HIGH-TECH BRIDGE®

INFORMATION SECURITY SOLUTIONS

CVE 2012-1889 Security Update Analysis

19TH July 2012

Brian MARIANI & Frédéric BOURLA

Timeline

- The 12th of June 2012 Microsoft published a security advisory with a temporary fix related to the msxml core services vulnerability which is heavily exploited in the wild.
- On June 18th 2012 Metasploit released a working exploit.
- On June 19th 2012 a 100% reliable exploit for Internet Explorer 6/7/8/9 on Windows XP/Vista, and Windows 7 SP1 was published by metasploit.
- On July 9th 2012 Microsoft finally released a security update in order to patch this vulnerability.

Some important details

- This document is the continuation of the previous publication: "Microsoft XML core services uninitialized memory vulnerability".
- In this new presentation we will analyze the security update released on July 9th 2012 which fixes several DLL libraries, specially the **msxml3.dll** one.
- The lab environment is an **English Windows XP SP3 workstation.**
- For simplicity, **ASLR** and **DEP** security options are **deactivated.**

MS12-043: Description of the security update for XML Core Services 3.0: July 10, 2012

Article ID: 2719985

View products that this article applies to.

Applies to

This article applies to the following:

- Microsoft XML Core Services 3.0 when used with:
  - Windows 7
  - Windows 7 Service Pack 1
  - Windows Server 2008 R2
  - Windows Server 2008 R2 Service Pack 1
  - Windows Server 2008 Service Pack 2
  - Windows Vista Service Pack 2
  - Windows Server 2008 Service Pack 2
  - Windows XP Service Pack 3
  - Windows XP Professional x64 Edition Service Pack 2
  - Windows Server 2003 Service Pack 2

Files' size comparison

- We identify all files implied in the security update process with monitoring tools, such as Process Monitor. Actually, the file which interests us is the **msxml3.dll** library.

- To successfully compare unpatched and patched files, we first make a copy of the unpatched library to an analysis directory.

- We apply the security update and we copy again the patched DLL file into the previous directory, with a new destination file name.

- After downloading and applying the security update and comparing the size of this particular file, we can notice a **tiny difference of 66 bytes.**

Binary Diffing

- Binary Diffing is a technique for performing automated binary differential analysis.
- This becomes very useful for reverse engineering patches as well as program updates.
- Some of the available binary diffing tools are:
  - Bindiff
  - PatchDiff
  - Darumgrim
  - Turbodiff
- Here, we used Turbodiff.

Turbodiff

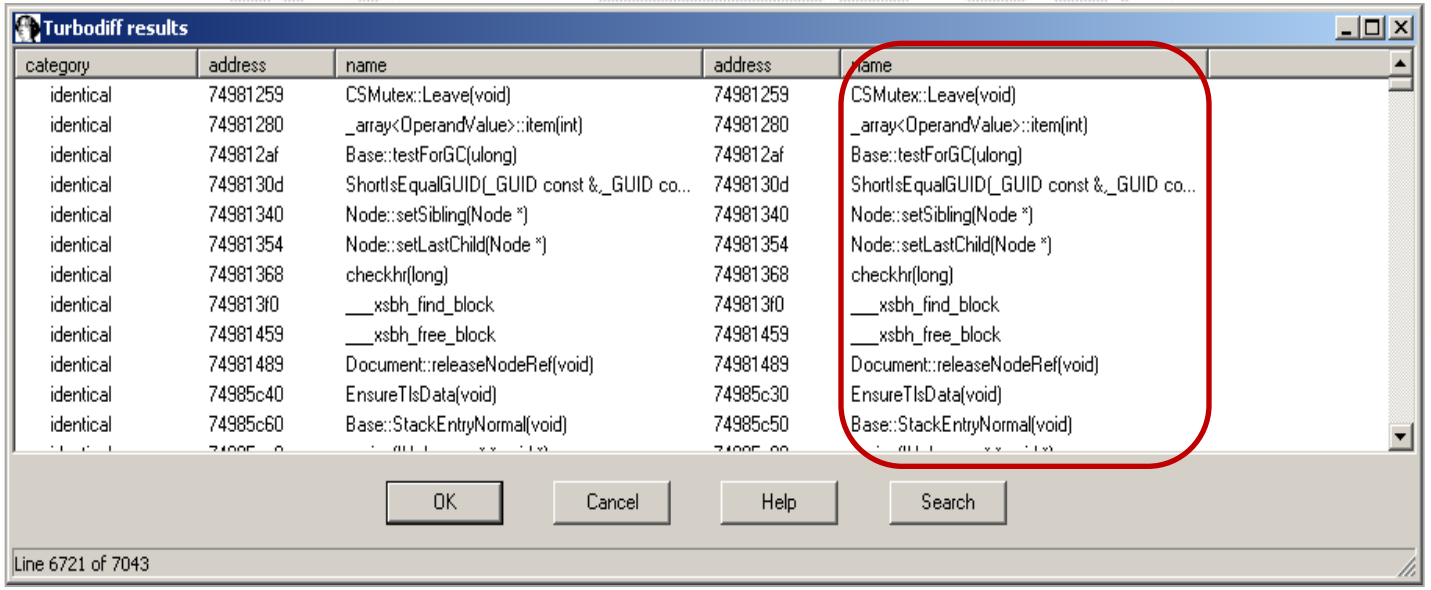
- Turbodiff was programmed by Nicolás Economou.
- It was presented at the Argentinian security conference Ekoparty in 2009.
- It is a heuristic based IDA Plugin aimed for binary diffing.
- This tools was developed in C++.
- It provides an Architecture Independent Diffing.

Turbodiff results (1)

- After analyzing the two binary files, turbodiff creates an ana file from the IDA idb file.

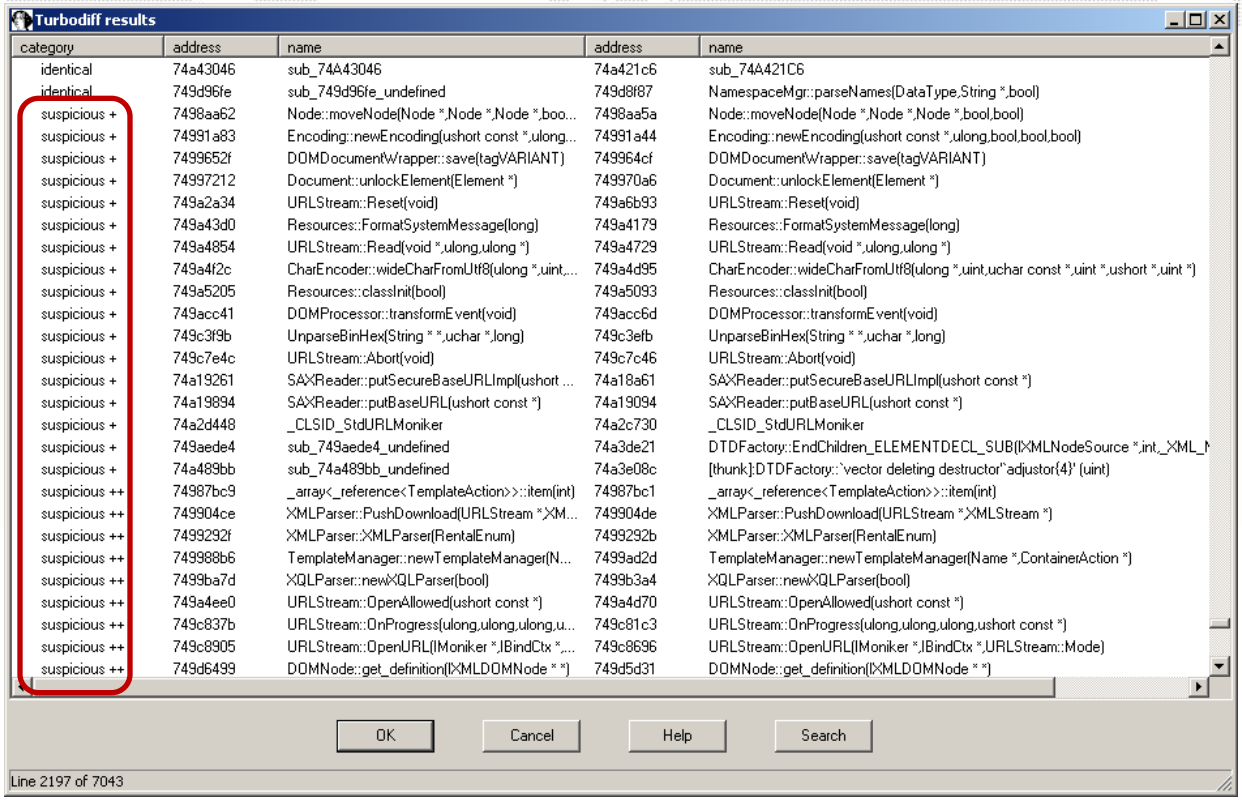
- The aforementioned ana file will be used later in order to detect the **suspicious** and **changed** functions.

- Later turbodiff displays its results:



Turbodiff results (2)

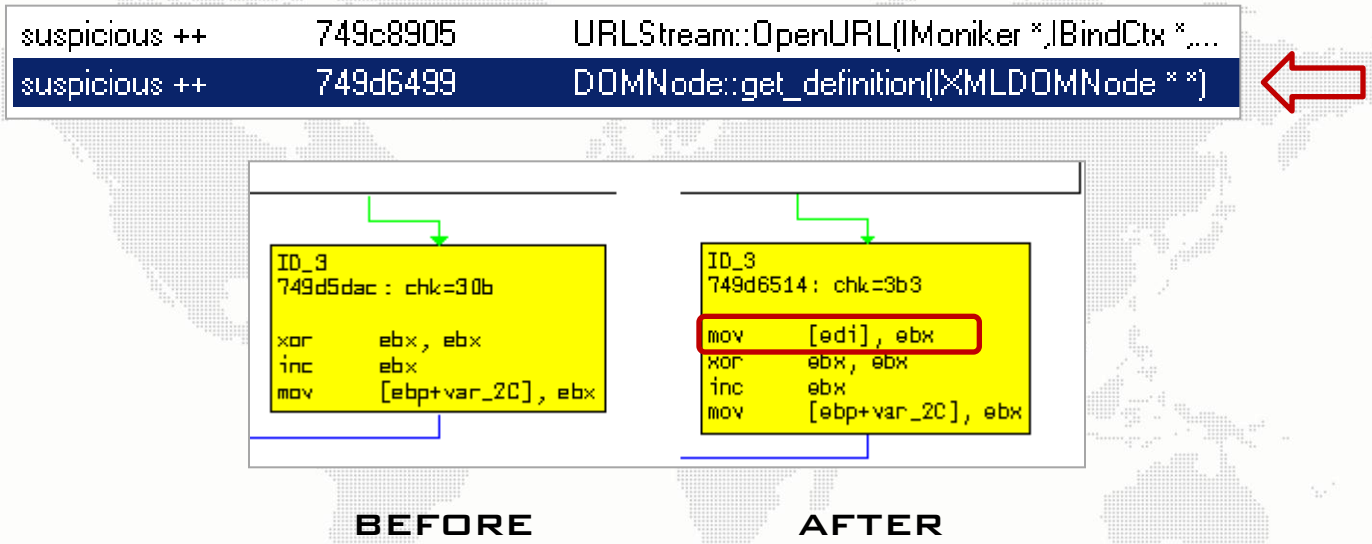
- After examining the differences between the two files:
  - 25 functions are marked as suspicious.
  - 72 functions are marked as changed.



category	address	name	address	name
identical	74a43046	sub_74A43046	74a421c6	sub_74A421C6
identical	749d96fe	sub_749d96fe_undefined	749d8f87	NamespaceMgr::parseNames(DataType,String *,bool)
suspicious +	7498aa62	Node::moveNode(Node *,Node *,Node *,boo...	7498aa5a	Node::moveNode(Node *,Node *,Node *,bool,bool)
suspicious +	74991a83	Encoding::newEncoding(ushort const *,ulong...	74991a44	Encoding::newEncoding(ushort const *,ulong,bool,bool,bool)
suspicious +	7499652f	DOMDocumentWrapper::save(tagVARIANT)	749964cf	DOMDocumentWrapper::save(tagVARIANT)
suspicious +	74997212	Document::unlockElement(Element *)	749970a6	Document::unlockElement(Element *)
suspicious +	749a2a34	URLStream::Reset(void)	749a6b93	URLStream::Reset(void)
suspicious +	749a43d0	Resources::FormatSystemMessage(long)	749a4179	Resources::FormatSystemMessage(long)
suspicious +	749a4854	URLStream::Read(void *,ulong,ulong *)	749a4729	URLStream::Read(void *,ulong,ulong *)
suspicious +	749a4f2c	CharEncoder::wideCharFromUtf8(ulong *,uint,...	749a4d95	CharEncoder::wideCharFromUtf8(ulong *,uint,uchar const *,uint *,ushort *,uint *)
suspicious +	749a5205	Resources::classInit(bool)	749a5093	Resources::classInit(bool)
suspicious +	749acc41	DOMProcessor::transformEvent(void)	749acc6d	DOMProcessor::transformEvent(void)
suspicious +	749c3f9b	UnparseBinHex(String * *,uchar *,long)	749c3efb	UnparseBinHex(String * *,uchar *,long)
suspicious +	749c7e4c	URLStream::Abort(void)	749c7c46	URLStream::Abort(void)
suspicious +	74a19261	SAXReader::putSecureBaseURLImpl(ushort ...	74a18a61	SAXReader::putSecureBaseURLImpl(ushort const *)
suspicious +	74a19894	SAXReader::putBaseURL(ushort const *)	74a19094	SAXReader::putBaseURL(ushort const *)
suspicious +	74a2d448	_CLSID_StdURLMoniker	74a2c730	_CLSID_StdURLMoniker
suspicious +	749aede4	sub_749aede4_undefined	74a3de21	DTDFactory::EndChildren_ELEMENTDECL_SUB(IXMLNodeSource *,int,_XML_N
suspicious +	74a489bb	sub_74a489bb_undefined	74a3e08c	[thunk]:DTDFactory::`vector deleting destructor'adjustor{4}' (uint)
suspicious ++	74987bc9	_array<_reference<TemplateAction>>::item(int)	74987bc1	_array<_reference<TemplateAction>>::item(int)
suspicious ++	749904ce	XMLParser::PushDownload(URLStream *,XM...	749904de	XMLParser::PushDownload(URLStream *,XMLStream *)
suspicious ++	7499292f	XMLParser::XMLParser(RentalEnum)	7499292b	XMLParser::XMLParser(RentalEnum)
suspicious ++	749988b6	TemplateManager::newTemplateManager(N...	7499ad2d	TemplateManager::newTemplateManager(Name *,ContainerAction *)
suspicious ++	7499ba7d	XQLParser::newXQLParser(bool)	7499b3a4	XQLParser::newXQLParser(bool)
suspicious ++	749a4ee0	URLStream::OpenAllowed(ushort const *)	749a4d70	URLStream::OpenAllowed(ushort const *)
suspicious ++	749c837b	URLStream::OnProgress(ulong,ulong,ulong,u...	749c81c3	URLStream::OnProgress(ulong,ulong,ulong,ushort const *)
suspicious ++	749c8905	URLStream::OpenURL(IMoniker *,IBindCtx *,...	749c8696	URLStream::OpenURL(IMoniker *,IBindCtx *,URLStream::Mode)
suspicious ++	749d6499	DOMNode::get_definition(IXMLDOMNode * *)	749d5d31	DOMNode::get_definition(IXMLDOMNode * *)

Turbodiff results (3)

- Let's check the changes in the **DOMNode::get_definition(IXMLDOMNode)** function which is the most important procedure involved in this vulnerability.



- As we can see the instruction **mov [edi], ebx** was added into the **get_definition** function.

- In order to understand this minor change let's analyzed the **whole process.**

Flow analysis (1)

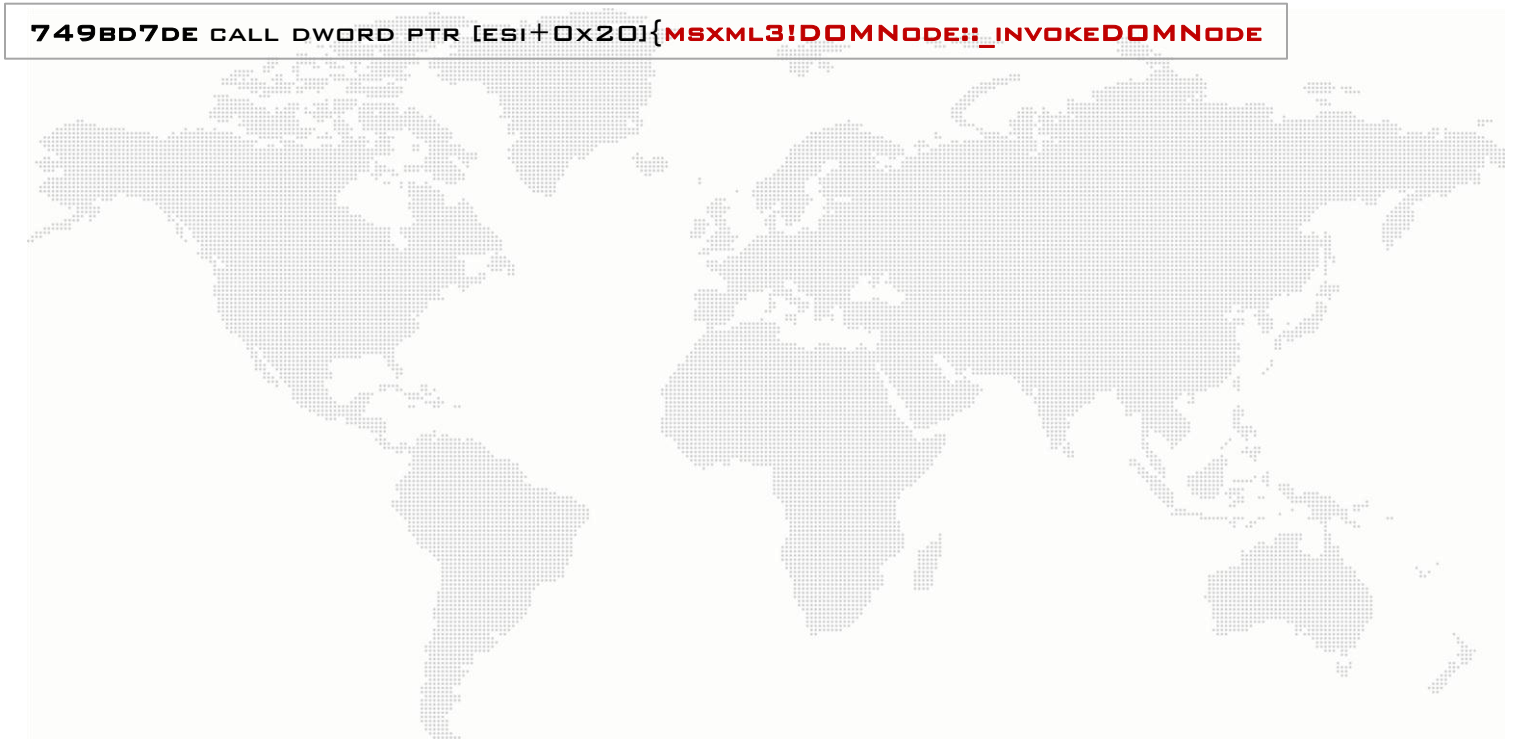
749bd756 _dispatchImpl::InvokeHelper

Flow analysis (2)

HIGH-TECH BRIDGE®
INFORMATION SECURITY SOLUTIONS

```
749bd756 _dispatchImpl::InvokeHelper
```

```
749bd7de call dword ptr [esi+0x20]{msxml3!DOMNode::_invokeDOMNode
```



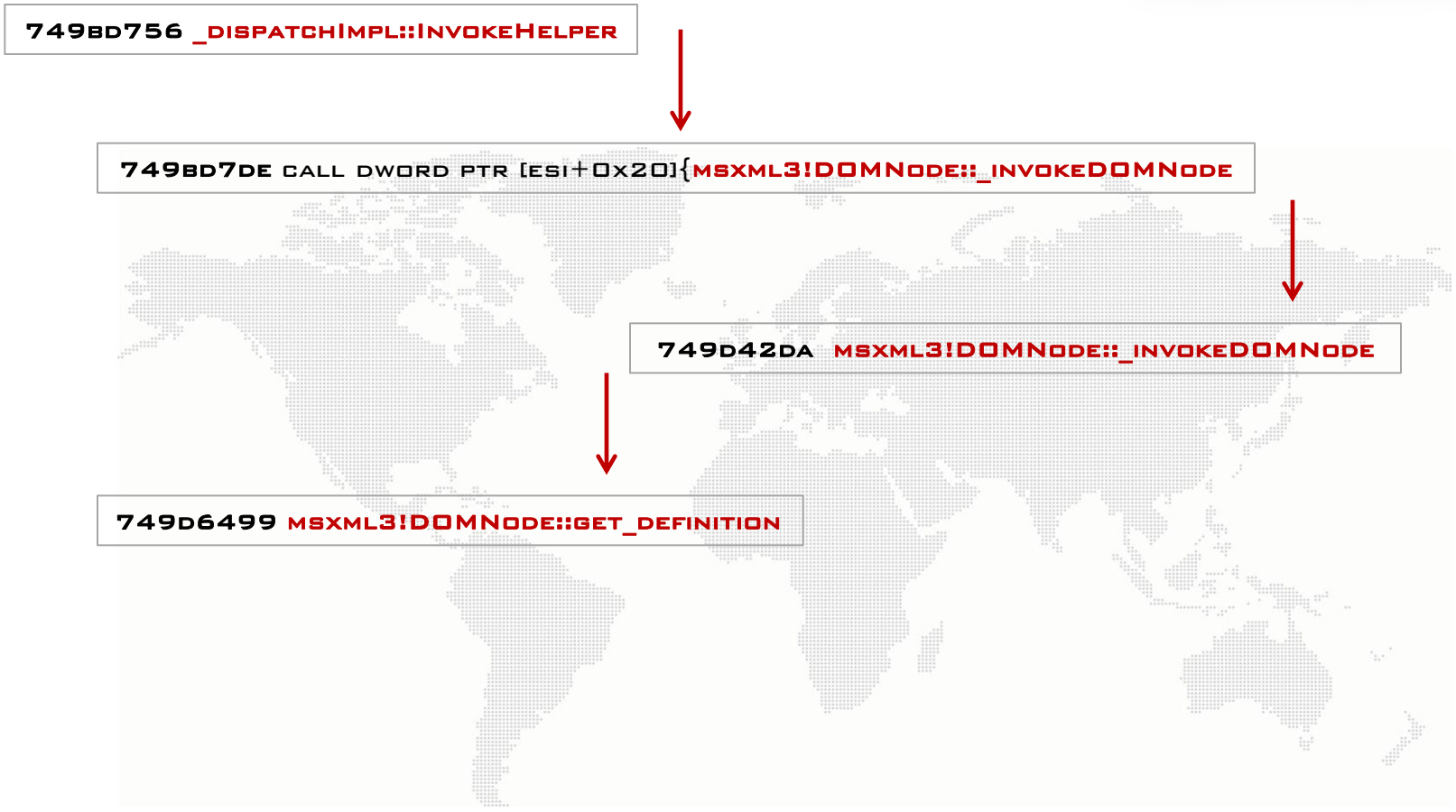
Flow analysis (3)

749bd756 _dispatchImpl::InvokeHelper

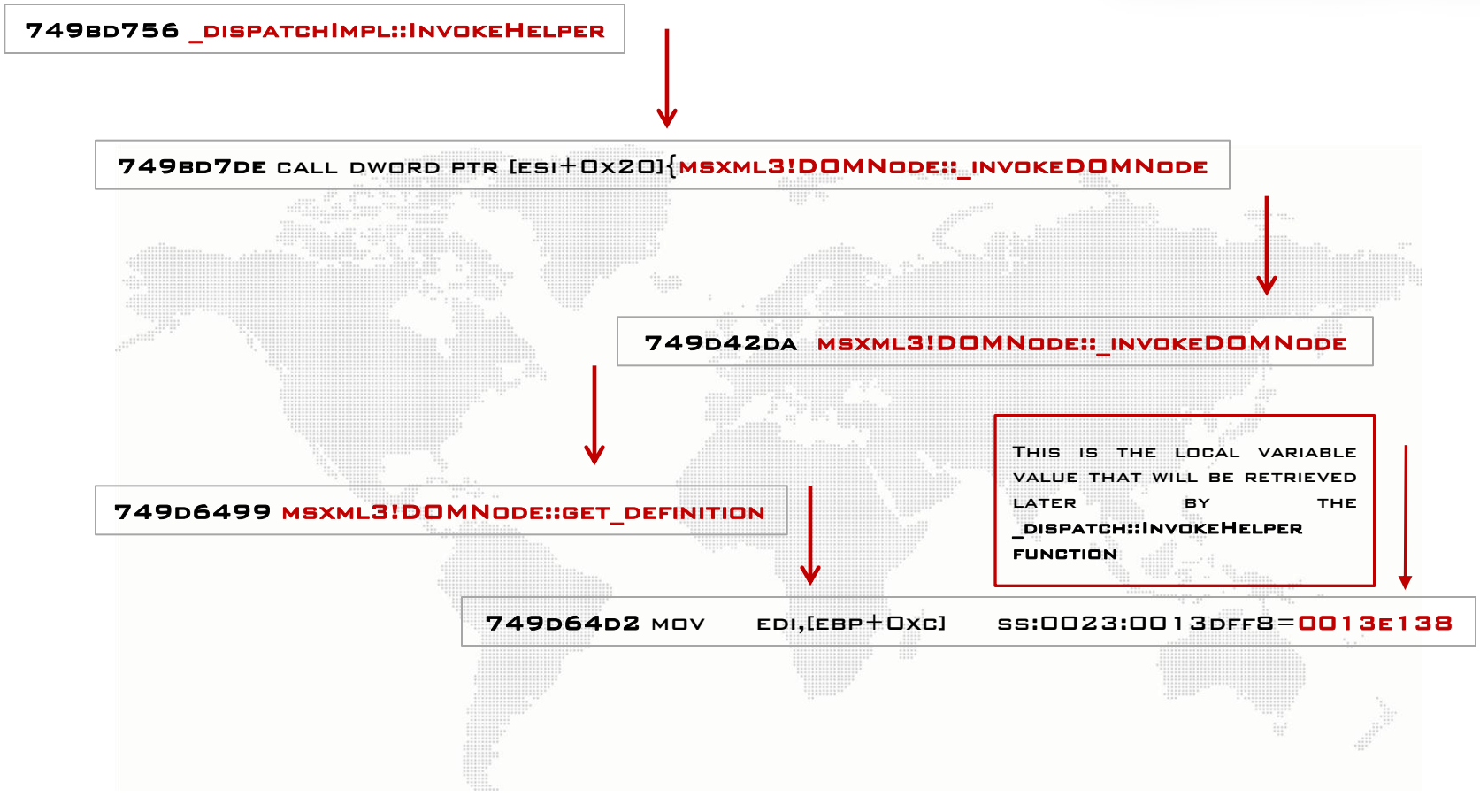
749bd7de call dword ptr [esi+0x20]{msxml3!DOMNode::_invokeDOMNode

749d42da msxml3!DOMNode::_invokeDOMNode

Flow analysis (4)



Flow analysis (5)



Flow analysis (6)

749bd756 _dispatchImpl::InvokeHelper

749bd7de call dword ptr [esi+0x20]{msxml3!DOMNode::_invokeDOMNode

749d42da msxml3!DOMNode::_invokeDOMNode

749d6499 msxml3!DOMNode::get_definition

This is the local variable value that will be retrieved later by the _dispatch::InvokeHelper function

749d64d2 mov edi,[ebp+0xc] ss:0023:0013dff8=0013e138

edi	13e138
esi	1cf48a4
ebx	0
edx	0
ecx	2360040
eax	13dfac
ebp	13dfec
eip	749d64d5

Flow analysis (7)

749bd756 _DispatchImpl::InvokeHelper

749bd7de call dword ptr [esi+0x20]{msxml3!DOMNode::_invokeDOMNode

749d42da msxml3!DOMNode::_invokeDOMNode

749d6499 msxml3!DOMNode::get_definition

This is the local variable value that will be retrieved later by the _dispatch::InvokeHelper function

749d64d2 mov edi,[ebp+0xc] ss:0023:0013dff8=0013e138

edi	13e138
esi	1cf48a4
ebx	0
edx	0
ecx	2360040
eax	13dfac
ebp	13dfec
eip	749d64d5

749d6514 mov [edi],ebx ds:0023:0013e138=0c0c0c08

edi	13e138
esi	1cf48a4
ebx	0

Flow analysis (8)

749bd756 _dispatchImpl::InvokeHelper

749bd7de call dword ptr [esi+0x20]{msxml3!DOMNode::_invokeDOMNode

749d42da msxml3!DOMNode::_invokeDOMNode

749d6499 msxml3!DOMNode::get_definition

This is the local variable value that will be retrieved later by the _dispatch::InvokeHelper function

749d64d2 mov edi,[ebp+0xc] ss:0023:0013dff8=0013e138

edi	13e138
esi	1cf48a4
ebx	0
edx	0
ecx	2360040
eax	13dfac
ebp	13dfec
eip	749d64d5

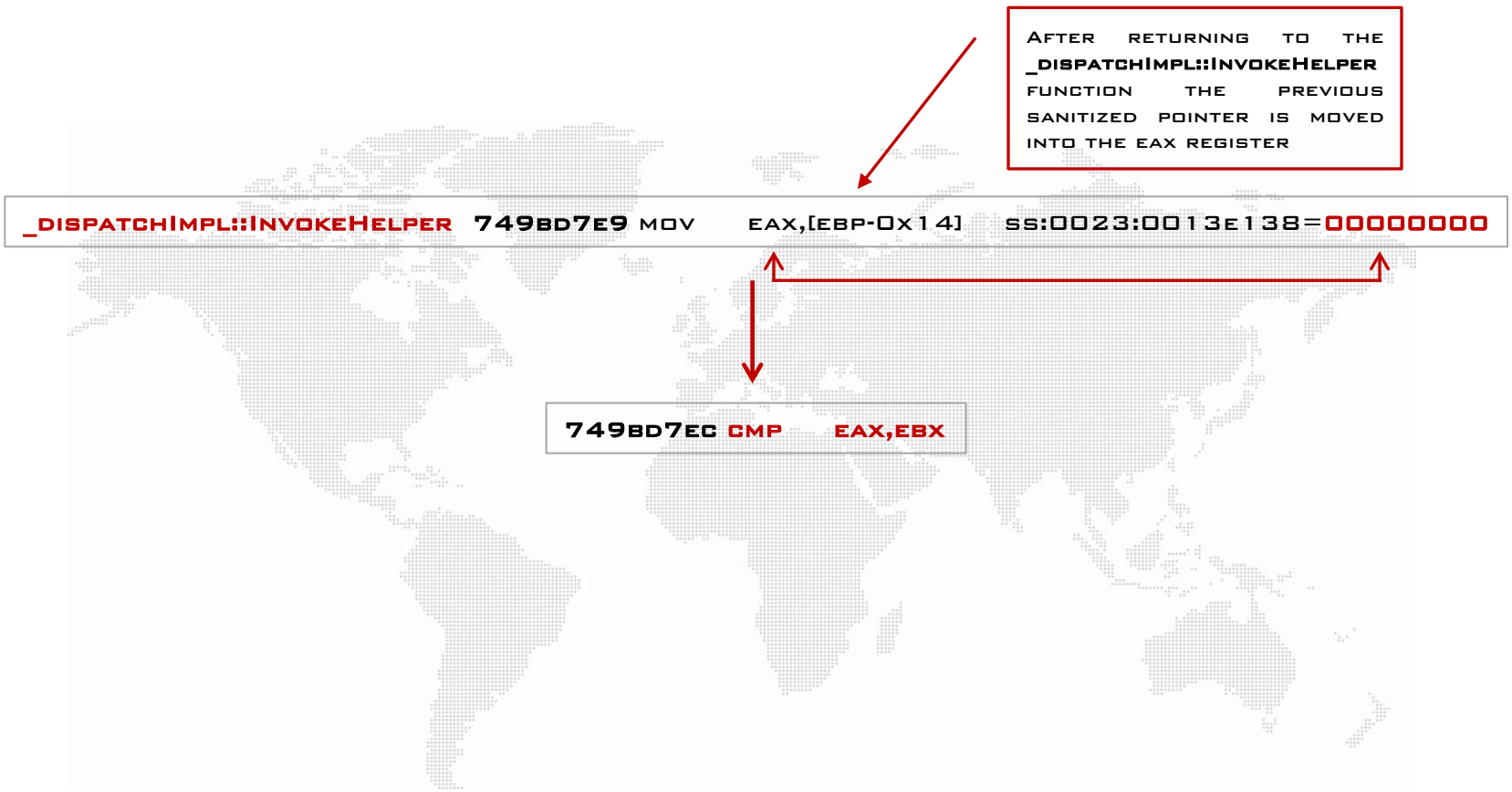
This instruction corresponds to the **security update.** The content of the edi will be initialized **to zero**

749d6514 mov [edi],ebx ds:0023:0013e138=0c0c0c08

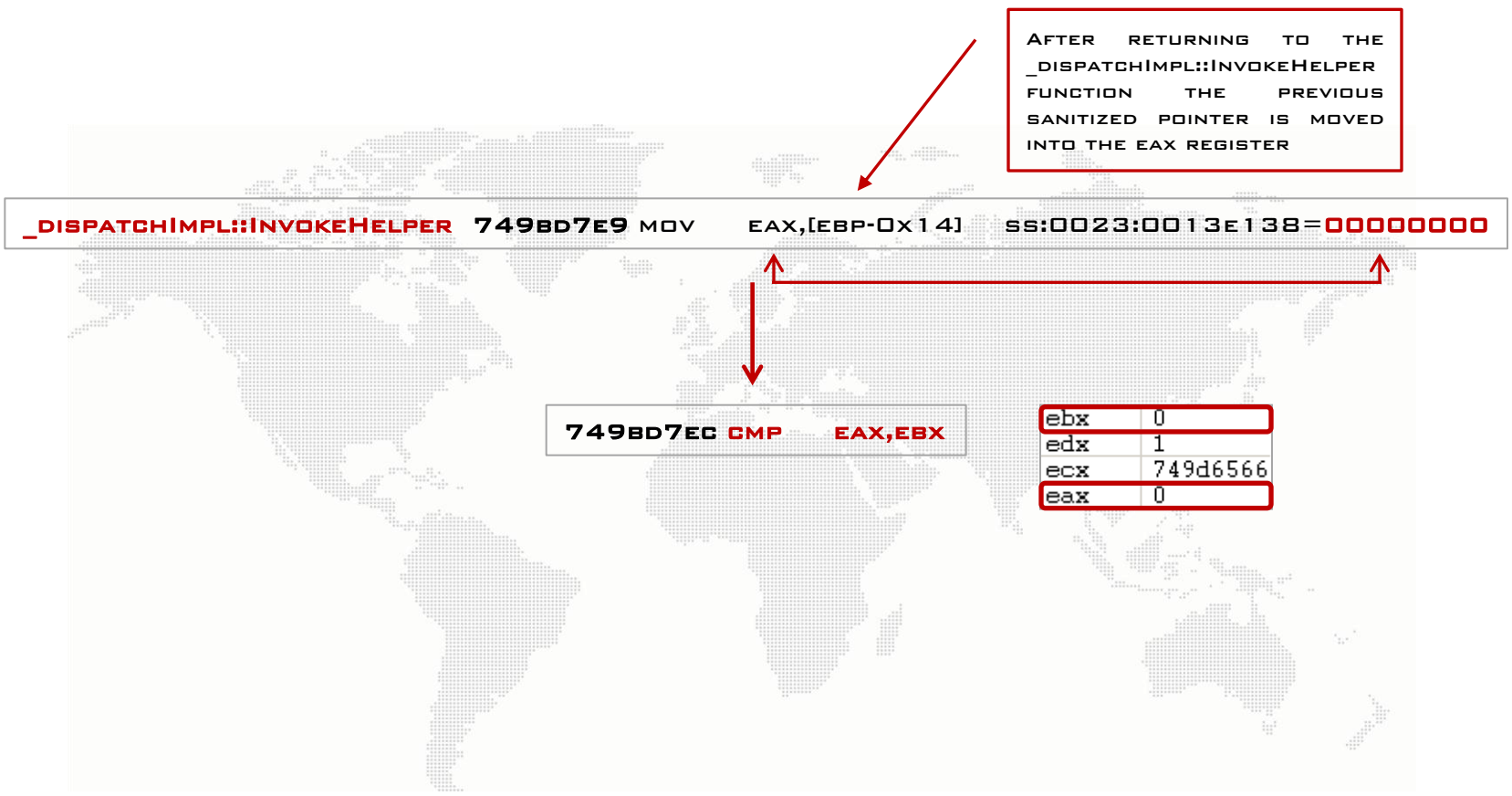
edi	13e138
esi	1cf48a4
ebx	0

Flow analysis (9)

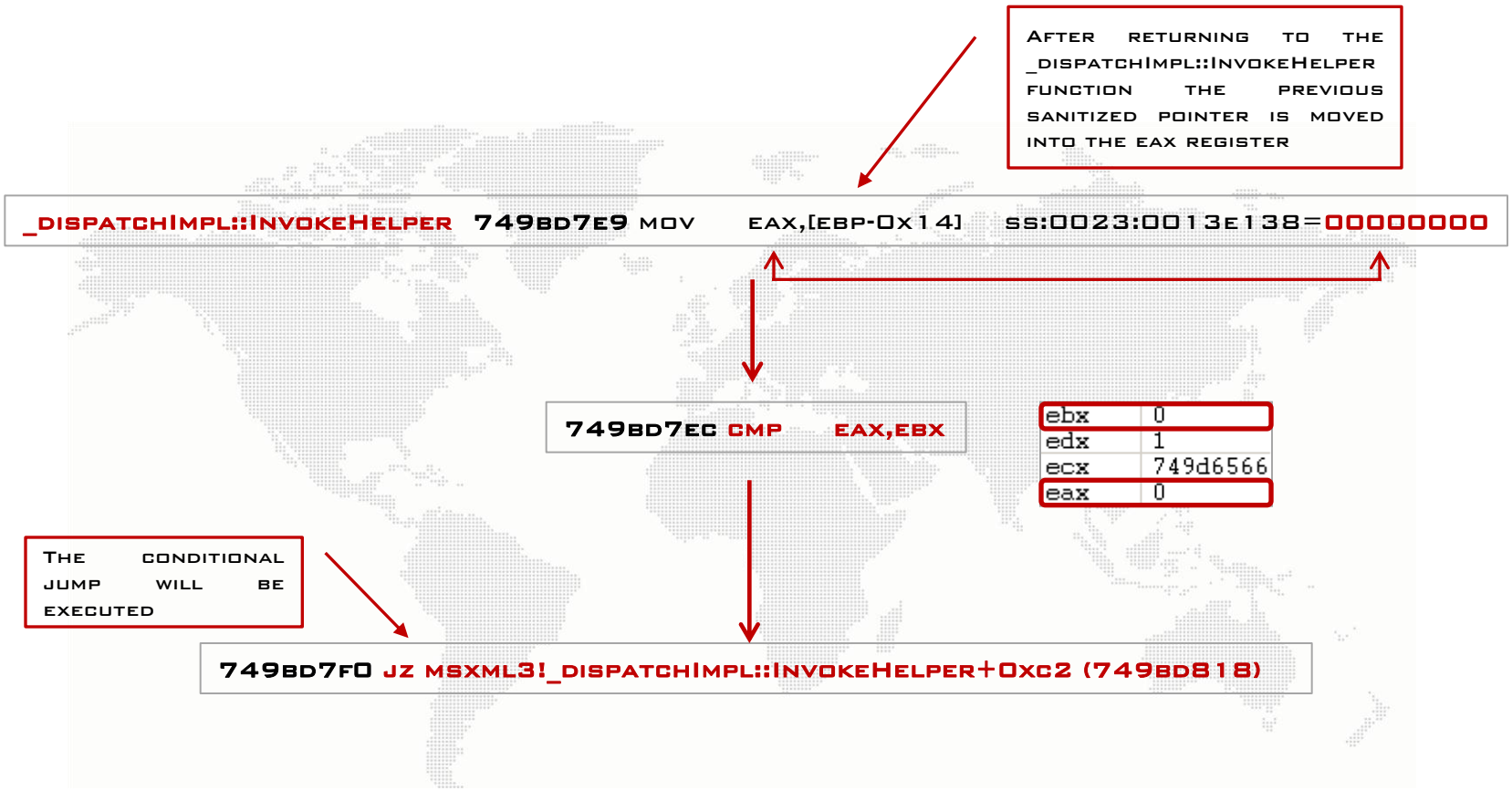
Flow analysis (10)



Flow analysis (11)

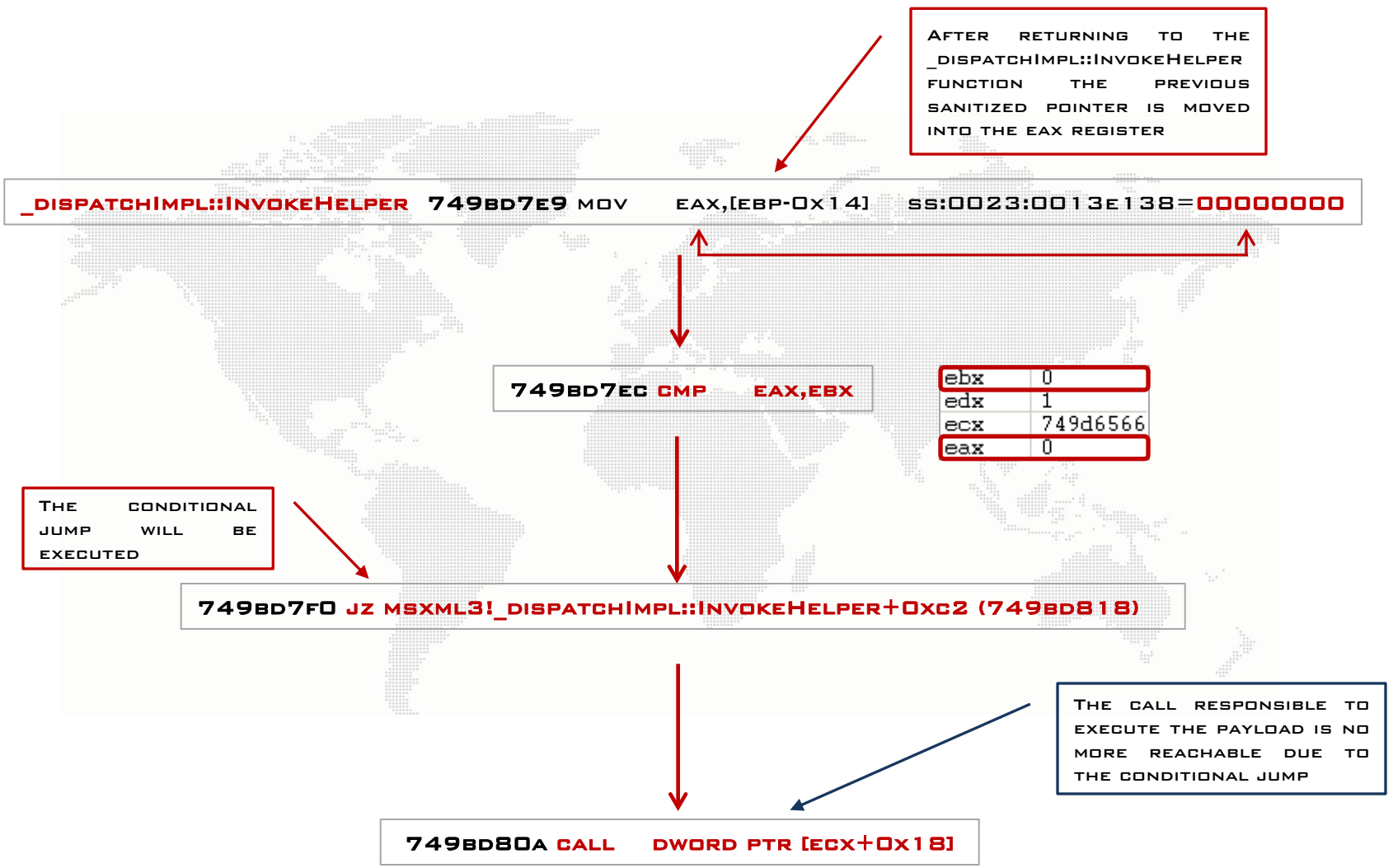


Flow analysis (12)



Flow analysis (13)

HIGH-TECH BRIDGE
INFORMATION SECURITY SOLUTIONS



Conclusions

- As we have seen **the main change** in the **XML** security update **for Windows XP-SP3** is the **mov [edi],ebx** instruction.

```
749d6514    891f        mov     [edi],ebx
```

- This instruction sanitizes the value that will be retrieved later by the **_DispatchImpl::InvokeHelper function.**

- If one modifies the two bytes instruction (**891f**) with NOP's instructions (**9090**) the whole security updated could be deactivate.

- Apply the security update (**KB2719985**) as soon as you can since this vulnerability is heavily exploited in the wild nowadays.

References

- http://www.microsoft.com/fr-fr/download/details.aspx?id=30290
- http://support.microsoft.com/kb/2719985
- http://www.openrce.org/forums/posts/82
- http://corelabs.coresecurity.com/index.php?module=Wiki&action=attachment&type=publication&page=Heuristicas_aplicadas_a_la_comparacion_%28_diffeo_%29_de_binarios&file=Economou_2009-binary_diffing.pdf

Acknowledgments

- Thanks to **Nicolas Economou** from **coresecurity** for allowing us to publish the document using its utility **Turbodiff** :]

- http://corelabs.coresecurity.com/index.php?module=Wiki&action=view&type=tool&name=**Turbodiff**

Thanks for reading

HIGH-TECH BRIDGE®
INFORMATION SECURITY SOLUTIONS

Your questions are always welcome!

brian.mariani@htbridge.ch

frederic.bourla@htbridge.ch