The Era of Cyber Espionage & Cyber Warfare
(Case Study: Stuxnet)

2012.7 Forensic Insight
Kevin Koo
(kevinkoo001@gmail.com)
The Era of Cyber Espionage and Warfare

[TED] Cracking *Stuxnet*, a 21st-century cyber weapon

Demystifying Stuxnet

Overview

Brief Statistics

Architecture

Analysis in details

[TED] Fighting viruses, defending the net

Conclusion
The Era of Cyber Espionage and Warfare

Voice of concerns

- Forbes: The Flame Cyber Espionage Attack: Five Questions We Should Ask
  http://www.forbes.com/sites/johnvillasenor/2012/06/04/the-flame-cyber-espionage-attack-five-questions-we-should-ask/

  - What is the true scope of cyberattacks going on today?
  - Will infection with some form of malware soon become the rule rather than the exception?
  - In the future, will nations outsource domestic espionage?
  - Is forging digital certificates fair game for nation states?
  - Is Flame one more reason to increase cybersecurity spending?
[TED] Cracking *Stuxnet*, a 21st-century cyber weapon

- **Ralph Langner’s Speech: Cracking *Stuxnet***
  - German control system security consultant
  - [2011.3] [http://www.youtube.com/watch?v=CS01Hmjv1pQ](http://www.youtube.com/watch?v=CS01Hmjv1pQ) (10m 40s)
Demystifying *Stuxnet*

- **Overview**
  - **Stuxnet: APT Attack**
  - **Who did it?**
  - **Relative Resources**

- **Analysis**
  - (Ref#1) [Symantec] W32.Stuxnet Dossier
  - (Ref#2) [ESET] Stuxnet_Under_the_Microscope
  - (Ref#3) [SecureView] Magazine 2nd quarter in 2011
Overview: APT Attack

- Highly Advanced Persistent Threat (지능형 지속가능 위협)
- Large and complex chunk
- Who did it? Or Who could do this?
  → Iran 핵발전소 겨냥하여 원심분리기 중단
  → 목표 시스템은 독일의 지멘스(siemens) 사의 원심분리기
  → Five zero-day vulnerabilities and one known for Windows OS
  → Target이 없으면 동작하지 않도록 설계
  → 최소 10개 이상의 팀이 수 개월~수 년간 치밀하게 준비 필요
  → 자금확보는?
- Malware 종합세트(?)
  Hooking, Injection, Rootkit, 0-day, Virus&Worm, C&C, Application Hacking, ...
Overview: WHO DID IT?

- It turns out that the attack was led by the US, code-named “Olympic Games” (2012.6.1) [http://www.nytimes.com/2012/06/01/world/middleeast/obama-ordered-wave-of-cyberattacks-against-iran.html](http://www.nytimes.com/2012/06/01/world/middleeast/obama-ordered-wave-of-cyberattacks-against-iran.html)

- Imagine what happened if:
  - Power grid were crashed
  - Air control system was compromised
  - National intelligence was taken over
  - Military weapon was manipulated: nuclear weapons, intercontinental missiles
## Overview: Relative Resources (1)

- 국제원자력기구 (IAEA, International Atomic Energy Agency)
  Implementation of the NPT Safeguards Agreement and relevant provisions of Security Council resolutions in the Islamic Republic of Iran (2/9/2011)

### C.1. Natanz: Fuel Enrichment Plant and Pilot Fuel Enrichment Plant

9. **Fuel Enrichment Plant (FEP):** There are two cascade halls at FEP: Production Hall A and Production Hall B. According to the design information submitted by Iran, eight units are planned for Production Hall A, with 18 cascades in each unit. No detailed design information has yet been provided for Production Hall B.

10. On 28 August 2011, 53 cascades were installed in three of the eight units in Production Hall A, 35 of which were declared by Iran as being fed with UF₆. Whereas initially each installed cascade comprised 164 centrifuges, Iran has subsequently modified 12 of the cascades to contain 174 centrifuges each. To date, all the centrifuges installed are IR-1 machines. As of 28 August 2011, installation work in the remaining five units was ongoing, but no centrifuges had been installed, and there had been no installation work in Production Hall B.

- 지멘스사의 제어 시스템 (Siemens control solutions for utilities)
Demystifying Stuxnet

Overview: Relative Resources (2)

- Princeton 大學 논문 (2008.6)
  Characteristics of the Gas Centrifuge for Uranium Enrichment and Their Relevance for Nuclear Weapon Proliferation (우라늄 농축 가스 원심분리 특성과 핵무기 증설 관련성)
  Cascade Interconnection with Partial Reconfiguration 부분 참조

As noted, it is plausible to assume that the first set of cascades (C1 and C2) are expanded, but essentially identical versions of the standard 164-machine cascade: these cascades were designed for the same type of centrifuge (P-1), produced a typical enrichment level (3.5%), and use a multiple of 164 machines (12 \times 164 = 1968). In contrast, all cascades of the HC-type, which represent about one third of the total number of machines (1896 out of 5832), generally require other cascade configurations. Figure 8 illustrates the configuration and further data on this enrichment strategy are summarized in Table 4. For the breakout scenario starting from natural uranium, the entire set of 5832 machines is used; for the scenario starting with preenriched feed, only the HC-type cascades are required.

- Mosaic theory: 보안 분석에서 기업 정보를 수집하는 방식

As a brief recap, a first-generation Iranian uranium enrichment cascade consists of 164 centrifuges that are not simply piped in a serial fashion but in groups, which are called stages. Centrifuges within one stage are piped in parallel. The resulting overall pattern is a belly-shaped curve. The exact shape of an IR-1 cascade was not publicly known but was computed in approximation by Alexander Glaser from Princeton, based on revelations of a talkative Gholam-Reza Agazadeh who let the world know that Iran used to group their IR-1 cascades into fifteen stages. From the IR-1 cascade structure computed by Alex we were able to link Stuxnet’s 417 attack code to Natanz – the match was simply too good to be a coincidence.
Demystifying *Stuxnet*

- **Overview: Analysis**
  - ✓ **Target:** a specific industrial control system (특정 산업 통제 시스템)
  - ✓ 분석 보고서
    - [ESET]
      [ESET](http://go.eset.com/us/resources/white-papers/Stuxnet_Under_the_Microscope.pdf)
    - [Symantec]
      [Symantec](http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf)

  - ✓ **Three Main Steps**
    1. **Windows Infection using zero-day vulnerabilities:**
       - Microsoft Windows Server Service RPC Handling Remote Code Execution Vulnerability (MS08-067)
       - Microsoft Windows Shortcut ‘LNK/PIF‘ Files Automatic File Execution Vulnerability (MS10-046)
       - Microsoft Windows Print Spooler Service Remote Code Execution Vulnerability (MS10-061)
       - Microsoft Windows Win32k.sys Local Privilege Escalation vulnerability (MS10-073)
       - Microsoft Windows Task Scheduler Escalation of Privilege vulnerability (MS10-092)
    2. **Step7 Software Infection:** Siemens' WinCC/PCS 7 SCADA control software
    3. **PLC(Programmable Logical Computer) infection**
# Demystifying Stuxnet

## W32.Stuxnet Timeline

<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>November 20, 2008</td>
<td>Trojan.Zlob variant found to be using the LNK vulnerability only later identified in Stuxnet.</td>
</tr>
<tr>
<td>January 25, 2010</td>
<td>Stuxnet driver signed with a valid certificate belonging to Realtek Semiconductor Corps.</td>
</tr>
<tr>
<td>March, 2010</td>
<td>First Stuxnet variant to exploit MS10-046.</td>
</tr>
<tr>
<td>June 17, 2010</td>
<td>Virusblokada reports W32.Stuxnet (named RootkitTamphider). Reports that it’s using a vulnerability in the processing of shortcuts/.lnk files in order to propagate (later identified as <a href="https://support.microsoft.com/en-us/kb/2286198">MS10-046</a>).</td>
</tr>
<tr>
<td>July 13, 2010</td>
<td>Symantec adds detection as W32.Temphid (previously detected as Trojan Horse).</td>
</tr>
<tr>
<td></td>
<td>Verisign revokes Realtek Semiconductor Corps certificate.</td>
</tr>
<tr>
<td>July 17, 2010</td>
<td>Esset identifies a new Stuxnet driver, this time signed with a certificate from JMicron Technology Corp.</td>
</tr>
<tr>
<td>July 19, 2010</td>
<td>Siemens report that they are investigating reports of malware infecting Siemens WinCC SCADA systems.</td>
</tr>
<tr>
<td></td>
<td>Symantec renames detection to W32.Stuxnet.</td>
</tr>
<tr>
<td>July 20, 2010</td>
<td>Symantec monitors the Stuxnet Command and Control traffic.</td>
</tr>
<tr>
<td>July 22, 2010</td>
<td>Verisign revokes the JMicron Technology Corps certificate.</td>
</tr>
<tr>
<td>August 2, 2010</td>
<td>Microsoft issues <a href="https://support.microsoft.com/en-us/kb/259525">MS10-046</a>, which patches the Windows Shell shortcut vulnerability.</td>
</tr>
<tr>
<td>August 6, 2010</td>
<td>Symantec reports how Stuxnet can inject and hide code on a PLC affecting industrial control systems.</td>
</tr>
<tr>
<td>September 14, 2010</td>
<td>Microsoft releases <a href="https://support.microsoft.com/en-us/kb/259525">MS10-061</a> to patch the Printer Spooler Vulnerability identified by Symantec in August.</td>
</tr>
<tr>
<td></td>
<td>Microsoft report two other privilege escalation vulnerabilities identified by Symantec in August.</td>
</tr>
<tr>
<td>September 30, 2010</td>
<td>Symantec presents at Virus Bulletin and releases comprehensive analysis of Stuxnet.</td>
</tr>
</tbody>
</table>
Demystifying *Stuxnet*

- **Brief Statistics**

  ![Graphs showing infected hosts and organizations, geographic distribution of infections, and percentage of Stuxnet infected hosts with Siemens software installed.](image-url)
Demystifying *Stuxnet*

- Architecture
  - Overview
  - Exports & Resources
  - Techniques to use
Demystifying *Stuxnet*

**Architecture - Overview**

* Stuxnet = a wrapper
  → stored in a **stub section**
  → Consists of:
    : large .dll file

(1) A **pointer** to the original stub section is passed to exports as a **parameter**

(2) How to call exports in the main .dll file
  → Loading the .dll file into memory and calling an export **directly**
  → Reading an **executable template** from its own resources, populate the template with appropriate data

: two encrypted configuration file

5.1 - 1/1/0 - 2 - 2010/09/22-15:15:47 127.0.0.1, [COMPUTER NAME] [DOMAIN NAME] [c:\a\1.zip:\proj.s7p]

5.1 - Major OS Version and Minor OS Version
1/1/0 – Flags used by Stuxnet
2 – Flag specifying if the computer is part of a workgroup or domain
127.0.0.1 – Up to IP addresses of the compromised computer (not in the June 2009 version).
[COMPUTER NAME] – The computer name.
[DOMAIN NAME] – The domain or workgroup name.
[c:\a\1.zip:\proj.s7p] – The file name of infected project file.
## Architecture – Exports & Resources

<table>
<thead>
<tr>
<th>DLL Exports</th>
<th>DLL Resources</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Export #</strong></td>
<td><strong>Function</strong></td>
</tr>
<tr>
<td>1</td>
<td>Infect connected removable drives, starts RPC server</td>
</tr>
<tr>
<td>2</td>
<td>Hooks APIs for Step 7 project file infections</td>
</tr>
<tr>
<td>4</td>
<td>Calls the removal routine (export 18)</td>
</tr>
<tr>
<td>5</td>
<td>Verifies if the threat is installed correctly</td>
</tr>
<tr>
<td>6</td>
<td>Verifies version information</td>
</tr>
<tr>
<td>7</td>
<td>Calls Export 6</td>
</tr>
<tr>
<td>9</td>
<td>Updates itself from infected Step 7 projects</td>
</tr>
<tr>
<td>10</td>
<td>Updates itself from infected Step 7 projects</td>
</tr>
<tr>
<td>14</td>
<td>Step 7 project file infection routine</td>
</tr>
<tr>
<td>15</td>
<td>Initial entry point</td>
</tr>
<tr>
<td>16</td>
<td>Main installation</td>
</tr>
<tr>
<td>17</td>
<td>Replaces Step 7 DLL</td>
</tr>
<tr>
<td>18</td>
<td>Uninstalls Stuxnet</td>
</tr>
<tr>
<td>19</td>
<td>Infects removable drives</td>
</tr>
<tr>
<td>22</td>
<td>Network propagation routines</td>
</tr>
<tr>
<td>24</td>
<td>Check Internet connection</td>
</tr>
<tr>
<td>27</td>
<td>RPC Server</td>
</tr>
<tr>
<td>28</td>
<td>Command and control routine</td>
</tr>
<tr>
<td>29</td>
<td>Command and control routine</td>
</tr>
<tr>
<td>31</td>
<td>Updates itself from infected Step 7 projects</td>
</tr>
<tr>
<td>32</td>
<td>Same as 1</td>
</tr>
</tbody>
</table>
Demystifying *Stuxnet*

- **Architecture - Techniques to use: Hooking**
  
  * Bypassing behavior blocking when loading DLLs
    - Hooking `ntdll.dll` to monitor requests to load specially crafted file names
      : `KERNEL32.DLL.ASLR.[HEXADECIMAL]` or `SHELL32.DLL.ASLR.[HEXADECIMAL]`
    - Hooked functions
      - `ZwMapViewOfSection`
      - `ZwCreateSection`
      - `ZwOpenFile`
      - `ZwCloseFile`
      - `ZwQueryAttributesFile`
      - `ZwQuerySection`
Demystifying *Stuxnet*

**Architecture - Techniques to use: Process Injection**

* Keep injected code in the trusted process
* Instruct the trusted process to inject the code into another currently running process

![Process Injection Table]

Ref. Symantec W32.Stuxnet Dossier p.14

<Running trusted process>
- Kaspersky KAV (avp.exe)
- McAfee (Mcshield.exe)
- AntiVir (avguard.exe)
- BitDefender (bdagent.exe)
- Etrust (UmxCfg.exe)
- F-Secure (fsdfwd.exe)
- Symantec (rtvscan.exe)
- Symantec Common Client (ccSvcHst.exe)
- Eset NOD32 (ekrn.exe)
- Trend Pc-Cillin (tmpproxy.exe)

<Registry>
- KAV v6 to v9
- McAfee
- Trend PcCillin

<Potential Target Process>
- Lsass.exe
- Winlogon.exe
- Svchost.exe
- The installed security product process
Demystifying Stuxnet

- Analysis in details
  - Installation: Preparation, Injection
  - Functionalities: Load point, C&C, Rootkit, Propagation
  - Techniques to use
Demystifying Stuxnet

Analysis in details - Installation: Preparation

Control flow for export 15

1) Windows Win32k.sys Local Privilege Escalation vulnerability (MS10-073)
2) Task Scheduler Escalation of Privilege vulnerability (MS10-092)

Two privilege escalation (or Elevation of Privilege) vulnerabilities:
Demystifying *Stuxnet*

- **Analysis in details - Installation: Infection routine flow (Export 16)**

1. **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\MS-DOS Emulation**
2. `%SystemDrive%\inf\mdmeric3.PNF, %SystemDrive%\inf\mdmcpq3.PNF, %SystemDrive%\inf\oem6C.PNF`
3. `%SystemDrive%\drivers\Mrxnet.sys, %SystemDrive%\drivers\Mrxcls.sys`
4. Infecting newly connected removable drives and for starting the RPC server
5. Hooking APIs for Step 7 project file infections
Demystifying Stuxnet

- **Analysis in details - Functionality: Load Point**
  - Drops Resource 242 MrxCls.sys via Export 16
  - The mrxcls.sys is a driver digitally signed with a compromised Realtek certificate. (suddenly revoked on July 16, 2010 by Verisign)
  - File checksums
    
    | Filename   | Size      | MD5               |
    |------------|-----------|-------------------|
    | mrxnet.sys | 17,400 bytes | cc1db5360109de3b857654297d262ca1 |
    | mrxcls.sys | 26,616 bytes | f8153747bae8b4ae48837ee17172151e |

  - Purpose of the rootkit drivers
    mrxnet.sys to hide the presence of the worm on removable drives
    mrxcls.sys to inject the worm into “services.exe” and two processes specific to Siemens software (Step7/S7 and WinCC)
  - The driver contains an encrypted data block.
    services.exe — %Windir%\inf\oem7A.PNF
    S7tgtopx.exe — %Windir%\inf\oem7A.PNF
    CCProjectMgr.exe — %Windir%\inf\oem7A.PNF
    explorer.exe — %Windir%\inf\oem7m.PNF
Analysis in details - Functionality: Command and Control

System data is gathered via Export 28, and Export 29 send payload to a target server.

- www[.].mypremierfutbol[.]com (Malaysia)
- www[.].todaysfutbol[.]com (Denmark)

The flags at offset 11h have the 4th bit set if at least one of the two registry values is found:

- HKEY_LOCAL_MACHINE\Software\Siemens\Step7, value: STEP7_Version
- HKEY_LOCAL_MACHINE\Software\Siemens\WinCC\Setup, value: Version
Demystifying *Stuxnet*

**Analysis in details - Functionality: Windows Rootkit**

- The ability to hide copies of its files copied to removable drives
- Registered as a boot start service
  
  HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MRxCls\"ImagePath" = "%System%\drivers\mrxcls.sys"

- Digitally signed driver file with a *legitimate digital certificate*

- Scans the filesystem driver objects:
  - \FileSystem\ntfs
  - \FileSystem\fastfat
  - \FileSystem\cdf

- The driver monitors “directory control“ IRPs, in particular “directory query“ notifications.

- Two types of files filtered out
  - Files with a “.LNK” extension having a size of 4,171 bytes.
  - Files named “~WTR[FOUR NUMBERS].TMP”, whose size is between 4Kb and 8Mb; the sum of the four numbers modulo 10 is null.
    - Copy of Copy of Copy of Shortcut to.Ink
    - Copy of Copy of Copy of Shortcut to.Ink
    - Copy of Copy of Shortcut to.Ink
    - Copy of Shortcut to.Ink
    - ~wtr4132.tmp
    - ~wtr4141.tmp
Demystifying *Stuxnet*

- Several questions about digital signatures (from SecureView Mags)

  - Digitally signed driver file with legitimate Realtek and JMicron digital certificates

<table>
<thead>
<tr>
<th>Name of Signer</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>JMicron Technology Corp</td>
<td>Not available</td>
</tr>
<tr>
<td>Email Address</td>
<td>Not available</td>
</tr>
<tr>
<td>Signing Time</td>
<td>1/25/2010 2:45:24 PM</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name of Signer</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Realtek Semiconductor Corp</td>
<td>Not available</td>
</tr>
<tr>
<td>Email Address</td>
<td>Not available</td>
</tr>
<tr>
<td>Signing Time</td>
<td>1/25/2010 2:45:24 PM</td>
</tr>
</tbody>
</table>

  Content Type: 06 8A 2E 06 01 04 01 82 37 02 01 04
  1.3.6.1.4.1.311.2.1.1.1: 30 0C 06 0A 2B 06 01 04 01 82 37 02 01 15
  Message Digest: 04 14 6A C5 D5 8A 5D F4 31 2D 7D 18 92 60 39
  1.3.6.1.4.1.311.2.1.12: 30 2E A0 10 4D 00 0A 4D 00 0A 4D 00 0A 4D 00 0A 4D 00 04

  Child Type: StringFileInfo
  Language/Code Page: 1033/1200
  Comments: change me
  CompanyName: change me
  FileDescription: change me
 FileVersion: 3.00
  InternalName: change me
  LegalCopyright: change me
  LegalTrademarks: change me
  OriginalFilename: change me
  ProductName: change me
  ProductVersion: 3.00

  How did the attackers manage to obtain the private keys required to sign them?
  * Were Realtek and JMicron involved in the operation and willingly sign the files?
  * Since both companies have development offices in China, are the Chinese involved?
Analysis in details - Functionality: Propagation Methods
Peer-to-Peer communication
Infected WinCC computers
Through network shares
Removable drives
Analysis in details - Functionality: Propagation Methods (P2P)

- RPC Server offers the following routines.
  0: Returns the version number of Stuxnet installed
  1: Receive an .exe file and execute it (through injection)
  2: Load module and execute export
  3: Inject code into lsass.exe and run it
  4: Builds the latest version of Stuxnet and sends to compromised computer
  5: Create process
  6: Read file
  7: Drop file
  8: Delete file
  9: Write data records

Example of an old client requesting latest version of Stuxnet via P2P
Demystifying *Stuxnet*

- **Analysis in details - Functionality: Propagation Methods (Infecting WinCC computers)**
  - Connects to a remote server running the WinCC database using a password hardcoded within the WinCC software
  - Two actions when found:
    - sends malicious SQL code
    - modifies an existing view
  - Sends an SQL statement that creates a table and inserts a binary value into the table

```sql
SET @ainf = @aind + '\sql%05x.dbi'
EXEC sp_addextendedproc sp_dumpdbilog, @ainf
EXEC sp_dumpdbilog
set @t=left(@t,len(@t)-charindex('\\',reverse(@t)))+'\GraCS\cc_tlg7.sav';
set @s = 'master..xp_cmdshell 'extrac32 /y ''+@t'' ''+@t+x''';
exec(@s);
```
Demystifying *Stuxnet*

- **Analysis in details - Functionality: Propagation Methods (Through network shares)**
  - Through either a scheduled job or using Windows Management Instrumentation (WMI)
  - Enumerate all user accounts of the computer and the domain, and try all available network resources either using the user’s credential token or using WMI operations with the explorer.exe token in order to copy itself and execute on the remote share.
Demystifying Stuxnet

Analysis in details - Functionality: Propagation Methods (Removable drive)

- LNK Vulnerability (MS010-046, CVE-2010-2568)

USB Execution Flow

![Diagram of USB execution flow]

**Autorun.inf header**

```
00000000: 4D5A9000 03000000 04000000 FFEE0000 MZ...........yy.
00000120: B8000000 00000000 40000000 00000000 ........@.........
00000200: 00000000 00000000 00000000 00000000 ........E........
00000300: 00000000 00000000 00000000 00000000 ........A........
000003E0: 0E1FEA0E 00E409CD 21B8014C CD215468 ...$i__II!Th
00000400: 69732070 72657374 656D656E 65727374 ....is program canno
00000460: 74206265 20727265 6E656420 44455320 t be run in DOS
00000520: 65616269 64656420 00000000 00000000 mode $...
000005C0: CF7A777C 686B192F 80B1B92F 08B1B92F Izv|.../|.../...
00000640: 00000090: ACDD642F 9D1B192F ACDD6622F 9C1B192F -Yd/|.../Yb/|...
00000680: 000000A0: 80B1B92F 6D1B192F ACDD6E2F DA1B192F .../.../.../...
```

**Autorun.inf footer**

```
00041000: 00DAE5B1 75746652 7566ED0D 0A6F526A ...
00041010: 65666465 65666465 69732070 72657374 ...
00041020: 33313533 33313533 3341422D 33353333 ...
00041030: 2D393941 392D3246 34363737 32333341 ...
00041040: 3434700D 0A ...
00041050: 61666465 61666465 2E5C4155 544F4255 ...
00041060: 5C4D5656 5C4D5656 5C4D5656 5C4D5656 ...
00041070: 75734025 77766656 6972255C 73797374 ...
00041080: 65663232 5C736366 6C633332 2E64646E ...
00041090: 2C2D3534 3936300A ...
000410A0: 00DA 55736541 75746650 4C41593D ...
000410B0: 30000A ...

?AVZdhnrndcnnGvqzdhFnnldcnn@gmailcnn@sr@@
[aorun]
objectDesciptor=\B315537-63AP-9512-99A9-2F4677235A44\nMenu\command=.\AUTORUN.INF
Menu=%\windir%\system32\shell32.dll,-8496
UseAutoPLAY=0
```

forensicinsight.org
Analysis in details - Functionality: Others (Removable drive)

- Trick to enhance the chances to be executed
- Real one: %WinDir%\System32\shell32.dll,-8496
Demystifying *Stuxnet*

- **Analysis in details - Step 7 Project File Infections**
  - Export 16 (Installation) calls Export 2 to hook specific APIs in the `s7tgtopx.exe`
    - In `s7apromx.dll`, `mfc42.dll`, and `msvcrt.dll`, `CreateFileA` is replaced to point to “CreateFileA_hook”.
    - In `ccprojectmgr.exe`, `StgOpenStorage` is replaced to point to “StgOpenStorage_hook”.
  - `CreateFileA_hook` to open S7P files for recording and infecting project folder
  - `*.S7P files → %Windir%/inf/oem6c.pnf`
  - `*.MCP files → oem6c.pnf`
  - `*.TMP files`
Demystifying Stuxnet

Analysis in details - Step 7 Project File Infections: S7P files

- Step 7 Project file
- A candidate for infection if:
  - It is not deemed too old (used or accessed in the last 3.5 years).
  - It contains a “wincproj” folder with a valid MCP file.
  - It is not a Step7 example project, checked by excluding paths matching “*\Step7\Examples\*”.

- Infection process consists of next steps:
  1. Stuxnet creates the following files:
     - xutils\listen\xr000000.md (an encrypted copy of the main Stuxnet DLL)
     - xutils\links\s7p0001.dbf (a copy of a Stuxnet data file (90 bytes in length)
     - xutils\listen\s700001.md (an encoded, updated version of the Stuxnet configuration data block)
  2. The threat scans subfolders under the “h0mSave7” folder. In each of them, Stuxnet drops a copy of a DLL it carries within its resources (resource 202). This DLL is dropped using a specific file name. The file name is not disclosed here in the interests of responsible disclosure and will be referred to as xyz.dll.
  3. Stuxnet modifies a Step7 data file located in Apilog\types.
Analysis in details - Step 7 Project File Infections: MCP files

- Created by WinCC
- A candidate for infection if:
  - It is not deemed too old (used or accessed in the last 3.5 years).
  - It contains a GracS folder with at least one .pdl file in it.

- Infection process consists of next steps:
  1. Stuxnet creates the following files:
     - GracS\cc_alg.sav (an encrypted copy of the main Stuxnet DLL)
     - GracS\db_log.sav (a copy of a Stuxnet data file, which is 90 bytes in length)
     - GracS\cc_alg.sav xutils\listen\s70000001.mdx (an encoded, updated version of the Stuxnet configuration data block)
  2. A copy of resource 203 is then decrypted and dropped to GracS\cc_tlg7.sav. This file is a Microsoft Cabinet file containing a DLL used to load and execute Stuxnet.

- WinCC DB is also infected during this infection process
Analysis in details - Step 7 Project File Infections: TMP files

- Validated filename: ~WRabcde.tmp where \((a+b+c+d+e) \mod 16 = 0\)
- Magic string (1st 8Bytes): LRW~LRW~
- Export #9 takes a Step7 Project path as input, then build paths:
  - ...\XUTILS\listen\XR000000.MDX
  - ...\XUTILS\links\S7P00001.DBF
  - ...\XUTILS\listen\S7000001.MDX
- Export #31 takes a Step7 Project path as input, then build paths:
  - ...\GracS\cc_alg.sav
  - ...\GracS\db_log.sav
  - ...\GracS\cc_tag.sav
Demystifying *Stuxnet*

- Analysis in details - Modifying PLCs (Terms)
  - **DB** (Data Blocks): program specific data
    Ex) numbers, structures
  - **SDB** (System Data Blocks): configuration info.
  - **OB** (Organization Blocks): entry point of programs
    → OB1: main EP of the PLC
    → OB35: standard watchdog
  - **FB** (Function Blocks): standard code blocks

Stuxnet code in the Step7 STL editor
Demystifying *Stuxnet*

- **Analysis in details - Modifying PLCs (Infection)**
  - Resource 208 is dropped by export #17
  - Replacement for Simatic’s `s7otbxdx.dll` file.

![Diagram](image)

- **Step 7 and PCL communicating via `s7otbxdx.dll`**
- **Communication with malicious version of `s7otbxdx.dll`**

93 out of 109 exports

16 out of 109 exports
Demystifying *Stuxnet*

- **Analysis in details - Modifying PLCs (Sequence Blocks)**
  - Sequence A,B then C
  - Initial Infection
    - The first thread runs an infection routine every 15 minutes. The targeted PLC information has previously been collected by the hooked exports, mainly s7db_open(). This infection routine specifically targets CPUs 6ES7-315-2 (series 300) with special SDB characteristics. The sequence of infection is A or B.
    - The second thread regularly queries PLC for a specific block that was injected by the first thread if the infection process succeeded. This block is customized, and it impacts the way sequences A or B run on the infected PLC.
  - The infection threat, sequences A and B
    - First, the PLC type is checked using the s7ag_read_szl API. It must be a PLC of type 6ES7-315-2.
    - The SDB blocks are checked to determine whether the PLC should be infected and if so, with which sequence (A or B).
    - If the two steps above passed, the real infection process starts. The DP_RECV block is copied to FC1869, and then replaced by a malicious block embedded in Stuxnet.
    - The malicious blocks of the selected infection sequence are written to the PLC.
    - OB1 is infected so that the malicious code sequence is executed at the start of a cycle.
    - OB35 is also infected. It acts as a watchdog, and on certain conditions, it can stop the execution of OB1.
  - Summary (key steps)
    - → SDB Check
    - → DP_RECV replacement
    - → OB1/OB35 infection
Demystifying *Stuxnet*

- **Analysis in details - Modifying PLCs (Sequence Blocks)**

  - Connections Between Blocks, Sequences A and B (targeting S7-315)

  ![Diagram showing connections between blocks and sequences]

  - Green: Stuxnet data blocks
  - Red: main blocks
  - White: standard Stuxnet blocks
  - Grey: system function blocks

<table>
<thead>
<tr>
<th>Infection OB1</th>
<th>Infection OB35</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC: FC1865</td>
<td>UC: FC1874</td>
</tr>
<tr>
<td>POP</td>
<td>POP</td>
</tr>
<tr>
<td>L: DW#16#DEADF007</td>
<td>L: DW#16#DEADF007</td>
</tr>
<tr>
<td>==D</td>
<td>==D</td>
</tr>
<tr>
<td>BEC</td>
<td>BEC</td>
</tr>
<tr>
<td>L: DW#16#0</td>
<td>L: DW#16#0</td>
</tr>
<tr>
<td>L: DW#16#0</td>
<td>L: DW#16#0</td>
</tr>
</tbody>
</table>
Demystifying Stuxnet

- Analysis in details - Modifying PLCs (Sequence Blocks)
  - State machine path of execution

![Diagram showing state machine path of execution with nodes and edges representing different states and transitions involving PLCs and frequency converters.]
Demystifying *Stuxnet*

- Analysis in details - Modifying PLCs (Sequence Blocks)
  - Connections Between Blocks, Sequences C (targeting S7-417 PLCs)
Demystifying *Stuxnet*

- **Analysis in details - Modifying PLCs (Sequence Blocks)**
  - Eight states in sequence C
    - State 0: Wait
    - State 1: Recording
    - State 2-6: Sabotage
    - State 7: Reset
  - Never happened due to the missing function in the DLL

**Affected peripherals within each cluster**

<table>
<thead>
<tr>
<th>Cluster Number</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peripherals in the Cluster</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>12</td>
<td>16</td>
<td>20</td>
<td>24</td>
<td>20</td>
<td>16</td>
<td>12</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>Peripheral Number</td>
<td>0-1</td>
<td>2-3</td>
<td>4-7</td>
<td>8-13</td>
<td>14-21</td>
<td>22-31</td>
<td>32-43</td>
<td>44-59</td>
<td>60-79</td>
<td>80-103</td>
<td>104-123</td>
<td>124-139</td>
<td>140-151</td>
<td>152-159</td>
<td>160-163</td>
</tr>
<tr>
<td>Peripherals affected</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>13</td>
<td>14</td>
<td>0</td>
<td>14</td>
<td>13</td>
<td>10</td>
<td>8</td>
<td>4</td>
</tr>
</tbody>
</table>

- 164 total
- 110 affected
Mikko Hypponen’s Speech: Fighting viruses, defending the net

- F-Secure CRO (Chief Research Officer)
- [Link](http://mikko.hypponen.com/)
- [Video](http://www.youtube.com/watch?v=cf3zxHuSM2Y) (17m 34s)
- [Clarified Networks Videos](https://www.clarifiednetworks.com/Videos)
Conclusion

Lessons Learned

- Cyber warfare has begun.
- Stuxnet was a huge threat and very complicated, targeted and sophisticated.
- It has started national support to attack against targeted enemy.
- It is like a mission-impossible game.
- Now is the time to foster cyber warriors with cyber weapons.
- APT is evolving: Duqu (2011.10), Flame (2012.6)

Last week, Kaspersky Lab announced the discovery of Flame, a malicious program with “complexity and functionality ... exceed[ing] those of all other cyber menaces known to date.” Once
NO QUESTIONS?