

Cunning with CNG: *SOLICITING SECRETS FROM SCHANNEL*

DEFCON 24

Why you might care

I.E. WHAT YOU GET TODAY

- ❑ Extracting TLS / SSL Keys (of various types) from memory
- ❑ Ability to decrypt TLS connections that use ephemeral key exchanges
 - ❑ For anything that uses Schannel: RDP, IE, Powershell, etc...pretty much anything .NET too
 - ❑ Past connections AND Future since the point of the cache is resumption
- ❑ Undocumented / partially documented structures elucidated
- ❑ TLS session caches mapped to the requesting processes, with SNIs
- ❑ A tool that does these things via Volatility/Rekall
- ❑ A paper that documents these things

How we get there

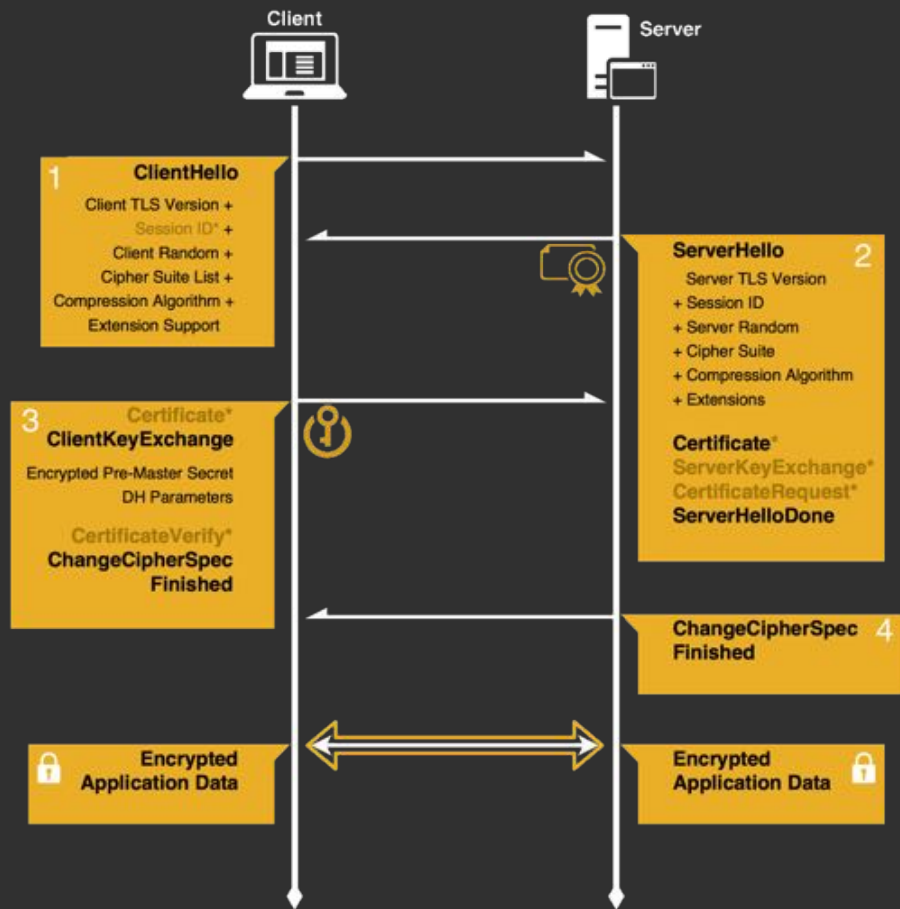
← THE AGENDA

1. Briefest of TLS Refreshers
2. How Schannel Works
3. The Secrets : }
4. The Other Forensic Artifacts!
5. A live demo >.>

A Disclaimer

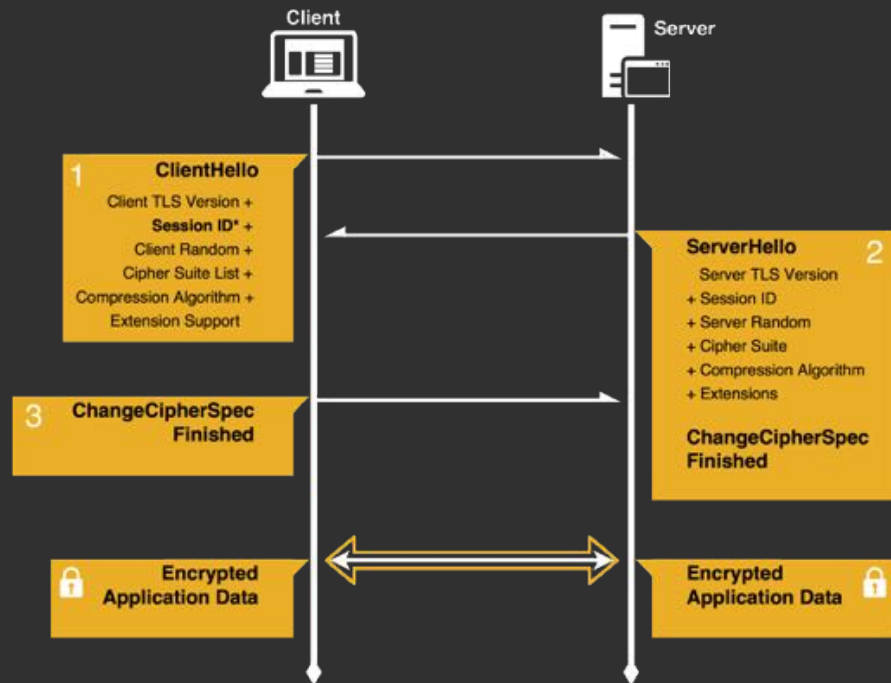
- ❑ This is NOT an exploit
 - ❑ It's the spec! :D
- ❑ Microsoft has done nothing wrong
 - ❑ To the contrary, their documentation was actually pretty great
- ❑ Windows doesn't track sessions for processes that load their own TLS libs
 - ❑ I'm looking at you Firefox and Chrome
- ❑ Windows doesn't track sessions for process that don't use TLS...
 - ❑ That'd be you teamviewer...
- ❑ This talk has nothing to do with Chanel
 - ❑ Sorry Aine.

The now infamous TLS Handshake



The ~~now~~ ^{DR:} infamous TLS Handshake

or, Session Resumption



Perfect Forward Secrecy < and what it means to TLS

What we *want* to do

- ❑ One time use keys, no sending secrets!

What TLS *actually* does

- ❑ Caches values to enable session resumption
 - ❑ recommends `An upper limit of 24 hours is suggested for session ID lifetimes`
- ❑ When using the session ticket extension, sending the encrypted state over the network
 - ❑ basically returning to the issue with RSA, but using a more ephemeral key...

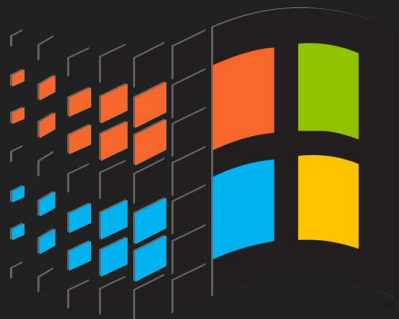
What implementations *also* do

- ❑ Store symmetric key schedules (so you can find the otherwise random keys...)
- ❑ Cache ephemeral keys and reuse for a while...



WHAT'S AN SCHANNEL?

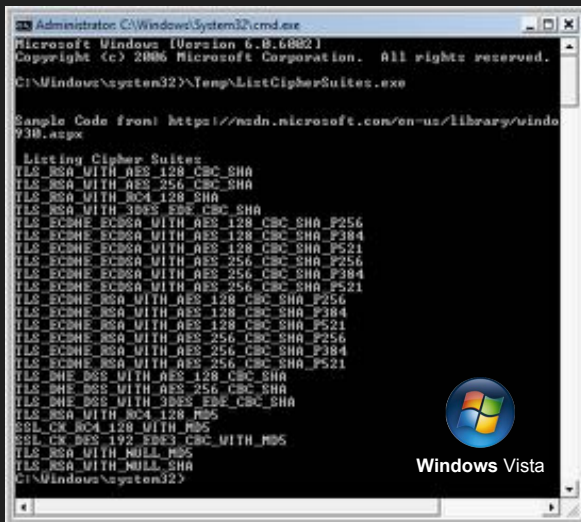
- ❑ It's TLS -> the Secure Channel for Windows!
- ❑ A library that gets loaded into the “key isolation process” **and** the “client” process
 - ❑ Technically a Security Support Provider (SSP)
- ❑ Spoiler: the key iso proc is LSASS



- ❑ Microsoft's CryptoAPI-Next Generation
- ❑ Introduced in Windows Vista
- ❑ Provides Common Criteria compliance
- ❑ Used to store secrets, also crypt them
 - ❑ The KSP & DPAPI for instance
- ❑ Important / reused keys are “isolated” from the less privileged/trusted “client” processes into the “key isolation process”
- ❑ Ncrypt is the “key storage router” and gateway to CNG Key Iso service

WHAT THE CNG?!

Schannel Cipher Suite Preferences

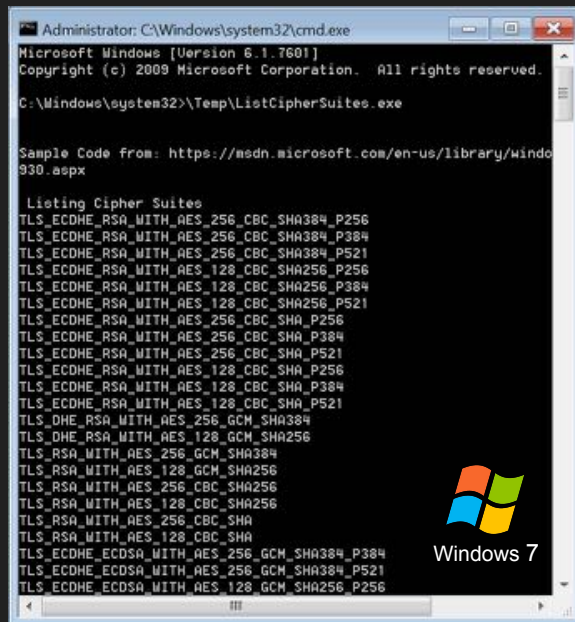


```
Administrator: C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.0.6002]
Copyright (c) 2006 Microsoft Corporation. All rights reserved.

C:\Windows\system32>C:\Temp>ListCipherSuites.exe

Sample Code from: https://msdn.microsoft.com/en-us/library/windows930.aspx

Listing Cipher Suites
TLS_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_AES_256_CBC_SHA
TLS_RSA_WITH_RC4_128_SHA
TLS_RSA_WITH_3DES_EDE_CBC_SHA
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA_P256
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA_P384
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA_P521
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA_P256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA_P384
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA_P521
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA_P256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA_P384
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA_P521
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA_P256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA_P384
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA_P521
TLS_DHE_DSS_WITH_AES_128_CBC_SHA
TLS_DHE_DSS_WITH_AES_256_CBC_SHA
TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA
TLS_RSA_WITH_RC4_128_MD5
TLS_RSA_WITH_RC4_128_SHA
TLS_RSA_WITH_NULL_MD5
C:\Windows\system32>
```

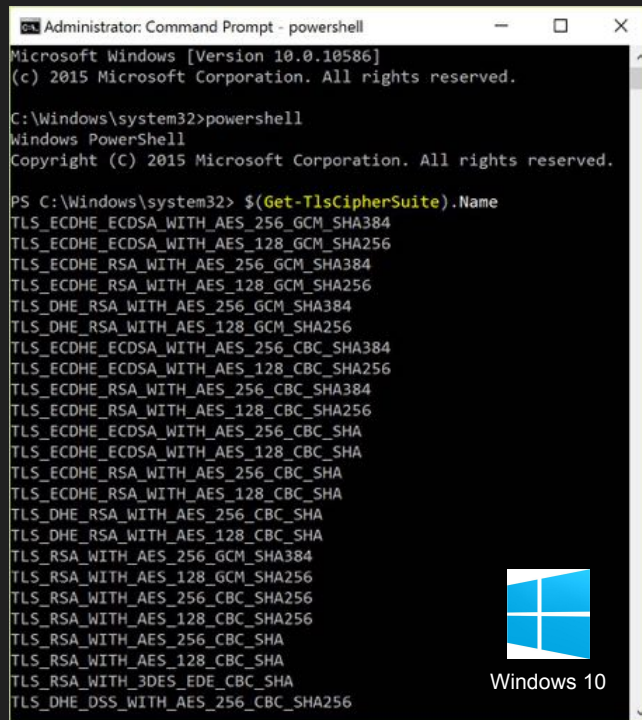


```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>C:\Temp>ListCipherSuites.exe

Sample Code from: https://msdn.microsoft.com/en-us/library/windows930.aspx

Listing Cipher Suites
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384_P256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384_P384
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384_P521
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256_P256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256_P384
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256_P521
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA_P256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA_P384
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA_P521
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA_P256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA_P384
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA_P521
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
TLS_RSA_WITH_AES_256_GCM_SHA384
TLS_RSA_WITH_AES_128_GCM_SHA256
TLS_RSA_WITH_AES_256_CBC_SHA256
TLS_RSA_WITH_AES_128_CBC_SHA256
TLS_RSA_WITH_AES_256_CBC_SHA
TLS_RSA_WITH_AES_128_CBC_SHA
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384_P384
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384_P521
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
```



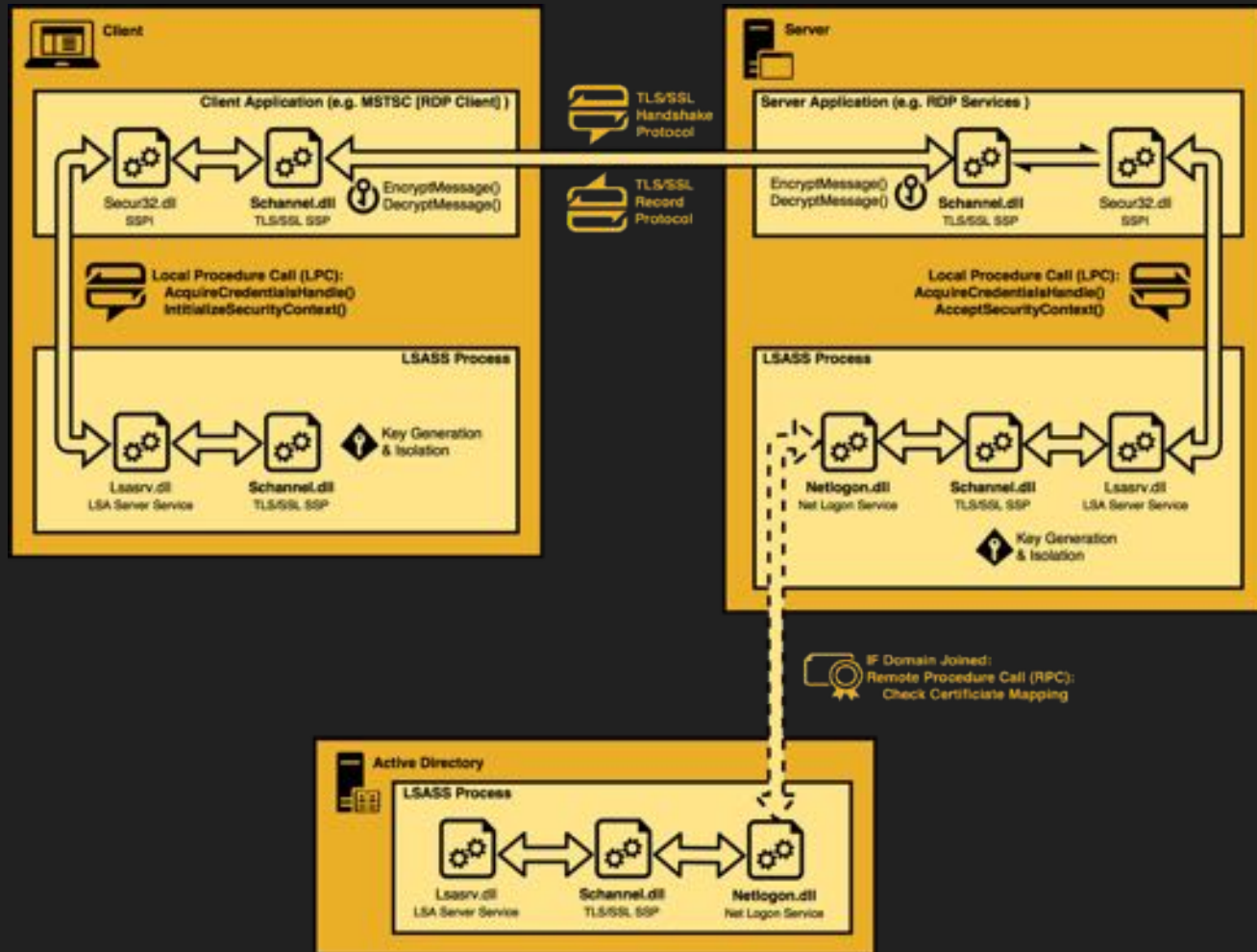
```
Administrator: Command Prompt - powershell
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Windows\system32>powershell
Windows PowerShell
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32>$(Get-TlsCipherSuite).Name
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
TLS_DHE_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_AES_256_GCM_SHA384
TLS_RSA_WITH_AES_128_GCM_SHA256
TLS_RSA_WITH_AES_256_CBC_SHA256
TLS_RSA_WITH_AES_128_CBC_SHA256
TLS_RSA_WITH_AES_256_CBC_SHA
TLS_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_3DES_EDE_CBC_SHA
TLS_DHE_DSS_WITH_AES_256_CBC_SHA256
```

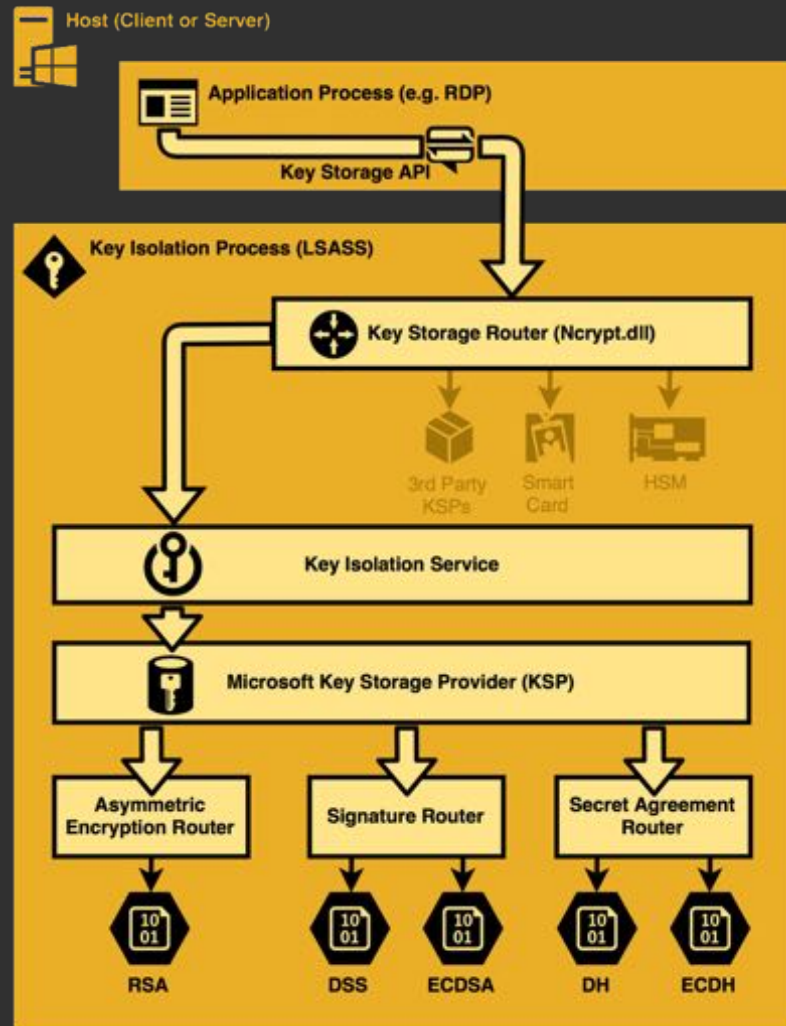
Schannel

by the docs



CNG Key Isolation

by the docs



Matching Session Keys

Basic Premis:

AES Keys are small and random

AES Key Schedules are larger and deterministic by design...they're a schedule.

Most implementations calculate schedule once and store it*

While a connection is active, both side NEED access to the symmetric keys used for encryption/verification

Matching Session Keys

So I scanned LSASS for cross-matched AES key schedules on both hosts...

And got nothing.

Well, no matches anyway.

A friendly neighborhood P.S.A.

ALL
V
RTFM

This announcement brought to you by an hour of wasted time

Matching Session Keys

RDP MSTSC AES Keys [Client]

```
C:\Windows\system32\cmd.exe
C:\TMP>findaes.exe rdp_mstsc.DMP
Searching rdp_mstsc.DMP
Found AES-256 key schedule at offset 0x3158ac:
b9 f0 65 ef 0a 27 33 62 0d 92 3d 2a 1e ba 24 3b 9a 1d 94 a8 70 d4 b5 ab 08 18 d6 f8 d8 04 1d 07
Found AES-256 key schedule at offset 0x3162ac:
b9 f0 65 ef 0a 27 33 62 0d 92 3d 2a 1e ba 24 3b 9a 1d 94 a8 70 d4 b5 ab 08 18 d6 f8 d8 04 1d 07
Found AES-256 key schedule at offset 0xcd71dc:
08 57 03 03 d2 71 5c bb 23 b1 69 1d b8 9a 2a ea 00 83 13 78 75 a5 84 a3 0f 21 af 5d 5c 4b 8c f9
Found AES-256 key schedule at offset 0xcd7bdc:
08 57 03 03 d2 71 5c bb 23 b1 69 1d b8 9a 2a ea 00 83 13 78 75 a5 84 a3 0f 21 af 5d 5c 4b 8c f9
Found AES-256 key schedule at offset 0xcfeadc:
00 75 6f 15 5c 70 a5 ec 8e 4c e3 c9 f3 b3 ff 33 80 04 ed 43 d4 a6 36 b7 6e 41 8f aa df 6c e1 b9
Found AES-256 key schedule at offset 0xcff4dc:
b0 75 6f 15 5c 70 a5 ec 8e 4c e3 c9 f3 b3 ff 33 80 04 ed 43 d4 a6 36 b7 6e 41 8f aa df 6c e1 b9
Found AES-256 key schedule at offset 0x171571c:
da 37 46 b4 a7 db e9 f5 b6 7f 27 ea a2 d3 26 c6 cc 65 30 42 f1 68 74 bb fb f5 c9 ef 64 f7 30 9c
Found AES-256 key schedule at offset 0x171611c:
da 37 46 b4 a7 db e9 f5 b6 7f 27 ea a2 d3 26 c6 cc 65 30 42 f1 68 74 bb fb f5 c9 ef 64 f7 30 9c
C:\TMP>
```

RDP SVCHost AES Keys [Server]

```
Select C:\Windows\system32\cmd.exe
C:\TMP>findaes.exe rdp_svchost.DMP
Searching rdp_svchost.DMP
Found AES-256 key schedule at offset 0x9bd7f50:
b9 f0 65 ef 0a 27 33 62 0d 92 3d 2a 1e ba 24 3b 9a 1d 94 a8 70 d4 b5 ab 08 18 d6 f8 d8 04 1d 07
Found AES-256 key schedule at offset 0x9c9b1c0:
b0 75 6f 15 5c 70 a5 ec 8e 4c e3 c9 f3 b3 ff 33 80 04 ed 43 d4 a6 36 b7 6e 41 8f aa df 6c e1 b9
Found AES-256 key schedule at offset 0x9c9bbc0:
00 75 6f 15 5c 70 a5 ec 8e 4c e3 c9 f3 b3 ff 33 80 04 ed 43 d4 a6 36 b7 6e 41 8f aa df 6c e1 b9
Found AES-256 key schedule at offset 0x9c9bf00:
da 37 46 b4 a7 db e9 f5 b6 7f 27 ea a2 d3 26 c6 cc 65 30 42 f1 68 74 bb fb f5 c9 ef 64 f7 30 9c
Found AES-256 key schedule at offset 0x9c9c900:
da 37 46 b4 a7 db e9 f5 b6 7f 27 ea a2 d3 26 c6 cc 65 30 42 f1 68 74 bb fb f5 c9 ef 64 f7 30 9c
Found AES-256 key schedule at offset 0x9ca8740:
b9 f0 65 ef 0a 27 33 62 0d 92 3d 2a 1e ba 24 3b 9a 1d 94 a8 70 d4 b5 ab 08 18 d6 f8 d8 04 1d 07
Found AES-256 key schedule at offset 0x9ca9140:
b9 f0 65 ef 0a 27 33 62 0d 92 3d 2a 1e ba 24 3b 9a 1d 94 a8 70 d4 b5 ab 08 18 d6 f8 d8 04 1d 07
Found AES-256 key schedule at offset 0x9cb97b0:
08 57 03 03 d2 71 5c bb 23 b1 69 1d b8 9a 2a ea 00 83 13 78 75 a5 84 a3 0f 21 af 5d 5c 4b 8c f9
Found AES-256 key schedule at offset 0x9cbalb0:
08 57 03 03 d2 71 5c bb 23 b1 69 1d b8 9a 2a ea 00 83 13 78 75 a5 84 a3 0f 21 af 5d 5c 4b 8c f9
C:\TMP>
```



The Session Key Structure

- ❑ Notice the value “31ss”
 - ❑ “31ss” -> “ss13”
 - ❑ Initially noticed while checking LSASS structs
- ❑ Structure is different in LSASS vs client process
- ❑ AES Key & Schedule highlighted
- ❑ Key and schedule appear multiple times in the same structure

```
0:000> db 000000d4`12d7adc0-100 L200
000000d4`12d7acc0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
000000d4`12d7acd0 00 00 01 01 00 00 00 00-00 00 00 00 00 00 00 00 .....
000000d4`12d7ace0 10 cb 25 12 d4 00 00 00-00 00 00 00 00 00 00 00 ..%.
000000d4`12d7acf0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 30 40 .....08
000000d4`12d7ad00 00 00 00 00 00 00 00 00-02 00 00 00 00 00 00 00 .....
000000d4`12d7ad10 ff ff ff ff 00 00 00 00-00 00 00 00 00 00 00 00 .....
000000d4`12d7ad20 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
000000d4`12d7ad30 7c 02 00 00 7c 02 00 00-33 6c 73 73 03 03 00 00 |.|.|.31ss...
000000d4`12d7ad40 28 c0 00 00 00 00 00 00-30 02 00 00 30 00 00 00 (.0...0...0
000000d4`12d7ad50 68 cf d7 ba de 11 55 b0-a0 9f 2f e6 23 33 54 0e h.....U.../.#3T.
000000d4`12d7ad60 61 df 6f a0 de af ea f4-cb 2e 28 0f 4d e3 2a 1b a.o.....(.M.*.
000000d4`12d7ad70 ae cb da 5e 2e 11 5f 14-9f a1 9f f2 dc 06 c0 e0 ...^.....
000000d4`12d7ad80 30 02 00 00 4b 53 53 4d-02 00 01 00 01 00 00 00 0...KSSM.....
000000d4`12d7ad90 10 00 00 00 00 01 00 00-20 00 00 00 cd 56 01 e) .....V..
000000d4`12d7ada0 62 04 f7 7d e5 bb 31 28-97 a9 9a be 1f 16 1e f4 b..).1(.....
000000d4`12d7adb0 9b 95 cd b5 eb 9a c3 14-e5 7b 36 46 00 00 00 00 9b 95 cd b5 eb 9a c3 14-e5 7b 36 46 00 00 00 00
000000d4`12d7adc0 cd 56 01 e1 62 04 f7 7d-e5 bb 31 28 97 a9 9a be cd 56 01 e1 62 04 f7 7d-e5 bb 31 28 97 a9 9a be
000000d4`12d7add0 1f 16 1e f4 9b 95 cd b5-eb 9a c3 14 e5 7b 36 4e 1f 16 1e f4 9b 95 cd b5 eb 9a c3 14 e5 7b 36 4e
000000d4`12d7ade0 ed 53 5b 38 8f 57 ac 45-6a ec 9d 6d fd 45 07 d3 .S[8.W.Ej...m.E..
000000d4`12d7adf0 4b 78 db 92 d0 ed 16 27-3b 77 d5 33 de 0c e3 75 Kx.....';w.3..u
000000d4`12d7ae00 11 42 c6 25 9e 15 6a 60-f4 f9 f7 0d 09 bc f0 de .B.%..j'.....
000000d4`12d7ae10 4a 1d 57 8f 9a f0 41 a8-a1 87 94 9b 7f 8b 77 ee J.W...A.....w.
000000d4`12d7ae20 28 b7 ee f7 b6 a2 84 97-42 5b 73 9a 4b e7 83 44 (. ....B[s.K..D
000000d4`12d7ae30 f9 89 bb 94 63 79 fa 3c-c2 fe 6e a7 bd 75 19 49 .....cy.<..n..u.I
000000d4`12d7ae40 bd 63 d5 8d 0b c1 51 1a-49 9a 22 80 02 7d a1 c4 .c....Q.I..")..
000000d4`12d7ae50 8e 76 89 88 ed 0f 73 b4-2f f1 1d 13 92 84 04 5a .v.....s./...2
000000d4`12d7ae60 f2 91 6b c2 f9 50 3a d8-b0 ca 18 58 b2 b7 b9 9c .k..P:....X....
000000d4`12d7ae70 b9 df df 56 54 d0 ac e2-7b 21 b1 f1 e9 a5 b5 ab ...VT:!(.....
000000d4`12d7ae80 d4 44 09 dc 2d 14 33 04-9d de 2b 5c 2f 69 92 c0 .D...-3...+\/i..
000000d4`12d7ae90 ac 26 90 ec f8 f6 3c 0e-83 d7 8d ff 6a 72 38 54 .6....<...jr8T
000000d4`12d7aea0 d4 43 29 de f9 57 1a da-64 89 31 86 4b e0 a3 46 .C)...W..d.1.K..F
000000d4`12d7aeb0 54 53 9f 6e 1e 73 a2 f3-le ce 5c aa f0 87 4a 49 TS.n.s....\...JI
```


The Session Key Structure

_SSL_SESSION_KEY	
4	cbStructLength
4	dwMagic ["ssl3"]
4	dwProtocolVersion
4/8	pvCipherSuiteListEntry
4	IsWriteKey
4/8	pvBcryptKeyStruct

_BCRYPT_KEY	
4	cbStructLength
4	dwMagic ["UUUR"]
4/8	pvBcryptProvider
4/8	pvBcryptSymmKey

_MS_SYMMETRIC_KEY	
4	cbStructLength
4	dwMagic ["MSSK"]
4	dwKeyType
...	...
4	KeyLength
?	SymmetricKey
?	SymmKeySchedule

```

Command - Dump
0:000> .foreach(key {s -[!w]a 0 17800000000000 3!ss}){.echo *** Session Key ***;dd $(^
*** Session Key ***
000000e1`7b047050 00000d2e                                     3!ss
000000e1`7b047054 73736c33
000000e1`7b047058 00000000`00000303
000000e1`7b047060 00007ffe`6fc11910 ncryptsslp!CipherSuiteList+0x1400
000000e1`7b047068 00000000`00000000
000000e1`7b047070 000000e1`7b0470c0 55555552`00000cbe
*
000000e1`7b0470c0 00000cbe
000000e1`7b0470c4 55555552 RUUU
000000e1`7b0470c8 000000e1`784e3af0 55555551`00000130
000000e1`7b0470d0 000000e1`7b0470e0 4d53534b`00000c80
000000e1`7b0470d8 00000000`00000000
*
000000e1`7b0470e0 00000c80 KSSM
000000e1`7b0470e4 4d53534b
000000e1`7b0470e8 00010002 00000005 00000010 00000001
000000e1`7b0470f8 00000100 00000001
000000e1`7b047100 000000e1`784e3c0 4d535341`00000028
000000e1`7b047118 00000020
* AES Key:
000000e1`7b04711c b0 75 6f 15 5c 70 a5 ec-8e 4c e3 c9 f3 b3 ff 33 .uo.\p...L.....3
000000e1`7b04712c 80 04 ed 43 d4 a6 36 b7-6e 41 8f aa df 6c e1 b9 ...C..6.nA...l..

*** Session Key ***
000000e1`7b047d90 00000d2e                                     3!ss
000000e1`7b047d94 73736c33
000000e1`7b047d98 00000000`00000303
000000e1`7b047da0 00007ffe`6fc11910 ncryptsslp!CipherSuiteList+0x1400
000000e1`7b047da8 00000000`00000001
000000e1`7b047db0 000000e1`7b047e00 55555552`00000cbe
*
000000e1`7b047e00 00000cbe
000000e1`7b047e04 55555552 RUUU
000000e1`7b047e08 000000e1`784e3af0 55555551`00000130
000000e1`7b047e10 000000e1`7b047e20 4d53534b`00000c80
000000e1`7b047e18 00000000`00000000
*
000000e1`7b047e20 00000c80 KSSM
000000e1`7b047e24 4d53534b
000000e1`7b047e28 00010002 00000005 00000010 00000001
000000e1`7b047e38 00000100 00000001
000000e1`7b047e40 000000e1`784e3c0 4d535341`00000028
000000e1`7b047e58 00000020
* AES Key:
000000e1`7b047e5c da 37 46 b4 a7 db e9 f5-b6 7f 27 ea a2 d3 26 c6 .7F.....*....6.
000000e1`7b047e6c cc 65 30 42 f1 68 74 bb-fb f5 c9 ef 64 f7 30 9c .e0B.ht.....d.0.

0:000> $(key)+!C L!r @ $t0 = $p;.echo *;dd @ $t0 L!;dc @ $t0+4 L!;dpp @ $t0+8 L!;dpp @
$t0+10 L2;.echo *;r $t0 = $p;dd @ $t0 L!;dc @ $t0+4 L!;dd @ $t0+8 L!;dpp @ $t0+20
L!;dd @ $t0+30+$ptrsize L!;.echo * AES Key::db @ $t0+34+$ptrsize L!dwo(@ $t0+30+
$ptrsize);.echo)
  
```

The Ncrypt SSL Provider [ncryptsslp.dll]

Ncryptsslp Validation function Symbols

```
Command - Dump \\vmware-host\Share...
0:000> x /1 ncryptsslp!*Validate*
ncryptsslp!SslpValidateEphemeralHandle
ncryptsslp!SslpValidateMasterKeyHandle
ncryptsslp!SslpValidateProvHandle
ncryptsslp!SslpValidateHashHandle
ncryptsslp!SslpValidateKeyPairHandle
0:000>
```

These functions do three things:

- ❑ Check the first dword for a size value
- ❑ Check the second dword for a magic ID
- ❑ Return the passed handle* if all is good

Master Key Validation Function Disassembly

```
Command - Dump \\vmware-host\Shared Folders\Documents\Challenges\zzTm...
0:000> uf ncryptsslp!SslpValidateMasterKeyHandle
ncryptsslp!SslpValidateMasterKeyHandle:
00007fff'df75b5b8 4885c9      test    rcx,rcx
00007fff'df75b5bb 7412        je      ncryptsslp!SslpValidateMasterKeyHandle+0x17

ncryptsslp!SslpValidateMasterKeyHandle+0x5:
00007fff'df75b5bd 833950      cmp     dword ptr [rcx],50h
00007fff'df75b5c0 720d        jb      ncryptsslp!SslpValidateMasterKeyHandle+0x17

ncryptsslp!SslpValidateMasterKeyHandle+0xa:
00007fff'df75b5c2 817904356c7373 cmp     dword ptr [rcx+4],73736C35h
00007fff'df75b5c9 7504        jne     ncryptsslp!SslpValidateMasterKeyHandle+0x17

ncryptsslp!SslpValidateMasterKeyHandle+0x13:
00007fff'df75b5cb 488bcl      mov     rax,rcx
00007fff'df75b5ce c3          ret

ncryptsslp!SslpValidateMasterKeyHandle+0x17:
00007fff'df75b5cf 33c0        xor     eax,eax
00007fff'df75b5d1 c3          ret
0:000>
```

*All handles in this case are explicitly pointers

SSL Magic	Size (x86)	Size (x64)	Validation Functions
ssl1	0xE4	0x130	SslpValidateProvHandle
ssl2	0x24	0x30	SslpValidateHashHandle
ssl3	?	?	<none>
ssl4	0x18	0x20	SslpValidateKeyPairHandle
ssl5	0x48	0x50	SslpValidateMasterKeyHandle
ssl6	0x18	0x20	SslpValidateEphemeralHandle
ssl7	?	?	<none>

Ncryptsslp “ssl3” symbols*

>_ Command - Dump C:\Defcon\Exa... - [] X

```
0:000> .foreach(magic {s -[1]a
00007fff`df750000 00007fff`df76f000 3lss}){ln
magic}
ncryptsslp!TlsGenerateSessionKeys+0x251
ncryptsslp!SPSslDecryptPacket+0x43
ncryptsslp!SPSslEncryptPacket+0x43
ncryptsslp!SPSslImportKey+0x19a
ncryptsslp!SPSslExportKey+0x76
ncryptsslp!SPSslFreeObject+0x1b
ncryptsslp!Ssl2GenerateSessionKeys+0x22c
ncryptsslp!Ssl2GenerateSessionKeys+0x294
```



ssl3 = session key struct

Ncryptsslp “ssl7” symbols*

>_ Command - Dump C:\Defcon\Exa... - [] X

```
0:000> lmm schannel
start          end          module
name
00007fff`ed1e0000 00007fff`ed254000  schannel

0:000> .foreach(magic {s -[1]a
00007fff`df750000 00007fff`df76f000 7lss}){ln
magic}
ncryptsslp!SPSslGenerateMasterKey+0x75
ncryptsslp!SPSslGenerateMasterKey+0x5595
ncryptsslp!SPSslGeneratePreMasterKey+0x15e
ncryptsslp!TlsDecryptMasterKey+0x 6b
```



ssl7 = pre-master secret struct?

The Master Secret

_SSL_MASTER_SECRET	
4	cbStructLength
4	dwMagic ["ssl5"]
4	dwProtocolVersion
0/4	dwUnknown1* [alignment?]
4/8	pvCipherSuiteListEntry
4	bIsClientCache
48	rgbMasterSecret
4	dwUnknown2 [reserved?]

```

Command - Dump \\vmware-host\Shared Folders\Documents\Challenges\zzTmp\...
0:000> .foreach(ms [s -[l]d 0 L?8000000000000000 'ssl5']) [.echo ***Raw Master Secret*** ^
***Raw Master Secret***
000000c9`86d9e980 50 00 00 00 35 6c 73 73-03 03 00 00 00 00 00 00 P...5lss.....
000000c9`86d9e990 10 1a 76 df ff 7f 00 00-00 00 00 00 01 7b 37 90 ..v.....{7.
000000c9`86d9e9a0 22 83 5c 78 ca 16 f7 12-53 00 39 fb 5d ea dd 03 ".\x....S.9.].
000000c9`86d9e9b0 a7 73 8a ba c7 a5 92 67-b7 45 97 2c 01 a3 25 15 .s.....g.E.,...%.
000000c9`86d9e9c0 44 0d fa d7 4c 45 c1 9a-25 a6 51 f1 00 00 00 00 D...LE..%.Q.....

***Parsed Master Secret***
000000c9`86d9e980 00000050
000000c9`86d9e984 73736c35 5lss
000000c9`86d9e988 00000000`00000303
000000c9`86d9e990 00007fff`df76la10 ncryptsslp!CipherSuiteList+0x1500
000000c9`86d9e998 00000000
* Secret:
000000c9`86d9e99c 01 7b 37 90 22 83 5c 78-ca 16 f7 12 53 00 39 fb .{7.".\x....S.9.
000000c9`86d9e9ac 5d ea dd 03 a7 73 8a ba-c7 a5 92 67 b7 45 97 2c ].s.....g.E.,
000000c9`86d9e9bc 01 a3 25 15 44 0d fa d7-4c 45 c1 9a 25 a6 51 f1 ..%.D...LE..%.Q.
*
000000c9`86d9e9cc 00000000

***Raw Master Secret***
000000c9`86d9f080 50 00 00 00 35 6c 73 73-03 03 00 00 00 00 00 00 P...5lss.....
000000c9`86d9f090 10 1a 76 df ff 7f 00 00-00 00 00 00 c3 05 82 ff ..v.....
000000c9`86d9f0a0 a5 6d ea 9c 0a ed 59 42-33 69 d8 ef b1 8a 79 6c .m....YB3i....yl
000000c9`86d9f0b0 60 5e 46 1e 7b 45 d0 12-88 71 14 c0 0b 32 86 ab `^F.{E...q...2..
000000c9`86d9f0c0 4a b4 84 1a fa 12 7a f0-3f 24 6b cb 00 00 00 00 J.....z.?%k.....

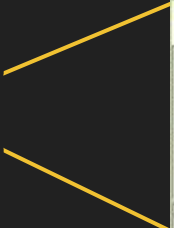
0:000> .foreach(ms [s -[l]d 0 L?8000000000000000 'ssl5']) [.echo ***Raw Master
Secret***;db ${ms}-4 Ldwo(${ms}-4);.echo;.echo ***Parsed Master Secret***;dd
${ms}-4 Ll;dc ms Ll;dp ${ms}+4 Ll;dps ${ms}+4+$ptrsize Ll;dd ${ms}+4+2*$ptrsize
Ll;.echo * Secret::;db ${ms}+3*$ptrsize L30;.echo *;dd ${ms}-4+dwo(${ms}-4)-4
Ll;.echo]

```

*Not present in x86 - either padding or part of previous member

The Master Secret

_SSL_MASTER_SECRET	
4	cbStructLength
4	dwMagic ["ssl5"]
4	dwProtocolVersion
0/4	dwUnknown1* [alignment?]
4/8	pvCipherSuiteListEntry
4	bIsClientCache
48	rgbMasterSecret
4	dwUnknown2 [reserved?]



```
Command - Dump \\vmware-host\Shared Folders\Documents\Challenges\zzTmp...
0:000> r @$t0 = 000000c9'86d9e980;dc @$t0 L2;dp @$t0+10 L1;.echo *;dpu poi($t0+10) L2
000000c9'86d9e980 00000050 73736c35 P...5lss
000000c9'86d9e990 00007fff'df761a10
*
00007fff'df761a10 0000c030'00000c00
00007fff'df761a18 00007fff'df762010 "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384"
0:000>
```

*Not present in x86 - either padding or part of previous member

Master Secret Mapped to Unique Identifier

- ❑ The Master Key is linked back to a unique ID through an “**NcryptSslKey**”
- ❑ The NcryptSslKey is referenced by an “**SessionCacheItem**”
- ❑ The SessionCacheItem contains either the SessionID, or a pointer and length value for a SessionTicket

At this point, we can identify and decrypt sessions robustly.

Schannel \$

Under the covers

X64 VOLATILITY VTYPE



```
'_SSL_SESSION_CACHE_CLIENT_ITEM': [ 0x140, {  
    'Vftable': [0x0, ['pointer64', ['void']]],  
    'NcryptKey': [0x10, ['pointer64', ['void']]],  
    'PublicCertificate': [0x18, ['pointer64', ['void']]],  
    'PublicKey': [0x28, ['pointer64', ['void']]],  
    'NcryptSslProv': [0x60, ['pointer64', ['void']]],  
    'SessionIdLen': [0x86, ['short short']],  
    'SessionId': [0x88, ['array', 0x20, ['unsigned char']]],  
    'ProcessId': [0xa8, ['unsigned long']],  
    'MaxLifeTime': [0xB0, ['unsigned long']],  
    'CertSerializedCertificateChain': [0xB0, ['pointer64',  
    ['void']]],  
    'UnkList1Flink': [0xB8, ['pointer64', ['void']]],  
    'UnkList1Blink': [0xC0, ['pointer64', ['void']]],  
    'UnkCacheList2Flink': [0xC8, ['pointer64', ['void']]],  
    'UnkCacheList2Blink': [0xD0, ['pointer64', ['void']]],  
    'ServerName': [0xF8, ['pointer64', ['void']]],  
    'CSessCacheManager': [0x110, ['pointer64', ['void']]],  
    'SessionTicket': [0x128, ['pointer64', ['void']]],  
    'SessionTicketLen': [0x130, ['int']],  
}],
```


Schannel \$

Under the covers

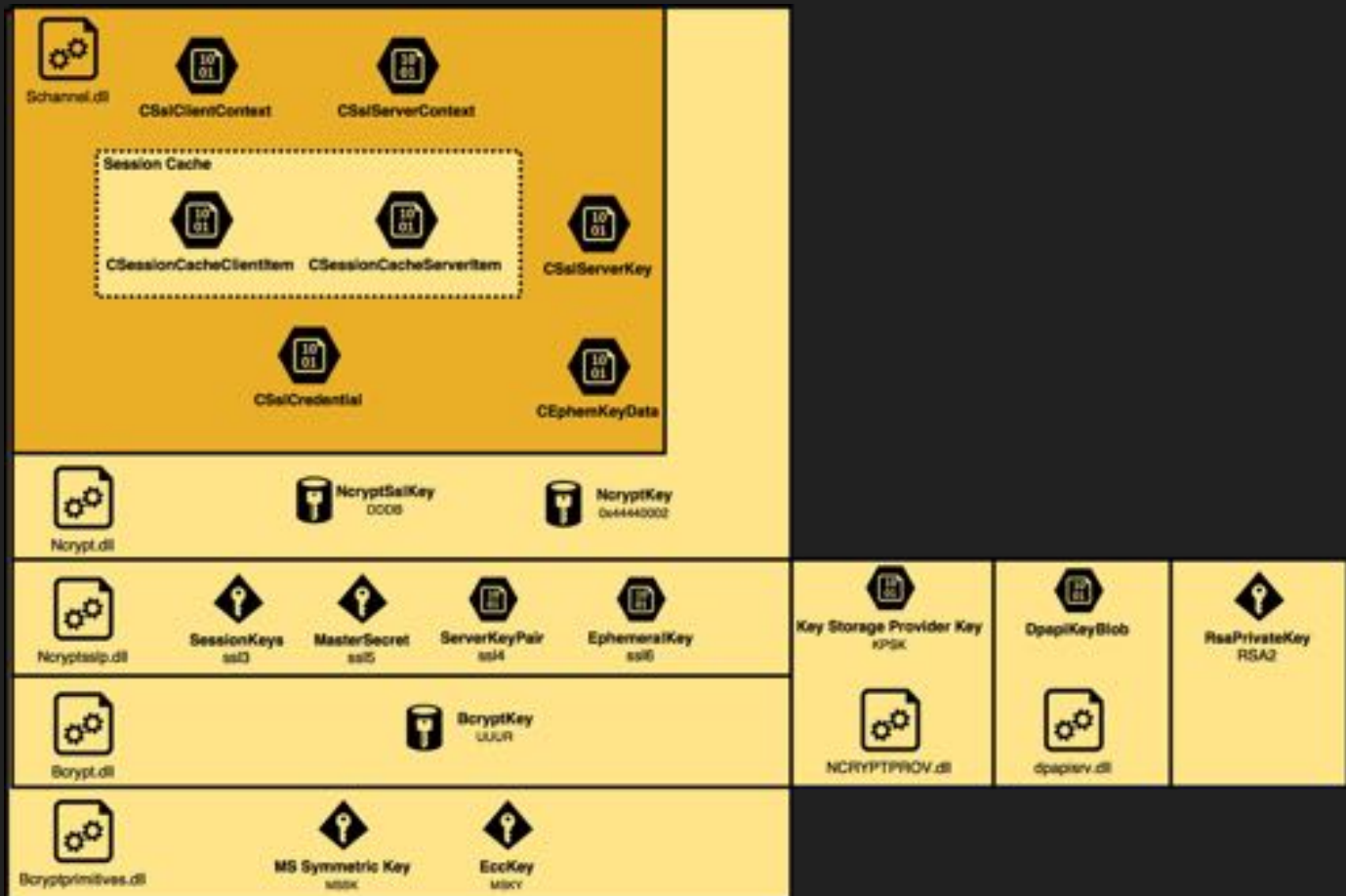
x64 VOLATILITY VTYPE



```
'_SSL_SESSION_CACHE_SERVER_ITEM': [ 0x110, {  
  'Vtable': [0x0, ['pointer64', ['void']]],  
  'NcryptKey': [0x10, ['pointer64', ['void']]],  
  'NcryptSslProv': [0x60, ['pointer64', ['void']]],  
  'SessionId': [0x88, ['array', 0x20, ['unsigned char']],  
  'ProcessId': [0xa8, ['unsigned long']],  
  'MaxLifeTime': [0xB0, ['unsigned long']],  
  'LastError?': [0xE8, ['unsigned long']],  
  'CSslCredential': [0xF0, ['pointer64', ['void']]],  
}],
```

Schannel \$

Under the covers



The Key Pairs

- ❑ The Server & Ephemeral Key Pairs use an identical structure
- ❑ The Key Type is compared with different values
 - ❑ ssl6 gets compared with a list stored in bcryptprimitives
 - ❑ ssl4 gets compared with a list stored in NCryptPROV
- ❑ The Key Storage Provider Key (KPSK) is referenced indirectly through an “Ncrypt Key” struct*

_SSL_KEY_PAIR	
4	cbStructLength
4	dwMagic ["ssl4" "ssl6"]
4	dwKeyType
4	dwUnknown1 [alignment?]
4/8	pvKspProvider
4/8	pvKspKey

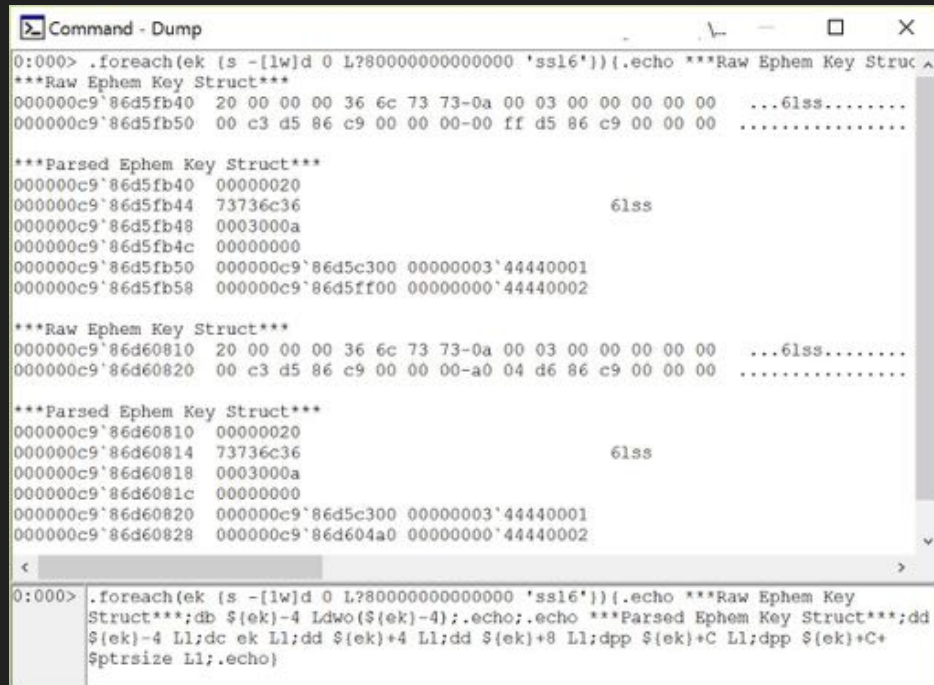
_NCRYPT_KEY	
4	cbStructLength
4	dwMagic [0x44440002]
4	dwKeyType
4	dwUnknown1 [alignment?]
4/8	pvKspProvider
4/8	pvKspKey

_KSP_KEY	
4	cbStructLength
4	dwMagic ["KSPK"]
4	dwKeyType
...	...
4/8	pvDpapiBlob
4/8	pvMSKY

*Not to be confused with an NcryptSslKeystruct

The Ephemeral Key Data

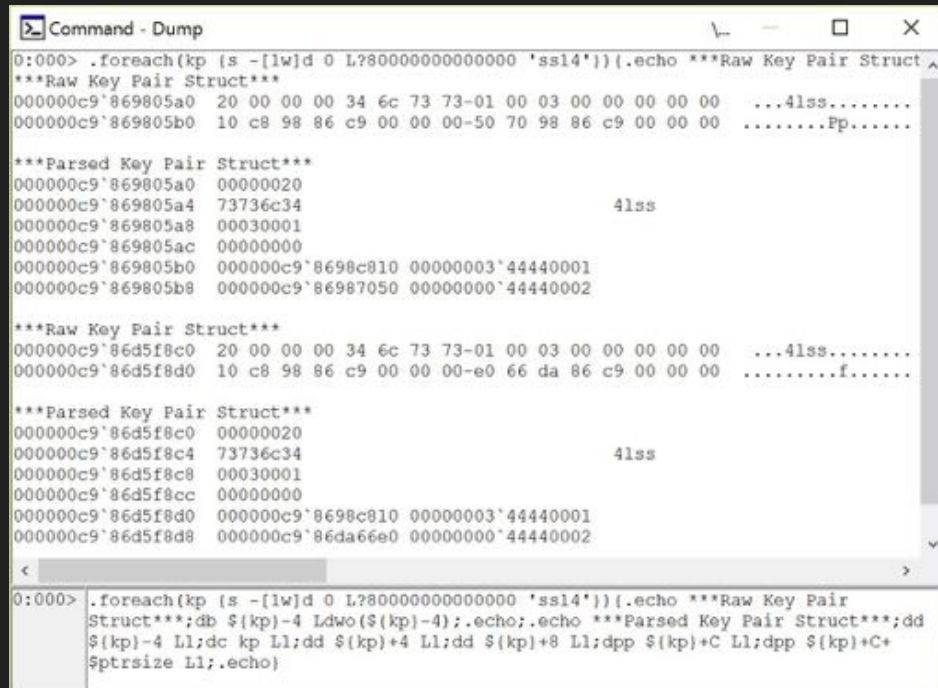
- ❑ Public Key is referenced by schannel!
CEphemKeyData
- ❑ Private Key is not stored in natively usable format, but is accessible
- ❑ The KPSK structure pointed to references another structure with the magic “MSKY” that appears to be the EccKey structure
- ❑ The KPSK structure has details about the curve selection / other valuable info



```
0:000> .foreach(ek [s -[lw]d 0 L?8000000000000000 'ssl6']) { .echo ***Raw Ephem Key Struct***  
***Raw Ephem Key Struct***  
000000c9`86d5fb40 20 00 00 00 36 6c 73 73-0a 00 03 00 00 00 00 00 ...6lss.....  
000000c9`86d5fb50 00 c3 d5 86 c9 00 00 00-00 ff d5 86 c9 00 00 00 .....  
  
***Parsed Ephem Key Struct***  
000000c9`86d5fb40 00000020  
000000c9`86d5fb44 73736c36 6lss  
000000c9`86d5fb48 0003000a  
000000c9`86d5fb4c 00000000  
000000c9`86d5fb50 000000c9`86d5c300 00000003`44440001  
000000c9`86d5fb58 000000c9`86d5ff00 00000000`44440002  
  
***Raw Ephem Key Struct***  
000000c9`86d60810 20 00 00 00 36 6c 73 73-0a 00 03 00 00 00 00 00 ...6lss.....  
000000c9`86d60820 00 c3 d5 86 c9 00 00 00-a0 04 d6 86 c9 00 00 00 .....  
  
***Parsed Ephem Key Struct***  
000000c9`86d60810 00000020  
000000c9`86d60814 73736c36 6lss  
000000c9`86d60818 0003000a  
000000c9`86d6081c 00000000  
000000c9`86d60820 000000c9`86d5c300 00000003`44440001  
000000c9`86d60828 000000c9`86d604a0 00000000`44440002  
  
<   
>  
0:000> .foreach(ek [s -[lw]d 0 L?8000000000000000 'ssl6']) { .echo ***Raw Ephem Key  
Struct***;db ${ek}-4 Ldwo(${ek}-4);.echo;.echo ***Parsed Ephem Key Struct***;dd  
${ek}-4 Ll;dc ek Ll;dd ${ek}+4 Ll;dd ${ek}+8 Ll;dpp ${ek}+C Ll;dpp ${ek}+C+  
$ptrsize Ll;.echo }
```

The Server (RSA) Private Key

- ❑ KSPK structure pointed to by the server key mostly resembles the file from disk
- ❑ The public information is parsed into fields
- ❑ The DPAPI protected private key blob is loaded into memory



```
0:000> .foreach(kp [s -[lw]d 0 L?8000000000000000 'ssl4']){.echo ***Raw Key Pair Struct***
***Raw Key Pair Struct***
000000c9'869805a0 20 00 00 00 34 6c 73 73-01 00 03 00 00 00 00 00 ...4lss.....
000000c9'869805b0 10 c8 98 86 c9 00 00 00-50 70 98 86 c9 00 00 00 .....Pp.....

***Parsed Key Pair Struct***
000000c9'869805a0 00000020
000000c9'869805a4 73736c34 4lss
000000c9'869805a8 00030001
000000c9'869805ac 00000000
000000c9'869805b0 000000c9'8698c810 00000003'44440001
000000c9'869805b8 000000c9'86987050 00000000'44440002

***Raw Key Pair Struct***
000000c9'86d5f8c0 20 00 00 00 34 6c 73 73-01 00 03 00 00 00 00 00 ...4lss.....
000000c9'86d5f8d0 10 c8 98 86 c9 00 00 00-e0 66 da 86 c9 00 00 00 .....f.....

***Parsed Key Pair Struct***
000000c9'86d5f8c0 00000020
000000c9'86d5f8c4 73736c34 4lss
000000c9'86d5f8c8 00030001
000000c9'86d5f8cc 00000000
000000c9'86d5f8d0 000000c9'8698c810 00000003'44440001
000000c9'86d5f8d8 000000c9'86da66e0 00000000'44440002

0:000> .foreach(kp [s -[lw]d 0 L?8000000000000000 'ssl4']){.echo ***Raw Key Pair
Struct***;db $(kp)-4 Ldwo($(kp)-4);.echo;.echo ***Parsed Key Pair Struct***;dd
$(kp)-4 Ll;dc kp Ll;dd $(kp)+4 Ll;dd $(kp)+8 Ll;dpp $(kp)+C Ll;dpp $(kp)+C+
$ptrsize Ll;.echo}
```

The Server (RSA) Private Key

```
defc0n% cd ./ProgramData/RSA/MachineKeys
defc0n% xxd -g 1 -s 0x165 f686aace6942fb7f7ceb231212eef4a4_7496afd3-d13f-4cf7-b6
d9-ca3d0c3ff959
0000165: 01 00 00 00 d0 8c 9d df 01 15 d1 11 8c 7a 00 c0 .....Z..
0000175: 4f c2 97 eb 01 00 00 00 66 68 6a f9 d8 1b d1 4a 0.....fhj....J
0000185: 85 fc 1a 77 28 7d 5c d1 04 00 00 00 2c 00 00 00 .....w()\\.....
0000195: 43 00 72 00 79 00 70 00 74 00 6f 00 41 00 50 00 C.r.y.p.t.o.A.P.
00001a5: 49 00 20 00 50 00 72 00 69 00 76 00 61 00 74 00 I..P.r.i.v.a.t.
00001b5: 65 00 20 00 4b 00 65 00 79 00 00 00 10 66 00 00 e..K.e.y....f..
00001c5: 00 01 00 00 20 00 00 00 8a 60 40 b8 f7 4f ec f9 .....?@...@...
00001d5: 37 6f cc 0b 14 82 e6 3f 40 79 65 5f 94 51 a3 75 7o.....?@y...Q.u
00001e5: 5a da e5 6f 81 89 ff d4 00 00 00 0e 80 00 00 Z.o.....
00001f5: 00 02 00 00 20 00 00 00 d2 41 1d a7 b8 f7 ce b4 .....A.....
0000205: 51 a6 85 13 39 0d da f1 00 54 ce e7 04 a8 e0 17 Q...9.....T.....
0000215: a7 9d c6 98 df 6f ef a3 50 05 00 00 2b 9f 70 ce .....o.P....+p.
0000225: 0c 3f fb f1 3f a6 78 87 0c 47 d9 b0 60 33 5d 27 .?..?..x..G...'}'
0000235: 82 af 5d eb b7 21 b2 36 2a 58 a2 88 56 61 69 8c .}...!.6*X..Vai.
0000245: 3e 11 20 ff 27 24 b5 dc e9 b2 fd 3d b0 c9 5e 31 >...'.S.....^1
0000255: e6 5e 5e de 81 a9 78 ea ea 16 c7 52 a4 70 9b 34 V^...x.....R.p.4
0000265: 7c 6c b8 9a 86 fb 02 d7 e5 a5 c2 e3 be 2e c7 65 |l.....nv.|
0000275: 21 f1 99 0a 5b 0d 34 98 ad 10 af 45 b7 79 f5 3e !...!.4.....E.y.>
0000285: 8a 95 be 29 83 be 68 74 78 64 d1 b3 db 13 2d 10 .....htxd.....-
0000295: 42 d0 95 f5 02 d4 9a 97 87 00 b1 6e 76 d0 7c B.....nv.|
00002a5: e1 67 d1 90 94 ea b0 9e a7 bd 37 12 2f 48 76 56 .g.....7./HvV
00002b5: 25 94 e9 cf 28 f6 ae 6e dc ba f3 77 0b b2 ce 26 %...(.n...w...&
00002c5: fa 33 32 0b b9 13 48 9a 77 0f b7 47 29 92 da c7 .32...H.w..G)...
00002d5: 7a 21 aa 12 04 8c 0b 27 6e fd 24 48 ab 91 8c 98 z!.....'n.$H..
00002e5: 3d 68 7c 0b 48 91 58 f7 6e d2 85 d8 a9 ec 2a ac =h|.H.X.n.....*
00002f5: 9d b3 39 e5 51 24 e1 d9 a1 eb 51 64 12 8b 2a c5 .9.Q$.A.Qd..*e
0000305: 62 4e ce a4 83 b1 e9 a7 0a a1 46 d5 46 fe 4b c3 bN.....F.F.K.
0000315: f2 8e fa d9 28 b9 38 86 1a 84 95 58 93 db d2 40 .....(.B...X...@
0000325: 5f 4b 47 bc 95 21 ce bc b3 a2 db 12 47 37 18 68 _KG...Q.....G7.h
0000335: fe c6 f9 55 9a 28 61 c5 c8 8a 55 07 04 ef 3a 2a _U.(a...U....*
0000345: 3b d2 b8 2e 26 09 6f c1 a4 b5 7d 82 93 35 a6 ;...&.o...u...5.
0000355: 00 aa 92 14 9c 77 10 af b9 05 93 af 3a 47 6d d2 .....w.....Gm.
0000365: a3 b8 d8 cf 98 72 72 e4 95 9e 07 ed 4d 7d 28 2e .....f.....M).
0000375: fe c8 d0 bd 42 75 26 fb e9 94 0c ea af 03 4d cd .....Bu&.....(
0000385: e1 3e 98 07 4e 3c 87 53 80 76 93 c6 bd 15 c3 47 .>..N<..S.v....G
0000395: 0c aa af 20 88 11 84 15 0b 71 64 32 35 fd a7 2d ... ..qd25..-
```

```
Command - Dump \\vmware-host\Shared Folders\Documents\Challenges\zzTmp\...
0:000> .foreach(key {s-[w]q 0x0 L7800000000000 schannel1CSessionCacheServerItem::v
0000000c9`85d06630 01 00 00 00 d0 8c 9d df-01 15 d1 11 8c 7a 00 c0 .....Z..
0000000c9`85d06640 4f c2 97 eb 01 00 00 00 66 68 6a f9 d8 1b d1 4a 0.....fhj....J
0000000c9`85d06650 85 fc 1a 77 28 7d 5c d1-04 00 00 00 2c 00 00 00 .....w()\\.....
0000000c9`85d06660 43 00 72 00 79 00 70 00 74 00 6f 00 41 00 50 00 C.r.y.p.t.o.A.P.
0000000c9`85d06670 49 00 20 00 50 00 72 00 69 00 76 00 61 00 74 00 I..P.r.i.v.a.t.
0000000c9`85d06680 65 00 20 00 4b 00 65 00 79 00 00 00 10 66 00 00 e..K.e.y....f..
0000000c9`85d06690 00 01 00 00 20 00 00 00 8a 60 40 b8 f7 4f ec f9 .....?@...@...
0000000c9`85d066a0 37 6f cc 0b 14 82 e6 3f-40 79 65 5f 94 51 a3 75 7o.....?@y...Q.u
0000000c9`85d066b0 5a da e5 6f 81 89 ff d4-00 00 00 00 0e 80 00 00 Z.o.....
0000000c9`85d066c0 00 02 00 00 20 00 00 00 d2 41 1d a7 b8 f7 ce b4 .....A.....
0000000c9`85d066d0 51 a6 85 13 39 0d da f1-00 54 ce e7 04 a8 e0 17 Q...9.....T.....
0000000c9`85d066e0 a7 9d c6 98 df 6f ef a3-50 05 00 00 2b 9f 70 ce .....o.P....+p.
0000000c9`85d066f0 0c 3f fb f1 3f a6 78 87-0c 47 d9 b0 60 33 5d 27 .?..?..x..G...'}'
0000000c9`85d06700 82 af 5d eb b7 21 b2 36-2a 58 a2 88 56 61 69 8c .}...!.6*X..Vai.
0000000c9`85d06710 3e 11 20 ff 27 24 b5 dc-e9 b2 fd 3d b0 c9 5e 31 >...'.S.....^1
0000000c9`85d06720 e6 5e 5e de 81 a9 78 ea-ea 16 c7 52 a4 70 9b 34 V^...x.....R.p.4
0000000c9`85d06730 7c 6c b8 9a 86 fb 02 d7-e5 a5 c2 e3 be 2e c7 65 |l.....nv.|
0000000c9`85d06740 21 f1 99 0a 5b 0d 34 98-ad 10 af 45 b7 79 f5 3e !...!.4.....E.y.>
0000000c9`85d06750 8a 95 be 29 83 be 68 74-78 64 d1 b3 db 13 2d 10 .....htxd.....-
0000000c9`85d06760 42 d0 95 f5 02 d4 9a 97-87 00 b1 6e 76 d0 7c B.....nv.|
0000000c9`85d06770 e1 67 d1 90 94 ea b0 9e-a7 bd 37 12 2f 48 76 56 .g.....7./HvV
0000000c9`85d06780 25 94 e9 cf 28 f6 ae 6e-dc ba f3 77 0b b2 ce 26 %...(.n...w...&
0000000c9`85d06790 fa 33 32 0b b9 13 48 9a-77 0f b7 47 29 92 da c7 .32...H.w..G)...
0000000c9`85d067a0 7a 21 aa 12 04 8c 0b 27-6e fd 24 48 ab 91 8c 98 z!.....'n.$H..
0000000c9`85d067b0 3d 68 7c 0b 48 91 58 f7-6e d2 85 d8 a9 ec 2a ac =h|.H.X.n.....*
0000000c9`85d067c0 9d b3 39 e5 51 24 e1 d9-41 eb 51 64 12 8b 2a c5 .9.Q$.A.Qd..*e
0000000c9`85d067d0 62 4e ce a4 83 b1 e9 a7-0a a1 46 d5 46 fe 4b c3 bN.....F.F.K.
0000000c9`85d067e0 f2 8e fa d9 28 b9 38 86-1a 84 95 58 93 db d2 40 .....(.B...X...@
0000000c9`85d067f0 5f 4b 47 bc 95 21 ce bc-b3 a2 db 12 47 37 18 68 _KG...Q.....G7.h
0000000c9`85d06800 fe c6 f9 55 9a 28 61 c5-c8 8a 55 07 04 ef 3a 2a _U.(a...U....*
0000000c9`85d06810 3b d2 b8 e8 26 09 6f c1-a4 b5 7d 82 93 35 a6 ;...&.o...u...5.
0000000c9`85d06820 00 aa 92 14 9c 77 10 af-b9 05 93 af 3a 47 6d d2 .....w.....Gm.
0000000c9`85d06830 a3 b8 d8 cf 98 72 72 e4-95 9e 07 ed 4d 7d 28 2e .....f.....M).
0000000c9`85d06840 fe c8 d0 bd 42 75 26 fb-e9 94 0c ea af 03 4d cd .....Bu&.....(
0000000c9`85d06850 e1 3e 98 07 4e 3c 87 53-80 76 93 c6 bd 15 c3 47 .>..N<..S.v....G
0000000c9`85d06860 0c aa af 20 88 11 84 15-0b 71 64 32 35 fd a7 2d ... ..qd25..-
<
0:000> .foreach(key {s-[w]q 0x0 L7800000000000 schannel1CSessionCacheServerItem::v
'vtable'}) {db poi (poi (poi (poi (poi (poi (poi (poi ($ {key}+F0)+48))+8)+10)+18)+
10)+D0) L700; echo}
```

Windows Vista

- ❑ CNG was introduced in Vista
- ❑ The Vista cache is different
- ❑ It's kinda proto-CNG
- ❑ Prior to Ncryptsslp (Sslp functions are in Ncrypt)
- ❑ Instead of Classes, the cache is just a doubly-linked list
- ❑ No RFC5088 support (no tickets)

x86 VOLATILITY VTYPE



```
'_SSL_SESSION_CACHE_CLIENT_ITEM': [ 0xf0, {  
    'Flink': [0x0, ['pointer', ['void']]],  
    'Blink': [0x4, ['pointer', ['void']]],  
    'ProcessId': [0x8, [['unsigned long']],  
    'MasterKey': [0x14, ['pointer', ['NcryptSslKey']],  
    'CipherSuiteId': [0x1C, ['pointer', ['void']]],  
    'ECCurveParam': [0x20, ['pointer', ['void']]],  
    'NcryptSslProv': [0x28, ['pointer', ['void']]],  
    'PublicCertificate': [0x2C, ['pointer', ['void']]],  
    'PublicCert2': [0x34, ['pointer', ['void']]],  
    'PublicKeyStruct': [0x3C, ['pointer', ['void']]],  
    'PublicCertStruct3': [0x44, ['pointer', ['void']]],  
    'ServerName': [0x80, ['pointer', ['void']]],  
    'SessionIdSize': [0x94, ['short short']],  
    'SessionId': [0x98, ['array', 0x20, ['unsigned char']]],  
    'ErrorCode': [0xEC, ['pointer64', ['void']]],  
}],
```


Windows Vista

```
Command - Dump \\vmware-host\Shared Folders\Documents\Challenge...

*** Cache Item ***
* ProcId:
001d76f0 00000cf8
* NcryptSslKey:
001d76fc 001dab40 00000018
* SNI:
001d7768 01f9e480 "live.sysinternals.com"
* SessionID:
001d7780 59 19 00 00 07 4a 6c cc-d6 b0 e2 b2 5f cd d1 30 Y....Jl.....0
001d7790 bf ee 06 b1 ec 20 e3 57-e3 79 52 72 d7 f5 a5 41 .....W.yRr...A

*** Cache Item ***
* ProcId:
001d7828 00000cf8
* NcryptSslKey:
001d7834 001dabe0 00000018
* SNI:
001d78a0 01fa3cb8 "www.torproject.org"
* SessionID:
001d78b8 ba ce 7b 7e ca 6d e8 15-92 e8 ae fb 08 bb 71 83 ..{~.M.....q.
001d78c8 e7 87 ed 78 e5 12 f3 c0-24 a3 b6 0b e8 a2 43 b9 ...X....$......C.

*** Cache Item ***
* ProcId:
001d7960 00000cf8
* NcryptSslKey:
001d796c 01fa3f98 00000018
* SNI:
001d79d8 01fa3d18 "urs.microsoft.com"
* SessionID:
001d79f0 99 0e 00 00 d8 3f de 02-53 c3 68 49 59 89 c2 c0 .....?.S.hIY...
001d7a00 71 ca bd 0f 5f 7b bd 59-08 6c df 44 8c a7 b7 7b q..._.{Y.l.D...{

*** Cache Item ***
* ProcId:
001d7e40 00000cf8
* NcryptSslKey:
001d7e4c 01fa3ed8 00000018
* SNI:
001d7eb8 01f75d88 "login.live.com"
* SessionID:
001d7ed0 f6 07 00 00 5d 3d bc aa-f7 91 9a 5e f5 3e b7 10 ....]=.....^.>..
001d7ee0 ab dc 7c d1 1f 3a 0a 95-08 02 80 cc ee 92 4c d1 ..l.:.....L.

*** Cache Item ***
* ProcId:
001d7bd0 00000cf8
* NcryptSslKey:
001d7bd4 01fa3f30 00000018

0:000> !list -x *.echo *** Cache Item ***;.echo * ProcId::dd @$extret+8 L1;.echo
* NcryptSslKey::dpp @$extret+14 L1;.echo * SNI::dpu @$extret+80 L1;.echo *
SessionID::dd @$extret+98 L20* 001d8ba0
```


The Forensic Context

- ❑ Active Connection = Security Context
- ❑ ProcessID for client process stored
- ❑ Server Name Indicator (SNI) stored in the cache as well
- ❑ Cache Lifetime of 10 hours
- ❑ Session IDs are arbitrary, but not always random
 - ❑ Schannel is the perfect example, can be fingerprinted
- ❑ If the system is a client, why would it have a server cache?
 - ❑ RDP for one, almost guaranteed to live 10 hours (unless there are 20,000 connections afterward)

Global Schannel Variables of Significance:

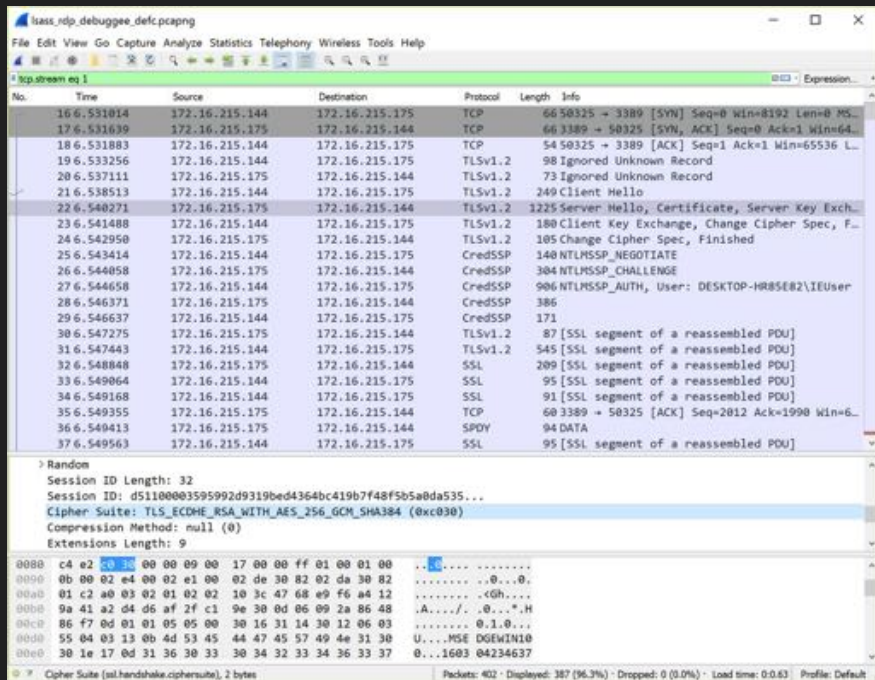
```
schannel!CSslGlobals::m_dwMaximumEntries  
schannel!CSslGlobals::m_dwClientLifespan  
schannel!CSslGlobals::m_dwServerLifespan  
schannel!CSslGlobals::m_dwSessionTicketLifespan
```

Extracting the Secrets

- ❑ Volatility & Rekall plugins
- ❑ By default (no args):
 - ❑ will automatically find lsass
 - ❑ will scan the heap
 - ❑ Can be configured to scan Writeable VADs, or full VAS
 - ❑ dumps to stdout in wireshark format
 - ❑ Can dump verbose object as json
- ❑ Hoping to have functionality integrated into PowerShell module soon
 - ❑ Got busy : <

```
defc0n% vol.py --plugins=./plugins --profile=Win10x64 -f ./Win10-Test-c2a4a77d.vm  
em lsasslkey  
Volatility Foundation Volatility Framework 2.5  
RSA Session-ID:b93c0000a110690b4ae9111bce5725c6c47a037b3c39c49c75ce51e1c2eb79ee M  
aster-Key:bc28467999b99fd3dfd3a24642c5d93b9ab43e51627f6e0145ef120ba98a1c3223f3dbe  
0154e30d7869bdb7ab66f5318  
RSA Session-ID:17330000f84a86aebb2c5de0af20e6d5c2cab95ab65043e14c6e19cee54ee17 M  
aster-Key:9dd750e12e6e4439b08326d4a1f9eba2d2fe65c2a26c2088e7cec22ce1d91e9f219b704  
547a2b2eccb9a81d557d5a1a  
RSA Session-ID:3c2c000024b8f70dd2613d8b13d0c4ac4daaefbe53ab4b7cb9763e80feccb4f1 M  
aster-Key:2d119c64695ffc9c143c136471f5625d8cde92d35721f5f2849b92639603799a45e1e60  
1786cbf89b00c186969d44983  
RSA Session-ID:d4170000da09f8596739215e216c496568fa66e42ac32b974d440949dff33d2b M  
aster-Key:44b503bef7842ea9a416fbf8b63b932b23b7b687fbf5297b253eac427877c8e11595e14  
c3f00c40bf2a0f4688de0b7aa  
RSA Session-ID:432a0000bf4f622f0fc119974a0ef30cd838c3a025b83abbdcdcbce7b2325d2d9 M  
aster-Key:552699d61e21d1b871af4b05a54003bf03eade60666dd1e54b94c3b5ec98f296db4ae99  
baed4e23882175e5fffd88be31  
RSA Session-ID:6f230000a021aac48d15544524c1454e4ec01d5adb305d8d9d57ab2b991dd597 M  
aster-Key:8bc9e9df653e3cbf533be84c6897787bd453b8cee9d5389e9c3659ebf997d9c8d0666aa  
dca5be2258f30b9251215a717
```

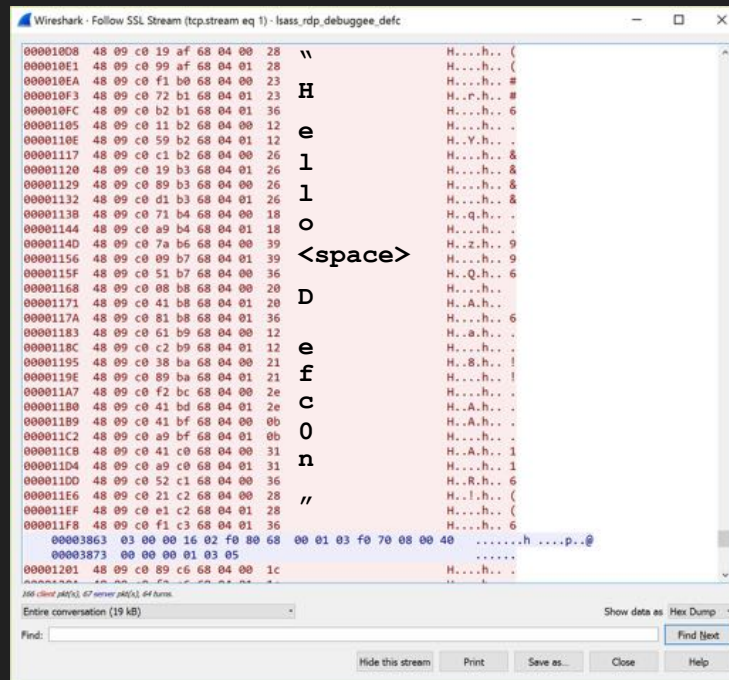
Decrypting an RDP Session (Ephemeral XCHG)



Wireshark packet capture of an RDP session. The packet list shows a TLS handshake sequence. The packet details pane for packet 22 (SSL segment of a reassembled PDU) shows the following information:

- Session ID Length: 32
- Session ID: d51100003595992d9319bed4364bc419b7f48f5b5a0da535...
- Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
- Compression Method: null (0)
- Extensions Length: 9

The packet bytes pane shows the raw data of the TLS segment, including the cipher suite and extensions.



Wireshark packet capture of an RDP session. The packet list shows a sequence of packets. The packet details pane for packet 100 (XCHG packet) shows the following information:

- Session ID Length: 32
- Session ID: d51100003595992d9319bed4364bc419b7f48f5b5a0da535...
- Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
- Compression Method: null (0)
- Extensions Length: 9

The packet bytes pane shows the raw data of the XCHG packet, including the cipher suite and extensions.

Decrypting an RDP Session (Ephemeral 🔑 XCHG)

^E
~~DO~~MO TIME



Fin

[@TinRabbit_](#)



Questions?

Special Thanks

For general support, helpful comments, their time, and encouragement.

- ❑ **Áine Doyle** - Badass Extraordinaire (OCSC)
- ❑ **Dr. John-Ross Wallrabenstein** - Sypris Electronics
- ❑ **Dr. Marcus Rogers** - Purdue Cyber Forensics Laboratory
- ❑ **Michael Hale Ligh (MHL)** - Volexity
- ❑ **Tatiana Ringenberg** - Sypris Electronics