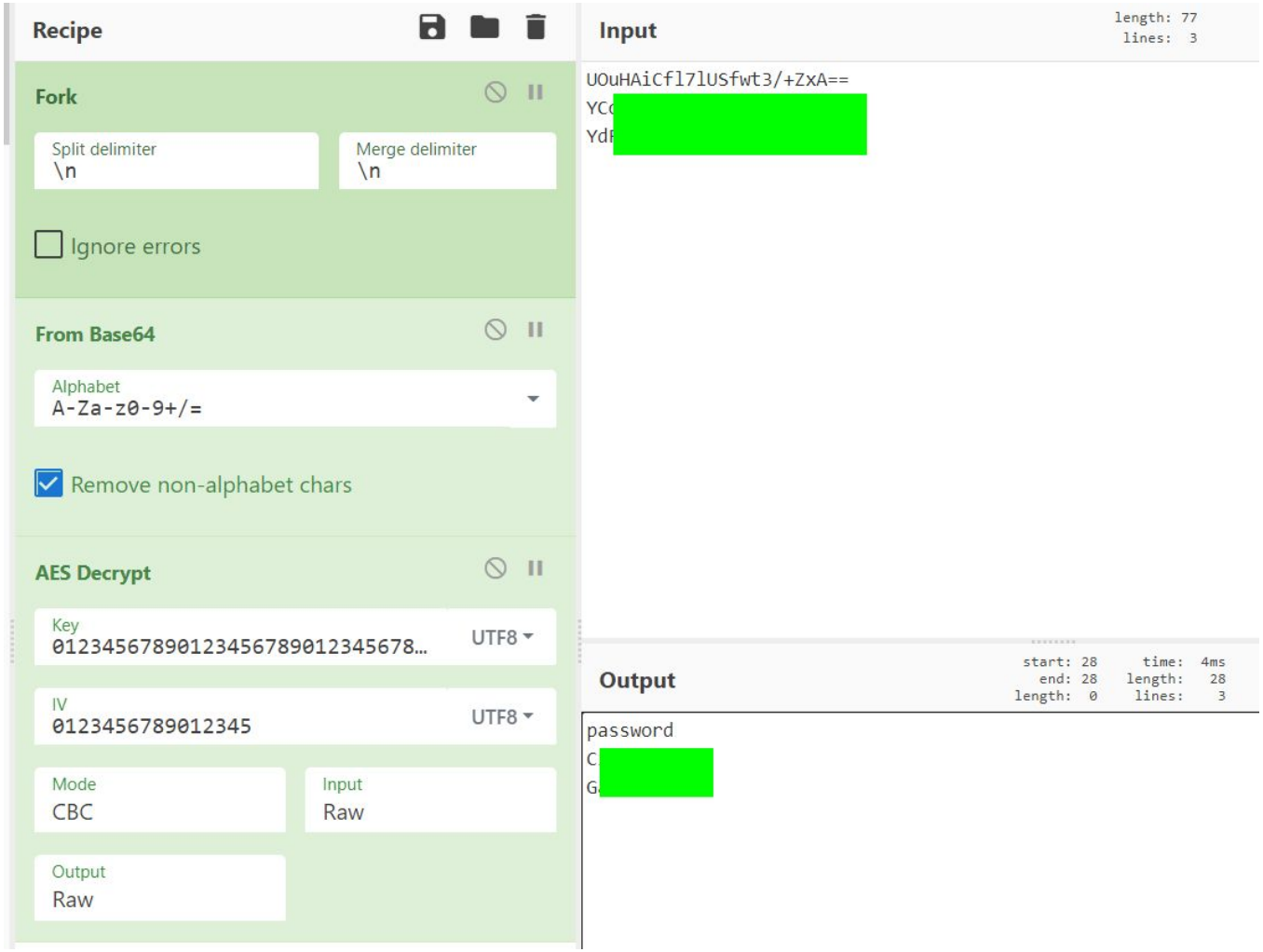

```
52          /// <summary>
53          /// Encrypts the given string.
54          /// </summary>
55          /// <param name="toEncrypt">The string to encrypt.</param>
56          /// <param name="iv">The Initialization Vector.</param>
57          /// <param name="key">The Encryption Key.</param>
58          /// <returns>The encrypted string.</returns>
59          // Token: 0x0600196D RID: 6509 RVA: 0x0007186C File Offset: 0x0006FA6C
60          public static string Encrypt(string toEncrypt, byte[] iv, byte[] key)
61          {
62              ICryptoTransform cryptoTransform = new RijndaelManaged
63              {
64                  KeySize = key.Length * 8
65              }.CreateEncryptor(key, iv);
66              byte[] bytes = Encoding.UTF8.GetBytes(toEncrypt);
67              byte[] inArray = cryptoTransform.TransformFinalBlock(bytes, 0, bytes.Length);
68              return Convert.ToBase64String(inArray);
69          }
70
71          /// <summary>
72          /// Decrypts the given string.
73          /// </summary>
74          /// <param name="toDecrypt">The string to decrypt.</param>
75          /// <param name="iv">The Initialization Vector.</param>
76          /// <param name="key">The Encryption Key.</param>
77          /// <returns>The decrypted string.</returns>
78          // Token: 0x0600196E RID: 6510 RVA: 0x000718B4 File Offset: 0x0006FAB4
79          public static string Decrypt(string toDecrypt, byte[] iv, byte[] key)
80          {
81              RijndaelManaged rijndaelManaged = new RijndaelManaged();
82              rijndaelManaged.KeySize = key.Length * 8;
83              byte[] array = Convert.FromBase64String(toDecrypt);
84              ICryptoTransform cryptoTransform = rijndaelManaged.CreateDecryptor(key, iv);
85              byte[] bytes = cryptoTransform.TransformFinalBlock(array, 0, array.Length);
86              return Encoding.UTF8.GetString(bytes);
87          }
88      }
```


dnSpy v6.1.7 (64-bit)
File Edit View Debug Window Help
C#
Start
Assembly Explorer
AESCryptographer
Acsis.Common.dll
PE
Type References
References
Resources
Acsis.Common
Acsis.Common.AlternateSource
Acsis.Common.API
Acsis.Common.API.Secure
Acsis.Common.BusinessResource
Acsis.Common.Cache
Acsis.Common.Collections
Acsis.Common.CommonEventArgs
Acsis.Common.CommunicationService
Acsis.Common.Compression
Acsis.Common.Configuration
Acsis.Common.Database
Acsis.Common.DataExchange
Acsis.Common.Deployment
Acsis.Common.DynamicWebService
Acsis.Common.EmailTemplateManagement
Acsis.Common.EncodeDecode
AESCryptographer @02000183
ApplicationIdentifier @02000184
BinaryEncoding @0200018A
Codec @02000195
DecodedResult @0200018B
DecodedResultState @02000190
DecodeException @02000196
EncodedResult @0200018D
Encoding @02000197
EncodingDef @02000198
EncodingException @0200019B
EncodingFactory @0200019C
EncodingHelper @0200018E
EncodingRequest @0200018C
EncodingTypeCollection @0200018F
Enums @02000192
ExposableField @02000199
ExposableFieldType @0200019A
FieldOverflowException @02000193
GS1128Element @02000185
GS1128Encoding @02000194
GS1PaddingSchemaElement @0200019D
ICodec @0200019E
IOVCompare @02000186
Iovec @02000187
IRawDevice @0200019F
NeedMoreIovException @02000188
28 memoryStream.Position = 0L;
29 string result = new StreamReader(cryptoStream).ReadToEnd();
30 cryptoStream.Close();
31 return result;
32 }
33
34 /// <summary>
35 /// Encrypts a string using the standard AES algorithm.
36 /// </summary>
37 /// <param name="stringToEncrypt">The string to encrypt</param>
38 /// <returns>The encrypted string</returns>
39 // Token: 0x0600112F RID: 4399 RVA: 0x000545C0 File Offset: 0x000527C0
40 public static string EncryptString(string stringToEncrypt)
41 {
42 ICryptoTransform transform = new AesManaged().CreateEncryptor(AESCryptographer.Key, AESCryptographer.IV);
43 MemoryStream memoryStream = new MemoryStream(1024);
44 byte[] bytes = Encoding.UTF8.GetBytes(stringToEncrypt);
45 CryptoStream cryptoStream = new CryptoStream(memoryStream, transform, CryptoStreamMode.Write);
46 cryptoStream.Write(bytes, 0, bytes.Length);
47 cryptoStream.FlushFinalBlock();
48 byte[] array = new byte[(int)memoryStream.Position];
49 memoryStream.Position = 0L;
50 memoryStream.Read(array, 0, array.Length);
51 cryptoStream.Close();
52 return Convert.ToBase64String(array);
53 }
54
55 /// <summary>The encryption key</summary>
56 // Token: 0x04000A6F RID: 2671
57 public static byte[] Key = Encoding.UTF8.GetBytes("01234567890123456789012345678901");
58
59 /// <summary>The initialization vector</summary>
60 // Token: 0x04000A70 RID: 2672
61 public static byte[] IV = Encoding.UTF8.GetBytes("0123456789012345");
62 }
63 }
64

Sandbox

```
0 references
static void Main(string[] args)
{
    string stringToDecrypt = "UOuHAiCfl7lUSfwt3/+ZxA== ";
    byte[] Key = Encoding.UTF8.GetBytes("01234567890123456789012345678901");
    byte[] IV = Encoding.UTF8.GetBytes("0123456789012345");

    ICryptoTransform transform = new AesManaged().CreateDecryptor(Key, IV);
    byte[] array = Convert.FromBase64String(stringToDecrypt);
    MemoryStream memoryStream = new MemoryStream(array.Length);
    CryptoStream cryptoStream = new CryptoStream(memoryStream, transform, Crypt
    memoryStream.Write(array, 0, array.Length);
    memoryStream.Position = 0L;
    string result = new StreamReader(cryptoStream).ReadToEnd();
    cryptoStream.Close();
    Console.WriteLine(result);
    Console.ReadLine();
}
```

```
password
```


Recipe
Fork
Split delimiter
\n
Merge delimiter
\n
Ignore errors
From Base64
Alphabet
A-Za-z0-9+/=
Remove non-alphabet chars
AES Decrypt
Key
0123456789012345678901234567 8...
UTF8
IV
0123456789012345
UTF8
Mode
CBC
Input
Raw
Output
Raw
Input
length: 77
lines: 3
UOuHAiCfl7lUSfwt3/+ZxA==
YC
Yd
Output
start: 28
end: 28
length: 0
time: 4ms
length: 28
lines: 3
password
C
G

Hey Rob, what's this?

=$|=_=-_$
=-$=+=$$~$.~+=-
=_~_=~-|_~-.+.*|+
|*.+|*$

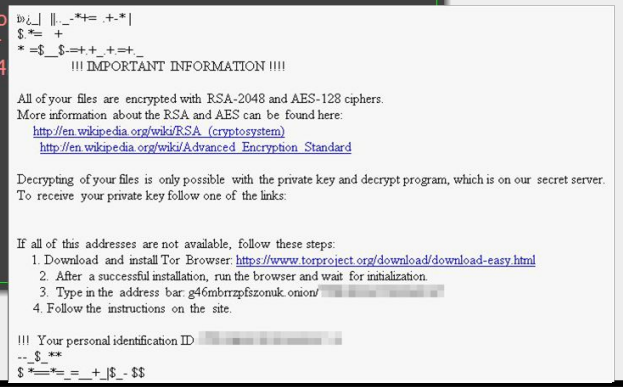
!!! IMPORTANT INFORMATION !!!!

All of your files are encrypted with RSA-2048 and AES-128 ciphers.
More information about the RSA and AES can be found here:
http://en.wikipedia.org/wiki/RSA_(cryptosystem)
http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

Decrypting of your files is only possible with the private key and decrypt program, which is on our secret server.
To receive your private key follow one of the links:

If all of this addresses are not available, follow these steps:
1. Download and install Tor Browser: https://www.torproject.o
2. After a successful installation, run the browser and wait for
3. Type in the address bar: g46mbrrzpfszonuk.onion/SU8DR34
4. Follow the instructions on the site.

!!! Your personal identification ID: [redacted] !!!
=_|_=-._+
=.|+*.|==+*.-|_
+$-_-.*_=~_==.=+~++
~_|+=_-~~.+_-*==|


»¿_| ||._-*+= .+-*|
$.*= +
* =$__$.=+.+_.+=+._
!!! IMPORTANT INFORMATION !!!!

All of your files are encrypted with RSA-2048 and AES-128 ciphers.
More information about the RSA and AES can be found here:
http://en.wikipedia.org/wiki/RSA_(cryptosystem)
http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

Decrypting of your files is only possible with the private key and decrypt program, which is on our secret server.
To receive your private key follow one of the links:

If all of this addresses are not available, follow these steps:
1. Download and install Tor Browser: https://www.torproject.org/download/download-easy.html
2. After a successful installation, run the browser and wait for initialization.
3. Type in the address bar: g46mbrrzpfszonuk.onion/[redacted]
4. Follow the instructions on the site.

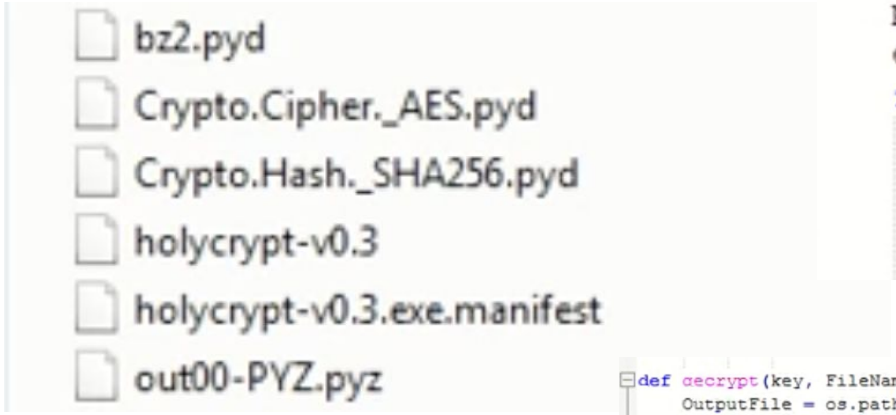
!!! Your personal identification ID [redacted]
--$_.**
$ *==*=_=__+_|$_- $$

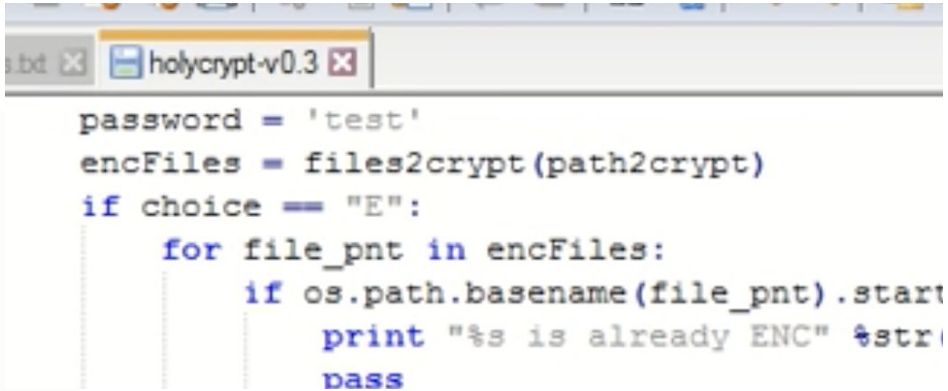
Locky Ransomware

Locky Ransomware

1. Downloads Public key from C&C
2. **CryptImportKey** using embedded key and public key from C&C
3. File encryption starts
4. Decryption without the private key is pretty close to impossible

HolyCrypt



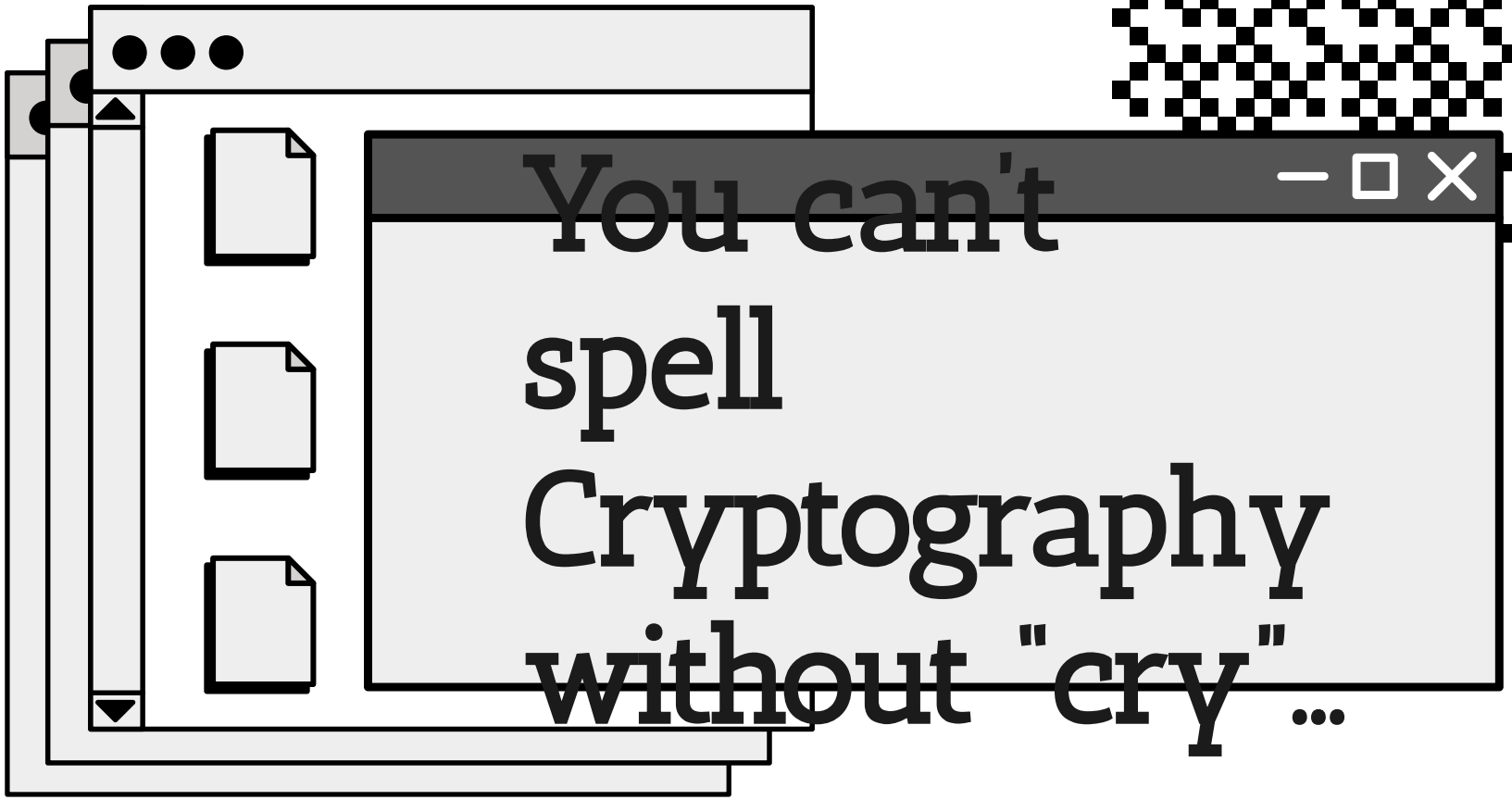



```
def decrypt(key, FileName):
    OutputFile = os.path.join(os.path.dirname(FileName), os.path.basename(FileName[11:]))
    chunkS = 64 * 1024
    with open(FileName, "rb") as infile:
        fileS = infile.read(16)
        IniVect = infile.read(16)

        decryptor = AES.new(key, AES.MODE_CBC, IniVect)

        with open(OutputFile, "wb") as outfile:
            while True:
                chunk = infile.read(chunkS)
                if len(chunk) == 0:
                    break
                outfile.write(decryptor.decrypt(chunk))
            outfile.truncate(int(fileS))
```

https://www.youtube.com/watch?v=r6BtA8p8kRU

Tools of the Trade

- CyberChef - https://gchq.github.io/CyberChef/
- DnSpy - https://github.com/dnSpy/dnSpy
- Hashcat - https://github.com/hashcat/hashcat
- John the Ripper - https://github.com/openwall/john
- JD-GUI - https://java-decompiler.github.io/


You can't
spell
Cryptography
without "cry"...

Thank you

Twitter: @mubix
Email: mubix@hak5.org
Come see me if you are looking for a job in Cyber Threat Intelligence :)