The Stuxnet Worm

Paul Mueller and Babak Yadegari

1 Overview of Stuxnet

Stuxnet is a sophisticated worm designed to target only specific Siemens SCADA (industrial control) systems.

It makes use of an unprecedented four 0-day vulnerabilities- attacks that make use of a security vulnerability in an application, before the vulnerability is known to the application's developers.

It also uses rootkits - advanced techniques to hide itself from users and anti-malware software - on both Windows and the control computers it targets. It employs two stolen digital certificates to sign its drivers, and its creators needed a large amount of knowledge of its targeted systems. See Figure 1 for an overview.

It was discovered in June 2010, but an early version appeared a year earlier. It is widely suspected of targeting Iran's uranium enrichment program, since it is rather specific about what it attacks, and this matches the Iranian Natanz enrichment plant.

One indication that Stuxnet targeted Iran's nuclear program is that it only targets facilities that have a certain specific physical layout. The layout of the centrifuges in a facility such as Natanz is called a cascade, and describes how the centrifuges are piped together; this is done in stages, and the centrifuges within one stage are piped in parallel.

Iranian President Ahmadinejad visited Natanz on April 8, 2008, and photos of the visit were published on his website as a photo-op. In one of the photos (Figure 2) is what appears to be a monitor showing the structure of a so-called IR-1 (for Iran-1 centrifuge) cascade. This structure, giving the number of centrifuges in each stage, matches the Stuxnet code exactly.

Iran's nuclear program launched in the 1950s with the Shah of Iran obtaining non weapons-related assistance from the United States' "Atoms for Peace" program. The program's inception was delayed because of the 1979 revolution and after that because of the Iran-Iraq war. However, Iran's new leaders were interested in continuing the nuclear program and started getting help from other countries to further it.

In 2002, an Iranian opposition group publicly revealed two undeclared nuclear facilities, resulting in Iran admitting to having constructed facilities for fuel enrichment and heavy water production, ostensibly for use in research reactors. Iran suspended its plans in 2003 but resumed them in 2006 and insists that it has the right to establish its own uranium enrichment program.

Iran maintains that its nuclear program is entirely peaceful. However, the IAEA (International Atomic Energy Agency) insists that Iran does not comply with the safeguard program it has agreed to, resulting in various sanctions against Iran by the UN Security Council [9]. It is widely believed that Iran is in fact working toward producing nuclear weapons.

Assuming that Stuxnet was intended to damage this suspected nuclear weapons program, it was somewhat effective: it may have destroyed 1,000 centrifuges at Natanz, about 11% of the total number installed at the time. Also, Iran doesn't have an unlimited number of centrifuges, and the ones they do have tend to fail relatively often, so such a decrease is significant, albeit not

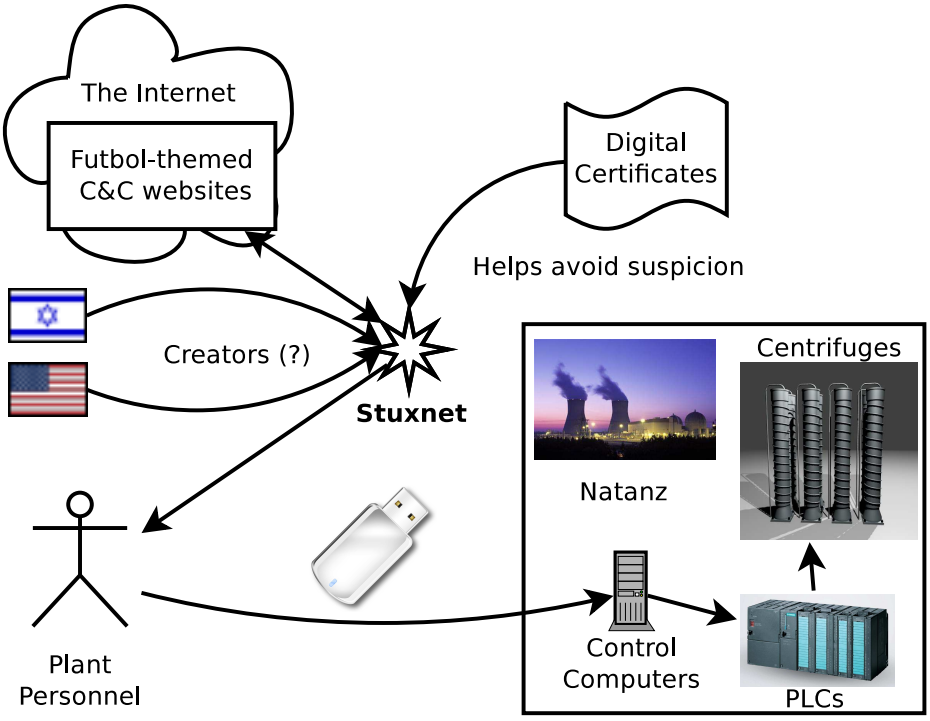

Figure 1: A high-level overview of Stuxnet.

Figure 2: Iran's president revealed the cascade structure at Natanz: The dark lines at the bottom probably denote the division between stages; yellow lines have been added above them to make the division clearer. from right to left- 4, 8, 12, 16, 20, 24, 20, 16. (Photo: Office of the Presidency of the Islamic Republic of Iran)

immediately fatal to the program. In addition, Stuxnet decreased production of enriched uranium and likely sowed chaos within the Iranian nuclear program.

Israel and the U.S. are the leading suspects as Stuxnet's creators. Neither are friends of Iran's current leadership (to put it mildly) and both, especially Israel, fear a nuclear-armed Iran.

Israel has said that cyberwarfare is an important part of its defense strategy, and has a military intelligence unit dedicated to it, according to Wikipedia. Furthermore, Israeli officials are said to have responded with "wide smiles" when they were asked in 2010 whether Israel created Stuxnet [5].

Furthermore, a video celebrating the operational successes of the head of the Israeli Defence Forces' Lieutenant General Gabi Ashkenazi, shown at his retirement party, showed Stuxnet as being among them, according to the Daily Telegraph [15].

Before Stuxnet had been discovered, according to Wikipedia, "John Bumgarner, a former intelligence officer and member of the United States Cyber-Consequences Unit (US-CCU)" had published an article describing a Stuxnet-like attack on centrifuges, and claimed that such attacks against nations enriching uranium in violation of international treaties are legitimate. This, combined with some US officials' not-quite-denials of involvement, raise suspicions that the US participated in Stuxnet's creation, despite official denials.

2 Operation of Stuxnet

One thing that differentiates Stuxnet from more run-of-the-mill malicious software is that its creators have incorporated lots of capabilities into it. These range from exploiting multiple zero-day vulnerabilities, modifying system libraries, attacking Step7 installations (Siemens' SCADA control software) and running an RPC server, to installing signed drivers on Windows operating systems. Figure 3 shows an overview of multiple components which are present in the malware. This section describes these components and their various purposes.

Stuxnet spreads via several vectors, no doubt selected to ultimately allow it to infect the PLCs it targets. It is capable of auto-updating, so that it can update old versions of itself to newer versions available on a local network. It communicates with command and control servers to provide information on its spread to its creators and also provide another way for it to be updated. It conceals its presence and the source of its destructive effects from plant personnel, who may be totally unaware that it is the cause of unexplained problems.

2.1 How Stuxnet Spreads

Stuxnet spreads readily, but it also contains safeguards to limit its spread. It only infects three computers from a given infected flash drive and is hardcoded to stop spreading itself after June 24, 2012.

As illustrated in Figure 4, Stuxnet employs several methods to spread itself:

Via USB flash drives The ultimate destination of Stuxnet is the computers that control the centrifuges. These are called PLCs (Programmable Logic Controllers), and are special-purpose computers, used for controlling electronic devices or systems, such as industrial systems.

The PLCs are connected to computers that control and monitor them, and typically, neither are connected to the Internet. Therefore, Stuxnet needs some other vector to reach those computers, and so it is capable of propagating via USB flash drives.

In the case of Natanz, the infected flash drives may have been introduced to the control computers via outside contractors working at the plant.

Different versions of Stuxnet use different ways to do this: recent versions use an Windows LNK vulnerability and older versions use an `autorun.inf` file vulnerability.

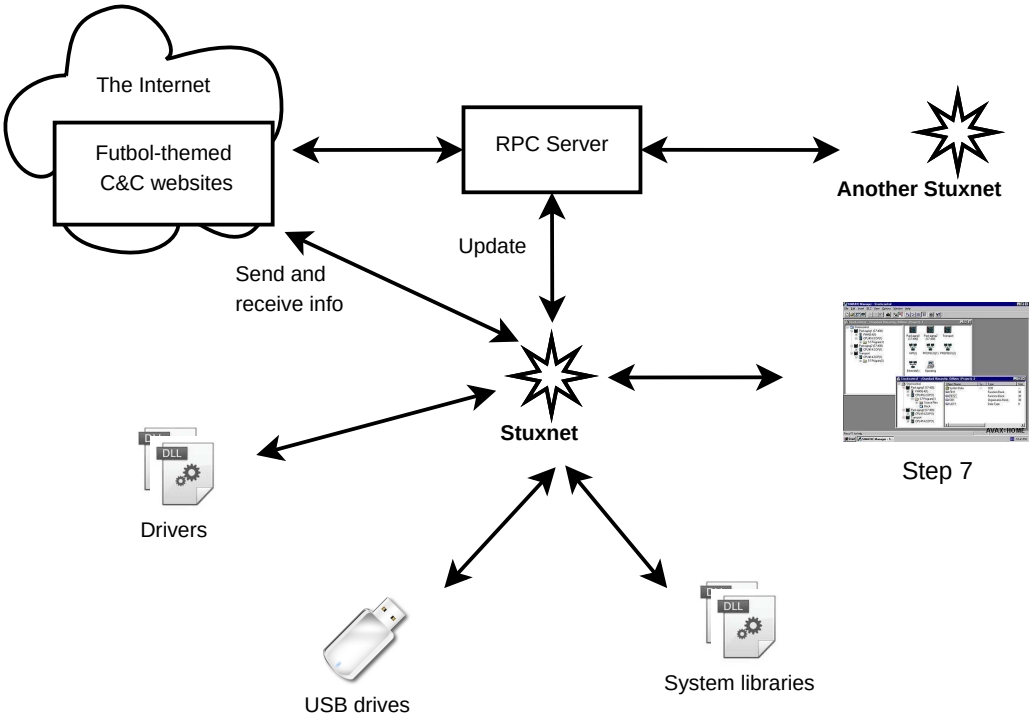

Figure 3: Stuxnet's various components

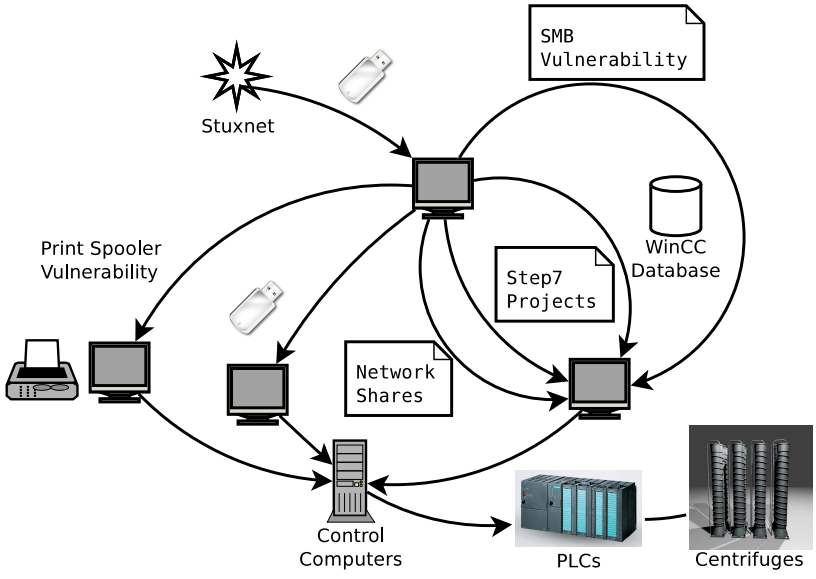

Figure 4: Stuxnet employs many ways to reach its target PLCs.

- LNK vulnerability (CVE-2010-2568)
  Stuxnet registers code to an infected Windows computer that, upon a USB drive being inserted, copies Stuxnet to the drive. Interestingly, an existing copy of Stuxnet on the external drive will be removed if that drive has already infected three computers.

  In addition to the Stuxnet DLL and a loader for it, the malware creates four `.lnk` files on the removable drive. These are used to execute the loader when a user views the drive; four are needed in order to target different versions of Windows.

- `autorun.inf` file
  An `autorun.inf` file is a file that causes Windows to automatically run a file on removable media when that media is inserted into the computer. Older versions of Stuxnet place an `autorun.inf` file on flash drives that are inserted into an infected computer.

  However, instead of using a separate file, it inserts the code for itself directly into the `autorun` file, along with valid commands to infect the computer using this code. Windows ignores the Stuxnet data portion, since it ignores invalid commands in an `autorun.inf` file.

Via WinCC Stuxnet searches for computers running Siemens WinCC, an interface to their SCADA systems. It connects using a password hardcoded into WinCC, and attacks its database using SQL commands to upload and start a copy of itself on the WinCC computer.

Via network shares Stuxnet can use Windows shared folders to propagate itself over a local network. It places a dropper file on any shares on remote computers, and schedules a task to execute it. ESET [11] says the task is scheduled to run the next day, whereas Symantec [7] claims it is scheduled for two minutes after the file is shared.

Via the MS10-061 print spooler 0-day vulnerability Stuxnet copies itself, places the copy on remote computers via this vulnerability, and then executes the copy, thereby infecting the remote machines. In brief, Stuxnet "prints" itself to two files in the `%system%` directory on each target machine, using the 0-day privilege escalation. It then executes the dropper file to infect the computer.

Via the MS08-067 SMB vulnerability If a remote computer has this vulnerability, Stuxnet can send a malformed path over SMB (a protocol for sharing files and other resources between computers); this allows it to execute arbitrary code on the remote machine, thereby propagating itself to it.

Via Step7 Projects Stuxnet will infect Siemens SIMATIC Step7 industrial control projects that are opened on an infected computer. It does this by modifying DLLs (Windows Dynamic Link Library; a library of shared objects: code, data, and resources) and an `.exe` file in the WinCC Simatic manager, so that they execute Stuxnet code as well. The additional code will insert Stuxnet into Step7 project directories.

2.2 Auto-updating

Stuxnet can update itself from infected Step7 projects. If an infected project is opened, and its version of Stuxnet is newer than the one already on the computer, the one on the computer will be updated.

Additionally, Stuxnet uses a built-in peer-to-peer network to update old instances of itself to the latest version present on a local network. Each copy starts an RPC (Remote Procedure Call) server, and listens for connections. Other instances, connecting via their RPC clients, are able to update themselves if their version is older, or update the server if it is older.

2.3 Command and Control servers

After Stuxnet establishes itself on a computer, it tries to contact one of two servers via HTTP:

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

It sends its IP address, some unknown data, and a payload consisting of, in part, information on the host OS, the host computer name and domain name, and a flag indicating if Siemens Step7 or WinCC, which Stuxnet targets, is installed [7].

The server may respond by sending a Windows executable, which it can specify to be loaded into the current process or a different one via RPC. This allows Stuxnet's authors to update it remotely, or to run entirely new malware on the infected computers.

Interestingly, the data sent to and from the server is encrypted, each way with a different 31 byte key (of seemingly random bytes) that is XORed with the data. However, both keys are static, and so don't provide much protection against someone who has intercepted multiple such communications.

The command and control servers have since been rendered impotent by redirecting their DNS entries (initially to nowhere and later to fake servers, set up by Symantec, to gather information on the worm.)

According to Symantec, Stuxnet is able to update itself to communicate with new command and control servers, but this hasn't been observed in the field.

2.4 Infection

The malware's main module consists of both user-mode and kernel-mode components. The user-mode functions are mainly designed to do several things: 1) inject it into a chosen process - add its own code into a running process, which results in the execution of that code in the target process's address space; 2) check for the appropriate platform; 3) escalate privileges; and 4) install two kernel-mode drivers, one for running Stuxnet after reboot and the other as a rootkit to hide its files.

2.4.1 User-Mode

The main module DLL exports 21 functions (Figure 5). Stuxnet starts by calling export 15. In this function, it first checks if it is running on an appropriate Windows version. Assuming it is and the machine is not infected already, it uses one of the zero-day exploits, depending on the version of Windows, to elevate its privileges.

Export 16 is then called to continue installation: it injects code into the `services.exe` process to infect removable drives and infects any Step7 projects it finds. It then checks a registry value and aborts the infection if it matches. It also checks to ensure the current date is not later than June 24, 2012, and checks if it is running the latest version. If all these checks pass, it will drop two driver files to install the driver files, `Mrxnet.sys` and `Mrxcls.sys` [7].

It is unclear if there is any special significance to the cutoff date of June 24th; it may be arbitrary, used to stop Stuxnet's spread after it had (presumably) already done "enough" damage.

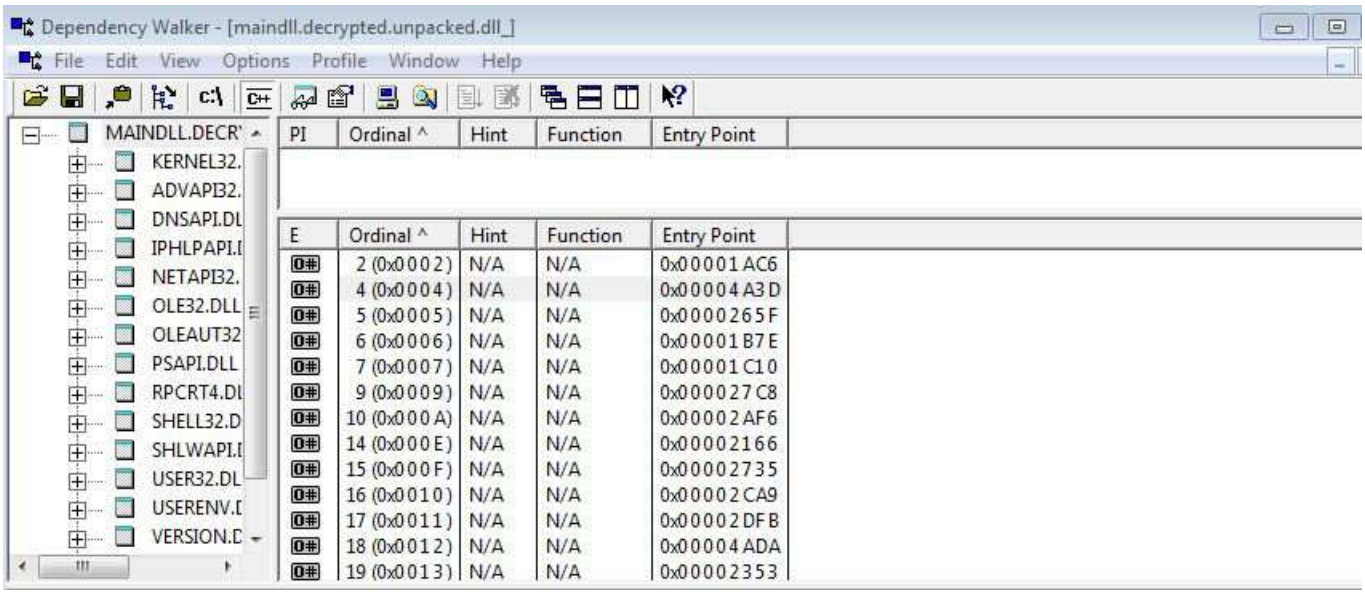

Figure 5: Stuxnet's main library export functions as shown by Dependency Walker

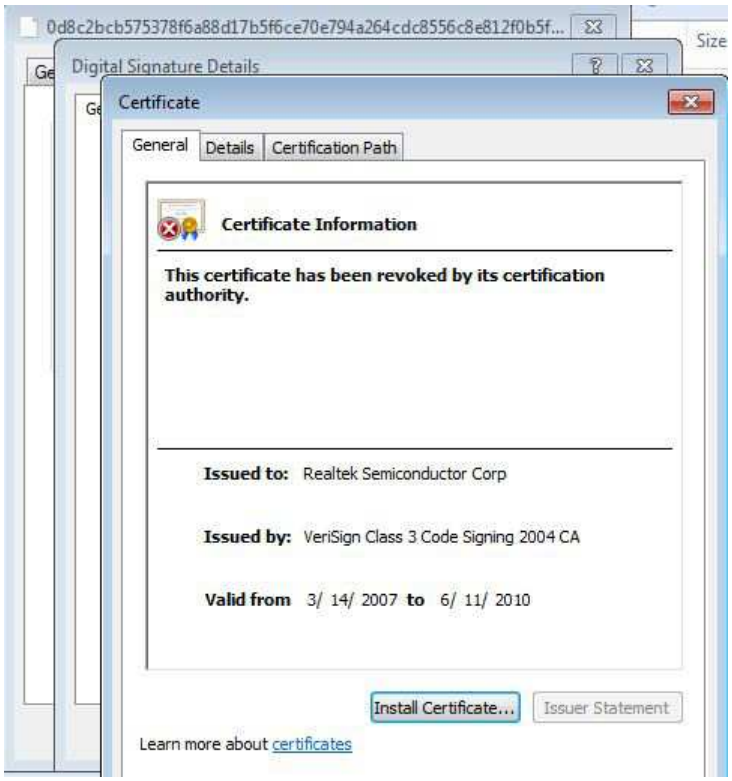

Figure 6: Stuxnet installs two drivers signed with stolen signatures

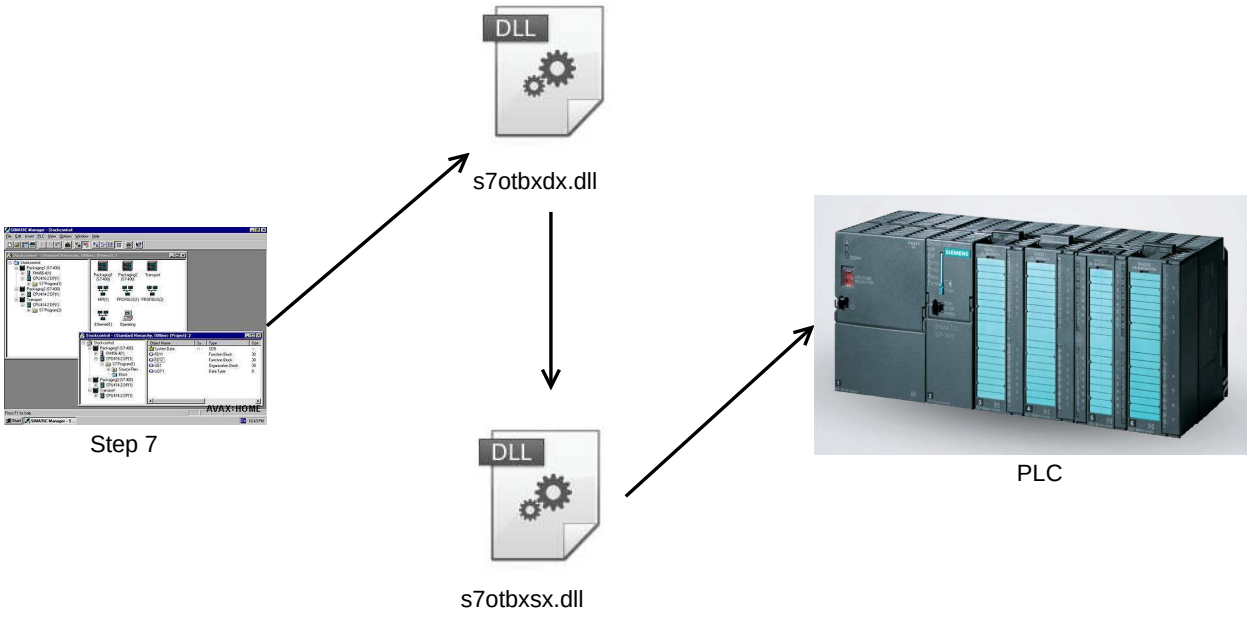

Figure 7: Stuxnet wraps the library used to communicate with the PLCs

2.4.2 Kernel-Mode

Stuxnet installs two kernel-mode drivers. `Mrxcls.sys` is a driver signed by a Realtek certificate as shown in Figure 6. When Stuxnet wants to install it onto the system, it marks it as a boot startup so it starts in the early stages of Windows boot. This driver first reads a registry key which has been written in the installation step and contains the information for injecting Stuxnet images into certain processes.

The other driver, `Mrxnet.sys`, is actually the rootkit and is also digitally signed by a Realtek certificate. It creates a device object and attaches it to the system's device objects so that it can monitor all requests sent to these objects. The purpose of this job is to hide files which meet certain criteria from users.

2.5 Attack phase

Stuxnet is not harmful to ordinary users since its aim is to modify Simatic PLCs manufactured by Siemens [14]. Step7 programs, which control and monitor these PLCs, use a library named `s7otbxdx.dll` to communicate with the actual PLC to read or modify its contents (codes). Stuxnet gets control over all requests sent to the PLC by wrapping this library.

In addition, Stuxnet writes its own malicious code to target certain specific PLCs. It hides its malicious code from users by returning original code blocks instead of the modified blocks upon a read request, as illustrated in Figure 7 [7].

The code Stuxnet infects PLCs with contains three attack sequences, named A, B, and C in Symantec's report. Sequences A and B are similar, with only slight differences, and have the same effects. Sequence C is more advanced but is incomplete and is never executed.

Sequences A and B perform their attacks by running the centrifuge rotors at too-low and too-high frequencies (as low and high as 2 and 1410 Hz, respectively). Interestingly, the periods in which they command the centrifuges to spin at these inappropriate speeds are quite short (50 and 15 minutes, respectively), and are separated by about 27 days between attacks, possibly indicating

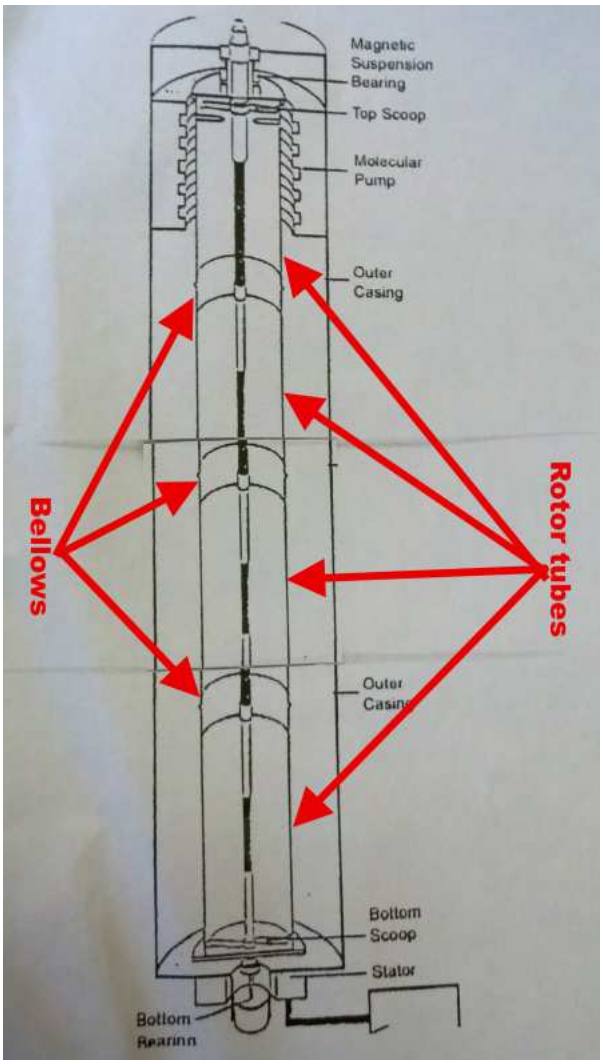

Figure 8: Diagram of a P-1 centrifuge. The Natanz centrifuges are based on the P-1. (Diagram: Institute for Science and International Security)

the designers wanted Stuxnet to operate very stealthily over long periods of time. See Figure 8 for a diagram of a centrifuge similar to the ones used at Natanz.

Although the time spans during which the centrifuges are slowed down or sped up are probably too short for them to actually reach the minimum and maximum values, they still result in significant slow-downs and speed-ups. The slow speeds are sufficient to result in inefficiently processed uranium, and the high speeds are probably sufficient to result in centrifuges actually being destroyed, as they are at the centrifuges' maximum speed limit.

2.6 Concealment of attack activities

Stuxnet hides itself from plant personnel by installing rootkits on infected Windows computers and on infected PLCs, in order to hide its files. By installing a driver on Windows computers, it hides itself by manipulating requests sent to devices.

By intercepting calls to `s7otbxdx.dll`, Stuxnet hides the malicious code it writes to PLCs to

sabotage centrifuges, and also prevents those malicious codes from being overwritten.

Before the malware runs an attack routine, it records the centrifuges' normal operating frequencies and feeds this recorded data to the WinCC monitor program during the attack. The result is that the system shows normal operation instead of alerting personnel to the anomalous frequencies the centrifuges are actually running at.

Stuxnet also modifies some routines on the PLCs, preventing a safe shutdown even if the operator finds out that the system is not operating normally [7].

3 Conclusion

Stuxnet is a malware of such sophistication that it is most likely the work of one or more nations, with Israel and/or the U.S. being the presumed author(s). It spread around the globe, but most of its infections were in Iran, and it seems that the only place it activated its payload was Natanz.

Now, almost two years after Stuxnet's discovery, Iran appears to have purged it from their Natanz equipment, according to a Reuters article [8]. The article was published in February 2012, but states that it is unknown exactly when the Iranians succeeded in clearing their systems of the infection, so they may have been clear for some time now.

It is likely that Stuxnet would have had a greater impact had it not been noticed by security researchers, who subsequently published detailed reports on it (See [7] and [11] for two good examples). It probably destroyed about 1,000 centrifuges and delayed Iran's nuclear weapon program, but likely didn't have as much impact as its creators had hoped for.

Now that the Iranians are on the alert for Stuxnet in specific and such threats in general, it will likely be harder to pull off another such attack against them. Thus, the best chance to derail or at least significantly delay Iran's nuclear weapons program has probably passed, which leaves us with the worrying prospect of Israel, with or without U.S. aid, militarily attacking Iran.

Stuxnet has also increased awareness of the vulnerability of industrial control systems, which haven't been the target of many attacks. This should result in them becoming more hardened against attack as time goes on, but this is balanced against the increased risk of such attacks.

Stuxnet has increased the likelihood that malware authors, be they nation-states or smaller entities, will perpetrate similar attacks in the future. It has proven such attacks possible, raised awareness of them and perhaps interest in them among malicious entities, and provided a sophisticated code base for malware authors to study and modify.

Presumably, assuming the U.S. and/or Israel were Stuxnet's creators, they did a cost-benefit analysis taking into account its potential to deter Iran's nuclear ambitions against the increased risk of similar attack against themselves and others. They must have concluded that the risk was worth it. Hopefully they won't be proven wrong in the coming years.

4 Overview of References

Symantec has produced a detailed, highly-recommended examination of Stuxnet [7]. ESET has produced a similarly detailed report, although with a somewhat different focus [11]. The Langner company, headed by Ralph Langner, was instrumental in deciphering Stuxnet. There is a lot of interesting information on Stuxnet and other topics on their blog [6]. Wikipedia has a good page on Stuxnet [5]. ISIS' website [1] has a wealth of information on Stuxnet, Iran's nuclear program, and other interesting topics. Finally, for some lighter reading, Wired.com has an interesting story about how Symantec, Langner, and others figured out what Stuxnet was [16].

References

[1] The website of isis. Technical report, World Wide Web, http://www.isis-online.org.

[2] David Albright, Paul Brannan, and Christina Walrond. Did stuxnet take out 1,000 centrifuges at the natanz enrichment plant? Technical report, World Wide Web, http://isis-online.org/uploads/isis-reports/documents/stuxnet_FEP_22Dec%2010.pdf, December 2010.

[3] David Albright, Paul Brannan, and Christina Walrond. Stuxnet malware and natanz: Update of isis december 22, 2010 report. Technical report, World Wide Web, http://isis-online.org/uploads/isis-reports/documents/stuxnet_update_15%Feb2011.pdf, February 2011.

[4] Mark Clayton. Stuxnet cyberweapon looks to be one on a production line, researchers say. Technical report, World Wide Web, http://www.csmonitor.com/USA/2012/0106/Stuxnet-cyberweapon-looks-to-be-%one-on-a-production-line-researchers-say, January 2012.

[5] Contributors. Stuxnet. Technical report, World Wide Web, http://en.wikipedia.org/wiki/Stuxnet.

[6] Ralph Langner et. al. The blog of langner.com. Technical report, World Wide Web, http://www.langner.com/en/blog/.

[7] Nicolas Falliere, Liam O Murchu, and Eric Chien. W32.stuxnet dossier (version 1.4). Technical report, World Wide Web, http://www.symantec.com/content/en/us/enterprise/media/security_respons%e/whitepapers/w32_stuxnet_dossier.pdf, February 2011.

[8] Mark Hosenball. Experts say iran has "neutralized" stuxnet virus. Technical report, World Wide Web, http://www.reuters.com/article/2012/02/14/us-iran-usa-stuxnet-idUSTRE81%D24Q20120214, February 2012.

[9] Nuclear Threat Initiative. Iran's profile. Technical report, World Wide Web, http://www.nti.org/country-profiles/iran/nuclear/, March 2012.

[10] Ralph Langner and associates. The prez shows his cascade shape. Technical report, World Wide Web, http://www.langner.com/en/2011/12/07/the-prez-shows-his-cascade-shape/, December 2011.

[11] Aleksandr Matrosov, Eugene Rodionov, David Harley, and Juraj Malcho. Stuxnet under the microscope (revision 1.31). Technical report, World Wide Web, go.eset.com/us/resources/white-papers/Stuxnet_Under_the_Microscope.pdf.

[12] Symantec Security Response. W32.duqu - the precursor to the next stuxnet. Technical report, World Wide Web, http://www.symantec.com/connect/w32_duqu_precursor_next_stuxnet, October 2011.

[13] Mark Russinovich. Analyzing a stuxnet infection with the sysinternals tools, part 1. Technical report, World Wide Web, http://blogs.technet.com/b/markrussinovich/archive/2011/03/30/3416253.a%spx, March 2011.

[14] Wikipedia. Simatic s5 plc. Technical report, World Wide Web, http://en.wikipedia.org/wiki/Simatic_S5_PLC/, February 2012.

[15] Christopher Williams. Israeli security chief celebrates stuxnet cyber attack. Technical report, World Wide Web, http://www.telegraph.co.uk/technology/news/8326274/Israeli-security-chi%ef-celebrates-Stuxnet-cyber-attack.html, February 2011.

[16] Kim Zetter. How digital detectives deciphered stuxnet, the most menacing malware in history. Technical report, World Wide Web, http://www.wired.com/threatlevel/2011/07/how-digital-detectives-deciphe%red-stuxnet/all/1, 2011.

5 Credits for Images Used in the Figures

- The US and Israel flag images come from the game Freeciv, and are licensed under the GPL (version 2).
- The nuclear power plant image is from MSN.com, via http://www.msnbc.msn.com/id/45524978/ns/technology_and_science-space/t/could-natural-nuclear-reactors-have-boosted-life/ (And yes, we know Natanz isn't actually a nuclear power plant).
- The PLC image is from alibaba.com
- The USB flash drive image is from psdgraphics.com
- The centrifuge image is from http://www.turbosquid.com/3d-models/blender-nuclear-centrifuge/663104