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

Windows Server Release Cadence
Delivering on the Promise of New Value

- ~4 YEARS
- ~2 YEARS
- ~2 YEARS

2003
MAJOR RELEASE
Windows Server 2003

2005
RELEASE UPDATE
Windows Server 2003 R2

2008
MAJOR RELEASE
Windows Server 2008

2009
RELEASE UPDATE
Windows Server 2008 R2

~2012
MAJOR RELEASE
Code Name “Windows 8”

Today

The picture courtesy of Microsoft
Microsoft – Significant vulnerabilities „roadmap“

MS03-026 – Buffer Overrun In RPC Interface (Blaster)

MS04-011 – LSASS Vulnerability (Rbot...)

MS06-014 – MDAC exploit (the base of Exploit Packs)

MS07-017 – Windows Animated Cursor Remote Code Execution Vulnerability

MS08-067 – Vulnerability in Server Service (Conficker)

MS10-046 – Vulnerability in Windows Shell (Stuxnet)

Microsoft Security Advisory (2269637) – Insecure Library Loading Could Allow Remote Code Execution
MS10-046 – “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
VirusBlokJAda 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

removable devices

- MS10-046

general attack vector

additional attack vectors

local network

- MS08-067
- MS10-061

Stuxnet propagation and installation vectors in MS Windows

- MS10-0XX

Win2000/XP

- MS10-073

privilege escalation

installation

Vista/Win7/Server 2008

- MS10-0XX
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

```cmd
rundll32.exe "C:\adins\DEFRAGdc2d0.TMP",DllGetClas
```

Run as: **NTAUTHORITY\SYSTEM**
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

```xml
<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

A vulnerability in win32k.sys
Based on faulty processing of keyboard layout files
Used for privilege escalation
Windows 2000 and Windows XP are affected

---

<table>
<thead>
<tr>
<th>Address</th>
<th>Size</th>
<th>Owner</th>
<th>Section</th>
<th>Contains</th>
<th>Type</th>
<th>Access</th>
<th>Initial</th>
</tr>
</thead>
<tbody>
<tr>
<td>009E0000</td>
<td>00038000</td>
<td>009E0000 (itself)</td>
<td></td>
<td></td>
<td>Priv</td>
<td>00021040</td>
<td>RWE</td>
</tr>
<tr>
<td>100000000</td>
<td>0000010000</td>
<td>RC_Data_10000000 (itself)</td>
<td>.text</td>
<td>PE header code</td>
<td>Imag</td>
<td>01001002</td>
<td>RWE</td>
</tr>
</tbody>
</table>

---

```
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:
%SYSTEM\32\% (privileged operation):
Windows\System32\winsta.exe and
Win32/Stuxnet – exploit #4: MS10-046

LNK File Format

- Header
- Shell Item Id List
- File Location Info
- Shell Item Id List

LNK Header

- struct LinkTargetIDList sLinkTargetIDList
  - WORD IDListSize: 2138
  - struct IDList sIDList[0]: CLSID_MyComputer
  - struct IDList sIDList[1]: CLSID_ControlPanel

Additional Info

CPL_FindCPLInfo()

CPL_LoadAndFindApplet()
Win32/Stuxnet – exploit #4: MS10-046

4 ways of storing the path to the payload:
\\.\STORAGE\Volume\??\USBSTOR\Disk\Ven\____\USB\Prod_FLASH_DRIVE\Rev\_123450001000000173&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\_123450001000000173&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
Code injection (existing processes)

1) Allocates a memory block
2) Patches Ntdll.dll system library
3) Calls LoadLibraryW API, exported from kernel32.dll, with a constructed library as a param: KERNEL32.DLL.ASLR.XXXXXXXX or SHELL32.DLL.ASLR.XXXXXXXX; XXXXXXX being a random hex number
4) Calls desired exported function
5) Calls FreeLibrary API function to free loaded library
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)
• fsdfwfd.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

CVE vulnerability

~WTR4141.TMP

Load and call entry point

~WTR4132.TMP

Extract dll from .stub section, load and call export 0x0F

Main dll

Execute export 0x10 in a new process

Main dll

Main dll

• kernel 32.dll
• FindFirstFileW
• FindNextFileW
• FindFirstFileExW
• ntdll.dll
• NtQueryDirectoryFile
• ZwQueryDirectoryFile

To hide files:
• with "\.LNK" extension sized 1471 bytes
• with "\.TMP" extension of which the name consists of 12 characters in the format ~WTRabcd.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
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.Ink
Copy of Copy of Shortcut to.Ink
Copy of Copy of Copy of Shortcut to.Ink
Copy of Copy of Copy of Copy of Shortcut to.Ink
~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_DEVICECHANGE
Can drop or remove files from removable devices
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

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

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 **50h**
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 occurrences
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

**Step 7** uses a library to access the PLC
S7otbxldx.dll

Stuxnet replaces the DLL with its own version
Replaces s7otbxldx.dll with a malicious DLL; original library renamed to s7otbxlsx.dll

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

Government or terrorists?
Why leaving traces then? #DEADF007, 19790509, Myrtus
MS10-046 related malware and its evolution

8 (+) malware families that got “inspired”

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

CVE-2010-2744/MS10-073 (win32k.sys) – since 2009/11!!!
Information value...
... needs to be considered and protected

I will not underestimate the danger from internet threats!
I will not underestimate the danger from internet threats!
I will not underestimate the danger from internet threats!
I will not underestimate the danger from internet threats!
I will not underestimate the danger from internet threats!
I will not underestimate the danger from internet threats!
I will not underestimate the danger from internet threats!
I will not underestimate the danger from internet threats!
I will not underestimate the danger from internet threats!
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