

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



8
7
3
1-
7
3
6
4
1
9
3
2-
9
6
4
6-
3
0
4
0

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?



ZERO DAY
INITIATIVE



“BlackMarket”

Broker?
Money?
Trust?



What happens when you sell to the black market?

TROOPERS 2014



McAfee Labs Detects Zero-Day Exploit Targeting Microsoft Office 0

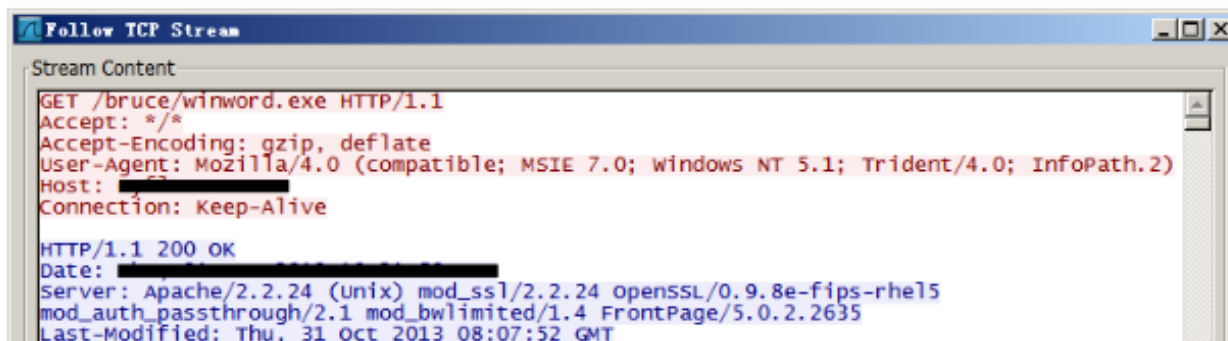
By Haifei Li on Nov 05, 2013

 Like 75  Share 36  +1 2  Tweet 76

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.



```
Follow TCP Stream
Stream Content
GET /bruce/winword.exe HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
Host: ██████████
Connection: Keep-Alive

HTTP/1.1 200 OK
Date: ██████████
Server: Apache/2.2.24 (unix) mod_ssl/2.2.24 OpenSSL/0.9.8e-fips-rhel5
mod_auth_passthrough/2.1 mod_bwlimited/1.4 FrontPage/5.0.2.2635
Last-Modified: Thu, 31 Oct 2013 08:07:52 GMT
```

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
```

Patch it!

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

Occurrences of: ULongAdd

Address	Function	Ins
.text:76039427	?ULongAdd@@YGJKPAK@Z	
.text:76131235	?NotifyExternalMemory@CRE...	
.text:76130CC7	?Configure@CRecCache@@@...	
.text:76130C44	?Configure@CRecCache@@@...	
.text:7612F39D	??0CImplReader_1@@@QAE@P...	
.text:7612F387	??0CImplReader_1@@@QAE@P...	
.text:7612F346	??0CImplReader_1@@@QAE@P...	
.text:7612F2DD	??0CImplReader_1@@@QAE@P...	
.text:7612E938	?AlignUp@CImplReader_1@...	
.text:7612B58B	?CopyImage@CBaseControlVi...	
.text:76115480	?GetFrame@CID3Parse@@@CG...	
.text:76115438	?GetFrame@CID3Parse@@@CG...	
.text:761153B0	?ExtendedHeaderLength@CID...	
.text:7610981E	?WSTRFromAnsi@@@YGJPAPA...	
.text:761078C7	sub_76107849	
.text:76104E48	?CreateOutputPins@CWAVEP...	
.text:76104DF4	?CreateOutputPins@CWAVEP...	
.text:76104DDC	?CreateOutputPins@CWAVEP...	
.text:76104C2E	?CreateOutputPins@CWAVEP...	
.text:76104C19	?CreateOutputPins@CWAVEP...	
.text:76104BBA	?CreateOutputPins@CWAVEP...	
.text:761039E5	??0CImplOldAviIndex@@@QAE...	
.text:761035A5	?ValidateSuperIndex@CImplSt...	
.text:7610351A	?ValidateStdIndex@CImplStdA...	
.text:7610179A	?BuildMT@CAviMSROutPin@...	
.text:761004DA	?SearchList@@@YGJPAUIAsyn...	call ?ULongAdd@@@YGJK
.text:76100421	?Search@CAviMSRFilter@@@A...	call ?ULongAdd@@@YGJK
Line 35 of 44		

Occurrences of: ULong[^\s]

Occurrences of: ULongAdd

Address	Function	Instruction
.text:35532BA1	_ConvertVideoInfoToVideoInf...	call ?ULongAdd@@@YGJK
.text:35620869	??0CImplReader_1@@@QAE@P...	call ?ULongAdd@@@YGJK
.text:356208F0	??0CImplReader_1@@@QAE@P...	call ?ULongAdd@@@YGJK
.text:35620930	??0CImplReader_1@@@QAE@P...	call ?ULongAdd@@@YGJK
.text:355335DD	_CheckMPEG1VideoInfoType@4	call ?ULongAdd@@@YGJK
.text:3557CC3B	?GetMaxConnectionsSize@CF...	call ?ULongAdd@@@YGJK
.text:3562094C	??0CImplReader_1@@@QAE@P...	call ?ULongAdd@@@YGJK
.text:3554D675	_CheckMPEG2VideoInfoType@4	call ?ULongAdd@@@YGJK
.text:355772FB	?CopyImage@CBaseControlVi...	call ?ULongAdd@@@YGJK
.text:3557CC28	?GetMaxConnectionsSize@CF...	call ?ULongAdd@@@YGJK
.text:356210E6	sub_356210D4	call ?ULongAdd@@@YGJK
.text:3557CC4C	?GetMaxConnectionsSize@CF...	call ?ULongAdd@@@YGJK
.text:3557CC8A	?GetMaxConnectionsSize@CF...	call ?ULongAdd@@@YGJK
.text:3557CCD3	?GetMaxConnectionsSize@CF...	call ?ULongAdd@@@YGJK
.text:35582C5D	?MediaTypeToText@@@YGJVC...	call ?ULongAdd@@@YGJK
.text:35592000	?GetMediaType@CAVDec@@@...	call ?ULongAdd@@@YGJK
.text:3559300C	?StartStreaming@CMJPGEnc...	call ?ULongAdd@@@YGJK
.text:35593B95	?Transform@CMJPGEnc@@@U...	call ?ULongAdd@@@YGJK
.text:355CBD7C	?CopyRGBSurfToDIB@CALloca...	call ?ULongAdd@@@YGJK
.text:355D45C8	?SetToVideoInfoHeader2@VP...	call ?ULongAdd@@@YGJK
.text:355EC6F1	?CopyRGBSurfToDIB@CALloca...	call ?ULongAdd@@@YGJK
.text:355FF06F	?BuildMT@CAviMSROutPin@...	call ?ULongAdd@@@YGJK
.text:355FF87F	?SearchList@@@YGJPAUIAsyn...	call ?ULongAdd@@@YGJK
.text:355FF95D	?Search@CAviMSRFilter@@@A...	call ?ULongAdd@@@YGJK
.text:35600A68	?ValidateSuperIndex@CImplSt...	call ?ULongAdd@@@YGJK
.text:35600ADB	?ValidateStdIndex@CImplStdA...	call ?ULongAdd@@@YGJK
.text:3560126F	??0CImplOldAviIndex@@@QAE...	call ?ULongAdd@@@YGJK
Line 45 of 46		

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

UInt8ToChar

ByteToInt8

ByteToChar

ShortToInt8

ShortToUChar

ShortToChar

UShortToUInt8

UShortToShort

IntToInt8

IntToUChar

IntToChar

UInt8Add

UShortAdd

UIntAdd

ULongAdd

SizeTAdd

ULongLongAdd

UInt8Sub

UShortSub

UIntSub

ULongSub

SizeTSub

ULongLongSub

StringCbGets

StringCbGetsEx

StringCbLength

StringCbPrintf

StringCbPrintfEx

StringCbVPrintf

StringCbVPrintfEx

StringCchCat

StringCchCatEx

StringCchCatN

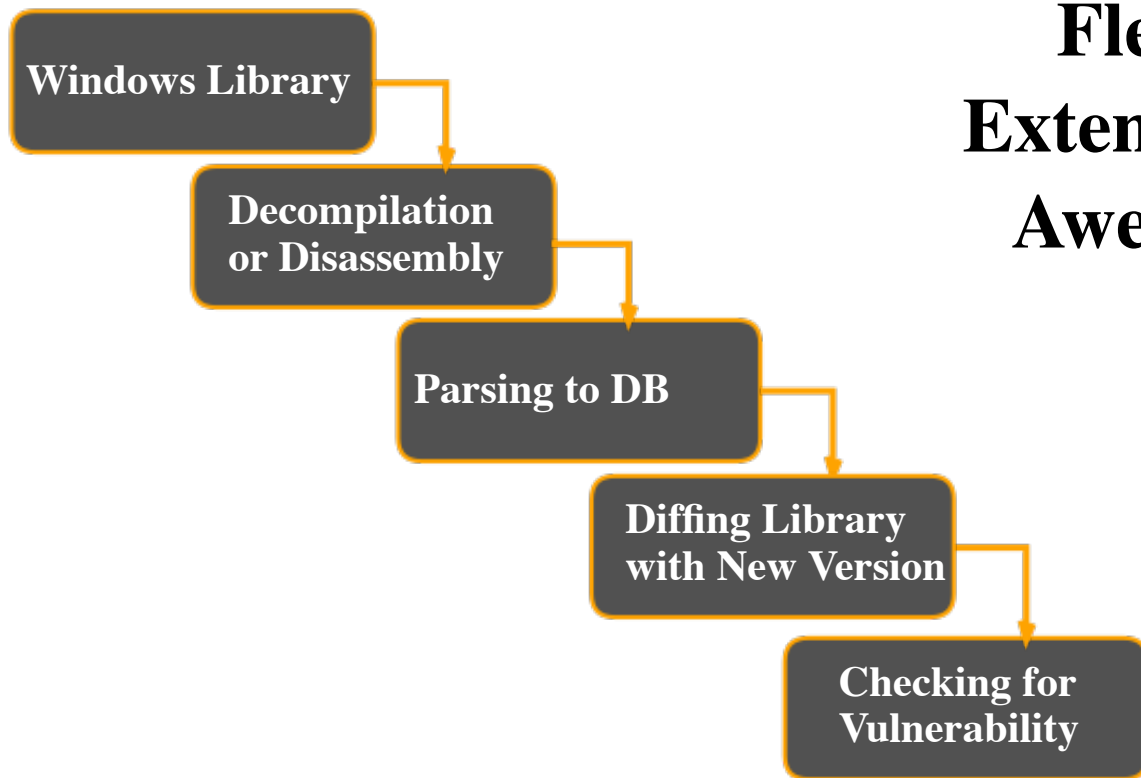
StringCchCatNEx

StringCchCopy

... and many many more



The Approach



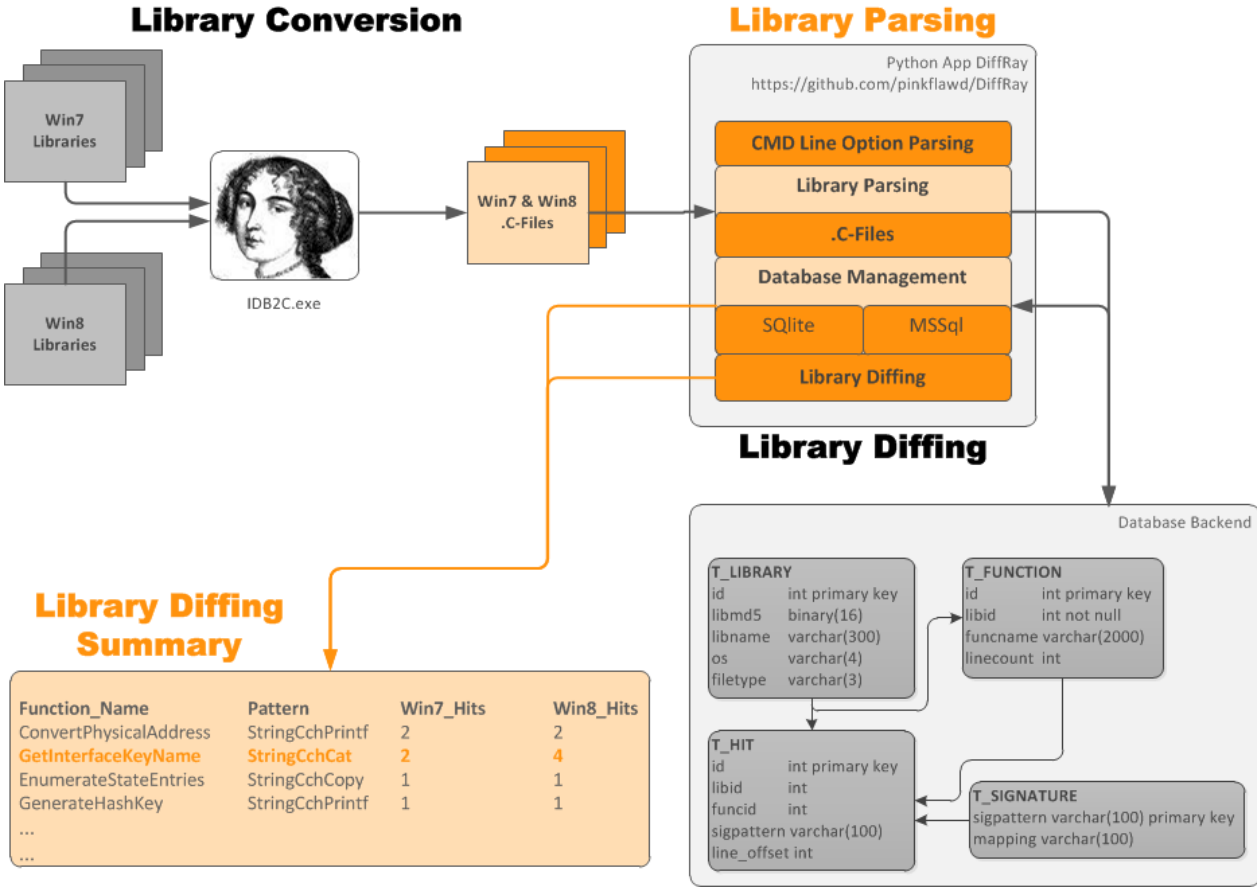
Flexible.
Extendible.
Awesome.



The Solution



Pretty, eh??



Getting the .C

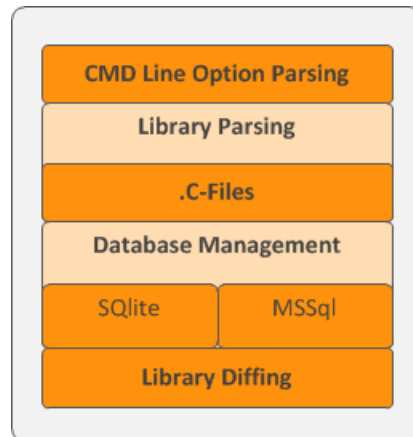
Library Conversion using IDA Pro



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

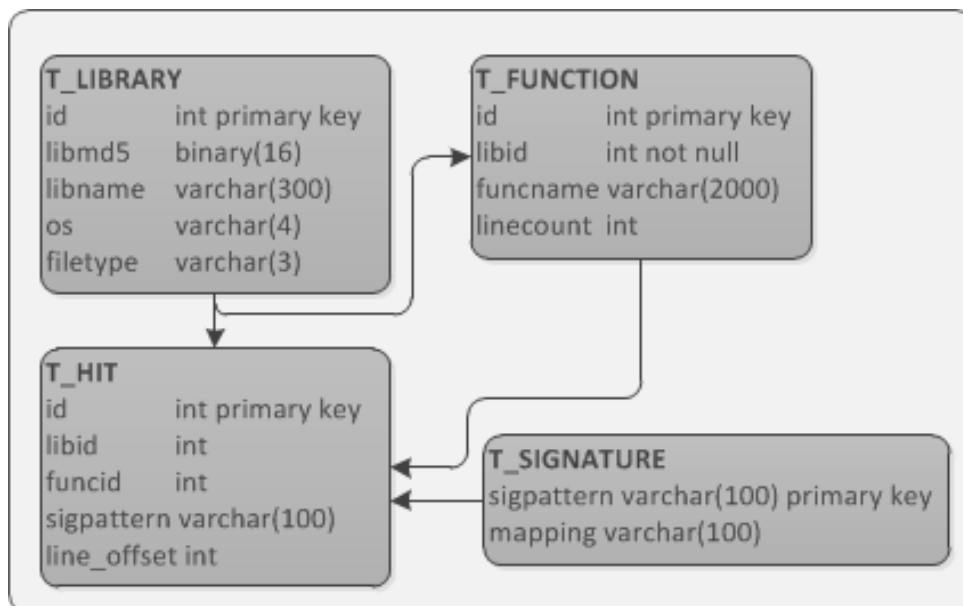
Library Parsing

- 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:

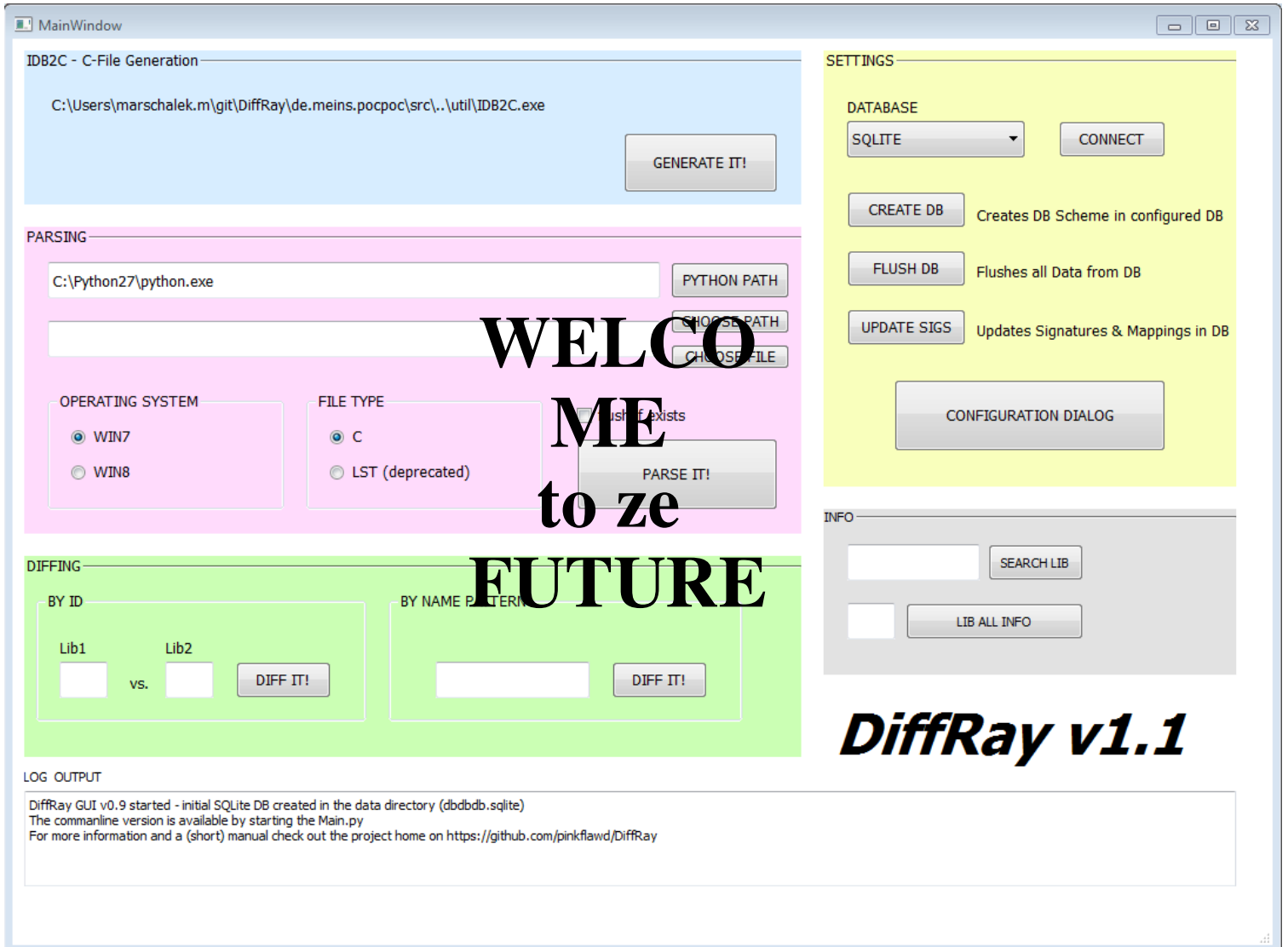
```
python [dir]\src\Main.py --search_libs [libname_pattern]
```

```
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]
```





DiffRay v1.1

DEMO TIME



Findings



Windows 7 (ULongAdd) bcrypt.dll!ConvertRsaPrivateBlobToFullRsa

```
text:6D80C94E      push    18h
text:6D80C950      pop     eax
text:6D80C951      lea    edi, [ebp+var_2C]
text:6D80C954      rep    movsd
text:6D80C956      lea    ecx, [ebp+var_4]
text:6D80C959      push   ecx
text:6D80C95A      push   [ebp+var_24]
text:6D80C95D      mov    [ebp+var_4], eax
text:6D80C960      push   eax
text:6D80C961      call   _ULongAdd@12 ; ULongAdd(x,x,x)
text:6D80C966      test   eax, eax
text:6D80C968      jnl   loc_6D80CB1E
text:6D80C96E      mov    esi, [ebp+var_20]
text:6D80C971      lea    eax, [ebp+var_4]
text:6D80C974      push   eax
text:6D80C975      push   esi
text:6D80C976      push   [ebp+var_4]
text:6D80C979      call   _ULongAdd@12 ; ULongAdd(x,x,x)
text:6D80C97E      test   eax, eax
text:6D80C980      jnl   loc_6D80CB1E
text:6D80C986      mov    ebx, [ebp+var_1C]
text:6D80C989      lea    eax, [ebp+var_4]
text:6D80C98C      push   eax
text:6D80C98D      push   ebx
text:6D80C98E      push   [ebp+var_4]
text:6D80C991      call   _ULongAdd@12 ; ULongAdd(x,x,x)
text:6D80C996      test   eax, eax
text:6D80C998      jnl   loc_6D80CB1E
text:6D80C99E      mov    edi, [ebp+var_18]
text:6D80C9A1      lea    eax, [ebp+var_4]
text:6D80C9A4      push   eax
text:6D80C9A5      push   edi
text:6D80C9A6      push   [ebp+var_4]
text:6D80C9A9      call   _ULongAdd@12 ; ULongAdd(x,x,x)
text:6D80C9AE      test   eax, eax
text:6D80C9B0      jnl   loc_6D80CB1E
```

000BD4E |6D80C94E: ConvertRsaPrivateBlobToFullRsa(x,x,x,x,x)+14

Windows 8 bcrypt.dll!ConvertRsaPrivateBlobToFullRsa

The screenshot shows a debugger window for the function `ConvertRsaPrivateBlobToFullRsa` in `bcrypt.dll`. The assembly code is displayed in a dark-themed window with various tabs like Hex View-A, Structures, Enums, Imports, and Exports. The code includes instructions such as `stosd`, `mov`, `push`, `pop`, `lea`, `rep movsd`, `add`, `cmp`, `jb`, `mov`, `lea`, `push`, `jmp`, and `short`. Three red arrows point to the instructions `mov ecx, [ebp+var_F0]`, `mov esi, [ebp+var_EC]`, and `mov ecx, [ebp+var_E4]`. A comment `; CODE XREF: ConvertRsaPrivateBlobToFullRsa(x,x,x,x)+16B↓j` is visible at the bottom right of the code block.

```
.text:1000BFEE      stosd
.text:1000BFEF      mov     [ebp+Dst], ecx
.text:1000BFF5      stosd
.text:1000BFF6      push   6
.text:1000BFF8      pop    ecx
.text:1000BFF9      mov    esi, edx
.text:1000BFFB      lea   edi, [ebp+var_F0]
.text:1000C001      rep   movsd
.text:1000C003      mov    ecx, [ebp+var_F0]
.text:1000C009      add    ecx, 18h
.text:1000C00C      mov    [ebp+var_C4], edx
.text:1000C012      cmp    ecx, 18h
.text:1000C015      jb    loc_1000C281
.text:1000C01B      mov    esi, [ebp+var_EC]
.text:1000C021      lea   eax, [esi+ecx]
.text:1000C024      cmp    eax, ecx
.text:1000C026      jb    loc_1000C281
.text:1000C02C      mov    ebx, [ebp+var_E8]
.text:1000C032      lea   edx, [ebx+eax]
.text:1000C035      cmp    edx, eax
.text:1000C037      jb    loc_1000C281
.text:1000C03D      mov    ecx, [ebp+var_E4]
.text:1000C043      lea   eax, [ecx+edx]
.text:1000C046      mov    [ebp+var_DC], eax
.text:1000C04C      cmp    eax, edx
.text:1000C04E      jb    loc_1000C281
.text:1000C054      cmp    [ebp+arg_0], eax
.text:1000C057      jnb   short loc_1000C069
.text:1000C059      mov    esi, 80090005h
.text:1000C05E      push  626h
.text:1000C063      loc_1000C063:      push  esi
.text:1000C064      jmp    loc_1000C28E
; CODE XREF: ConvertRsaPrivateBlobToFullRsa(x,x,x,x)+16B↓j
```

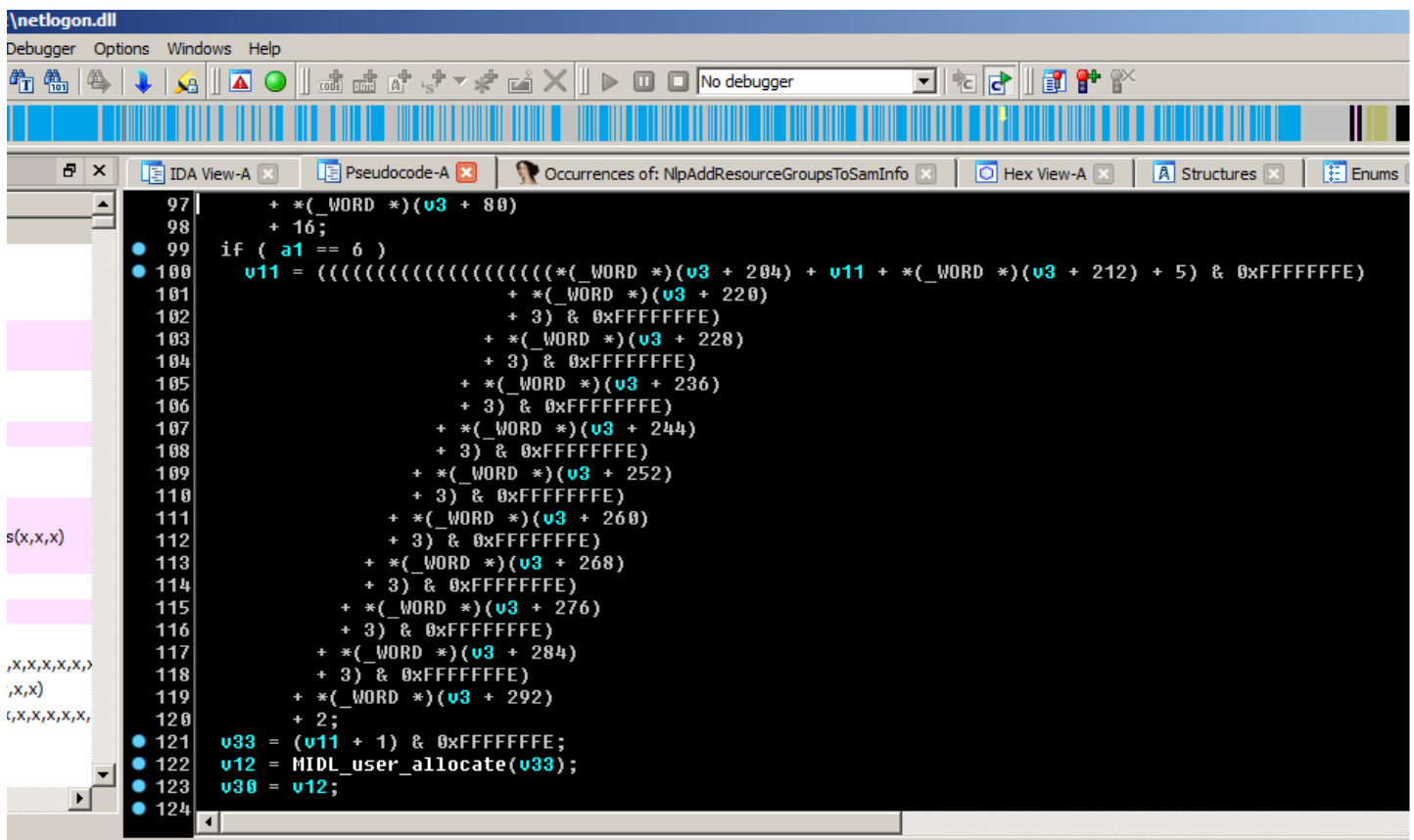
Windows 8 (ULongAdd)

netlogon.dll! NlpAddResourceGroupsToSamInfo

```
syswin8\netlogon.dll
Options Windows Help
No debugger
IDA View-A x Pseudocode-A x Occurrences of: NlpAddResourceGroupsToSamInfo x Hex View-A x Structures x Enums x Im
109 }
110 if ( a1 == 6 )
111 {
112     if ( RtlULongAdd(size, *((_WORD *)v3 + 106) + *((_WORD *)v3 + 102) + 4, &size) < 0
113         || (v12 = 2, NetpULongPtrRoundUp(size, 2, &size) < 0)
114         || RtlULongAdd(size, *((_WORD *)v3 + 110) + 2, &size) < 0
115         || NetpULongPtrRoundUp(size, 2, &size) < 0
116         || RtlULongAdd(size, *((_WORD *)v3 + 114) + 2, &size) < 0
117         || NetpULongPtrRoundUp(size, 2, &size) < 0
118         || RtlULongAdd(size, *((_WORD *)v3 + 118) + 2, &size) < 0
119         || NetpULongPtrRoundUp(size, 2, &size) < 0
120         || RtlULongAdd(size, *((_WORD *)v3 + 122) + 2, &size) < 0
121         || NetpULongPtrRoundUp(size, 2, &size) < 0
122         || RtlULongAdd(size, *((_WORD *)v3 + 126) + 2, &size) < 0
123         || NetpULongPtrRoundUp(size, 2, &size) < 0
124         || RtlULongAdd(size, *((_WORD *)v3 + 130) + 2, &size) < 0
125         || NetpULongPtrRoundUp(size, 2, &size) < 0
126         || RtlULongAdd(size, *((_WORD *)v3 + 134) + 2, &size) < 0
127         || NetpULongPtrRoundUp(size, 2, &size) < 0
128         || RtlULongAdd(size, *((_WORD *)v3 + 138) + 2, &size) < 0
129         || NetpULongPtrRoundUp(size, 2, &size) < 0
130         || RtlULongAdd(size, *((_WORD *)v3 + 142) + 2, &size) < 0
131         || NetpULongPtrRoundUp(size, 2, &size) < 0
132         || RtlULongAdd(size, *((_WORD *)v3 + 146) + 2, &size) < 0 )
133     {
134         NlPrintRoutine(256, L"NlpAddResourceGroupsToSamInfo: Integer overflow in length calculation at line %d\n",
135             return -1073741675; 7248-8111-6932-1904-2648
136     }
137 }
```

Windows 7

netlogon.dll! NlpAddResourceGroupsToSamInfo



```
\netlogon.dll
Debugger Options Windows Help
No debugger
IDA View-A Pseudocode-A Occurrences of: NlpAddResourceGroupsToSamInfo Hex View-A Structures Enums
97 | + *(_WORD *)(v3 + 80)
98 | + 16;
99 | if ( a1 == 6 )
100 |   v11 = ((((((((((((((((((((((((*(_WORD *)(v3 + 204) + v11 + *(_WORD *)(v3 + 212) + 5) & 0xFFFFFFFF)
101 |     + *(_WORD *)(v3 + 220)
102 |     + 3) & 0xFFFFFFFF)
103 |     + *(_WORD *)(v3 + 228)
104 |     + 3) & 0xFFFFFFFF)
105 |     + *(_WORD *)(v3 + 236)
106 |     + 3) & 0xFFFFFFFF)
107 |     + *(_WORD *)(v3 + 244)
108 |     + 3) & 0xFFFFFFFF)
109 |     + *(_WORD *)(v3 + 252)
110 |     + 3) & 0xFFFFFFFF)
111 |     + *(_WORD *)(v3 + 260)
112 |     + 3) & 0xFFFFFFFF)
113 |     + *(_WORD *)(v3 + 268)
114 |     + 3) & 0xFFFFFFFF)
115 |     + *(_WORD *)(v3 + 276)
116 |     + 3) & 0xFFFFFFFF)
117 |     + *(_WORD *)(v3 + 284)
118 |     + 3) & 0xFFFFFFFF)
119 |     + *(_WORD *)(v3 + 292)
120 |     + 2;
121 | v33 = (v11 + 1) & 0xFFFFFFFF;
122 | v12 = MIDL_user_allocate(v33);
123 | v30 = v12;
124 |
```

Windows 8 /ULongLongToUInt twext.dll! EscapeField

```
0 ; Attributes: bp-based frame
1
2 ; int __stdcall _EscapeField(LPCWSTR psz, int)
3 ?_EscapeField@@YGJPBGAPAG@Z proc near ; CODE XREF: SHGetParsingNameFromPropertyS
4
5 var_C          = dword ptr -0Ch |
6 var_8          = dword ptr -8
7 uBytes        = dword ptr -4
8 psz           = dword ptr 8
9 arg_4         = dword ptr 0Ch
10
11         mov     edi, edi
12         push   ebp
13         mov     ebp, esp
14         mov     eax, [ebp+arg_4]
15         sub     esp, 0Ch
16         push   ebx
17         mov     ebx, [ebp+psz]
18         push   esi
19         xor     esi, esi
20         push   edi
21         mov     [eax], esi
22         lea   eax, [ebp+var_8]
23         push   eax ; unsigned int *
24         push   ebx ; lpString
25         call  _lstrlenW@4 ; lstrlenW(x)
26         push   3
27         pop    ecx
28         mul   ecx
29         push   edx
30         push   eax ; unsigned __int64
31         call  ?ULongLongToUInt@@YGJ_KPAI@Z ; ULongLongToUInt(unsigned __
32         test   eax, eax
33         jmp   738145B0
```

Windows 7 Integer overflow twext.dll! EscapeField

```
.text:06D161E0 ; Attributes: bp-based frame
.text:06D161E0
.text:06D161E0 ; int __stdcall _EscapeField(LPCWSTR psz, int)
.text:06D161E0 ?_EscapeField@@YGJPBGPAPAG@Z proc near ; CODE XREF: SHGetParsingNameFromProperty
.text:06D161E0
.text:06D161E0 var_C          = dword ptr -0Ch
.text:06D161E0 uBytes        = dword ptr -8
.text:06D161E0 var_4         = dword ptr -4
.text:06D161E0 psz          = dword ptr 8
.text:06D161E0 arg_4        = dword ptr 0Ch
.text:06D161E0
.text:06D161E0          mov     edi, edi
.text:06D161E2          push   ebp
.text:06D161E3          mov     ebp, esp
.text:06D161E5          mov     eax, [ebp+arg_4]
.text:06D161E8          and     dword ptr [eax], 0
.text:06D161EB          sub     esp, 0Ch
.text:06D161EE          push   edi
.text:06D161EF          push   [ebp+psz] ; lpString
.text:06D161F2          call   ds:__imp__lstrlenW@4 ; lstrlenW(x)
.text:06D161F8          mov     edi, eax
.text:06D161FA          imul   edi, 3
.text:06D161FD          inc     edi
.text:06D161FE          cmp     edi, 20000h
.text:06D16204          ja     loc_6D16307
.text:06D16204          and     [ebp+uBytes], 0
.text:06D1620E          lea   eax, [ebp+uBytes]
.text:06D16211          push   eax ; uBytes
.text:06D16212          push   edi ; int
.text:06D16213          call   ??$LocalAllocArray@G@@YGJIPAPAG@Z ; LocalAllocArray<ushort>
.text:06D16218          mov     [ebp+var_4], eax
```


Drrrrivers...



Windows 8 cng.dll!

```
.text:0003EAE7  
.text:0003EAE7 loc_3EAE7: ; CODE XREF: ConvertRsaPrivateBlobToFullRsa(x,x,x,x)+C9↑j  
.text:0003EAE7 lea ecx, [esp+158h+var_14C]  
.text:0003EAE8 mov [esp+158h+var_14C], eax  
.text:0003EAE9 push ecx  
.text:0003EAEF mov edx, ebx  
.text:0003EAF0 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 loc_3EB61: ; CODE XREF: ConvertRsaPrivateBlobToFullRsa(x,x,x,x)+140↑j  
.text:0003EB61 cmp [ebp+arg_4], eax  
.text:0003EB64 jnb short loc_3EB70
```

Windows 7 cng.dll!

```
text:0000000000035ADE      add     ecx, r12d
text:0000000000035AE1      mov     [rax], ecx
text:0000000000035AE3      test   r14, r14
text:0000000000035AE6      jnz    short loc_35AEF
text:0000000000035AE8      jmp     ebx, ebx
text:0000000000035AEA      jmp     loc_35CC1
text:0000000000035AEF      ; -----
text:0000000000035AEF      loc_35AEF:                                ; CODE XREF: ConvertRsaPrivateBlobToFullRsa+106↑j
text:0000000000035AEF      cmp     dword ptr [rsp+218h+arg_18], ecx
text:0000000000035AF6      jnb    short loc_35B02
text:0000000000035AF8      mov     ebx, 80090028h
text:0000000000035AFD      jmp     loc_35CC1
text:0000000000035B02      ; -----
text:0000000000035B02      loc_35B02:                                ; CODE XREF: ConvertRsaPrivateBlobToFullRsa+116↑j
text:0000000000035B02      lea    rax, [rbx+18h]
text:0000000000035B06      mov     edx, edx ; int
text:0000000000035B08      lea    rcx, [rsp+218h+var_1B8] ; void *
text:0000000000035B0D      lea    rbx, [rax+r15]
text:0000000000035B11      lea    r8d, [rdx+20h] ; size_t
text:0000000000035B15      mov     [rsp+218h+arg_0], rax
text:0000000000035B1D      lea    r13, [rbx+rbp]
text:0000000000035B21      call   memset
text:0000000000035B26      mov     rcx, [rsp+218h+arg_0]
text:0000000000035B2E      lea    r11, [rsp+218h+var_1B8]
text:0000000000035B33      lea    rax, [rsp+218h+var_178]
text:0000000000035B3B      mov     r9d, ebp
text:0000000000035B3E      mov     [rsp+218h+var_1C8], r11
text:0000000000035B43      mov     [rsp+218h+var_1D0], 1
text:0000000000035B4B      mov     [rsp+218h+var_1D8], rax
text:0000000000035B50      mov     [rsp+218h+var_1E0], edi
text:0000000000035B54      lea    rax, [rsi+r13]
text:0000000000035B58      mov     r8, rbx
text:0000000000035B5B      mov     [rsp+218h+var_1E8], rax
text:0000000000035B60      mov     edx, r15d
text:0000000000035B63      mov     dword ptr [rsp+218h+var_1F0], esi
text:0000000000035B67      mov     [rsp+218h+var_1F8], r13
```

Windows 8

ksecdd.dll! SspiCopyAuthIdentity

```
text:00018BF9 ; -----
text:00018BF9
text:00018BF9 loc_18BF9: ; CODE XREF: SspiCopyAuthIdentity(x,x)+3C1j
text:00018BFD movzx edx, word ptr [ebx+10h]
text:00018C00 lea eax, [ebp+arg_0]
text:00018C02 push 40h
text:00018C03 pop ecx
text:00018C04 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@YGPAKK@Z ; SspiLocalAlloc(ulong)
```

Windows 7

ksecdd.dll! SspiCopyAuthIdentity

```
0000000018A97 test    rax, rax
0000000018A9A jz     loc_188A0
0000000018AA0 mov    eax, [rbx+18h]
0000000018AA3 imul  eax, edi
0000000018AA6 bt     dword ptr [rbx+2Ch], 10h
0000000018AAB jnb   short loc_18AB3
0000000018AAD add    eax, 7
0000000018AB0 and   eax, 0FFFFFFF8h
0000000018AB3 loc_18AB3:
0000000018AB3 mov    rdx, [rbx+10h] ; CODE XREF: SspiCopyAuthIdentity+25B1j
0000000018AB7 mov    r8d, eax ; void *
0000000018ABA call  memmove ; size_t
0000000018ABF mov    r11d, [rbx+18h]
0000000018AC3 mov    [r12+18h], r11d
0000000018AC8 loc_18AC8:
0000000018AC8 cmp    qword ptr [rbx+20h], 0 ; CODE XREF: SspiCopyAuthIdentity+2191j
0000000018ACD jz     short loc_18B35
0000000018ACF mov    eax, [rbx+28h]
0000000018AD2 movzx  edi, r13w
0000000018AD6 xor    edx, edx
0000000018AD8 add    eax, r14d
0000000018ADB mov    r13d, 0FFFFFFF8h
0000000018AE1 imul  eax, edi
0000000018AE4 add    eax, 7
0000000018AE7 and   eax, r13d
0000000018AEA div    edi
0000000018AEC mov    [r12+28h], eax
0000000018AF1 lea   ecx, [rax+1]
0000000018AF4 imul  ecx, edi ; unsigned __int32
0000000018AF7 call  ?SspiLocalAlloc@YAPEAXKQZ ; SspiLocalAlloc(ulong)
0000000018AFC mov    rcx, rax ; void *
0000000018AFF mov    [r12+20h], rcx
0000000018B04 test   rax, rax
0000000018B07 jz     loc_188A0
0000000018B0D mov    eax, [rbx+28h]
```

Windows 8

srvnet.dll!

SrvNetAllocatePoolWithTagPriority

```
text:0001A753 ; Attributes: bp-based frame
text:0001A753
text:0001A753 ; int __stdcall SrvNetAllocatePoolWithTagPriority(PPOOL_TYPE PoolType, int, ULONG Tag, EX_POOL_PRIORITY Priority)
text:0001A753         public _SrvNetAllocatePoolWithTagPriority@16
text:0001A753         _SrvNetAllocatePoolWithTagPriority@16 proc near
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
text:000000000001F8F0 public SrvNetAllocatePoolWithTagPriority
text:000000000001F8F0 SrvNetAllocatePoolWithTagPriority proc near
text:000000000001F8F0 ; CODE XREF: SrvLibLookasideCreatePool+4E1p
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
```



Windows 7 cryptdlg!DecodeAttrSequence

```
ccurrences of: Decode x Hex View-A x Structures x Enums x Imports x Exports x
.text:64481D28 and [ebp+cbEncoded], 0
.text:64481D2C
.text:64481D2C loc_64481D2C: ; CODE XREF: DecodeAttrSequence(uLong,char c
.text:64481D2F mov eax, [ebp+uBytes]
.text:64481D2F mov ecx, [ebp+arg_18]
.text:64481D32 mov [ebp+pcbStructInfo], eax
.text:64481D35 mov eax, [ebp+var_C]
.text:64481D38 mov eax, [eax+4]
.text:64481D3B lea eax, [eax+ecx*8]
.text:64481D3E lea ecx, [ebp+pcbStructInfo]
.text:64481D41 push ecx ; pcbStructInfo
.text:64481D42 push edi ; pvStructInfo
.text:64481D43 push 0 ; dwFlags
.text:64481D45 push dword ptr [eax] ; cbEncoded
.text:64481D47 push dword ptr [eax+4] ; pbEncoded
.text:64481D4A push 16h ; lpszStructType
.text:64481D4C push 1 ; dwCertEncodingType
.text:64481D4E call ds:__imp__CryptDecodeObject@28 ; CryptDecodeObject(x,x,x,x,x
.text:64481D54 test eax, eax
.text:64481D56 jz loc_64481E0A
.text:64481D5C mov eax, [ebx+4]
mov ecx, [ebp+cbEncoded]
mov [ecx+eax], esi
push dword ptr [edi] ; lpString
call ds:__imp__lstrlenA@4 ; lstrlenA(x)
inc eax
push eax ; Size
mov [ebp+pcbStructInfo], eax
push dword ptr [edi] ; Src
push esi ; Dst
call _mencpy
add esi, [ebp+pcbStructInfo]
mov ecx, [ebp+4]
```



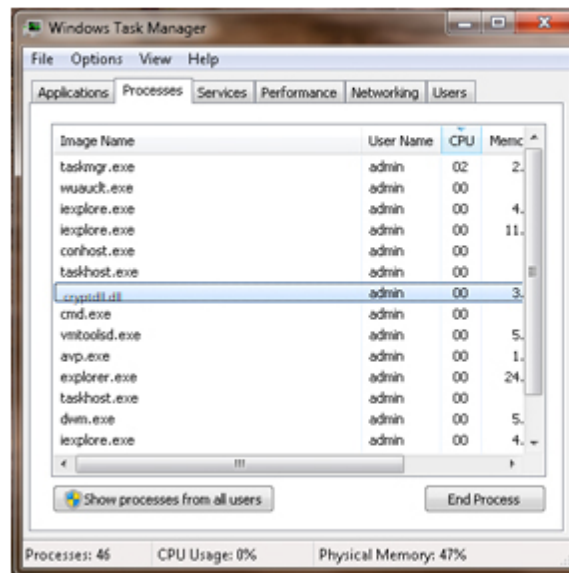
Windows 8 cryptdlg!DecodeAttrSequence

```
Hex View-A | Structures | Enums | Imports | Exports
push     edx             ; pcbStructInfo
push     esi             ; pvStructInfo
mov     [ebp+pcbStructInfo], eax
mov     eax, [edi+4]
push     ebx             ; dwFlags
push     dword ptr [eax+ecx*8] ; cbEncoded
push     dword ptr [eax+ecx*8+4] ; pbEncoded
push     16h             ; lpszStructType
push     1                ; dwCertEncodingType
call    ds:__imp__CryptDecodeObject@28 ; CryptDecodeObject(x,x,x,x,x,x,x)
test    eax, eax
jz     loc_64481E79
mov     eax, [ebp+arg_14]
mov     eax, [eax+4]
mov     ecx, [ebp+Dst]
mov     edx, [ebp+pbEncoded]
mov     [edx+eax], ecx
mov     [ebp+pcbStructInfo], ebx
mov     edx, [esi]
mov     eax, [ebp+pcbStructInfo]
?StringCchLengthA@@YGJPDIPAI@Z ; StringCchLengthA(char const *,uint,uint *)
mov     eax, eax
short  loc_64481DC4
mov     eax, [ebp+pcbStructInfo]
test    eax, eax
jz     short loc_64481DC4
inc     eax
push    eax              ; Size
mov     [ebp+pcbStructInfo], eax
push    dword ptr [esi] ; Src
push    [ebp+Dst]       ; Dst
call    _memcpy
```

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  
);
```

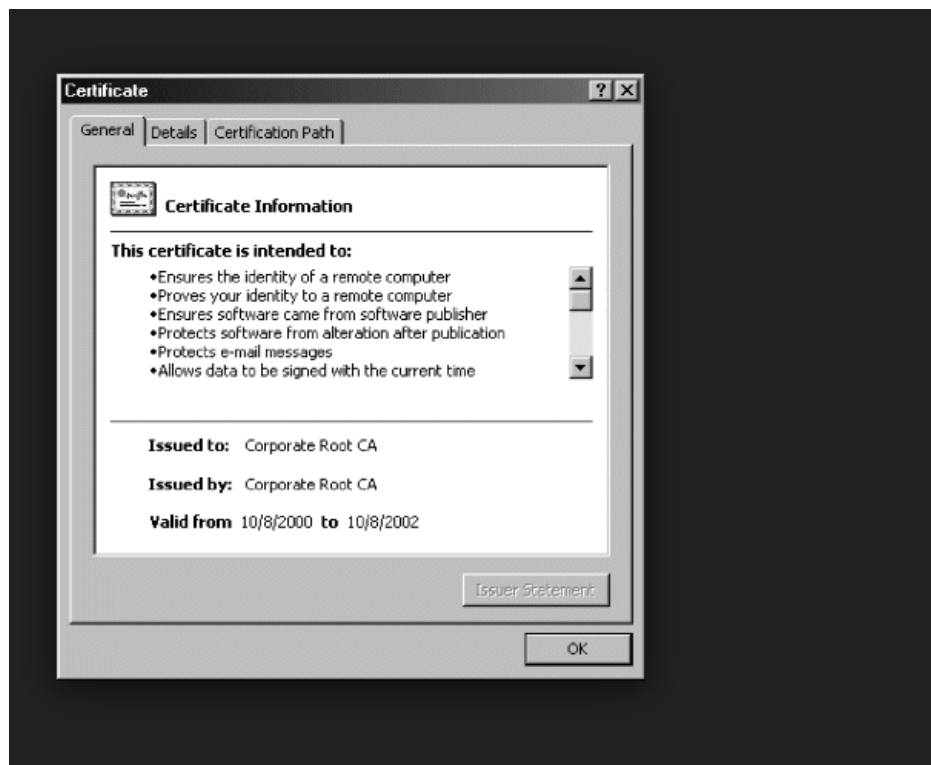
pcbStructInfo [in, out]

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 Duffing.

"Most of all I love your vulnerability."