TROOPERS

Make the world a safer place.

What Happens In Windows 7 Stays In Windows 7

Moti Joseph & Marion Marschalek

Troopers Conference 2014

About Us

Joseph Moti
Security Researcher

Marion Marschalek
Malware Analyst

8731-7364-1932-9646-3040

Agenda

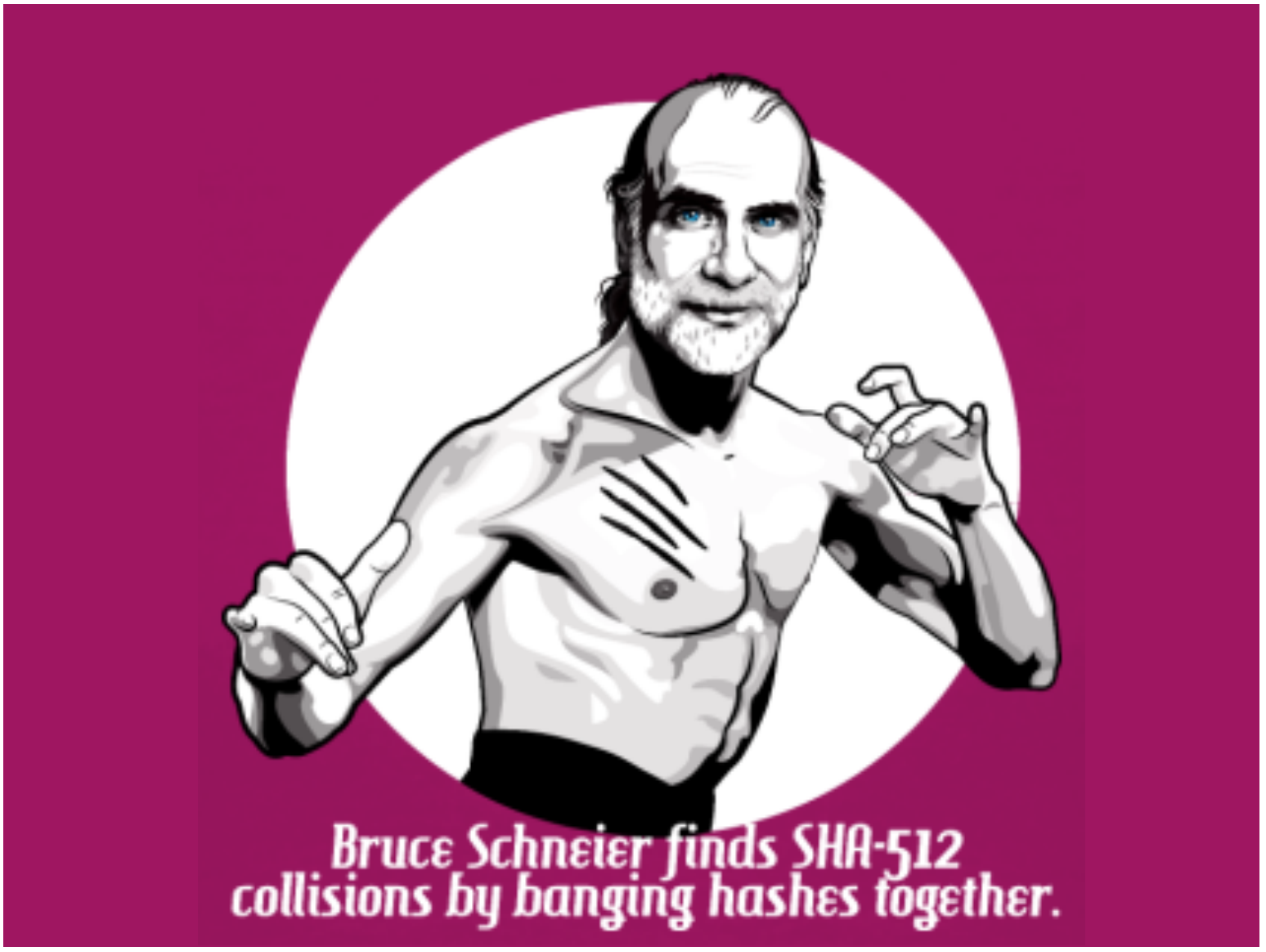
- Vulnerabilities
- Automated Vulnerability Search
- An Approach
- A Solution as Proof of Concept
- Demo ;)
- Whats next?

Intro


Got a bug in your software?
Can I haz it??

Chuck Norris On Security.

Vulnerabilities are **software mistakes** in specification and design, but mostly mistakes **in programming**. Any large software package will have thousands of mistakes. Once discovered, they can be **used to attack systems**. This is the point of security patching: eliminating known vulnerabilities. But many systems don't get patched, so the **Internet is filled with known, exploitable vulnerabilities**.


Bruce Schneier finds SHA-512
collisions by banging hashes together.

How to find vulnerabilities?

- Application Penetration Testing
- Fuzzing
- Reverse Engineering
- Source Code Review
- Or.. Being more advanced:
  - Tracking software bugs, introducing bugs into software, reversing security patches


Who is interested in finding them?
Hackers
Software Companies
Criminals
Governments
Media

How much does a 0-day vulnerability cost?

ADOBE READER	$5,000–$30,000
MAC OSX	$20,000–$50,000
ANDROID	$30,000–$60,000
FLASH OR JAVA BROWSER PLUG-INS	$40,000–$100,000
MICROSOFT WORD	$50,000–$100,000
WINDOWS	$60,000–$120,000
FIREFOX OR SAFARI	$60,000–$150,000
CHROME OR INTERNET EXPLORER	$80,000–$200,000
IOS	$100,000–$250,000

“White Market”

When or why to sell to white market?

“BlackMarket”

Broker?
Money?
Trust?

What happens when you sell to the black market?

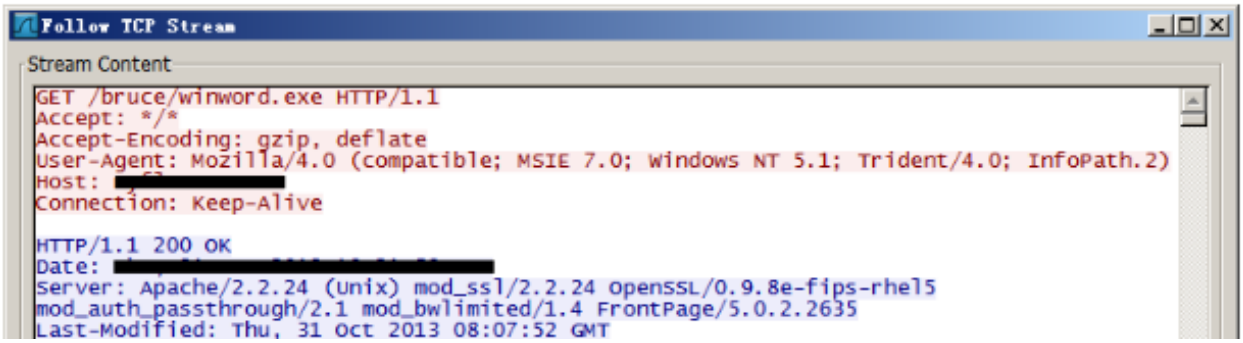
McAfee Labs Detects Zero-Day Exploit Targeting Microsoft Office

By Haifei Li on Nov 05, 2013

Last Thursday morning (October 31), our Advanced Exploit Detection System (AEDS), which we discussed in an earlier post, detected a suspicious sample targeting Microsoft Office. After some investigation, we confirmed this is a zero-day attack.

Considering the importance of this incident, we shared our findings immediately with the Microsoft Security Response Center and worked closely with them in the last couple days. Today, as Microsoft has publicly released the security advisory with mitigations and workarounds, we feel it is time to share some detail of this zero-day attack.

Here is the traffic captured by this attack on a fully updated version of Office 2007 running on Windows XP SP3.



And why automate it?

It‘s faster!!

- The hacker – can break more
- The software company – can fix faster
- Criminals – can make more money
- Governments – can ... [SECRET]
- Media – has more to write about

The Approach

What happens in Windows7 stays in Windows7...

Win7

quartz.dll

```
xor     eax, eax
inc     eax
shl     eax, cl
...
shl     eax, 2
push    eax         ; cb
call
 ds:__imp__CoTaskMemAlloc@4
```

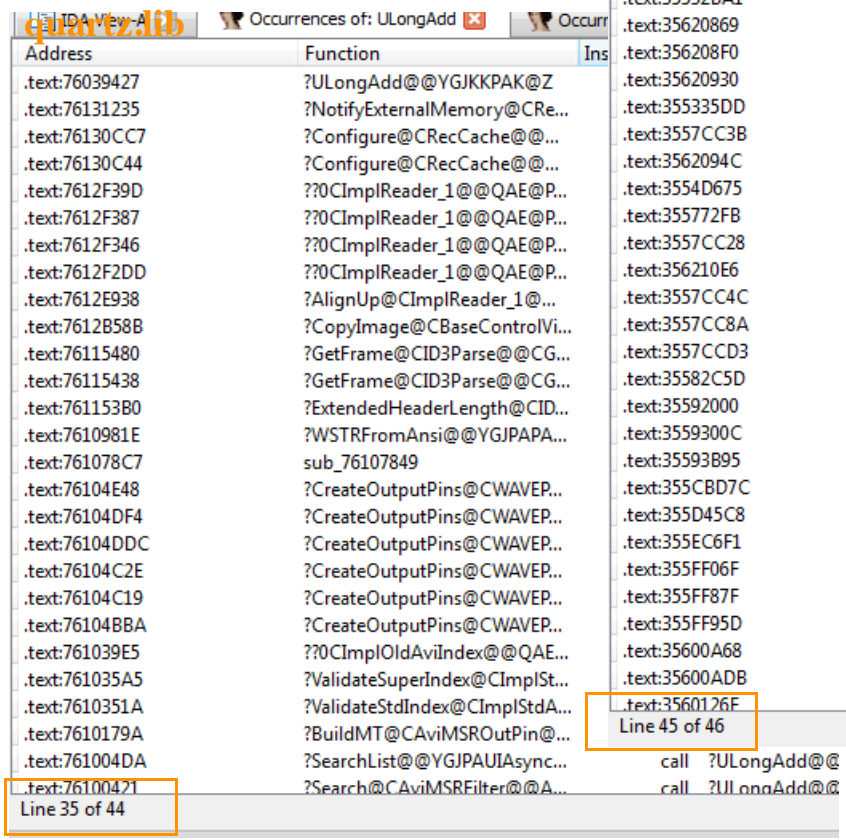
Win8

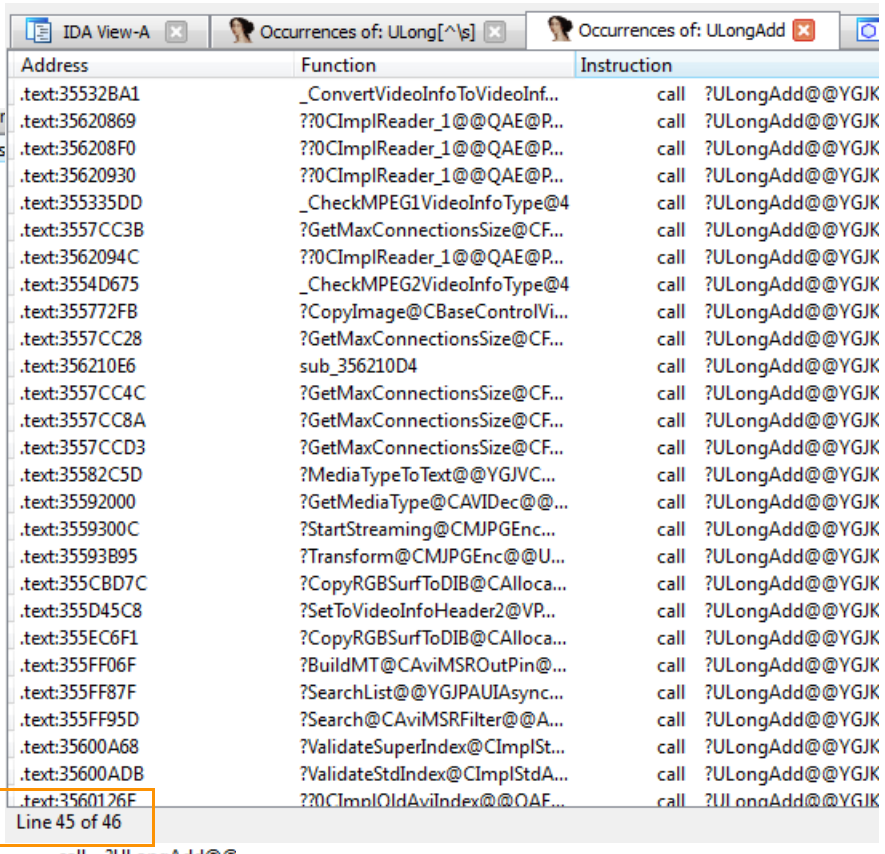
quartz.dll

```
lea     ecx, [ebp+cb]
push    ecx
push    4
push    eax
mov     [esi], eax
call    ?ULongMult@@YGJKKPAK@Z
test    eax, eax
...
push    [ebp+cb]    ; cb
call
 ds:__imp__CoTaskMemAlloc@4
```

Counting Function Calls

Win7
quartz.lib





Win8
quartz.lib

Spot The Patch

Win7
quartz.lib

```
cmp     cx, 8
jbe     short loc_7609B8EC
mov     eax, 80040220h
jmp     short loc_7609B933
-----------------------------------------

                        ; CODE XREF:
xor     eax, eax
inc     eax
shl     eax, cl
push    esi
mov     esi, [ebp+arg_0]
push    edi
mov     [esi], eax
shl     eax, 2
push    eax             ; cb
call    ds:__imp__CoTaskMemAlloc@4 ;
mov     edi, [ebp+arg_4]
mov     [edi], eax
test    eax, eax
jnz     short loc_7609B914
and     [esi], eax
```

Win8
quartz.lib

```
lea     ecx, [ebp+cb]
push    ecx             ; unsigned __int3
push    4               ; int
push    eax             ; int
mov     [esi], eax
call    ?ULongMult@@YGJKKPAK@Z ; ULongMul
test    eax, eax
jns     short loc_355B131C
mov     eax, 80070216h
jmp     short loc_355B1351
-----------------------------------------

                        ; CODE XREF: COve
push    [ebp+cb]        ; cb
call    ds:__imp__CoTaskMemAlloc@4 ; CoTa
mov     [edi], eax
test    eax, eax
jnz     short loc_355B1334
and     [esi], eax
mov     eax, 8007000Eh
jmp     short loc_355B1351
```

Intsafe.h & Strsafe.h

- Searching for security patches:
  - Type Conversion
  - Safe Math Functions
  - Buffer Boundary Checks on Strings

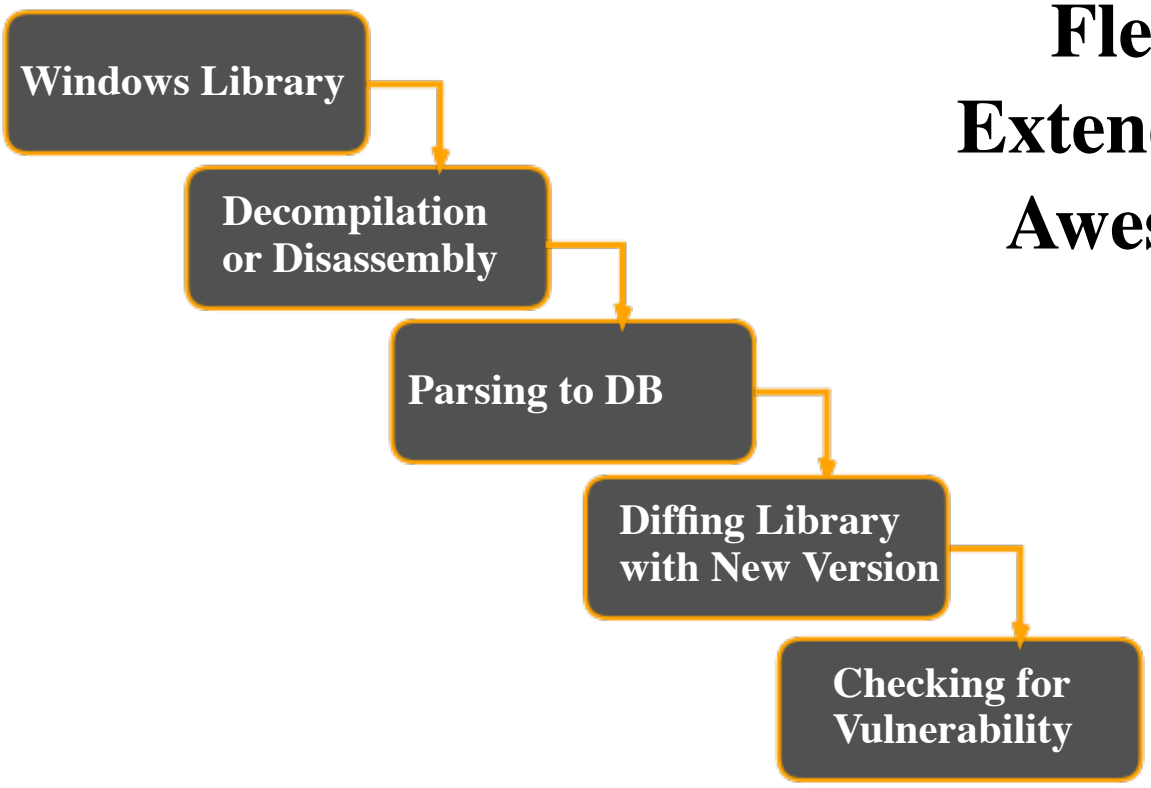
- Set of 130 Signatures of ‚Safe Functions‘

‚Safe Functions‘

UInt8ToInt8	UInt8Add	StringCbGets
UInt8ToChar	UShortAdd	StringCbGetsEx
ByteToInt8	UIntAdd	StringCbLength
ByteToChar	ULongAdd	StringCbPrintf
ShortToInt8	SizeTAdd	StringCbPrintfEx
ShortToUChar	ULongLongAdd	StringCbVPrintf
ShortToChar	UInt8Sub	StringCbVPrintfEx
UShortToUInt8	UShortSub	StringCchCat
UShortToShort	UIntSub	StringCchCatEx
IntToInt8	ULongSub	StringCchCatN
IntToUChar	SizeTSub	StringCchCatNEx
IntToChar	ULongLongSub	StringCchCopy

... and many many more ....

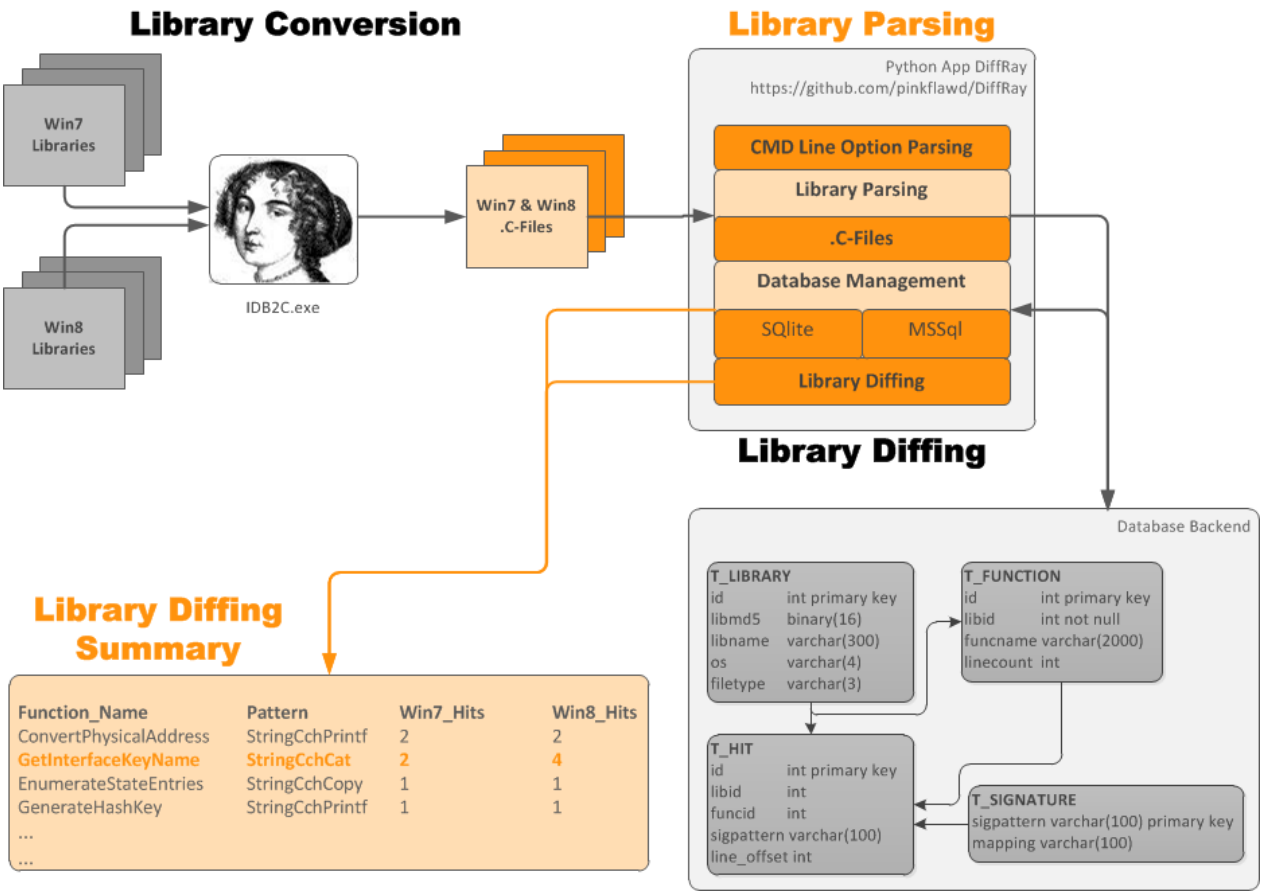
The Approach



Flexible.
Extendible.
Awesome.

The Solution

Pretty, eh??


Library Conversion
Win7 Libraries
Win8 Libraries
IDB2C.exe
Win7 & Win8 .C-Files
Library Parsing
Python App DiffRay
https://github.com/pinkflawd/DiffRay
CMD Line Option Parsing
Library Parsing
.C-Files
Database Management
SQlite
MSSql
Library Diffing
Library Diffing
Database Backend
T_LIBRARY
id int primary key
libmd5 binary(16)
libname varchar(300)
os varchar(4)
filetype varchar(3)
T_FUNCTION
id int primary key
libid int not null
funcname varchar(2000)
linecount int
T_HIT
id int primary key
libid int
funcid int
sigpattern varchar(100)
line_offset int
T_SIGNATURE
sigpattern varchar(100) primary key
mapping varchar(100)
Library Diffing Summary
Function_Name Pattern Win7_Hits Win8_Hits
ConvertPhysicalAddress StringCchPrintf 2 2
GetInterfaceKeyName StringCchCat 2 4
EnumerateStateEntries StringCchCopy 1 1
GenerateHashKey StringCchPrintf 1 1
...
...

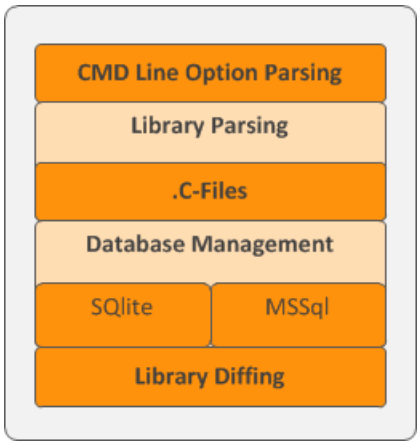
Getting the .C

Library Conversion using IDA Pro

means: .dll -> .idb -> .c

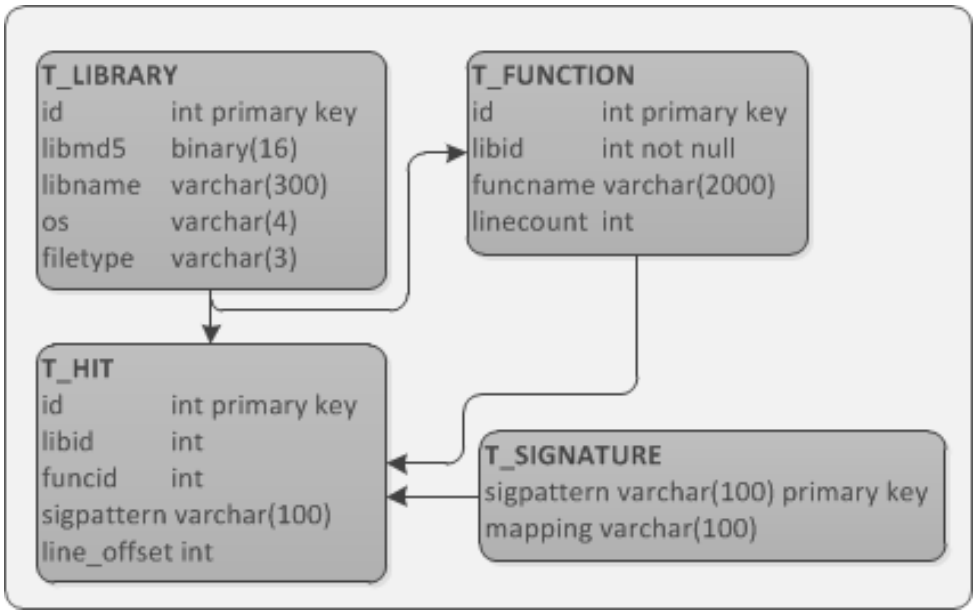
Library Parsing

- DiffRay on *https://github.com/pinkflawd/DiffRay*
- Parses a library / directory of libraries
- Manages libraries , functions and signature hits
- Diff libraries functionwise
  - Based on library ID or library name pattern



The Database

MSSql or SQLite



Diff it!

- Compare libraries on a function basis
- Extract hits per function per signature

Function_Name	Pattern	Win7_Hits	Win8_Hits
ConvertPhysicalAddress	StringCchPrintf	2	2
GetInterfaceKeyName	StringCchCat	2	4
EnumerateStateEntries	StringCchCopy	1	1
GenerateHashKey	StringCchPrintf	1	1
1611-1610-9232-2136-5206			
...			

DiffRay HowTo: Configuration

- **signatures.conf** – whatever symbols you‘re searching for
- **sig_mappings.conf** – mappings for signatures
- **logger.conf** – logging output and formatting, details to be found at *http://docs.python.org/2/howto/logging.html*
- **mssql.conf** – MSSql access credentials

DiffRay HowTo: CMD Parsing

Maintenance:

```
python [dir]\src\Main.py --create-scheme --update-sigs
python [dir]\src\Main.py --parse [library_path]
             --os [Win7|Win8] --type [C|LST]

python [dir]\src\Main.py --dirparse [directory_path]
             --os [Win7|Win8] --type [C|LST]


python [dir]\src\Main.py --flushall
```

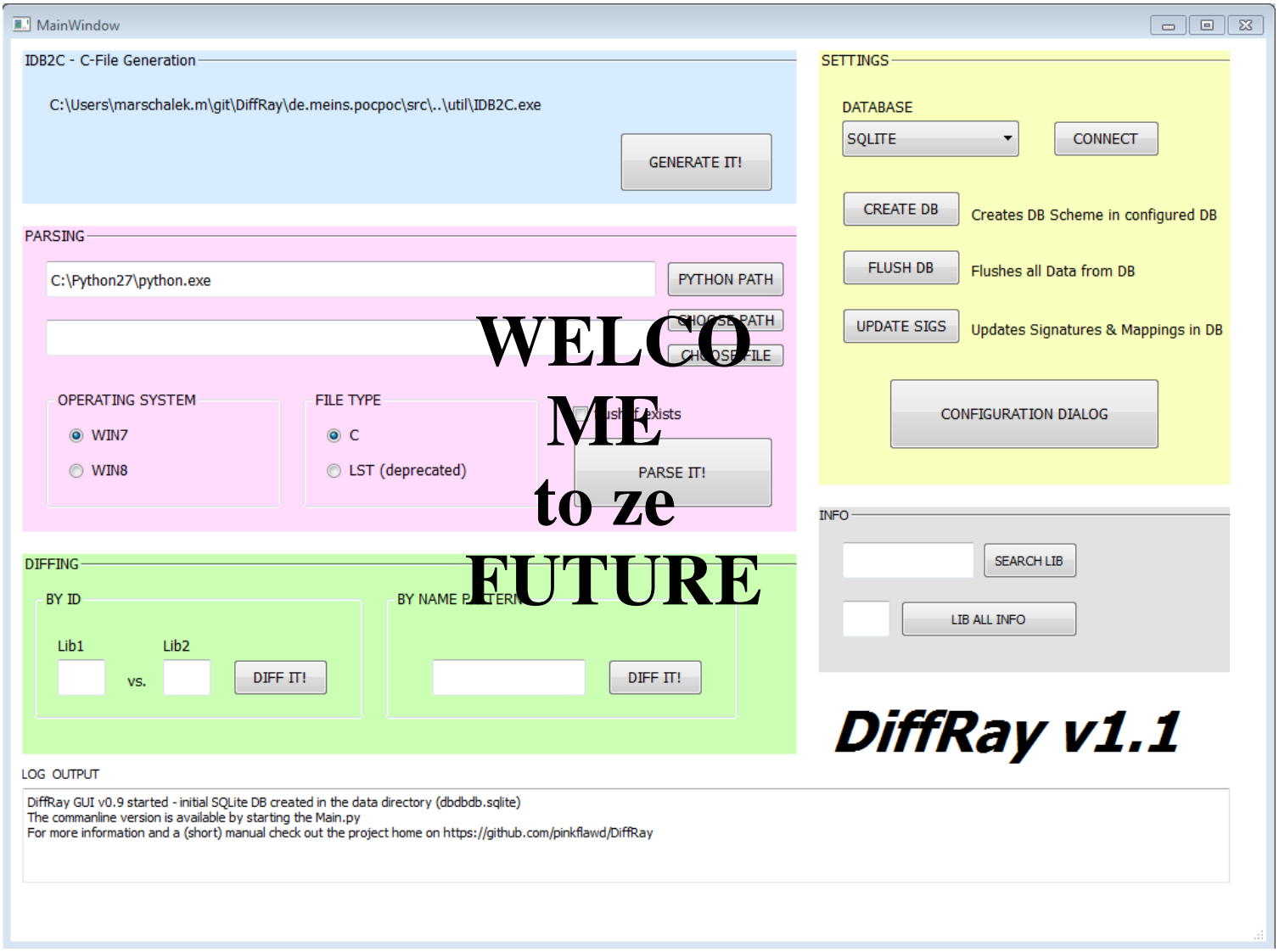
Switches:

```
--backend [mssql|sqlite]
--no-flush
```

DiffRay HowTo: CMD Diffing

Info Output & Diffing:

```
python [dir]\src\Main.py –-search_libs [libname_pattern]

python [dir]\src\Main.py –-lib_all_info [lib_id]

python [dir]\src\Main.py –-diff
              --lib_1 [win7lib] --lib_2 [win8lib]

python [dir]\src\Main.py –-diff_byname [libname_pattern]
```

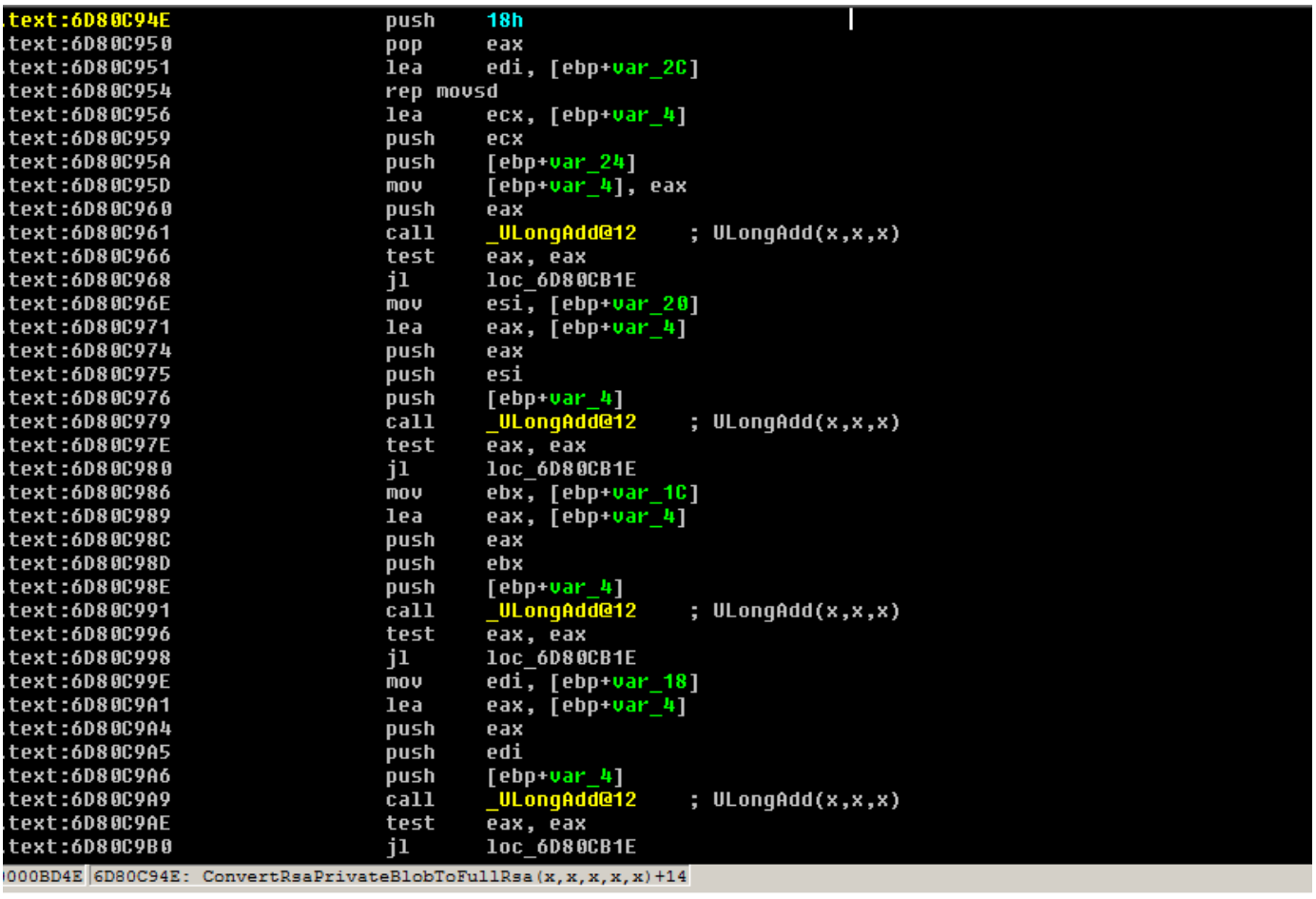


WELCO ME to ze FUTURE

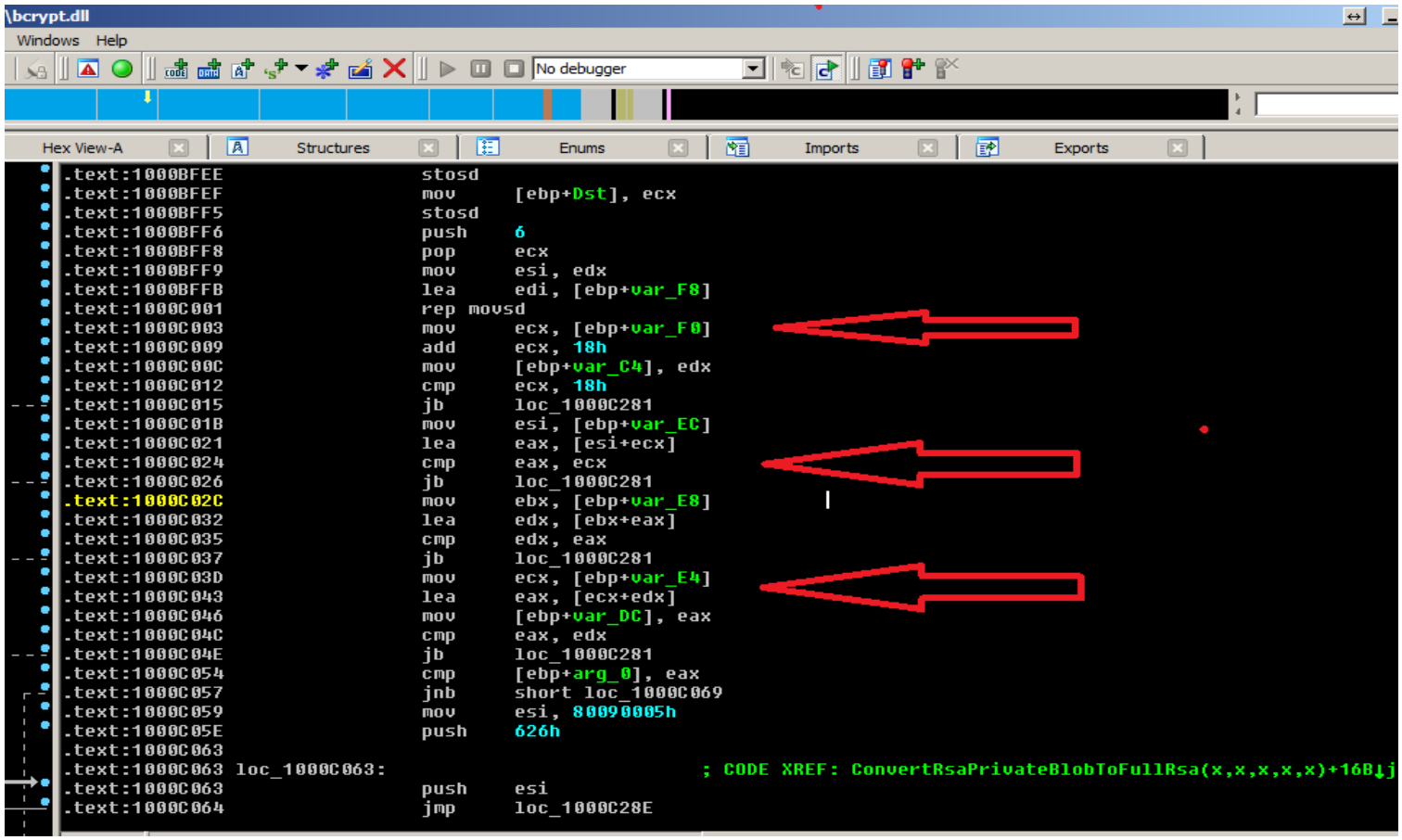
DEMO TIME

Findings

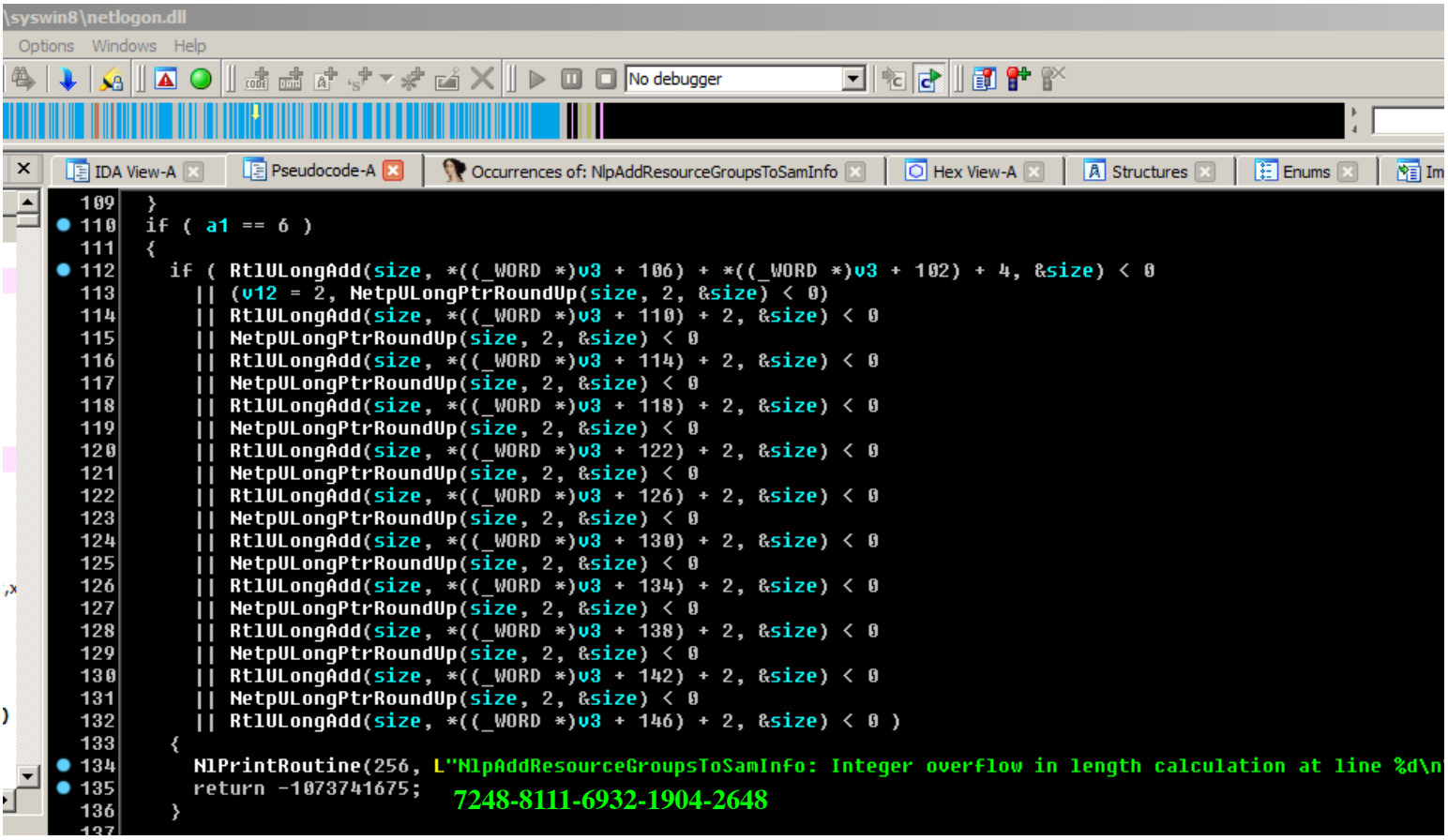
Windows 7 (ULongAdd)
bcrypt.dll!ConvertRsaPrivateBlobToFullRsa



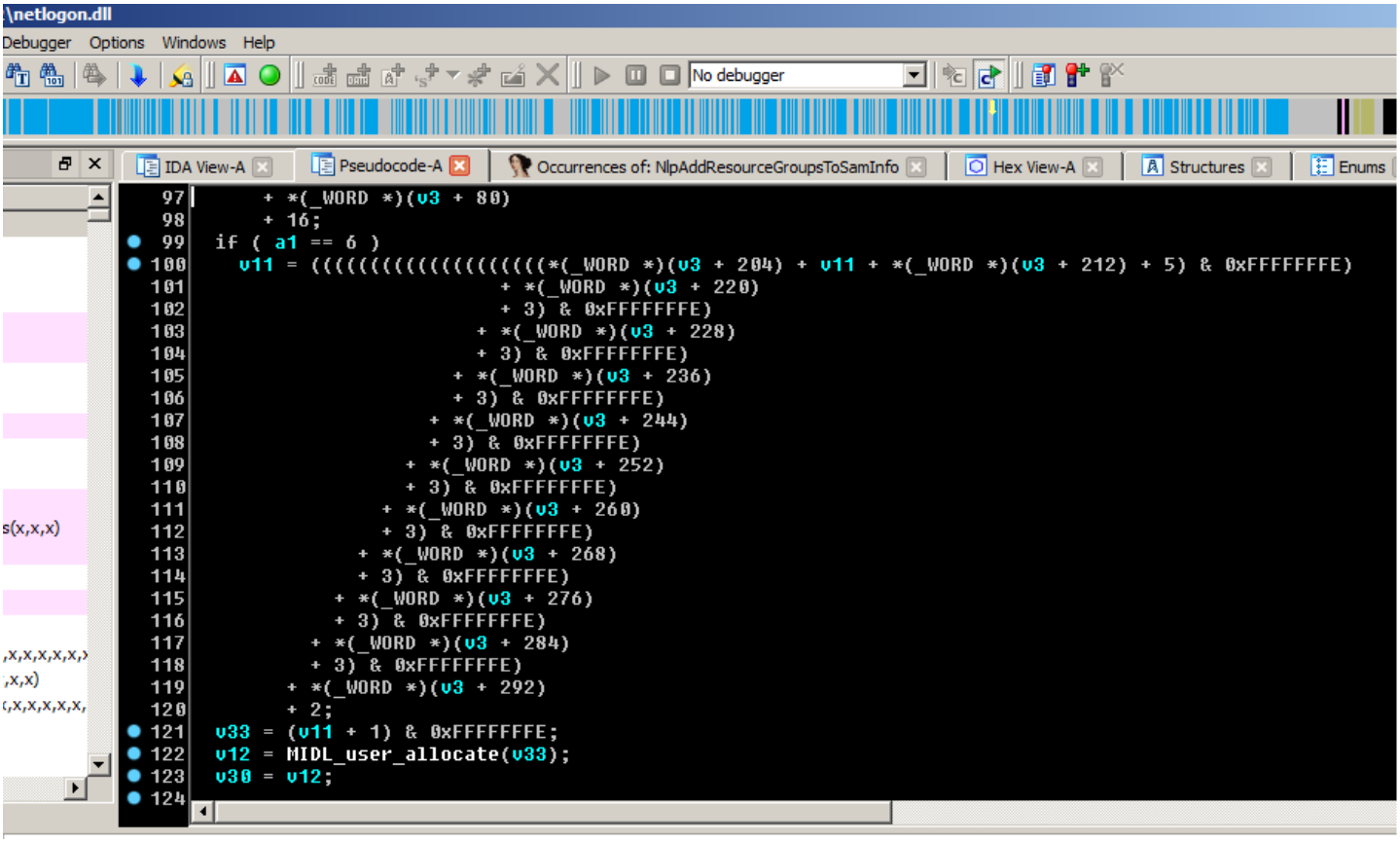
Windows 8
bcrypt.dll!ConvertRsaPrivateBlobToFullRsa



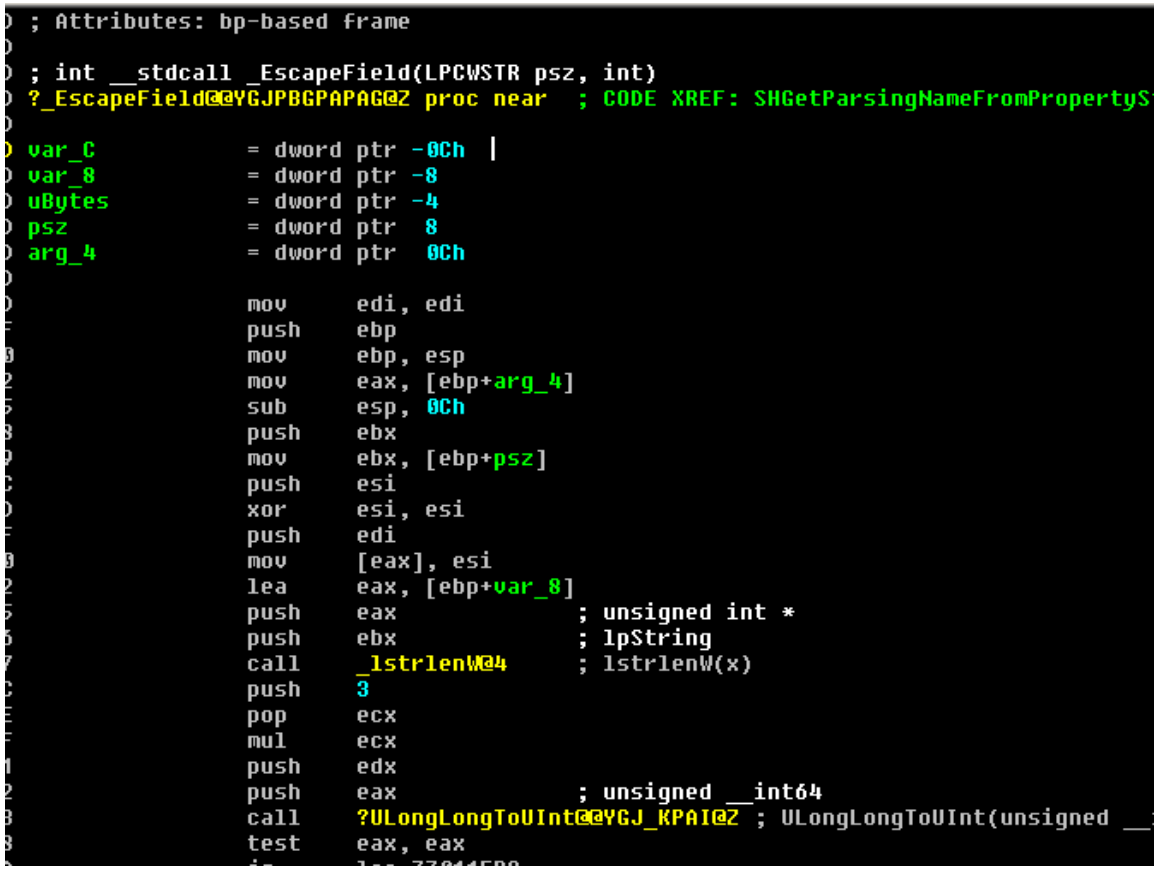
Windows 8 (ULongAdd)
netlogon.dll! NlpAddResourceGroupsToSamInfo



Windows 7
netlogon.dll! NlpAddResourceGroupsToSamInfo

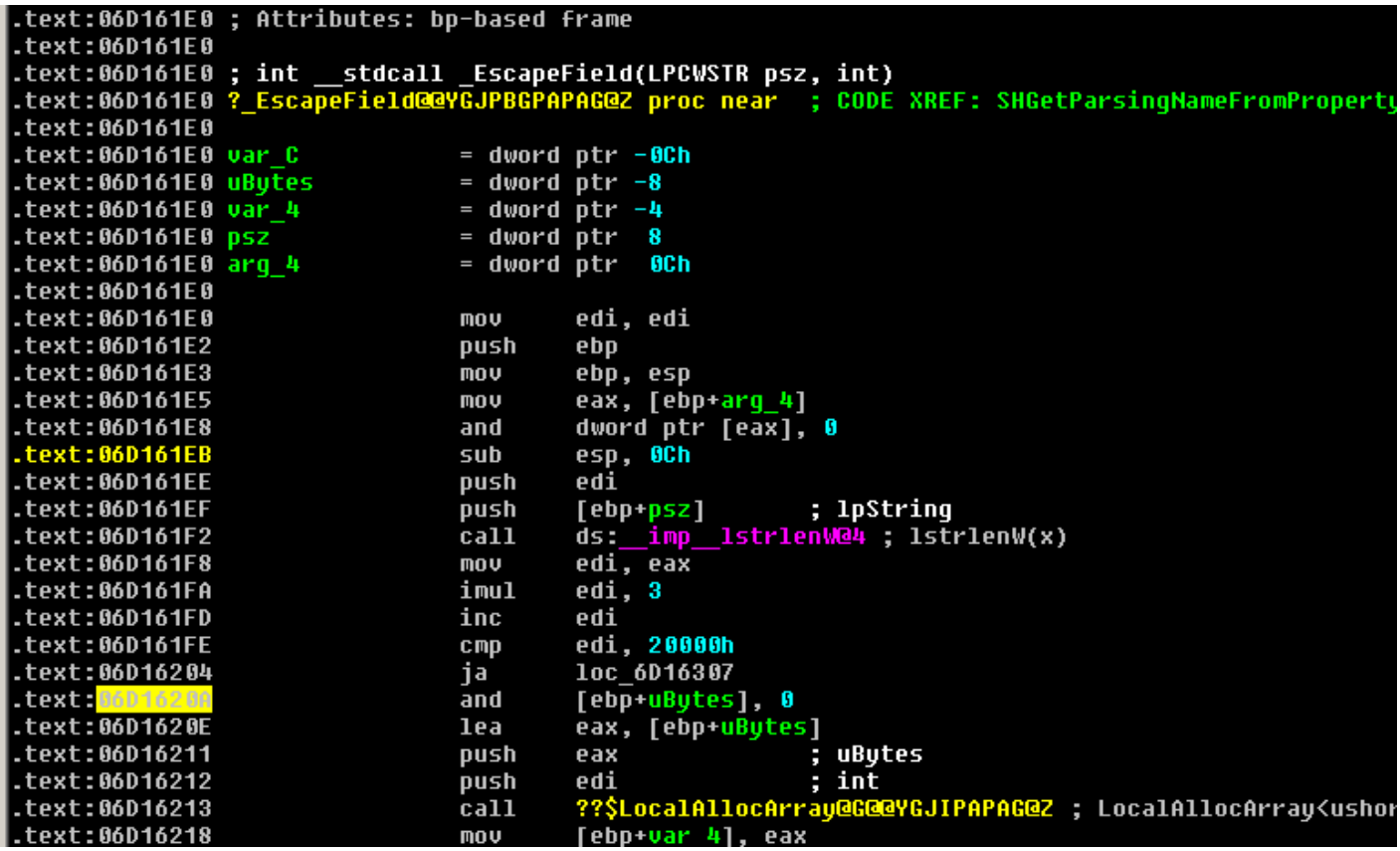


Windows 8 /ULongLongToUint twext.dll! EscapeField



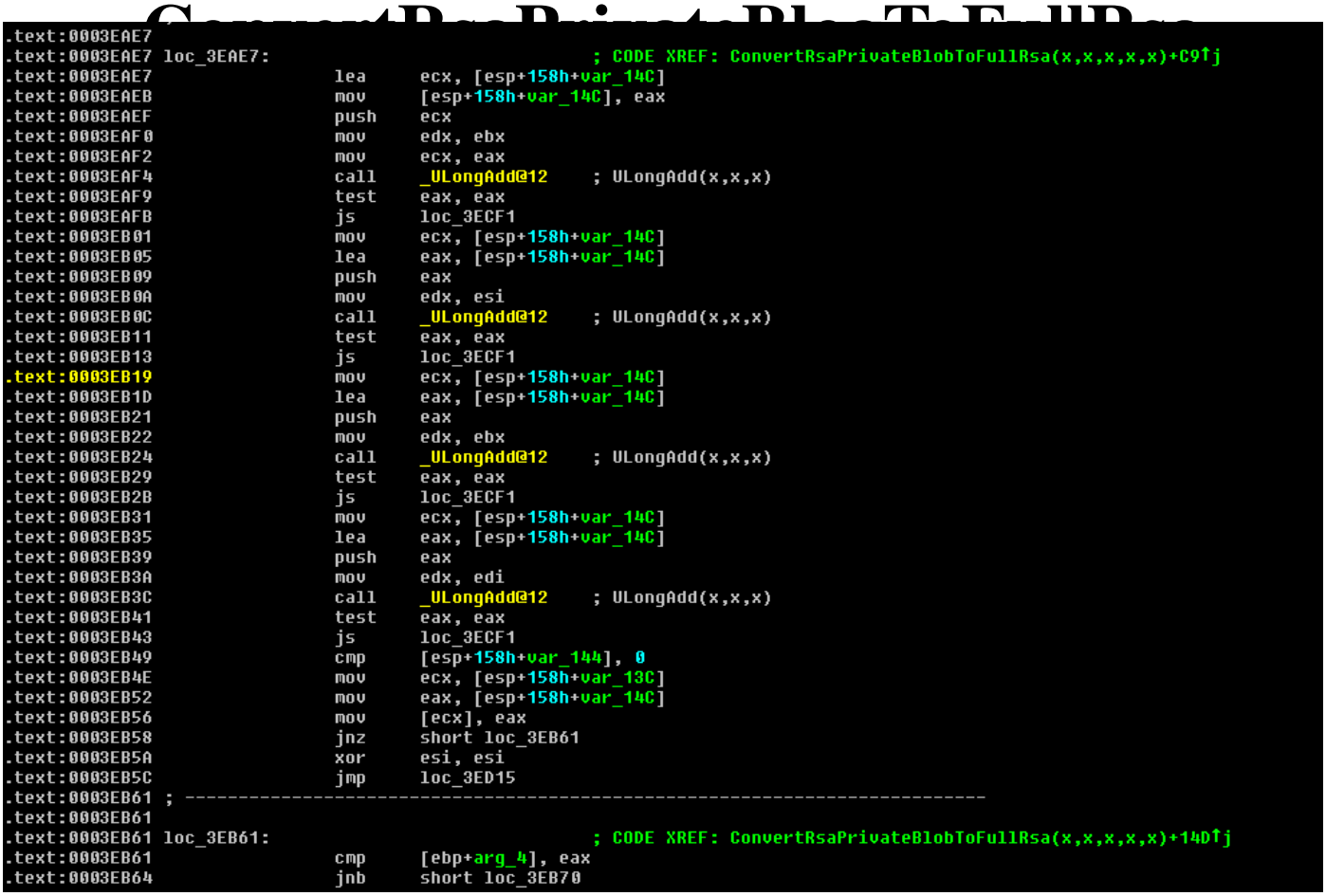
```
; Attributes: bp-based frame

; int __stdcall _EscapeField(LPCWSTR psz, int)
?_EscapeField@@YGJPBGPAPAG@Z proc near  ; CODE XREF: SHGetParsingNameFromPropertyS

var_C           = dword ptr -0Ch
var_8           = dword ptr -8
uBytes          = dword ptr -4
psz             = dword ptr  8
arg_4           = dword ptr  0Ch

                mov     edi, edi
                push    ebp
                mov     ebp, esp
                mov     eax, [ebp+arg_4]
                sub     esp, 0Ch
                push    ebx
                mov     ebx, [ebp+psz]
                push    esi
                xor     esi, esi
                push    edi
                mov     [eax], esi
                lea     eax, [ebp+var_8]
                push    eax             ; unsigned int *
                push    ebx             ; lpString
                call    _lstrlenW@4     ; lstrlenW(x)
                push    3
                pop     ecx
                mul     ecx
                push    edx
                push    eax             ; unsigned __int64
                call    ?ULongLongToUInt@@YGJ_KPAI@Z ; ULongLongToUInt(unsigned __
                test    eax, eax
```

Windows 7 Integer overflow twext.dll! EscapeField



Drrrivers...

Windows 8
cng.dll!
ConvertRsaPrivateBlobToFullRsa



```
.text:0003EAE7
.text:0003EAE7 loc_3EAE7:                              ; CODE XREF: ConvertRsaPrivateBlobToFullRsa(x,x,x,x,x)+C9↑j
.text:0003EAE7                 lea     ecx, [esp+158h+var_14C]
.text:0003EAEB                 mov     [esp+158h+var_14C], eax
.text:0003EAEF                 push    ecx
.text:0003EAF0                 mov     edx, ebx
.text:0003EAF2                 mov     ecx, eax
.text:0003EAF4                 call    _ULongAdd@12    ; ULongAdd(x,x,x)
.text:0003EAF9                 test    eax, eax
.text:0003EAFB                 js      loc_3ECF1
.text:0003EB01                 mov     ecx, [esp+158h+var_14C]
.text:0003EB05                 lea     eax, [esp+158h+var_14C]
.text:0003EB09                 push    eax
.text:0003EB0A                 mov     edx, esi
.text:0003EB0C                 call    _ULongAdd@12    ; ULongAdd(x,x,x)
.text:0003EB11                 test    eax, eax
.text:0003EB13                 js      loc_3ECF1
.text:0003EB19                 mov     ecx, [esp+158h+var_14C]
.text:0003EB1D                 lea     eax, [esp+158h+var_14C]
.text:0003EB21                 push    eax
.text:0003EB22                 mov     edx, ebx
.text:0003EB24                 call    _ULongAdd@12    ; ULongAdd(x,x,x)
.text:0003EB29                 test    eax, eax
.text:0003EB2B                 js      loc_3ECF1
.text:0003EB31                 mov     ecx, [esp+158h+var_14C]
.text:0003EB35                 lea     eax, [esp+158h+var_14C]
.text:0003EB39                 push    eax
.text:0003EB3A                 mov     edx, edi
.text:0003EB3C                 call    _ULongAdd@12    ; ULongAdd(x,x,x)
.text:0003EB41                 test    eax, eax
.text:0003EB43                 js      loc_3ECF1
.text:0003EB49                 cmp     [esp+158h+var_144], 0
.text:0003EB4E                 mov     ecx, [esp+158h+var_13C]
.text:0003EB52                 mov     eax, [esp+158h+var_14C]
.text:0003EB56                 mov     [ecx], eax
.text:0003EB58                 jnz     short loc_3EB61
.text:0003EB5A                 xor     esi, esi
.text:0003EB5C                 jmp     loc_3ED15
.text:0003EB61 ; ---------------------------------------------------------------------------
.text:0003EB61
.text:0003EB61 loc_3EB61:                              ; CODE XREF: ConvertRsaPrivateBlobToFullRsa(x,x,x,x,x)+14D↑j
.text:0003EB61                 cmp     [ebp+arg_4], eax
.text:0003EB64                 jnb     short loc_3EB70
```

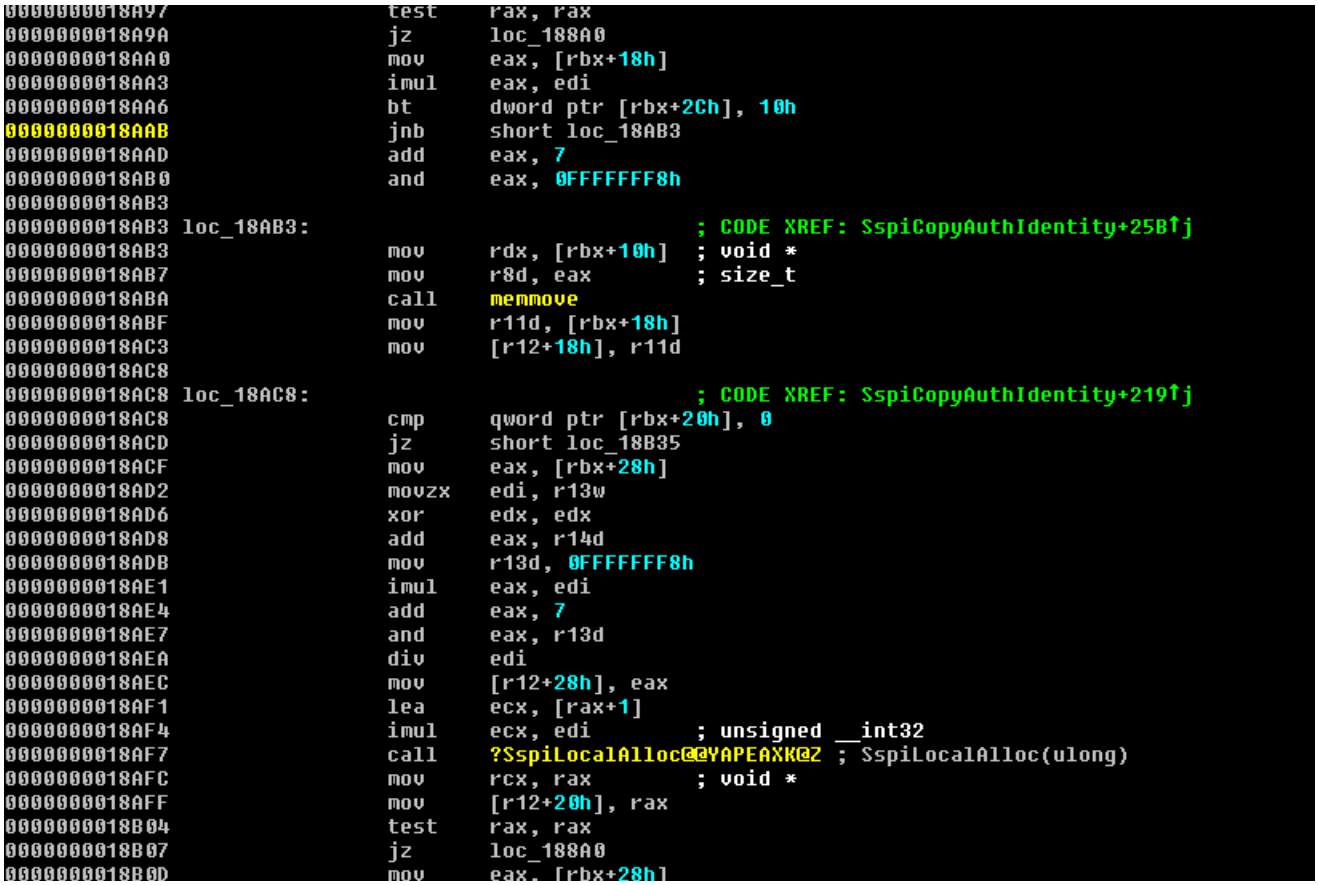
Windows 7
cng.dll!
ConvertRsaPrivateBlobToFullRsa

```
text:000000000035ADE                add     ecx, r12d
text:000000000035AE1                mov     [rax], ecx
text:000000000035AE3                test    r14, r14
text:000000000035AE6                jnz     short loc_35AEF
text:000000000035AE8                xor     ebx, ebx
text:000000000035AEA                jmp     loc_35CC1
text:000000000035AEF ; ---------------------------------------------------------------------------
text:000000000035AEF
text:000000000035AEF loc_35AEF:                             ; CODE XREF: ConvertRsaPrivateBlobToFullRsa+106↑j
text:000000000035AEF                cmp     dword ptr [rsp+218h+arg_18], ecx
text:000000000035AF6                jnb     short loc_35B02
text:000000000035AF8                mov     ebx, 80090028h
text:000000000035AFD                jmp     loc_35CC1
text:000000000035B02 ; ---------------------------------------------------------------------------
text:000000000035B02
text:000000000035B02 loc_35B02:                             ; CODE XREF: ConvertRsaPrivateBlobToFullRsa+116↑j
text:000000000035B02                lea     rax, [rbx+18h]
text:000000000035B06                xor     edx, edx        ; int
text:000000000035B08                lea     rcx, [rsp+218h+var_1B8] ; void *
text:000000000035B0D                lea     rbx, [rax+r15]
text:000000000035B11                lea     r8d, [rdx+20h]  ; size_t
text:000000000035B15                mov     [rsp+218h+arg_0], rax
text:000000000035B1D                lea     r13, [rbx+rbp]
text:000000000035B21                call    memset
text:000000000035B26                mov     rcx, [rsp+218h+arg_0]
text:000000000035B2E                lea     r11, [rsp+218h+var_1B8]
text:000000000035B33                lea     rax, [rsp+218h+var_178]
text:000000000035B3B                mov     r9d, ebp
text:000000000035B3E                mov     [rsp+218h+var_1C8], r11
text:000000000035B43                mov     [rsp+218h+var_1D0], 1
text:000000000035B4B                mov     [rsp+218h+var_1D8], rax
text:000000000035B50                mov     [rsp+218h+var_1E0], edi
text:000000000035B54                lea     rax, [rsi+r13]
text:000000000035B58                mov     r8, rbx
text:000000000035B5B                mov     [rsp+218h+var_1E8], rax
text:000000000035B60                mov     edx, r15d
text:000000000035B63                mov     dword ptr [rsp+218h+var_1F0], esi
text:000000000035B67                mov     [rsp+218h+var_1F8], r13
```

Windows 8
ksecdd.dll! SspiCopyAuthIdentity

```
text:00018BF9 ; ---------------------------------------------------------------------------
text:00018BF9
text:00018BF9 loc_18BF9:                              ; CODE XREF: SspiCopyAuthIdentity(x,x)+3C↑j
text:00018BF9                 movzx   edx, word ptr [ebx+10h]
text:00018BFD                 lea     eax, [ebp+arg_0]
text:00018C00                 push    40h
text:00018C02                 pop     ecx
text:00018C03                 push    eax
text:00018C04                 mov     [ebp+arg_0], ecx
text:00018C07                 call    _RtlULongAdd@12 ; RtlULongAdd(x,x,x)
text:00018C0C                 test    eax, eax
text:00018C0E                 js      loc_18B83
text:00018C14                 movzx   edx, word ptr [ebx+18h]
text:00018C18                 lea     eax, [ebp+arg_0]
text:00018C1B                 mov     ecx, [ebp+arg_0]
text:00018C1E                 push    eax
text:00018C1F                 call    _RtlULongAdd@12 ; RtlULongAdd(x,x,x)
text:00018C24                 test    eax, eax
text:00018C26                 js      loc_18B83
text:00018C2C                 movzx   edx, word ptr [ebx+2Ch]
text:00018C30                 lea     eax, [ebp+arg_0]
text:00018C33                 mov     ecx, [ebp+arg_0]
text:00018C36                 push    eax
text:00018C37                 call    _RtlULongAdd@12 ; RtlULongAdd(x,x,x)
text:00018C3C                 test    eax, eax
text:00018C3E                 js      loc_18B83
text:00018C44                 movzx   edx, word ptr [ebx+20h]
text:00018C48                 lea     eax, [ebp+arg_0]
text:00018C4B                 mov     ecx, [ebp+arg_0]
text:00018C4E                 push    eax
text:00018C4F                 call    _RtlULongAdd@12 ; RtlULongAdd(x,x,x)
text:00018C54                 test    eax, eax
text:00018C56                 js      loc_18B83
text:00018C5C                 mov     esi, [ebp+arg_0]
text:00018C5F                 cmp     esi, 0FFFFh
text:00018C65                 ja      loc_18B83
text:00018C6B                 mov     ecx, esi
text:00018C6D                 call    ?SspiLocalAlloc@@YGPAXK@Z ; SspiLocalAlloc(ulong)
```

Windows 7
ksecdd.dll! SspiCopyAuthIdentity

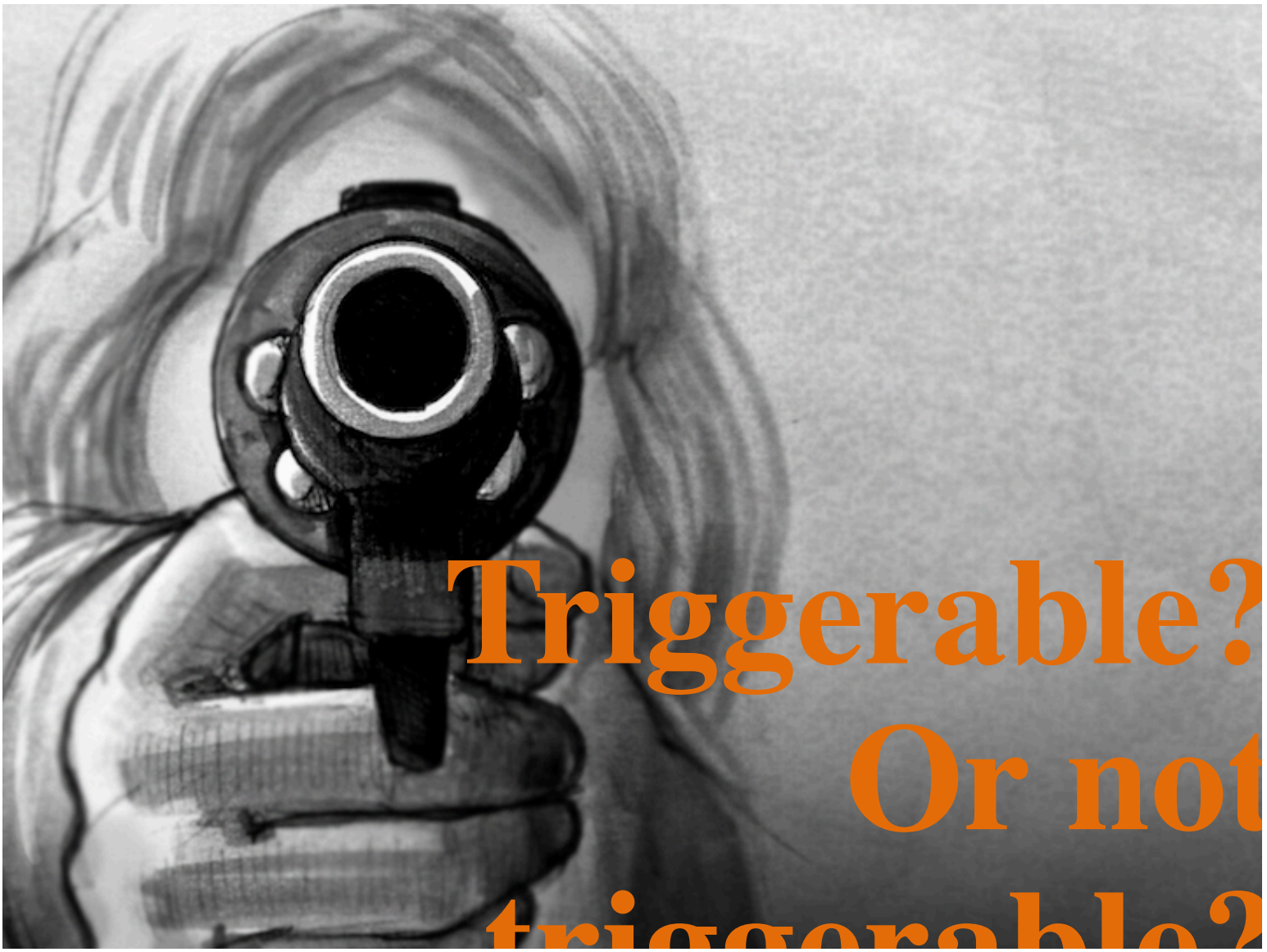


Windows 8
srvnet.dll!
SrvNetAllocatePoolWithTagPriority

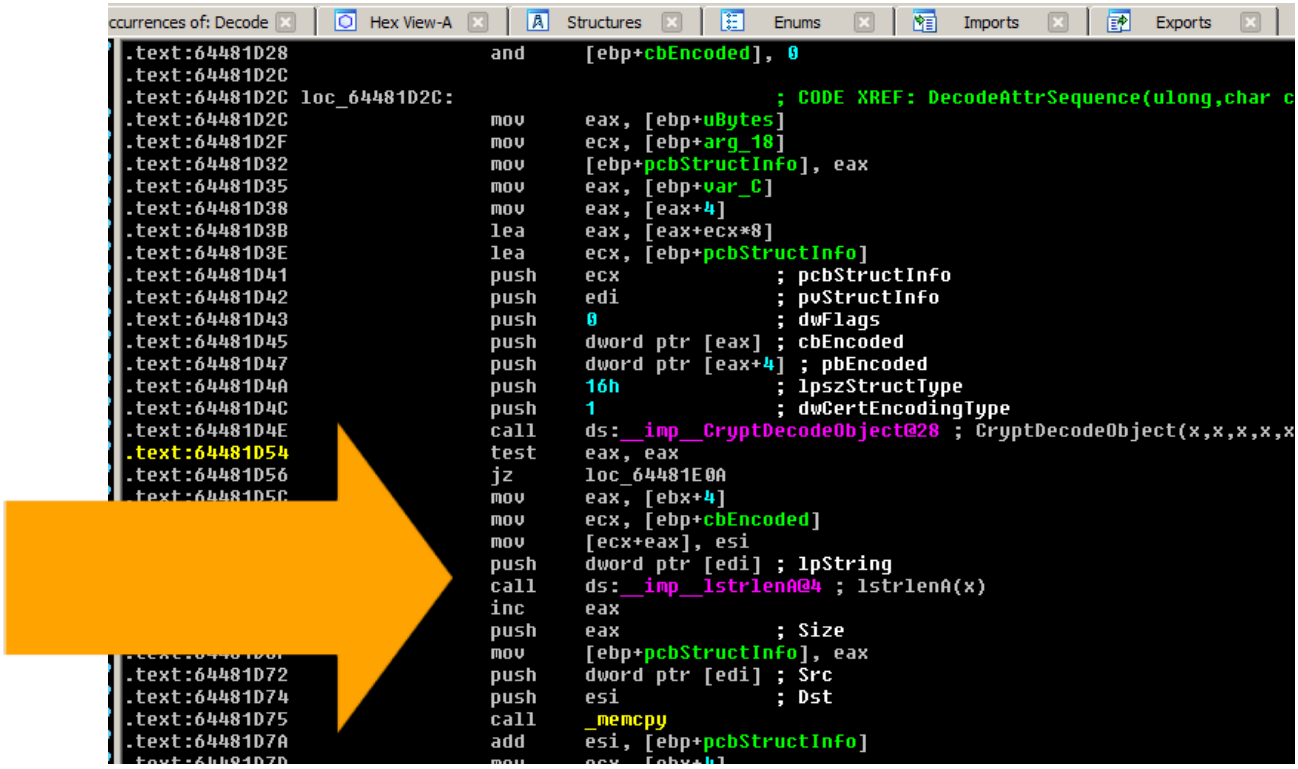
```
text:0001A753 ; Attributes: bp-based frame
text:0001A753
text:0001A753 ; int __stdcall SrvNetAllocatePoolWithTagPriority(POOL_TYPE PoolType, int, ULONG Tag, EX_POOL_PRIORITY Priority)
text:0001A753                 public _SrvNetAllocatePoolWithTagPriority@16
text:0001A753 _SrvNetAllocatePoolWithTagPriority@16 proc near
text:0001A753
text:0001A753 NumberOfBytes   = dword ptr -4
text:0001A753 PoolType        = dword ptr  8
text:0001A753 arg_4           = dword ptr  0Ch
text:0001A753 Tag             = dword ptr  10h
text:0001A753 Priority        = dword ptr  14h
text:0001A753
text:0001A753                 mov     edi, edi
text:0001A755                 push    ebp
text:0001A756                 mov     ebp, esp
text:0001A758                 push    ecx
text:0001A759                 mov     edx, [ebp+arg_4]
text:0001A75C                 lea     eax, [ebp+NumberOfBytes]
text:0001A75F                 push    eax
text:0001A760                 push    16
text:0001A762                 pop     ecx
text:0001A763                 call    _RtlSizeTAdd@12 ; RtlSizeTAdd(x,x,x)
text:0001A768                 test    eax, eax
text:0001A76A                 jns     short loc_1A778
```

Windows 7
srvnet.dll!
SrvNetAllocatePoolWithTagPriority

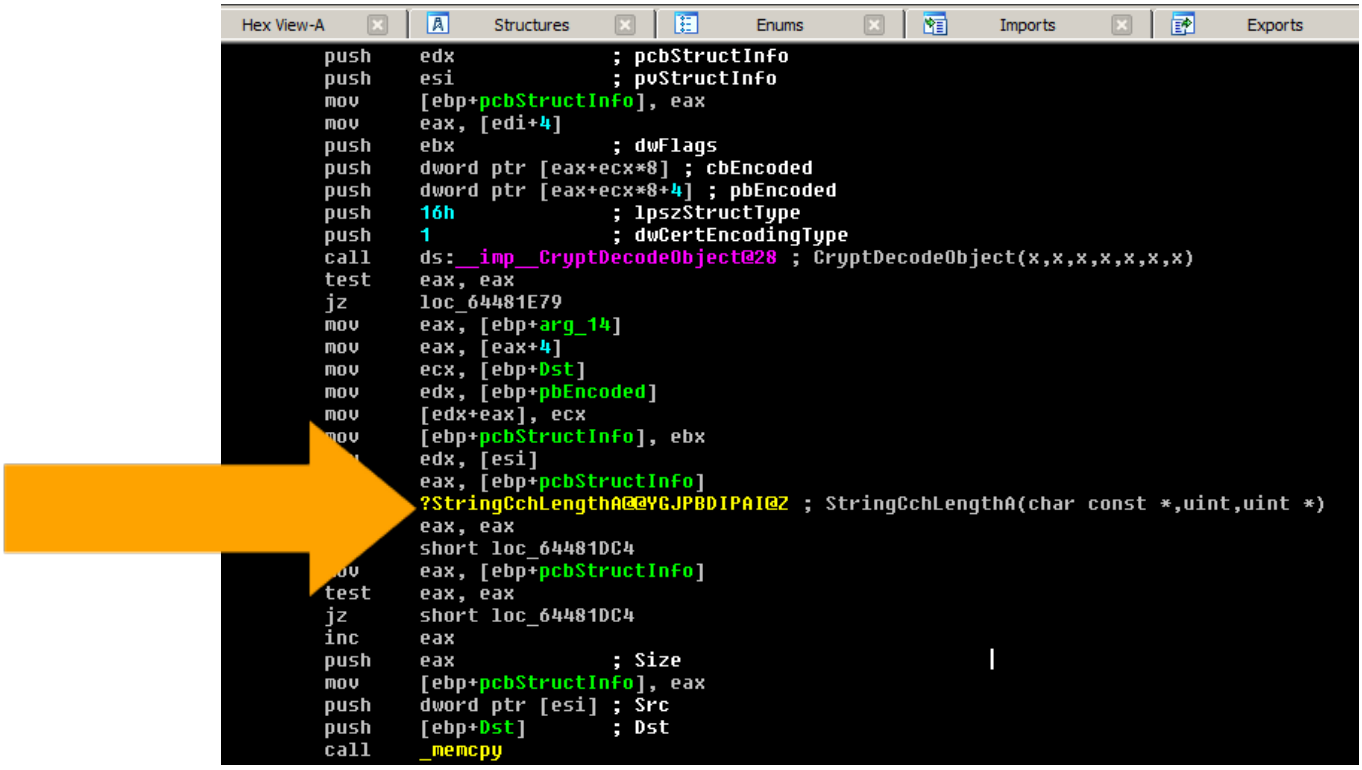
```
.text:000000000001F8F0
.text:000000000001F8F0
.text:000000000001F8F0                  public SrvNetAllocatePoolWithTagPriority
.text:000000000001F8F0 SrvNetAllocatePoolWithTagPriority proc near
.text:000000000001F8F0                                          ; CODE XREF: SrvLibLookasideCreatePool+4E↓p
.text:000000000001F8F0
.text:000000000001F8F0 arg_0           = qword ptr  8
.text:000000000001F8F0
.text:000000000001F8F0                  mov     [rsp+arg_0], rbx
.text:000000000001F8F5                  push    rdi
.text:000000000001F8F6                  sub     rsp, 20h
.text:000000000001F8FA                  lea     edi, [rdx+10h]
.text:000000000001F8FD                  mov     ebx, ecx
.text:000000000001F8FF                  mov     edx, edi        ; NumberOfBytes
.text:000000000001F901                  call    cs:__imp_ExAllocatePoolWithTagPriority
.text:000000000001F907                  mov     r11, rax
.text:000000000001F90A                  test    rax, rax
.text:000000000001F90D                  jz      short loc_1F96A
.text:000000000001F90F                  mov     [rax], edi
.text:000000000001F911                  mov     [rax+4], ebx
.text:000000000001F914                  test    ebx, ebx
.text:000000000001F916                  jz      short loc_1F945
.text:000000000001F918                  cmp     ebx, 4
.text:000000000001F91B                  jz      short loc_1F945
.text:000000000001F91D                  cmp     ebx, 1
.text:000000000001F920                  jz      short loc_1F927
```


Triggerable?
Or not
triggerable?

Windows 7 cryptdlg!DecodeAttrSequence



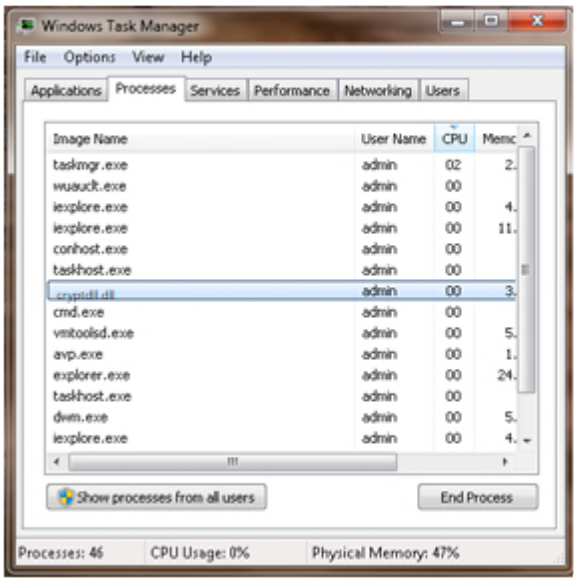
Windows 8 cryptdlg!DecodeAttrSequence



What's CryptDll.dll??

What is cryptdll.dll doing on my computer?

cryptdll.dll is a module associated with Cryptography Manager from Microsoft Corporation. This file is part of Microsoft® Windows® Operating System Non-system processes like cryptdll.dll originate from software you installed on your system. Since most applications store data in your system's registry, it is likely that over time your registry suffers fragmentation and accumulates invalid entries which can affect your PC's performance. It is recommended that you check your registry to identify slowdown issues.



StringCchLength

Determines whether a string exceeds the specified length, in characters.

StringCchLength is a replacement for the following functions:

- strlen, wcslen, _tcslen

Syntax

C++ Copy

```
HRESULT StringCchLength(
  _In_   LPCTSTR psz,
  _In_   size_t cchMax,
  _Out_  size_t *pcch
);
```

Parameters

psz [in]
Type: **LPCTSTR**

The string whose length is to be checked.

cchMax [in]
Type: **size_t**

The maximum number of characters allowed in *psz*, including the terminating null character. This value cannot exceed **STRSAFE_MAX_CCH**.

CryptDecodeObject API

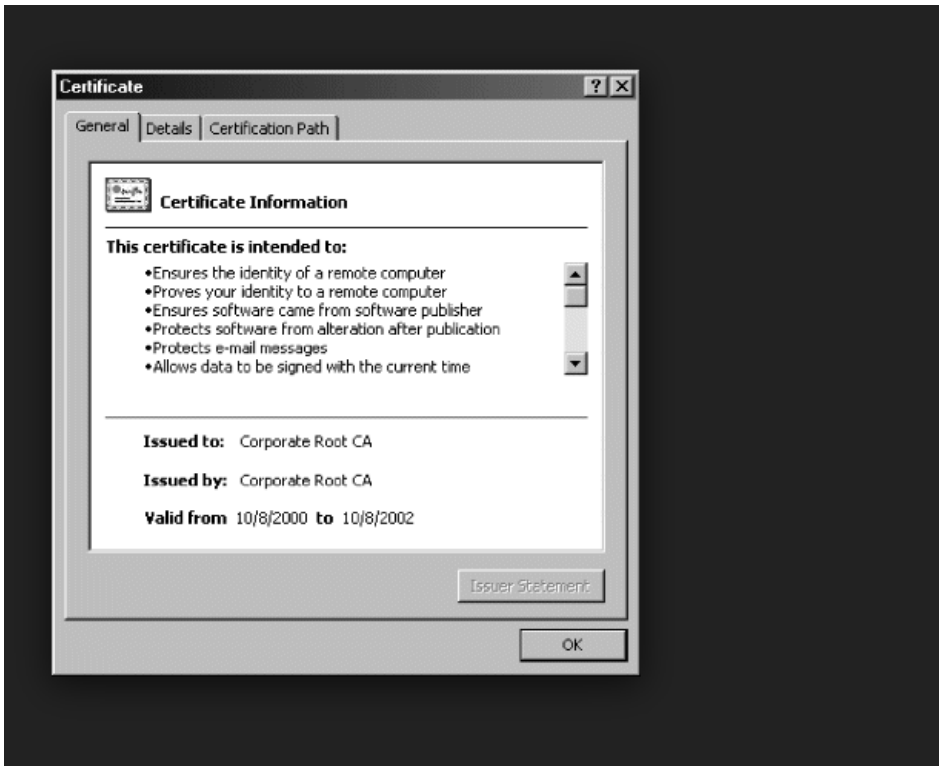
```
BOOL WINAPI CryptDecodeObject(
  _In_     DWORD dwCertEncodingType,
  _In_     LPCSTR lpszStructType,
  _In_     const BYTE *pbEncoded,
  _In_     DWORD cbEncoded,
  _In_     DWORD dwFlags,
  _Out_    void *pvStructInfo,
  _Inout_  DWORD *pcbStructInfo
);
```

pcbStructInfo [in, out]

A pointer to a **DWORD** value specifying the size, in bytes, of the buffer pointed to by the *pvStructInfo* parameter. When the function returns, this **DWORD** value contains the size of the decoded data copied to *pvStructInfo*. The size contained in the variable pointed to by *pcbStructInfo* can indicate a size larger than the decoded structure, as the decoded structure can include pointers to other structures. This size is the sum of the size needed by the decoded structure and other structures pointed to.

Note When processing the data returned in the buffer, applications must use the actual size of the data returned. The actual size can be slightly smaller than the size of the buffer specified on input. (On input, buffer sizes are usually specified large enough to ensure that the largest possible output data fits in the buffer.) On output, the variable pointed to by this parameter is updated to reflect the actual size of the data copied to the buffer.

Certificate DialogBox



What‘s Next

Whats Next

- Possible Extensions
  - Win8, we‘re coming!!
  - Extended signatures
  - Symbolic Execution FTW
- Improvements
  - Transparent DB library
- Known issues
  - Duplicate hits, false positives, slooow, output is not handy

Happy Diffing.

"Most of all I love your vulnerability."