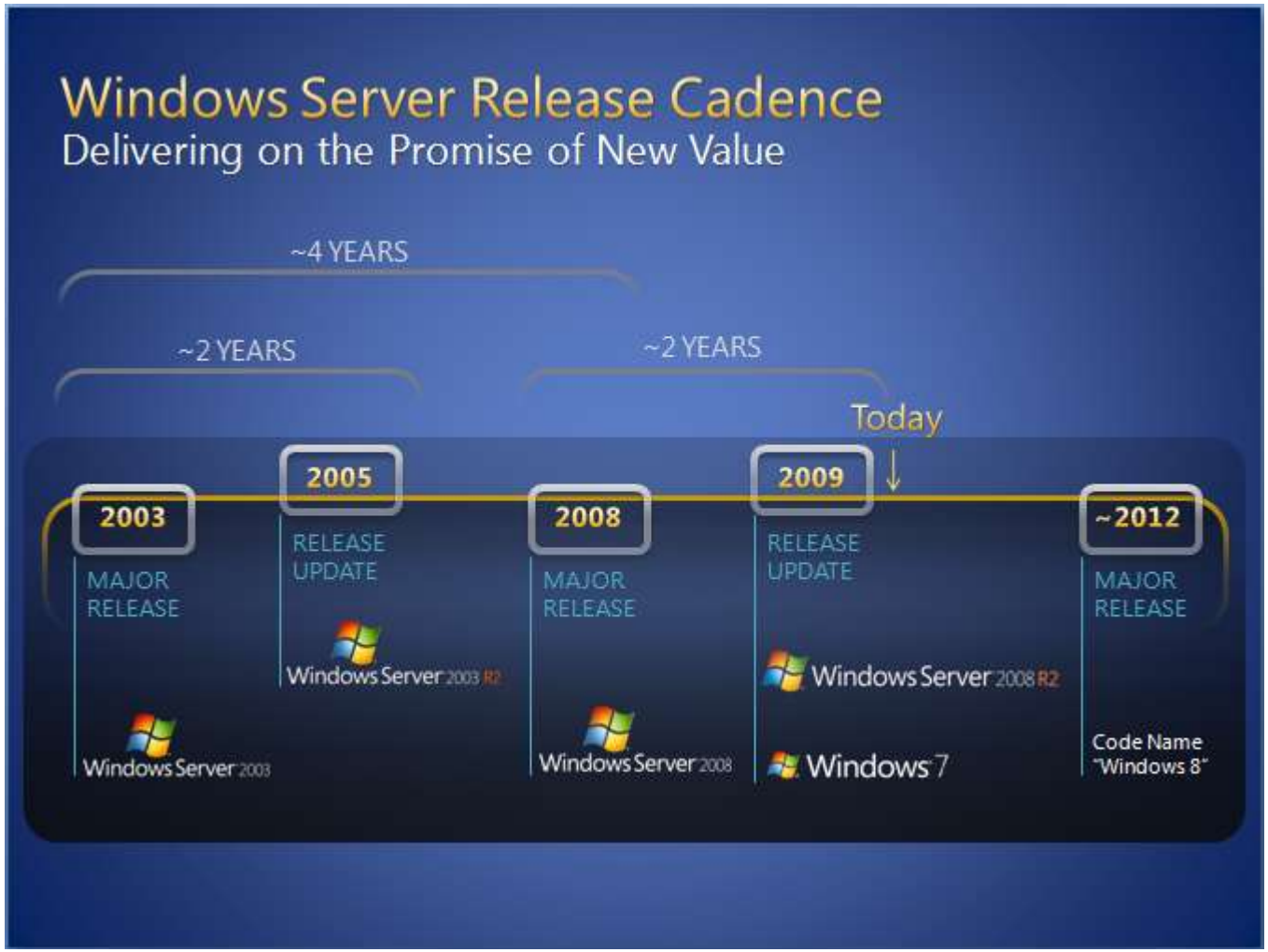
2010: The Year of the Exploit

Juraj Malcho (malcho@eset.sk)
Alexandr Matrosov
Eugene Rodionov
David Harley
Liam O'Murchu (Symantec)

Microsoft – Windows Server Releases Roadmap



The picture courtesy of Microsoft

Microsoft – Significant vulnerabilities „roadmap“

MS**03-026 –** Buffer Overrun In RPC Interface (Blaster)

MS**04-011 –** LSASS Vulnerability (Rbot**...)**

MS**06-014 –** MDAC exploit (the base of Exploit Packs)

MS**07-017 –** Windows Animated Cursor Remote Code Execution Vulnerability

MS**08-067 –** Vulnerability in Server Service (Conficker)

MS**10-046 –** Vulnerability in Windows Shell (Stuxnet)

Microsoft Security Advisory (**2269637) – Insecure Library Loading Could Allow Remote Code Execution**

MS**10-046** – "LNK exploit"

Windows Shell vulnerability

Discovered in the wild as a **0**-day

Out-of-band patch released on August **2nd 2010**

Affects all Windows versions
Spreading (not only) via removable devices regardless of security settings

MITRE code CVE-**2010-2568**

Win32/Stuxnet

Win**32**/Stuxnet

VirusBlokAda identified Stuxnet and the LNK exploit on June **17th** (Trojan-Spy.**04850**)
Microsoft and others only took a notice a month later

Realtek Semiconductors notified on June **24th**
regarding the certificate problem
Allegedly, the notification was ignored

July 13th – The Moment of Truth
Win32/Rootkit.TmpHider
July 6th 2010: Win32/Rootkit.Agent.NTK

Gradual unfurling of the truth about Stuxnet
At first seemed to be spyware
Only in September was it found to be a tool of destruction

Win32/Stuxnet – what's so special?

Targeted attack
Not only an eye-opener for the general public, but even for many in the IT security industry

Uncompromisingly professional
Created by a team of people

0-day vulnerability portfolio
4 0-day vulnerabilities: MS10-046, MS10-061, MS10-073, MS10-0XX + MS08-067

Signed!
Compromised Realtek & JMicron certificates

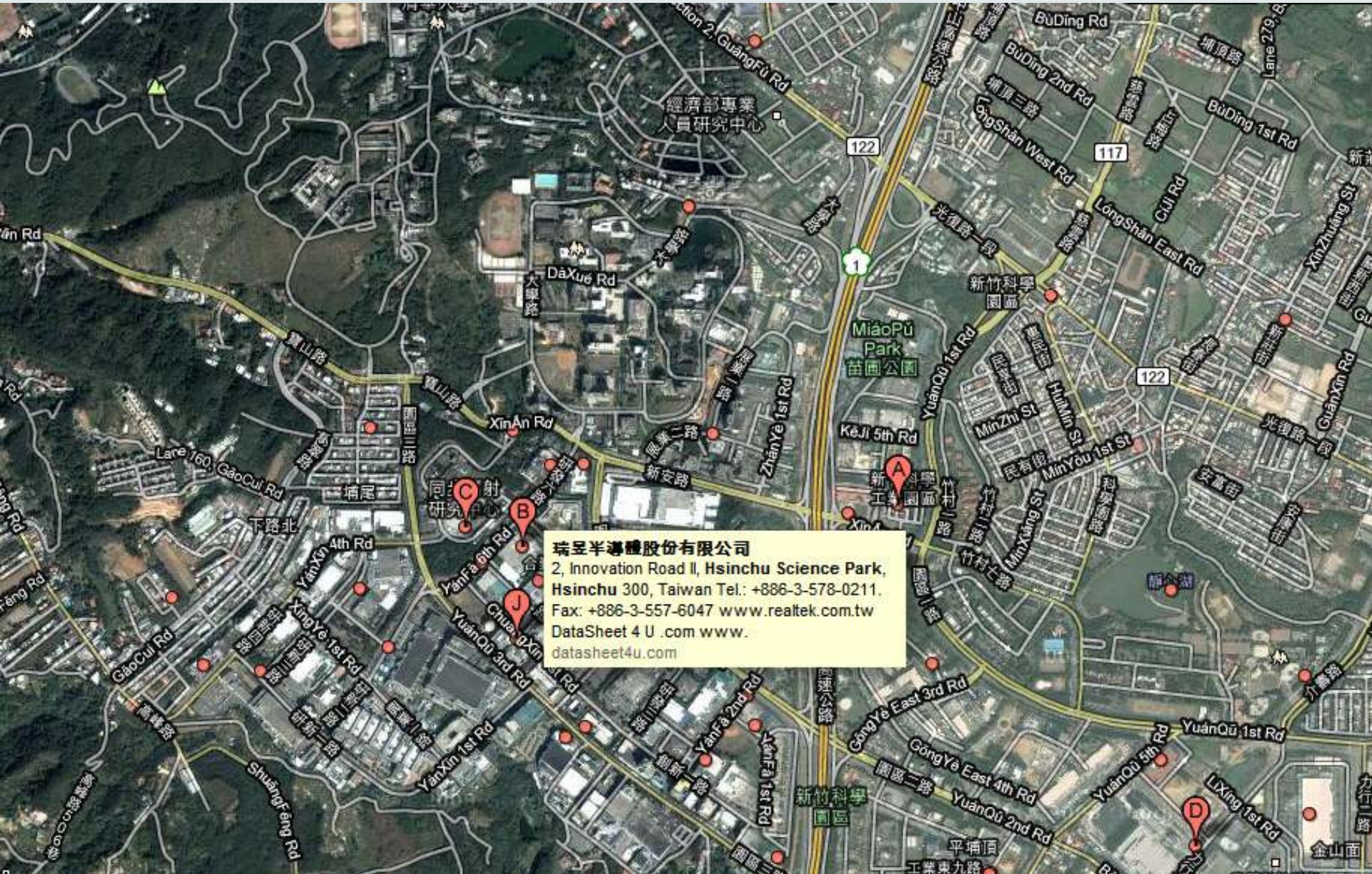
Weeks of exhaustive analysis
The effect on Siemens Simatic SCADA SW
Speculation about other possible targets

Win32/Stuxnet – invisible

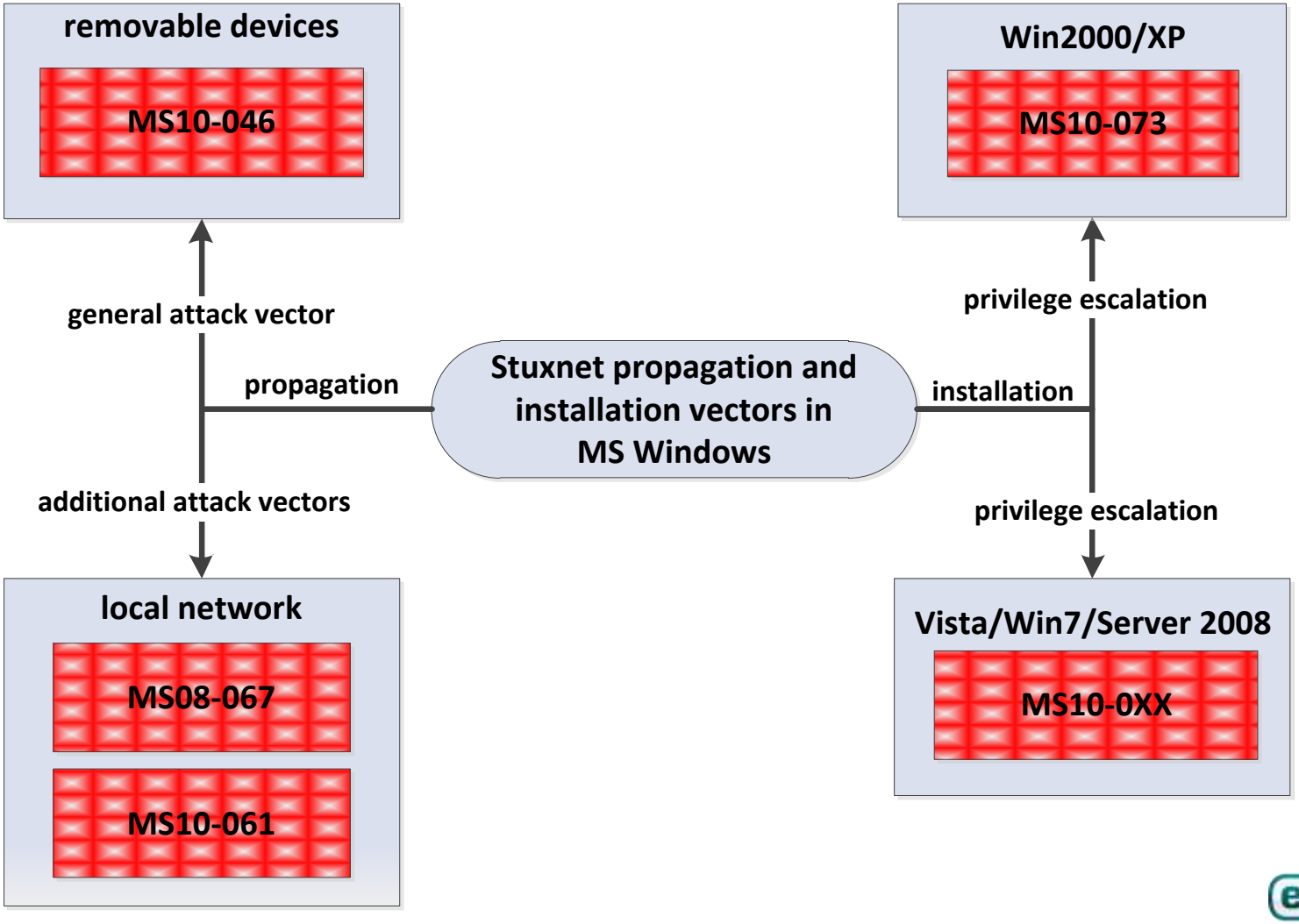
First variants January/March/June **2009**
Vulnerability arsenal was limited by then:
MS08-067
MS10-061
MS08-025 (win32k.sys!NtUserMessageCall)
autorun.inf

Significant upgrade in January **2010**
Another driver added
Signed by Realtek Technologies certificate
New 0-day vulnerabilities added:
MS10-046, MS10-061, MS10-073, MS10-0XX

Win32/Stuxnet – signatures



Win32/Stuxnet – vulnerabilities



Win32/Stuxnet – exploit #0: MS08-067

netapi32.dll!NetPathCanonicalize

```
\\remote\hello\.\you\..\world\
```

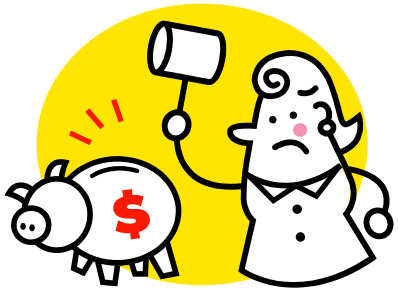
```
\\remote\hello\you\..\world\
```

```
\\remote\hello\world\
```

Win32/Stuxnet – exploit #0: MS08-067

netapi32.dll!NetPathCanonicalize

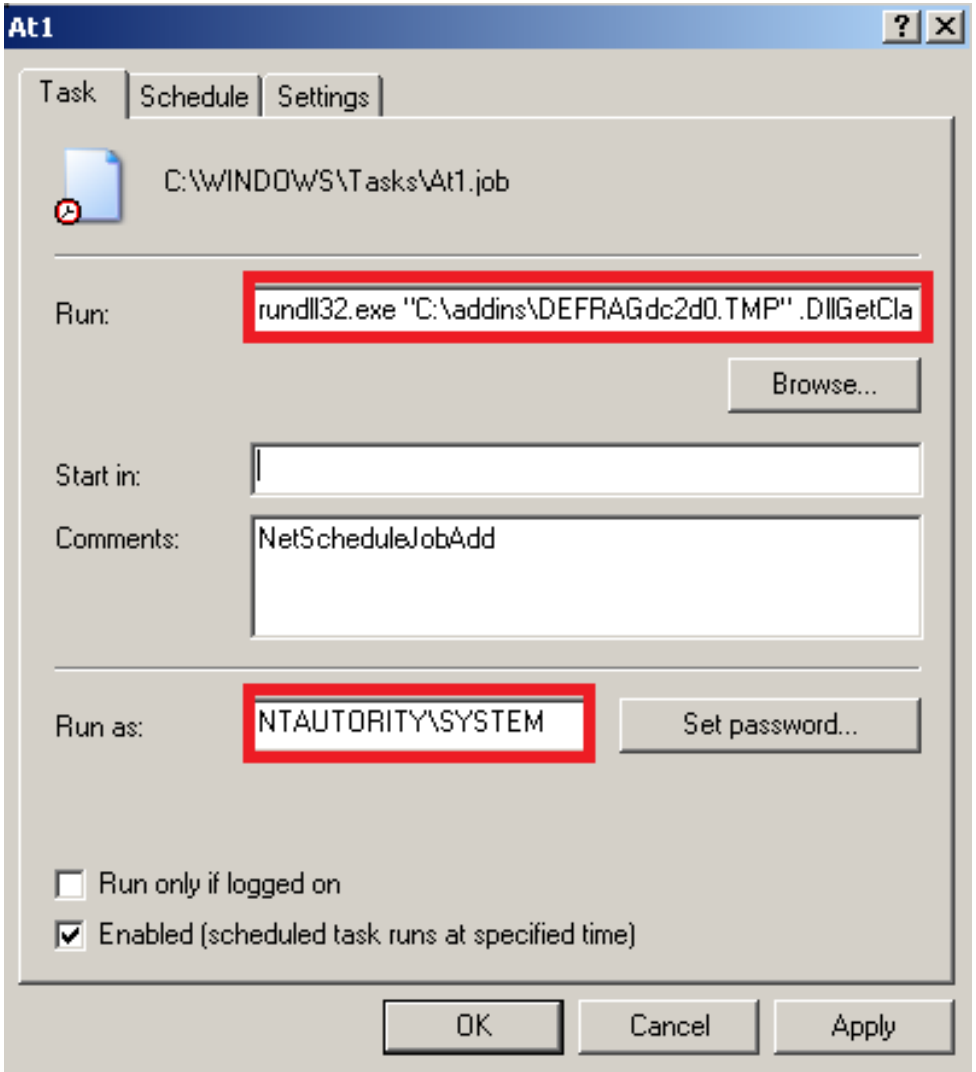
```
\\remote\..\..\hello_world\
```



```
..\hello_world\
```

Win32/Stuxnet – exploit #0: MS08-067

c$ and admin$ shares scan



Win32/Stuxnet – exploit #1: MS10-0XX

A vulnerability in Task Scheduler service

Scheduled tasks integrity checking problem

Used for privilege escalation

Windows Vista and above

```
- <Principals>
  - <Principal id="LocalSystem">
      <UserId>S-1-5-18</UserId>
      <RunLevel>HighestAvailable</RunLevel>
    </Principal>
  </Principals>
- <Actions Context="LocalSystem">
  - <Exec>
      <Command>C:\WINDOWS\NOTEPAD.EXE</Command>
      <Arguments />
    </Exec>
  </Actions>
</Task>
```

Win32/Stuxnet – exploit #2: MS10-073

Address	Size	Owner	Section	Contains	Type	Access	Initial
009E0000	00038000	009E0000 (itself)			Priv 00021040	RWE	RWE
10000000	00001000	RC_Data_ 10000000 (itself)		PE header	Imag 01001002	R	RWE
10001000	00004000	RC_Data_ 10000000	.text	code	Imag 01001002	R	RWE

```
60636261  E8 01000000     CALL 60636267
60636266  0059 F0         ADD BYTE PTR DS:[ECX-10],BL
60636269  66:0FBA29 00    BTS WORD PTR DS:[ECX],0
6063626E  72 1D           JB SHORT 6063628D
60636270  53              PUSH EBX
60636271  E8 04000000     CALL 6063627A
60636276  0000            ADD BYTE PTR DS:[EAX],AL
60636278  8600            XCHG BYTE PTR DS:[EAX],AL
6063627A  5B              POP EBX
6063627B  8B13            MOV EDX,DWORD PTR DS:[EBX]
6063627D  FF7424 04       PUSH DWORD PTR SS:[ESP+4]
60636281  FF7424 14       PUSH DWORD PTR SS:[ESP+14]
60636285  8BC2            MOV EAX,EDX
60636287  FFD0            CALL EAX
60636289  C603 00         MOV BYTE PTR DS:[EBX],0
6063628C  5B              POP EBX
6063628D  33C0            XOR EAX,EAX
6063628F  C2 0C00         RETN 0C
```

Win32/Stuxnet – exploit #3: MS10-061

A vulnerability in Printer Spooler

Shared printers problem
"Known" since 2009/04

Used to spread over the network

All Windows versions vulnerable

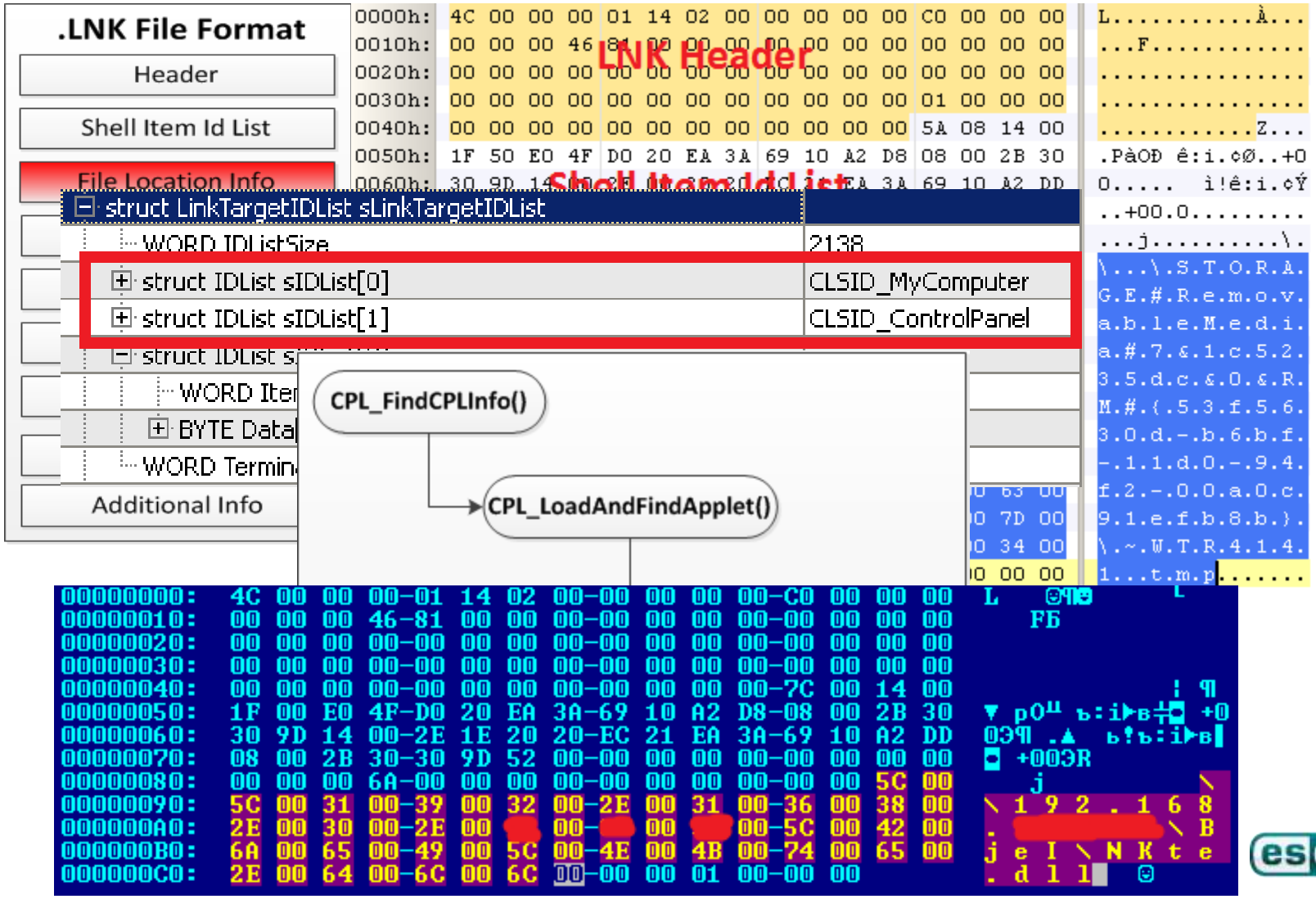
A problem in verifying the identity of the printing client

Instead of being sent to a printer files are dropped to:
%SYSTEM32% (privileged operation):
Windows\System32\winsta.exe and

Printer Document View Help

Document Name	Status	Owner	Pages	Size	Submitted	Port
Default	printing	Guest	n/a	502 KB	22:15:43 16.09.2010	winsta.exe
Default		Guest	n/a		22:15:44 16.09.2010	wbem\mof\sysnullevnt.mof

Win32/Stuxnet – exploit #4: MS10-046



Win**32**/Stuxnet **–** exploit #**4:** MS**10-046**

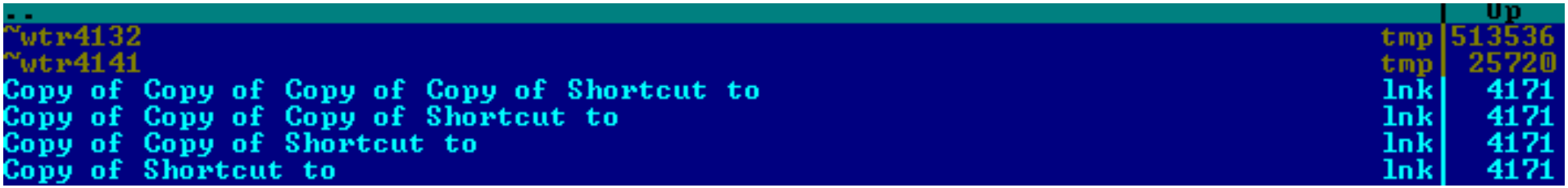
4 ways of storing the path to the payload:

\\.\STORAGE#Volume#_??_USBSTOR#Disk&Ven_____USB&Prod_FLASH_DRIVE&Rev_#1234500010000000173&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b**}\~WTR4141.tmp**

\\.\STORAGE#Volume#1&19f7e59c&0&_??_USBSTOR#Disk&Ven_____USB&Prod_FLASH_DRIVE&Rev_#1234500010000000173&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b**}\~WTR4141.tmp**

\\.\STORAGE#RemovableMedia#8&1c5235dc&0&RM#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b**}\~WTR4141.tmp**

\\.\STORAGE#RemovableMedia#7&1c5235dc&0&RM#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b**}\~WTR4141.tmp**



Win32/Stuxnet – User mode functionality

Large DLL is the main body

Everything else (including kernel mode drivers) in the resources

Name	Virtual S	aw Size	Characteristics
00000200	000002	000021	00000224
Byte[8]	Dword	word	Dword
.text	000539	053A0	60000020
.rdata	00011A	011C0	E0000040
.data	00003D	003400	C0000040
.xdata	000113	011400	40000040
.cdata	000007	000800	C0000040
.rsrc	000A8F	0A900	40000040
.reloc	000099	009A0	42000040

RC Data

- 201
- 202
- 203
- 205
- 208
- 209
- 210
- 221
- 222
- 240
- 241
- 242
- 250

Name	Address	Ordinal
_1	100019D5	1
_2	10001AC6	2
_4	10004A3D	4
_5	1000265F	5
_6	10001B7E	6
_7	10001C10	7
_9	100027C8	9
_10	10002AF6	10
_14	10002166	14
_15	10002735	15
_16	10002CA9	16
_17	10002DFB	17
_18	10004ADA	18
_19	10002353	19
_22	10001C15	22
_24	10003579	24
_27	10001CA2	27
_28	10003602	28
_29	1000368B	29
_31	10002926	31
_32	10001A4E	32
DllEntryPoint	10042AA6	

original

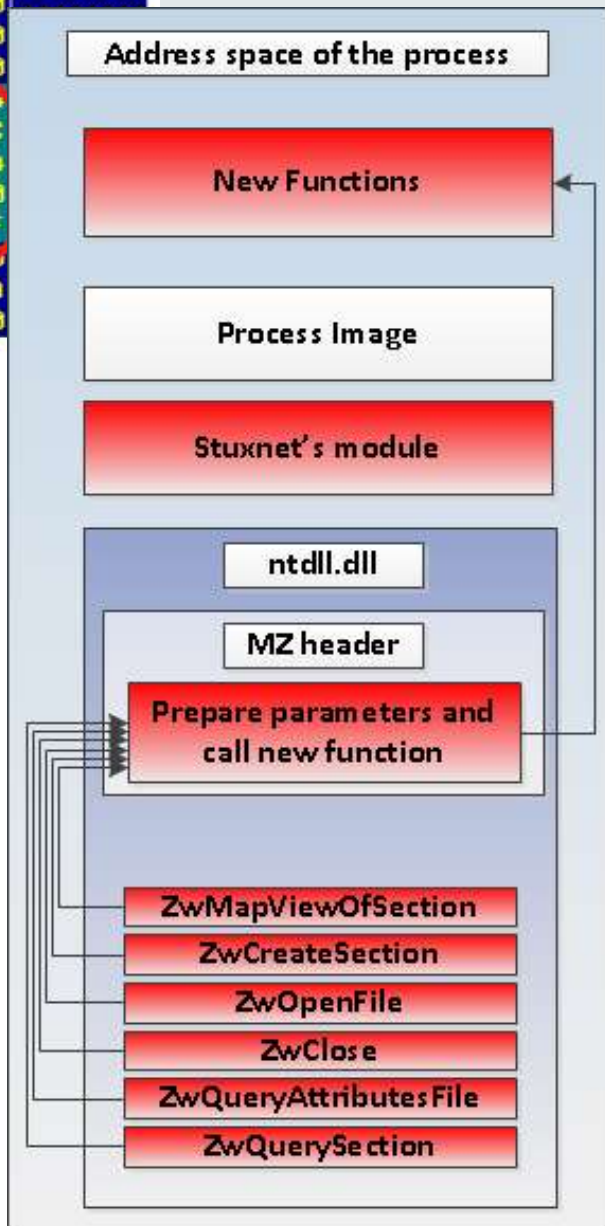
```
7C900000 4D 5A 90 00 03 00 00 00 MZh.Ъ...
7C900008 04 00 00 00 FF FF 00 00 Ъ...яя..
7C900010 B8 00 00 00 00 00 00 00 ё.......
7C900018 40 00 00 00 00 00 00 00 @.......
7C900020 00 00 00 00 00 00 00 00 ........
7C900028 00 00 00 00 00 00 00 00 ........
7C900030 00 00 00 00 00 00 00 00 ........
7C900038 00 00 00 00 E0 00 00 00 ....a...
7C900040 0E 1F BA 0E 00 B4 09 CD ЪЪєЪ.г.H
7C900048 21 B8 01 4C CD 21 54 68 !ёЪLH!Th
7C900050 69 73 20 70 72 6F 67 72 is progr
7C900058 61 6D 20 63 61 6E 6E 6F am canno
7C900060 74 20 62 65 20 72 75 6E t be run
7C900068 20 69 6E 20 44 4F 53 20  in DOS
7C900070 6D 6F 64 65 2E 0D 0D 0A mode....
7C900078 24 00 00 00 00 00 00 00 $.......
```

pathced

```
7C900000 4D 5A 90 00 03 00 00 00 MZh.Ъ...
7C900008 04 00 00 00 FF FF 00 00 Ъ...яя..
7C900010 B8 00 00 00 00 00 00 00 ё.......
7C900018 40 00 00 00 00 00 00 00 @.......
7C900020 00 00 00 00 00 00 00 00 ........
7C900028 00 00 00 00 00 00 00 00
7C900030 00 00 00 00 00 00 00 00
7C900038 00 00 00 00 E0 00 00 00
7C900040 3B 10 49 AB B2 00 EB 14
7C900048 B2 01 EB 10 B2 02 EB 0C
7C900050 B2 03 EB 08 B2 04 EB 04
7C900058 B2 05 EB 00 52 E8 04 00
7C900060 00 00 F2 00 AC 00 5A FF
7C900068 20 69 6E 20 44 4F 53 20
7C900070 6D 6F 64 65 2E 0D 0D 0A
7C900078 24 00 00 00 00 00 00 00
```

disassembled

```
ZwMapViewOfSectionHandler:
                mov     dl, 0
                jmp     short loc_1004966C
; ---------------------------------------
ZwCreateSectionHandler:
                mov     dl, 1
                jmp     short loc_1004966C
; ---------------------------------------
ZwOpenFileHandler:
                mov     dl, 2
                jmp     short loc_1004966C
; ---------------------------------------
ZwCloseHandler:
                mov     dl, 3
                jmp     short loc_1004966C
; ---------------------------------------
ZwQueryAttributesFilleHandler:
                mov     dl, 4
                jmp     short loc_1004966C
; ---------------------------------------
ZwQueryHandler:
                mov     dl, 5
                jmp     short $+2
loc_1004966C:
                push    edx
                call    JmpToNewFunction
```



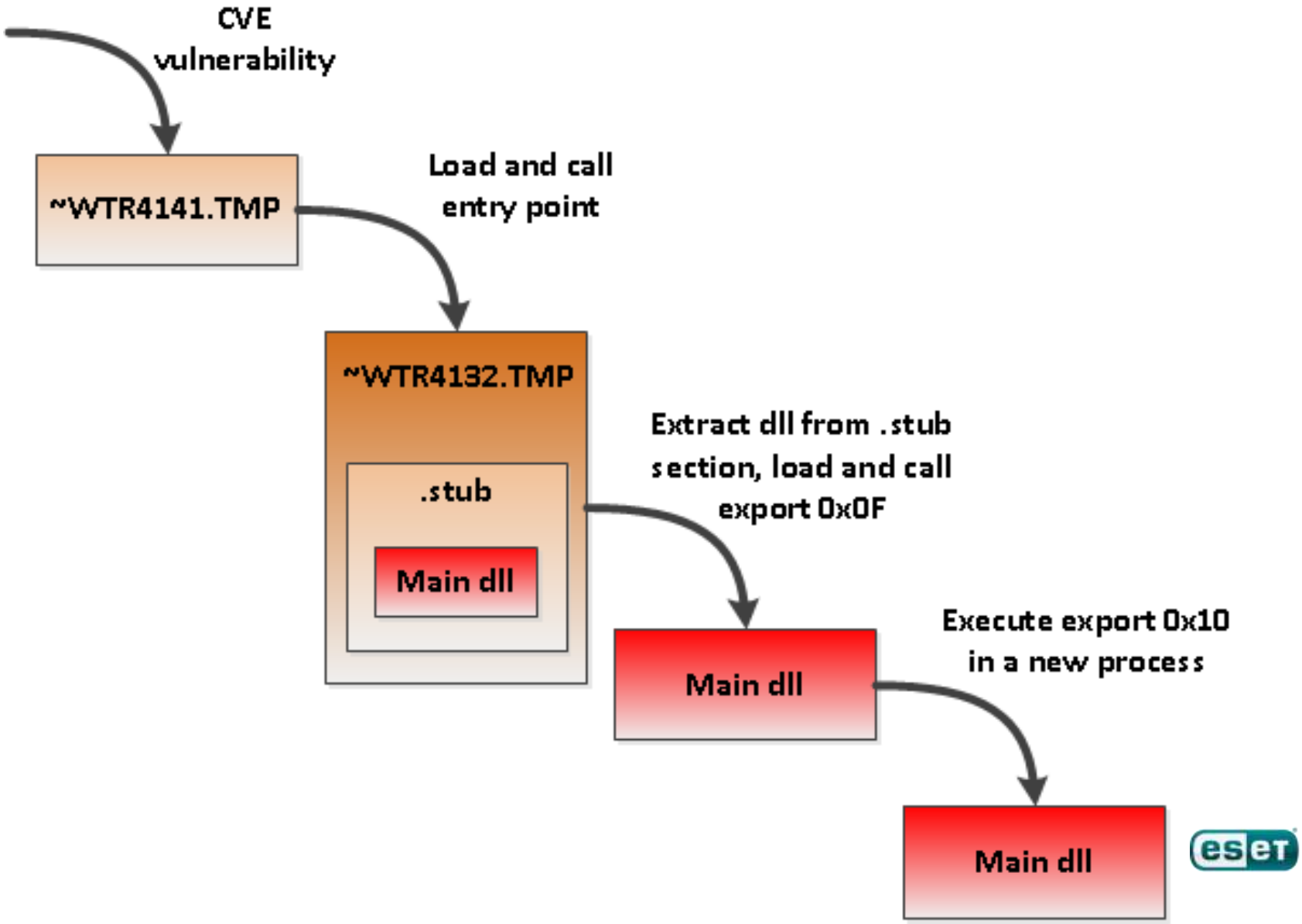
Code injection (new processes)

1) Creates a host process

2) Replaces the image with the code to load specified module/export (as in previous scenario)

Host processes:
- lsass.exe (system process)
- avp.exe (Kaspersky)
- mcshield.exe (McAfee VirusScan)
- avguard.exe (AntiVir Personal Edition)
- bdagent.exe (BitDefender Switch Agent)
- UmxCfg.exe (eTrust Configuration Engine from CA)
- fsdfwd.exe (F-Secure Anti-Virus suite)
- rtvscan.exe (Symantec Real Time Virus Scan service)
- ccSvcHst.exe (Symantec Service Framework)
- ekrn.exe (ESET Antivirus Service Process)
- Tmproxy.exe (PC-cillin / TrendMicro)

Installation – ~WTR4141.TMP



Win32/Stuxnet – the exports (1)

Export #2

Called in address space of the process with name s7tgtopx.exe and CCProjectMgr.exe

Hooks monitor opening files with the extension .S7P & .MCP

Siemens Simatic Step7 software

Export #5

Checks whether the kernel-mode driver MrxCls.sys is properly installed in the system

Export #6

Return current version of Stuxnet installed

Win32/Stuxnet – the exports (2)

Export #9, #31

Builds Stuxnet's dropper from the files located in the system and runs it:

- %Dir%\XUTILS\listen\XR000000.MDX
- %Dir%\XUTILS\links\S7P00001.DBF
- %Dir%\XUTILS\listen\S7000001.MDX

Export #18

Completely removes the malware from the system and perform full cleanup

Win32/Stuxnet – the exports (3)

Export #16

Installs the malware's components:

- Drops and installs kernel-mode drivers: MrxNet.sys and MrxCls.sys
- Drops the main dll in %SystemRoot%\inf\oem7A.PNF
- Drops Stuxnet's configuration data in %SystemRoot%\inf\mdmcpq3.PNF
- Creates tracing file in %SystemRoot%\inf\oem6C.PNF
- Drops data file in %SystemRoot%\inf\mdmeric3.PNF
- Injects the main dll into services.exe process and executes the function exported as ordinal 32
- Injects the main dll into the s7tgtopx.exe process if any exists, and executes exported function 2 there

Win32/Stuxnet – the exports (4)

Export #17

Replaces s7otbxdx.dll with a malicious DLL; original library renamed to s7otbxdx.dll

Wrapper plus hooks 16 functions:

- s7_event
- s7ag_bub_cycl_read_create
- s7ag_bub_read_var
- s7ag_bub_write_var
- s7ag_link_in
- s7ag_read_szl
- s7ag_test
- s7blk_delete
- s7blk_findfirst
- s7blk_findnext
- s7blk_read
- s7blk_write
- s7db_close
- s7db_open
- s7ag_bub_read_var_seg
- s7ag_bub_write_var_seg

Win32/Stuxnet – the exports (5)

Export #19

Prepares the files to propagate through USB flash drives:
Copy of Shortcut to.lnk
Copy of Copy of Shortcut to.lnk
Copy of Copy of Copy of Shortcut to.lnk
Copy of Copy of Copy of Copy of Shortcut to.lnk
~WTR4141.TMP
~WTR4132.TMP

Export #22

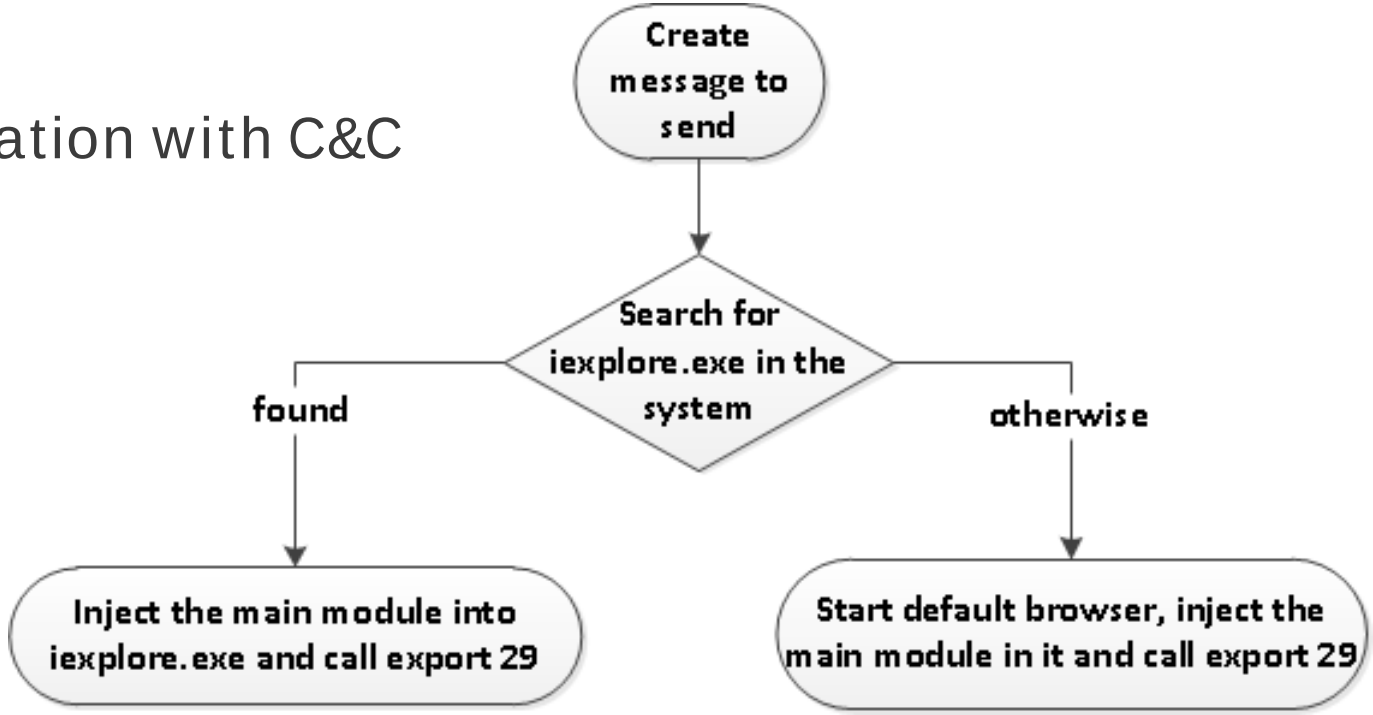
Network distribution (via exploits) + RPC-based communication

Export #27

Implements RPC server to handle remote calls

Win32/Stuxnet – the exports (6)

Export #28
Communication with C&C



Export #29
Data exchange with C&C – send data/receive a binary to execute

Win32/Stuxnet – the exports (7)

Export #32

Starts the RPC server (must be called from services.exe)
Monitors WM_DEVICE_CHANGE
Can drop or remove files from removables

Win32/Stuxnet – RPC features

RpcProc1 – Returns the version of the worm
RpcProc2 – Loads a module passed as a parameter into a new process and executes specified exported function
RpcProc3 – Loads a module passed as a parameter into the address of the process executing this function and calls its exported function number 1
RpcProc4 – Loads a module passed as a parameter into a new process and executes it
RpcProc5 – Builds the worm dropper
RpcProc6 – Runs the specified application
RpcProc7 – Reads data from the specified file
RpcProc8 – Writes data into the specified file
RpcProc9 – Deletes the specified file
RpcProc10 – Works with the files of which the names are intercepted by hooks set up in function number 2 and writes information in tracing file

Win32/Stuxnet – Resources

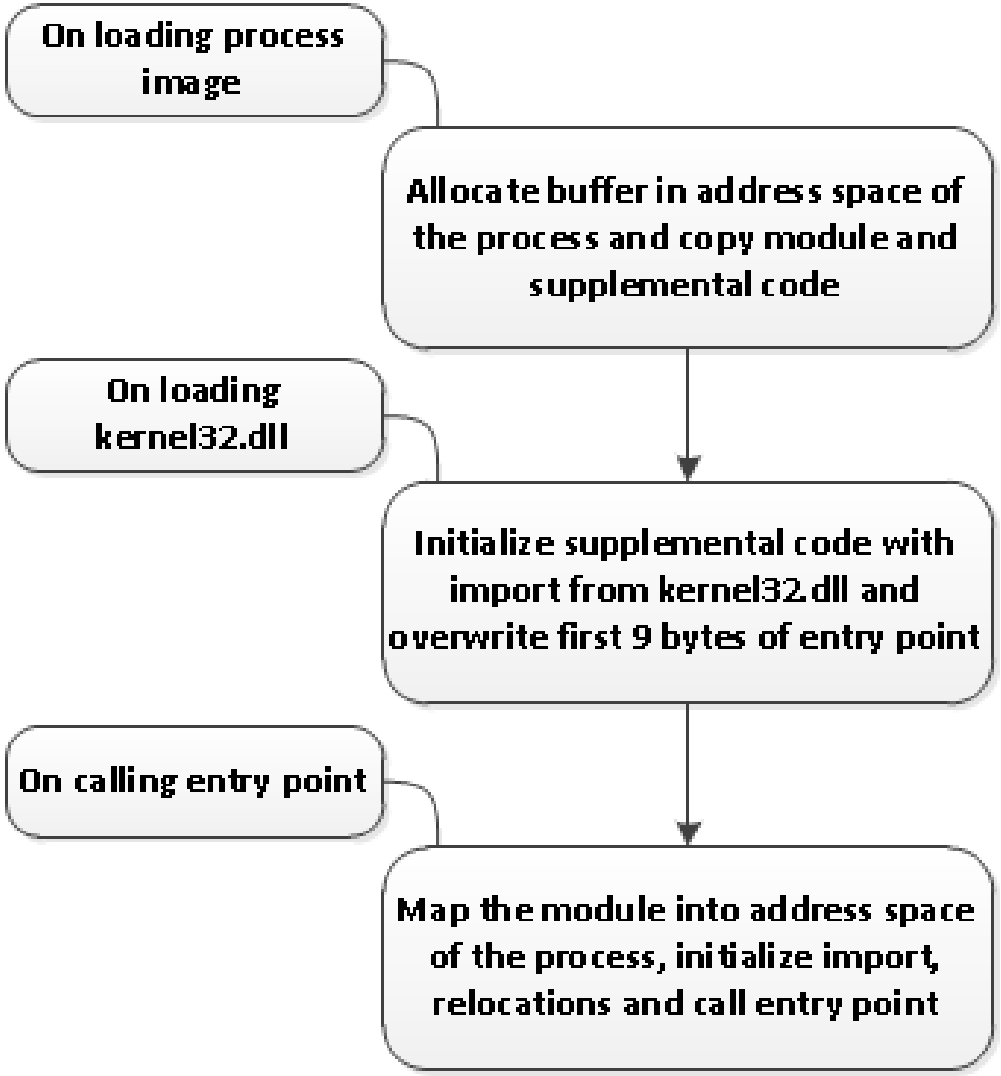
Resource ID	Description
201	Kernel-mode driver (MrxCls.sys) responsible for injecting code into certain processes
202	A proxy dynamic link library
203	A .cab file with dynamic link library inside
205	Configuration data for MrxCls.sys
208	A dynamic link library – fake s7otbldx.dll (Siemens SCADA module)
209	Encrypted data file drop to %WINDIR%\help\winmic.fts
210	Template PE-file, used to construct dropper (~WTR4132.TMP)
221	Module used for distribution of the worm by exploiting RPC vulnerability
222	Module used for distribution of the worm by exploiting MS10-061 vulnerability
240	.LNK file template, used to create .LNK files exploiting vulnerability
241	~WTR4141.TMP – dynamic link library, used to load dropper (~WTR4132.TMP) while infecting system
242	Kernel-mode driver (MrxNet.sys) responsible for concealing files exploiting LNK vulnerability and infecting system
250	Module used to escalate privileges by exploiting 0-day vulnerability in Win32k.sys

Win32/Stuxnet – Kernel mode functionality

Digitally signed drivers
Rootkit functionality

MRXCLS.SYS
Injector/Stealthy export calls

MRXNET.SYS
Files hiding



Win32/Stuxnet – bot config data

%WINDIR%\inf\mdmcpq3.pnf

Encrypted, 1860 bytes

- URLs of C&C servers
- Activation time – the time and date after which the worm is active
- Deactivation time – the time after which the worm becomes inactive and deletes itself
- Version of the malware
- The minimum quantity of files that the removable drive should contain to drop malicious .LNK files successfully
- Other information about its propagation and functioning

C&C

- www.mypremierfutbol.com
- www.todaysfutbol.com

http://www.mypremierfutbol.com/index.php?data=data-to-send

Win32/Stuxnet – PLCs

Programmable Logic Controller

Monitors Input and Output lines

Sensors on input
Switches/equipment on output
Many vendors

Stuxnet seeks specific models

s7-300 & s7-400

Win32/Stuxnet – HW Configuration

PLC config stored in System Data Blocks
Stuxnet parses these blocks

Looks for magic bytes **2C CB 00 01** at offset **50**h
Signifies a Profibus network card attached – CP 342-5

Looks for **7050h and 9500h**
Must have more than 33 of these values

Injects different code based on number of occurences

Win32/Stuxnet – Step7, STL, MC7

Simatic or Step7 software
Used to write code in STL or other languages

STL is compiled to MC7 byte code
MC7 byte code is transferred to the PLC

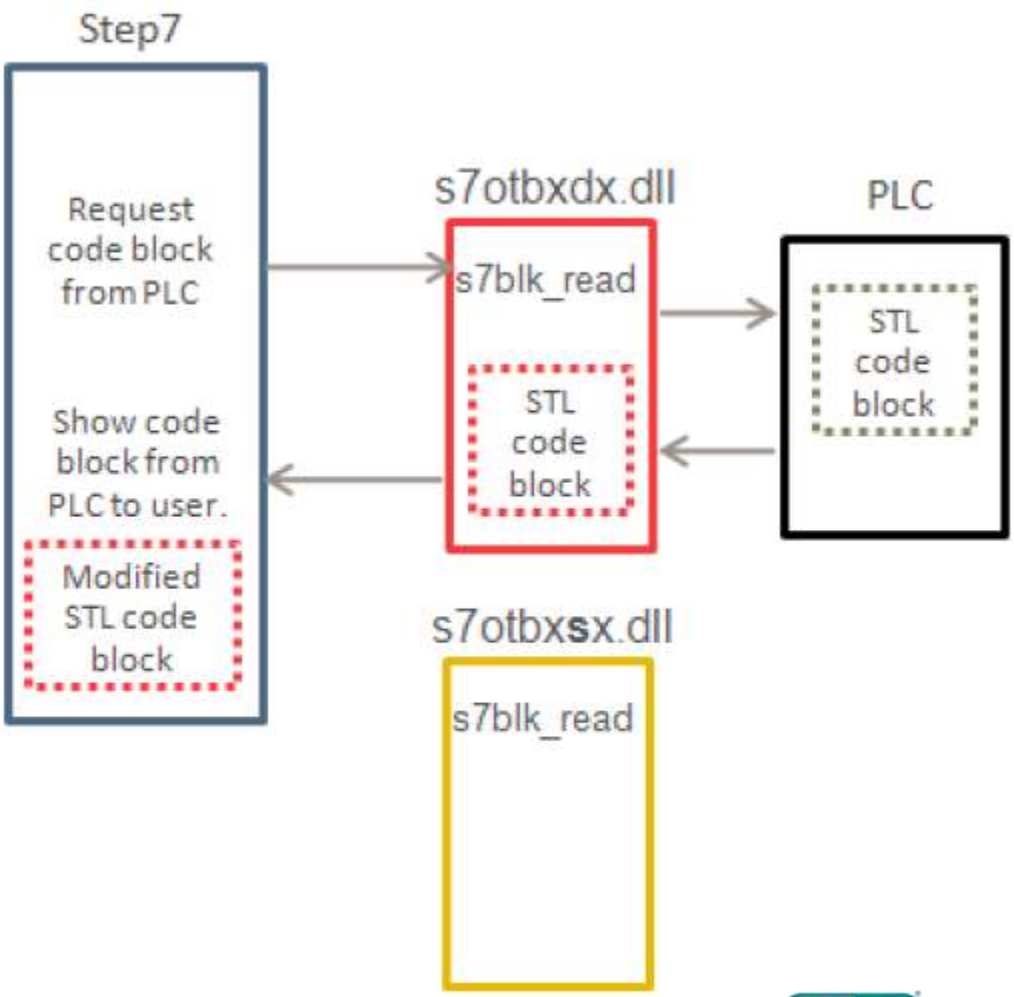
MC7 code is transferred to the PLC
Control PC can now be disconnected

Win32/Stuxnet – Man in the Middle

Step7 uses a library to access the PLC
S7otbxdx.dll

Stuxnet replaces the DLL with its own version
Replaces s7otbxdx.dll with a malicious DLL; original library renamed to s7otbxsx.dll

Stuxnet intercepts reads and writes to the PLC and changes the code at this point



Win32/Stuxnet – MC7 Byte Code

Stuxnet contains at least **70** binary blobs of data
Encoded and stored in fake DLL

This is the MC7 code to be injected to the PLCs
Can only be understood after being converted to STL

Even though the code is readable, still unsure what it means
Starts to make sense only on the targeted system

Win32/Stuxnet – OB1 & OB35

OB1 = main() on PLCs
Stuxnet inserts its own code at the beginning of OB1 so it runs first

OB35 is a 100 ms interrupt routine
Used to monitor inputs that require fast action
Stuxnet infects OB35 too

Stuxnet will return clean version of these functions when they are read from the PLC

Win32/Stuxnet – the real payload

Stuxnet contains hundreds lines of code

It's difficult to understand the real world actions without knowing what's connected on the inputs and outputs

```
UC FC 1865
POP
L DW#16#DEADF007
==D
BEC
L DW#16#0
L DW#16#0
```

Tampers with Frequency Converter Drivers

Sets them on low vs high (2~1410Hz) every 13/27 days

Win32/Stuxnet – facts vs speculation & myth

Frequency converter drives

Fararo Paya in Teheran, Iran
Vacon NX Finland

Such drives "are regulated for export in the US by the Nuclear Regulatory Commission," because one of their main uses is for uranium enrichment...

```
.rdata:00011D95                 db    0
.rdata:00011D96                 db    0
.rdata:00011D97                 db    0
.rdata:00011D98 aBMyrtusSrcObjf db 'b:\myrtus\src\objfre_w2k_x86\i386\guava.pdb',0
.rdata:00011DC4                 db    0
.rdata:00011DC5                 db    0
.rdata:00011DC6                 db    0
.rdata:00011DC7                 db    0
```

Government or terrorists?

Why leaving traces then? #DEADF007, 19790509, Myrtus

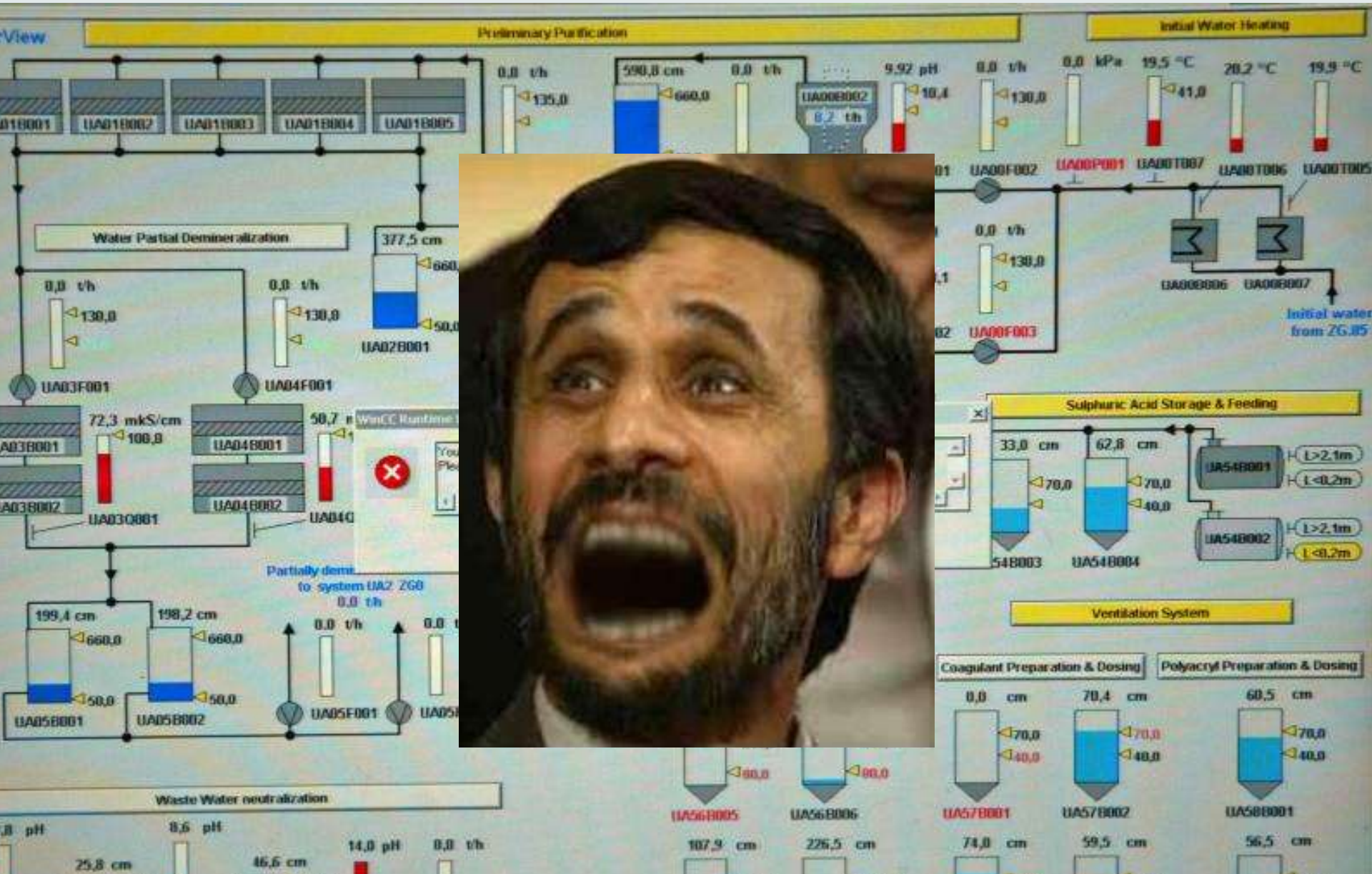
MS10-046 related malware and its evolution

8(+) malware families that got "inspired"

Malware name	first appearance ITW	LNK exploit added	signed	advanced	prevalence	targeted
LNK/Exploit.CVE-2010-2568	2010/07/16 (2008/11/20)	2008/11/20	N/A			N/A
Win32/Stuxnet	2009/01	2010/03 (2010/01?)				
Win32/Autorun.VB.{RL, RP, RT, RU, SN}	2010/07/18	2010/07/22				
Win32/Sality.NBA	2003/07/06	2010/07/24				
Win32/Agent.OTB	2010/01	2010/07/26				
Win32/TrojanDownloader.Chymine.A	2010/07/13	2010/07/26				
Win32/Delf.NVR (xbot)	2010/07/09	2010/07/27				
CN "0-day"	2010/08/02	2010/08/02				
Win32/Agent.OSW aka Dottun (fanny)	2008/02	2010/03???				

CVE-2010-2744/MS10-073 (win32k.sys) – since 2009/11!!!

Information value…



… needs to be considered and protected

Education is a necessary part of defense

Gathering data
Not so difficult online

Mining and exploring it
To find the right target

Marketing folks know this
Cyber criminals are no different

Careful what you say
Once it's online it's hard to withdraw it

Nothing comes for free
Mistrust information you didn't ask for or people you don't know

Questions?

Juraj Malcho (malcho@eset.sk)
Alexandr Matrosov
Eugene Rodionov
David Harley
Thanks to Liam O'Murchu & all Stuxnet Reverse Engineers