

ExploitSpotting: Locating Vulnerabilities Out Of Vendor Patches Automatically

Defcon 18
August 1st, 2010
Las Vegas, USA

Jeongwook Oh
Sr. Security Researcher
WebSense Inc.

Why?

- I worked on a security product last 5 years.
 - The IPS and vulnerability scanner needed signatures
- We needed technical details on the patches
 - The information was not provided by the vendors
 - In recent years, a program called MAPP appeared from Microsoft, but many times it's not enough
- You have two options in this case:
 - Use your own eye balls to compare disassemblies
 - **Use binary diffing tools**
- Patch analysis using **binary diffing tools** is the only healthy way to obtain some valuable information out of the patches.

How?

- I'll show you whole process for a typical binary diffing
 - You should grab an idea what binary diffing is
- The example shown next will show the typical example of binary diffing process
 - The patch(MS10-018) is for "CVE-2010-0806" vulnerability.

Example: CVE-2010-0806 Patch Description from CVE Web Page

<http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-0806>

Use-after-free vulnerability in the Peer Objects component (aka **iepeers.dll**) in Microsoft Internet Explorer 6, 6 SP1, and 7 allows remote attackers to execute arbitrary code via vectors involving access to an invalid pointer after the deletion of an object, as exploited in the wild in March 2010, aka "Uninitialized Memory Corruption Vulnerability."

CVE-2010-0806 Patch Analysis

Acquire Patches

- Download the patch by visiting patch page(MS10-018) and following the OS and IE version link.
 - For XP IE 7, I used following link from the main patch page to download the patch file. (<http://www.microsoft.com/downloads/details.aspx?FamilyID=167ed896-d383-4dc0-9183-cd4cb73e17e7&displaylang=en>)

Cumulative Security Update for Internet Explorer 7 for Windows XP (KB980182)

Brief Description

This update addresses the vulnerability discussed in Microsoft Security Bulletin MS10-018. To find out if other security updates are available for you, see the Overview section of this page.

On This Page

↓ [Quick Details](#)

↓ [System Requirements](#)

↓ [Additional Information](#)

↓ [Overview](#)

↓ [Instructions](#)

↓ [Related Resources](#)

Download

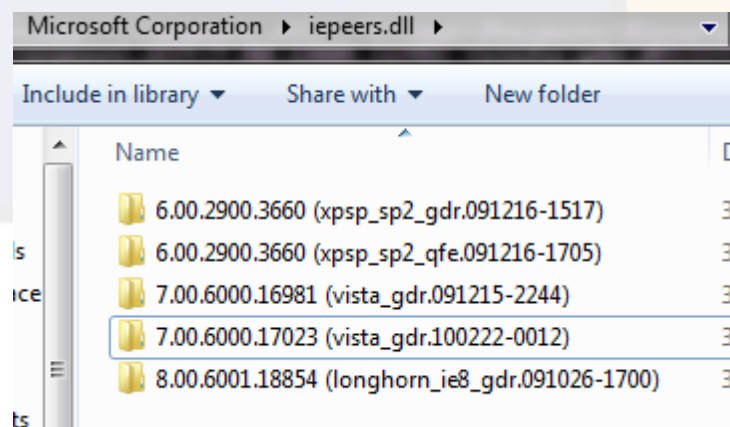
Quick Details

File Name:	IE7-WindowsXP-KB980182-x86-ENU.exe
Version:	980182
Security Bulletins:	MS10-018
Knowledge Base (KB) Articles:	KB980182
Date Published:	3/24/2010

CVE-2010-0806 Patch Analysis

Acquire unpatched files

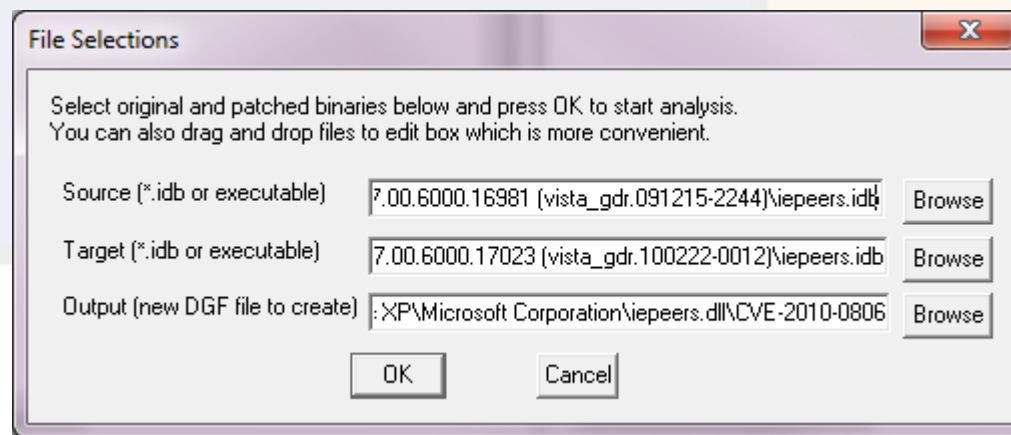
- You need to collect unpatched files from the operating system that the patch is supposed to be installed.
 - I used SortExecutables.exe from DarunGrim2 package to consolidate the files. The files will be inside a directory with version number string.



CVE-2010-0806 Patch Analysis

Load the binaries from DarunGrim2

- Launch DarunGrim2.exe and select "File → New Diffing from IDA" from the menu
 - You need to wait from few seconds to few minutes depending on the binary size and disassembly complexity.



CVE-2010-0806 Patch Analysis

Binary Level Analysis

- Now you have the list of functions
- Find any eye catching functions
 - Like following, the match rate(the last column value) 86% and 88% is a strong indication that it has some minor code change which can be a security patch.

Original	Unmat...	Patched	Unmat...	Different	Matched	Mat...
<input type="checkbox"/> ??1?\$CComObject@VCHomePage@@...	0	??1?\$CComObject@VCHomePage@@...	0	0	0	0%
<input type="checkbox"/> ?Invoke@?SDispatchImpl@UIClientCap...	0	?Invoke@?SDispatchImpl@UIClientCap...	0	0	0	0%
<input type="checkbox"/> ?Invoke@?SDispatchImpl@UIHomePa...	0	?Invoke@?SDispatchImpl@UIHomePa...	0	0	0	0%
<input type="checkbox"/> ?setAttribute@CPersistDataPeer@@U...	0	?setAttribute@CPersistDataPeer@@UA...	2	4	17	86%
<input type="checkbox"/> ?setAttribute@CPersistUserData@@U...	0	?setAttribute@CPersistUserData@@UA...	1	4	17	88%
<input type="checkbox"/> _SHRegGetValueW@z8	0	_SHRegGetValueW@z8	0	0	1	100%
<input type="checkbox"/> _PathAddBackslashW@4	0	_PathAddBackslashW@4	0	0	1	100%
<input type="checkbox"/> bsearch	0	bsearch	0	0	1	100%

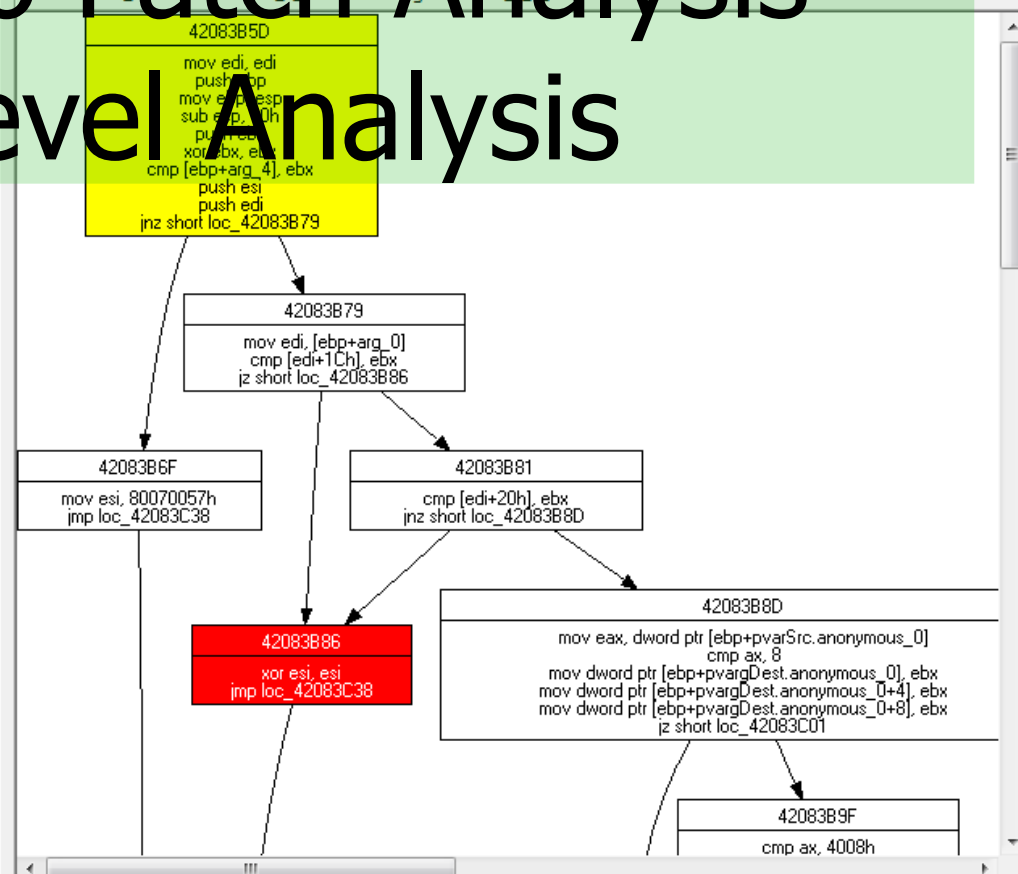
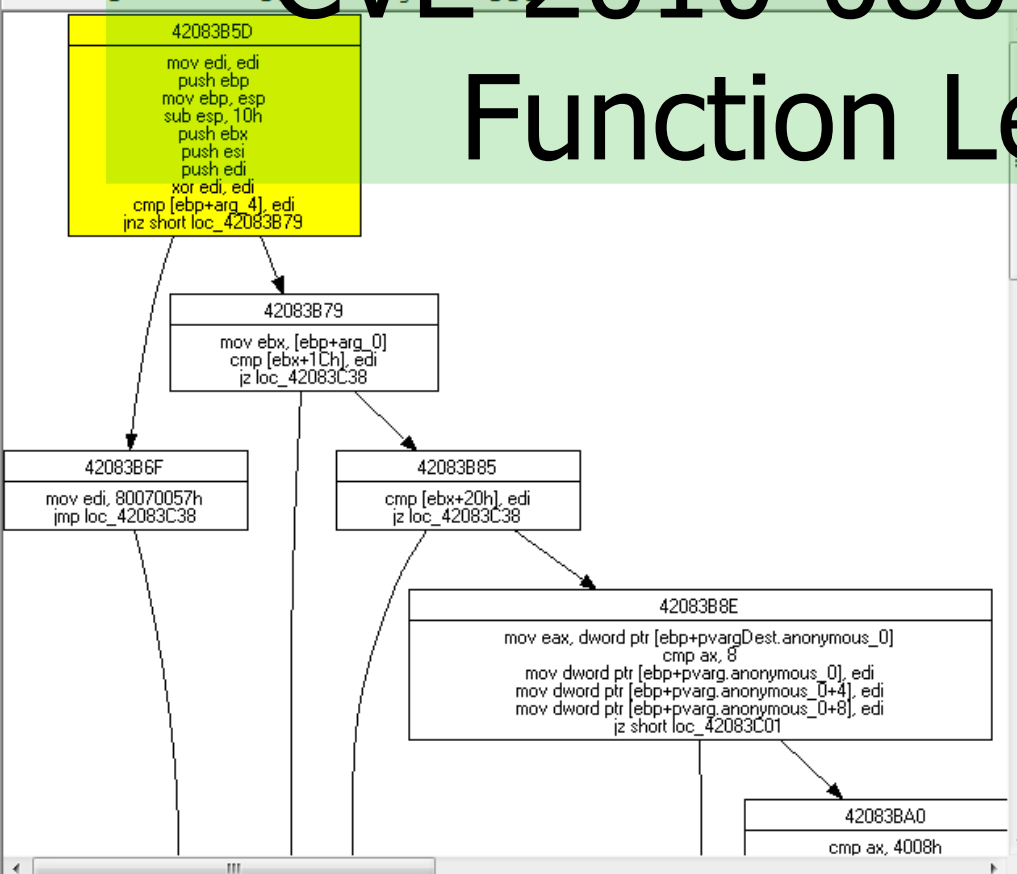
CVE-2010-0806 Patch Analysis

Function Level Analysis

- If you click the function match row, you will get a matching graphs.
- Color codes
 - The white blocks are matched blocks
 - The yellow blocks are modified blocks
 - The red blocks are unmatched blocks
- Unmatched block means that the block is inserted or removed.
 - So in this case, the red block is in patched part which means that block has been inserted in the patch.

CVE-2010-0806 Patch Analysis

Function Level Analysis



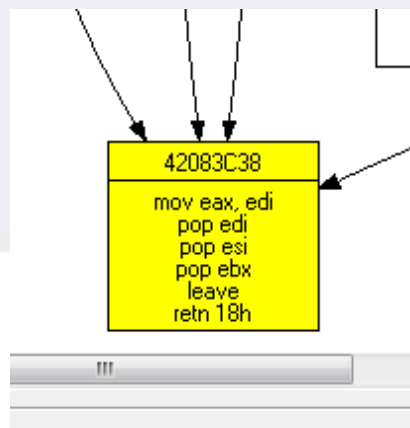
List Of Matches

Original	Patched	Match R...	Type	Fingerprint(Original)	Fingerprint(Patched)	Parent(Original)	Parent(Patched)
<input type="checkbox"/>	42083BFC				5c01020402		
<input type="checkbox"/>	42083B86				d801020102		
<input type="checkbox"/>	42083BF3	76%	Tree	1202027a040201025c010...	1202027a04020102	42083BE7	42083BE7
<input type="checkbox"/>	42083BE7	100%	Tree	8f0502	8f0502	42083BDC	42083BDC
<input type="checkbox"/>	42083BEE	100%	Tree	8f0502	8f0502	42083BDC	42083BDC
<input type="checkbox"/>	42083BB2	85%	Tree	8f05028f01025c01020402...	8f05028f01028f05025c01...	42083BAC	42083BAB
<input type="checkbox"/>	42083BD5	100%	Tree	7a010204027e01020401	7a010204027e01020401	42083BAC	42083BAB
<input type="checkbox"/>	42083BDC	100%	Tree	1b010101017a04010501	1b010101017a04010501	42083BCF	42083BCF
<input type="checkbox"/>	42083BAC	100%	Tree	1b01010501	1b01010501	42083BA6	42083BA5
<input type="checkbox"/>	42083BCF	100%	Tree	7e01020401	7e01020401	42083BA6	42083BA5
<input type="checkbox"/>	42083C2E	100%	Tree	5c010204028f0102120202	5c010204028f0102120202	42083C0D	42083C0D
<input type="checkbox"/>	42083C2C	100%	Tree	d801020102	d801020102	42083C0D	42083C0D
<input type="checkbox"/>	42083BA6	100%	Tree	1b01010501	1b01010501	42083BA0	42083B9F
<input type="checkbox"/>	42083C0D	100%	Tree	7a010204027a01020302d...	7a010204027a01020302d...	42083C01	42083C01
<input type="checkbox"/>	42083C0A	100%	Tree	5c01020402	5c01020402	42083C01	42083C01

CVE-2010-0806 Patch Analysis

Function Level Analysis

- So we just follow the control flow from the red block and we can see that esi is eventually set as return value(eax).
- We can guess that the patch is about sanitizing return value when some condition is not met or something.



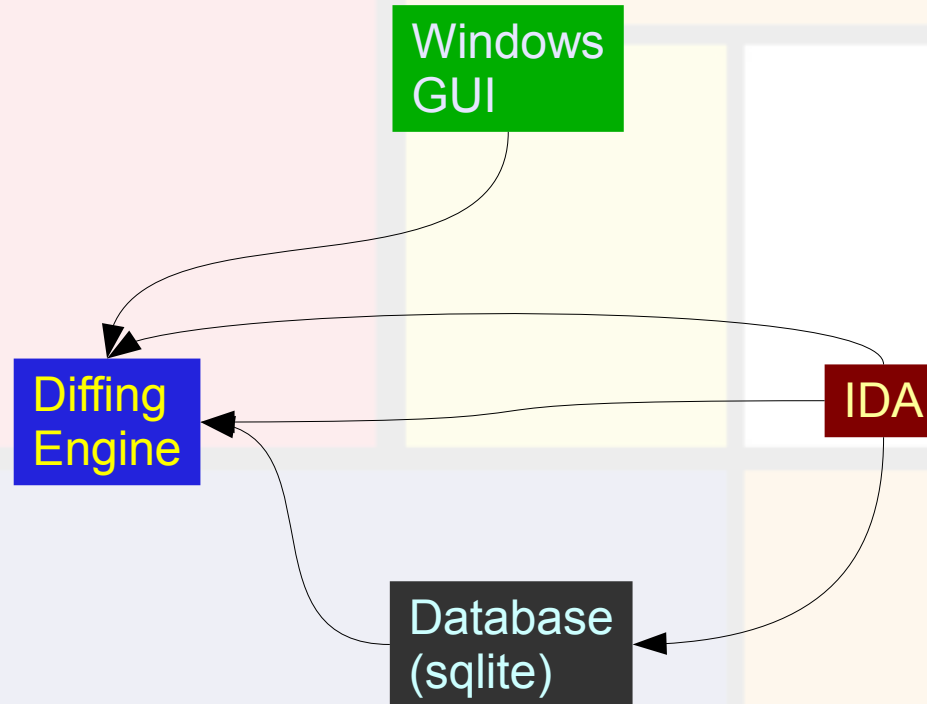
The Problems with Current Binary Diffing Tools

- Managing files are boring job.
 - Downloading patches
 - Storing old binaries/ Loading the files manually
- How do we know which function has security updates, not feature updates?
 - Just go through every modified functions?
 - How about if the modified functions are too many?

The Solution = DarunGrim 3

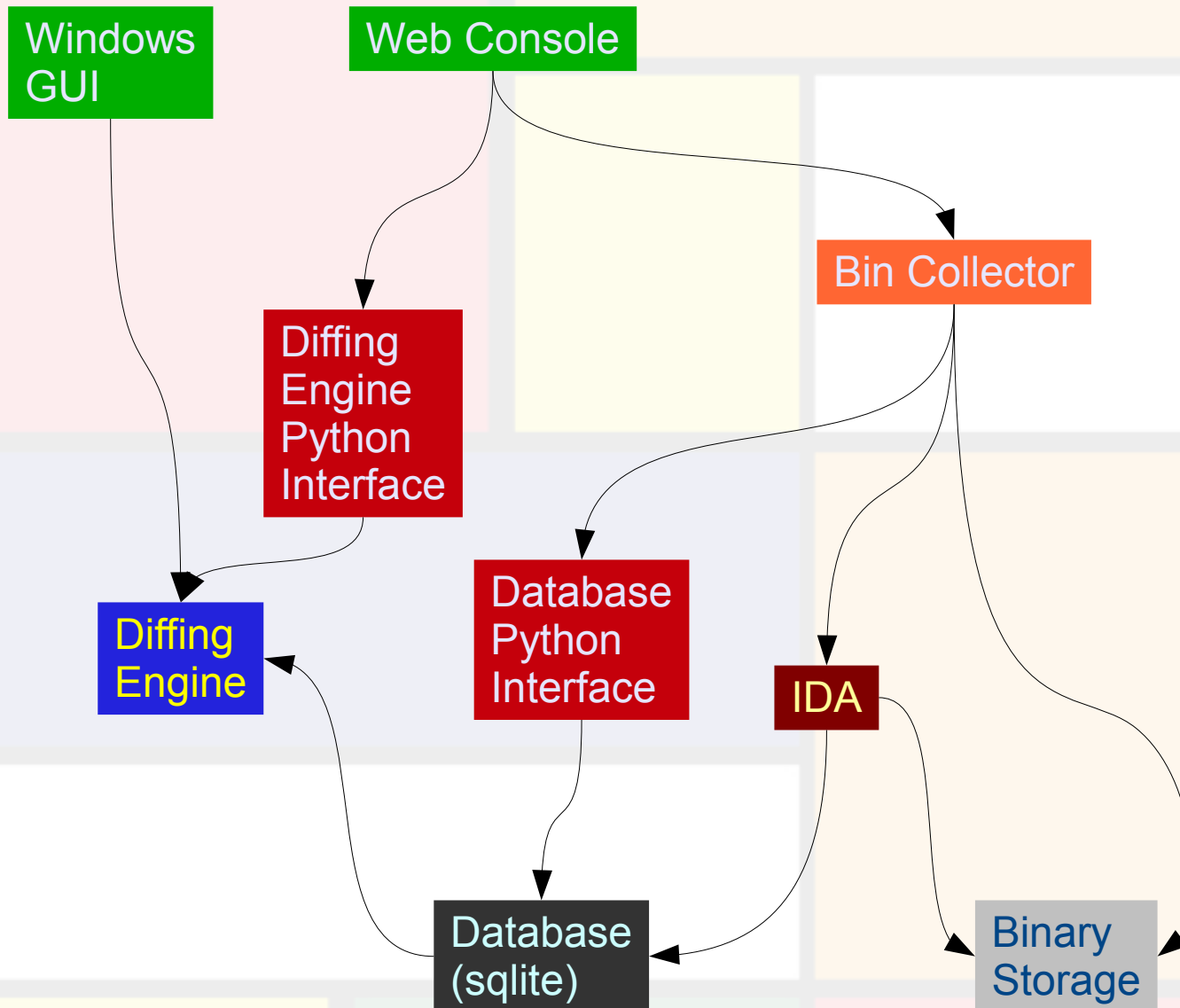
- **Bin Collector**
 - Binary Managing Functionality
 - Automatic patch download and extraction
 - Supports Microsoft Binaries
 - Will support other major vendors soon
- **Security Implication Score**
 - Shows you what functions have more security related patches inside it.
- **Web Interface**
 - User friendly
 - By clicking through and you get the diffing results

Architecture Comparison DarunGrim 2



Architecture Comparison

DarunGrim 3



Performing Diffing

- **Interactive**
- **Non-Interactive**

Performing Diffing: Interactive

- **Using DarunGrim2.exe UI**
 - Just put the path for each binary and DarunGrim2.exe will do the rest of the job.
- **DarunGrim2.exe + Two IDA sessions**
 - First launch DarunGrim2.exe
 - Launch two IDA sessions
 - First run DarunGrim2 plugin from the original binary
 - Secondly run DarunGrim2 plugin from the patched binary
 - DarunGrim2.exe will analyze the data that is collected through shared memory
- **Using DarunGrim Web Console: a DarunGrim 3 Way**
 - User friendly user interface
 - Includes "Bin Collector"/"Security Implication Score" support

Performing Diffing: Non-Interactive

- Using DarunGrim2C.exe command line tool
 - Handy, Batch-able, Quick
- Using DarunGrim Python Interface: a DarunGrim 3 Way
 - Handy, Batch-able, Quick, Really Scriptable

Diffing Engine Python Interface

```
import DarunGrimEngine
```

```
DarunGrimEngine.DiffFile( unpatched_filename, patched_filename,  
    output_filename, log_filename, ida_path  
)
```

- Perfoms diassembly using IDA
- Runs as a background process
- Runs DarunGrim IDA plugin automatically
- Runs the DiffEngine automatically on the files

Database Python Interface

```
import DarunGrimDatabaseWrapper

database = DarunGrimDatabaseWrapper.Database( filename )
for function_match_info in database.GetFunctionMatchInfo():
    if function_match_info.non_match_count_for_the_source > 0 or
function_match_info.non_match_count_for_the_target > 0:
        print function_match_info.source_function_name +
hex(function_match_info.source_address) + '\t',
        print function_match_info.target_function_name +
hex(function_match_info.target_address) + '\t',
        print str(function_match_info.block_type) + '\t',
        print str(function_match_info.type) + '\t',
        print str( function_match_info.match_rate ) + "%" + '\t',

        print database.GetFunctionDisasmLinesMap( function_match_info.source_file_id,
function_match_info.source_address )
        print database.GetMatchMapForFunction( function_match_info.source_file_id,
function_match_info.source_address )
```

Bin Collector

- Binary collection & consolidation system
 - Toolkit for constructing binary library
- It is managed through Web Console
 - It exposes some python interface, so it's scriptable if you want
- The whole code is written in Python
- It maintains indexes and version information on the binary files from the vendors.
- Download and extract patches automatically
 - Currently limited functionality
- Currently it supports Microsoft binaries
 - Adobe, Oracle binaries will be supported soon

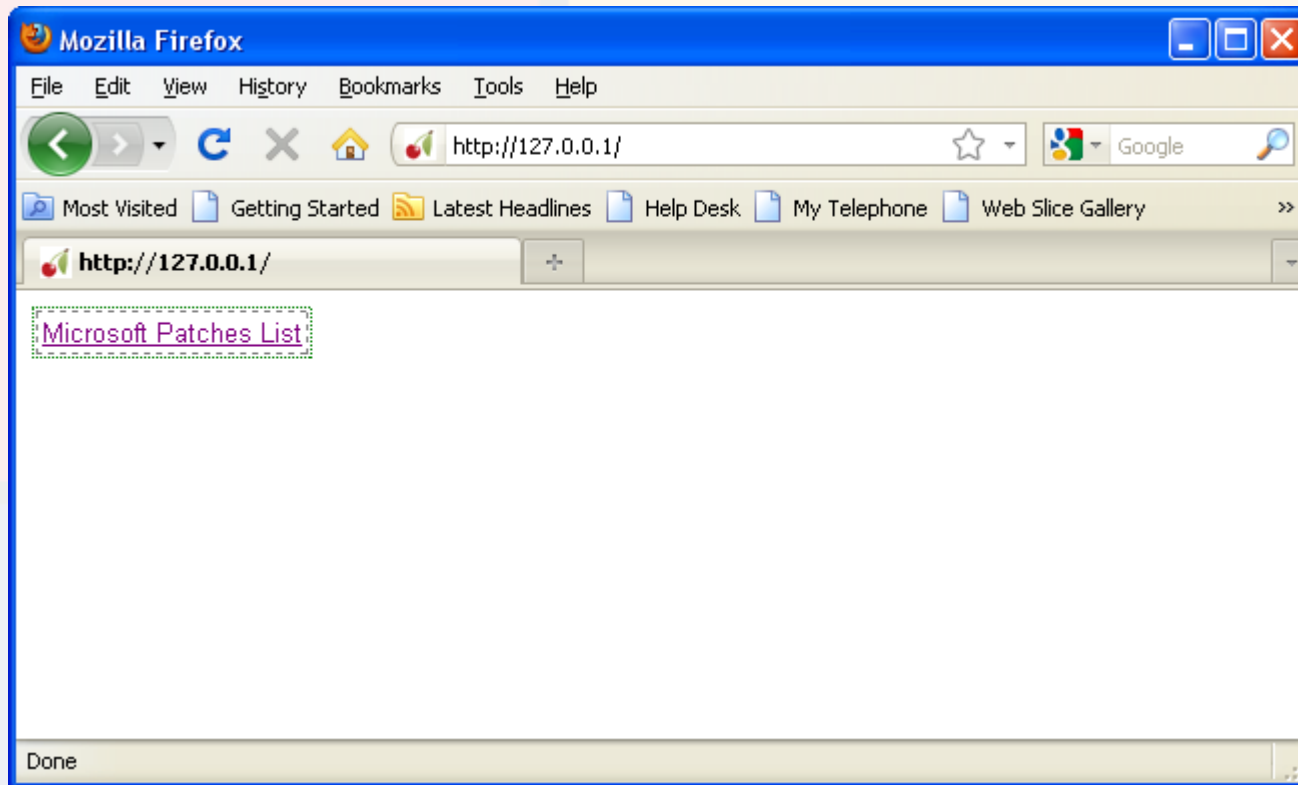
Bin Collector

Collecting Binaries Automagically

- It visits each vendors patch pages
 - Use mechanize python package to scrap MS patch pages
 - Use BeautifulSoup to parse the html pages
- It extracts and archives binary files
 - Use sqlalchemy to index the files
- Use PE version information to determine store location
 - <Company Name>\<File Name>\<Version Name>
- You can make your own archive of binaries in more organized way

Web Console Work Flow

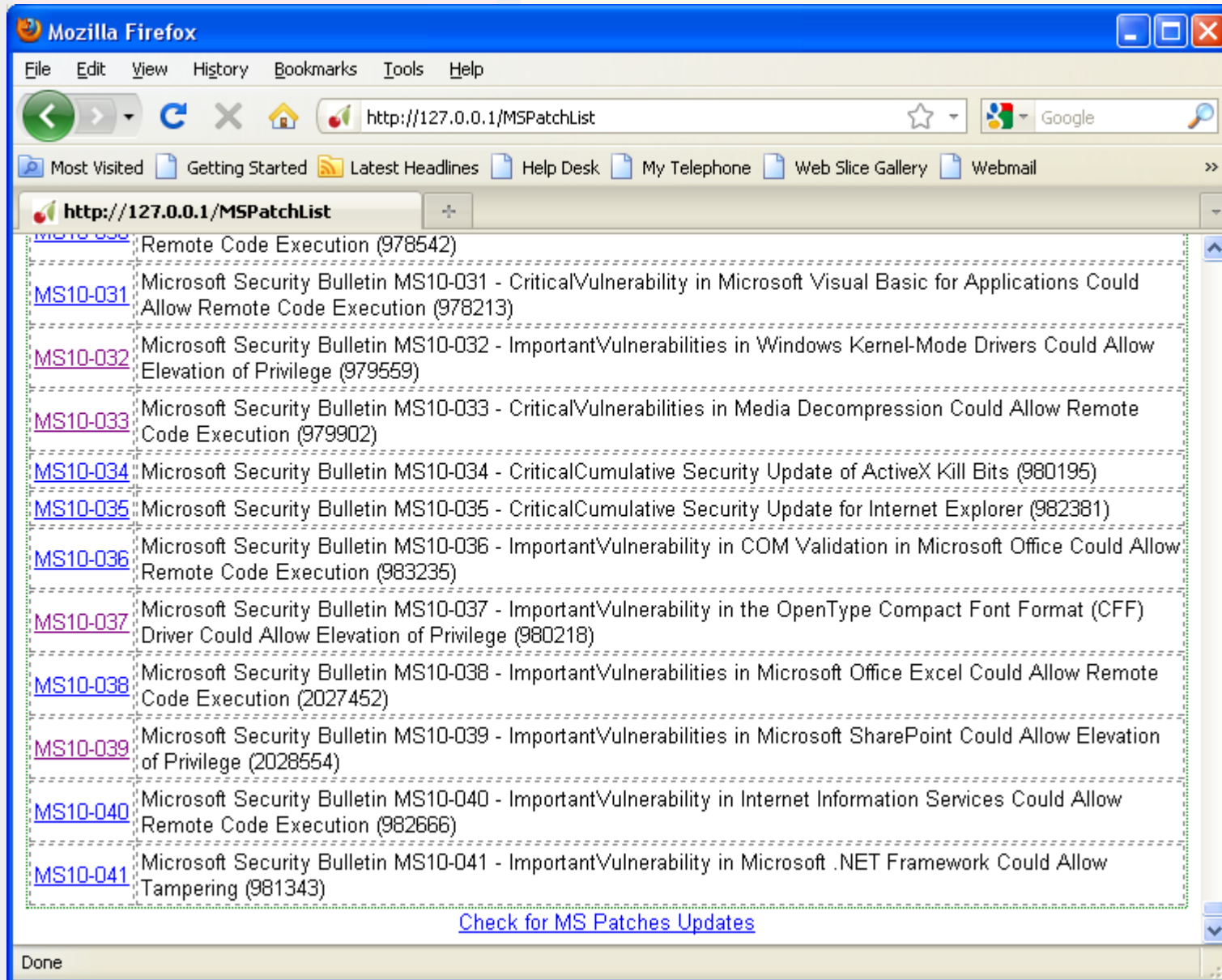
Select Vendor



We only support Microsoft right now.
We are going to support Oracle and Adobe soon.

Web Console Work Flow

Select Patch Name



The screenshot shows a Mozilla Firefox browser window displaying a list of Microsoft Security Bulletins (MS10-031 to MS10-041) on the website <http://127.0.0.1/MSPatchList>. The browser interface includes the menu bar (File, Edit, View, History, Bookmarks, Tools, Help), the address bar, and a search bar. The list of bulletins is as follows:

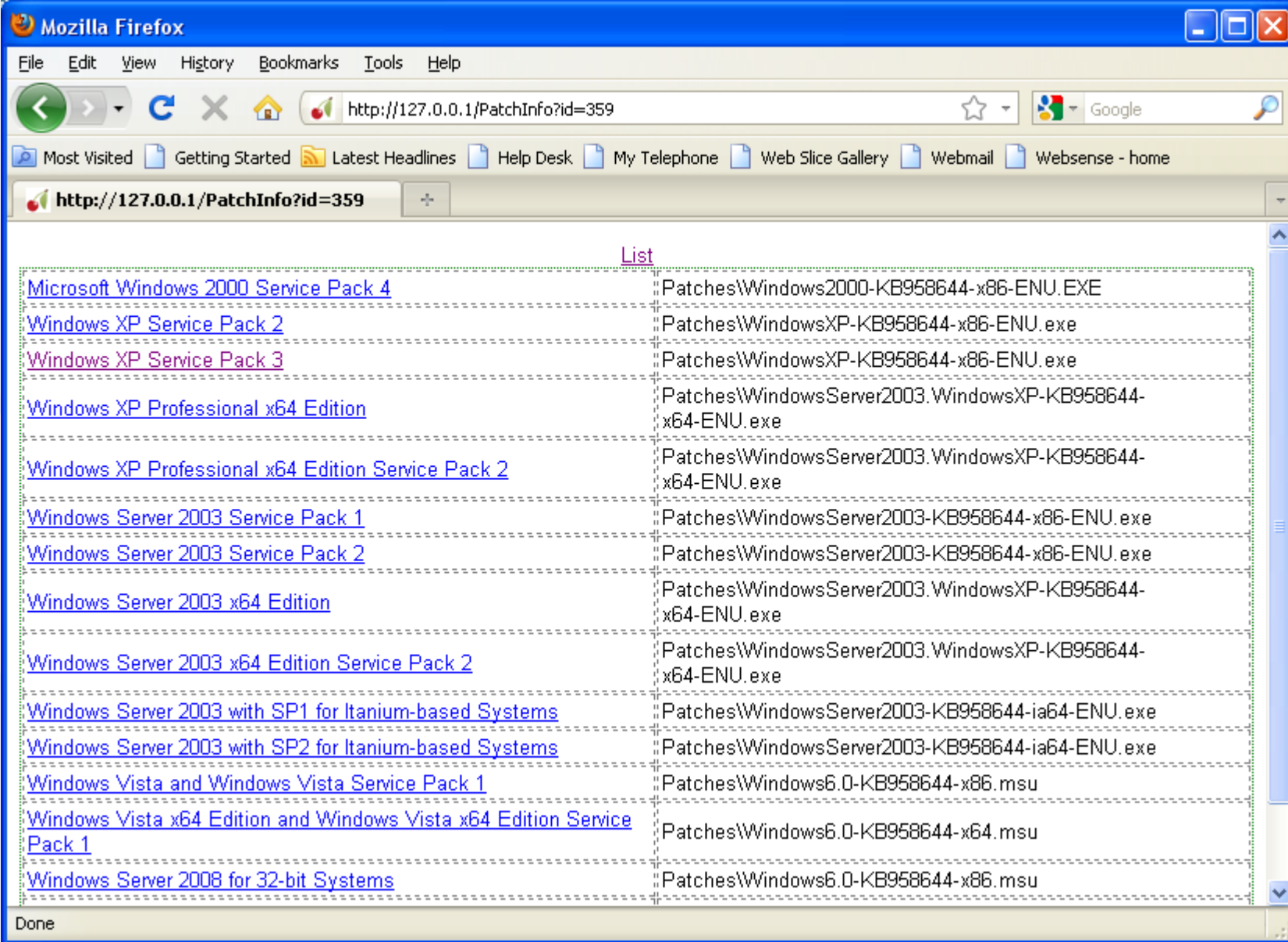
MS10-030	Remote Code Execution (978542)
MS10-031	Microsoft Security Bulletin MS10-031 - CriticalVulnerability in Microsoft Visual Basic for Applications Could Allow Remote Code Execution (978213)
MS10-032	Microsoft Security Bulletin MS10-032 - ImportantVulnerabilities in Windows Kernel-Mode Drivers Could Allow Elevation of Privilege (979559)
MS10-033	Microsoft Security Bulletin MS10-033 - CriticalVulnerabilities in Media Decompression Could Allow Remote Code Execution (979902)
MS10-034	Microsoft Security Bulletin MS10-034 - CriticalCumulative Security Update of ActiveX Kill Bits (980195)
MS10-035	Microsoft Security Bulletin MS10-035 - CriticalCumulative Security Update for Internet Explorer (982381)
MS10-036	Microsoft Security Bulletin MS10-036 - ImportantVulnerability in COM Validation in Microsoft Office Could Allow Remote Code Execution (983235)
MS10-037	Microsoft Security Bulletin MS10-037 - ImportantVulnerability in the OpenType Compact Font Format (CFF) Driver Could Allow Elevation of Privilege (980218)
MS10-038	Microsoft Security Bulletin MS10-038 - ImportantVulnerabilities in Microsoft Office Excel Could Allow Remote Code Execution (2027452)
MS10-039	Microsoft Security Bulletin MS10-039 - ImportantVulnerabilities in Microsoft SharePoint Could Allow Elevation of Privilege (2028554)
MS10-040	Microsoft Security Bulletin MS10-040 - ImportantVulnerability in Internet Information Services Could Allow Remote Code Execution (982666)
MS10-041	Microsoft Security Bulletin MS10-041 - ImportantVulnerability in Microsoft .NET Framework Could Allow Tampering (981343)

[Check for MS Patches Updates](#)

Done

Web Console Work Flow

Select OS

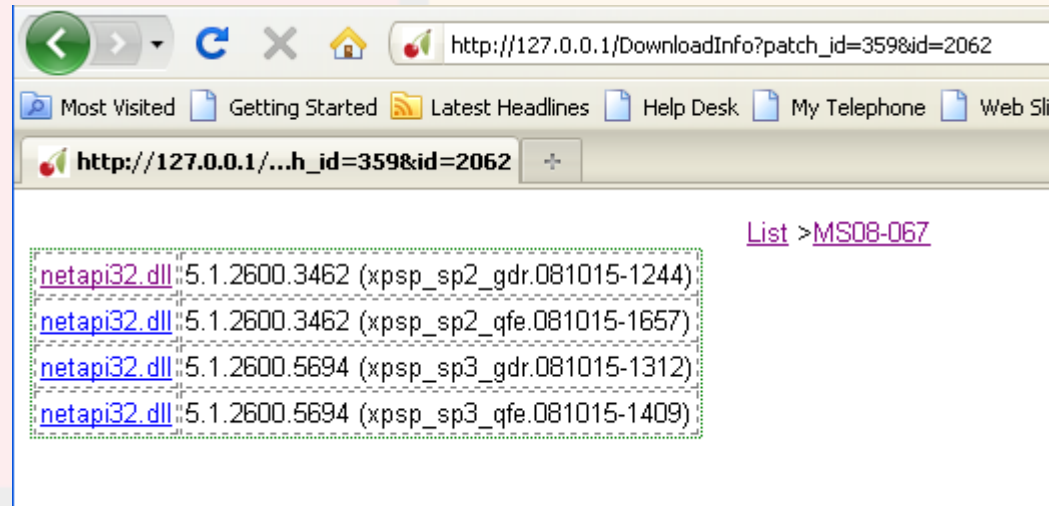


The screenshot shows a Mozilla Firefox browser window with the address bar displaying `http://127.0.0.1/PatchInfo?id=359`. The browser interface includes a menu bar (File, Edit, View, History, Bookmarks, Tools, Help), a search bar with the Google logo, and a toolbar with various icons. The main content area displays a table of Windows patches, with a "List" link above it. The table has two columns: the first column contains links to patch information, and the second column contains the file names of the patches. The status bar at the bottom shows "Done".

List	
Microsoft Windows 2000 Service Pack 4	Patches\Windows2000-KB958644-x86-ENU.EXE
Windows XP Service Pack 2	Patches\WindowsXP-KB958644-x86-ENU.exe
Windows XP Service Pack 3	Patches\WindowsXP-KB958644-x86-ENU.exe
Windows XP Professional x64 Edition	Patches\WindowsServer2003.WindowsXP-KB958644-x64-ENU.exe
Windows XP Professional x64 Edition Service Pack 2	Patches\WindowsServer2003.WindowsXP-KB958644-x64-ENU.exe
Windows Server 2003 Service Pack 1	Patches\WindowsServer2003-KB958644-x86-ENU.exe
Windows Server 2003 Service Pack 2	Patches\WindowsServer2003-KB958644-x86-ENU.exe
Windows Server 2003 x64 Edition	Patches\WindowsServer2003.WindowsXP-KB958644-x64-ENU.exe
Windows Server 2003 x64 Edition Service Pack 2	Patches\WindowsServer2003.WindowsXP-KB958644-x64-ENU.exe
Windows Server 2003 with SP1 for Itanium-based Systems	Patches\WindowsServer2003-KB958644-ia64-ENU.exe
Windows Server 2003 with SP2 for Itanium-based Systems	Patches\WindowsServer2003-KB958644-ia64-ENU.exe
Windows Vista and Windows Vista Service Pack 1	Patches\Windows6.0-KB958644-x86.msu
Windows Vista x64 Edition and Windows Vista x64 Edition Service Pack 1	Patches\Windows6.0-KB958644-x64.msu
Windows Server 2008 for 32-bit Systems	Patches\Windows6.0-KB958644-x86.msu

Web Console Work Flow

Select a File



GDR(General Distribution): a binary marked as GDR contains only security related changes that have been made to the binary

QFE(Quick Fix Engineering)/LDR(Limited Distribution Release): a binary marked as QFE/LDR contains both security related changes that have been made to the binary as well as any functionality changes that have been made to it.

Web Console Work Flow

Initiate Diffing

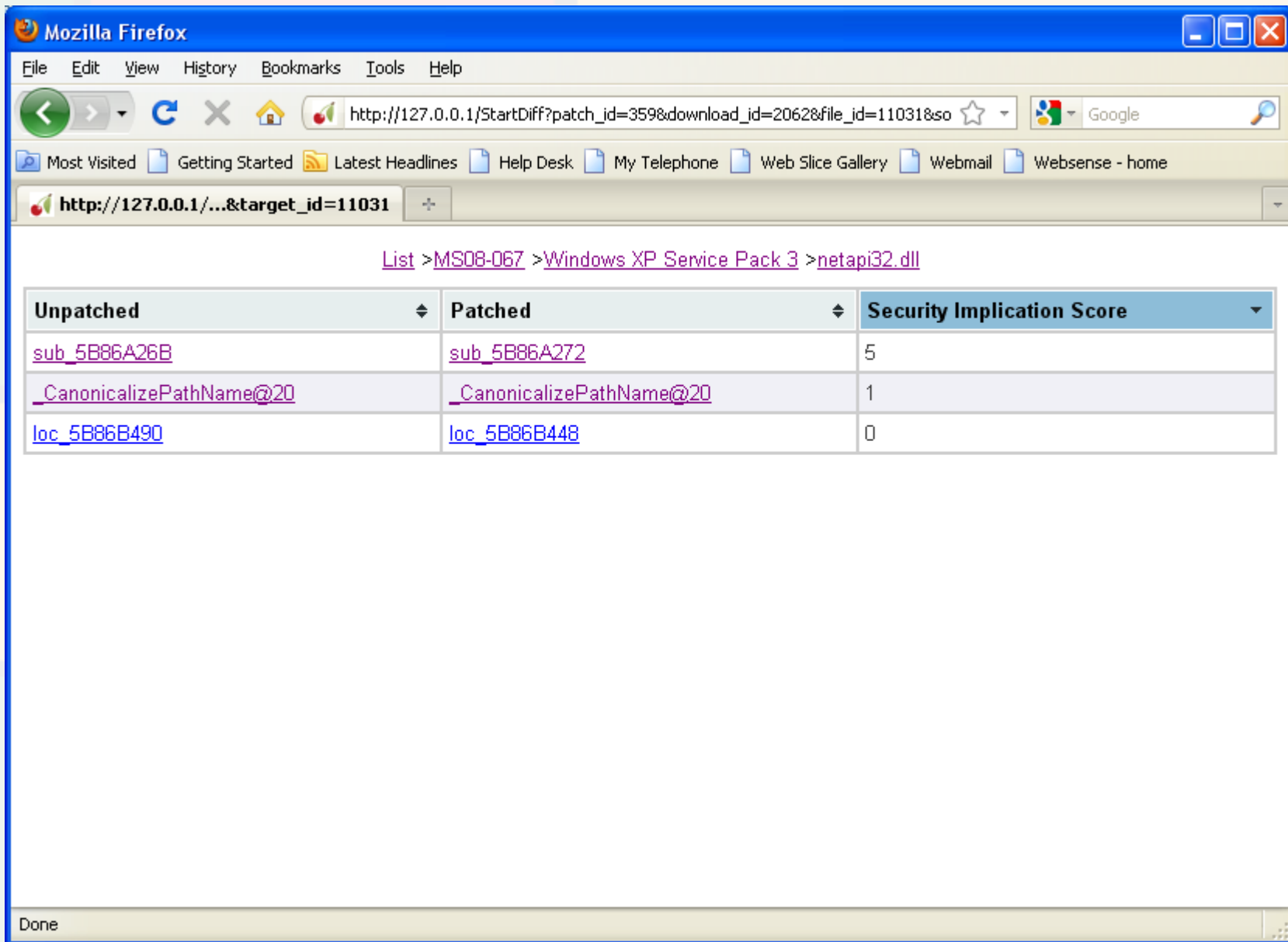
[List](#) > [MS08-067](#) > [Windows XP Service Pack 3](#)

Company Name	Microsoft Corporation
Operating System	xpsp
Service Pack	sp2
Filename	netapi32.dll
Unpatched Filename	MS06-070: T:\mat\Projects\Binaries\Windows XP\Microsoft Corporation\netapi32.dll\5.1.2600.2976 (xpsp_sp2_gdr.060817-0106)\netapi32.dll
Patched Filename	MS08-067: T:\mat\Projects\Binaries\Windows XP\Microsoft Corporation\netapi32.dll\5.1.2600.3462 (xpsp_sp2_gdr.081015-1244)\netapi32.dll

The unpatched file is automatically guessed based on the file name and version string.

Web Console Work Flow

Check the results



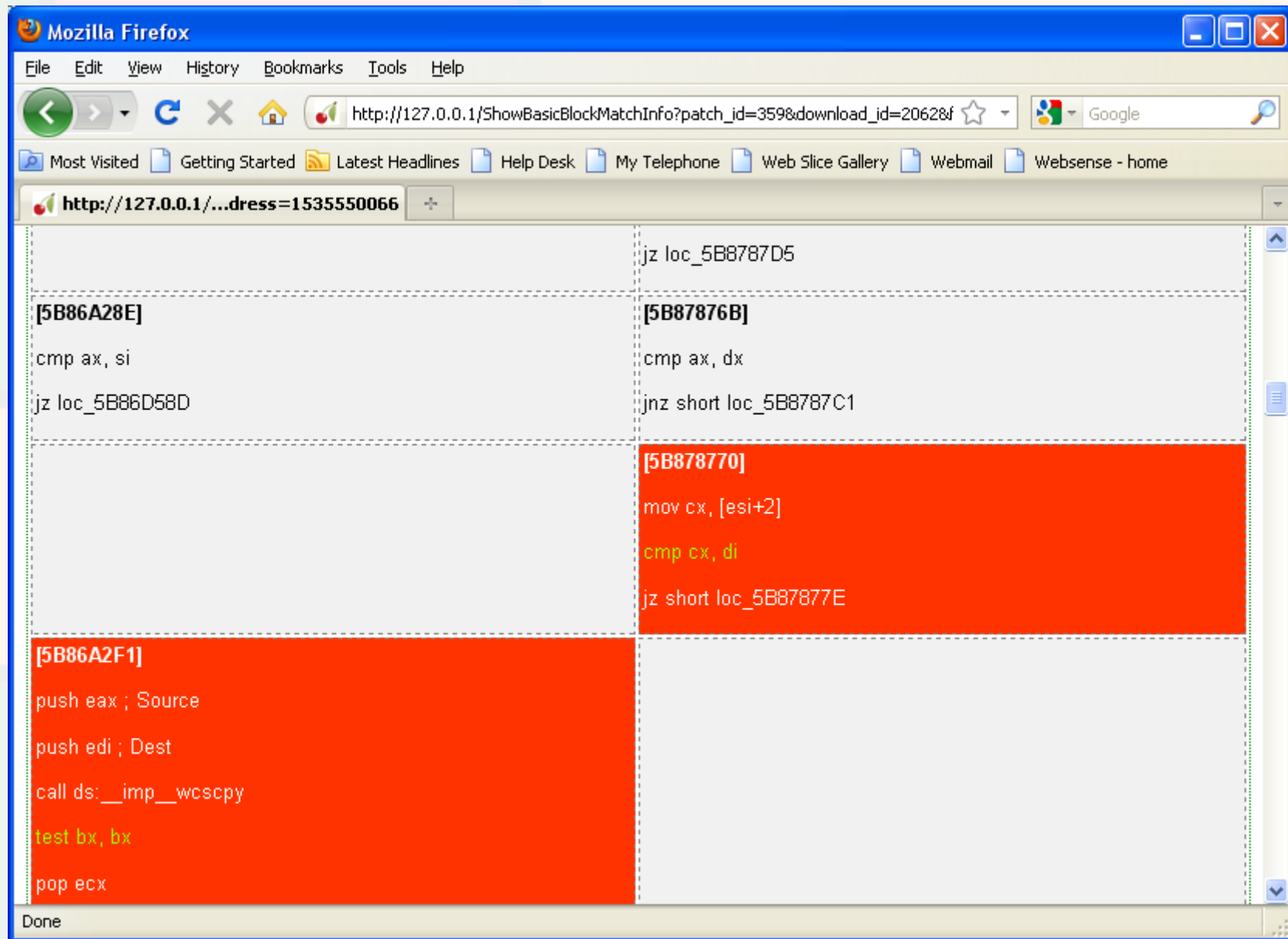
The screenshot shows a Mozilla Firefox browser window displaying a web console results page. The address bar shows the URL: http://127.0.0.1/StartDiff?patch_id=359&download_id=2062&file_id=11031&so. The page content includes a breadcrumb trail: [List](#) > [MS08-067](#) > [Windows XP Service Pack 3](#) > [netapi32.dll](#).

Unpatched	Patched	Security Implication Score
sub_5B86A26B	sub_5B86A272	5
_CanonicalizePathName@20	_CanonicalizePathName@20	1
loc_5B86B490	loc_5B86B448	0

Done

Web Console Work Flow

Check the results



The screenshot shows a Mozilla Firefox browser window displaying assembly code. The address bar shows the URL `http://127.0.0.1/ShowBasicBlockMatchInfo?patch_id=359&download_id=20628f`. The main content area is divided into two columns of assembly instructions. The left column contains instructions for block `[5B86A28E]` and `[5B86A2F1]`. The right column contains instructions for block `[5B87876B]` and `[5B878770]`. The instructions in the right column are highlighted in red, indicating they are the current focus of the debugger. The status bar at the bottom shows "Done".

```
File Edit View History Bookmarks Tools Help
http://127.0.0.1/ShowBasicBlockMatchInfo?patch_id=359&download_id=20628f
Most Visited Getting Started Latest Headlines Help Desk My Telephone Web Slice Gallery Webmail Websense - home
http://127.0.0.1/...dress=1535550066
jz loc_5B8787D5
[5B86A28E] cmp ax, si
jz loc_5B86D58D
[5B87876B] cmp ax, dx
jnz short loc_5B8787C1
[5B878770] mov cx, [esi+2]
cmp cx, di
jz short loc_5B87877E
[5B86A2F1] push eax ; Source
push edi ; Dest
call ds: __imp__wcsncpy
test bx, bx
pop ecx
Done
```

Reading Results

- Locate security patches as quickly as possible
- Sometimes the diff results are not clear because of a lot of noises.
- The noise is caused by
 - Feature updates
 - Code cleanup
 - Refactoring
 - Compiler option change
 - Compiler change

Identifying Security Patches

- Not all patches are security patches
- Sometimes it's like finding needles in the sand
- We need a way for locating patches with strong **security implication**

Identifying Security Patches

Security Implication Score

- DarunGrim 3 provides script interface to the Diffing Engine
- DarunGrim 3 provides basic set of pattern matching
- We calculate Security Implication Score using this Python interface
 - The pattern matching should be easy to extend as the researcher get to know new patterns
 - You can add new patterns if you want.

Examples

- Examples for each vulnerability classes.
- DarunGrim2 and DarunGrim3 examples are shown.
- Security Implication Scores are shown for some examples.

Stack Based Buffer Overflow: MS06-070

List >MS06-070 >Microsoft Windows XP Service Pack 2 —>netapi32.dll

Unpatched	Patched	Security Implication Score
_NetpManagelPCConnect@16	_NetpManagelPCConnect@16	6
sub_5B88F5EB	sub_5B869B96	2

Stack Based Buffer Overflow: MS06-070/_NetpManageIPCCConnect@16

```
cmp word ptr [esi], 5Ch
```

```
push edi
```

```
mov edi, [ebp+Str]
```

```
mov [ebp+var_2B4], eax
```

```
lea eax, [ebp+UseName]
```

```
mov [ebp+ParmError], ebx
```

```
jz short loc_5B885189
```

```
push edi
```

```
mov edi, [ebp+Str]
```

```
push ebx ; Str
```

```
mov [ebp+var_2B8], eax
```

```
lea esi, [ebp+UseName]
```

```
call ds:__imp__wcslen
```

```
cmp eax, 101h
```

```
pop ecx
```

```
jbe short loc_5B885199
```

```
[5B885184]
```

```
push ebx
```

```
push offset aNetpmanageipcc; "NetpManageIPCCConnect: server  
name %ws t"...
```

```
call _NetpLogPrintHelper
```

```
pop ecx
```

```
pop ecx
```

```
push 57h
```

```
pop eax
```

```
jmp loc_5B8853D4
```

Stack Based Buffer Overflow: Signatures

- Pattern matching for string length checking routines is a good sign for stack or heap based overflow.
- There are variations of string length check routines.
 - `strlen`, `wcslen`, `_mbslen`, `_mbstrlen`

Stack Based Buffer Overflow(Logic Error): MS08-067

- Conficker worm exploited this vulnerability to propagate through internal network.
- Easy target for binary diffing
 - only 2 functions changed.
 - One is a change in calling convention.
 - The other is the function that has the vulnerability

Stack Based Buffer Overflow(Logic Error): MS08-067

List >MS08-067 >Windows XP Service Pack 2 >netapi32.dll

Unpatched	Patched	Security Implication Score
sub_5B86A26B	sub_5B86A272	20
_CanonicalizePathName@20	_CanonicalizePathName@20	1
loc_5B86B490	loc_5B86B448	0

Stack Based Buffer Overflow (Logic Error): MS08-067

The screenshot displays a static analysis tool interface. The main area is split into two panels, 'sub_5B86A51B' on the left and 'sub_5B86A51A' on the right, each showing a control flow graph (CFG) with red nodes and black edges. The graphs illustrate the flow of execution, with some nodes highlighted in yellow. Below the graphs is a 'List of Matches' table with the following data:

Original	Patched	Mat...	Type	Fingerprint(Original)	Fingerprint(Patched)	Parent(Original)	Pa
<input type="checkbox"/> 5B878B09	5B878B25	100%	Fingerprint	5c010204027a01020102	5c010204027a01020102	0	0
<input type="checkbox"/> 5B86D865	5B86A561	100%	Tree	2c01022c0102	2c01022c0102	5B86D85C	5B

Stack Based Buffer Overflow(Logic Error): MS08-067

```
List >MS08-067 >Windows XP Service Pack 2 >netapi32.dll >Functions
```

Unpatched: sub_5B86A26B	Patched: sub_5B86A272
[5B86A26B]	[5B86A272]
mov edi, edi	mov edi, edi
push ebp	push ebp
mov ebp, esp	mov ebp, esp
push ecx	sub esp, 0Ch
mov ecx, [ebp+arg_0]	push ebx
mov ax, [ecx]	push esi
push ebx	mov esi, eax
push esi	push edi
push edi	push esi ; Str
xor ebx, ebx	call ds: __imp__wcslen
xor edi, edi	jmp loc_5B878750
cmp ax, 5Ch	pop ecx
push 2Fh	push 2Fh
mov [ebp+var_4], ebx	pop edx
pop esi	lea eax, [esi+eax*2+2]
jz loc_5B86D58D	push 5Ch
	mov [ebp+var_8], eax
	mov ax, [esi]
	xor ebx, ebx
	pop edi
	cmp ax, di
	mov [ebp+pszDest], ebx

Stack Based Buffer Overflow(Logic Error): MS08-067

```
mov [ebp+var_4], esi
jmp loc_5B86A2AE

[5B878800]
push eax ; pszSrc
mov eax, [ebp+var_8]
sub eax, ebx
sar eax, 1
push eax ; cchDest
push ebx ; pszDest
call _StringCchCopyW@12;
StringCchCopyW(x,x,x)
cmp word ptr [ebp+var_C], 0
jz loc_5B869FBC

[5B87881A]
cmp esi, ebx
mov [ebp+pszDest], ebx
```

Stack Based Buffer Overflow(Logic Error): MS08-067

StringCchCopyW

<http://msdn.microsoft.com/en-us/library/ms647527%28VS.85%29.aspx>

Syntax

```
HRESULT StringCchCopy(  
    __out LPTSTR pszDest,  
    __in  size_t  cchDest,  
    __in  LPCTSTR pszSrc  
);
```

Copy

Compared to the functions it replaces, **StringCchCopy** provides additional processing for proper buffer handling in your code. Poor buffer handling is implicated in many security issues that involve buffer overruns. **StringCchCopy** always null-terminates a non-zero-length destination buffer.

Behavior is undefined if the strings pointed to by *pszSrc* and *pszDest* overlap.

Stack Based Buffer Overflow: Signatures

- Pattern matching for safe string manipulation functions are good sign for buffer overflow patches.
 - **Strsafe Functions**
 - StringCbCat, StringCbCatEx, StringCbCatN, StringCbCatNEx, StringCbCopy, StringCbCopyEx, StringCbCopyN, StringCbCopyNEx, StringCbGets, StringCbGetsEx, StringCbLength, StringCbPrintf, StringCbPrintfEx, StringCbVPrintf, StringCbVPrintfEx, StringCchCat, StringCchCatEx, StringCchCatN, StringCchCatNEx, StringCchCopy, StringCchCopyEx, StringCchCopyN, StringCchCopyNEx, StringCchGets, StringCchGetsEx, StringCchLength, StringCchPrintf, StringCchPrintfEx, StringCchVPrintf, StringCchVPrintfEx
 - **Other Safe String Manipulation Functions**
 - strcpy_s, wcsncpy_s, _mbstrcpy_s
 - strcat_s, wscat_s, _mbscat_s
 - strncat_s, _strncat_s_l, wcsncat_s, _wcsncat_s_l, _mbsncat_s, _mbsncat_s_l
 - strncpy_s, _strncpy_s_l, wcsncpy_s, _wcsncpy_s_l, _mbsncpy_s, _mbsncpy_s_l
 - sprintf_s, _sprintf_s_l, swprintf_s, _swprintf_s_l

Stack Based Buffer Overflow: Signatures

- Removal of unsafe string routines is a good signature.
 - strcpy, wcsncpy, _mbncpy
 - strcat, wscat, _mbscat
 - sprintf, _sprintf_l, swprintf, _swprintf_l, __swprintf_l
 - vsprintf, _vsprintf_l, vswprintf, _vswprintf_l, __vswprintf_l
 - vsnprintf, _vsnprintf, _vsnprintf_l, _vsnwprintf, _vsnwprintf_l

Integer Overflow

MS10-030

List > MS10-030 > Microsoft Outlook Express 6 > inetcomm.dll

Unpatched	Patched	Security Implication Score
?RootProps_EndChildren@CHTTPMailTransport@@QAEJXZ	?ContactInfo_EndChildren@CHTTPMailTransport@@QAEJXZ	5
_STR_ATT_COMBINED	_STR_ATT_RENDERED	4
?ResponseSTAT@CPOP3Transport@@AAEXXZ	?ResponseSTAT@CPOP3Transport@@AAEXXZ	4
?ResizeMsgSeqNumTable@Cimap4Agent@@UAGJK@Z	?ResizeMsgSeqNumTable@Cimap4Agent@@UAGJK@Z	4
_STR_ATT_NORMSUBJ	_STR_ATT_RENDERED	3
_STR_ATT_PRIORITY	_STR_ATT_RENDERED	3
?ResponseGenericList@CPOP3Transport@@AAEXXZ	?ResponseGenericList@CPOP3Transport@@AAEXXZ	3
?ProcessTransactTestResponse@CNNTPTTransport@@AAEJXZ	?StartLogon@CNNTPTTransport@@AAEXXZ	3
?GetMsgSeqNumToUIDArray@Cimap4Agent@@UAGJPAPAKPAK@Z	?GetMsgSeqNumToUIDArray@Cimap4Agent@@UAGJPAPAKPAK@Z	3
_STR_ATT_SERVER	_STR_ATT_FORMAT	2
??1CActiveMovie@@@UAE@XZ	??1CBGIImage@@@UAE@XZ	2
?CheckForCompleteResponse@Cimap4Agent@@AAEXPADKPAW4IMAP_RESPONSE_ID@@@Z	?CheckForCompleteResponse@Cimap4Agent@@AAEXPADKPAW4IMAP_RESPONSE_ID@@@Z	2
_STR_ATT_STOREMSGID	_STR_ATT_RENDERED	1
_STR_ATT_FORWARDTO	_STR_ATT_FORMAT	1
?ExclusiveUnlock@CExShareLockWithNestAllowed@@@QAEXXZ	?ExclusiveUnlock@CExShareLock@@@QAEXXZ	1

Integer Overflow

MS10-030

Integer Comparison Routine

```
[7618DCF0]
mov ecx, ebx
shl ecx, 2
lea eax, [esi+584h]
push ecx ; unsigned __int32
lea edi, [esi+580h]
push eax ; void **
mov [edi], ebx
call ?HrAlloc@@@YGJPAPAXK@Z; HrAlloc(void **,ulong)
test eax, eax
mov [ebp+var_10], eax
jl short loc_7618DD36
```

```
[7618DE07]
lea eax, [ebp+var_C]
push eax ; unsigned __int32 *
push 4
pop ecx
mov eax, ebx
mul ecx
push edx
push eax ; unsigned __int64
mov [ebp+var_C], edi
mov [ebp+var_10], edi
call ?ULongLongToULong@@@YGJ_KPAK@Z;
ULongLongToULong(unsigned __int64,ulong *)
cmp eax, edi
mov [ebp+var_14], eax
jl short loc_7618DE68
```

Integer Overflow

MS10-030

[7618DCCE]

```
push [ebp+lpSrc] ; lpSrc
mov edi, ds:__imp__StrToIntA@4; StrToIntA(x)
call edi ; StrToIntA(x); StrToIntA(x)
push [ebp+var_8] ; lpSrc
mov ebx, eax
call edi ; StrToIntA(x); StrToIntA(x)
cmp dword ptr [esi+578h], 0
mov [ebp+var_C], eax
jnz short loc_7618DD2C
```

[7618DDE4]

```
push [ebp+lpSrc] ; lpSrc
mov edi, ds:__imp__StrToIntA@4; StrToIntA(x)
call edi ; StrToIntA(x); StrToIntA(x)
push [ebp+var_8] ; lpSrc
mov ebx, eax
call edi ; StrToIntA(x); StrToIntA(x)
xor edi, edi
cmp [esi+578h], edi
mov [ebp+var_18], eax
jnz short loc_7618DE5E
```


Integer Overflow Signatures

- Additional string to integer conversion functions can be used to check sanity of an integer derived from string.
 - **ULONGLongToULong Function**
 - In case of multiplication operation is done on 32bit integer values, it can overflow. This function can help to see if the overflow happened.
 - **atoi, _atoi_l, _wtoi, _wtoi_l or StrToInt Function** functions might appear on both sides of functions.

Integer Overflow

JRE Font Manager Buffer Overflow(Sun Alert 254571)

T:\mat\Projects\ResearchTools\Binary\StaticAnalysis\DarunGrim2\src\Bin\fontmanager.dll-6.0.10.6-6.0.120.4.dbg

File Graphs Help

sub_6D2C4A60

```

push esi
call sub_6D2C4922
6D2C4A74
push edi
mov edi, [esp+10h]
lea eax, [edi+0Ah]
cmp eax, 2000000h
jnb short loc_6D2C4A8D
6D2C4A8D
xor eax, eax
6D2C4A83
push eax; size_t
call ds:malloc
pop ecx
jmp short loc_6D2C4A8F
6D2C4A8F
test eax, eax
jnz short loc_6D2C4A9A
6D2C4A9A
mov dword ptr [eax], 0AA53C5AAh
mov [eax+4], edi
lea ecx, [eax+edi]
mov byte ptr [ecx+8], 5Ah
mov byte ptr [ecx+9], 0F0h

```

sub_6D244AF2

```

cmp edi, eax
jnb short loc_6D244B2B
6D244B14
lea ecx, [edi+0Ah]
cmp ecx, eax
jnb short loc_6D244B25
6D244B25
xor eax, eax
6D244B1B
push ecx; size_t
call ds:malloc
pop ecx
jmp short loc_6D244B27
6D244B27
test eax, eax
jnz short loc_6D244B32
6D244B32
mov dword ptr [eax], 0AA53C5AAh
mov [eax+4], edi
lea ecx, [eax+edi]
mov byte ptr [ecx+8], 5Ah
mov byte ptr [ecx+9], 0F0h
mov edi, [esi+8]
cmp [esi+4], edi
6D244B2B
push 2718h
jmp short loc_6D244B7A

```

List Of Matches

Original	Patched	Match R...	Type	Fingerprint(Original)	Fingerprint(Patched)	Parent(Original)	Parent(Patched)
<input type="checkbox"/>	6D244B27				cc201020102		
<input type="checkbox"/>	6D244B1B				cc8f0102cc120202cc860...		
<input type="checkbox"/>	6D244B25				cc801020102		
<input type="checkbox"/>	6D2C4AD9	100%	Tree	cc1b01020102	cc1b01020102	6D2C4AD3	6D244B68
<input type="checkbox"/>	6D2C4ACC	100%	Tree	cc2c0102cc1b01020102	cc2c0102cc1b01020102	6D2C4AC6	6D244B5E
<input type="checkbox"/>	6D2C4AD3	100%	Fingerprint	cc7a03020102cc2c0402	cc7a03020102cc2c0402	0	0
<input type="checkbox"/>	6D2C4ADD	100%	Tree	cc8f0502	cc8f0502	6D2C4ABD	6D244B55
<input type="checkbox"/>	6D2C4AC6	100%	Tree	cc1b03020502	cc1b03020502	6D2C4ABD	6D244B55
<input type="checkbox"/>	6D2C4AE8	100%	Fingerprint	cc860102cc0601020502c...	cc860102cc0601020502c...	0	0
<input type="checkbox"/>	6D2C4ABD	100%	Fingerprint	cc7a01020402ccd801020...	cc7a01020402ccd801020...	0	0
<input type="checkbox"/>	6D2C4AB6	100%	Tree	cc8f0502	cc8f0502	6D2C4A9A	6D244B32
<input type="checkbox"/>	6D2C4AE2	100%	Tree	cc8f0102cc100702	cc8f0102cc100702	6D2C4AB6	6D244B4E
<input type="checkbox"/>	6D2C4A9A	100%	Fingerprint	cc7a02020502cc7a020...	cc7a02020502cc7a020...	0	0

Integer Overflow

JRE Font Manager Buffer Overflow(Sun Alert 254571)



Original

```
.text:6D2C4A75      mov     edi, [esp+10h]
.text:6D2C4A79      lea    eax, [edi+0Ah]
.text:6D2C4A7C      cmp    eax, 2000000h
.text:6D2C4A81      jnb   short loc_6D2C4A8D

.text:6D2C4A83      push   eax          ; size_t
.text:6D2C4A84      call  ds:malloc
```

Patched

```
.text:6D244B06      push   edi

Additional Check:
.text:6D244B07      mov    edi, [esp+10h]
.text:6D244B0B      mov    eax, 2000000h
.text:6D244B10      cmp    edi, eax
.text:6D244B12      jnb   short loc_6D244B2B

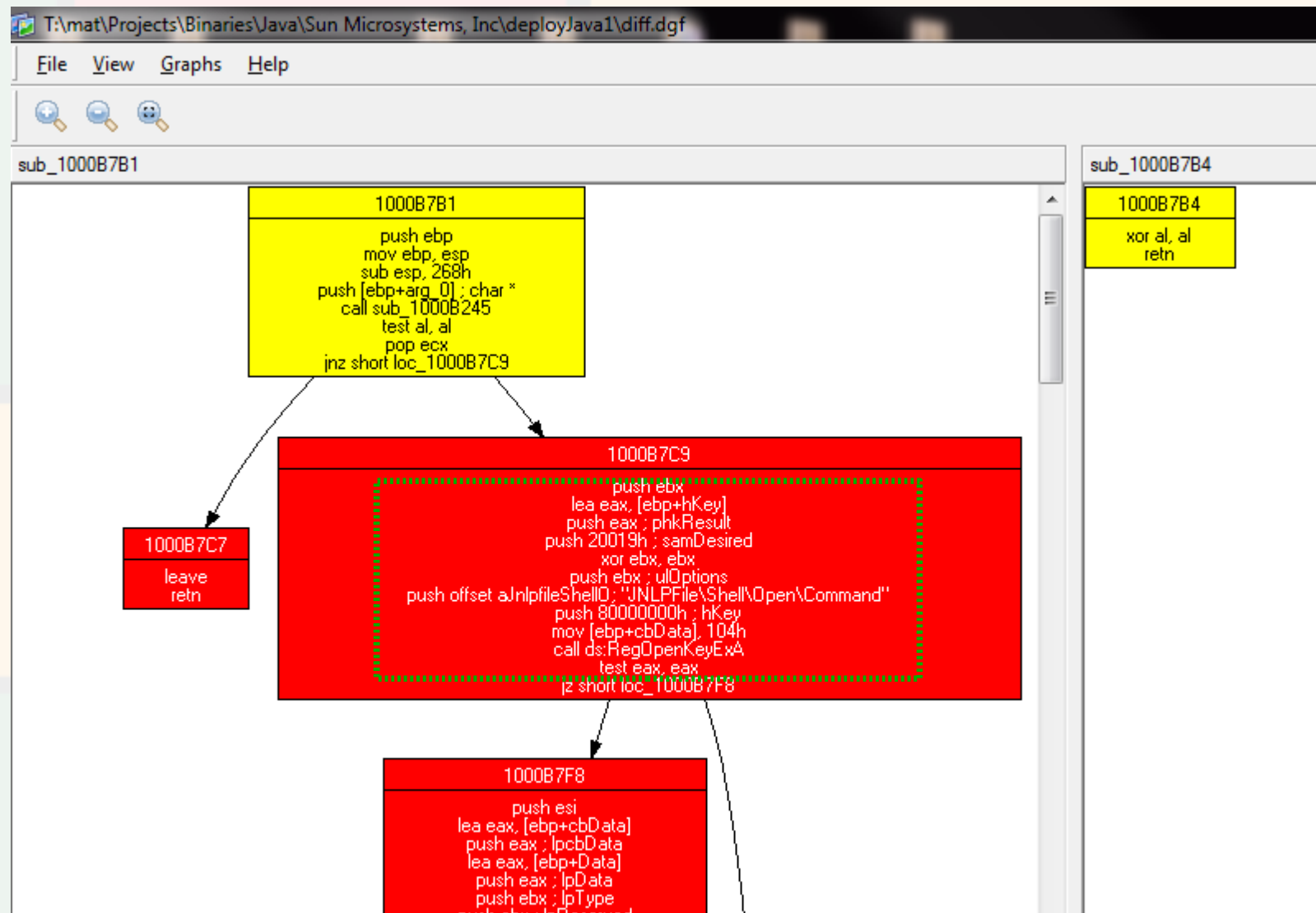
.text:6D244B14      lea   ecx, [edi+0Ah]
.text:6D244B17      cmp   ecx, eax
.text:6D244B19      jnb   short loc_6D244B25

.text:6D244B1B      push   ecx          ; size_t
.text:6D244B1C      call  ds:malloc
```

Integer Overflow Signatures

- Additional `cmp` x86 operation is a good sign of integer overflow check.
 - It will perform additional range check for the integer before and after of the arithmetic operation
 - Counting additional number of "`cmp`" instruction in patched function might help deciding integer overflow.

Insufficient Validation of Parameters Java Deployment Toolkit



Insufficient Validation of Parameters

Java Deployment Toolkit

- Unpatched one has whole a lot of red and yellow blocks.
 - The whole function's basic blocks have been removed.
 - This is the quick fix for @taviso's 0-day.
- The function is responsible for querying registry key for JNLPFile Shell Open key and launching it using CreateProcessA API.

Insufficient Validation of Parameters Signatures

- If validation of parameters are related to process creation routine, we can check if the original or patched function has a process creation related APIs like **CreateProcess Function** in modified functions.

Invalid Argument

MS09-020:WebDav case

?ScConvertToWide@@YJPDPA... 0 ?ScConvertToWide@@YJPDPA... 0 10 16 80%

?ScConvertToWide@@YJPDPAIPAG0H@Z

```
lea ebx, [eax+1]
mov eax, [ebp-12Ch]
push dword ptr [eax]; cchWideChar
mov eax, [ebp-124h]
push dword ptr [ebp-130h]; lpWideCharStr
sub eax, esi
push ebx; cchMultiByte
push dword ptr [ebp-128h]; lpMultiByteStr
neg eax
sbb eax, eax
and eax, 8
push eax; dwFlags
push dword ptr [ebp-124h]; CodePage
call edi; MultiByteToWideChar(x,x,x,x,x,x); MultiByteToWideChar(
test eax, eax
jnz short loc_6F069752
```

6F0696F2

```
call ds:__imp__GetLastError@0; GetLastError()
cmp eax, 7Ah
```

Patched

Original

6F0696B9

```
push dword ptr [ebx]; cchWideChar
mov esi, ds:__imp__MultiByteToWideChar@24; MultiByteToWideChar(x,x,x,x,x,x)
push dword ptr [ebp-12Ch]; lpWideCharStr
sub eax, ecx
lea edi, [eax+1]
push edi; cchMultiByte
push dword ptr [ebp-124h]; lpMultiByteStr
push 8; dwFlags
push dword ptr [ebp-128h]; CodePage
call esi; MultiByteToWideChar(x,x,x,x,x,x); MultiByteToWideChar(x,x,x,x,x,x)
test eax, eax
jnz short loc_6F069739
```

6F0696E1

```
mov ebx, ds:__imp__GetLastError@0; GetLastError()
call ebx; GetLastError(); GetLastError()
cmp eax, 7Ah
jnz short loc_6F069707
```


Invalid Argument

MS09-020:WebDav case

Flags has changed

Original

```
lea ebx, [eax*4]
mov eax, [ebp-12Ch]
push dword ptr [eax]; cchWideChar
mov eax, [ebp-124h]
push dword ptr [ebp-130h]; lpWideCharStr
sub eax, esi
push ebx; cchMultiByte
push dword ptr [ebp-128h]; lpMultiByteStr
neg eax
sbb eax, eax
and eax, 8
push eax; dwFlags
push dword ptr [ebp-124h]; CodePage
call edi; MultiByteToWideChar(x,x,x,x,x); MultiByteToWideChar(
```

Patched

```
push dword ptr [ebx]; cchWideChar
mov esi, ds: __imp__MultiByteToWideChar@24; MultiByteToWideChar(x,x,x,x,x)
push dword ptr [ebp-12Ch]; lpWideCharStr
sub eax, ecx
lea edi, [eax+1]
push edi; cchMultiByte
push dword ptr [ebp-124h]; lpMultiByteStr
push 8; dwFlags
push dword ptr [ebp-128h]; CodePage
call esi; MultiByteToWideChar(x,x,x,x,x); MultiByteToWideChar(x,x,x,x,x)
```

Invalid Argument

MS09-020:WebDav case

What does flag 8 mean?

MSDN([http://msdn.microsoft.com/en-us/library/dd319072\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/dd319072(VS.85).aspx)) declares like following:

MB_ERR_INVALID_CHARS

Windows Vista and later: The function does not drop illegal code points if the application does not set this flag.

Windows 2000 Service Pack 4, Windows XP: Fail if an invalid input character is encountered. **If this flag is not set, the function silently drops illegal code points.** A call to GetLastError returns ERROR_NO_UNICODE_TRANSLATION.

Invalid Argument

MS09-020:WebDav case

Broken UTF8 Heuristics?

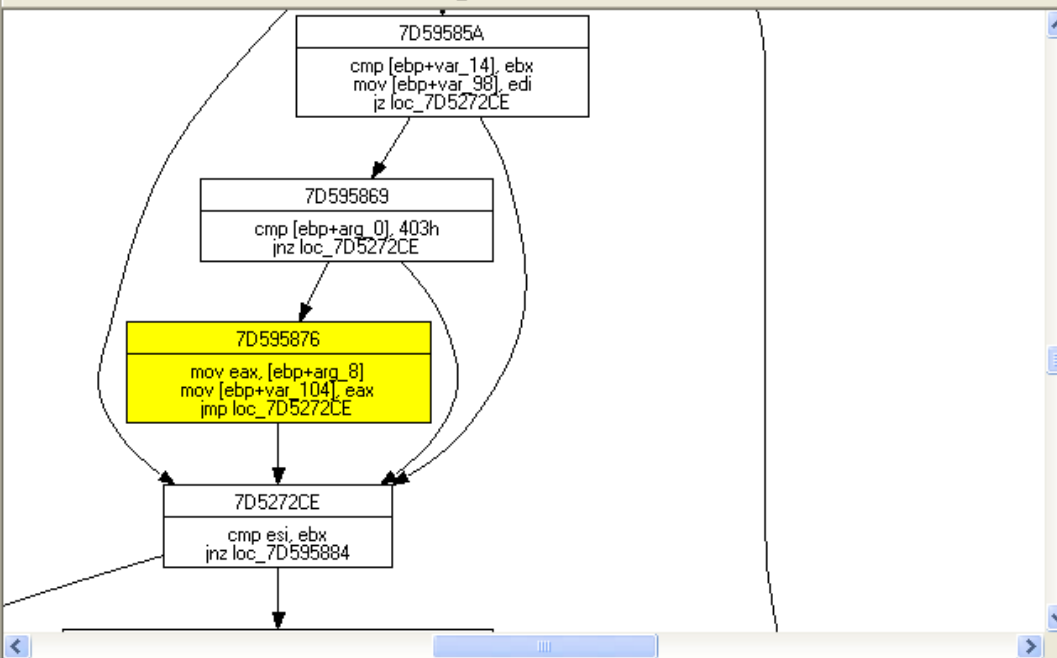
```
6F0695EA mov    esi, 0FDE9h
''''
6F069641 call   ?FlsUTF8Url@@YIHPBD@Z ;
FlsUTF8Url(char const *)
6F069646 test   eax, eax
if(!eax)
{
    6F0695C3 xor    edi, edi
    6F06964A mov    [ebp-124h], edi
}else
{
    6F069650 cmp    [ebp-124h], esi
}
...
6F0696C9 mov    eax, [ebp-124h]
6F0696D5 sub    eax, esi
6F0696DE neg    eax
6F0696E0 sbb    eax, eax
6F0696E2 and    eax, 8
```

Insufficient Validation of Parameters Signatures

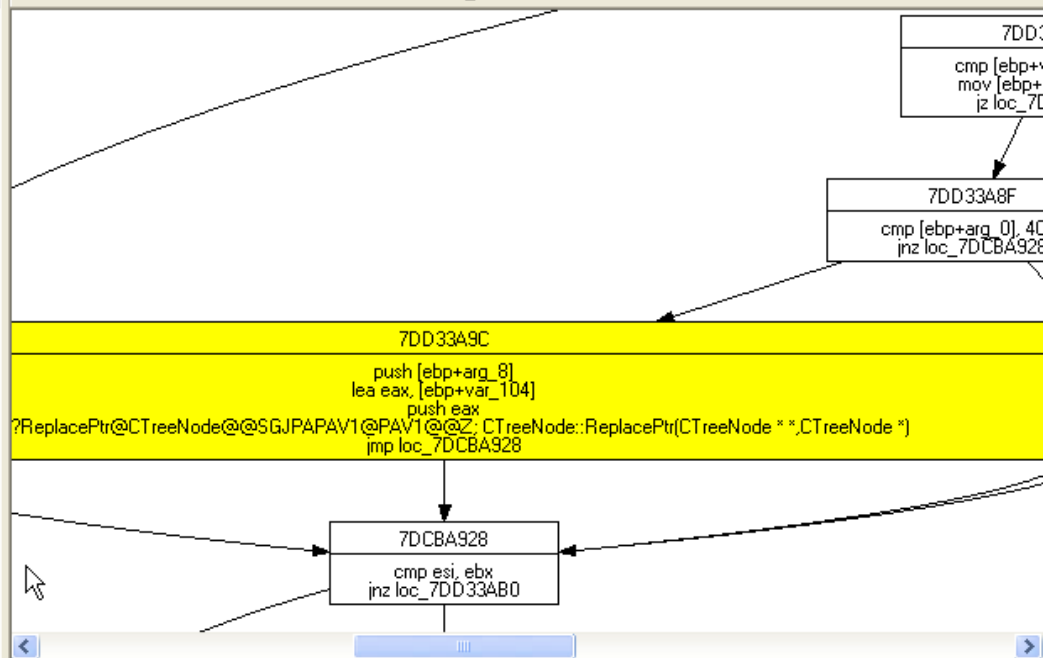
- This issue is related to string conversion routine like **MultiByteToWideChar Function**, we can check if the modified or inserted, removed blocks have these kinds of APIs used in it.
 - If the pattern is found, it's a strong sign of invalid parameter checks.

Use-After-Free: CVE-2010-0249-Vulnerability in Internet Explorer Could Allow Remote Code Execution

?FireEvent@CElement@@QAEJPBUPROPERTYDESC_BASIC@@HPAVCTreeNode@@@JPAUEVENTINFO@@@H

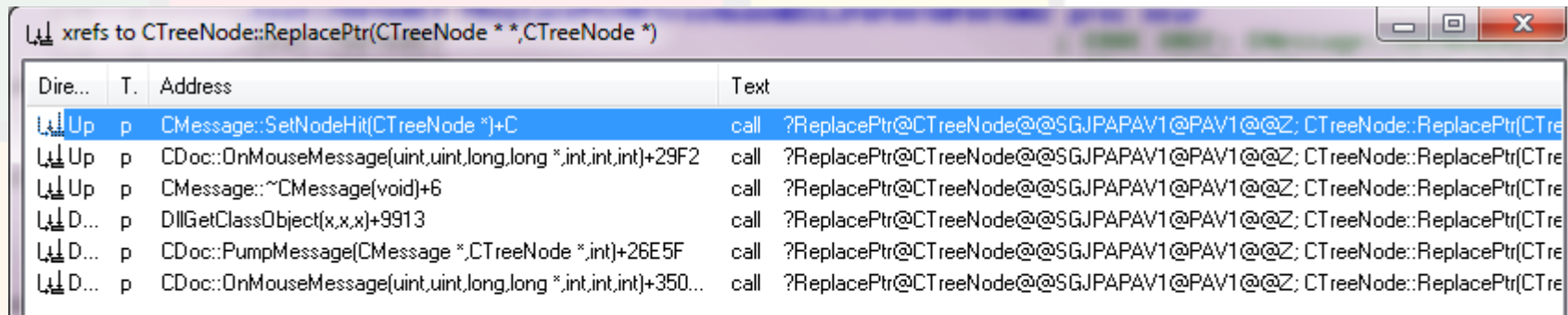


?FireEvent@CElement@@QAEJPBUPROPERTYDESC_BASIC@@HPAVCTreeNode@@@JPAUEVENTINFO@@@H



Use-After-Free: CVE-2010-0249-Vulnerability in Internet Explorer Could Allow Remote Code Execution

Unpatched



The screenshot shows a debugger window titled "xrefs to CTreeNode::ReplacePtr(CTreeNode *,CTreeNode *)". The window displays a call stack with the following entries:

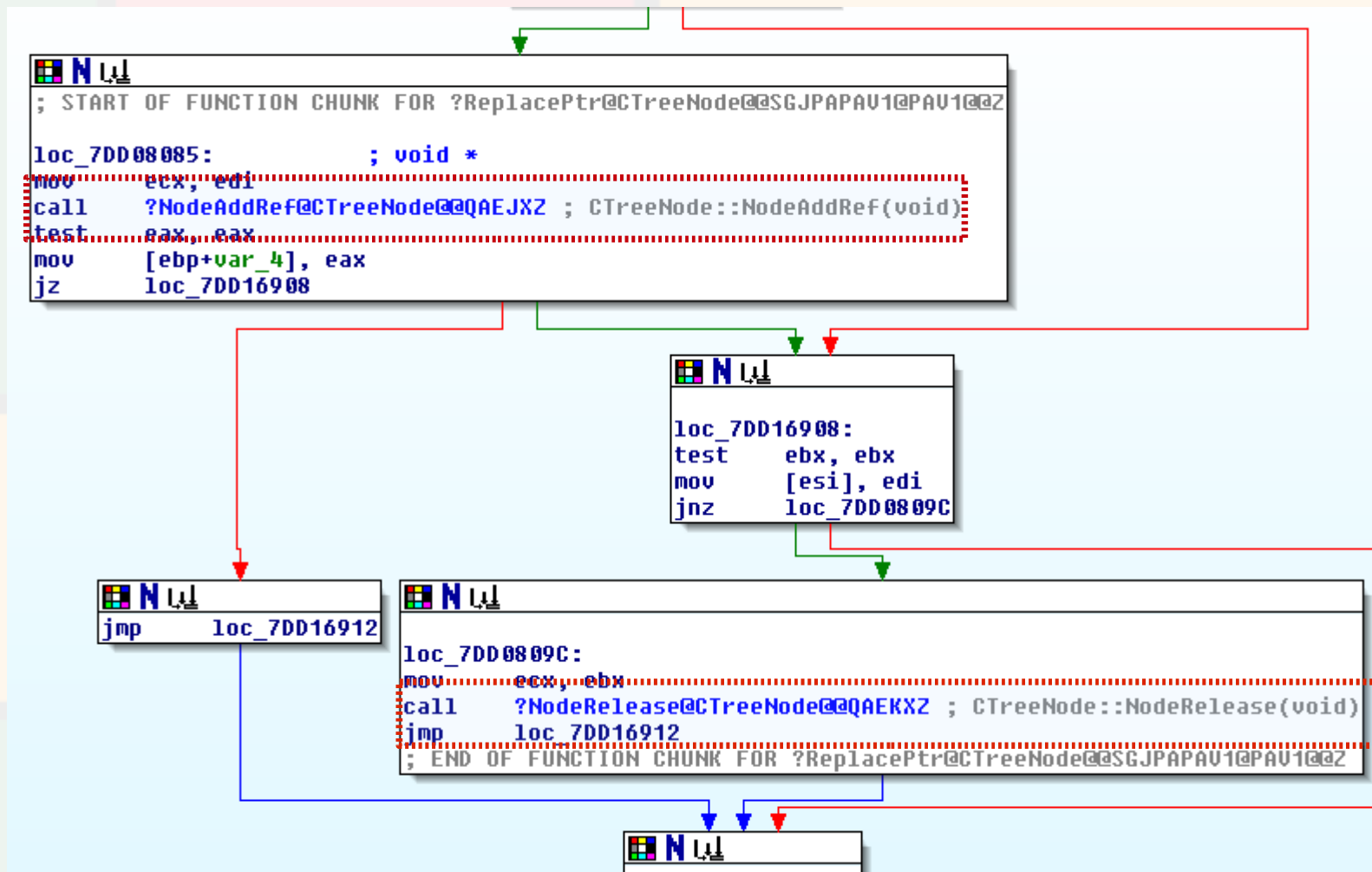
Dir...	T.	Address	Text
Up	p	CMessage::SetNodeHit(CTreeNode *)+C	call ?ReplacePtr@CTreeNode@@SGJPAPAV1@PAV1@@@Z; CTreeNode::ReplacePtr(CTreeNode *,CTreeNode *)
Up	p	CDoc::OnMouseMessage(uint,uint,long,long *,int,int,int)+29F2	call ?ReplacePtr@CTreeNode@@SGJPAPAV1@PAV1@@@Z; CTreeNode::ReplacePtr(CTreeNode *,CTreeNode *)
Up	p	CMessage::~CMessage(void)+6	call ?ReplacePtr@CTreeNode@@SGJPAPAV1@PAV1@@@Z; CTreeNode::ReplacePtr(CTreeNode *,CTreeNode *)
D...	p	DllGetObject(x,x,x)+9913	call ?ReplacePtr@CTreeNode@@SGJPAPAV1@PAV1@@@Z; CTreeNode::ReplacePtr(CTreeNode *,CTreeNode *)
D...	p	CDoc::PumpMessage(CMessage *,CTreeNode *,int)+26E5F	call ?ReplacePtr@CTreeNode@@SGJPAPAV1@PAV1@@@Z; CTreeNode::ReplacePtr(CTreeNode *,CTreeNode *)
D...	p	CDoc::OnMouseMessage(uint,uint,long,long *,int,int,int)+350...	call ?ReplacePtr@CTreeNode@@SGJPAPAV1@PAV1@@@Z; CTreeNode::ReplacePtr(CTreeNode *,CTreeNode *)

Use-After-Free: CVE-2010-0249-Vulnerability in Internet Explorer Could Allow Remote Code Execution

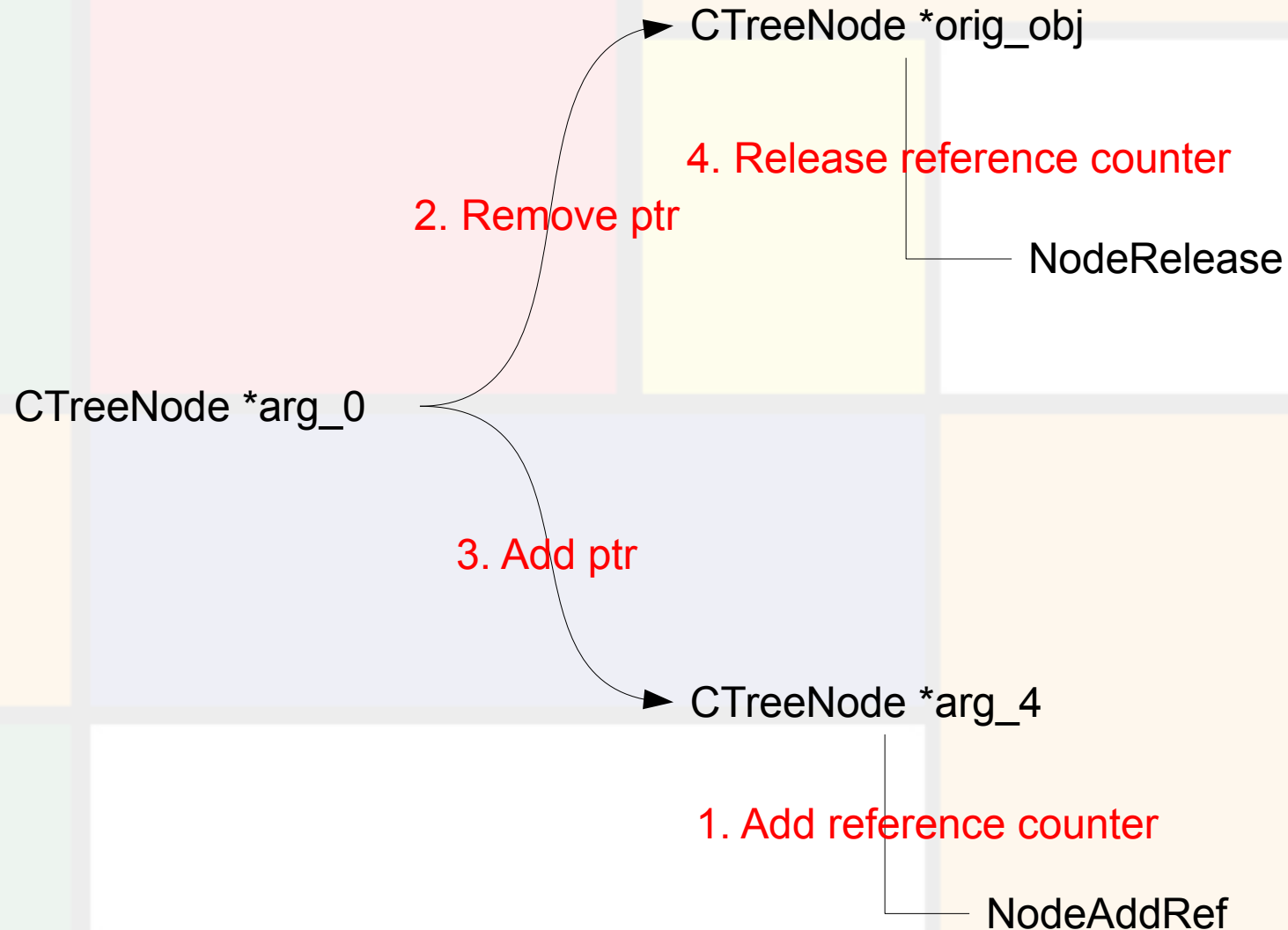
Patched

Dire...	T.	Address	Text
...	p	CDoc::HandleSelectionMessage(CMessage *,int,EVENTINFO *,HM_TYPE)+...	call ?ReplacePtr@CTreeNode@@SGJPAPAV1@PAV1@@@Z; CTreeNode::ReplacePtr(C...
...	p	CDoc::OnMouseMessage(uint,uint,long,long *,int,int,int)+29EE	call ?ReplacePtr@CTreeNode@@SGJPAPAV1@PAV1@@@Z; CTreeNode::ReplacePtr(C...
...	p	CDoc::OnMouseMessage(uint,uint,long,long *,int,int,int)+352FD	call ?ReplacePtr@CTreeNode@@SGJPAPAV1@PAV1@@@Z; CTreeNode::ReplacePtr(C...
...	p	CDoc::PumpMessage(CMessage *,CTreeNode *,int)+270B9	call ?ReplacePtr@CTreeNode@@SGJPAPAV1@PAV1@@@Z; CTreeNode::ReplacePtr(C...
...	p	CElement::BubbleEventHelper(CTreeNode *,long,long,long,int,int *)+78F25	call ?ReplacePtr@CTreeNode@@SGJPAPAV1@PAV1@@@Z; CTreeNode::ReplacePtr(C...
...	p	CElement::BubbleEventHelper(CTreeNode *,long,long,long,int,int *)+78F5B	call ?ReplacePtr@CTreeNode@@SGJPAPAV1@PAV1@@@Z; CTreeNode::ReplacePtr(C...
...	p	CElement::FireEvent(PROPERTYDESC_BASIC const *,int,CTreeNode *,long,...	call ?ReplacePtr@CTreeNode@@SGJPAPAV1@PAV1@@@Z; CTreeNode::ReplacePtr(C...
...	p	CElement::FireEventMouseEnterLeave(CTreeNode *,CMessage *,short,short,I...	call ?ReplacePtr@CTreeNode@@SGJPAPAV1@PAV1@@@Z; CTreeNode::ReplacePtr(C...
...	p	CElement::FireEventMouseEnterLeave(CTreeNode *,CMessage *,short,short,I...	call ?ReplacePtr@CTreeNode@@SGJPAPAV1@PAV1@@@Z; CTreeNode::ReplacePtr(C...
...	p	CElement::FireStdEvent_MouseHelper(CTreeNode *,CMessage *,short,short,I...	call ?ReplacePtr@CTreeNode@@SGJPAPAV1@PAV1@@@Z; CTreeNode::ReplacePtr(C...
...	p	CElement::FireStdEvent_MouseHelper(CTreeNode *,CMessage *,short,short,I...	call ?ReplacePtr@CTreeNode@@SGJPAPAV1@PAV1@@@Z; CTreeNode::ReplacePtr(C...
...	p	CElement::Fire_ActivationHelper(long,CElement *,long,int,int,int,EVENTINFO *...	call ?ReplacePtr@CTreeNode@@SGJPAPAV1@PAV1@@@Z; CTreeNode::ReplacePtr(C...
...	p	CElement::Fire_ActivationHelper(long,CElement *,long,int,int,int,EVENTINFO *...	call ?ReplacePtr@CTreeNode@@SGJPAPAV1@PAV1@@@Z; CTreeNode::ReplacePtr(C...
...	p	CElement::Fire_ActivationHelper(long,CElement *,long,int,int,int,EVENTINFO *...	call ?ReplacePtr@CTreeNode@@SGJPAPAV1@PAV1@@@Z; CTreeNode::ReplacePtr(C...
...	p	CElement::Fire_ActivationHelper(long,CElement *,long,int,int,int,EVENTINFO *...	call ?ReplacePtr@CTreeNode@@SGJPAPAV1@PAV1@@@Z; CTreeNode::ReplacePtr(C...
...	p	CElement::Fire_ActivationHelper(long,CElement *,long,int,int,int,EVENTINFO *...	call ?ReplacePtr@CTreeNode@@SGJPAPAV1@PAV1@@@Z; CTreeNode::ReplacePtr(C...
...	p	CElement::Fire_ActivationHelper(long,CElement *,long,int,int,int,EVENTINFO *...	call ?ReplacePtr@CTreeNode@@SGJPAPAV1@PAV1@@@Z; CTreeNode::ReplacePtr(C...
...	p	CElement::Fire_onlayoutcomplete(int,ulong)+32	call ?ReplacePtr@CTreeNode@@SGJPAPAV1@PAV1@@@Z; CTreeNode::ReplacePtr(C...
...	p	CElement::fireEvent(ushort *,tagVARIANT *,short *)+131	call ?ReplacePtr@CTreeNode@@SGJPAPAV1@PAV1@@@Z; CTreeNode::ReplacePtr(C...
...	p	CElement::fireEvent(ushort *,tagVARIANT *,short *)+152	call ?ReplacePtr@CTreeNode@@SGJPAPAV1@PAV1@@@Z; CTreeNode::ReplacePtr(C...

Use-After-Free: CVE-2010-0249-Vulnerability in Internet Explorer Could Allow Remote Code Execution



Use-After-Free: CVE-2010-0249-Vulnerability in Internet Explorer Could Allow Remote Code Execution



Use-After-Free: CVE-2010-0249-Vulnerability in Internet Explorer Could Allow Remote Code Execution Signatures

- Original binary was missing to replace pointer for the tree node.
 - Freed node was used accidentally.
 - ReplacePtr in adequate places fixed the problem
- We might use ReplacePtr pattern for use-after-free bug in IE.
 - Adding the pattern will help to find same issue later binary diffing.

Conclusion

- Binary Diffing can benefit IPS rule writers and security researchers
 - Locating security vulnerabilities from binary can help further binary auditing
 - There are typical patterns in patches according to their bug classes.
 - **Security Implication Score** by DarunGrim3 helps finding security patches out from feature updates
 - The **Security Implication Score** logic is written in Python and customizable on-demand.



Questions?