CCDCOE

NATO Cooperative Cyber Defence
Centre of Excellence Tallinn, Estonia

Stuxnet Facts Report

A Technical and Strategic Analysis

LTC Marco De Falco

Tallinn 2012

Disclaimer

This publication is a product of the NATO Cooperative Cyber Defence Centre of Excellence (the Centre) and it represents the views and interpretations of the Centre. This publication does not represent the opinions or policies of NATO and is designed to provide an independent position.

Third-party sources are quoted as appropriate and the Centre is not responsible for the content of the external sources referenced in this publication. The Centre assumes no responsibility for any loss or harm arising from the use of information contained in this publication. Copies of this publication may be distributed for non-profit and non-commercial purpose only.

Contact

NATO Cooperative Cyber Defence Centre of Excellence
Filtri tee 12, Tallinn 10132, Estonia
publications@ccdcoe.org
www.ccdcoe.org

This study is dedicated to ReSIA and to CGA.

Table of Contents

List of Figures

N.B.: All images in this document are released in the public domain or used under the "fair use" rule.

List of Tables

Introduction

At some time in every developing technology, a new outcome will mark an important step and will be remembered as the opening of a new "era". For instance, in aviation this occurred when the reaction engine was built and quickly replaced the propeller engine. In biology, it occurred with the discovery of DNA (Watson and Crick, 1953). In malware technology, this has happened with the development and use of Stuxnet.
Although the initial basic operation of Stuxnet reflects the classical malware actions (network vulnerability exploitation, rootkit installation, data exfiltration, Internet self-upgrade, etc.), Stuxnet differs from any other precedent worm because of its complexity, flexibility, potentiality, combination of features, multi-role performance and goal. It is as if, in the wild and dangerous environment of malware, a new breed of worm has evolved by learning all the most successful, state-of-the-art attack and survival strategies, and is now using them to hit very specific targets. Stuxnet is definitely the founder of this new breed of *superworms*.

Understanding PLCs and ICS environments

Stuxnet is a piece of malware which has been written expressly for targeting industrial systems, not personal computers, and this is one of its several peculiarities. Personal computers are infected only because they are the "natural gateway" through which the worm can attack the industrial systems.
To fully understand how this is possible, and why the worm performs certain operations, one has to understand the ICS (Industrial Control System) environment it was conceived for.
In the industrial world, machines such as robotic arms, conveyor belts or complex hydraulic systems are operated and controlled by specialised computers called PLCs (Programmable Logic Controllers). Those computers are particularly effective in the aspects of robustness, speed, safety, modularity, and capability of interfacing.
The PLCs – like any computer – have a CPU, a memory, and many I/O ports, through which they basically do three things:

- read inputs from the external world (sensors);
- activate or deactivate electrical contacts which operate devices (relays or drivers);
- interact with human users, mainly for displaying information, receiving commands and being programmed.

Fig. 1: The Siemens Simatic S7-300 PLC (CPU 315), equipped with a 24 V power supply and four I/O modular units. This is exactly the PLC model hunted by Stuxnet.

The input and output ports can be digital or analogue. Digital ports can have only two states (on/off), while analogic ports are able to measure (input) or provide (output) variable voltage or variable current in a given interval.

The brand which is the *de facto* standard in industrial PLCs is Siemens, with its architecture *Simatic*.
So, resuming, there is a three-tier architecture:

Denomination:	**PG/HMI** (Programming Gateway / Human-Machine Interface)	**PLC** (Programmable Logic Controller)	**Field Device**
Function:	Uploads executable instructions into the PLC (i.e. via a serial or Profibus interface), controls and monitors PLC operations	Manages the connected device and reports status of operations to a visualisation workstation	Ad-hoc operation. (i.e. regulates the rotation speed of an electric engine)
Example:	Intel-based computer with Windows XP O.S.	Simatic S7-300/400	Frequency driver for electric motors

Fig. 2: block diagram of an ICS environment

Personal computers used in the architecture (mostly Windows PCs) do not directly control the process. The control is carried out through Programmable Logic Controllers, in this case Simatic PLCs from Siemens. PCs only work as SCADA/HMI (Supervisory Control And Data Acquisition / Human Machine Interface) workstations, i.e. machines that display information to the operators and communicate with PLCs.
In the Simatic architecture, Siemens provides the users with several programs that constitute an integrated environment conceived to easily and efficiently manage all the needs related to PLCs. This is the STEP 7 environment, a Siemens software product from which it is possible to define a so-called "project" that will contain the metadata describing the configuration of the managed PLCs, along with many other types of data. Then, according to this configuration, from STEP 7 it is possible to access the real PLC and upload these data. Next, always within the STEP 7 environment, the user can start to write his/her own STL program – that will instruct the PLC about what to do and how to react to external data read from sensors (input ports) – and can upload it inside the memory of the managed PLC. Finally, through the Siemens WinCC software which runs on personal computers, the user can start the program uploaded in the PLC, and receive from it information on how the process is going.

General presentation of the threat

Technically speaking, Stuxnet is a worm, as it spreads over a network without the need for executable files or user intervention. It can also replicate through USB removable devices.
Its name is derived from the file *mrxnet.sys* that the worm installs for rootkit purposes (see later → *Infection history*).

The worm can present itself in three different forms, each one containing executable code:

1) the main dropper worm's code (`~WTR4132.TMP`). This is the main executable code, the one which spreads through the network and on USB storage devices. Its UPX-packed size is around 0.5 MB, while the unpacked form has a size of around 1.2 MB. It is basically a large DLL, from which the other parts listed below are dynamically extracted.
2) the LNK shortcut code. This is a specially-crafted Windows shortcut (.LNK) file which is written by the worm code in removable USB drives as it finds them accessible in the infected system. As soon as the shortcut is displayed within a Windows Explorer window, the worm code to which it is linked is executed. To put it more simply, just browsing a removable media drive which contains these .LNK files using an application that displays shortcut icons (such as Windows Explorer or Total Commander) starts the execution of the malware without any additional user interaction.
3) the rootkit code. This is the code portion which is installed on a system after the dropper code is executed, and is made up of two files – `mrxnet.sys` and `mrxcls.sys` – which are installed as device drivers files, signed with a Realtek or with a JMicron certificate. The two files respectively take care of hiding the worm files and preventing behaviour-based antimalware products from detecting the viral activities.

For a complete list of all Stuxnet files, see Annex E.

The worm is highly adaptive, in the sense that it will put into action a number of tactics aimed at maximising the probability of successful infection (in terms of different operating systems) and circumventing eventual defences found on the target systems.

The following operating systems are affected (only 32-bit versions):

- Windows 2000 (Professional, Server, and Advanced Server)
- Windows XP
- Windows 2003 Server
- Windows Vista
- Windows 2008 Server
- Windows 7

Infection history

Stuxnet was first discovered by Belarusian security company VirusBlokAda (www.anti-virus.by) on June 17, 2010, in the computers of one of its customers, who asked the company for technical help with some unexplainable system reboots.[1] The malware was found on 14 systems, the majority of which were located in Iran.

Sergey Ulasen and Oleg Kupreev, security experts working in the company, started conducting an initial analysis[2] and lebelled the components of Stuxnet with the names "*Trojan-Spy.0485*" and "*Malware-Cryptor.Win32.Inject.gen.2*". They soon discovered what are still now believed to be the most significant features of Stuxnet: that it was spreading via USB removable media thanks to the .LNK zero-day vulnerability, and that it was installing for rootkit purposes two driver files (`mrxnet.sys` and `mrxcls.sys`), which they identified as "*Rookit.Tmphider*" and "*SScope.Rookit. Tmphider2*", both signed with the digital signature of Realtek Semiconductor Corp. On June 24, 2010, Realtek was made aware of the issue but did not reply.

By mid-July 2010, an advisory about the worm appeared on the VirusBlokAda website. By that time the worm final payload was still unknown.

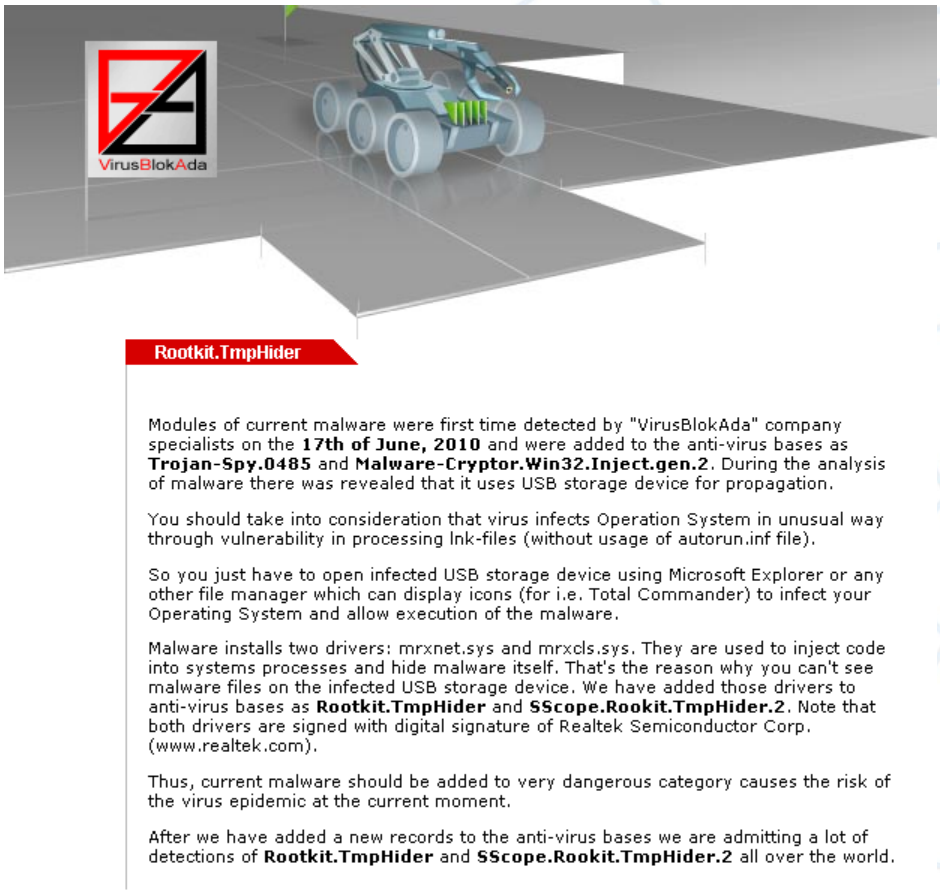

Fig.3: The first communication about Stuxnet (initially called "TmpHider") appeared on the VirusBlokAda website in July, 2010.

After disclosing this information, VirusBlokAda then contacted Microsoft but did not receive a response.

---

1 For more information on this particular issue, refer to paragraph → "*The mistakes of Stuxnet*" (page 29)

2 "*Trojan-Spy.0485 And Malware-Cryptor.Win32.Inject.gen.2 Overview*" ("Обзор вредоносных программ Trojan-Spy.0485 и Malware-Cryptor.Win32.Inject.gen.2"), by S. Ulasen and O. Kupreev (July 9, 2010). Available at: ftp://anti-virus.by/pub/docs/russian/Rootkit.TmpHider.pdf.

On July 13, 2010, the appearance of the threat became known to security company Symantec. Just 72 hours later, Symantec analysts documented about 14000 unique IP addresses infected with the worm and started attempts to contact the controlling servers.
On July 14, 2010, the Microsoft group manager of response communications, Jerry Bryant, informed KrebsOnSecurity.com (a blog posting security news that emphasised the existence of the new threat) that his company was "*investigating new public claims of a malware propagating via USB storage devices*", and that upon completion of the investigations it would take "appropriate action to protect users".
Microsoft dubbed the worm "*Stuxnet*", from a combination of text strings (".stub" and "MrxNet.sys") found in the code.
Independent security researcher Frank Boldewin[3] reported a few days later that he was able to examine the malware, and that it appeared to be searching for Siemens WinCC SCADA software, adding that it looked "*like this malware was made for espionage*".

What was very clear from the beginning was that Stuxnet could attack industrial Siemens PLCs, but this was just the tip of the iceberg.
As long as worldwide security experts were reverse-engineering the malware, new stunning details came to the surface, each day showing more and more of its initially underestimated complexity and sophistication.
A month after its first discovery, Microsoft and Siemens said the worm was actively targeting Windows PCs that managed large-scale industrial-control systems in manufacturing and utility firms.

Despite its first discovery in June 2010, Stuxnet is believed to have been in the wild since at least one year earlier (see: → "*Data exfiltration*", in the following chapter). Once VirusBlokAda discovered Stuxnet, antivirus vendors were able to search through their sample databases and find even earlier samples of the worm. The earliest samples dated back to June 2009. The first wave of Stuxnet attacks probably started at this time (consisting of 10 initial infections targeting five organisations inside Iran),[4] resulting in 12000 infections within one year.
Moreover, one of the Stuxnet components (`mrxnet.sys`) has a compile date set to January 1st, 2009.

Most important dates

20 November 2008	First use of the LNK vulnerability, by Trojan.Zlob .
June 2009	Oldest recorded occurrence of Stuxnet's activity (first version): first wave of attacks targeting five organisations inside Iran.
17 June 2010	First discovery of Stuxnet by VirusBlokAda. The version discovered was set to limit its spreading to three consecutive infections.
8 July 2010	Microsoft Security Essentials is the first antivirus capable of detecting Stuxnet (by checking `winsta.exe`) as *TrojanDropper:Win32.Stuxnet.A.*
16 July 2010	Verisign revokes the Realtek Semiconductor digital certificate used to sign the Stuxnet rootkit files.
17 July 2010	ESET identifies a new version of Stuxnet using a counterfeit digital certificate from JMicron Technology Corp.
February 2011	Release of patches from Microsoft fixing all the exploited vulnerabilities.
01 June 2011	Expiration date for using the Print Spooler vulnerability (see→*Operating mode*).
24 June 2012	Self-kill date (after this date Stuxnet auto-terminates itself).

(For a more exhaustive chronology of Stuxnet-related events, refer to Appendix F)

[3] http://www.wilderssecurity.com/showpost.php?p=1712134&postcount=22

[4] Source: Symantec (http://www.symantec.com/connect/fr/blogs/updated-w32stuxnet-dossier-available). Further attacks followed in July 2009, March 2010, April 2010 and May 2010. Shortest span between compile time and initial infection is 12 hours.

Operating mode

- Being alive on an infected system, Stuxnet tries to propagate itself in different ways:
  a) via the local network;
     - using the zero-day *Microsoft Windows Print Spooler Service Remote Code Execution Vulnerability (CVE-2010-2729 / BID 43073)*[5] or the two-year old *Microsoft Windows Server Service RPC Handling Remote Code Execution Vulnerability (CVE-2008-4250/ BID 31874)*[6]. The Print Spooler vulnerability consists of the acceptance of a specially crafted print request sent to a networked printer, containing arbitrary executable code which is run by the computer sharing the printer (the host computer is forced to write a dropper named `winsta.exe` in the `%SystemRoot%\system32` directory, and a file named `sysnullevent.mof` in the `%SystemRoot%\system32\wbem\mof` directory, which is then automatically executed as a WMI binary managed object file). Note also that Stuxnet will attempt to use this vulnerability only if the current date is before June 1, 2011.
     - by copying itself in accessible shared folders (using the security credential tokens of the users found in the local computer / domain or through a WMI Explorer impersonation).
     - by copying and executing itself on remote computers running a WinCC database server.

  b) via USB removable storage devices (mainly USB memory sticks);
     If Stuxnet detects that a USB storage device is connected to the system on which it resides, then it copies itself and generates on the USB device a specially crafted .LNK file, and waits for users of other systems to display its content. By doing so, they also get infected because of the (0-day) *Microsoft Windows Shortcut 'LNK/PIF' Files Automatic File Execution Vulnerability (CVE-2010-2568 / BID 41732)*[7] that Stuxnet is capable of exploiting.
     In earlier versions of Stuxnet, the worm was spreading via USB removable media through the usage of autorun.inf.

  c) via infection of STEP 7 folders.
     Stuxnet searches for STEP 7 projects (.S7P files) in the infected system. If it finds any of these, it modifies the main index files and copies itself in their folders. STEP 7 folders are often copied from one computer to another for documentation or development purposes. When a user opens such an infected folder on a clean system, the worm is executed and spreads the infection.

  To summarise, a total of seven methods are used by Stuxnet for spreading.

- Once inside a new system, depending on the Windows version found, the worm uses the 0-day vulnerability *Windows Task Scheduler Privilege Escalation Vulnerability (CVE-2010-3888)*[8] or *Windows Win32K Keyboard Layout Vulnerability (CVE-2010-2743)*[9] to gain elevated privileges and install a rootkit. In order to bei undetectable by anti-virus software and to have very privileged access to the host system, the rootkit functionality is installed as two hardware driver-level executable modules (device drivers `mrxnet.sys` and `mrxcls.sys`), which run in kernel mode. Since this is a "suspicious" operation, to do this Stuxnet uses counterfeit identification certificates to prove their origin from a trusted source to Windows.
  In order to be executed on every system start, the worm then sets the Windows registry entries `HKLM\System\CurrentControlSet\Services\MRXCLS` and `HKLM\System\CurrentControlSet\Services\MRXNET` so that the two drivers are started as services.

---

[5] Fixed with **KB2347920** patch, released by Microsoft on September 2010 within the **MS10-061** security bullettin. The exploit was already known in April 2009, when the security magazine Hakin9 described it (http://hakin9.org/print-your-shell).

[6] First used by the Conficker worm in November 2008. Fixed with **KB958644** patch, released by Microsoft on October 2008 within the **MS08-67** security bullettin.

[7] Fixed with **KB2286198** patch, released by Microsoft in August 2010 within the **MS10-046** security bullettin.

[8] Fixed with **KB2305420** patch, released by Microsoft in December 2010 within the **MS10-092** security bullettin (Vista/Win7).

[9] Fixed with **KB981957** patch, released by Microsoft on 12 October 2010 within the **MS10-073** security bullettin (WinXP).

- Stuxnet then starts its Remote Procedure Call (RPC) server and listens for incoming connections from other infected machines possibly residing on the local network. This feature enables an infected system to execute the following functions within any other infected machine to which it it can connect:
  - get the malware version
  - send a module and have it executed remotely in a new or in an existing (e.g. `lsass.exe`) process
  - download the worm dropper (built on-demand right at the time of the request)
  - run any specified application
  - read a file
  - write a file
  - delete a file

  The RPC server installed by Stuxnet in the infected systems is identified as a unique software object through the Globally Unique IDentifier (GUID) *000204e1-0000-0000-c000-000000000046*. Using this GUID, those systems are enabled to identify, communicate with, and update one another.
  This feature allows all malware instances to automatically update each other over the LAN, even if they cannot reach to the command-and-control (C&C) server due to a firewall or lack of Internet connectivity.

- Finally, it searches on the local computer for the Siemens WinCC software, which would indicate that the machine is a computer used for controlling an industrial PLC (Programmable Logic Controller), also known as a "Human-Machine Interface" (HMI) workstation.
  To determine if WinCC is installed, Stuxnet looks in the Windows system folder for the file S7OTBXDX.DLL, used by WinCC systems. Once found, it renames the file to S7OTBXSX.DLL and then replaces it with a modified version (extracted from the main wrapper file as resource 208). The new .DLL has the same exports as the original but with code modifications on the following functions:

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

  These functions are generally used to access, read, write, and delete code blocks on the PLC. In an infected system, when these functions are called, Stuxnet will execute additional instructions before calling the true functions contained in S7OTBXSX.DLL. By intercepting these functions, it can modify the data sent to or received from the PLC, acting as an MITM-like attack.

- Next, Stuxnet tries to contact a remote server. In attempting to do this, it first tests for an active Internet connection by trying to open an HTTP session to the following non-malicious URLs:
  - www.windowsupdate.com
  - www.msn.com

After a connection is established, it then connects to the following URL(s) to send and receive commands from a remote user:

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

It then generates the following URL and posts it to the server:

- `http://www.mypremierfutbol.com/index.php?data={data}`

Where {*data*} is a XOR encrypted hexadecimal value that contains the IP address, computer name, domain, OS version of the infected machine and whether WinCC or STEP 7 are installed or not.
The server may respond to the infected machine by sending back arbitrary code to be executed (most likely an updated version of the malware).

- As a next move, Stuxnet start a search for STEP 7 projects. (see first point, letter c)
- At this point, all the install and setup operations are done. Using the S7OTBXDX.DLL and the WinCC default credentials (userid=*WinCCConnect* password=*2WSXcder*)(*vulnerability CVE-2010-2772*), Stuxnet accesses the PLCs and verifies what type of CPU they have. If CPUs are type 6ES7-315-2 or 6ES7-417, then it checks what type of field devices are connected to them by reading the PLC's system data blocks (SDB). If the devices found are Vacon or Fararo Paya frequency drive converters, Stuxnet records the frequency configuration data set in the PLC, and then it begins intercepting commands, altering their operation. Stuxnet has the ability to upload its own attack code to the PLCs. By doing so, "Stuxnet changes the output frequency [of the converters] for short periods of time to 1410 Hz and then to 2 Hz and then to 1064 Hz. Modification of the output frequency essentially sabotages the automation system from operating properly",[10] causing mechanical stress to the centrifuges (which can lead to failure) and corrupting the quality of the processed uranium. The attack sequence – intended as the commands given to the frequency converters – is different depending on the CPU type.
- As a last move, to cover its tracks and finish its attack in a truly impeccable way, Stuxnet – after hijacking the sent commands – replays reassuring fake data to the operator (previously recorded), discarding the real ones coming from the PLC's sensors, so that everything on the HMI station looks to be in order. This is a PLC rootkit functionality, and so far seems to be the first one of its kind to appear in the wild.

Note that those operations described above are performed by the last version of Stuxnet. At least another two previous versions were discovered which use different operations. For instance, the .LNK exploit was added to the Stuxnet code only in March 2010. Previously, the worm was trying to spread across removable devices through the use of a specially crafted AUTORUN.INF file which also contained the worm itself (≈260 KB).

In summary, these are all the vulnerabilities exploited by Stuxnet:

	Vulnerability ID		MS	0-day	Vulnerability description
	CVE	BID			
1	CVE-2008-4250	31874	08-067	No	Windows Server Service RPC Handling Remote Code Execution
2	CVE-2010-2568	41732	10-046	**Yes**	Windows Shortcut 'LNK/PIF' Files Automatic File Execution
3	CVE-2010-2729	43073	10-061	**Yes**	Windows Print Spooler Service Remote Code Execution
4	CVE-2010-2743	43774	10-073	**Yes**	Windows Kernel Win32K.sys Keyboard Layout Privilege Escalation
5	CVE-2010-2772	41753	10-092	Yes	Siemens Simatic WinCC Default Password Security Bypass
6	CVE-2010-3888	44357	10-073	**Yes**	Windows Task Scheduler Privilege Escalation

Tab. 1: List of the vulnerabilities exploited by Stuxnet

[10] Eric Chien, Symantec's Connect Security Response blog (http://www.symantec.com/connect/blogs/stuxnet-breakthrough), 12-Nov-2010

Technical considerations

The capability of causing real damage through PLCs' code injection makes Stuxnet a milestone in malware history. Before its appearance on the world stage, cyber attacks were limited to website defacements, denial-of-service attacks, or unauthorised accesses into systems to steal sensitive data or to cause classic system disruption/ shutdown.
With Stuxnet, the games have changed. Stuxnet "*is a carefully developed, specifically targeted malware, intended to cause subtle but substantial damage to key infrastructure. While a defacement or a DoS attack could be recovered within some days or weeks, Stuxnet-like attacks have the capability to set back their victims by years.*"[11]
The right definition for the outcome of this "intelligent" malware could be "continuous stealth disruption", meaning that the damages are protracted over a long period of time but in a way that they remain virtually undetectable until it is too late to recover without large losses.
Due to its refined capability to strongly radicate in a system, hide itself and exfiltrate data, Stuxnet, from an intelligence perspective, can be defined as an effective "*Advanced Persistent Threat*".[12]
It is a worm, but does not require networks to spread, nor need a host to be transported. This is not new, but nowadays quite obsolete. Not relying on networks for diffusion is a strategy that was common to "first-generation" viruses – those born in the early 90s replicated and spread themselves through the use of removable disks (i.e. *boot sector* viruses). The Stuxnet's writers did this because SCADA systems are not normally connected to the Internet for security reasons.

A short history of malware exploiting removable media

Malware infections vectored through removable media are not new. In fact, most of the early personal computer infections were passed by floppy disks because at that time PCs were not yet connected to networks. However, those primitive viruses did need a direct user intervention to replicate, i.e. the user had to recall and execute them.

The following is a historical summary of malware able to replicate by disks, CDs, and USB devices:

- *Elk Cloner*: the earliest known malware to spread by removable media. Released in 1982, this virus attacked the Apple II DOS v3.3.
- *Brain*: the first MS DOS floppy disk-based virus. Showed up in 1986.
- *Roron*: in 2002, the Roron worm made use of autorun.inf on networked drives.
- *Bancos*: in 2004, the Bancos worm exploited CD-ROM discs.
- *SillyFD-AA*: the first malware to spread via USB; this 2007 worm did so by creating its own autorun.inf file.
- *Conficker*: probably the most known and sophisticated worm (after Stuxnet), spread in 2008.
- *Stuxnet*: it is worth pointing out that this worm uses *both* the autorun.inf method and also a zero-day vulnerability in the .LNK files to spread through removable storage devices.

It can use *multiple* zero-days exploits to gain access to a Windows computer, but can also infect a machine through known vulnerabilities or simple network shares.
Stuxnet uses Windows computers only as a means of spreading and delivering its payload.
Once a system is infected, Stuxnet hides inside rootkit and waits, checking every five seconds to see if its exact parameters are met on the system.
Stuxnet was designed to work silently and efficiently, most likely to maximise its life expectancy in infected systems.
Basically, Stuxnet uses 'man-in-the-app' attacks, intercepting commands sent to and answers received from the PLC.

[11] Peter Bright, in blog article "*Stuxnet apparently as effective as a military strike*" (http://arstechnica.com/tech-policy/news/2010/12/stuxnet-apparently-as-effective-as-a-military-strike.ars).
[12] Ralph Langner, a worldwide known industrial security analyst, has also used for Stuxnet this attribute.

Targeting and final payload

Stuxnet is extremely selective in delivering its final payload, i.e. in modifying the PLC's operations. This happens only if the malware is executed in an environment with the following very specific software and hardware configuration:

- the Windows computer (on which Stuxnet is alive) must be:
  1) running Siemens WinCC software (Siemens control environment for Simatic PLCs),
  2) connected to a S7 Simatic PLC;
- the connected PLC must be:
  3) equipped with a 6ES7-315-2 (S7-300) or a 6ES7-417 CPU (S7-400),
  4) equipped with one or more Profibus CP 342-5 communications processor modules (detected through hex value `2CCB0001` found at offset `50`h in SDBs),
  5) connected to at least 33 Vacon-branded or Fararo Paya-branded frequency converter drives (detected by counting multiple hex values `7050` and/or `9500` found in SDBs);
- the frequency converters must be:
  6) operating at a speed between 807 and 1210 Hz.

Fig. 4: The Vacon NX frequency driver (a.k.a. frequency converter). This device is controlled by the S7 PLC through a CP-342 module, and regulates accordingly the speed of the electric motor of the centrifuge to which is connected.

When *all* the above conditions exist, Stuxnet finally recognises its ideal target, enters the PLC's memory and changes the program running inside of it. It acts very discreetly (we could say "stealthily"), and because of this even more "smartly" than one could think.
Stuxnet does not generate a highly disruptive incident, which could potentially lead to some kind of explosion or dangerous event. It just modifies, in an unperceivable way, the normal operation of the electrical engines which drive the centrifuges. In the nuclear enrichment process, the centrifuges need to rotate at a definite high speed for long periods of time in order to extract the enriched uranium. According to ISIS,[13] the nominal rotation frequency of the IR-1 centrifuges used at the Iranian enrichement site of Natanz is 1064 Hz, but it seems that the frequency was kept lower to reduce breakage (probably at 1007 Hz, according to other sources).
If the centrifuges are abruptly stopped while they spin at that high speed, or suddenly accelerated when they rotate at low speed, then not only can mechanical stress be induced which could lead to a failure, but the process of isolating the heavier isotopes in those centrifuges can also be disrupted, causing the final grade of obtained uranium to be of lower quality.

In summary, the worm goes into sabotage action if and only if a large number of very specific conditions are met. It examines the connected Programmable Logic Controller and checks its model of processor. Then it proceeds to check if there are particular data at a particular offset in two System Data Blocks. It is designed to sabotage not only one particular PLC, but – more appropriately – one particular PLC connected to one particular type of device (in a certain number) operating within specific parameters; a very specific combination of hardware and data that will match maybe only a few dozen systems in a world of millions of industrial automations.
Once the malware finds its victim, it inserts a few lines of a low level language called MC7 (used for programming Siemens PLCs) at the top of the OB1 and OB35 memory blocks. OB35 is a particular memory address (a.k.a. "watchdog block") where a special routine, used to control highly critical processes that need very fast responses, is located. This routine takes priority over any other process or event, and is run every 100 milliseconds. The injected code in OB35 is placed at the very top of the loop and waits for a specific event to occur, i.e. the sabotage routine. When the event occurs the injected code prevents the execution of any other "high-priority" routine (that could be contained in

[13] Institute for Science and International Security. See also →*ISIS* (Glossary).

the rest of the OB35), rendering whatever safety control mechanism – such as an emergency shutdown – that exists ineffective.
It is interesting to note that the MC7 code injected in the PLC varies depending on the type of the detected frequency drivers, confirming once again the very deep knowledge possessed by the malware's writers about the enrichment hardware. They seemed to know exactly the best "attack sequence" needed to damage the frequency drivers.
To be precise, Stuxnet contains in its main body three different PLC infection sequences, which have been dubbed attack sequences "A", "B", and "C". While sequences A and B are conceived to be injected into S7-300 PLCs (equipped with CPU 6ES7-315-2), sequence C was developed to be injected in S7-400 PLCs (equipped with CPU 6ES7-417) but appears to have been disabled, probably because it was not completed.
Sequence A is optimised for Vacon drivers, and sequence B for Fararo Paya drivers.

The fact that Stuxnet delivers its final disruptive payload only to very specific PLCs implies that, for the home user, there is not a such a great danger, considering also that the worm has a built-in auto-destruction feature that will make it kill itself after June 24, 2012. Wherever (or whenever) the searched conditions are not met, Stuxnet remains dormant, causing no damage.

Self-upgrade

Like other advanced malware (e.g. Conficker), Stuxnet is able to upgrade itself. It can do this in two ways:

- by connecting to its C&C server (if Internet connectivity is available), using the HTTP protocol;
- by contacting other infected machines which are found on the same local network through a peer-to-peer (P2P) mechanism (RPC client/ server) and asking which version of the worm they are running. If these machines are infected with a more recent version, Stuxnet will download from them the new version and install this latest one via a process injection.
  The interesting thing here is that if an older version is found on the contacted machine, then the roles are reversed. In this case the requestor will become the server, and will send to the remote computer a copy of itself. This will ensure that new versions of the worm are deployed as much as possible.

This feature is essential for the effectiveness of the worm, as it gives it the capability to adapt and evolve, and theorically could also give it the capability to survive new threats (i.e. by implementing countermeasures to new ways of being detected or neutralised).

Moreover, this self-upgrading capability can be exploited and used as a backdoor functionality, since the C&C server could instruct the worm to download any type of code (including Remote Access Tools or immediate self-kill instructions, if necessary).

Digital certificates impairment

One of the most interesting features discovered in Stuxnet is its capability of installing device drivers by using counterfeit (but still valid) digital certificates.
Starting with Windows Vista and Windows Server 2008, Microsoft introduced driver signing requirements to enforce trust relationships between kernel-level code and the publishers of that code. Basically this means that if a driver is not digitally signed, Windows will stop installing it and will alert the user (see Figure 5).

Fig. 5: The security warning message issued by Windows 7 for drivers not digitally signed.

The Stuxnet components that are installed as device drivers were signed with certificates issued to Realtek Semiconductors and JMicron, two companies that both have offices in Hsinchu Science Park, Taiwan, which could indicate the possibility of physical theft of their private key. The second counterfeit certificate was generated and used[14] shortly after Stuxnet was discovered (July 2010). This could indicate that Stuxnet had not completed its goal, and consequently it was necessary for its authors to give it a new fake "identity" to allow the worm to continue its job.
In today's computer security world, usage of digital certificates to sign files (programs, data, documents or anything else) is believed to be one of the most secure ways to prove the origin of a file is from a trusted source. Digital certificates and signatures can be considered as the main pillar of secure authentication and, therefore, play an essential role in distributing data: files digitally signed by trusted manufacturers are considered as safe.

How digital certificates work

The working principle of digital certificates relies on *asymmetric encryption*. To be extremely concise, asymmetric encryption is a type of encryption in which the key used for encrypting a plain text to cipher text is different to the one used for decrypting the same cipher text back to the original plain text. This allows the spread of the *encrypting* key to everyone (therefore, called the *public* key) while retaining strictly secret the *decrypting* key (therefore, called the *private* key).
The file in which the public key – along with other ID information – is released to everyone is called a *digital certificate*.
To let everyone know the public key of any person within an organisation that uses asymmetric encryption, special servers acting as a centralised repositories for certificates are available, on which directory services are run – typically by means of LDAP (Lightweight Directory Access Protocol) – for quickly retrieving the requested data.
By using your own private key (assuming that you are the only one to know it), you can prove your identity through the use of a simple challenge/ response: anyone who wants to verify your identity will send to you some specific data encrypted with the public key and will wait to receive back the plain form. If your answer matches the unencrypted data then you are trusted to be who you claim to be.
A good way to securely store private keys is to have them hard-coded inside smart cards, as not even the legitimate user can know or directly access them.

As stated in the above explanations, it is obvious that the reliability of a certificate is entirely based on the fact that the related private key is securely stored by its owner in a highly safeguarded

[14] embedded within the device driver file `jmidebs.sys`. It is worth noting that this file has a compilation date of July 14th, 2010, a day before the news about Stuxnet broke out.

environment (like an anti-tampering, externally unaccessible memory storage device installed in a physically well-protected location).
If for any reason the storage security is bypassed, and the secrecy of the private key is compromised, the entire trust system collapses. This is the reason why private keys – especially those used by software companies for signing their productcs, as this could lead to potentially disastrous security incidents – should be kept in very secure storage, in the same way as done by Certification Authorities and sub- Certification Authorities through the usage of so-called HSMs (Hardware Security Modules).
Since the private key is a physical file, it can potentially be stolen just like any other digital data. In the Stuxnet case it is still unclear how such confidential information as private keys could have been exfiltrated. Basically, this could have occurred in two ways: the worm's authors may have paid insiders to obtain them, or they may have stolen them by resorting to direct hacking or to some type of infostealing malware (e.g. some variants of the ZeuS banking trojan appear to have the ability to steal client-side certificates[15]).

Replication

Although Stuxnet does replicate through networks, its main way to infect machines is through USB sticks. This means that someone needed to physically bring a USB stick to the SCADA system and connect it. This action could have been carried out – knowingly or otherwise – by an external contractor (e.g. the Russian technicians from RosAtom, or even some technical representative from Siemens).

Someone has argued[16] about this aspect that Stuxnet "contains some glaring shortcomings" (see also: → *The mistakes of Stuxnet*), because it infected a massive number of computer systems around the world rather than what appears to have been its intended target, namely the computers connected to the Iranian centrifuges in the Natanz enrichment plant.
Why have the authors not implemented a simple geolocation check to narrow the spreading? Simply because this would not have worked. Normally SCADA systems are not connected to the Internet for security reasons, so they don't use the same IP addressing schema. They use internal IP addressing (RFC 1918), which gives absolutely no information on their geographic location. Even if they had not used RFC 1918 addressing (i.e. IANA-assigned public IP addresses), there would have been a replication of existing addresses (since those networks are isolated) and, therefore, again totally inconsistent with their true geographical position.
However, if the networks were isolated, why try to connect outside and to exfiltrate data? Because the SCADA networks were not the only ones expected to be infected. Internet-connected networks were also the targets, for the above mentioned reasons (exfiltration and update), and the "bridge" between these two types of network were the USB removable media, widely exploited.

An even more realistic explanation is that the authors simply did not have the time to take this detail into consideration. Their goal to stop the Iranian nuclear programme may have been so pressing that they decided that an unlimited spreading was an acceptable collateral effect.

There is also another explanation: it is possible that, to achieve the desired goals, Stuxnet's designers needed a massive spreading of the worm within Iran (i.e. from one computer to another within different zones of the same facility, or between facilities), but there was no countermeasure to prevent it from propagating further. In fact, there is an even bigger and much less visible advantage to use a weapon like Stuxnet on a large scale: this lies in the fact that Stuxnet has the potential to hit not only Natanz, but also any other nuclear enrichment site with a similar configuration, even those not yet discovered. To Stuxnet, infecting a known site or a top secret one does not make any difference: what the worm looks for in its spread is a combination of well-defined elements. If they are all present,

[15] "ZeuS on the Hunt", by Dmitri Tarakanov (http://www.securelist.com/en/analysis/204792107#2). For more information, see also: http://www.thetechherald.com/articles/Zeus-botnet-plundering-the-masses-and-snatching-certificates.

[16] John Markoff, newspaper article "*A Silent Attack, but Not a Subtle One*", 27-Sep-2010, The New York Times (http://www.nytimes.com/2010/09/27/technology/27virus.html) .

then Stuxnet understands to be in the presence of an industrial plant commissioned to refining uranium, and sabotages it. And, as we have seen, the fact that a site is not connected to the Internet (or even to a closed network) is not enough to prevent Stuxnet from infecting it. What the ear and the arm of traditional intelligence cannot reach, cyber weapons may...

Data exfiltration

Because of its data exfiltration (it evades personal and corporate firewalls by injecting itself into the `iexplorer.exe` process) and self-upgrading capability, Stuxnet can be considered a highly environment-adaptive malware. This is also confirmed by the fact that at least three different versions of this worm were found in the wild. It is very likely that, as Stuxnet was providing its creators with the projects of the SCADA systems stolen from the computers it was infecting, new, more targeted and more effective versions were developed and released on the "battlefield".
An interesting finding about the age of Stuxnet can be deduced from the information related to the two URLs the worm tries to connect to, behind which the command-and-control (C&C) servers reside.
The DNS registration data (whois) gathered for the domain *mypremierfutbol.com* gives this result:

```
Registrant:
  Domains by Proxy, Inc.
  DomainsByProxy.com
  15111 N. Hayden Rd., Ste 160, PMB 353
  Scottsdale, Arizona
  United States

  Domain Name: MYPREMIERFUTBOL.COM
     Created on: 24-Dec-08
     Expires on: 24-Dec-10
     Last Updated on: 24-Dec-08

  Administrative Contact:
     Private, Registration mypremierfutbol.com@domainsbyproxy.com
```

The result of whois mypremierfutbol.com

Of course the date on which the domain was registered (24 December 2008) does not indicate the exact release of the malware, but gives a good idea about the preceding development times and suggests that the worm was ready to be deployed at this time.

To be precise, Stuxnet does not implement a specific feature focused on exfiltrating data. Rather it reports to its C&C whether or not STEP 7 projects are found on the infected machine. However, Stuxnet can be easily "taught" to do anything thanks to its self-upgrading capability, which allows the worm to receive and execute additional code. So, when Stuxnet was finding valuable STEP 7 projects, it is very likely that then an additional "info-stealing" tool was sent to Stuxnet to upload those files to the C&C server.
An interesting consideration is that, thanks to Stuxnet's data exfiltration capabilities, whoever was controlling the malware should have collected enough information to understand how the centrifuges were used, either for Low Enriched Uranium (power plant fuel) or Highly Enriched Uranium (suitable for warheads) production, and should thus be substantially aware if Iran has been producing fuel or warheads.

Stealth capabilities

Not only can Stuxnet steal code and design projects, but it can also hide itself using classic Windows rootkit behaviours.
Stuxnet uses different methods to hide itself and its operations:

- Directory list filtering
  In the early stage of the infection[17], the file `~WTR4141.tmp` hooks the following APIs exported by `Kernel32.dll` to hide the malware files in the removable drives:
  - `FindFirstFileW`
  - `FindNextFileW`
  - `FindFirstFileExW`
  - `NtQueryDirectoryFile`
  - `ZwQueryDirectoryFile`

  After the rootkit functionalities are installed, the file `MrxNet.sys` will intercept commands sent to the file systems, and will filter out (hide) from the results any file which:
  - ends with a .LNK extension and has a size of 4171 bytes;
  - is named `~WTRnnnn.TMP` and has a size between 4 KB and 8 MB, where '`n`' is a decimal digit, such that the sum of the four digits must be 0 or a multiple of ten (for example: 4+4+0+2=10).
- File timestamp modification
  When copying PNF files in `%Windir%\inf`, Stuxnet sets their timestamp to match those of other PNF files in the directory.
- Debugging awareness
  Stuxnet will abort its execution at startup (when `MrxCls.sys` is loaded) if it detects that Windows is started in Safe Mode or if kernel-mode debugging is enabled. In addition, processes that would usually be targets for injection are not targeted if they are being debugged.
- DLL mapping to memory
  Instead of drives, Stuxnet decrypts and writes its DLL files in memory, from where they can be recalled without using the *LoadLibrary* method, which is normally monitored by anti-malware programs. This sophisticated stealth feature (unseen before Stuxnet) prevents executable code from being written to disk, which could be considered suspicious, and bypasses checks of host-based IDS.
- DLL injection in running processes
  When Stuxnet needs to load and execute a specific malicious function, it does not create its own process but injects code in a process already running. The processes chosen for injection are those listed below, which being anti-malware tools are considered as trusted, and also other system-related trusted processes such as `lsass.exe`, `svchost.exe` and `services.exe`.
  Stuxnet uses any of the following 10 processes if found in memory (all security related):
  - `avp.exe` (Kaspersky Anti Virus)
  - `avguard.exe` (AntiVir)
  - `bdagent.exe` (BitDefender)
  - `ccSvcHst.exe` (Symantec Common Client Norton Antivirus)
  - `ekrn.exe` (Eset NOD32)
  - `fsdfwd.exe` (F-Secure)
  - `Mcshield.exe` (McAfee)
  - `rtvscan.exe` (Symantec)
  - `tmpproxy.exe` (Trend Pc-Cillin)
  - `UmxCfg.exe` (Computer Associates eTrust Antivirus)

  Stuxnet also injects code in the `iexplore.exe` process to communicate with its C&C server, in order to bypass firewalls.

Moreover, Stuxnet is also able to hide the code injected in the PLCs and the resulting outcomes (therefore, being the first publicly known "PLC rootkit"), preventing the examination of the following objects:

[17] As soon as the .LNK vulnerability is exploited (infection via USB removable media).

- Executable code (organisation blocks) in PLCs
  Stuxnet intercepts commands aimed to read these code blocks and returns fake data, so when anyone using an infected computer (typically, a HMI station) tries to view the code blocks on a PLC he/she will not see the code injected by the malware but, instead, the code precedent to the injection. This is done by hooking the enumeration, read, and write functions of the `s7otbxdx.dll` file, a DLL responsible for accessing the PLCs from within the WinCC environment.
- Feedback process data from PLCs to HMI stations
  Stuxnet, before starting to modify the PLCs' operation, will record the values read by the PLCs' sensors, and "re-play" them back later to the operators, discarding the real feedback data ("*like a pre-recorded security tape in a bank heist, so that it would appear that everything was operating normally*"[18]).

Encryption and obfuscation

According to many malware experts, encryption and obfuscation are not the smartest features of Stuxnet. Stuxnet uses only limited encryption and obfuscation techniques, namely as follows:

- It applies several layers of encryption for executable code portions (a.k.a. "onion" encryption)
- it uses very weak XOR encryption for sending *to* the C&C server information about the infected machine (data is XOR-ed with the hex value `FF`)
- it uses weak XOR encryption for the HTTP payload received *from* the C&C server (XOR-ed with the following 31-byte hexadecimal string: `67A96E28900D58D6A45DE27266C04A57885AB05C6E45561ABD7C715E42E4C1`)
- it uses XOR, NOT and bit-shift encryption for PNF configuration files stored in `C:\Windows\inf` (weak encryption)
- the rootkit files (`mrxcls.sys` and `mrxnet.sys`, dropped in `C:\Windows\System32\Drivers`) are encrypted as resource 201 and resource 204 in the main .DLL file
- the peer-to-peer networking feature built into the worm is encrypted to FIPS 140-2 standards[19].

Compared with Stuxnet, other malware such as Conficker used much more complex encryption techniques[20] and were heavily obfuscated. Some malware analysts have also observed that Stuxnet would have been much more difficult to reverse engineer and detect by antivirus software if the worm had used a polymorphic engine to disguise itself, a technique now as much as some 20 years old.[21]
However, Stuxnet uses proprietary C/C++ structures for hosting the executable code and for all data passed to the main subroutines that are not recognised by disassemblers and decompilers, therefore, resulting in some kind of "plain" obfuscation.

Additional considerations

With its elegant and extremely targeted architecture, Stuxnet demonstrates a perfect combination of classical intelligence (on-site operatives), cyber intelligence (Project 7 data exfiltration) and digital warfare (PLC attack). This had never been seen before, and is one of the main reasons investigators were persuaded that it was a result reachable only by a well-organised state. Ralph Langner likens its complexity to "*the arrival of an F-35 fighter jet on a World War I battlefield*".

The Stuxnet malware is extraordinary in three aspects. First, Stuxnet shows a level of sophistication and technical excellence never seen before, which surpasses any previous worm.

---

[18] Newspaper article: "*Israeli Test on Worm Called Crucial in Iran Nuclear Delay*", by William J. Broad, John Markoff and David E. Sanger (published January 15, 2011; http://nyti.ms/esygjV).

[19] as disclosed by Brian Tillet of Symantec at IdentEvent 2010 (reporteb by The Atlantic, 4 Nov. 2010, http://www.theatlantic.com/technology/archive/2010/11/the-stuxnet-worm-more-than-30-people-built-it/66156/).

[20] Conficker was using MD6 related crypto algorithms and even signing digitally updates and P2P communications.

[21] The first polymorphic (i.e. dynamically self- mutating) virus was first developed between 1991 and 1992 by a Bulgarian virus writer called Dark Avenger (whose identity has never been discovered).

Secondly, it was designed to target specific programmable logic controllers. To date, this has been the first instance of industrial cyber-sabotage achieved through malware. Since attacks like the one performed by Stuxnet can inflict substantial physical damage, the border between the digital and the real world has now become less definite, and this points out a new set of problems that the cyber security community will have to deal with.
Finally, Stuxnet uses valid counterfeit digital certificates, issued with the private key of their owner, which is a novelty for malware.

Considering their combination of multiple features and complexity, threats like Stuxnet are rather new in the cyber security panorama, and are especially interesting to nation states for two reasons: they combine cyber intelligence and cyber sabotage in one tool, and they are basically impossible to trace to their creators and/or users. Under this perspective, Stuxnet is a real breakthrough and represents a weapon of substantial power which can effectively provide returns on a strategic level.

In his presentation[22] about how to build a cyber army, Charlie Miller hypothesised that enemy networks or hosts that are isolated from the Internet could still be reached thanks to hidden 3G modems that on-site operatives would have secretly installed. The creators of Stuxnet chose a different approach that seems to be even more effective as it leaves no hardware that could be discovered and does not need to rely on the presence of a cellular network.

"*What we're seeing with Stuxnet is the first view of something new that doesn't need outside guidance by a human – but can still take control of your infrastructure*," said Michael Assante, former chief of industrial control systems cyber security research at the US Department of Energy's Idaho National Laboratory. "*This is the first direct example of weaponized software, highly customized and designed to find a particular target*."

The application vulnerability contained in the WinCC software, which consists of a hard-coded authentication in the database back-end, deserves separate discussion.
OWASP (the Open Web Application Security Project) defines this type of vulnerability as "high" in severity and even "very high" in the likelihood of being exploited. Even more clearly, OWASP explicitly reports in its web pages[23] that "*if hard-coded passwords are used, it is almost certain that malicious users will gain access through the account in question*".
To further complicate this issue, it seems that – according to Siemens[24] – these default credentials must be kept in place, or the SCADA components will not interoperate.
Needless to say, implementing a system design of an ICS in such a way is at the very least hazardous, and is rather unprofessional from a company as large as Siemens.

Many have remarked that having a Windows operating system connected to industrial control devices is unacceptable from a cyber security perspective. Unfortunately, having Windows in a control room is a very common situation. All the big SCADA/ HMI packages (the so-called visualisation applications that display to operators what is going on in a controlled process) are built on Windows WinCC, Wonderware inTouch, or PCIM. Only very few factories implement visualisation through Unix, Linux or QNX.

[22] CCDCOE Conference on Cyber Conflict, June 2010, Tallinn (www.ccdcoe.org/conference2010).

[23] https://www.owasp.org/index.php/Use_of_hard-coded_password.

[24] On 19 July 2010, Siemens spokesman Michael Krampe said in a statement that customer guidance given to customers to counter Stuxnet "won't include advice to change default settings as that could impact plant operations".

Summary of Stuxnet's features

- Extremely targeted, multiple triggers.
- Six vulnerabilities exploited (of which four are operating system zero-days and one an application zero-day). Never heard of before in malware history.
- Uses valid certificates which appear to be issued by trusted parties to install rootkit as device drivers.
- Reports information about the infected machine to its C&C server.
- Can download upgrades or other malware from its C&C server.
- Designed to sabotage centrifuges used for the uranium enrichment process.
- Extensive knowledge of Windows, Simatic architecture, and uranium enrichement process ICSs.
- Good knowledge of SQL.
- Refined stealth and rootkit capabilities.
- Fail-safes: self-terminating on 24-6-2012 (two checks), limitations set[25] on propagations.

Comparative analysis with other malwares

A quick comparative overview of Stuxnet and the most significant worms spread in the last decade is given in Table 2. The table allows the reader to compare which feature belongs to each worm, identifying commonalities and differences.

Some features of Stuxnet were already observed in previous infections of other malware, such as Conficker (one of the most complex worms ever seen before Stuxnet). In contrast, some others are very specific to Stuxnet, if not unique (highlighted in bold font). Specifically, the most remarkable features that make Stuxnet unique are its usage of five zero-day vulnerabilities, the use of valid digital certificates and the fact that it targets industrial PLCs.

The penultimate column of the table contains data particular to Slammer (a.k.a. "Sapphire"), an incredibly virulent worm that crippled the Internet in 2003. The Slammer data have been reported here to illustrate how different it was to Stuxnet, and to highlight the progress in malware evolution. Although both worms are very efficient at propagating, they can be considered as opposite: Stuxnet is large, silent, subtle, extremely targeted, slowly proceeding and yet very sophisticated in its action; Slammer is very small, ultra-fast, does not try to hide itself and does not install anything (no files, it only "lives" in memory), yet it is highly disruptive and generates a extremely large amount of DDoS. The attack of Stuxnet on the Iranian industrial systems lasted more than one year; Slammer flooded the Internet in just two hours,[26] infecting 75000 hosts in the first 10 minutes.
If we wanted to make an analogy with ways of causing problems for a town, we could say that Slammer instantly shuts down the electrical power, while Stuxnet releases micro-quantities of arsenic into the aqueduct each day for several months. The first operating mode is immediate and very noticeable, but after some hours of blackout you can recover. The second one is very slow and would be noticeable only after months, but it would end up killing you.

[25] By Michael Joseph Gross, April 2011, http://www.vanityfair.com/culture/features/2011/04/stuxnet-201104.

[26] Infection started at 05:30 UTC on January 25, 2003. According to [David Moore, Vern Paxson, Stefan Savage, Colleen Shannon, Stuart Staniford, Nicholas Weaver, "Inside the Slammer Worm," IEEE Security and Privacy, pp. 33-39, July-August, 2003], Slammer was the fastest spreading worm in malware history, capable of attaining a peak scanning rate of 55 million hosts per second.

Feature	Purpose (specific to Stuxnet or generic)	*Stuxnet (2010)*	*Conficker (2008)*	*Sasser (2004)*	*Slammer (2003)*	*Blaster (2003)*
Obfuscation / Encryption	Reverse engineering countermeasure	Y	Y	Y	N	N
Compression (UPX or other)	Reverse engineering countermeasure, size limitation	Y	Y	Y	N	Y
Dynamic mutation (polymorphic or metamorphic code)	Escape antivirus fingerprinting/reverse engineering	N	Y	N	N	N
Replicates through network vulnerability (worm)	No user intervention required (distinctive worm feature)	Y	Y	Y	Y	Y
Uses 0-day vulnerability	No protection against exploited vulnerability in the short run	Y	Y	N	N	N
Uses multiples 0-day vulnerabilities	Maximise probability and speed of infection	Y	N	N	N	N
Installs rootkit/exploitable command shell	Remote access, invisibility to user and programs	Y	Y	N	N	Y
Installs network server	Remote download (replication/ update method)	Y	Y	Y	N	Y
Kernel-mode loader	Avoid detection by security programs (no files)	Y	N	N	N	N
Uses valid certificates for driver installation	Device driver-level rootkit installation	Y	N	N	N	N
Replicates through network shares	Replication efficiency	Y	Y	N	N	N
Replicates through admin shares	Replication efficiency	N	Y	N	N	N
Replicates through removable storage	Network-free autonomous replication	Y	Y	N	N	N
Replicates through emails (mass-mailing worm)	Replication efficiency (normally user-driven)	N	N	Y	N	N
Replicates through file infection (virus)	Replication efficiency	N	N	N	N	N
Replicates through web browser vulnerability	Replication efficiency, user-driven	N	Y	N	N	N
Uses password dictionary attacks	Identity theft or illegal access to protected resources	N	Y	N	N	N
Self-updating	[Adaptative] evolution. Maximise flexibility/survival capability	Y	Y	N	N	N
Exfiltrates data	Intelligence	Y	N	N	N	N
Exfiltrates user credentials	Identity theft	N	N	N	N	N
Installs keylogger	Intelligence, spyware, data theft	N	N	N	N	N
Installs as system service	Auto-run at system startup	Y	Y	N	N	N
Modifies "Run" registry key	Auto-run at system startup	N	Y	Y	N	Y
Creates custom registry keys (apart from above)	Internal data saving, duplicate infection avoidance	Y	Y	N	N	N
Targets Programmable Logic Controllers (PLCs)	Ultimate target service	Y	N	N	N	N
Spam generator (spambot)	Ultimate target service	N	Y	N	N	N
DoS generator	Ultimate target service	N		N	Y	Y
Scareware/ ransomware installation	Ultimate target service	N	Y	N	N	N
Uses stealth techniques	Escape antimalware programs / user awareness	Y	Y		N	
Disables anti-virus/ anti-malware software	Avoid detection by security programs	Y	Y	N	N	N
Disables auto-update	Avoid vulnerability patching	N	Y	N	N	N
File size (unexecuted, in bytes)	-	≈501 K	≈160 K	15872	376	6176

Tab. 2: Comparison of features of famous worms

Technical development of Stuxnet

Stuxnet was mostly written in a C/C++ development environment. An analysis of the Stuxnet executables files shows the following characteristics:

- Use of C++
- Use of C++ exception handling
- Use of C++ classes
- Use of C structures for all data passed to the main subroutines (over 40 user-defined structures)

Some other interesting information can be obtained from timestamps and headers of the Stuxnet components. For example, the file `~wtr4141.tmp` has a timestamp which indicates that the date of compilation is 03/02/2010. Version 9.0 of the linker indicated that attackers used MS Visual Studio 2008 for developing Stuxnet's components. File `~wtr4141.tmp` is digitally signed, and the timestamp indicates that the signature on the date of signing coincides with the time of compilation.

MRXCLS.sys (the module that loads in kernel mode the different components of the worm upon need) is probably the most interesting piece of software in the Stuxnet code structure. In his excellent technical analysis[27] of this software module, Geoff Chappell cleverly emphasises the extreme "independence" of MRXCLS.sys from Stuxnet's other components, noting that it is so general that it could be used to load *any* type of malware: MRXCLS.sys "simply" reads from a configuration file the names of the processes whose execution it has to watch for, and for each such process the names of DLLs it has to inject into that process's address space.
"*Although MRXCLS may presently be distributed only with Stuxnet, its code knows absolutely nothing about anything else in Stuxnet […] It's even possible that the Stuxnet writers don't have the MRXCLS source code*", Chappell says.
The timestamp of MRXCLS.sys is also very interesting, since it indicates that it was compiled on the 1st January 2009. The fact that version 8.0 of the linker was used to build it suggests that MS Visual Studio 2005 was used for its development. Using a different version of the linker may again indicate that the software development was done by a group of people with a clear division of responsibilities.

Once again, these are strong indicators that Stuxnet's different parts were developed independently. This is particularly believed to be case for MRXCLS.sys, which could have even been bought from an external party.

Investigations of malware samples gathered in the wild show that the development of Stuxnet went through at least four different versions:

1) a first version not yet using the LNK vulnerability, without signed driver files (June 2009). This was probably a "reconnaissance" version, sent out to gather the SCADA blueprints from the Iranian STEP 7 computers (see: *Technical consideration →Data exfiltration*) which it was able to infect;
2) a second version using the LNK vulnerability and a counterfeit digital certificate from RealTek (March 2010);
3) a third version, which differs from the second version only by an increase in the amount of time the worm remains on USB removable media, from 21 to 90 days (April 2010);
4) a fourth version using a counterfeit digital certificate from JMicron (July 2010).

It is also possible that a link between Stuxnet and Conficker exists, for the following reasons:

- Conficker first appeared in November 2008 (*Conficker.A*), and other four variants (from *Conficker.B* to *Conficker.E*) followed up until April 2009. The first variant of Stuxnet was found in June 2009.
- Both exploit the MS08-067 vulnerability.
- Both use USB mass storage devices to spread.
- Both are uncommonly complex.

[27] Geoff Chappell, " *The MRXCLS.SYS Malware Loader* " (http://www.geoffchappell.com/notes/security/stuxnet/mrxcls.htm).

It is probable that the authors of Stuxnet studied Conficker to learn from it. It is worth remembering that the only non-zero day vulnerability used by Stuxnet is the CVE-2008-4250 (MS08-067), i.e. the one disclosed by Conficker in 2008/2009. John Bumgarner, a well-known cyber security expert, was even moved to say that Conficker.C could have been used in 2009 by attackers to open backdoors on computers in Iran, so they could then be infected with Stuxnet.[28] Bumgarner's assertions – which are not fully shared in the security community – are mainly based on three facts: the two worms exploit a common vulnerability, their files have overlapping timestamps and both seem to have no evident financial return.

Manpower

According to Symantec's security expert, Brian Tillett, traces of more than 30 programmers were found in the Stuxnet source code.[29] The task of testing the worm in a faithful test bed site alone would have taken 10 developers at least six months.[30]
Other sources hypothesise that "*building the worm cost at least 3 million dollars and required a team of as many as 10 skilled programmers working about six months*". [31]
Microsoft has estimated that at least 30 cyber experts have together spent more than 10,000 man-days (equivalent to 27 man-years) in the project.
According to Siemens engineers, to create this malware would take months if not years of work if done by one person.
The New York Times reported that Stuxnet was developed jointly by Americans and Israelis over the past two years (see later → *Origins of Stuxnet*).
Finally, Langner's estimation for Stuxnet code is 15000 Lines of Code.

By interpolating the above estimations, and correlating them with those from Charlie Miller, Langner, and other OSINT sources, the following data can be empirically hypothesised:

Profile	**Number**
ICS consultant	2
SCADA/ PLC architect/ engineer	2
Simatic PLC programmer	3
Nuclear fuel production expert	2
Windows internal system programmer	5-10
Vulnerability analyst	2
Exploit writer	3
Quality assurance operator	3
Lab field tester	3
IT/ C&C infrastructure maintainer	5
On-site intelligence operator	1-10?
On-site installer	1-2?
TOTAL	**≈ 45**

Tab. 3: Estimation of knowledge profiles of personnel needed for the Stuxnet development, deployment and operational management

Note that the above estimation includes not only technical human resources needed for designing, developing and testing the malware, but also agents needed to gather information and eventually for the local deployment of the cyber weapon.

[28] Interview of John Bumgarner (chief technology officer for the U.S. Cyber Consequences Unit) to Reuters (http://www.reuters.com/article/2011/12/08/us-cybersecurity-iran-idUSTRE7B10AP20111208).
[29] http://www.theatlantic.com/technology/archive/2010/11/the-stuxnet-worm-more-than-30-people-built-it/66156/ (4 Nov, 2010).
[30] http://www.guardian.co.uk/politics/2011/may /30/stuxnet-attack-cyber-war-iran (30 May, 2011).
[31] Frank Rieger (Chief Technology Officer at GSMK), September 2010 (http://www.bloomberg.com/news/2010-09-24/stuxnet-computer-worm-may-be-aimed-at-iran-nuclear-sites-researcher-says.html).

The overall number of human resources given in our estimations is also somewhat compatible with Kaspersky's estimation given in December 2011 (about 30 persons),[32] which does not include the intelligence operators.

The mistakes of Stuxnet

Apparently, perfection does not belong to the world of software development – either malware or legitimate application. Bugs are always ready (and very likely) to enter any programme without invitation. Stuxnet, being fundamentally a piece of software, is no exception to this unwritten but widely known rule.
In spite of its undoubted complexity and the expertise of its creators, Stuxnet seems, in fact, to have some faults in its code.
These findings have induced some experts, after the initial clamour, to reconsider their view of Stuxnet as a masterpiece of computer programming.
According to Tom Parker, a security consultant who spoke about the worm at the 2011 Black Hat DC conference,[33] the developers of Stuxnet made many mistakes, resulting in a code of not such a high quality. He mentioned that the implementation of the command-and-control mechanism was trivial, and that it was also unlikely that the creators wanted the massive propagation over the Internet, even doubting that a western state was responsible for developing it.
Parker said, however, that the worm was very effective on a number of levels and it was highly unlikely that only one person developed the worm on their own. He hypothesised, therefore, that two separate groups could have developed Stuxnet, namely an "elite group" of talented programmers to produce the code and exploits, and a less technically qualified group to adapt the code for its final use.

Regardless of speculation, it is undeniable that there are some evident incongruences in Stuxnet. For instance, cryptography has been implemented in a strange manner: on one hand, Stuxnet uses naïve XOR encryption for the outbound data (from the infected machine to the Command & Control server) communication protocol; on the other, it uses FIPS 140-2-compliant encryption for the P2P networking (code updates). This is another clue that could be taken as an indication that different functionalities of the malware were written by different authors, maybe even by different development teams that worked in parallel to maximise the time exploitation, as posited by many.
Moreover, Stuxnet also contains a macroscopic bug in its code related to the exploitation of the Windows Print Spooler Vulnerability: the abnormal inflation of the `winsta.exe`[34] executable file, which is also a very evident symptom useful to quickly and simply diagnose its presence in a system[35]. This bug is serious because it causes effects which are very noticeable[36], thus potentially nullifying all the advanced stealth techniques used by the malware to hide itself.
Another remarkable bug causing "Blue Screens of Death" (BSODs) and system reboots on infected systems was related to `MrxCls.sys`. The bug was due to the fact that in some cases Windows was starting `MrxCls.sys` *before* `HAL.DLL` (the Hardware Abstraction Layer component) was completely loaded, causing the operating system to crash since device drivers require the HAL to be active in order to operate. Normally the HAL always starts loading before device drivers but, for unknown reasons, there were some Windows machines in Iran that were particularly slow in loading the HAL, and didn't finish loading before the Stuxnet device driver started its execution. This resulted in an endless crash-and-reboot sequence, and is the symptom that pushed the Iranians to ask for the help of VirusBlokAda

[32] http://www.securelist.com/en/images/pictures/klblog/208193184.png.

[33] https://threatpost.com/en_us/blogs/stuxnet-authors-made-several-basic-errors-011811.

[34] The name given to the file is aimed to make out that it is the WinStation Monitor utility (~40 KB), which is a tool included with the Microsoft Windows 2000 Server Resource Kit, used to monitor Terminal Services sessions. Usually the file winsta.exe is not included in the standard Windows installations and is not likely to be installed in %Windir%\system32\ but rather in C:\Program Files (x86)\Resource Kit\.

[35] First cases of this issue were reported in June 2010. Microsoft Security Essential was the first anti-malware tool that was able to discover Stuxnet, just by carrying out a simple check on %Windir%\system32\winsta.exe.

[36] Infected PCs slow down because the winsta.exe file gets bigger and decreases the available hard disk space, practically until complete depletion. The file size has been reported in some cases to reach up to 200 GB. In such a situation, Windows will alert the user of the low disk space available.

(see → *Infection history*). For the record, this problem was corrected in Duqu (see → *Duqu, the new Stuxnet?*).

Basically, the weaknesses discovered in Stuxnet were the following:
- weak code obfuscation (UPX packing + some XOR)
- weak encryption of the C&C protocol
- poor spread control over the Internet (excessive propagation)
- `winsta.exe` bug
- missing code portion for attacking Siemens PLCs equipped with CPU S7-415 (the so-called attack sequence "C")
- BSODs at system startup

The reasons could have been:
- lack of time
- lack of expertise
- intentional non-application of features that could have triggered anti-malware tools (encryption and obfuscation)

All the shortfalls and design faults discovered in Stuxnet can possibly be explained by the fact that the Stuxnet operation was given less time than planned.
According to the New York Times,[37] in January 2009 US President George Bush authorised a covert operation – of which Stuxnet is believed to have been a part – to sabotage the electrical and computer systems around Natanz. When President Obama took office and was informed about the programme, he wanted to quicken its execution. This sudden acceleration could have forced the malware writers to give up on test and quality control of the developed code.
Certainly, time was a very critical factor in the Stuxnet operation; a constraint that can also be confirmed by the fact that a version not exploiting the LNK vulnerability was released in an early status of the threat history. This could have meant that the Stuxnet authors were somewhat in an hurry, and decided to spread the worm even without a zero-day vulnerability capable of infecting air-gapped systems.

The lack of expertise is an unlikely theory. Stuxnet shows so many sophisticated features of stunning elegance – led by its multiple zero-day exploits – that it is rather difficult to believe that its authors were not that expert. A more plausible speculation would be the supposition that two or more different groups worked separately on developing different parts of the malware and had poor (or no) inter-communication, as already touched on.

The intentional non-application of features is a valid point that was also supported by Mikko Hypponen, chief research officer at F-Secure, who hypothesised that Stuxnet's authors may not have added encryption and anti-debugging features "*because they wanted to make the program look as 'normal' as possible*", referring to the fact that most antivirus labs use automated heuristic analysis to find 'suspicious' samples. By not adding encryption, "*Stuxnet didn't look suspicious. It looked like an automation toolkit that would install signed device drivers*."
However, let us be honest: was obfuscation and encryption such a high priority in the Stuxnet implementation, and really so necessary? "*If you were an administrator, would you question a file in the Windows\System32 folder named MrxNet.sys, written by RealTek and verified with a legitimate certificate?*"[38]

[37] Newspaper article: "*U.S. Rejected Aid for Israeli Raid on Iranian Nuclear Site*", by David E. Sanger, published January 10, 2009 (http://www.nytimes.com/2009/01/11/washington/11iran.htm).

[38] http://www.darkreading.com/advanced-threats/167901091/security/attacks-breaches/229500805/stuxnet-how-it-happened-and-how-your-enterprise-can-avoid-similar-attacks.html.

Goals achieved by Stuxnet

Despite the initial belief that the Russian-built nuclear power plant (steam turbine) in Bushehr, 745 miles south of Tehran, was the most probable target of Stuxnet, evidence soon clearly showed that the real target was the uranium enrichment facility (gas centrifuges) in Natanz.[39]

Stuxnet was not designed to damage the Windows computers it infects (hardware, software, or data), nor the Siemens PLCs, nor the frequency converters controlled by the PLCs. Its final targets are the IR-1 centrifuges used to enrich uranium, which are spun by these frequency converters. However, there are controversial opinions about how effective the worm was in accomplishing its goal.

According to ISIS,[40] "*between 2009 and 2010, Iran decommissioned and replaced about 1000 IR-1 centrifuges in the Fuel Enrichment Plant (FEP) at Natanz, implying that these centrifuges broke. This level of breakage exceeded expectations and occurred during an extended period of relatively poor centrifuge performance.*
The number of IR-1 centrifuges installed at the FEP climbed steadily to a peak of almost 9,000 in November 2009 before falling in late 2009 or early 2010 ostensibly due to the effects of the Stuxnet malware. Currently, the number of installed centrifuges has leveled off at around 8,000."

Fig. 6: An IR-1 centrifuge in Natanz. It is essentially a replica of a P-1 Pakistani centrifuge. Consisting of a two metre-long aluminum rotor, it can rotate at peripheral speeds of up to 350 metres per second, equivalent to 1260 km/h (close to the speed of sound).

According to Ralph Langner, a German security expert who has been particularly active in analysing Stuxnet, by acting on the centrifuges the worm has set back the Iranian nuclear programme by two years. Interviewed by the Israeli newspaper *The Jerusalem Post*, the researcher said that "*it will take two years for Iran to get back on track. This was nearly as effective as a military strike, but even better since there are no fatalities and no full-blown war. From a military perspective, this was a huge success.*"[41]

Langner's and ISIS estimates, however, were recently downsized substantially by FAS[42] and the IAEA.
In an interview[43] released to the Washington Post on February 14 2011, Yukiya Amano, Director General of the International Atomic Energy Agency, said that the Stuxnet cyber attack on Iran seemed to have had no evident effect on its nuclear programme. Amano said that "*Iran is somehow producing uranium enriched to 3.5 percent and 20 percent […] steadily*" and "*constantly*".
Later, in a confidential report[44] released to the United Nations in November 2011, IAEA reported that the Iranian nuclear programme indeed reached a very advanced state and that Iran "*has carried out activities relevant to the development of a nuclear explosive device*".
Consequently, it has been supposed that the information about 1000 damaged gas centrifuges in Natanz could have been intentionally spread by Iran as disinformation to

---

39 First hypothesised by Frank Rieger on 22 Sep, 2010, in newspaper article "*Trojaner „stuxnet" Der digitale Erstschlag ist erfolgt*", Frankfurter Allgemeine Zeitung (http://www.faz.net/aktuell/feuilleton/debatten/digitales-denken/trojaner-stuxnet-der-digitale-erstschlag-ist-erfolgt-1578889.html).

40 ISIS report, 22 Dec, 2010 (http://www.isis-online.org/isis-reports/detail/did-stuxnet-take-out-1000-centrifuges-at-the-natanz-enrichment-plant/).

41 http://www.jpost.com/IranianThreat/News/Article.aspx?id=199475.

42 "*Using Enrichment Capacity to Estimate Iran's Breakout Potential*", Federation of American Scientists issue brief, January 2011 (http://www.fas.org/pubs/_docs/IssueBrief_Jan2011_Iran.pdf).

43 http://www.iaea.org/newscenter/transcripts/2011/wp140211.html.

44 GOV/2011/65 "*Implementation of the NPT safeguards agreement and relevant provisions of United Nations Security Council resolutions in the Islamic Republic of Iran*" (http://isis-online.org/uploads/isis-reports/documents/IAEA_Iran_8Nov2011.pdf).

make believe that their nuclear programme was effectively delayed by Stuxnet.
There are many possible explanations as to why the Stuxnet worm was not the definitive solution to stop Iran's uranium enrichment process.
The first one – as has been already said in this document – is that Stuxnet was released in too big a hurry. It was able to attack only the centrifuges connected to PLCs equipped with CPU S7-315, because the code for attack sequence against the S7-417 CPU was not developed in time.
The second one is that the Iranians were able to recover from the Stuxnet attacks and clean the infected computers and PLCs more quickly than expected. This would mean that Iran has cyber defence capabilities that were until now greatly underestimated, a suspicion that could be fed by the well-known case of the Comodo hack which occurred in March 2011 and more recently by the capture of the American RQ-170 Sentinel Unmanned Aerial Vehicle (UAV), which occurred over Iran in December 2011[45].

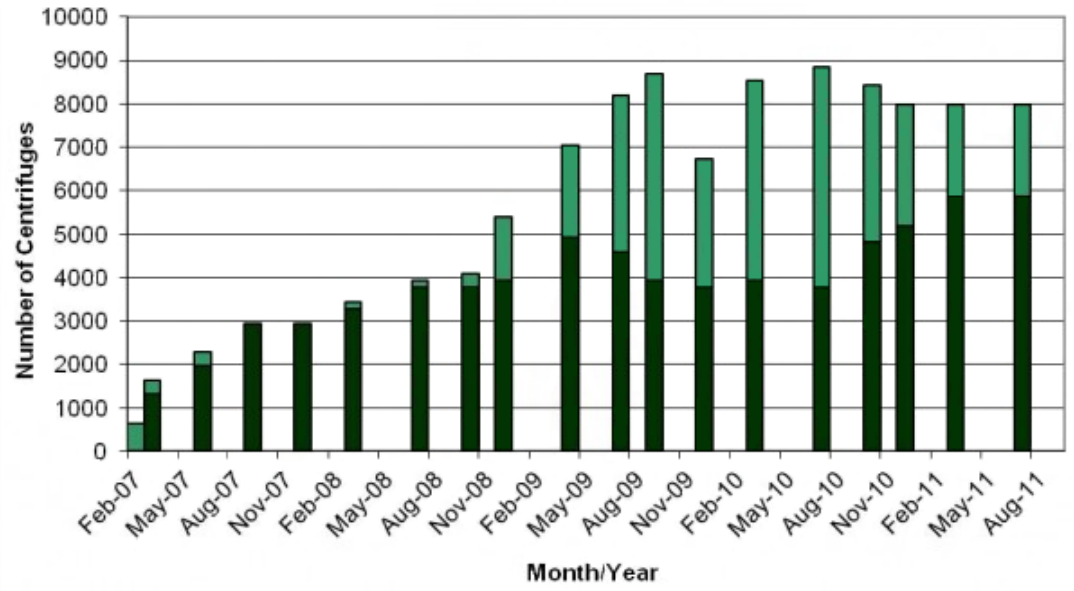

Fig. 7: Number of centrifuges working in Natanz from February 2007 to August 2011.The dark green bars indicate the number of centrifuges processing the uranium hexafluoride (UF_6), while the light green bars indicate the overall number of installed centrifuges. The decrease noticeable between November 2009 and February 2010 is allegedly the result of the sabotage performed by Stuxnet.

[45] The capture could have been achieved by jamming radio communications and/or sending fake GPS data to the UAV.

Origins of Stuxnet

The reason for the existence of Stuxnet is quite clear and supported by effective reverse-engineering analysis: Stuxnet is a cyber weapon built to sabotage the uranium enrichment centrifuges.
Unfortunately, the same cannot be said for the creators, about which only prudent speculations can be hypothesised.

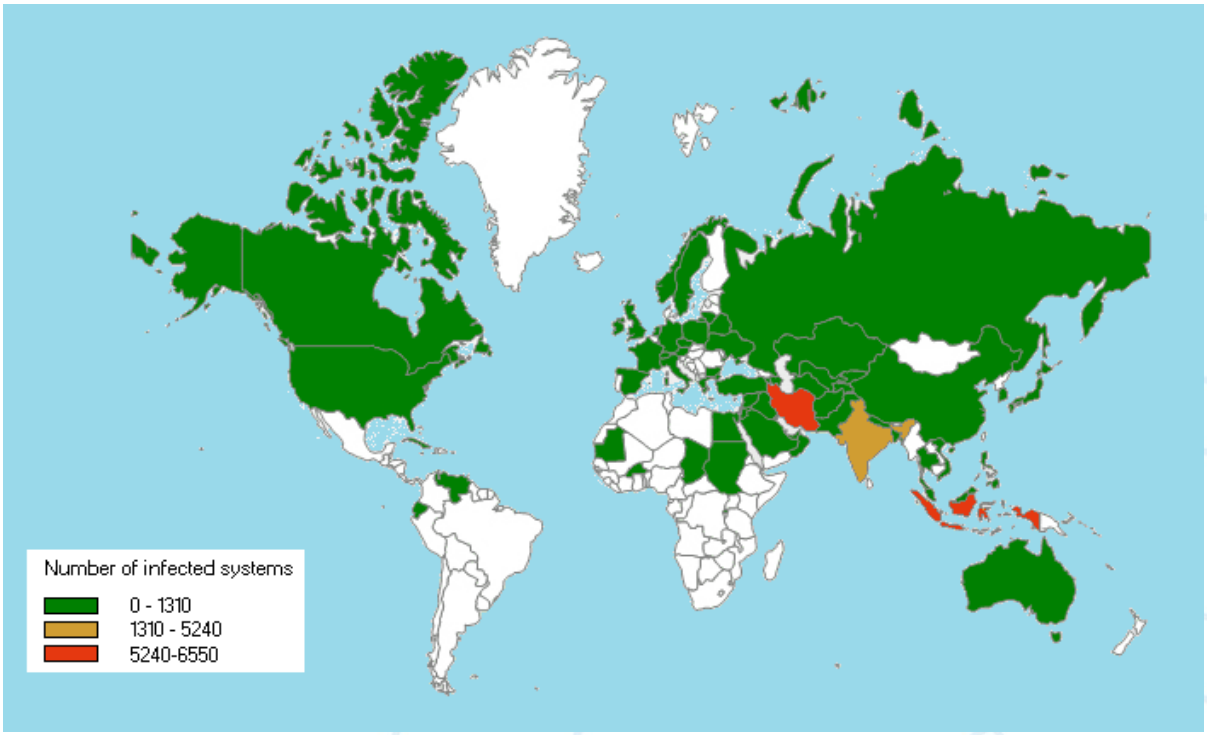

Fig. 8: Geographic distribution of the Stuxnet infection (October 2010). Areas in white indicate countries for which no official data are available. The most hit countries are Iran, Indonesia and India.

Indications in Stuxnet's code that could suggest its authorship are few and rather cryptic. There are some text strings in the code that contain the word "Myrtus",[46] and there is a value written in the Windows registry by the worm which seems to be a date ("19790509"), which some link to May 9 1979, the date of the execution of Jewish Iranian businessman and philanthropist Habib Elghanian.[47]
However, to be absolutely objective, there is nothing in the code that could be interpreted as definitive evidence pointing to any individual or organisation. It is also possible that those traces were expressly left in Stuxnet's code with the intentional purpose of deception.
What can be asserted for sure is that whoever was behind Stuxnet had a very deep knowledge of SCADA systems, and particularly of the Simatic PLCs and STEP 7 environment.
In addition, the typical goals which cybercriminals or hackers attempt to reach – such as spam, DDoS, botnet installations, personal data exfiltration, identity theft – are not the goal of Stuxnet, making it once more substantially different to other malware and highly unusual from a behavioural analysis perspective.
In addition, using so many unpatched vulnerabilities in just one malware is unheard of outside Stuxnet, again suggesting that whoever wrote this malware was someone who had a much more sophisticated intention than the typical cyber-criminals. Cyber-criminals/terrorists or hackers would not waste their time in looking for *four* different zero-day vulnerabilities; rather they would have exploited the first found vulnerability as soon as possible.

[46] The full string is: "`b:\myrtus\src\objfre_w2k_x86\i386\guava.pdb`".
[47] Habib Elghanian was a Jewish self-made millionaire living in Iran and acted as the symbolic head of the Iranian Jewish community in the 1970s. His death, decided by an Islamic revolutionary tribunal after the fall of the Shah, is considered the first execution made by the Iranian Islamic government against a Jew.

As the infamous post-commentator *n3td3v*[48] correctly observed about Stuxnet, "*This proves there is no cyber terrorism threat* [behind Stuxnet]*, if there was the US would be shut down by now. It would take two minutes with this exploit to shut down the entire US's critical infrastructure.*"[49]
What is clear beyond any speculation or skepticism is that Stuxnet was altogether a large and expensive operation, with no obvious financial return. The enormous knowledge and workload pulled into the worm's code to successfully attack the Iranian PLCs suggest that whoever developed Stuxnet probably had the same types of software and hardware (PLCs, frequency converters, centrifuges), which are very expensive, on which to run tests. Only a nation state, or maybe a very few private organisations, can afford this.
The abovementioned considerations suggest that the malware was developed with either the direct involvement or support of intelligence agencies or nation-states and designed for sabotage. Due to its complexity and the related need for major financial resources, many security experts stated that it could have only been created by a hostile government.[50]

US-Israel joint venture

The most popular theory is that Stuxnet could have been developed under a joint effort between the USA and Israel. This speculation is rather easy to understand, considering the following factors:

- Iran was the country most hit by the malware;
- Israel felt very seriously threatened by the development of the Iranian nuclear programme;
- In 2007, Israel bombed a secret Syrian nuclear reactor in a desert area in the east of the country called al-Kibar, in the Deir Al-Zur region (apparently built with North Korea's support), essentially for the same reasons explained above;
- The USA and Israel have a very strong cyber warfare capability (respectively the first and fourth in the world, according to McAfee projections).

According to the New York Times,[51] Stuxnet could have been developed within a joint US-Israeli operation, and could have been tested in the secret nuclear research center of Dimona, in the Israeli Negev desert. *"The Dimona complex […], the heavily guarded heart of Israel's never-acknowledged nuclear arms program, became a critical testing ground in a joint American and Israeli effort to undermine Iran's efforts to make a bomb of its own".* It is believed that in Dimona "*Israel spun nuclear centrifuges virtually identical to Iran's at Natanz, where Iranian scientists are struggling to enrich uranium*", to test and hone the effectiveness of the computer worm.

Fig. 9: The Negev Nuclear Research Center, a.k.a. the "Dimona complex". It is located in the Negev desert, about 13 km south-east of the city of Dimona.

Dimona is truly the ideal site for secretly developing a sophisticated cyber weapon. It is heavily guarded, is approachable only by

[48] n3td3v is a very well known Internet post-commentator, particularly active on security-related posts and mailing lists. Due to his way to interact with other users, he was often defined as a "troll" and also banned as such from discussions.

[49] First posted on July 20, 2010 1:49 PM PDT at http://news.cnet.com/8301-27080_3-20011159-245.html. Also available at http://www.1-script.com/forums/computer-security/the-taliban-al-qaeda-aren-t-interested-in-cyber-terrorism-13191-.htm.

[50] Eugene Kaspersky, co-founder of Kaspersky lab, Inc.: "*According to information from the code, we understand this is high-end malware. To develop such malware needs a million dollar budget. I'm afraid it's quite obvious that this malware is not done by ordinary cyber-criminals*." (Interview to *Computerworld*, Melbourne, 22 March, 2011).

[51] Newspaper article: "*Israeli Test on Worm Called Crucial in Iran Nuclear Delay*", by William J. Broad, John Markoff and David E. Sanger (published January 15, 2011; http://nyti.ms/esygjV).

staff, far away from curious eyes and it has basically all the facilities necessary to achieve full uranium processing: an enrichement plant, a fuel fabrication area and a small nuclear reactor. It is generally believed to be the area where Israel could have unofficially developed its nuclear weapons.
Interestingly, both the USA and Israel have never denied the claims that they were involved with Stuxnet's development. In May 2011, the US Deputy Defense Secretary William Lynn, when directly asked by a CNBC journalist[52] if the "*US was in any way involved in the development of Stuxnet*", tried to dodge the question and only after being pressed finally answered that "*this is not something that we're going to be able to answer at this point.*"

A possible framework

N.B. the views expressed in the following section are purely theoretical speculation and do not necessarily reflect in any way the official policy or position of the author, CCDCOE, NATO, NATO bodies or any NATO country.

Many speculations were made about the authors of Stuxnet. According to expert opinion, the most followed theory is that Stuxnet could have been developed through a joint effort between the USA, Israel and Germany .
The USA put on the table IT and nuclear power production experts, Israel put on-site intelligence operatives (for information gathering and infiltration ops) and the skills of its famous secret cyberwar division Unit 8200, and Germany (or – more likely – Siemens) put the knowledge of the Simatic PLCs architecture. The result was an ultra-technical joint task force of hackers, provided with a superbly equipped lab in which the Iranian industrial systems were carefully reproduced and on which they were able to test the best ways and configurations to deliver an incredibly efficient cyber weapon. For sure they had same PLCs, PGs, SCADA software and also several enrichment centrifuges owned by the Iranians.
It is worth remembering that Unit 8200 was allegedly responsible in 2007 for shutting down the Syrian air defence radars just minutes before Israeli aircraft were able to bomb the Al-Kibar syrian nuclear reactor (Operation "Orchard"). It is also worth remembering that in 2010 it was funded by the Israeli government with a large amount of money (maybe for the excellent return).
Unit 8200[53] (located in Har Avital, Golan Heights) is the largest unit in the Israeli Defence Forces and is comparable, in skills and competence, to the American NSA, except that it is a fully military, top-secret organisation, led by a brigadier general whose identity remains classified.
*Several signs point to confirmation of this theory. The first and most significant is that – according to the NewYork Times which cites unidentified intelligence and military experts – officials from Israel broke "*into wide smiles when asked whether Israel was behind the attack, or knew who was.*"[54]*
*Moreover, in early 2008 "*the German company Siemens cooperated with the Idaho National Laboratory (INL), in the United States, to identify the vulnerabilities of computer controllers that the company sells to operate industrial machinery around the world – and that American intelligence agencies have identified as key equipment in Iran's enrichment facilities. Siemens said that this was part of routine efforts to secure its products against cyberattacks. Nonetheless, it gave the Idaho National Laboratory – which is part of the Energy Department, responsible for America's nuclear arms – the chance to identify vulnerabilities in the Siemens systems that were later exploited by Stuxnet."
The Siemens involvement seems to also be believed by the Iranian regime.[55]

52 Correspondent Melissa Lee in the documentary "*CodeWars: America's Cyber Threat*" (http://www.youtube.com/watch?v=_9Gt2Ek4inM).

53 For more information: http://www.upi.com/Top_News/Special/2011/05/11/Enter-Unit-8200-Israel-arms-for-cyberwar/UPI-93881305142086/.

54 Newspaper article: "*Israeli Test on Worm Called Crucial in Iran Nuclear Delay*", by William J. Broad, John Markoff and David E. Sanger (published January 15, 2011; http://nyti.ms/esygjV).

55 http://www.guardian.co.uk/world/2011/apr/17/iran-siemens-stuxnet-cyber-attack.

Finally, we would like to report some words from Ralph Langner, probably one of the three[56] most knowledgeable security scientists in the world, about Stuxnet:
*"*Stuxnet required an extreme amount of intelligence about the Natanz plant layout, a full understanding of the IR-1 operation (presumably with a mockup test system available), and an extreme amount of insider knowledge of the Siemens products involved. This limits the search for the originators to very few organizations in the world.*"*

A further credibility to these claims comes from John Bumgarner, Research Director for Security Technology at the United States Cyber Consequences Unit (US-CCU), in an article[57] published in May 2010 (prior to Stuxnet's discovery). He suggested that a cyber attack against uranium centrifuges of those countries violating international treaties on nuclear weapons, by manipulating their rotational speed, would be even more effective than conventional strikes (given that they are contained in heavily hardened structures) and significantly disrupt their nuclear programme.

Why not the Russians?

The Russian origin of Stuxnet might not be so unlikely. In Iran, in the immediate aftermath of the malware's discovery, there were rumours that many interrogations were made of Russian technicians about their possible involvement.[58] Having worked on the completion of the Bushehr nuclear power plant, the Russians know in detail the innermost secrets of the site. For them, connecting USB sticks to SCADA workstations is an operation that they could have done hundreds of times without generating any suspicion.[59]
But why hit a customer with such an attack? There are many reasons, the first of which would be to prevent an Israeli conventional military attack too close to the Russian border. Moreover, they could also fear that a sabotage leading to a nuclear accident would basically stop the Russian Federation from exploiting the promising nuclear market in the Middle East. Right now, with Russia being the only supplier of nuclear fuel for Iran, the Iranian problems of producing enriched uranium by themselves would turn economically beneficial for the Russians. In addition, Russia has been repeatedly indicated as a country having one of the best cyber warfare capabilities in the world,[60] not to mention being behind the cyber attacks released against Georgia in 2008.
Last, but not least, the political faction close to President Medvedev is pushing for a slowdown in relations with Tehran and sees a nuclear-empowered Iran as a potential rival to its southern borders.

Fig. 10: Russian contractors in the Bushehr nuclear power plant control room, setting up the SCADA systems.

Why not the Chinese?

Even the Chinese could have done a cost / benefit analysis, after which they could have concluded that

[56] Along with Lian O'Morchu (Symantec's Supervisor of NAM Security Response Operations) and Eric Chien (Symantec's Security Response technical director).

[57] http://www.crows.org/images/stories/pdf/IOI/IO%20Journal_Vol2Iss2_0210.pdf.

[58] http://temi.repubblica.it/limes/iran-sotto-attacco-le-nuove-frontiere-della-cyberwar/15341.

[59] "*Was Russia behind Stuxnet?*" by Panayotis A. Yannakogeorgos (http://the-diplomat.com/2011/12/10/was-russia-behind-stuxnet/2/).

[60] In its "*Virtual Criminology Report 2009 - Virtually Here: The Age of Cyber Warfare*", McAfee indicated Russia as one of the top five "countries developing advanced offensive cyber capabilities", along with USA, China, France and Israel (http://www.mcafee.com/us/resources/reports/rp-virtual-criminology-report-2009.pdf).

a hypothetical Israeli military attack against Iran could be probable. Such a case would, as well as compromising the huge Chinese investment made in the Middle-East, have caused a strong brake on their economic growth because of the subsequent chaos. Alternatively, Beijing could have been concerned about the Iranian nuclear proliferation and decided to stop it, while showing, however, a reverent attitude towards its third largest oil supplier.
Another theory suggests that Stuxnet may be a weapon of China against India to stop or slow down their race to the moon. China and India are competing with each other to see who will be the first to reach the moon. The Chinese government said they could land an astronaut on it in 2025, while India claimed they could do the same in 2020. This theory could be somehow confirmed by the fact that India has been one of the countries most hit by Stuxnet after Iran. On the other hand, in China the Xinhua state news agency and other media reported – probably exaggerating – that more than six million personal computers and 1000 industrial facilities were infected by Stuxnet.[61]
China could also have used its agreement with Microsoft,[62] which allows the Chinese government to examine the Windows source code, in order to discover more easily zero-day vulnerabilities.
The Chinese responsibility was strongly supported by Jeffrey Carr,[63] who repeatedly in different articles and in his paper "*Dragons, Tigers, Pearls, and Yellowcake: Four Stuxnet Targeting Scenarios*" depicted China as the most likely author of the worm. Carr's theory is that China created Stuxnet to target countries with large amounts of natural resources such as copper, iron, and gold, which are especially valuable to China's high economic growth.
It is interesting to note that there have been multiple episodes of cyber attacks – most of which were attributed to China[64] – against mining companies (especially Australian companies, such as BHP Billiton, Rio Tinto and Fortescue Metal) that could strengthen the validity of this theory. However, their timelines do not match Stuxnet's chronology, and they seem to have had a goal of cyber espionage rather than cyber sabotage. The attacks were conducted between 2010 and 2011, according to the Australian press.[65]

Other possibilities

For the sake of completeness, there are also other theories regarding Stuxnet's origins that, although less plausible, deserve to be cited. They are mostly reported by commentators who are sceptical about the USA and/or Israeli responsibility.

One of these theories hypothesises that there is a small chance that Stuxnet originated in Taiwan or that it has a Taiwan connection, given that the Verisign certificate for the `~WTR4141.tmp` file was obtained with the private key stolen from Realtek Semiconductor. Another version of the worm comes with a second certificate issued with another key stolen from the JMicron Technology Corporation. Both of these companies have headquarters in the Hsuchin Science Park office park in Taiwan.

Germany has also been mentioned as a possible creator of Stuxnet, mainly because of the deep expertise shown by the author(s) of the worm in dealing with industrial systems built by Siemens – which is a German company – but the evidence for this theory is rather circumstantial.
Being a German industrial giant, Siemens AG would definitely have the capability of independently writing the malware. It is the largest industrial conglomerate in Europe, with a reported global revenue of more than 76 billion euros for the year of 2009. The galaxy of Siemens subsidiaries ranges from software development, to telephony, to large industrial plants.

---

61 Source: AFP (http://www.google.com/hostednews/afp/article/ALeqM5iFRHUmI2w6HaAFZq-wUNre813wcA?docId=CNG.f6fba55ad8f5e329c0c25bad9aa7b8d3.651).

62 http://www.informationweek.com/news/software/operating_systems/225400063.

63 "[...] *based solely on the known facts, I consider China to be the most likely candidate for Stuxnet's origin.*" Blog article: "*Stuxnet's Finnish-Chinese Connection*" (14 Dec 2010, The Firewall - the world of security; http://www.forbes.com/sites/firewall/2010/12/14/stuxnets-finnish-chinese-connection/).

64 http://www.news.com.au/breaking-news/chinese-cyber-attacks-on-bhp-billiton-rio-tinto-and-fortescue-metals-group/story-e6frfku0-1225855710032.

65 http://www.theaustralian.com.au/business/mining-energy/miners-under-cyber-attack-from-everywhere/story-e6frg9df-1226065199596.

Siemens could possibly have acted independently to undermine Iran's nuclear programme, having "means, motive, and opportunity". The means were the technical knowledge that only a manufacturer can have about its own devices, especially in conjunction with the not insignificant detail that Siemens has been building power plants for over thirty years (Siemens AG was initially entrusted with the construction of the Busher nuclear plant; see: → *Geo-political considerations*).
The motive may have been a desire to distance its name from the production of the Iranian regime's WMDs (image damage), plus the fact that Iran is clandestinely using Siemens products (economic damage), because any technological material (including software) coming from western countries is placed under embargo according to UN resolution 1737 (December 27, 2006) and subsequent resolutions. And the opportunity? There have been plenty of them, considering that the license keys for the Simatic Manager engineering software are shipped by Siemens to its customers on USB sticks,[66] which the customer has to plug into the machine in order to begin the installation of the software.
Iran has publicly accused Germany of providing technical advice to Israel and USA for the sabotage of the ICSs serving its uranium enrichment plant, helping them launch the Stuxnet cyber attack.[67] Finally, it is worth noting that while it was first discovered on an Iranian computer,[68] the first real infection of a plant was found to be in Germany.

The Williams analysis

Believe it or not, almost everything that has happened with Stuxnet was clearly spelled out one year before its first appearance on the world scene.
In an analysis[69] written in July 2009, Dan Williams, a Reuters correspondant in Jerusalem, reported that Scott Borg, director of the US Cyber Consequences Unit, said that "*Israel can definitely be assumed to have advanced cyber-attack capabilities*", and that if you were to imagine how Israel could use its cyber warfare knowkledge against Iran, "… *malware could be inserted to corrupt, commandeer or crash the controls of sensitive sites like uranium enrichment plants.*"
Continuing further, Borg said that "*as Iran's nuclear assets would probably be isolated from outside computers, hackers would be unable to access them directly. Israeli agents would have to conceal the malware in software used by the Iranians or discreetly plant it on portable hardware brought in, unknowingly, by technicians. A contaminated USB stick would be enough*."
The description of the attack modalities against Iran hypothesised by Borg is so similar to Stuxnet's operative effect and methods that the Williams article seems to be more of a report than a forecast.
But that is not all. In his analysis Williams also writes of an Iranian businessman, namely Ali Ashtari, who was executed by the Iranian regime as an Israeli spy for having sold to it "counterfeited" telecommunications equipments for a secret military project. This case is unbelieveably similar to that of Habib Elghanian, the Iranian businessman who was also executed by Iran as a Jewish spy, and whose date of birth written in "sortable" format YYYYMMDD (19790509) is the exception marker used by Stuxnet to self-terminate if it is found in the Windows registry.
As a last consideration, it is also very interesting to note that both Scott Borg and John Bumgarner, the cyber security expert previously mentioned who also suggested the possibility of using cyber attacks to disrupt uranium enrichment centrifuges, work for the same company (US-CCU).

[66] R.Langner, "Intercept, Infect, Infiltrate" (http://www.langner.com/en/2011/02/22/intercept-infect-infiltrate/).
[67] http://www.reuters.com/article/2011/04/17/iran-nuclear-stuxnet-idAFPOM73176820110417.
[68] See paragraph: →"*Infection history*", page 9.
[69] http://www.reuters.com/article/2009/07/07/us-israel-iran-cyberwar-analysis-idUSTRE5663EC20090707.

Prevention, mitigation and counter-measures

Removing the worm from an infected control system is not that difficult but can be a rather long process because of the extent of the infection, since Stuxnet tries to spread to other computers over local area networks. Siemens has provided its customers with an effective removal tool, developed by Trend Micro, along with detailed instructions on how to proceed.[70] Microsoft's MSRT (Malicious Software Removal Tool)[71] is also able to remove it. The removal procedures must not be limited to Windows machines, but also need to encompass the Simatic PLCs, as the modified code will remain in their memory.

For network administrators who need to remotely perform the detection of the malware on their networks, free network tools are also available. These tools work by sending spoofed packets similar to those sent by known Stuxnet variants. Since any infected host will respond to this spoofed packet, network administrators can easily identify which machines are infected within the network.

Nevertheless, it is generally agreed that the most effective way to prevent the worm from spreading is to make use of zone-based defences, as described in the ANSI/ISA-99.02.01 and IEC-62443 standards. The concept is as old as firewalls, and is applied by breaking up the network into security zones.[72] Between the zones, industrial firewalls are installed with rules that block the protocols that Stuxnet uses for infection and communications (HTTP, RPC, and MSSQL). This way, if a Stuxnet infection occurs, it should be limited to a small number of machines in a single zone.

Unfortunately, it seems that Stuxnet has carefully chosen for its communication needs the same protocols used by Simatic Windows software, so merely blocking them will stop the communication between those components if they are located across segmented zones. Consequently, a better solution would be to rely on personal firewalls (possibly from third-parties rather than Windows firewalls) installed on each Windows machine that would authorise only selected processes (belonging to SCADA applications) for sending out the indicated protocols.

Protocol	Port number	Common application in SCADA / ICS	Used by Stuxnet for
HTTP	TCP 80	HMI Web Clients Historian Web Clients	Connection to Internet Command & Control Server
RPC-DCOM	TCP 135 Random TCP ports between 1024-65535	OPC Classic, Certificate Services, Group Policy	Worm P2P Upgrade System
RPC-SMB	TCP 139 TCP 445 UDP 139 UDP 445	File and Print Sharing, Event Log, Netlogon, WinCC Web Nav	Open Shares File Share Exploit Print Spooler Exploit
MSSQL	TCP 1443	WinCC Client-Server Interaction	WinCC SQL Server Infections

Tab. 4: Network protocols used by SCADA/ICS processes and by Stuxnet

Other, more in-depth, actions that can be taken are the following:

1) DO NOT ALLOW ANY INTERNET CONNECTIVITY IN THE PROCESS CONTROL NETWORK.
2) Disable removable storage support (USB/SD card/floppies), or limit usage to specific vendor devices (PnP Vendor ID / PnP_Device_ID filtering).
3) If the PLCs are controlling a particularly sensitive automation process, isolate SCADA/HMI/PG workstations from any TCP/IP network (disable or remove network card/WiFi/BT) and allow communications only through industrial interfaces (ModBus, CanBus, Profibus, RS-485, etc.)
4) Do not allow any new or external computer not previously checked to connect to the process control network. Use NAC (802.1X / RADIUS and/or MAC address filtering) solutions to rule the access.

[70] http://support.automation.siemens.com/WW/llisapi.dll/43876783?func=ll&objId=43876783.

[71] Starting from version released in August, 2010.

[72] As also suggested by Siemens in the paper "Security concept - PCS 7 and WinCC - Basic document" (Doc. No. 04/2008 A5E02128732-01).

5) Do not use locally-attached shared printers. Instead, use network printers (i.e. printers with a network card, not Windows-based!) and TCP/IP printing protocols (RAW/LPR on port 9100).
6) If an installation or a file copy is needed, perform the following actions in COMMAND LINE interface or text-based interface (no WIMPs): FTP from a local, authenticated distribution server ONLY, in unexecutable format (i.e. password-protected zip files) with random numeric extension (12 digits), cmdline extract, SHA-1/MD5 verify, AV scan, install. All FTP-ed modules should came only from certified vendors/ manufacturers which should always give file fingerprint data and a digital certificate over a secure channel. Otherwise, files should be carefully checked in a sandbox environment and fingerprinted internally.
7) Have full disk images of the control workstations stored in a secure storage environment and ready for quick re-installation. Re-install regularly (e.g. once a month) and reset PLCs.
8) Configure Windows machines for a whitelist of permitted executable processes (e.g. using AppLocker or applying a Software Restriction Policy through Group Policy editor).
9) Organisations using the Simatic architecture of PLCs must be cautious of project files obtained from external sources, the most likely source of infection being trusted parties whose systems have been compromised by the worm.
10) Fake systems (PG/HMI stations and PLCs) acting as basic honeypots would also help in detecting suspicious activity.
11) Install industrial IDS that perform deep packet inspection of the Siemens S7 protocol to monitor network traffic to and from S7 PLCs, as also suggested by ICS-CERT.

Neutralising factors

Stuxnet is a Windows malware, so it cannot infect machines running a non-Windows operating system (i.e. Linux or QNX).
In a Windows environment, due to the conditions checked by Stuxnet during its execution, any one of the following situations will cause the malware to terminate itself:

- system date set to a value greater than (after) 24-June-2012;
- machine running any Windows 64-bit operating system;
- presence in the registry of the key "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\MS-DOS Emulation" with a string value named "`NTVDM TRACE`", containing the value "`19790509`" (which is believed to be an infection marker).

Moreover, a fully patched Windows machine is immune to Stuxnet, given that Microsoft fixed all of the vulnerabilities exploited by the malware on February 2011.

Misconceptions and errors

Like any news that quickly spreads in the wild because of its sensationalism, there are many details belonging to Stuxnet which are far from true.

1) *Stuxnet can hide itself in a PLC and from there infect a machine connected to it*.
   This is false. Rather than "infecting" a PLC, Stuxnet simply rewrites its organisation blocks (OBs) and does its best to hide it. The code injected cannot spread out from the targeted PLC, to another PLC or to a Windows machine.
2) *Stuxnet can infect non-networked Windows machines through USB sticks*.
   This is correct but misleading. Although Stuxnet will copy itself only to USB removable drives, basically it could be vectored through any removable media, including (but not limited to):
   - any USB removable storage (USB flash drives, USB hard disks, mobile phones, picture frames, etc.)
   - SD cards
   - CD or DVD
   - floppy disks (old but still in use)

For instance, if a user copies the full content of a Stuxnet-infected USB stick to a SD card, and then reads that SD card in a clean Windows computer with Explorer, that computer will also become infected.

Similarly, browsing an infected computer through a network share can spread the virus.

3) *Disabling the autorun feature of Windows will stop the worm from spreading via USB removable media*.
   This is one of the biggest and most dangerous mistakes. Stuxnet, more than relying on autorun.inf for being executed, exploits a system vulnerability related to .LNK files. Therefore, disabling the autorun feature will have absolutely no effect on the malware.
4) *Stuxnet is able to spread trough a network using a limited dictionary attack (weak passwords) to access network SMB shares.*
   Again, this is only partially correct. Stuxnet DOES use network shares to infiltrate a Windows system, but do not use weak passwords for doing so. Instead, it will use users' security tokens gathered from the infected machine and WMI (Windows Management Instrumentation) sessions to try to bypass user authentication in the attacked computers.

Verified and hypothetical evolutions of Stuxnet

The effort put into developing Stuxnet was great. As already said, it is estimated that a team of at least 30 of the world's top cyber experts have spent more than 10,000 man-days[73] in its creation, and the outcome is definitely satisfactory.
It is, therefore, unlikely that such a masterpiece of software – a real gem in malware programming - would be just dropped after its disclosure to the public domain.
Already in 2010, Langner foresaw that the malware was going to be reused, along with minor improvements and modifications needed to re-adapt it to new tasks. Considering its level of quality it is highly likely that the software development process of Stuxnet has followed strict management requirements such as versioning, technical documentation, object modularity and – thus – re-use.
According to other sources,[74] Stuxnet has also attracted the interest of criminal hackers, who have learned from it and in some cases have managed to "master" the malware for the purpose of reselling it to potential customers. In fact, Stuxnet has led to the emergence of new malware, as well as modifications of worms already existing, based on its exploits.

CVE-2010-2568 ***(MS10-046)*** *Windows LNK Shortcut Remote Code Execution*	***CVE-2010-2729*** ***(MS10-061)*** *Windows Print Spooler Service Remote Code Execution*	***CVE-2010-2743*** ***(MS10-073)*** *Windows Kernel Win32K.sys Keyboard Layout Privilege Escalation*	***CVE-2010-3888*** ***(MS10-092)*** *Windows Task Scheduler Privilege Escalation*
• *Trojan.Zlob*[75] • WORM_**SALITY**.RL • Chymine • WORM_OTORUN.ASH • WORM_OTORUN.KR • TROJ_**ZBOT**.BXW • WORM_**WEBMONER**.JC • Win32/Delfi (aka XBot) • Win32/Agent.OTB • *Vobfus.BK/C (aka VBNA)*	*(none so far; probably because this vulnerability is exploitable only in LANs with shared printers and, therefore, rather useless on the Internet)*	• Win32**.Carberp** (banking trojan) • Win32/Agent.OSW (aka Dottun)	• Win32**.Carberp** (banking trojan) • Trojan.Win32.**Ctfmon**.A (TLD4)

Tab. 5: Malware re-using Stuxnet's zero-day vulnerability exploits (ESET/ Trend Micro/ Symantec naming convention). Names in italics are malware not re-adapted, i.e. originally exploiting the vulnerabilities from their birth.

[73] Microsoft estimation. Equivalent to 27 man-years.

[74] Kaspersky, SearchSecurity.org, Eset, *et al*.

[75] Some variants of the Zlob trojan were the first to use the LNK vulnerability, in November 2008 (around seven months before the oldest known Stuxnet activity). It is possible that these malware instances were released by Stuxnet's authors during or before its development for experimental purposes.

For instance, the cyber criminals behind the TDL4 rootkit, a variant of the TDSS (Alureon) rootkit that caused many problems of Microsoft Windows "blue screens" (BSODs) in 2010, have copied the Stuxnet worm, exploiting the zero-day vulnerability *Windows Task Scheduler Privilege Escalation (CVE-2010-3888).*

In August 2010, the *Windows LNK Shortcut Remote Code Execution (CVE-2010-2568)* vulnerability disclosed by Stuxnet started to also be used by malware[76] such as *Vobfus* (a worm whose name was derived from the fact that it is coded in **V**isual Basic and is highly **obfus**cated, a.k.a. *W32.Changeup.C* and *VBNA*) and *Sality*, which spread around faster than the worm itself.

In September 2010, Eset, an international developer of antivirus software and computer security solutions, reported[77] the detection of several malicious programs exploiting the same vulnerability, belonging to a class of downloader threats (when installed on a computer, the programs download other malware such as the keylogger Win32/Spy.Agent.NSO) which were, therefore, given the name of *Win32/TrojanDownloader.Chymine family*. Eset also reported that their researchers observed some known malware, like the Win32/Autorun.VB.RP trojan, to have been "refurbished" with the CVE-2010-2568 vulnerability in order to include the related exploit as a new propagation vector. In addition, the virus analysts identified several variants of the virus Win32 Sality, as well as Trojan Zeus/Zbot, which were modified to use this vulnerability.

The possibility of the re-adaptation of Stuxnet with new payloads was seriously contemplated by the US Department of Homeland Security, and reported by the US Congressional Research Service to the US Congress on the second week of December in 2010. The report emphasised that a cyber weapon such as Stuxnet, if conveniently re-engineered, could be used to inflict widespread damage to US critical infrastructure services, hitting a wide range of targets as opposed to a narrow target such as Iran's nuclear facilities.

Moreover, plugin modules exploiting the 0-day vulnerabilities used by Stuxnet were easily available for penetration test environments such as Metasploit as far back as August 2010.

However, not everyone is persuaded that Stuxnet marks the beginning of a new era. According to Richard Aldrich, professor of international relations at the University of Warwick and a historian of Britain's signals intelligence agency GCHQ[78], "*Stuxnet is something that will work brilliantly the first time, less well the second time (and) hardly at all the third. Maybe Stuxnet and similar forms of attack have already had their day*". In effect, many believe that this type of weapon could become obsolete in a matter of months if left unused. It is also believed that Stuxnet served as a wake-up call that has driven the cyber defences of nation states to become increasingly effective. As often occurs, the truth probably sits in the middle.

Fig. 11: Life cycle of a cyber attack. As believed by many, once a cyber weapon is used its effectiveness quickly degrades.

[76] http://blogs.technet.com/b/mmpc/archive/2010/07/23/protection-for-new-malware-families-using-lnk-vulnerability.aspx.

[77] http://go.eset.com/us/resources/white-papers/chymine-whitepaper.pdf.

[78] Government Communications Headquarters. Centre for Her Majesty's Government's Signal Intelligence (SIGINT) activities.

Duqu, the "new Stuxnet"?

On October 18, 2011, Symantec claimed on its blog to the computer security community to have identified, through its affiliated technical laboratory CrySis[79] (Laboratory of Cryptography and System Security, Budapest University of Technology and Economics), what appeared to be a new malware derived directly from Stuxnet (enough to be called "the new Stuxnet", "the son of Stuxnet", or "Stuxnet 2.0"). The malware was dubbed *Duqu* because it was creating files with the prefix "~DQ".
According to CrySys researchers, who were the first in the world to identify the malware, probably at the end of August 2011, Duqu "*is a threat nearly identical to Stuxnet*" but appears to be aimed at data exfiltration (info-stealing) rather than cyber sabotage. To penetrate target systems, the malware uses accurately socially-engineered e-mails with an attached Word file. Duqu exploits a zero-day vulnerability[80] (CVE-2011-3402) in the Windows kernel (`win32k.sys`), related to the rendering of TrueType fonts which are embedded in a specifically malformed Word file. Therefore, when a Word file of this kind is open, the malicious code is executed and installs its components (DLLs and system drivers) via a delayed-action Trojan dropper.
Once installed on a system, Duqu contacts a C&C server and sends and receives steganographed data in the form of JPG files. It has been reported that Duqu can download other malware such as TROJ_SHADOW.AF[81], a trojan capable of collecting various information about the machine on which is run, into the infected system.
The malware is scheduled to run only for a limited period of time (CrySyS reported eight days in August 2011), and is removed automatically from the infected system after a configurable number of days from its installation.
Some findings published in November 2011 by Kaspersky conclude that there are at least 12 versions of Duqu, and that every single instance of the malware is slightly different from the others, probably because it has been adapted and optimised for a particular target just before an attack, including the use of a different C&C server.

It is believed that Duqu has caused reported incidents in Hungary, the UK, the USA, Iran, Sudan and possibly Austria and Indonesia. E-mails sent to one of the victims as far back as April 2011 were discovered.
According to Trend Micro[82], Duqu is similar to Stuxnet in the following aspects:

- It uses UPX compression
- It uses 2 configuration files
- It spreads through network shares
- It checks safe mode and kernel debugger
- It uses digital certificates to sign executable modules
- It injects the DLL to a target process via SYS component (kernel-mode loader)
- It hooks Windows APIs
- It is made up of multiple components that inter-communicate
- It uses RPC (remote procedure call)

Liam O Murchu, manager of operations for Symantec Security Response, said that their analysis "*shows that 50 percent of the code in Duqu is exactly the same as code used in Stuxnet. This means that the creators of Duqu had access to the source code from Stuxnet*."[83]

[79] http://crysys.hu/.
[80] Microsoft has released security advisory MS11-087 to address this issue. The vulnerability addressed is the TrueType Font Parsing Vulnerability (CVE-2011-3402).
[81] Trend Micro naming convention.
[82] http://about-threats.trendmicro.com/relatedthreats.aspx?language=us&name=DUQU %20Uses%20 STUXNET-Like%20 Techniques%20to%20Conduct%20Information%20Theft.
[83] http://www.darkreading.com/advanced-threats/167901091/security/attacks-breaches/231901226/waiting-for-son-of-stuxnet-to-attack.html.

Statistically,[84] non-related malware samples have less than a 25% code match (static code similarity). A 50% percent match between Stuxnet and Duqu is, therefore, remarkable, and very unlikely to be achieved without a "common heritage" in the original source code.

On the other hand, some researchers[85] state that this significant similarity between the two malicious programs could have been attained even without access to Stuxnet's source code, through good reverse engineering and re-compilation of the executables.

To summarise, although investigations into Duqu are still in progress, the following points can be stated:

- the goal of Duqu is to steal data;
- Duqu contains code structures similar to those used by Stuxnet;
- Duqu uses some very specific techniques unique to Stuxnet, specifically the use of kernel-mode loader drivers and their signature with "legitimate" digital certificates. It is still unknown if these certificates were issued with stolen keys or were generated by the attackers through a compromised certificate authority. It is also interesting to note that the digital certificate used to authenticate the driver comes from a Taiwanese company (C-Media Electronics Inc.), as was the case with Stuxnet. To date, Duqu and Stuxnet are the only two malware instances known to use a kernel driver signed with a legitimate cerfificate;
- Duqu is highly configurable and customisable; each investigated attack seems to be using different exploit files;
- unlike Stuxnet, Duqu infiltrated targeted systems within a spear phishing attack, through email attachments;
- unlike Stuxnet, the techniques used to conceal the exfiltrated data are good, and encryption is stronger;
- unlike Stuxnet, Duqu does not access or re-program PLCs of any type, nor does it affect SCADA systems;
- unlike Stuxnet, Duqu appears not to replicate in any way and, therefore, belongs to a class of malware different to that of viruses or worms .

For the above reasons, it is somehow misleading to describe Duqu as "the new Stuxnet", even if there is a high likelihood that it was written by the same hands that wrote Stuxnet.
According to Kaspersky, Duqu is not the successor of Stuxnet, but rather a "cousin". Duqu seems to have been developed concurrently with Stuxnet, probably starting in 2007, and by the same programmers. Kaspersky supposes[86] that Duqu and Stuxnet are parts of the same sophisticated cyber arsenal, which probably encompasses many other specialised malwares (Kaspersky talks of at least another three), conceived to be used for specific purposes upon need. This arsenal has been dubbed "the Tilded Platform", because of the tilde symbol (~) frequently used in filenames by both malwares.
Kaspersky believes that the Tilded Platform traces back to at least 2007 because the compilation date of Duqu's components is set to August 31, 2007.
There is also a good possibility that Stars, the info-stealing malware that Iranians claimed to have discovered on their computers in April 2011,[87] is closely related to Duqu, or is even the same malware. Again according to Kaspersky, the timeline of events coincides, and the fact that a JPG image of the NGC 6745 galaxy was found embedded in the keylogger component of Duqu is probably the reason why the Iranians dubbed it "Stars". Duqu could, therefore, be an updated version of Stars.
It is also interesting to mention a theory from John Bumgarner, who asserts that Stuxnet's operators started carrying out reconnaissance in 2007 using Duqu, which exfiltrated information about components used in Iran's nuclear and critical infrastructure facilities. If this were true, then it should be stated that Stuxnet is the son of Duqu, rather than the opposite.

---

84 Jeremy Sparks, SC Magazine (http://www.scmagazine.com/duqu-father-son-or-unholy-ghost-of-stuxnet/article/215851/).

85 http://www.nsslabs.com/blog/2011/11/duqu-analysis-and-detection-tool.html.

86 http://www.reuters.com/article/2011/12/28/us-cybersecurity-stuxnet-idUSTRE7BR1EV20111228.

87 http://www.mehrnews.com/en/newsdetail.aspx?NewsID=1297506.

In our opinion, the spreading of Duqu has been a test. Given that its worldwide distribution has been limited to areas that are very different and far apart, in countries that do not seem to have much in common, and that it is quickly self-destructing and active only for a limited time window; chances are that Duqu is experimental malware, still under refinement, which its authors wanted to test in the wild in order to understand its efficiency.

Lessons learned

There are a lot of important lessons that can be learned from Stuxnet. The most meaningful are:

- Cyber threats can effectively target physical assets just as conventional weapons can.
- Air gapped/ isolated systems are not automatically secure.
- Removable media such as USB Flash Drives (UFD) are a very serious threat to cyber security.
- Digital certificates can be impaired and do not offer real protection if their compliance is not strictly enforced and checked.
- Device drivers can be installed in Windows even if their digital certificates are revoked (e.g. when there is no Internet access for verifying CRLs), and also without any warning to the users. This is a behaviour that is deeply unsecure for any operating system.
- Private keys must be saved and protected with extreme care, possibly by using smart cards and HMSs. To access this kind of data, strong authentication (two or three factors) is a must. Biometric solutions are still very effective for this purpose.
- Hard-coded passwords are unacceptable from a security perspective. Default passwords are of course necessary, but systems must force users to change them at their first access, without exceptions.
- Emergency access to systems should not be done via hard-coded credentials, rather only through physical intervention (smart card, DIP switches, etc.).
- Compartmentalise and isolate: do not allow inter-network communications; do not allow Internet access; do not allow any data exchange to key systems (USB connections).

Aftermath / Consequences of the Stuxnet case

- Much more attention is being paid to cyber warfare issues that – despite worldwide evidence – were until now believed by most people to be "experimental" or not really able to cause significant damage.
- Re-thinking of security of SCADA systems, with probable creation of ad-hoc international working groups that will review the SCADA protocols and introduce encryption and mutual authentication for communication between HMI stations, PG and PLCs.
  Automation manufacturers could also seriously consider the possibility of developing a more secure, industrial process-oriented, compact operating system that can replace Windows for hosting control, programming and visualisation applications. A customised version of Linux (or BSD Unix, as in the case of Kylin) could be tailored for this purpose.
- Stricter rules for the management of digital certificates, such as secure methods for storing the private key (i.e. through smart cards or Hardware Security Modules, like those of Certification Authorities) and quality certifications for organisations who rely on PKI for signing and distributing potentially dangerous software.
- Stuxnet has shown to the world how powerful and effective a cyber weapon can be, making the unthinkable real. It is obvious that states will now do their best to develop an equivalent capability. We must, therefore, expect more and better malware of this kind in the future, even if used only in specific situations and for very specific targets (copycat attacks).
- Whoever was behind Stuxnet (be it a single country or – more likely – a coalition), they now have a substantial advantage over other countries in cyber warfare experience. The people who programmed Stuxnet are precious human resources that any organisation would keep operational

and ready for another succesful strike. It is, therefore, highly probable that in the future some other smart malware will pop out from the same authors.

- Stuxnet is also an example for other subjects to imitate, such as criminal organisations, hackers or terrorist groups. Althought directed to different targets, the technical quality of this malware is very high and worth understanding for re-exploitation purposes (see previous paragraph → "*Possible evolutions of Stuxnet*").

Geopolitical considerations

Developing the capability for energy production through the exploitation of nuclear fission sounds rather out of place for a country like Iran, a state which owns one of the biggest oilfields in the world (that will probably last for the next 80 years) and that ranks fourth in the list of the biggest oil producers in the world (about 5% of the world total, of which only 40% is used for internal needs). Consider also that the investment costs for the nuclear programme easily exceeds two billion dollars,[88] money that could instead be spent on more compelling needs, such as public instruction (Iran has an estimated 20% illiteracy rate) or the healthcare system.

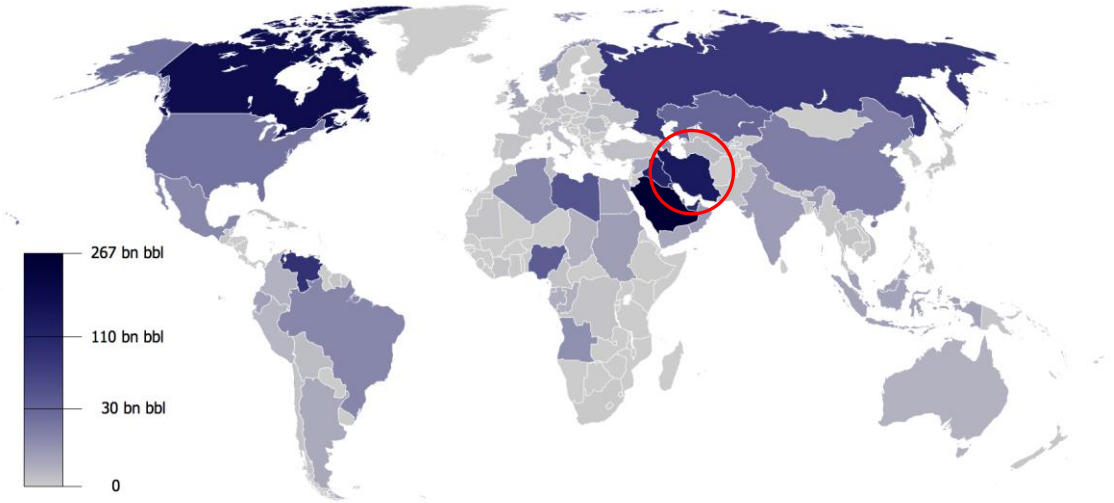

Fig. 12: World Oil Reserves (2009). In 2009, Iran (circled in red) produced more than 1539 million barrels of crude oil (one barrel=159 lt.)[1]. Its natural reserves, considering the extraction rate assessed in 2009, are estimated to last for the next 89 years.

These considerations, especially when considered alongside the strong opposition expressed by Ahmadinejad in violent verbal aggressions against Israel, which he said "*must be wiped off the map*"[89], have caused deep international concern about the possibility that Iran could use the uranium enrichment process and its nuclear programme in general to feed the construction of nuclear WMDs.

The history of the Iranian nuclear programme dates back to the 1950s, when the Shah Mohammad Reza Pahlavi signed several agreements with the United States and other European countries, specifically France and Germany, which possessed nuclear technologies. The site chosen for the first nuclear plant, whose building was committed to the German companies Siemens and AEG-Telefunken, was the city of Bushehr. The building of the reactor began in 1975.
After the Iranian revolution, relations with western countries were interrupted and, consequently, the development of the project stopped. In addition, during the war with Iraq (1980-1988), the Bushehr plant was damaged by Iraqi bombardments.

---

88 One billion dollars for the Bushehr site and one billion dollars for the Natanz enrichment site.

89 IRIB News, "*Ahmadinejad: Israel must be wiped off the map*", 26-Oct-2005, (http://web.archive.org/web/20070927213903/http://www.iribnews.ir/Full_en.asp?news_id=200247).

The nuclear programme was restarted in 1995 with an agreement between Moscow and Tehran for the re-construction of the Bushehr nuclear plant. The delivery of the plant was scheduled for 1999, but the date was postponed several times due to technical and financial problems, and also because of strong pressure from the international community, which feared that Tehran's nuclear programme had military goals.

Actually, the massive support given by Russia in building the Bushehr nuclear power plant (Rosatom and Atomstroyexport), can be clearly interpreted as an attempt to re-establish an effective political influence in the Near-Eastern region that was lost following the collapse of the Soviet Union.
As confirmation of this, along with its engagement in the Iranian nuclear programme, Russia is also providing powerful armament to Syria (S-1 "Pantsir" and S-300[90] surface-to-air missiles, SS-N-26 "Yakhont" anti-ship missiles, MiG-29 "Fulcrum" aircrafts) and to Iran (surface-to-air missiles of the Tor-M1[91] type), and was granted permission to install a new naval base on the Syrian coast. Additionally, the Russian companies Stroitransgaz and Gazprom will ensure the transit of Syrian gas to Lebanon, the latter not having yet come to an agreement with Israel (with which it has been officially in a state of war since 2006) to exploit the large reserves located offshore in the international waters just in front of the two countries.

Fig. 13: The Russian SS-N-26 (in the photo, launched from a ship) is believed to be one of the most lethal anti-ship missiles in the world. Such a weapon, which is owned by Syria, could easily sink Israeli and American frigates.

Going a little deeper, and considering also future Russian engagements in Turkey to build a nuclear power plant and an oil pipeline, some analysts have seen in these Russian moves the will to support the Tehran-Damascus-Ankara triangle (at least before the Turkish condemnation of Syrian massive repression against internal turmoil) in the face of US, Israeli and European hostility, thus effectively tipping the strategic balance in the Middle East.

Fig. 14: Map of Iranian nuclear-related sites

[90] The S-300 is a whole family of more than 25 different Russian SAMs, some of which are identified by NATO/ DoD with the reporting names of SA-N-6, SA-10 "*Grumble*", SA-12 "*Gladiator*", SA-20 "*Gargoyle*".

[91] Originally developed by the Soviet Union under the designation name 9K330, and identified by NATO with the reporting name of SA-15 "*Gauntlet*".

In this highly critical geo-political scenario, a conventional attack on Iran's nuclear installation, such as an Israeli air strike, would cause dangerous reactions not only from the surrounding countries but also from Russia and China. More than once, through the voice of President Dmitry Medvedev, Russia has spoken of its possible direct involvement and warned Israel and the United States against an armed conflict in Iran that could degenerate into a large-scale war where anything could be expected, "*including use of nuclear weapons.*"[92]
Being well aware of these extremely heavy implications, in 2008 President Bush – according to the New York Times[93] - had already rejected a request by Israeli Prime Minister Ehud Olmer for specialised bunker-busting bombs and for permission to fly over Iraq to attack the Natanz plant.
In such a situation, resorting to a cyber weapon – whose usage is notoriously always difficult if not impossible to attribute to someone – to compromise Iran's nuclear capability seems to be a truly *ideal solution*. This gives to cyber weapons the same effectiveness of a conventional military strike, but without the risk of casualties or of degenerating into overt war.

The following table summarises the kind of advantages a cyber weapon like Stuxnet can provide, compared with a conventional attack:

	CONVENTIONAL (air strike) ✈	**CYBER** (malware) 🖫
Financial resources	Very high	Medium to high
Casualties	High	Basically null
Loss risk (equipment and lives)	Low to medium	Basically null
Loss costs (equipment and lives)	Very high	Low
Deniability	Difficult	Easy
Attribution	Easy	Very difficult
Side-effects costs	High (geo-political fallouts)	Very low
Stealth	Partial	Yes
Preparation time	Short	Variable
Execution time span	Very short	Medium to very long
Predictability	Good	Difficult
Effect visibility (results)	Almost immediate	Difficult to quantify in the short run
Probability of success	High	Medium
Additional goals (incidental or planned)	Limited	Intelligence, network mapping/ exploitation, information gathering
Re-usability	Only in some cases	Yes (mostly with adaptation)

Tab. 6: Differences between a conventional war attack and a cyber war attack

Other covert operations against the Iranian nuclear programme

To correctly understand the Stuxnet story, and to frame it into the world strategic scenario to which it belongs, it is important to obtain the whole picture.
Attempts to slow down or stop the Iranian nuclear programme were also made by means other than cyber attacks. Beginning in 2008, several conventional sabotage operations against Iran's personnel and structures were conducted.
The most recent episode occurred on January 11, 2012, when Mostafa Ahmadi Roshan, a University professor in Tehran and deputy director of the Natanz enrichment facility, was killed.

[92] Medvedev's interview with George Stephanopoulos, April 12, 2010 (http://abcnews.go.com/GMA/transcript-george-stephanopoulos-interviews-russian-president-dmitry-medvedev/story?id=10348116 &page=8).
[93] http://www.nytimes.com/2009/01/11/washington/11iran.htm.

In December 2011, no fewer than seven people died in an explosion in a steel factory in Yazd (central Iran, 500 km southeast of Tehran), a location often reported as possibly concealing covert nuclear facilities.
On November 28, 2011, another explosion took place in the uranium conversion facility of Isfahan (340km south of Tehran), involved in processing uranium which is then fed to the Natanz fuel enrichment plant and, and operating under IAEA surveillance.
On November 12, 2011, a huge explosion killed 17 and wounded a further 15 at the Bid Ganeh base, a site near Qom, about 35 km west of Tehran. A military missile expert, the head of the missile program General Hasan Moghaddam, was killed. There is some speculation (Time magazine quoted "*western intelligence sources*"[94]) that the Mossad was allegedly responsible for the event and that at the Bid Ganeh base a new uranium enrichment plant was secretly under development. Other sources report that Shabab missiles were stored in the Bid Ganeh base, capable of reaching targets within a 2000 km range and, therefore, being ideal vectors for launching nuclear warheads against Israel.
On November 29, 2010, Dr. Majid Shahriari was killed and Dr. Fereydoon Abassi was seriously injured in two sticky-bomb attacks, reportedly identical to the one that killed Roshan. The two Iranian scientists were also both involved in Iran's nuclear development programme, with Shahriari – according to Iranian media – heading the team responsible for developing the technology of a nuclear reactor core, and Abassi (now Head of the Atomic Energy Organisation of Iran) being directly engaged in the nuclear weaponisation programme.
All the above-mentioned attacks on Iranian scientists followed an identical *modus operandi*: assailants on motorbikes approached the vehicles of the victims and attached an improvised explosive device (IED) to the doors, which exploded seconds later.
On July 23, 2011, another nuclear scientist called Darioush Rezaeinejad was shot and killed outside his daughter's kindergarten by two gunmen on a motorcycle.
Finally, in 2007, the scientist Ardeshir Hassanpour was killed while carrying out his nuclear work at Isfahan, in an alleged radioactive poisoning that the private intelligence company Stratfor attributed to Israel. In its report about the incident, Stratfor explained with a cynical but sharp logic the ongoing assassinations: "*Decapitating a hostile nuclear program by taking out key human assets is a tactic that has proven its effectiveness over the years, particularly in the case of Iraq. In the months leading up to the 1981 Israeli airstrike on Iraq's Osirak reactor – which was believed to be on the verge of producing plutonium for a weapons program – at least three Iraqi nuclear scientists died under mysterious circumstances.*" A very clear concept, recently reaffirmed by David Albright, president of the Institute for Science and International Security: "*The idea clearly is to try to disrupt operations that could lead to a nuclear weapon, and to make their scientists feel less secure and less capable of doing their work*." However, many western commentators have condemned these sabotages and homicides, warning that such actions could have the effect of increasing international tension and pushing Iran to extreme reactions, rather than effectively endangering the Iranian nuclear programme.
The Iranian regime has repeatedly accused Israel and the United States of most of the attacks. While the US has always strongly denied its involvement, Israel held a more ambiguous position, refusing to officially comment.

The facts listed above are undeniable indicators that well-trained operatives, probably working for foreign powers, are actively trying to prevent Iran from developing nuclear WMDs, confirming somewhat that a covert war is already in progress by several years . The need to stop Iran from developing the atomic bomb is too critical to rely only on a cyber weapon, no matter how sophisticated.

Evolution of the geo-political situation

Iran continues to develop its nuclear programme, despite continuous invitations from the international community for greater collaboration in showing its peaceful intent, official reports from AIEA and other important arms control organisations documenting incontrovertibly the Iranian efforts for

[94] http://www.time.com/time/world/article/0,8599,2099376,00.html

building nuclear WMDs, increasingly stricter international sanctions and Israel openly requesting a conventional military strike. The recent Iranian threat to close the strait of Hormuz in retaliation for other possible international sanctions has increased the international tension in the Gulf even more, causing the US to immediately send warships to patrol the area. A few days later, the EU decided to impose a complete embargo against Iranian oil export, starting from July 1st 2012.
At the time of writing this report, it looks like Iran has never been so alone against the whole world.
Unless Iran really withdraws from any belligerent will of having nuclear weapons, the possible developments do not leave much room for optimism, and the possibility of an Israeli military strike in the next months seems increasingly likely, despite American vetos. Under this perspective, we sadly have to say that we wish that Stuxnet was much more effective, if this could avert the occurence of a real conflict.

A step beyond: what next in cyber warfare?

After debating the Stuxnet facts, many have concluded that the border between the "virtual world" and "real world" (or, more correctly, between the electronic domain and the physical domain: digital/ analogue) has been crossed.
This is something that is only relatively new and not so surprising after all. Interaction between computers and analogue I/O has been happening for more than 30 years in the industry, especially in the fields of automotive serialised building and digitally-controlled manufacturing machines (objects manipulation and processing). Actually, due to the fall in price of electronics and high scale integration, in the domestic setting all kinds of devices are also starting to be connected to networks (e.g. burglar alarm systems, fridges, Internet TV, remote-controlled home heating systems, meteorological sensors, irrigation systems, etc.). This leads to many possibilities in the years to come.
Let us consider three large domains of application of ICSs: the chemical industries, space programmes, and medical systems. The speculations made here could be easily and quickly classified as sci-fi, but before Stuxnet was discovered and analysed, the same would have been said about "sabotaging a country's nuclear development programme through the use of a cyber weapon".

Oil processing plants are very important for the economy. Sabotaging these installations would slow down terribly (or even stop) the sale of oil from producer to consumer states. This could be done not only to harm a country's economy, but also to manipulate the price of oil for financial gain (criminal intent): buying oil at a low price before the sabotage and re-sellling it at much higher price after the sabotage.

Chemical plants producing pesticide substances would also be likely targets for cyber sabotage, as these kinds of installations can be quite easily converted for the production of chemical weapons, including nerve agents (Sarin, VX). Such threats have existed in the past in Libya and Iraq, and those situations are very similar to that in Iran (which claims the uranium enrichment process is carried out only for peaceful purposes).

Cyber-sabotage of space programmes (NASA, Russia, ESA, China, India) is also a realistic possibility, given the strategic returns that satellites provide in terms of intelligence, communications, and guidance. For instance, it is currently rumoured that some countries are very concerned about the strong acceleration that India has put in its space programme, as they see these technology improvements as directed not only to enhance ICBM navigation and targeting support, but also to develop a MIRV (Multiple Independently targetable Reentry Vehicle) capability. States that would feel threatened by this situation (such as the bordering Pakistan, which is a historical enemy of India and also a nuclear power, or China, which is competing with India in the race to land on the moon,[95]) could consider the attempt to slow down the Indian space programme through the means of cyber weapons a valid option, to prevent the development of the above-mentioned capabilities. Since software is

[95] Article "*Did The Stuxnet Worm Kill India's INSAT-4B Satellite*?", by Jeffrey Carr, 29-Sep-2010, http://www.forbes.com/sites/firewall/2010/09/29/did-the-stuxnet-worm-kill-indias-insat-4b-satellite/

deeply devoted to control engines and the stabilisation of rockets vectoring satellites and/or human crews during all phases of the space mission (ignition, lift-off, escape and orbital insertion), destroying a rocket through software manipulation is easy, so easy that in the past this has happened even *without* a sabotage intention. In 1996, a software programming error (integer overflow) caused the rocket Ariane V, launched from the French Guiana, to explode after 36 seconds of flight, resulting in the loss of the vector and of the onboard satellite with total damages costing over 500 million dollars.

Although medical systems are more difficult to access because of different architecture standards (an industry standard[96] for medical devices exists but this is only for the capture of data), malware programs that identify people (by searching the patient's name or ID in X-rays, echo, ultra-sonography, CAT, PET machines) and try to harm them by re-programming life support systems (respiratory machines, X-rays scanners, anaesthetic ventilators, breathing blenders, intravenous infusion devices, cardio monitors) are a concrete possibility.

A precedent in history already exists in this area. The French/Canadian medical system AECL CGR Therac-25, a computer-controlled[97] advanced medical linear accelerator (linac), caused the deaths of no fewer than five people between 1985 and 1987 by exposing patients to an overdose of beta radiation (about 100 times the intended dose), due to a software bug acting on the electron beam power regulator[98]. It is also worth remembering the case of the X-ray department at Lund University Hospital, which in 2004 was attacked by the Sasser worm. As a result, all four X-ray machines in the hospital were disabled for several hours, forcing the personnel to redirect emergency patients to other hospitals.

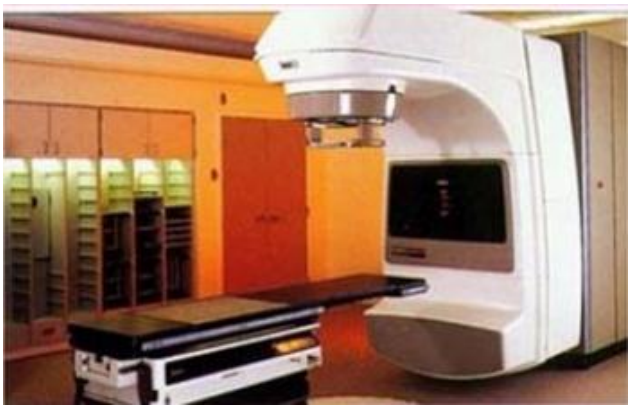

Fig. 15: the Therac-25 medical appliance

[96] ANSI/IEEE 1073 Medical Information Bus (MIB) data communication standard.

[97] The Therac-25 was controlled by a PDP-11, a 16-bit computer by Digital Equipment Corporation.

[98] "*Medical devices: the Therac-25*", by Nancy Leveson, University of Washington (http://sunnyday.mit.edu/papers/ therac.pdf).

Conclusions and Summary

Stuxnet has been defined in many ways: “the most important malware of the decade", “a milestone in cyber threats history”, “the world's first cyber super weapon”, “the most sophisticated cyberweapon ever created“, “a watershed event“[99], “an eye-opener for our whole industry“[100], and even a “game changer”[101].

The real novelty of Stuxnet is the fact that it is a true cyber weapon, providing real damages in the physical world with which we all interact every day.

Beyond speculation and sensationalism, it is evident that the Stuxnet worm marks a new age in cyber warfare.

Attacks on industrial plants that are capable of causing bodily harm are considered within the so-called problems of CIP (Critical Infrastructure Protection). Until now, these cases have consisted of assumed DDoS attacks or intrusions by attackers in industrial control systems that could activate or disable specific features in order to cause damage.

The possibility of a "stand-off" attack, all led by malware without the direct ("synchronous") control of a human attacker, had not yet been taken into account, and is this main feature that makes Stuxnet such a powerful weapon. Stuxnet is a kind of "fire-and-forget" cyber weapon, programmed from birth to search and damage a specific type of industrial plant. Once "set free" it will do everything by itself, but without "calling home" to let people know what it is doing and what it has found, or to receive instructions and/or additional modes of operation.

"*Stuxnet is a 100-percent-directed cyber attack aimed at destroying an industrial process in the physical world*", said Ralph Langner.[102] "*This is not about espionage, as some have said. This is a 100 percent sabotage attack*."

In conclusion, it doesn't really matter if Stuxnet has been able to effectively delay the Iranian nuclear programme by two months or two days. Regardless of its real proven effectiveness, Stuxnet, more than anything else, is one thing: a demonstration that cyber warfare and weaponised software are a reality, and that they can achieve unprecedented results in damaging critical infrastructure.

Call it an eye opener or a wake-up call, we will always remember Stuxnet for this.

99 Janet Napolitano, Head of US Department of Homeland Security (Berkeley, 25 Apr 2011).

100 Heikki Hiltunen, Executive Vice President of Vacon.

101 Mikko Hyppönen, F-Secure Chief Research Officer: *"I do believe Stuxnet will stay in history books as a game-changer*".

102 http://www.csmonitor.com/USA/2010/0921/Stuxnet-malware-is-weapon-out-to-destroy-Iran-s-Bushehr-nuclear-plant.

APPENDIX A
Example schema of a SCADA/PLC Industrial Control System

The schema below depicts a simple industrial control system (ICS) with its SCADA (Supervisory Control And Data Acquisition) component controlling PLCs for water flow. The task of the ICS system is to move water according to certain *setpoints*, which are thresholds upon which an action must be taken. Note that, although the real SCADA system only contains the components framed within the yellow rectangle (HMI + SV CPU + DB), the full ICS is often - but improperly - called SCADA.

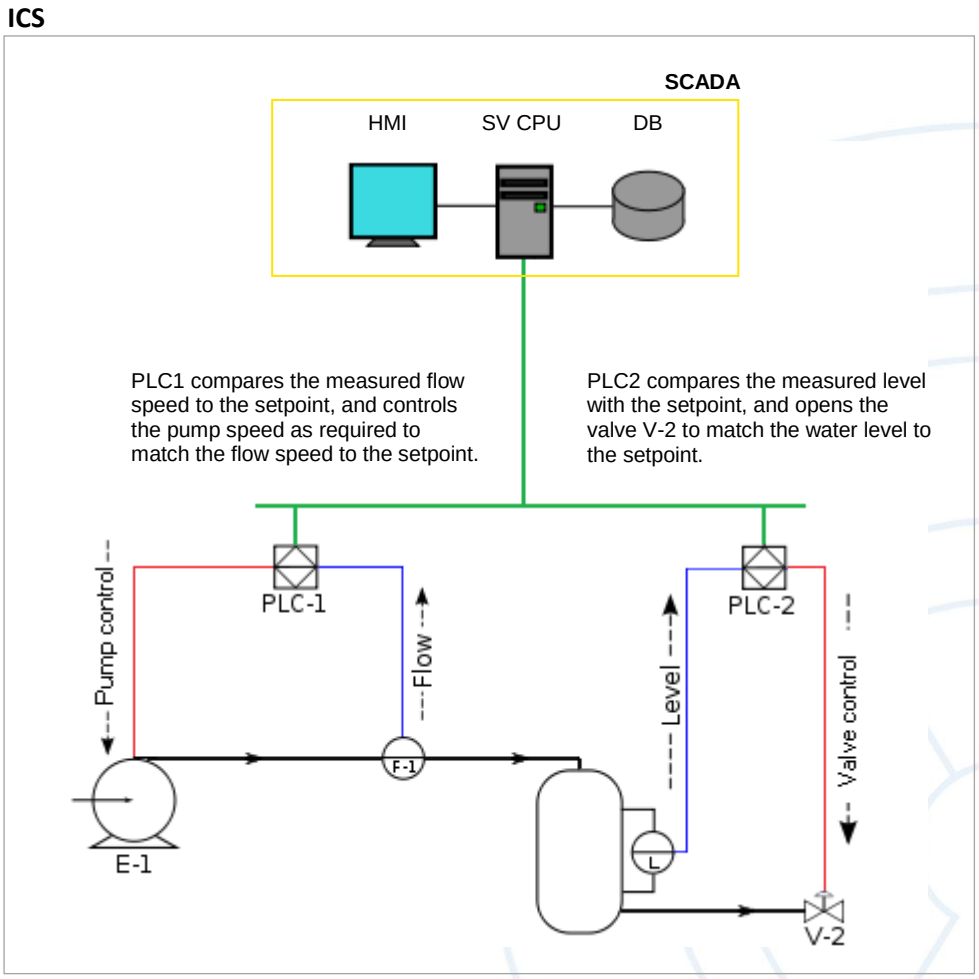

Fig. 16: ICS/SCADA/PLC schema

E-1	Electrical pump
F-1	Flow-meter (speed of flow)
L	Level-meter
V-2	Output electro-valve
HMI	Human Machine Interface
SV CPU	Supervisor station
DB	Database

Process control network
PLC signal IN (from sensors)
PLC signal OUT (to actuators)
Water flow

In detail, the SCADA system:
- starts and stops the operation of PLCs
- reads values from the PLCs
- displays data, status of processes, and alerts to a human operator (HMI)
- if needed, saves these values in a database (DB)
- if needed, sets values in the PLCs (e.g. setpoints)

APPENDIX B
Uranium processing for civilian nuclear use

Nuclear reactors use uranium as fuel for producing heat[104]. Although uranium is a relatively common element on Earth, not all the natural deposits can be conveniently exploited, depending on their U^{235} isotope concentration (the fissile part of the mineral). Important uranium mines are currently operating in Kazakhstan, Canada, and Australia, which are the top three producers in the world (producing over 60% of the total production).

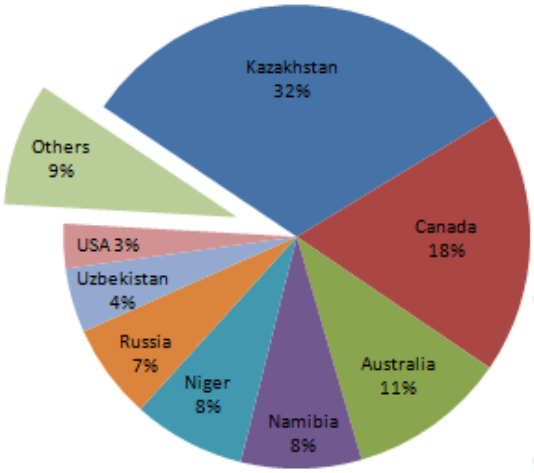

Fig. 17: Uranium production by country, in 2010. The total production reached 53663 tonnes[103].

Once extracted, the raw uranium is not yet suitable to start any nuclear fission reaction[105]. To become useable in this sense, it must undergo several processing steps which are aimed to increase the concentration of fissile uranium (U^{235}) from 0.5-0.7% to 3-5%, summarised in the following table:

Phase	*Obtained product*	*Rel. mass* (t. / %)	*Description*
Extraction ↓	Raw mineral	150000 (100%)	Extracted by digging and crushing hard rocks (pitchblende) or through in-situ leaching. The raw uranium is typically UO_2, containing both U^{235} (0.7%) and U^{238} (99.3%) isotopes.
Refinement ↓	"Yellowcake"	200 (0.13%)	Concentrated uranium oxide powder (70% to 90% of U_3O_8), obtained through milling and chemical treatment (leaching solutions and precipitation) of the raw mineral
Conversion ↓	Uranium Hexafluoryde (UF_6)	170 (0.11%)	In this process, the yellowcake is combined with fluorine (hydrofluoridric acid) to form uranium hexafluoride (UF_6), which can assume solid, liquid or gaseous consistency depending upon temperature and pressure[106].
Enrichment ↓	Enriched UF_6 (LEU)	24 (0.016%)	Purified uranium metal enriched in the U^{235} isotope (3-5%), obtained through isotope separation, carried out using gaseous *diffusion* or *centrifugation*. The enrichment basically separates U^{235} from U^{238}, the latter being known as "depleted uranium" (1:6 ratio). **This is the process targeted by Stuxnet.**
Fabrication ↓	Enriched uranium dioxide (UO_2)	24 (0.016%)	Enriched UF_6 gas is converted to uranium dioxide, which is formed into ceramic fuel *pellets* by baking it at a high temperature (over 1400°C).
Encasing	Fuel rods		Pellets of uranium dioxide are encased in long metal tubes made of neutron-transparent zirconium alloy, which are arranged in fuel assemblies. Uranium dioxide fuels are used in PWR, BWR and Candu reactors.

Tab. 7: Phases of uranium processing

[103] Data source: World Nuclear Association Market Report data (http://www.world-nuclear.org/info/inf23.html).

[104] Assuming complete fission, one kilogram of uranium 235 can theoretically produce about 80 terajoules of energy, which is the same amount of energy obtainable from 3000 tonnes of coal (N.B.: 1 tonne=1000 kg).

[105] At least for most reactors. Some types of reactors (russian RBMK and canadian CANDU) can use natural uranium, but are not very common. Moreover, RBMK reactors (the type involved in the Chernobyl disaster) are considered obsolete and unreliable from a safety perspective.

[106] At normal atmospheric pressure (1 bar), UF_6 is a solid below a temperature of 57°C and a gas at temperatures above.

The process detailed in the previous table can be pictorially summarised in the figure below:

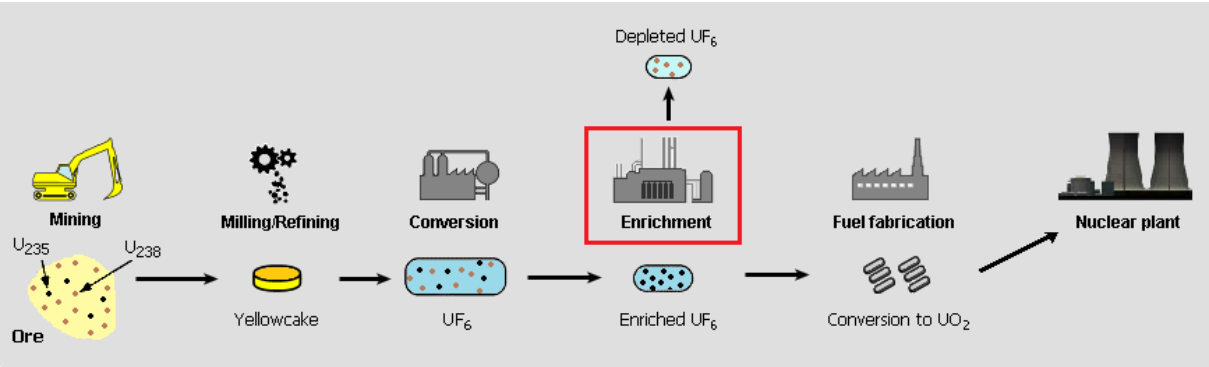

Fig. 18: Graphical depiction of uranium processing for use in nuclear power plants. The red squared icon represents the type of facility targeted by Stuxnet.

In 2011, the price of mined uranium (yellowcake) was, according to the World Nuclear Association[107], around 120 $/kg. The price of enriched uranium in fuel rods is about 2500 $/kg (roughly 20 times the price of the yellowcake). It is however expected that in the future the uranium price will increase, especially after 2013, when the agreement between Russia and USA for the conversion[108] of nuclear warheads HEU into civilian nuclear fuel will cease.

Commercial nuclear power plants use fuel that is typically uranium 235 enriched to around 3%. A 1000 MW reactor (such as the Bushehr reactor, see Annex C) operates with approximately 75 tonnes of fuel loaded, with approximately 25 tonnes replaced each year (equivalent to 200 tonnes of yellowcake, costing approximately 50 million dollars).

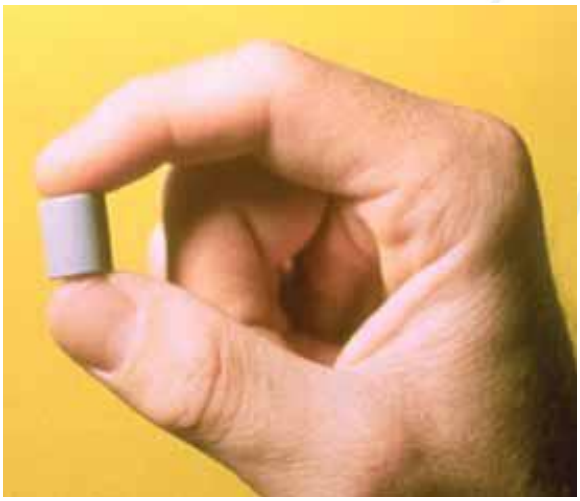

Fig. 19: A uranium dioxide fuel pellet (7 g.). Five fuel pellets meet the electricity needs of a household for one year. A large Westinghouse pressurised water reactor contains 193 fuel assemblies, nearly 51000 fuel rods and approximately 18 million fuel pellets.

According to OECD[109] statistics and Iranian announcements[110], in 2010 Iran produced about 20 tonnes of natural uranium, less than 0.04% of the world production, for an equivalent economic value lower than 2.5 million dollars.

Such a small production, which represents approximately one tenth of the amount needed to feed a nuclear reactor, hardly justifies the need for enrichment facilities, which are known to be very capital intensive, each requiring an investment of over 1 billion dollars[111].

As ISIS has observed, "*Iran's enormous investment into Natanz is unlikely to pay off in terms of utilising LEU in power reactors*"[112].

---

107 "*The Nuclear Fuel Cycle*", NWA (http://www.world-nuclear.org/education/nfc.htm)

108 So-called "*Megaton to Megawatt*" programme, started in 1995 within the USA-USSR WMD non-proliferation agreement. This programme has eliminated the equivalent of 16,000 nuclear warheads.

109 Organisation for Economic Co-operation and Development.

110 http://www.guardian.co.uk/world/2010/dec/05/iran-nuclear-power-domestic-uranium.

111 In 2010, the international community asked Iran to stop its enrichment facilities and offered to enrich uranium up to 20% in Russia, Turkey or France, according to its preference. This would have reached the two-fold goal of transparency and expense containment, but Tehran refused.

112 http://isis-online.org/isis-reports/detail/natanz-enrichment-site-boondoggle-or-part-of-an-atomic-bomb-production-comp/.

Enrichment process details

Uranium enrichment separates the fissile U-235 isotope from the U-238 isotope, and is carried out by *diffusion* or *centrifugation*.

Diffusion enrichment works by exploiting the different speeds at which U_{235} and U_{238} pass through a membrane.

Centrifuge enrichment works by passing the gas through spinning cylinders. The centrifugal force moves the heavier U_{238} to the outside of the cylinder, leaving a higher concentration of U_{235} on the inside (the U_{235} gravitating toward the centre of the rotor tube).

Currently, centrifuge enrichment is the preferred technology, as its energy requirements are 95% lower than the requirements of a comparably sized gaseous diffusion plant per unit of enrichment.

Since the desired enrichment level cannot be achieved in one centrifuge, several machines must be connected in series and parallel in what is called a "cascade". A centrifuge enrichment plant is made up of multiple cascades.

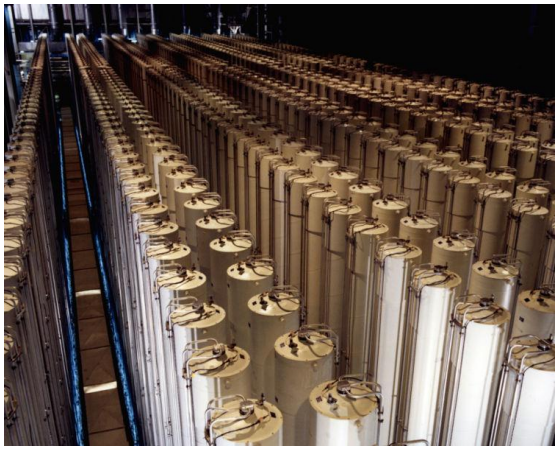

Fig. 20: Cascades of gas centrifuges in a US fuel enrichment plant in Piketon, Ohio.

APPENDIX C
Bushehr nuclear reactor technical data

The Bushehr nuclear reactor is a Russian technology-based VVER-1000 model[113] (pressurised, light-water-cooled and -moderated reactor), expressely adapted for the Iranian site.
The VVER-1000 design was developed between 1975 and 1985, based on the requirements of a new Soviet nuclear standard that incorporated some international practices, particularly in the area of plant safety. This type of reactor meets most modern safety standards (it has an emergency core-cooling system and a containment building), but also has some deficiencies.
Although the original design of VVER-1000 reactors has some shortcomings (notoriously its fire protection and electronic control-and-protection systems), the Bushehr reactor has been constructed and will operate under full International Atomic Energy Agency (IAEA) safeguards.

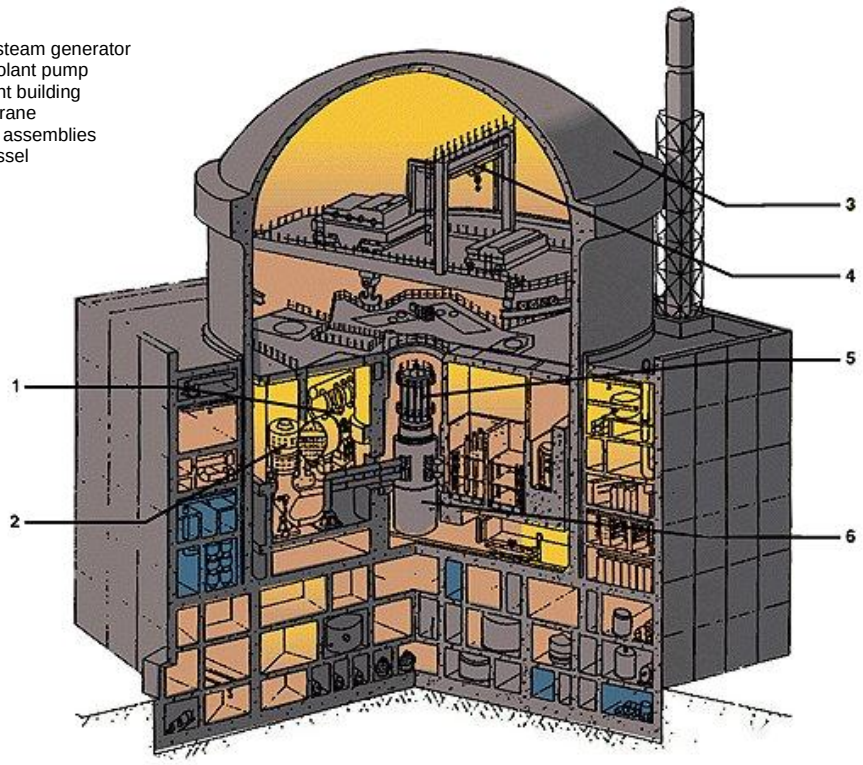

Fig. 21: The Russian VVER-1000 Pressurised Light Water reactor plant layout.

Principal strengths

- Unlike Russian nuclear power plants, which use a reinforced concrete structure (steel-lined, pre-stressed, large-volume concrete containment structure) lined with steel sheets to provide the *leak-tight enclosure system* (LES), Bushehr's LES comprises a protective shell made of a steel sphere 56 metres in diameter. This difference between Russian VVERs and Bushehr is part of the legacy of the Iranian reactor's inception as a German-designed plant on which work started in the 1970s. Russian contractors undertook to complete the first unit at the plant as a VVER-1000 reactor in 1994, some 15 years after work on the original Siemens KWU (Kraftwerk Union AG) plant had been abandoned, integrating their Russian reactor design with existing infrastructure.

[113] The VVER acronym comes from Russian *Водо-водяной энергетический реактор*, transliterated as ***V**odo-**V**odyanoi **E**nergetichesky **R**eactor*; and translatable as "Water-Water Energetic Reactor". The western denomination for such a type of nuclear reactor is *Pressurised Water Reactor* (PWR).

- Evolutionary design incorporating safety improvements over VVER-440 Model V213 plants. The Soviet approach to standardisation was based on the continued use of components that had performed well in earlier plants.
- Use of four coolant loops and horizontal steam generators (both considered improvements by Soviet designers).
- Redesigned fuel assemblies that allow better flow of coolant, and improved control rods.
- Plant worker radiation levels reportedly lower than in many western plants, apparently due to selection of materials, high-capacity system for purifying primary coolant and water-chemistry control.

Principal deficiencies (in original VVER-1000)

- Substandard plant instrumentation and controls. Wiring of emergency electrical system and reactor-protection system does not meet western standards for separation (control and safety functions are inter-connected in ways that may allow failure of a control system to prevent operation of a safety system).
- Fire-protection systems do not appear to differ substantially from earlier VVER models, which do not meet western standards.
- Quality-control, design and construction significantly deficient by western standards.
- Protection measures for control-room operators essentially unchanged from earlier VVER-440 Model V213 design, which does not meet western standards. Unlike all US nuclear plants, and most in western countries, VVER-1000s have no on-site "technical support center" to serve as a command post for stabilising the plant in an emergency.
  Technical support centres were incorporated in US and many western nuclear plants following the accident at Three Mile Island Unit 2 in 1979.
- Operating and emergency procedures fall far short of western standards and vary greatly among operators of VVER-1000 plants.
- Higher power densities and smaller volume of primary and secondary systems result in a somewhat less forgiving and stable reactor.

Current operational status of Bushehr's reactor

According to AtomStroyExport, the plant's primary circuit was hydraulically tested up to 250 kg/cm^2, 40% above normal operating pressure. The test confirmed that the main and auxiliary equipment in the two circuits are functioning efficiently within plant design parameters.
Hydraulic testing of the secondary circuit at the Bushehr reactor was completed in January 2010.

The initial 90 tons (82 metric tonnes) of nuclear fuel was delivered by Russia in eight shipments between 2007 and 2008, sufficient for one year's consumption.
On 21 August, 2010, the nuclear fuel was loaded into the reactor core.
In February 2011 the reactor operation was suspended because Russian specialists spotted faults in a pump unit of the cooling system, and Rosatom insisted the fuel be removed from the reactor and the reactor inspected for possible damage.
In April 2011 Atomstroyexport began to re-load fuel rods into the reactor after cleaning its interior and the plant's main circulation pipeline.
On 8 May, 2011, the first self-initiated nuclear fission took place in the reactor, and the reactor was run at minimum capacity for a few days.
On May 4, 8, and 11, 2011, the second shipment of uranium (30 tons) was delivered to Iran by Russia.
On 3 September, 2011, Iran claimed that the Bushehr power plant was fully operational and that it was stably connected to the Iranian power grid.

The Bushehr reactor was originally scheduled to begin operation in late 2006 but the project was beset with several delays. Iran had planned for the plant to be fully operational by the end of June 2011.
The Bushehr nuclear power plant will be operated by a Russian-Iranian joint venture during its first year of operation.

Background

The US and other western countries urged Russia for years to abandon the project, fearing that it could help Tehran to develop nuclear weapons.
Those fears were allayed in September 2006 by a deal committing Iran to use Russian-supplied nuclear fuel that will be returned to Russia after use.
As a consequence of this agreement Iran should not need to process uranium and, therefore, there are ongoing international concerns over current Iran's uranium enrichment programme.
Iran is now under UN nuclear watchdog investigation and UN sanctions over its defiance of international demands to halt uranium enrichment and do more to back up its claim that its nuclear programme is purely peaceful.

(Sources: www.world-nuclear-news.org, insp.pnnl.gov)

APPENDIX D
Natanz fuel enrichment plant (FEP)

Geographic position:	33°43'24.43" N, 51° 43' 37.55" E (about 30 km NNW of the town of Natanz, and about 20 Km SE of Kashan Air Base)
Location name:	Chaleh Qareh, Natanz, Ishafan province, Iran

The Natanz Iranian nuclear facility is a hardened Fuel Enrichment Plant (FEP) extending on a surface of approximately 100,000 square metres that is built eight metres underground and protected by a concrete wall 2.5 metres thick, itself protected by another concrete wall.
In 2004, the roof was hardened with reinforced concrete and covered with 22 metres of earth. The complex consists of two 25,000 square metre halls and a number of administrative buildings.
This once secret site was one of two that were revealed in August 2002 by Alireza Jafarzadeh, an active dissident to the Iranian government.
The underground buildings are estimated to be sized to hold over 50,000 centrifuges (this is the originally stated number of centrifuges that Natanz should be capable to hold).
It is currently believed that there are approximately 8,000 IR-1 installed centrifuges at Natanz, of which about 6,000 are producing low enriched uranium.

(Source: Wikipedia)

Fig. 22: Satellite view of the Natanz fuel enrichment plant. Enrichment centrifuges are housed in underground structures, which in the above picture are located in the areas indicated by the red rectangles.

APPENDIX E
List of Stuxnet's files and registry keys

Path\File	Hash (MD5)
~wtr4132.tmp	74ddc49a7c121a61b8d06c03f92d0c13
~wtr4141.tmp	055a3421813caf77e1387ff77b2e2e28
%windir%\system32\drivers\mrxcls.sys	f8153747bae8b4ae48837ee17172151e
%windir%\system32\drivers\mrxnet.sys	cc1db5360109de3b857654297d262ca1
%windir%\inf\mdmcpq3.pnf	variable (configuration data)
%windir%\inf\mdmeric3.pnf	b834ebeb777ea07fb6aab6bf35cdf07f
%windir%\inf\oem6c.pnf	variable (log data)
%windir%\inf\oem7a.pnf	ad19fbaa55e8ad585a97bbcddcde59d4
%windir%\system32\s7otbxdx.dll	7a4e2d2638a454442efb95f23df391a1

Tab. 8: Hashes of Stuxnet files

N.B.:
- The hash fingerprints of ~wtr4132.tmp and ~wtr4141.tmp may vary, depending on the malware version. Hashes listed above are the most common found in the wild.
- %windir% is a Windows system environment variable which contains the full path spec to the Windows operating system directory

File	Size (B)	Function
~wtr4132.tmp	513536	Dropper (main wrapper file, contains all resources needed by the worm)
~wtr4141.tmp	25720	User-mode rootkit
mdmcpq3.pnf	variable	Configuration data (size varies)
mdmeric3.pnf	90	Configuration data
mrxcls.sys	26616	Kernel-mode loader
mrxnet.sys	17400	Kernel-mode rootkit
oem6c.pnf	323848	Log file (encrypted, logs operations, infected projects, and other)
oem7a.pnf	498176	Main payload (encrypted DLL)
s7otbxdx.dll	298000	Simatic Manager DLL replacement

Tab. 9: Size and function of Stuxnet files

Upon its first execution, Stuxnet creates the following six registry keys:

```
HKLM\System\CurrentControlSet\Enum\Root\LEGACY_MRXCLS
HKLM\System\CurrentControlSet\Enum\Root\LEGACY_MRXCLS\0000
HKLM\System\CurrentControlSet\Enum\Root\LEGACY_MRXNET
HKLM\System\CurrentControlSet\Enum\Root\LEGACY_MRXNET\0000
HKLM\System\CurrentControlSet\Services\MRxCls
HKLM\System\CurrentControlSet\Services\MRxNet
```

APPENDIX F
STUXNET CHRONOLOGY – TIMELINE OF SIGNIFICANT EVENTS

Technical aspects		*Geopolitical aspects*
24 June 2012 – Stuxnet's self-kill date	< 2012 <	**12 Feb 2012** – Two Israeli diplomats are targeted in two car bomb attacks in New Delhi and in Tbilisi. Israel's PM accuses Iran of retaliation for the Iranian scientists killed.
		29 January 2012 – The Pentagon declares that Iran will probably be able to build a nuclear weapon within one year
		11 January 2012 – Iranian nuclear scientist Mostafa Roshani is killed in a car bomb attack
December 2011 – Kaspersky formulate the "Tilded Platform" theory	< 2011 <	**11 December 2011** – An explosion in a steel factory (suspected covert nuclear site) in Yazd (Iran) kills seven
		4 Dec 2011 – Iran captures a CIA drone flying over northeastern Iran, probably through an EW / cyber attack
		28 November 2011 – An explosion detonates in in the Iranian uranium conversion facility of Isfahan
13 Nov 2011 – Iran says it is being targeted by the Duqu malware		**12 November 2011** – A huge explosion kills 17 at the Iranian military Bid Ganeh base
		8 Nov 2011 – IAEA reports that Iran carried out significant activities aimed at developing nuclear explosive devices
		31 October 2011 – Israeli Prime Minister Benjamin Netanyahu addresses Knesset to support a military attack on Iran over its nuclear programme
18 Oct 2011 – Symantec reveals the existence of *Duqu* malware		**3 Sep 2011 –** Iran announces that the Busher nuclear power plant is fully operational and stably connected to the Iranian power grid
		23 July 2011 – Iranian nuclear scientist Darioush Rezaeinejad is shot to death
01 June 2011 – Expiration date for using the Print Spooler vulnerability		
25 Apr 2011 – Iran claims discovery of a new espionage virus on its computer systems, dubbed "*Stars*"		**18 Apr 2011** – Iran openly accuses Siemens to have helped the US and Israel in developing Stuxnet
3 March 2011 – ISA99 Committee launches Cyber Threat Gap Analysis Task Group on Stuxnet		
13 Feb 2011- Anonymous discloses Stuxnet source code, exfiltrated from the website of security company HBGary Federal		
		15 Jan 2011 – An article in the New York Times theorises US and Israel responsibility in developing Stuxnet
22 Dec 2010 – ISIS reports that Stuxnet has damaged more than 1000 enrichement gas centrifuges in Natanz	< 2010 <	**29 Nov 2010** – Iranian nuclear scientist Majid Shahriari is killed and Iranian nuclear scientist Fereydoun Abasi Davani is injured in two different car bomb attacks
		29 Nov 2010 - Iranian President Mahmoud Ahmadinejad for the first time admits that malware has damaged centrifuges of Iran's uranium enrichment facilities
		14 Nov 2010 - Langner points out that a coalition of nation states appears to be behind Stuxnet, limiting the names of suspects to Israel, USA, Germany, Russia
1 Oct 2010 – According to China's state news agency Xinhua, about 1000 enterprises and more than six million individual computers have been infected by Stuxnet		
September 2010 – Exploits using Stuxnet's vulnerabilities are sold on the black market for €10,000 each		
17 July 2010 – ESET identifies a new version of Stuxnet using a counterfeit digital certificate from JMicron		
16 July 2010 – Verisign revokes the Realtek Semiconductor digital certificate used to sign the Stuxnet rootkit files		
17 June 2010 – VirusBlokAda discovers Stuxnet		
March 2010 – Second wave of Stuxnet's attacks. First Stuxnet variant to exploit .LNK vulnerability (MS10-046).		
22 June 2009 – First wave of Stuxnet's attacks targeting five organisations inside Iran	< 2009 <	
April 2009 – Magazine Hakin9 publishes information about the Print Spooler Vulnerability (later identified as MS10-061)		
		January 2009 – President Bush authorises covert operations against the Iranian nuclear program
24 Dec 2008 – Domain registration of Stuxnet's C&C server	< 2008 <	
20 November 2008 – first use of the LNK vulnerability, by Trojan.Zlob		**1 June 2008** – Siemens and Idaho National Laboratory start testing PCS7 equipments for vulnerabilities
		January 2008 – Israel requests US approval for military conventional attack against Iran

Fig. 23: Timeline of Stuxnet-related events

Glossary

0-day	See: → "*Zero-day*"
802.1x	An IEEE security standard ruling the physical access of devices to networks.
AG	(from the German: "Automatisierungsgerät", automation device) See: →"*PLC*". A.k.a. "*SPS*" (Speicher-Programmierbare Steuerung)
AIEA	("*Agéncie Internationale pour l'Energie Atomique*") See: → "*IAEA*"
API	Application Program Interface. A specialised portion of code whose purpose is to give programs or users a simplified way to request functions to another (mostly lower level) executable program. In Windows, system APIs are available in the form of DLLs which are used by programs to obtain specific functionalities from the operating system, and are grouped in three API main groups: *kernel*, *GDI* and *user*. "Hooking" of these DLLs (i.e. intercepting their workflow) is a typical malware behaviour.
APT	Advanced Persistent Threat. A term commonly used in cyber threats, and particularly in cyber espionage, to designate malware developed by well organised groups (such as intelligence agencies of nation-states) with the capability and the intent to persistently and effectively target a specific entity.
ANSI/ISA-99.02.01	("*Security for Industrial Automation and Control Systems: Establishing an Industrial Automation and Control Systems Security Program*"). A security standard developed by ISA (International Society of Automation) and approved by ANSI (American National Standards Institute), which describes the elements contained in a cyber security management system for use in the industrial automation and control systems environment and provides guidance on how to meet the requirements described for each element.
autorun.inf	A file found in root directories of Windows drives which is automatically read by the operating system when the drive is mounted. It normally contains a link to an executable file, which is also executed by the operating system.
Backdoor	A hidden access to a system. Differs from rootkit because it can be implemented consciously and for legitimate purposes and because it does not try to hide its operations.
BID	Bugtraq Identifier. A unique number used for identifying vulnerabilities listed in the Bugtraq security mailing list. Alternative to CVE.
BSD	Berkeley Software Distribution.
BSOD	Blue Screen of Death. A fatal error (officially known as a *Stop Error* or a *Bug Check error*) which halts irrecoverably the execution of Windows, most commonly due to hardware faults or device driver conflicts. It is so called because of the error message appearing on a blue background.
BWR	Boiling Water Reactor. A type of nuclear reactor developed by the Idaho National Laboratory and General Electric in the mid-1950s, in which the water heated by the nuclear core turns into steam and directly drives a steam turbine. BWRs are considered less secure than →*PWR*s.
C&C	Command-and-Control. A server to which malware software connects to report its operations and receive commands, executable code, or updates.
CRL	Certificate Revocation List. An online, real-time updated list of digital certificates that have been revoked by a Certification Authority. This list should be always checked by any entity relying on PKI authentication and signing provided by that Certification Authority.

CVE	Common Vulnerabilities and Exposures. A worldwide-known conventional code used for uniquely identifying a vulnerability and its associated exploits. The CVE database is maintained by MITRE under the funding of US DHS.
Digital certificate	A file that contains a public cryptographic key through which authentication and asymmetric cryptography can be done.
DHS	Department of Homeland Security. A Cabinet department of the US federal government responsible for identifying and countering internal threats and domestic emergencies.
DLL	Dynamic Link Library. A file containing executable code, data, and resources (in any combination) which is dynamically loaded into memory and executed by Windows upon need. A DLL is organised in "sections", among which the most important are the following: `.text` (executable code), `.rdata` (read-only data), `.data` (initialised data), `.idata` (table of the imported symbols), `.reloc` (relocations table), and `.edata` (exported functions list).
DLL injection	A technique used to run code within the address space of a process by forcing it to load a dynamic-link library.
Dropper	A malware component designed to install some sort of malware (virus, backdoor, etc.) to a target system. The malware code can be contained within the dropper (single-stage) or the dropper may download the malware to the target machine once activated (two-stage). After installing the final malware the dropper usually terminates its execution as its primary function has been accomplished. See also: →"*Trojan downloader*".
Exploit	Executable code which use a specific vulnerability, expressly written for gaining unauthorised access or permissions in a system.
Export	A function (a.k.a. "*export function*") contained in a DLL that can be used by external programs through the invocation of its entry point name (often with parameter passing) for obtaining specific functionalities. Normally a DLL contains several export functions whose names are listed in the "`.edata`" section.
FAS	Federation of American Scientists
Field PG	Field Programming Gateway. A portable personal computer used for programming a PLC at its installation location. See also: → "*PG*".
FEP	Fuel Enrichment Plant. A facility which enriches uranium in its fissile isotope.
FIPS 140-2	Federal Information Processing Standard Publication 140-2 ("*Security Requirements for Cryptographic Modules*"). A US government document prepared by the National Institute for Standards and Technology (NIST) that recommends optimal security standards for implementing secure strong encryption, which can be used for protecting *sensitive but unclassified* (SBU) data. Initial publication was released on 25 May 2001, and last updated on 3 December 2002.
HAL	Hardware Abstraction Layer. The kernel component of an operating system that provides software interfaces to programs for accessing the hardware in a simplified manner.
HEU	Highly Enriched Uranium. Uranium that has a concentration of U^{235} or U^{233} greater than 20%. Uranium enriched over 20% is suitable for construction of nuclear weapons, even if "weapons-grade" HEU generally refers to uranium enriched to at least 90%.
Historian	A software that logs and archives data, making them ready for enquiries when needed. It is a common function of a SCADA system.
HMI	Human-Machine Interface. A system (device and/or software) which presents process data to a human operator, and through which the human operator controls

	and manages the process. Often incorrectly used as a synonym of SCADA.
HSM	Hardware Security Module. A physical device that generates, stores securely, and manages digital cryptographic (mostly asymetric) keys, accelerating cryptography processes and providing strong authentication to access critical keys. HSMs basically provide both logical and physical protection of high-value cryptographic keys (especially private keys) from non-authorised use and/or potential attacks. The goals of a HSM are: (a) onboard secure key generation, (b) onboard secure storage, (c) secure use of the cryptographic keys, (d) offloading application servers for complete asymmetric and symmetric cryptography. HSM modules come in the form of a plug-in card or an external networking device that can be attached directly to the server or general purpose computer. The widely accepted security standard for the definition of operations of HSM devices is the → FIPS 140-2.
IAEA	(International Atomic Energy Agency) A United Nations agency set up in 1957 (as the world's "Atoms for Peace" organisation) with the purpose of promoting cooperation and control in the field of peaceful use of nuclear energy.
ICS	Industrial Control System. A a general term that encompasses several types of control systems used in industrial production, including SCADA systems, distributed control systems (DCS), and programmable logic controllers (PLC), often found in the industrial sectors and critical infrastructures.
ICS-CERT	Industrial Control Systems Cyber Emergency Response Team. A US security CERT which deals with ICS security, managed and operated by the DHS Control Systems Security Programme (CSSP). Located at at Idaho National Labs, Idaho Falls.
IEC-62443	IEC-62443 (Security for Industrial Process Measurement and Control: Network and System Security). A series of ICS security related standard published by the International Electrotechnical Commission, strictly linked with ISA99 standards. It is considered the leading generic control system security standard series. See also → "*IEC 62443-2-1:2010*".
IEC-62443-2-1:2010	IEC 62443-2-1:2010(E) ("*Industrial communication networks – Network and system security – Part 2-1: Establishing an industrial automation and control system security program* ") is a standard published by the International Electrotechnical Commission which defines the elements necessary to establish a cyber security management system (CSMS) for industrial automation and control systems (IACS) and provides guidance on how to develop those elements. The elements of the CSMS described in this standard are mostly policy, procedure, practice and personnel-related, describing what shall or should be included in the final CSMS for the organisation.
IDS	Intrusion Detection System. A signature-based or behaviour-based system devoted to identifying incoming cyber attacks or threats.
Impersonation	A security concept unique to Windows NT, that allows a server application to temporarily "be" the client in terms of access to secure objects.
IR-1	The codename given to the Iranian enrichment gas centrifuges operating in the Natanz FEP. The IR-1 is a home-made Iranian centrifuge based on the Pakistani centrifuge called the P1, which was in its turn derived from a Dutch design.
ISIS	Institute for Science and International Security. A non-profit institution, founded in 1993, dedicated to informing the public about science and policy issues affecting international security, with particular focus on nuclear proliferation. It is currently led by founder and former United Nations IAEA nuclear inspector David Albright. (website: http://isis-online.org/)
Kernel-mode	A CPU execution mode in which the code being run has complete and unrestricted access to the underlying hardware. Kernel mode can execute any CPU instruction and reference any memory address, and is generally reserved for the lowest-level, most trusted functions of the operating system. Crashes in kernel mode are

	catastrophic and will completely halt the machine. Also called "*Ring 0*", "*system*", "*supervisor*", or "*privileged*" mode. Contrasts with → *User mode.*
Kylin	An operating system developed by the Chinese National University of Defense Technology, based on Mach and FreeBSD and approved for use in key servers by the Chinese Army. It is believed that the deployment of Kylin was made to protect Chinese systems from offensive cyber capabilities of foreign states, given they have been designed and developed for targeting computers equipped with less secure operating systems like Windows, Linux, and UNIX.
LEU	Low Enriched Uranium. Uranium that has a concentration of U^{235} or U^{233}lower than 20%. Not suitable for building nuclear weapons. LEU for civilian nuclear reactors normally has a concentration of U^{235} between 3% and 5%.
MAC address	(Medium Access Control address) A six-byte worldwide-unique code which physically identifies a network interface. A.k.a. "hardware address".
MC7	The assembly language (bytecode) that runs on Simatic S7 PLCs. It is basically the result of the compilation of the instructions originally written in STL.
Metamorphic	Said of an executable program that has the capability of dynamic self-reshaping. Typically, a metamorphic malware automatically recodes itself at each infection to avoid signature detection by using transformation techniques such as register renaming, code permutation, code expansion, code shrinking and garbage code insertion.
MIRV	Multiple Independently targetable Reentry Vehicle. Payloads released in multiple quantity by a flying vector, each one of them capable of having a different trajectory. This feature particularly applies to ICBMs (such as the *Minuteman* missile family) which are capable of delivering multiple, independently-guided nuclear warheads.
MITM	Man-In-The-Middle. A type of network attack in which a third-party inserts itself fraudulently in a communication between two parties, and acts to each one of the two parties as the counterpart.
Mutex	A program object or executable code that negotiates *mutual exclusion* among threads. Because of its *exclusive* nature, only one instance of a mutex process can exist at the same time. Therefore, a mutex is a common way to ensure that *one and only one* instance of a same process is executed at a given time by an operating system, avoiding concurrent executions.
NAC	Network Access Control. A family of network security solutions attempting to enforce network access rules based on authenticated identities, at least for user end-stations such as laptops and desktop computers. In practice, a NAC system will isolate a device from the network if it does not comply with sufficient authentication requirements.
OB	Organisation Block. In a Simatic PLC, a block of executable instructions stored in a specific area of memory. Each OB is identified by a two or three-digit number (e.g. OB35). Some OBs are reserved for specific functionalities.
PCS 7	Process Control System 7. A Siemens PLC belonging to the Simatic M7 family.
PE	Program Executable. A file format for executable binaries (programs, object code, device drivers and DLLs), used in 32-bit and 64-bit versions of Windows operating systems.
PG	Programming Gateway (or also "Programmiergerät"). A personal computer used for programming PLCs. In the Simatic architecture, a PG can be connected to the PLC in different ways (serial interface, MPI interface, ProfiBus interface, etc.)
PG/PC	Programming Gateway/Personal Computer. See: → "*PG*"
PLC	Programmable Logic Controller. A programmable electronic board which is able to control connected devices and to receive external feedback signals. It is basically a

	simple, robust digital computer used for automation of industrial processes, such as machinery control in factories.
Polymorphic	Said of a malware capable of dynamic self-encrypting.
Process Control Network	In a SCADA/ICS, the network that connects PLCs with HMI stations and/or PG. Can be a serial (RS-449), Profibus, or Modbus networks.
Profibus	(Process Field Bus) A standard for field bus communication in automation technology, first promoted in 1989 by the German Department of Education and Research. Profibus connectors have the same mechanical interface as serial DB9 interfaces. It is estimated that five million Profibus nodes (out of a total of 30 million installed by the end of 2009) are used in the process industry.
PWR	Pressurised Water Reactor. A type of nuclear reactor in which the water sent to cool the core and moderate the fission reaction is pressurised to about 150 atm to avoid boiling, and does not directly drive a steam turbine but instead heats a steam generator which in turns drives a steam turbine. Contrasts with →*BWR* (Boiling Water Reactor). PWRs have a lower thermal efficiency than BWRs, but are believed to be more reliable and secure because of the closed design of their cooling system that prevents contamination of the turbine in the event of a leak of radioactive material inside the reactor core.
QNX	A real-time Unix-like operating system especially used in embedded systems and industrial control. QNX has a very limited size thanks to its microkernel.
RADIUS	(Remote Authentication Dial-In User Service) The *de-facto* authentication, authorisation and accounting protocol (AAA) used for accessing TCP/IP networks.
Resource	Portions of data contained in a DLL file. Contrasts with → *export* , which is executable code.
Rootkit	A software that, once installed in a computer, gives to an external attacker the power to have remote administrative access and to hide this kind of operations (e.g. by tampering with system log files).
S7-300 **S7-400**	An industrial high-end PLC belonging to the Simatic family, using the Profibus protocol to control the devices it is connected to. It is made up of different components (CPU, power supply, I/O units, etc.).
SCADA	Supervisory Control And Data Acquisition. An umbrella term for defining any type of computer system which monitors and controls industrial processes, providing notifications to human operators.
SDB	System Data Block. A memory location which contains information about how a Simatic PLC is configured. System Data Blocks are created depending on the number/ type of hardware modules that are connected to the PLC.
Shellcode	A typical payload code through which it is possible to open a remote command shell on a vulnerable computer.
Simatic	A family of industrial automation solutions developed by Siemens
Simatic Manager	The Simatic Manager engineering software is the central component of the STEP 7 environment from which all other tools can be accessed and all data belonging to an automation project can be managed. It is a graphical operative framework which allows online / offline management of S7 objects (projects, application programs, blocks, hardware stations and tools). The Simatic Manager allows the execution of the following operations: - project and library management, - invocation of STEP 7 tools, - online access to the PLC.
STEP 7	A development environment available from Siemens for generating configuration data and executable code to be uploaded into Simatic PLCs.

	STEP 7 includes different functions for all the phases of an automation project: configuration and parameterisation of the hardware, programming, testing, commissioning and maintenance, project documentation, operation and diagnostic functions.
STL	(Statement List) A low-level mnemonic programming language used to program Simatic PLCs.
Trojan	A program that performs malicious operations but that claims or appears to be safe. Normally is used to install malware and does not replicate.
Trojan-downloader	A type of trojan that, once installed on a computer, waits until a network/ Internet connection becomes available, and then downloads files from remote sites, usually via HTTP or FTP. Once downloaded, the trojan-downloader installs and runs these files on the infected computer, without the user's knowledge or authorisation. After its primary download / execution routine is completed, the trojan may also proceed to a secondary payload routine.
Troll	A forum user which interacts with other users in a provocative way with the intent of causing verbal fighting, violating netiquette and often being offensive.
UF6	Uranium Hexafluoride
UPX	Ultimate Packer for eXecutables. An open source tool which uses the ULC compression algorithm to decrease the size of executable programs.
UFD	USB Flash Drive. A removable and rewritable data storage device which consists of a flash memory provided with an integrated Universal Serial Bus (USB) interface.
User mode	A CPU excution mode in which the code being executed has no ability to directly access hardware or reference memory, but must delegate these actions to system calls. Due to the protection afforded by this sort of isolation, crashes in user mode are always recoverable. Every process started by Windows (with the exception of the System process) runs in user mode. Also called *non-privileged* or *untrusted* mode.
Virus	A program which is able to replicate itself by attaching to other programs
Worm	A program who is able to replicate itself into other computers by spreading over a network to which they are connected, normally by means of exploiting a specific vulnerability present in the target computers.
Vulnerability	A security bug in a software that can be exploited for attacks.
WinCC	A software developed by Siemens running on Windows computers for visualising and controlling the operation of Simatic PLCs and their associated devices. It is basically the software implementation of the HMI functions of a SCADA system.
WMD	Weapons of Mass Destruction
WMI	Windows Management Instrumentation. A Microsoft implementation that provides a standardised set of methods for accessing system data on networked machines. Instrumentation refers to an ability to monitor or measure the level of a sotware's performance, to diagnose errors and to write trace information.
Zero-day	A type of attack that exploits a so far unknown vulnerability for which a fix does not yet exist. A good zero-day exploit can be sold by cybercriminals in its early days of discovery for a price ranging between 10,000 and 100,000 USD, depending on its effectiveness and vulnerable operating systems.

Sources of definitions:
Wikipedia (6), International Society for Automation (1), International Electrotechnical Commission (1), F-Secure (1).
All other definitions belong to the author.

Bibliography

Agence France Presse (AFP), "*Stuxnet 'cyber superweapon' moves to China*", 30- Sep- 2010, Google hosted News (http://www.google.com/hostednews/afp/article/ALeqM5iFRHUmI2w6HaAFZq-wUNre813wcA).

Albright, David; and Christina Walrond, "*Performance of the IR-1 Centrifuge at Natanz*", 18-Oct-2011, Institute for Science and International Security (http://isis-online.org/uploads/isis-reports/documents/IR1_Centrifuge_Performance_18October2011.pdf)

Albright, David; Brannan, Paul; Stricker, Andrea; and Christina Walrond, "*Natanz Enrichment Site: Boondoggle or Part of an Atomic Bomb Production Complex?*", 21-Sep-2011, Institute for Science and International Security (http://isis-online.org/isis-reports/detail/natanz-enrichment-site-boondoggle-or-part-of-an-atomic-bomb-production-comp/)

Amano, Yukiya (IAEA, Director General), "*Implementation of the NPT safeguards agreement and relevant provisions of United Nations Security Council resolutions in the Islamic Republic of Iran*" (Doc. no. GOV/2011/65), 08-Nov-2011 (http://isis-online.org/uploads/isis-reports/documents/IAEA_Iran_8Nov2011.pdf)

American National Standard Institute / Institute for Electric and Electronic Engineers, ANSI/IEEE 1073 Medical Information Bus (MIB) data communication standard

Australian Associated Press, "*Chinese cyber attacks on BHP Billiton, Rio Tinto and Fortescue Metals Group*",19-Apr-2010 (http://www.news.com.au/breaking-news/chinese-cyber-attacks-on-bhp-billiton-rio-tinto-and-fortescue-metals-group/story-e6frfku0-1225855710032)

Baheli, Nima, "*Iran sotto attacco, le nuove frontiere della cyberwar*", 07-Oct-2010, Limes (http://temi.repubblica.it/limes/iran-sotto-attacco-le-nuove-frontiere-della-cyberwar/15341)

Barzashka, Ivanka, "*Using Enrichment Capacity to Estimate Iran's Breakout Potential*", Federation of American Scientists issue brief, 21-Jan-2011, FAS website (http://www.fas.org/pubs/_docs/IssueBrief_Jan2011_Iran.pdf)

Bell, Stephen, "*Cut-price Stuxnet successors possible: Kaspersky*" (Interview to Eugene Kaspersky, 22-Mar-2011, Melbourne), 28-Mar-2011, Computerworld (http://computerworld.co.nz/news.nsf/news/cut-price-stuxnet-successors-possible-kaspersky)

Boldewin, Frank, et al., Wilders Security forums, thread "*Rootkit.TmpHider*", 14-Jul-2010 (http://www.wilderssecurity.com/showpost.php?p=1712134&postcount=22)

Borger, Julian, "*Iran unveils use of locally mined uranium for the first time*", 05-Dec-2010, The Guardian (http://www.guardian.co.uk/world/2010/dec/05/iran-nuclear-power-domestic-uranium)

Broad, William J.; Markoff, John; and David E. Sanger, newspaper article "*Israeli Test on Worm Called Crucial in Iran Nuclear Delay*", 15-Jan-2011, The New York Times (http://nyti.ms/esygjV) [accessed 10-Jan-2012]

Bumgarner, John, "Computers as Weapons of War", May 2010*, IO Journal* Vol.2 Issue 2 (http://www.crows.org/images/stories/pdf/IOI/IO%20Journal_Vol2Iss2_0210.pdf)

Byres, E. (Tofino Security); Ginter, Andrew (Abterra Technologies); Langill, Joel (ScadaHacker.com), white paper "*How Stuxnet Spreads – A Study of Infection Paths in Best Practice Systems*", 22-Feb-2011 (http://abterra.ca/papers/How-Stuxnet-Spreads.pdf)

Carìa, Raimondo; e Cazzin, Diego, "*Stuxnet, la nuova frontiera dei disastri*", 19-Apr-2011, Limes (http://temi.repubblica.it/limes/stuxnet-la-nuova-frontiera-dei-disastri/22300)

Carr, Jeffrey (Taia Global), "*Dragons, Tigers, Pearls, and Yellowcake: Four Stuxnet Targeting Scenario*", 16-Nov-2010 (http://nanojv.files.wordpress.com/2011/03/dragons_whitepaper_updated1.pdf)

Carr, Jeffrey, article "*Did The Stuxnet Worm Kill India's INSAT-4B Satellite*?", 29-Sep-2010 (9http://www.forbes.com/sites/firewall/2010/09/29/did-the-stuxnet-worm-kill-indias-insat-4b-satellite/)

Carr, Jeffrey, blog article "*Stuxnet's Finnish-Chinese Connection*", 14-Dec-2010, The Firewall - the world of security (http://www.forbes.com/sites/firewall/2010/12/14/stuxnets-finnish-chinese-connection/)

Carsten, Köhler, "*Print Your Shell*", 01-Apr-2009, Hakin9 security magazine (http://hakin9.org/print-your-shell/)

Chapell, Geoff, *The MRXCLS.SYS Malware Loader*, 21-Oct-2010 (http://www.geoffchappell.com/notes/security/stuxnet/mrxcls.htm [accessed 21-Jan-2012])

Chen, Thomas M., "*Stuxnet, the Real Start of Cyber Warfare?*", November/December 2010, IEEE Network magazine (http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=05634434)

Chien, Eric (Symantec), "*Stuxnet: A Breakthrough*",Symantec's Connect Security Response blog (http://www.symantec.com/connect/blogs/stuxnet-breakthrough), 12-Nov-2010

Clayton, Mark, "*Stuxnet malware is 'weapon' out to destroy ... Iran's Bushehr nuclear plant?*", 21-Sep-2010, The Christian Science Monitor (http://www.csmonitor.com/USA/2010/0921/Stuxnet-malware-is-weapon-out-to-destroy-Iran-s-Bushehr-nuclear-plant)

Clendenin, Mike, "*China Gets A Peek At Microsoft Source Code*", 04-Jun-2010, InformationWeek (http://www.informationweek.com/news/software/operating_systems/225400063)

Davis, Michael A., *"Stuxnet Reality Check: Are You Prepared for a Similar Attack?*", May 2011 (http://www.savidtech.com/wp-content/themes/sti/images/stories/PDF/S2840511_DR_stuxnet.pdf [accessed 10-Mar-2012])

Davis, Michael, "*Stuxnet: How It Happened And How Your Enterprise Can Avoid Similar Attacks*", 17-May-2011, DarkReading (http://www.darkreading.com/advanced-threats/167901091/security/attacks-breaches/229500805/stuxnet-how-it-happened-and-how-your-enterprise-can-avoid-similar-attacks.html) [accessed 10-Jan-2012

Dehghan, Saeed Kamali, "*Iran accuses Siemens of helping launch Stuxnet cyber-attack*", 17-Apr-2011, The Guardian (http://www.guardian.co.uk/world/2011/apr/17/iran-siemens-stuxnet-cyber-attack)

Falliere, N., O Murchu, L., Chien, L. (Symantec), *"W32.Stuxnet Dossier (Symantec Security Response")*, 11-Feb-2011 (Version 1.4)

Finkle, Jim (Reuters), "*Insight: Did Conficker help sabotage Iran program*",08-Dec-2011, Interview of John Bumgarner (U.S. Cyber Consequences Unit) (http://www.reuters.com/article/2011/12/08/us-cybersecurity-iran-idUSTRE7B10AP20111208)

Finkle, Jim, "*Stuxnet weapon has at least 4 cousins: researchers*", 28-Dec-2011, Reuters U.S. (http://www.reuters.com/article/2011/12/28/us-cybersecurity-stuxnet-idUSTRE7BR1EV20111228)

Fisher, Dennis, "*Stuxnet Authors Made Several Basic Errors*", 18-Jan-2011,ThreatPost (https://threatpost.com/en_us/blogs/stuxnet-authors-made-several-basic-errors-011811)

Francis Allan Tan Seng & Elda Dimakiling (Microsoft Malware Protection Center), "*Protection for New Malware Families Using .LNK Vulnerability* ", 23-Jul-2010, Technet blogs (http://blogs.technet.com/b/mmpc/archive/2010/07/23/protection-for-new-malware-families-using-lnk-vulnerability.aspx)

F-Secure, Virus and Threat descriptions, Trojan-Dropper:W32/Stuxnet (http://www.f-secure.com/v-descs/trojan-dropper_w32_stuxnet.shtml [accessed 11-Mar-2012])

Ginter, Andrew, Industrial Defender - *The Stuxnet Worm and Options for Remediation*, by last updated 6-Aug-2010 (http://j-j.co.za/wp-content/uploads/2010/08/stuxnet_08.2010.pdf)

Gostev, Aleksander, "*The Mistery of Duqu: Part One*", 20-Oct-2011, Kaspersky's SecureList blog (http://www.securelist.com/en/blog/208193182/The_Mystery_of_Duqu_Part_One)

Gostev, Aleksander, Securelist, "*The Duqu Saga Continues: Enter Mr. B. Jason and TV's Dexter*", 11-Nov-2011 (https://www.securelist.com/en/blog/208193243/The_Duqu_Saga_Continues_Enter_Mr_B_Jason _and_TVs_Dexter [accessed 11-Mar-2012])

Gross, Michael Joseph, "*A Declaration of Cyber-War*", April 2011, Vanity Fair (http://www.vanityfair.com/culture/features/2011/04/stuxnet-201104) [accessed 12-Jun-2011]

Harley, David (ESET), "*Chim Chymine: A Lucky Sweep?*", 11-Sep-2010, Virus Bullettin (http://go.eset.com/us/resources/white-papers/chymine-whitepaper.pdf)

Higgins, Kelly Jackson, "*Waiting For 'Son Of Stuxnet' To Attack*", 19-Oct-2011 (http://www.darkreading.com/advanced-threats/167901091/security/attacks-breaches/231901226/waiting-for-son-of-stuxnet-to-attack.html)

Hipolito, Joahnna, Trend Micro Threat Encyclopedia, "*DUQU Uses STUXNET-Like Techniques to Conduct Information Theft*" (http://about-threats.trendmicro.com/relatedthreats.aspx?language=us &name=DUQU%20Uses%20STUXNET-Like%20Techniques%20to%20Conduct%20Information%20Theft)

Hopkins, Nick, "*Stuxnet attack forced Britain to rethink the cyber war* ", 30 May 2011, The Guardian (http://www.guardian.co.uk/politics/2011/may/30/stuxnet-attack-cyber-war-iran)

Hounshell, Blake, "*Six mysteries about Stuxnet*", 27-Apr-2010, Foreign Policy newspaper article (http://blog.foreignpolicy.com/posts/2010/09/27/6_mysteries_about_stuxnet)

INSP, "*The VVER-1000*" (nuclear reactor design), 19-Nov-1998 (http://insp.pnnl.gov/-profiles-reactors-vver1000.htm)

IRIB News, "*Ahmadinejad: Israel must be wiped off the map*", 26-Oct-2005, (http://web.archive.org/web/20070927213903/http://www.iribnews.ir/Full_en.asp?news_id=200247).

ISIS report, 22 Dec 2010 (http://www.isis-online.org/isis-reports/detail/did-stuxnet-take-out-1000-centrifuges-at-the-natanz-enrichment-plant/)

Katz, Yaakov, "*Stuxnet virus set back Iran's nuclear program by 2 years*", 15-Dec-2010, The Jerusalem Post (http://www.jpost.com/IranianThreat/News/Article.aspx?id=199475)

Keizer, Greg, *Stuxnet struck five targets in Iran, say researchers*, 11-Feb-2011, ComputerWorld magazine (http://www.computerworld.com/s/article/9209160/Stuxnet_struck_five_targets_in_Iran_say_researchers [accessed 10-Jan-2012])

Krebs, Brian, "*'Stuxnet' Worm Far More Sophisticated Than Previously Thought*", 14-Sep-2010, KrebsOnSecurity blog (http://krebsonsecurity.com/2010/09/stuxnet-worm-far-more-sophisticated-than-previously-thought/ [accessed 11-Mar-2010])

Krebs, Brian, "*Experts Warn of New Windows Shortcut Flaw*", 15-Jul-2010, KrebsOnSecurity blog (http://krebsonsecurity.com/2010/07/experts-warn-of-new-windows-shortcut-flaw/ [accessed 11-Mar-2012])

Langner, Ralph, "*Intercept, infect, infiltrate*", 22-Feb-2011, Langner blog (http://www.langner.com/en/2011/02/22/intercept-infect-infiltrate/)

Langner, Ralph, "*Stuxnet Deep Dive*" (video), 12-Jan-2012, Digital Bond (http://www.digitalbond.com/2012/01/31/langners-stuxnet-deep-dive-s4-video/)

Langner, Ralph, "*The first deployed cyber weapon in history: Stuxnet's architecture and implications* " (video), 09-Jun-2011, International Conference on Cyber Conflict (http://www.youtube.com/watch?v=n7UVyVSDSxY)

Larimer, Jon (Senior Researcher, IBM X-Force), *An inside look at Stuxnet*, 10-Nov-2010

(http://blogs.iss.net/archive/papers/ibm-xforce-an-inside-look-at-stuxnet.pdf [accessed 12-May-2011])

Lee, Melissa, "*CodeWars: America's Cyber Threat*" (video documentary), May 2011,CNBC (http://video.cnbc.com/gallery/?video=3000022244)

Leveson, Nancy, "*Medical devices: the Therac-25*", University of Washington (http://sunnyday.mit.edu/papers/therac.pdf)

Madrigal, Alexis, "*The Stuxnet Worm? More Than 30 People Built It*", 4-Nov-2010, The Atlantic (http://www.theatlantic.com/technology/archive/2010/11/the-stuxnet-worm-more-than-30-people-built-it/66156/)

Markoff, John, "*A Silent Attack, but Not a Subtle One*", 27-Sep-2010, The New York Times (http://www.nytimes.com/2010/09/27/technology/27virus.html)

Matrosov A., Rodionov E., Harley D., Malcho J., "*Stuxnet under the Microscope*" (Revision 1.31), ESET, (http://go.eset.com/us/resources/white-papers/Stuxnet_Under_the_Microscope.pdf)

McAfee, "*Virtual Criminology Report 2009 - Virtually Here: The Age of Cyber Warfare*" (http://www.mcafee.com/us/resources/reports/rp-virtual-criminology-report-2009.pdf)

Mehr News Agency (MNA), " *Iran target of new cyber attack*", 25-Apr-2011 (http://www.mehrnews.com/en/newsdetail.aspx?NewsID=1297506)

Microsoft Security TechCenter, "*MS08-67 Security Bullettin - Vulnerability in Server Service Could Allow Remote Code Execution (958644)*", 23-Oct-2008 (http://technet.microsoft.com/en-us/security/bulletin/ms08-067)

Microsoft Security TechCenter, "*MS10-046 Security Bullettin - Vulnerability in Windows Shell Could Allow Remote Code Execution (2286198*")*,* 02-Aug- 2010 (http://technet.microsoft.com/en-us/security/bulletin/MS10-046)

Microsoft Security TechCenter, "*MS10-061 Security Bullettin - Vulnerability in Print Spooler Service Could Allow Remote Code Execution (2347290)*", 14-Sep-2010 (http://technet.microsoft.com/en-us/security/bulletin/MS10-061)

Microsoft Security TechCenter, *MS10-073 Security Bullettin - Vulnerabilities in Windows Kernel-Mode Drivers Could Allow Elevation of Privilege (981957)*, 12-Oct-2010 (http://technet.microsoft.com/en-us/security/bulletin/MS10-073)

Microsoft Security TechCenter, *MS10-092 Security Bullettin - Vulnerability in Task Scheduler Could Allow Elevation of Privilege (2305420)*, 14-Dec-2010 (http://technet.microsoft.com/en-us/security/bulletin/MS10-092)

Microsoft Security TechCenter, *MS11-087 Security Bullettin - Vulnerability in Windows Kernel-Mode Drivers Could Allow Remote Code Execution (2639417)*, 13-Dec-2011 (http://technet.microsoft.com/en-us/security/bulletin/ms11-087)

Miller, Charlie, "*Kim Jong-il and me: How to build a cyber army to attack the U.S.*", CCDCOE Conference on Cyber Conflict, June 2010, Tallinn (www.ccdcoe.org/conference2010)

Moore, D., Paxson, V., Savage, S., Shannon, C., Staniford, S., Weaver, N. *The Spread of the Sapphire/Slammer Worm*, CAIDA, http://www.caida.org/publications/papers/2003/sapphire/sapphire.html

Moore, D., Paxson, V., Savage, S., Shannon C., Staniford S., Weaver N., "*Inside the Slammer Worm*", IEEE Security and Privacy, pp. 33-39, July-August 2003

Mostafavi, Ramin (Reuters)," *Iran accuses Siemens over Stuxnet virus attack*", 17-Apr-2011 ()

N3td3v, CNET News forum on article "*Details of the first-ever control system malware (FAQ)*", 20-Jul-2010 1:49 PM PDT (http://news.cnet.com/8301-27080_3-20011159-245.html)

O Murchu, Liam (Symantec), "*Stuxnet - Infecting Industrial Control Systems*" (VirusBulletin Conference 2010, Vancouver), 01-Sep-2010 (http://www.virusbtn.com/pdf/conference_slides/2010/OMurchu-VB2010.pdf [accessed 10-Jan-2012])

O Murchu, Liam (Symantec), "*Stuxnet - Modus* Operandi", 01-Mar-2011 (http://www.thei3p.org/docs/events/cybercprpresentation_stuxnet.pdf) [accessed 31-Oct-2011]

Ockham, William, "*Did Duqu fix the bug that revealed Stuxnet?*", 20-Oct-2011, EmptyWheel blog (http://www.emptywheel.net/2011/10/20/did-duqu-fix-the-bug-that-revealed-stuxnet/ [accessed 18-Feb-2012])

OWASP (The Open Web Application Security Project), "*Use of hard-coded password* ", 21-Feb-2009, OWASP website (https://www.owasp.org/index.php/Use_of_hard-coded_password)

Pinna, Lorenzo, "*SuperQuark - Stuxnet: il virus informatico di nuova generazione*"(video documentary), August 2011, RAI (http://www.rai.tv/dl/RaiTV/programmi/media/ContentItem-24b4f8f0-e236-4e86-8c9a-9a96623d1ddf.html)

Putz, Ulrike, "*Mossad Behind Tehran Assassinations, Says Source*", 08-Feb-2011, Spiegel Online International (http://www.spiegel.de/international/world/0,1518,777899,00.html [accessed 20-Feb-2012])

Ragan, Steve,"*Zeus botnet plundering the masses and snatching certificates*", 05-Aug-2010, The Tech Herald (http://www.thetechherald.com/articles/Zeus-botnet-plundering-the-masses-and-snatching-certificates)

Rieger, Frank, newspaper article "*Trojaner „stuxnet" Der digitale Erstschlag ist erfolgt*", Frankfurter Allgemeine Zeitung, 22-Sep-2010 (http://www.faz.net/aktuell/feuilleton/debatten/digitales-denken/trojaner-stuxnet-der-digitale-erstschlag-ist-erfolgt-1578889.html)

Rieger, Frank, September 2010 (http://www.bloomberg.com/news/2010-09-24/stuxnet-computer-worm-may-be-aimed-at-iran-nuclear-sites-researcher-says.html)

Saher, Mohamed & Molinyawe, Matthew (NSS Labs), "*Duqu Analysis & Detection Tool*", 03-Nov- 2011, NSS Labs website (http://www.nsslabs.com/blog/2011/11/duqu-analysis-and-detection-tool.html)

Sanger, David E., "*U.S. Rejected Aid for Israeli Raid on Iranian Nuclear Site*", 10-Jan-2009, The New York Times (http://www.nytimes.com/2009/01/11/washington/11iran.htm)

Sanger, David E., *The New York Times - U.S. Rejected Aid for Israeli Raid on Iranian Nuclear Site*, published 10-Jan-

2009 (http://www.nytimes.com/2009/01/11/washington/11iran.htm [accessed 1-Ja-2011])

Schneier, Bruce, "*Stuxnet*", 07-Oct-2010, Schneier on Security blog (http://www.schneier.com/blog/archives/2010/10/stuxnet.html)

Siemens, "*SIMATIC WinCC / SIMATIC PCS 7: Information about Malware / Viruses / Trojan horses*" ,04-Jan-2011, Siemens International Industry Online Support (http://support.automation.siemens.com/WW/llisapi.dll/43876783?func=ll&objId=43876783)

Siemens, "*Stuxnet Malware - Official communication*" (presented by Thomas Brandstetter at CIP seminar), 02-Nov-2010 (http://ciip.files.wordpress.com/2010/11/the_stuxnet_malware.pdf)

Siemens, white paper "*Security concept PCS 7 and WinCC - Basic document*" (Doc. No. 04/2008 A5E02128732-01), April 2008 (http://cache.automation.siemens.com/dnl/jE/jE2MjIwNQAA_26462131_HB/wp_sec_b.pdf)

Sparks, Jeremy, "*Duqu: father, son, or unholy ghost of Stuxnet?",* 02-Nov-2011, SC Magazine (http://www.scmagazine.com/duqu-father-son-or-unholy-ghost-of-stuxnet/article/215851/)

Stark, Holger, "*Mossad's Miracle Weapon - Stuxnet Virus Opens New Era of Cyber War* , 8-Aug-2011, Spiegel Online International (http://www.spiegel.de/international/world/0,1518,778912,00.html [accessed 18-Dec-2011])

Stephanopoulos, George, *Interview to Dmitry Medvedev*, 12-Apr-2010 (http://abcnews.go.com/GMA/transcript-george-stephanopoulos-interviews-russian-president-dmitry-medvedev/story?id=10348116 &page=8)

Symantec, "*Updated W32.Stuxnet Dossier is Available* ", 14-Feb-2011, Symantec Connect Security Response blog (http://www.symantec.com/connect/blogs/updated-w32stuxnet-dossier-available).

Zetter, Kim, "*ThreatLevel - Clues Suggest Stuxnet Virus Was Built for Subtle Nuclear Sabotage*", 15-Nov-2010 (http://www.wired.com/threatlevel/2010/11/stuxnet-clues/ [accessed 11-Mar-2012])

Tarakanov, Dmitri, "*ZeuS on the Hunt*" (http://www.securelist.com/en/analysis/204792107#2).

Thabet, Amr, *The Code Project – Stuxnet Malware Analysis Paper*, 09-Sep-2011 (http://www.codeproject.com/Articles/246545/Stuxnet-Malware-Analysis-Paper [accessed 10-Dec-2011])

The Economist, "*A worm in the centrifuge*", 30-Sep-2010 (http://www.economist.com/node/17147818)

ThreatExpert, *ThreatExpert Submission Summary* on submitted threat sample, 2-Jul-2011, http://www.threatexpert.com/report.aspx?md5=74ddc49a7c121a61b8d06c03f92d0c13 [accessed 11-Mar-2012]

Tillet, Brian (Symantec), IdentEvent 2010 (reported by The Atlantic, 4 Nov. 2010, http://www.theatlantic.com/technology/archive/2010/11/the-stuxnet-worm-more-than-30-people-built-it/66156/)

Travis, G., Balas, E., Ripley, D., Wallace, S., "*Analysis of the "SQL Slammer" worm and its effects on Indiana University and related institutions"*, February 2003 (http://paintsquirrel.ucs.indiana.edu/pdf/SLAMMER.pdf)

Ulasen, S., and Kupreev, O., "*Trojan-Spy.0485 And Malware-Cryptor.Win32.Inject.gen.2 Overview*" ("Обзор вредоносных программ Trojan-Spy.0485 и Malware-Cryptor.Win32.Inject.gen.2"), 09-Jul-2010, VirusBlokAda (ftp://anti-virus.by/pub/docs/russian/Rootkit.TmpHider.pdf)

United Nations Security Council, *Resolution 1737 (2006)*, 27-Dec-2006 (http://www.iaea.org/newscenter/focus/iaeairan/unsc_res1737-2006.pdf)

United Press International, "*Enter Unit 8200: Israel arms for cyberwar*", 11-May-2011, UPI (http://www.upi.com/Top_News/Special/2011/05/11/Enter-Unit-8200-Israel-arms-for-cyberwar/UPI-93881305142086/)

Warrell, Andrew, "*Miners under attack 'from everywhere*'", 30-May-2011, The Australian (http://www.theaustralian.com.au/business/mining-energy/miners-under-cyber-attack-from-everywhere/story-e6frg9df-1226065199596)

West, Ben, "*Dirty tricks and sticky bombs in Iran*", 20-Dec-2010, Asia Times Online (http://www.atimes.com/atimes/Middle_East/LL04Ak01.html)

Weymouth, Lally (Washington Post), "*Iran Still Steadily Producing Uranium*", interview with Yukiya Amano, 14-Feb-2011, IAEA (http://www.iaea.org/newscenter/transcripts/2011/wp140211.html)

Vick, Karl, "*Was Israel Behind a Deadly Explosion at an Iranian Missile Base?*", 13-Nov- 2011, Time (http://www.time.com/time/world/article/0,8599,2099376,00.html)

Wikipedia, Natanz fuel enrichment plant data (http://en.wikipedia.org/wiki/Nuclear_facilities_in_Iran#Natanz)

Wilders Security Forums, "Rootkit.TmpHider", Page 7 of 16, 20-Jul-2010

(http://www.wilderssecurity.com/showthread.php?p=1716086#post1716086)

Williams, Dan (Reuters), "*Wary of naked force, Israelis eye cyberwar on Iran*", 7-Jul-2009 (http://www.reuters.com/article/2009/07/07/us-israel-iran-cyberwar-analysis-idUSTRE5663EC20090707)

von Eitzen, Chris, "*Duqu exploits previously unknown vulnerability in Windows kernel*", The H Security, 02-Nov-2011 (http://www.h-online.com/security/news/item/Duqu-exploits-previously-unknown-vulnerability-in-Windows-kernel-1370369.html)

World Nuclear Association, Market Report data (http://www.world-nuclear.org/info/inf23.html)

World Nuclear News, Bushehr related news (http://www.world-nuclear-news.org/results.aspx?sparam=bushehr)

Yannakogeorgos, Panayotis A., "*Was Russia behind Stuxnet?*", 10-Dec-2011, The Diplomat (http://the-diplomat.com/2011/12/10/was-russia-behind-stuxnet/2/)

Index

N.B.: please also check the glossary for specific term definitions.