Matthew “j00ru” Jurczyk, Gynvael Coldwind
HISPASEC

# CASE STUDY OF RECENT WINDOWS VULNERABILITIES

# Eyjafjallajökull

*Photo by Árni Friðriksson*

# Me, Myself and I

Reverse Engineering

Pentesting

Creating Tools

Gynvael Coldwind

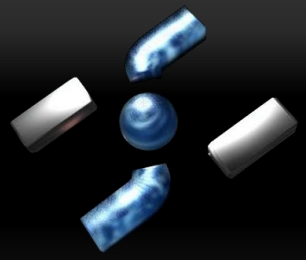
Vulnerability Research

Matthew “j00ru” Jurczyk

http://gynvael.coldwind.pl/

http://j00ru.vexillium.org/

IT Security Reseracher
HISPASEC



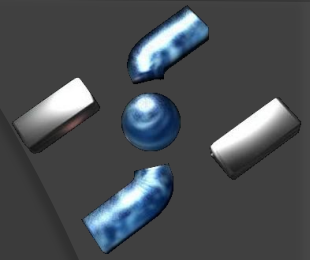
# Me, Myself and I

Matthew “j00ru” Jurczyk

# Me, Myself and I

Gynvael Coldwind

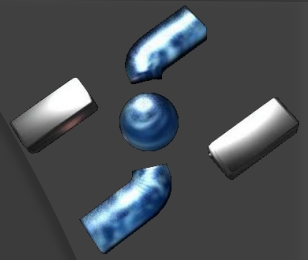
# 7 vulnerabilities agenda



- CSRSS Local Elevation of Privileges
- Registry Link Unicode Parity Buffer Overflow DoS
- Registry Link 16-bit Integer Wrap Buffer Overflow Local Elevation of Privileges
- Registry Link Access Control List NULL Pointer Dereference DoS
- Registry Link Race Condition DoS
- Registry Link Cross-Hive Registry Information Disclosure
- Registry Link Cross-Hive Local Elevation of Privileges

# CSRSS Local Elevation of Privileges

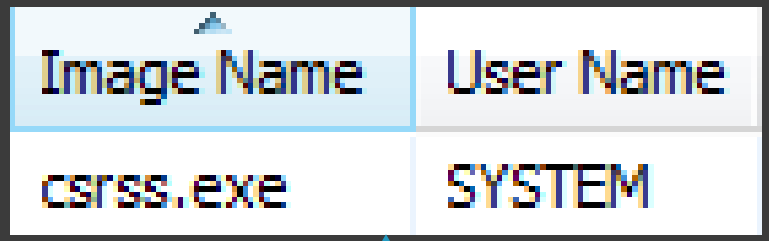
**CVE-2010-0023**



Affected Windows versions:
- Windows 2000 x86 SP4
- Windows XP x86 SP2 & SP3
- Windows XP x86-64 SP2
- Windows Server 2003 x86 SP2
- Windows Server 2003 x86-64 SP2
- Windows Server 2003 Itanium SP2

# DEMO 1

# CSRSS – what is it?




Winsrv.dll

Csrsrv.dll

Basesrv.dll

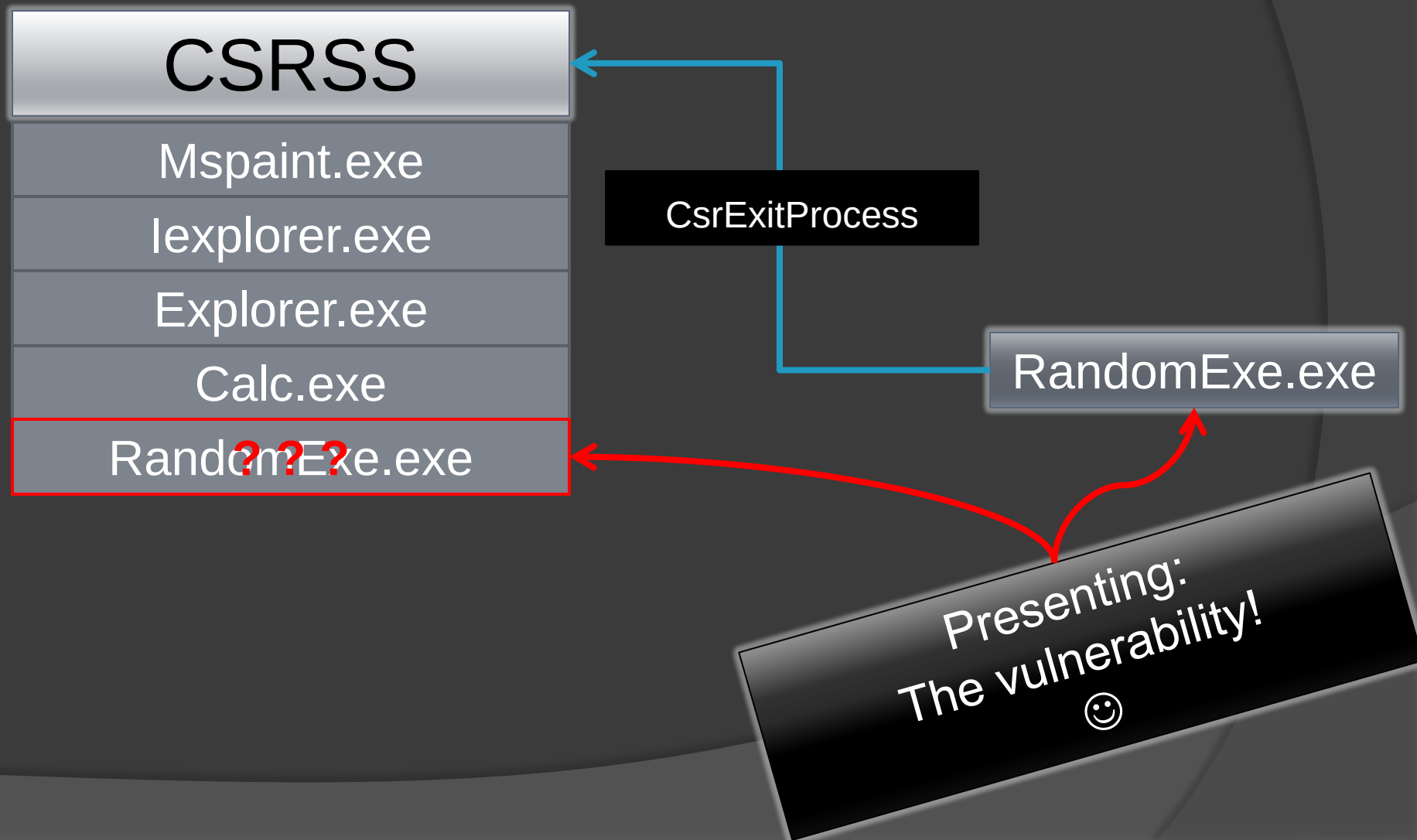
Client/Server Runtime Subsystem

Windows Subsystem

Communicates with processes via (Advanced) LPC

Manages a list of processes
The list is used to kill them

# CSRSS – vulnerability?

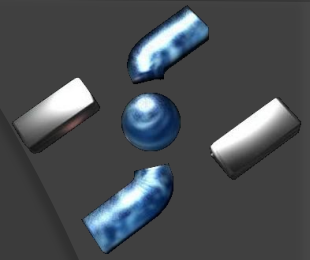


# CSRSS – how the exploit works


**Plant exploit** → **Admin log in** → **Come back**

trigger

# CSRSS – what does work?



Not everything from the API works when CSRSS thinks the process is dead!

Sending key strokes

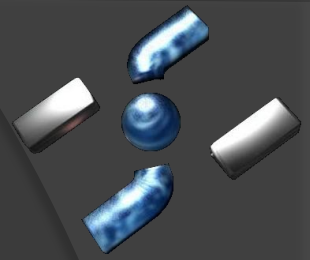
Displaying windows

Enumerating windows

Making screenshots

Keylogging

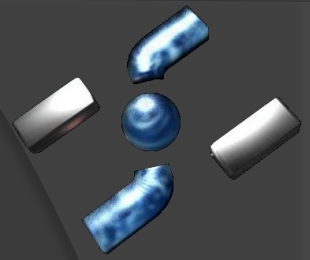
# CSRSS – how the exploit works



**Immortal Exploit**

1. Send the CsrExitProcess opcode
2. Log off
3. Wait for another user (admin!) to log on
4. Take a screen shot*
5. Display the screen shot top most*
6. Run “net localgroup administrators add EvilNotMaid”*
7. Let the user log off, and relog as admin ☺

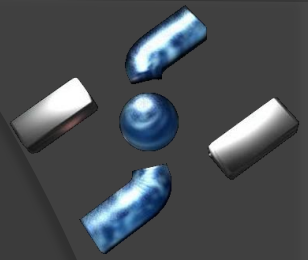
# CSRSS - how the exploit works



DEMO 2
Again

# Registry Link Unicode Parity Buffer Overflow DoS

**CVE-2010-0235**



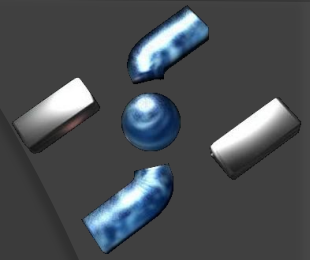
Affected Windows versions:
- Windows 2000 SP4
- Windows XP SP2 & SP3
- Windows Server 2003 SP2
- Windows Vista Gold

# DEMO 3

# Registry – what is a Link?

No magic here ☺
Just plain simple “this key points somewhere else”

# Registry – what is a Link? Technically…



REG_LINK type key

Default Value = destination key

UNICODE

\Registry\User*SID*\Software

EXAMPLE

2-bytes per char

# Registry – How to create a link?

REGLN by Antoni Sawicki
http://www.tenox.tc/out/#regln

**OR**

2

```
NtCreateKey with REG_OPTION_CREATE_LINK
NtSetValueKey with REG_LINK
```

```
L"SymbolicLinkValue"
```

# Registry – the vulnerability?

```
NtSetValueKey(
  IN HANDLE KeyHandle,
  IN PUNICODE_STRING ValueName,
  IN ULONG TitleIndex OPTIONAL,
  IN ULONG Type,
  IN PVOID Data,
  IN ULONG DataSize );
```

3

1

-1

-3

...

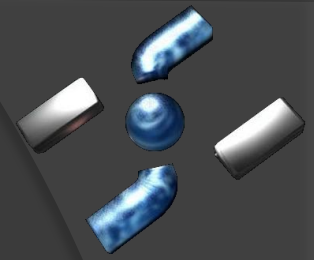
What if DataSize is odd? E.g. 3?

```
Count = DataSize;
while(Count)
{
   […]
   Count -= sizeof(WCHAR);
   […]
}
```

2-bytes per char

Presenting: The vulnerability! ☺

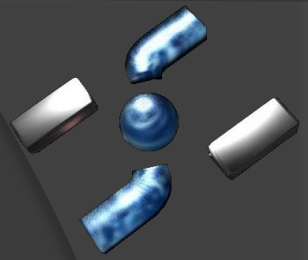
Registry Link Unicode Parity Buffer Overflow DoS



# DEMO 4
# Again

# Registry Link 16-bit Integer Wrap Buffer Overflow Local Elevation of Privileges

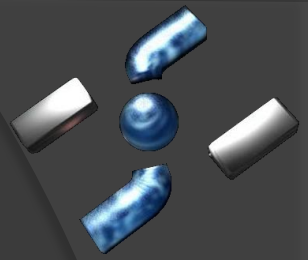
**CVE-2010-0236**

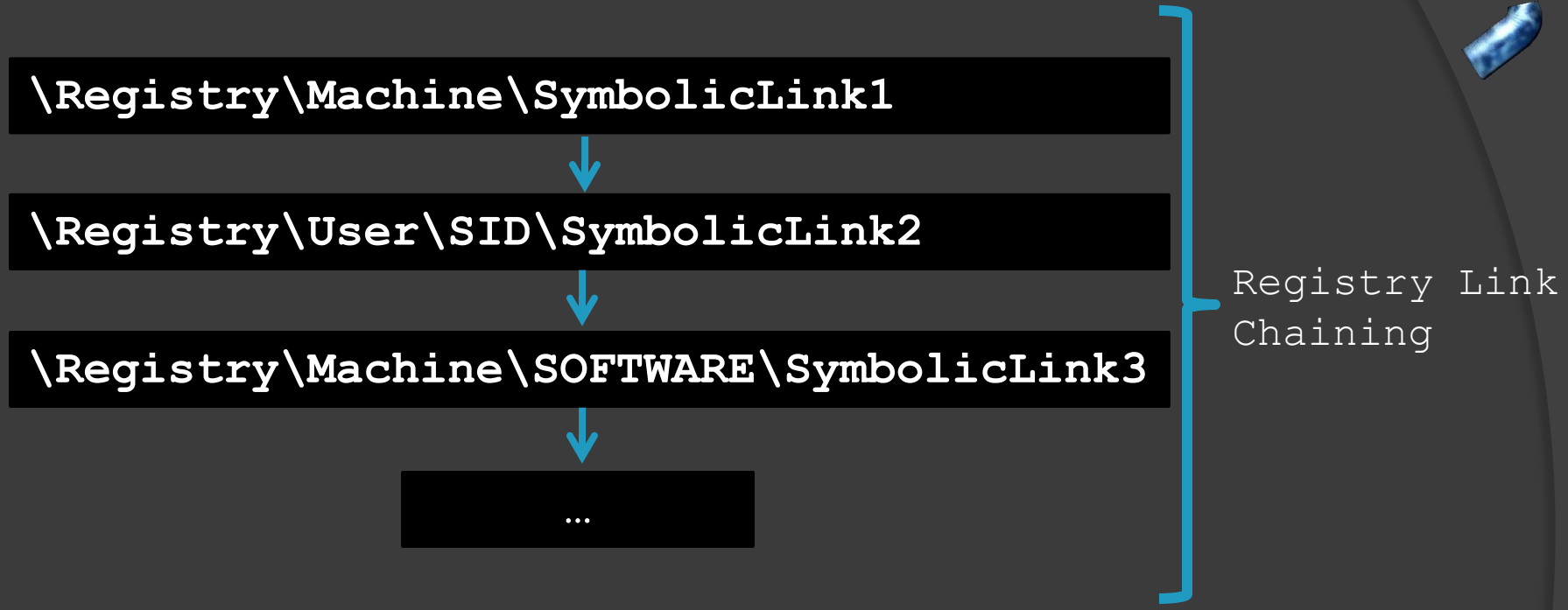


Affected Windows versions:
- Windows 2000 SP4
- Windows XP SP2 & SP3
- Windows Server 2003 SP2
- Windows Vista Gold

# DEMO 5

# Creating Symbolic Link Chains – what is it?






Each key points to the successive key (link) …

… till a normal key is encountered

# Chained Symbolic Link Management

## CmpGetSymbolicLink function

1. Get the L”SymbolicLinkValue” value contents

2. Length = (USHORT)ValueLength + sizeof(WCHAR);

3. Check if Length > 0xFFFF ← **Sanity check fail ☺**

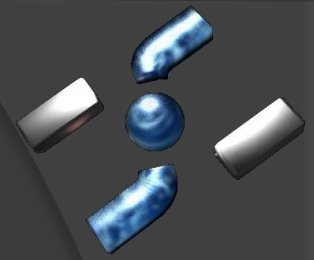
4. If Length > Current Buffer’s Length

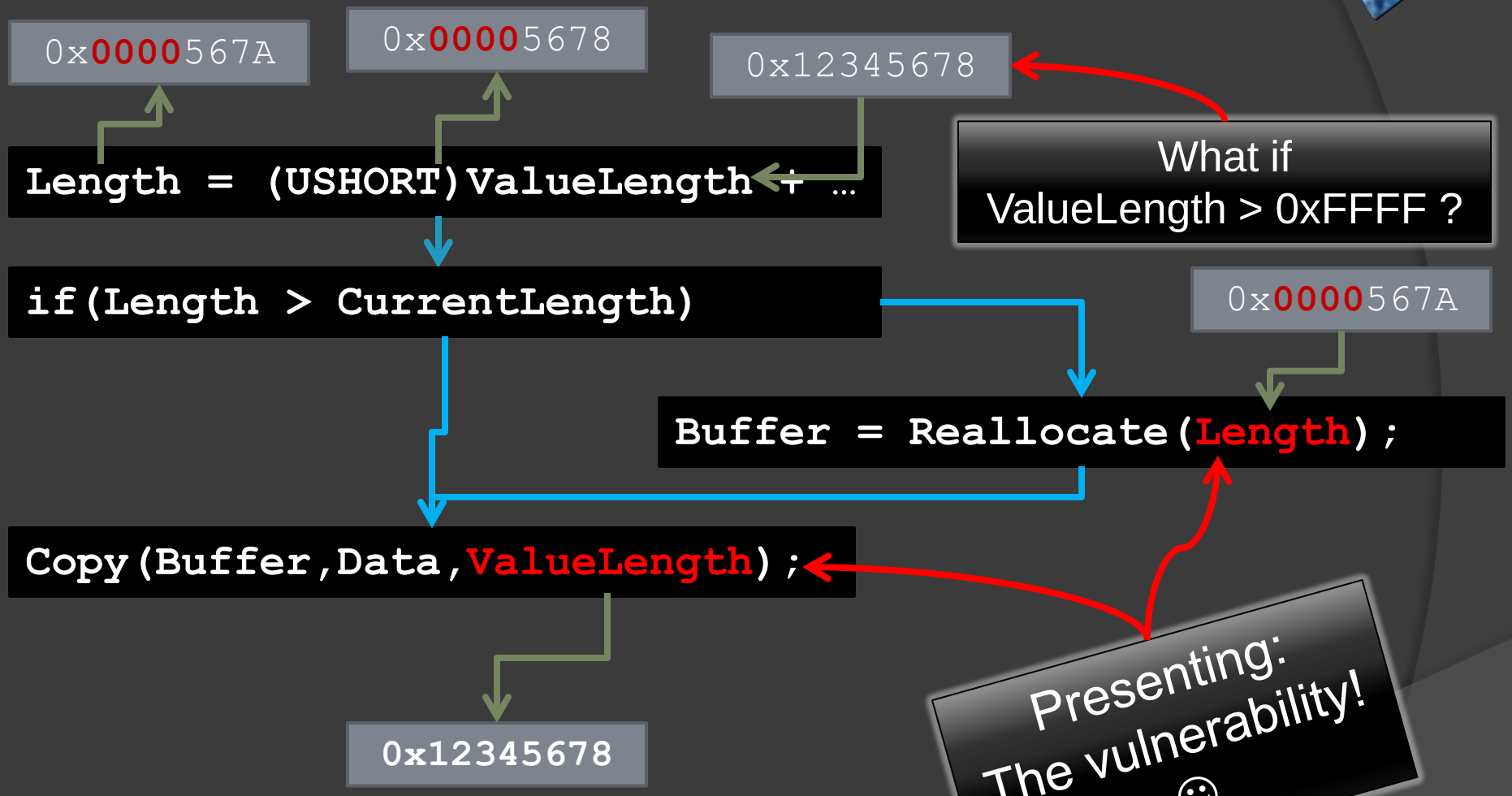
4.1. Reallocate the existing buffer, using Length

5. Copy the value data into the buffer, using ValueLength

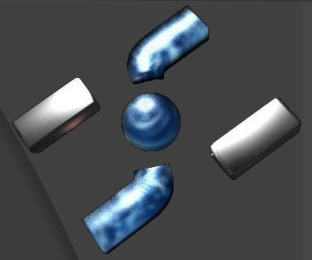
# Chained Symbolic Link Management

## CmpGetSymbolicLink function





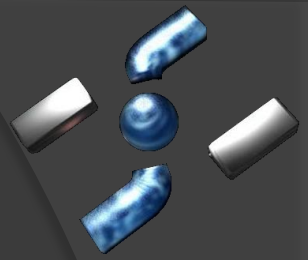
Registry Link 16-bit Integer Wrap Buffer Overflow Local Elevation of Privileges



# DEMO 6 Again

# Registry Link Access Control List NULL Pointer Dereference DoS

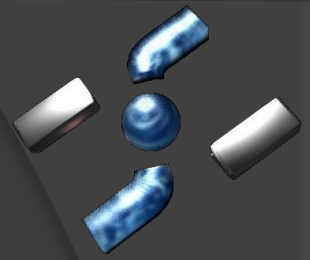
**CVE-2010-0234**



Affected Windows versions:
- Windows 2000 SP4
- Windows XP SP2 & SP3
- Windows Server 2003 SP2
- Windows Vista Gold, SP1, SP2
- Windows Server 2008 Gold, SP2

# DEMO 7

# Registry Symbolic Link capabilities



Transparent for reading:

- RegQueryValue
- RegQueryMultipleValues
- RegQueryInfoKey

Transparent for writing:

- RegSetKeyValue
- RegSetValue
- RegCreateKey

How about security rights?

# Registry Symbolic Link – security access rights

The kernel fails to parse the symbolic link name

CompleteName parameter:

```
kd> dt nt!_UNICODE_STRING
      +0x000 Length          : 0
      +0x002 MaximumLength   : 0
      +0x004 Buffer          : (null)
```

NtOpenKey

↓

ObOpenObjectByName

↓

ObpLookupObjectName

↓

CmpParseKey

↓

CmpGetSymbolicLink

↓

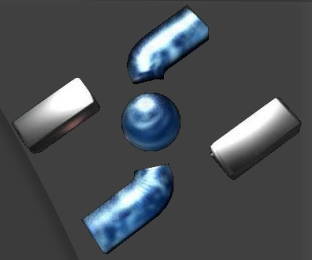
ExFreePoolWithTag(NULL);

↓

KeBugCheckEx(0x40);

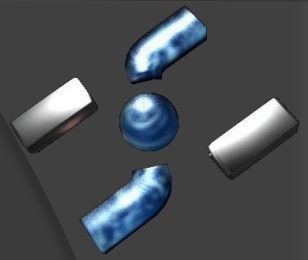

Presenting:
The vulnerability!
☺

# Registry Symbolic Link – security access rights



DEMO 8
AGAIN

# Registry Link Race Condition DoS

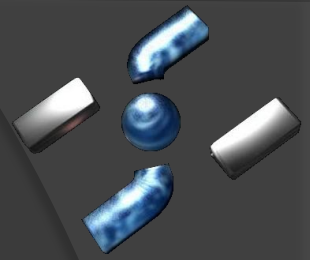


**CVE-2010-0238**

Affected Windows versions:
- Windows 2000 SP4
- Windows XP SP2 & SP3
- Windows Server 2003 SP2
- Windows Vista Gold

# DEMO 9

# Registry Link Race Condition DoS



Is registry access through links thread-safe?

Mmm… nope!

Fine, how do we check it?

Let’s reference a link through MAANY threads! ☺

# Registry Link Race Condition DoS

100 seems to be enough

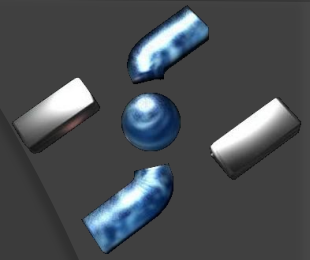
```
for( int i=0;i<NumberOfThreads;i++ )
{
  CreateThread(ThreadRoutine);
}
```

```
while(1)
{
  RegOpenKeyEx(RegistryLink);
}
```

The results?

You've seen it already ☺

# Registry Link Race Condition DoS - details



**User-mode address**

```
NtOpenKey(KeyHandle,Access,ObjectAttributes)
```

```
ObOpenObjectByName(ObjectAttributes,…
```

```
ObpLookupObjectName(RootDirectory~,ObjectName,…
```

On multiple references…

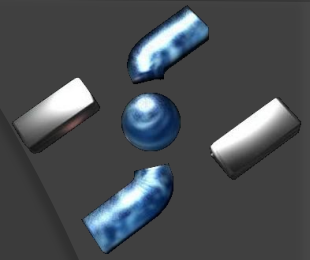
… the synchronization fails ☺

# Registry Link Race Condition DoS - details

## Just like that:

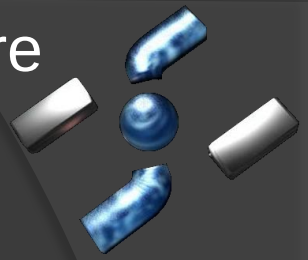
```
eax=00f8000f ebx=f40b6c68 ecx=e1dab000 edx=00000011
esi=0052005c edi=00000000 eip=80563ed6 esp=f40b6bd0
ebp=f40b6c28 iopl=0         nv up ei ng nz ac pe cy
cs=0008  ss=0010  ds=0023  es=0023  fs=0030  gs=0000             efl=00010297

nt!ObpLookupObjectName+0x355:
0008:80563ed6 6683395c        cmp     word ptr [ecx],5Ch  ds:0023:e1dab000=????
```

# Registry Link Race Condition DoS



DEMO 9
AGAIN

# Registry Link Cross-Hive Registry Information Disclosure



**CVE-2010-0237**
**2 in 1**

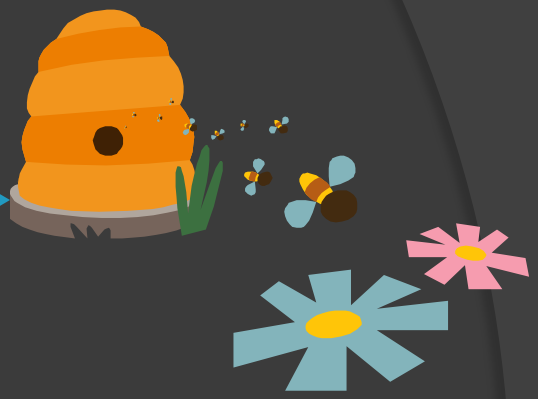
Affected Windows versions:
- Windows 2000 x86 SP4
- Windows XP x86 SP2 & SP3
- Windows XP x86-64 SP2

# DEMO 10

# Registry Hive – what is it ?

Registry split into hives

Each hive is in a separate file



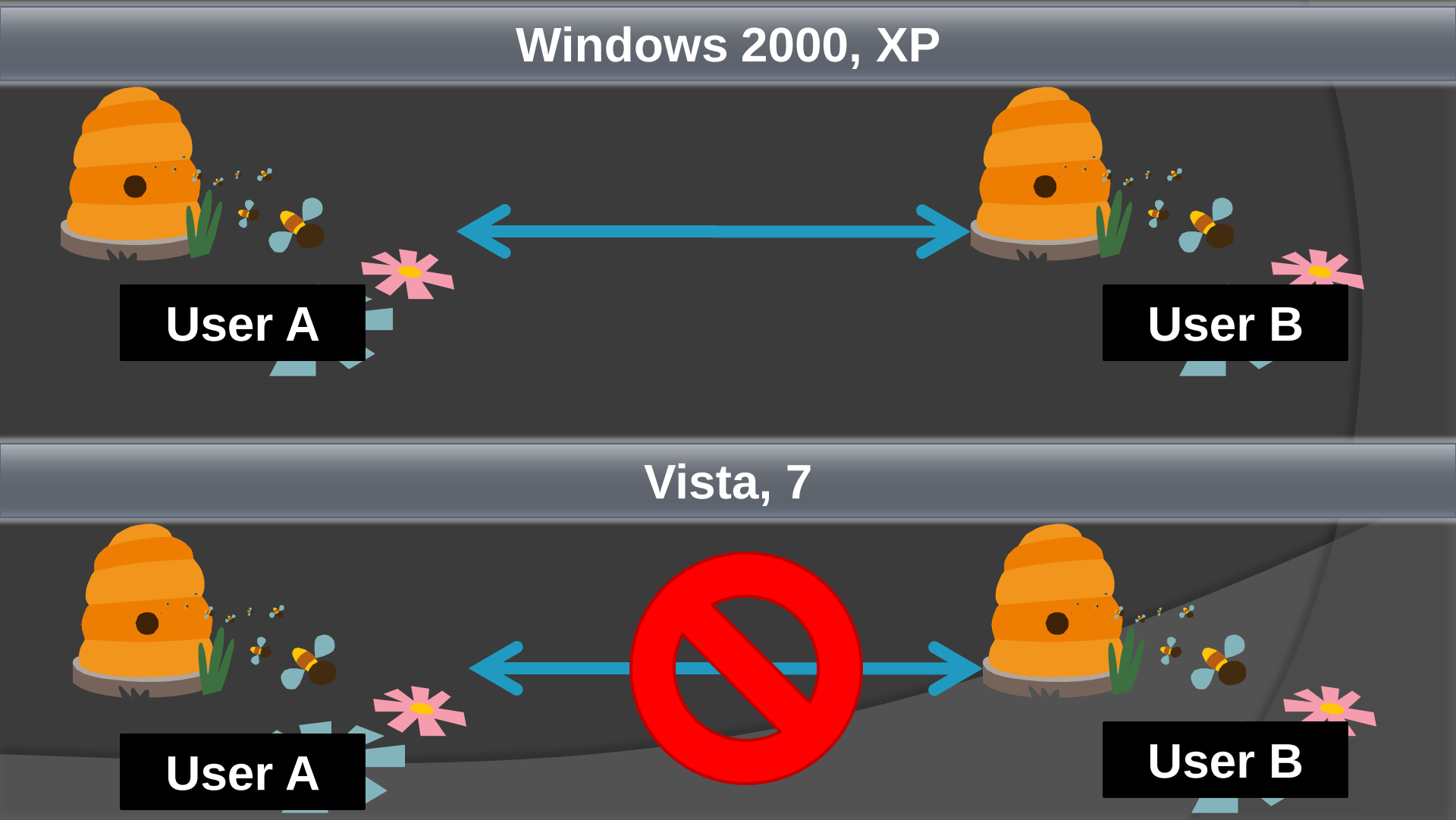
| Registry hive | Supporting files |
|---|---|
| **HKEY_CURRENT_CONFIG** | System, System.alt, System.log, System.sav |
| **HKEY_CURRENT_USER** | Ntuser.dat, Ntuser.dat.log |
| **HKEY_LOCAL_MACHINE\SAM** | Sam, Sam.log, Sam.sav |
| **HKEY_LOCAL_MACHINE\Security** | Security, Security.log, Security.sav |
| **HKEY_LOCAL_MACHINE\Software** | Software, Software.log, Software.sav |
| **HKEY_LOCAL_MACHINE\System** | System, System.alt, System.log, System.sav |
| **HKEY_USERS\.DEFAULT** | Default, Default.log, Default.sav |

# Registry Hive – links between untrusted hives

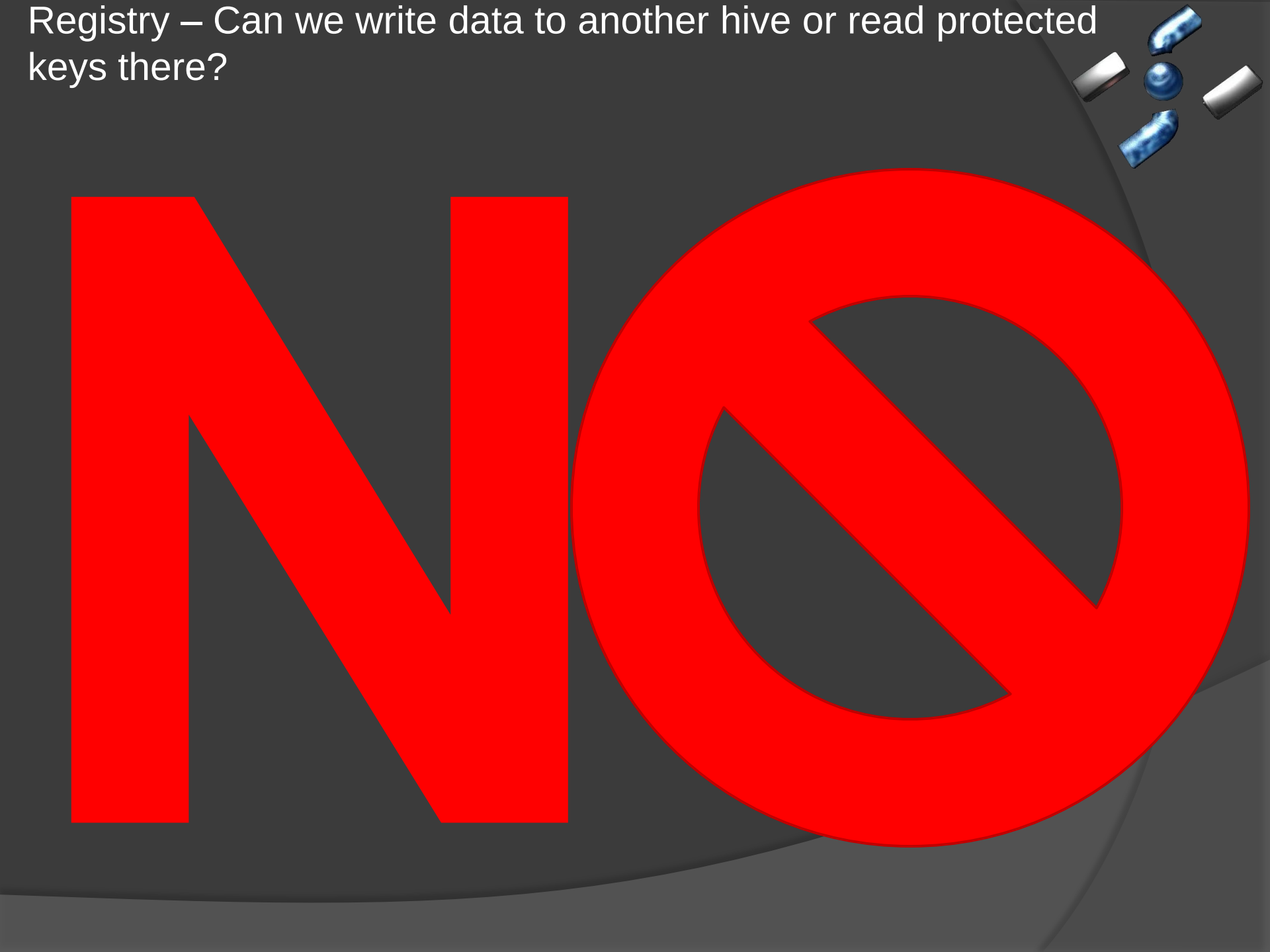
Do links between hives work ?

YES – in case of trusted hives

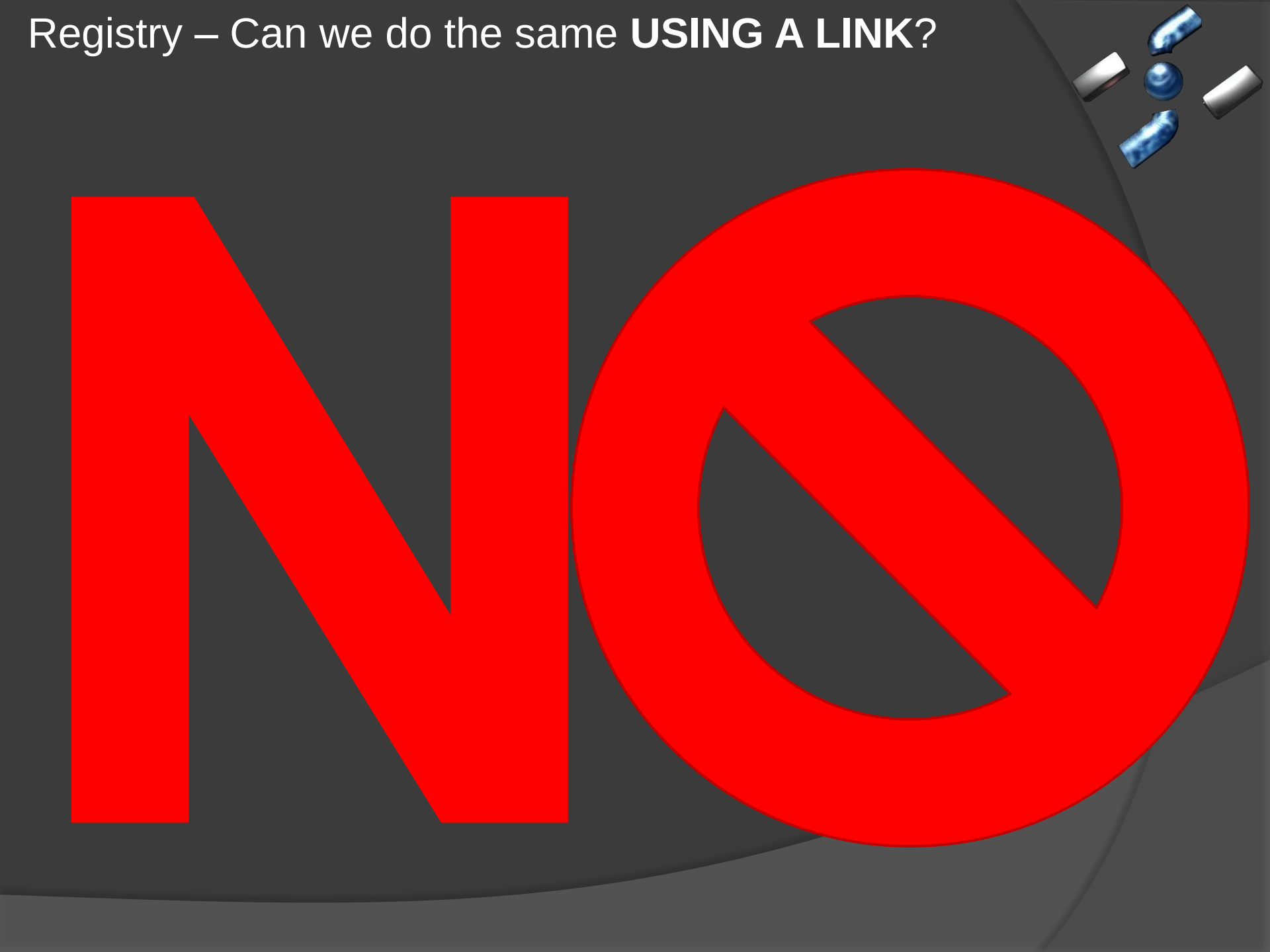
It depends in other case



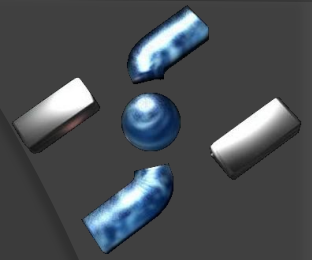
Registry – Can we write data to another hive or read protected keys there?



# Registry – Can we do the same **USING A LINK**?



# Registry – Do we know someone who can?



**Ehm, yeah, sure…**
**Any admin-level user or a process he owns**
**Or any SYSTEM process**

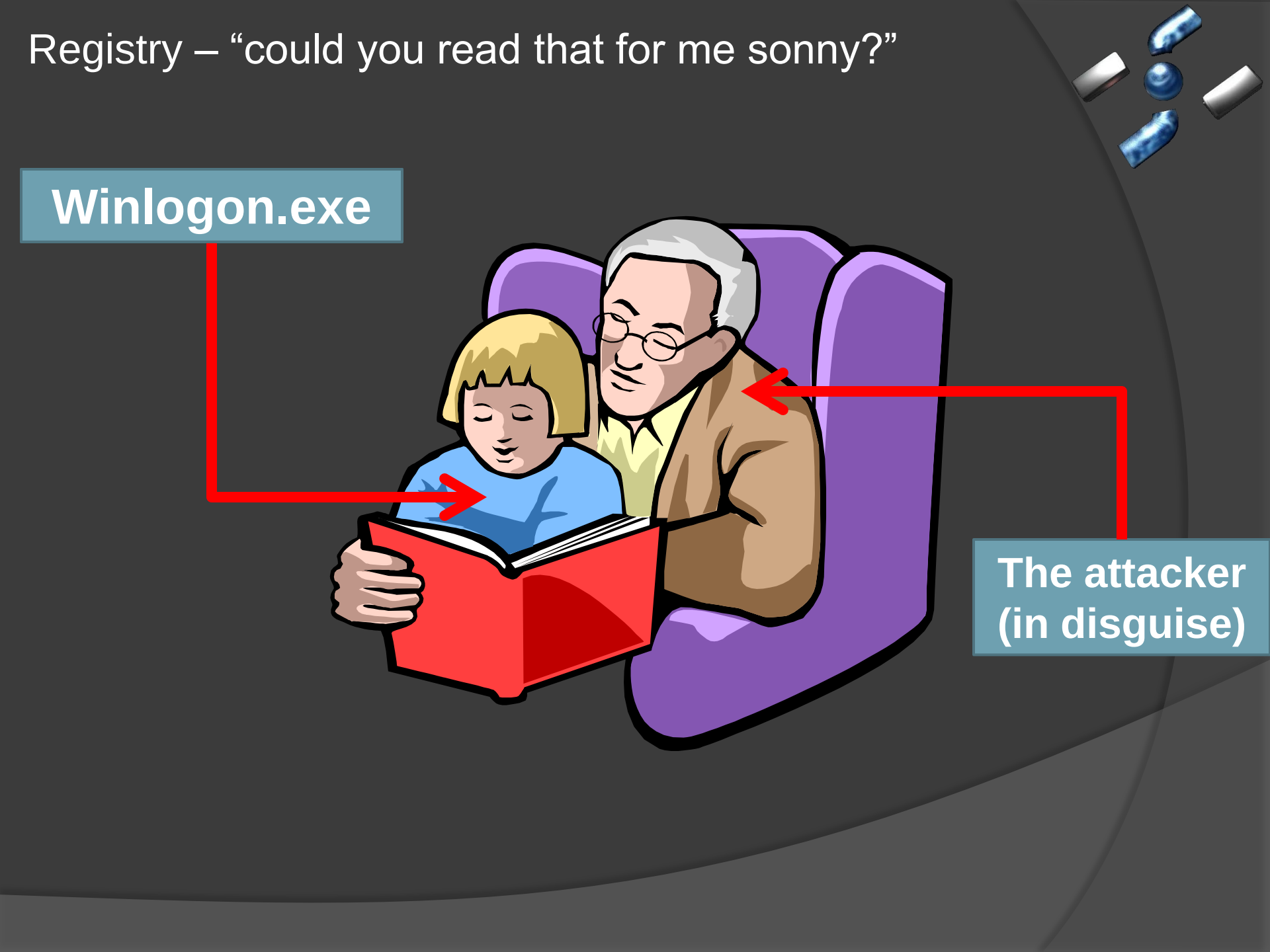
↓

**But why would they???**
(… disclose the data or overwrite sth)

↓

**Because we TRICK them into doing it!**
**(using registry links of course ☺)**

# Registry – “could you read that for me sonny?”



# Registry – “could you read that for me sonny?”

**Winlogon.exe**

Handles the user logging in

Fetches the environment variables from the registry

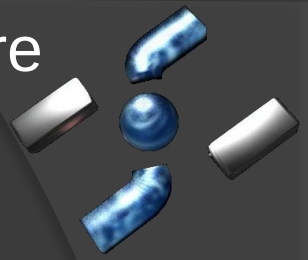
One of the things it does…

Let’s redirect this!

**HKEY_CURRENT_USER**\Environment\

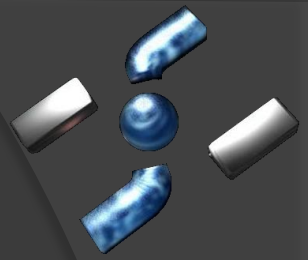
Presenting:
The vulnerability!
☺

```
15:38:00 gynvael >set
ALLUSERSPROFILE=C:\ProgramData
APPDATA=C:\Users\gynvael\AppData\Roaming
AVR32_HOME=d:\bin\avr
CommonProgramFiles=C:\Program Files (x86)\Common Files
CommonProgramFiles(x86)=C:\Program Files (x86)\Common Files
CommonProgramW6432=C:\Program Files\Common Files
COMPUTERNAME=EMERALD
```



# DEMO 10 again ☺

# Registry Link Cross-Hive Local Elevation of Privileges



**CVE-2010-0237**
**2 in 1**

Can we use this to write something someplace?

# DEMO 11

# Registry Link Priv. Escal. – How?

**Winlogon.exe**

Handles the user logging in

Setup the paths to My Documents
Desktop, Send To, etc.

Another thing it does…

Let's redirect this!

**HKEY_CURRENT_USER**\Software\Microsoft\Windows\
CurrentVersion\Explorer**Shell Folders**

Presenting:
The vulnerability!
☺

| | | |
|---|---|---|
| Local AppData | REG_SZ | C:\Users\gynv |
| My Music | REG_SZ | C:\Users\gynva |
| My Pictures | REG_SZ | C:\Users\gynvael\Pictures |
| My Video | REG_SZ | C:\Users\gynvael\Videos |

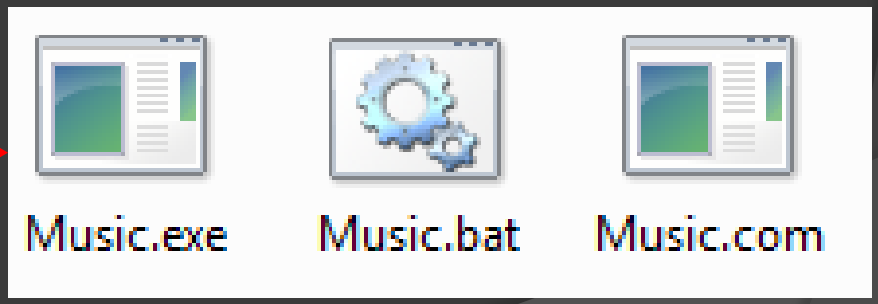
# Registry Link Priv. Escal. – Where do we redirect?

**HKEY_CURRENT_USER**\Software\Microsoft\Windows\CurrentVersion\Explorer**Shell Folders**

| | | |
|---|---|---|
| Local AppData | REG_SZ | C:\Users\gynvael\AppData\Local |
| My Music | REG_SZ | C:\Users\gynvael\Music |
| My Pictures | REG_SZ | C:\Users\gynvael\Pictures |
| My Video | REG_SZ | C:\Users\gynvael\Videos |

*"Run C:\Users\Attacker\Music??? It will just open the view of that folder lol"*

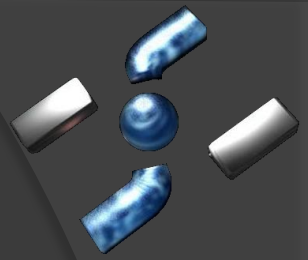
Thank you Microsoft!




**HKEY_USERS*AdminsSID***Software\Microsoft\Windows\CurrentVersion**\Run**

Registry Link Cross-Hive Local Elevation of Privileges

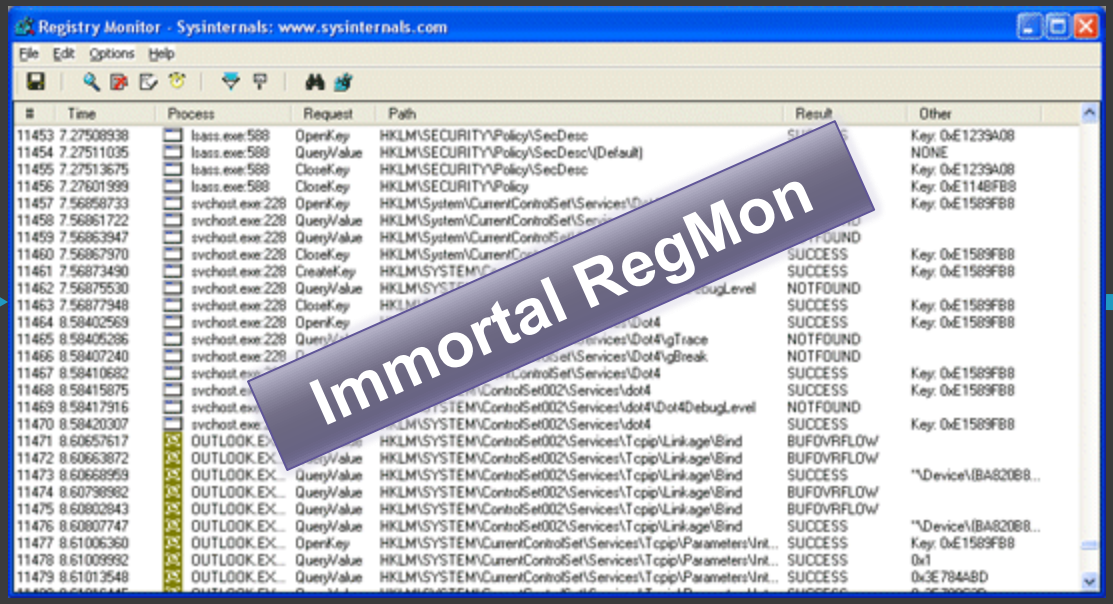
**CVE-2010-0237**
**2 in 1**



# DEMO 11
# Again

# Summary & random thoughts

CSRSS Local Elevation of Privileges

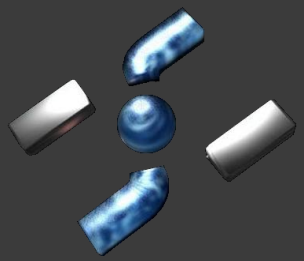



Registry Link Cross-Hive Registry Information Disclosure

Registry Link Cross-Hive Local Elevation of Privileges

# Contact

Matthew “j00ru” Jurczyk
http://j00ru.vexillium.org/
mailto: j00ru@vexillium.org



HISPASEC
http://hispasec.com/
http://virustotal.com/

Gynvael Coldwind
http://gynvael.coldwind.pl/
mailto: gynvael@coldwind.pl