Investigating Stuxnet and Flame

Peter Muli Titus

titusmpeter[at]gmail[dot]com

Stuxnet

The computer worm Stuxnet discovered in June 2010 was designed targeting industrial programmable logic controllers (PLCs) that allow the computerization of electromechanical procedures; such as the ones used to control factory machinery on assembly lines, or centrifuges used to separate nuclear material. By exploiting zero-day flaws, the Stuxnet worm targeted machines running Microsoft Windows operating system and their networks, then sought out the Siemens Step7 software. It compromised Iranian PLCs by collecting data on industrial systems and triggering the fast-spinning nuclear centrifuges to tear apart. Stuxnet’s architecture and design are not unique to particular domains, and it can be custom-made as a platform targeting modern PLC and SCADA systems, like in power or automobile plants. Stuxnet ruined almost 1/5 of nuclear centrifuges in Iran (Computer Emergency Response Team, 2010).

Stuxnet contains three modules: a rootkit component that is responsible for hiding all malicious processes and files, preventing the detection of Stuxnet’s presence; a worm responsible for executing all routines associated with the main attack payload; and a link file that executes the propagated worm copies automatically. Stuxnet is usually introduced to its target setting through an infected USB drive. The worm then disseminates across the host network while scanning the nodes controlling a PLC for Siemens Step7 software. If both criteria are absent Stuxnet becomes inactive within the infected computer until both the conditions are satisfied, then it introduces the infected rootkit on the Step7 software and PLC, amending the codes and issuing unpredicted commands to the PLC while at the same time returning a loop of standard system operation values to the users (ESET, 2010).

Stuxnet originally spread via Microsoft Windows OS, targeting industrial control systems manufactured by Siemens. While Stuxnet was not the first case of hackers targeting industrial systems, nor the first intentional cyber warfare act, it was the first revealed malware that subverted and spied on industrial systems. It was also the first malware to include a PLC rootkit. Initially, the worm spreads indiscriminately, but it comprises an extremely specific malware payload designed to target supervisory control and data acquisition (SCADA) systems manufactured by Siemens that are configured to monitor and control specific industrial procedures. Stuxnet infects the PLCs by overthrowing the Step-7 software application used in reprogramming these devices. Different Stuxnet variants targeted five Iranian establishments, with the main target being the uranium enrichment infrastructure (Jonathan, 2010).

Stuxnet, unlike conventional malware, does little damage to networks and computers that fail to meet its specific requirements. The programmers put a lot of due diligence to make sure that only the selected targets were affected. While the Stuxnet worm is unrestrained, it makes itself passive if it fails to locate Siemens software on infected computers. It contains safety measures that prevent each of the infected computers from disseminating the worm to more than three computers, and delete itself on a specified date. Stuxnet encompasses, amid other things, source code for a man-in-the-middle (MIM) attack. The MIM imitates sensor signals for industrial process control so that an infected system fails to shut down due to sensed irregular behavior. The worm comprises of a layered attack targeting three different systems: Windows OS; WinCC, STEP7, and Siemens PCS 7 industrial software applications running on Windows OS; and Siemens S7 PLCs (Karnouskos, 2011).

Stuxnet attacked systems running Windows OS using unparalleled four zero-day attacks. The total zero-day exploits used is uncommon, as they are extremely valued, and creators of malware do not usually waste the usage of four dissimilar ones in the same worm. The Stuxnet worm is unusually large at 500 kilobytes and written in several programming languages that are also uneven for malware. Its Windows component of malware spreads comparatively quickly and universally (Karnouskos, 2011).

The malware contains both kernel-mode rootkit and user-mode capability under Windows, with digitally signed device drivers, with the private keys of two stolen certificates from Realtek and JMicron, both situated in Taiwan's Hsinchu Science Park. The driver signing helped the malware successfully install kernel-mode rootkit without notifying the host, and therefore remaining undetected for a reasonably long period of time. Two websites in Malaysia and Denmark were configured as control and command servers for Stuxnet, allowing for periodical updates, and for industrial reconnaissance to be conducted through uploading of information (ESET, 2010).

Once installed, Stuxnet infects project files that belong to the Siemens WinCC/PCS 7 SCADA control software (Step 7), then overthrows an essential WinCC communication library, s7otbxdx.dll. By accomplishing this, Stuxnet intercepts communications between the target Siemens PLC devices and the WinCC software running on Windows OS. The malware is capable of installing itself on the PLC devices without being noticed, and consequently mask its manifestation from WinCC if the controller software tries to read an infected memory block from the PLC system. The entire Stuxnet code has not been divulged, but its payload targets SCADA configurations that meet the criteria it is programmed to ascertain (Bogdan, 2010).

Stuxnet obliges particular slave variable-frequency drives to be attached to the Siemens S7-300 system under attack, as well as its associated modules. It solitary attacks PLC systems containing variable-frequency drives from specific vendors: Fararo Paya and Vacon, based in Iran and Finland respectively. Likewise, it monitors the attached motors' frequency and only attacks systems spinning between 807 Hz and 1210 Hz. Industrial implementations of motors with such parameters are varied and may comprise gas centrifuges or pumps. The Stuxnet malware is installed into the PLC's memory block DB890 that monitors the system's Profibus messaging. When definite criteria are met, the malware periodically alters the frequency to 1410 Hertz, then to 2 Hertz, and then to 1064 Hertz, thus affecting the process of the associated motors by changing their speeds of rotation. It also mounts a rootkit that hides the malware inside the system, masking the variations in rotation speed from the monitoring systems (Siemens, 2010).

The Iranian government in April 2011 stated that an investigation on the malware had established that the United States and Israel governments were behind the attack. The intelligence agencies of three European countries agreed that Stuxnet was a joint Israel-United States effort (Kaspersky Lab, 2010).

Flame

A new malware thought to be related to Stuxnet was found in May 2012. Researchers called the malware "Flame" after one of its modules. Kaspersky Lab analyzed the code of Flame and concluded that there was a strong relationship between

Stuxnet and Flame. An initial version of Stuxnet consisted of code for propagating infections through USB drives that was nearly identical to a module in that exploited the same susceptibility (Marc, 2012).

Flame, also known as sKyWIper, Skywiper, and Flamer is modular computer malware that attacks computer systems running the Microsoft Windows OS. The malware is used for directed cyber espionage mostly in Middle Eastern countries. Flame’s discovery was announced by MAHER Center of Iranian National CERT, Kaspersky Lab, Computer Emergency Response Team (CERT), and CrySyS Lab of the Budapest University of Technology and Economics on 28 May 2012. It was stated that Flame was arguably the most sophisticated malware ever encountered and probably the most complex malware ever to be found. Flame has the capability of spreading to other systems via a USB stick or a local area network (LAN). It captures screenshots, records audio, keyboard activity and network traffic. The malware also records VoIP conversations and can transform infected workstations into Bluetooth beacons and attempt to copy contact information from close Bluetooth-enabled devices (Kaspersky Lab, 2012).

The captured data, alongside locally stored documents, are transferred to one of the several control and command servers scattered across the globe. The malware then awaits additional instructions from the servers. Flame is employed in a targeted fashion and is able to evade current security software via rootkit functionality. After a system gets infected, the malware can spread to other systems via a USB stick or over a local network (Kaspersky Lab, 2012).

According to Kaspersky Lab estimates, Flame had at first infected roughly 1,000 machines, with its victims including educational institutions, governmental organizations, and private individuals. At that time, May 2012, 65% of the infections occurred in Iran, the Palestinian Territories, Israel, Sudan, Lebanon, Syria, Saudi Arabia, and Egypt. The majority of targets were within Iran. The Flame malware was also reported in North America and Europe. The source code for Flame supports a "kill" command used for wiping all the traces left by the malware from a computer. The initial Flame infections stopped functioning after it was publicly exposed, and the "kill" command was issued (Kaspersky Lab, 2012).

At 20 megabytes, Flame is uncharacteristically large for a malware, written partly in a scripting language called LUA, linked with compiled C++ code, allowing other attack modules to load after the first infection. Flame uses five different methods of encryption and SQLite database for storing structured information. The code injection method used by Flame is stealthy; the malware modules do not appear in the module listing loaded into a process (Iran Computer Emergency Response Team, 2012).

The malware memory pages have a READ, WRITE, and EXECUTE protection that make them not accessible by user-mode applications. Flame's internal code has very few parallels with other malware. It, however, exploits two of the same security susceptibilities previously used by Stuxnet to infect computer systems. The malware defines the installed antivirus software and then adapts its own behavior to reduce the likelihood of detection. Additional pointers of compromise include registry and mutex, such as installing of a fake driver which the malware uses to retain diligence on the compromised system (Iran Computer Emergency Response Team, 2012).

Flame was not designed to deactivate automatically, but it chains a "kill" function for eliminating all the traces of its operation and files from a system on delivery of a module from the controllers. The malware was signed with a fake certificate from the Microsoft Enforced Licensing Intermediate PCA certificate authority. The developers recognized a Microsoft Terminal Server Licensing Service certificate that used the MD5 hashing algorithm (known to be weak) and was inadvertently enabled for code validation. They then produced a fake copy of the certificate that was then used for signing some modules of the malware to make them seem to have come from Microsoft (Iran Computer Emergency Response Team, 2012).

Flame, unlike Stuxnet, appears to have been designed purely for spying. Using the sinkholing modus operandi, Kaspersky Lab demonstrated that a vast majority of the malware targets was within Iran, with the attackers seeking particularly text files, AutoCAD drawings, and PDFs. The malware seemed to gather technical illustrations for intelligence purposes. A grid of 80 servers across Europe, North America, and Asia was used to access the infected machines remotely (Kaspersky Lab, 2012).

On June 19, 2012, *The Washington Post* newspaper published an article claiming that the Flame malware was a joint development by the CIA, NSA, and Israel’s military, conducted at least five years prior to its discovery. The venture was said to be part of a top-secret effort dubbed Olympic Games, intended to gather intelligence in the preparation for a campaign of cyber sabotage aimed at decelerating Iranian nuclear efforts (Kaspersky Lab, 2012).

Remediation

For industrial sites running control systems, what should be done depends very much on the level of vulnerability of the control systems, how stringent the availability and safety requirements are, and what the existing security platform looks like. There is a variety amid security postures of control systems. Some systems are well patched, secured by anti-virus and other protections, with digital signatures regularly updated. Such systems are protected as soon as new anti-virus signatures get through their site's alteration control processes. There are other sites that run old O.S versions without antivirus and are unable to patch those systems due to the risk triggering a malfunction or breaching their vendor support contract terms (Department of Homeland Security, 2010).

Some control systems require very stringent availability and safety requirements e.g. refineries, nuclear sites, and electric power grids. There are significant impacts in case of a catastrophe at one of these locations, hence very constricted change control systems. All the changes, however small, have to be evaluated as to the risk they pose to the physical process operation (Department of Homeland Security, 2010).

What victims should do in light of the Flame, Stuxnet or any other malware is to re-evaluate their security posture in light of the new threat. Depending on the security posture that exists, sites should re-evaluate following proper recommendations. Industry experts agree that the right security stance for industrialized sites is a defense in complexity posture; with defense layers including procedures, policies, physical security, training, personnel screening, computer security, and many other fundamentals (Kaspersky Lab, 2012).

Bibliography

Computer Emergency Response Team. (2010). *Stuxenet: A worm which targets SCADA Systems*. Retrieved on 20th Feb 2015 from http://www.cert-ist.com/public/en/SO_detail?code=stuxnet

David Kushner. (2013). *How Kaspersky Lab tracked down the malware that stymied Iran's nuclear-fuel enrichment program.* Retrieved on 19th Feb 2015 from http://spectrum.ieee.org/telecom/security/the-real-story-of-stuxnet

Department of Homeland Security. (2010). *Control Systems.* Retrieved on 21st Feb 2015 from www.us-cert.gov/control_systems/

ESET. (2010). Stuxnet Under the Microscope. Revision 1.31.

Fildes, Jonathan. (2010). *Stuxnet worm targeted high-value Iranian assets*. Retrieved on 20th Feb 2015 from spectrum.ieee.org/podcast/telecom/security/how-stuxnet-is-rewriting-the-cyberterrorism-playbook

Iran Computer Emergency Response Team. (2012). *Identification of a New Targeted Cyber-Attack*. Retrieved on 18th Feb 2015 from www.certcc.ir/index.php?name=news&file=article&sid=1894&newlang=eng

Kaspersky Lab. (2010). *Kaspersky Lab provides its insights on Stuxnet worm.* Retrieved on 19th Feb 2015 from www.kaspersky.com/about/news/virus/2010/Kaspersky_Lab_provides_its_insights_on_Stuxnet_worm

Kaspersky Lab. (2012). *Resource 207: Kaspersky Lab Research Proves that Stuxnet and Flame Developers are Connected*. Retrieved on 18th Feb 2015 from www.kaspersky.com/about/news/virus/2012/Resource_207_Kaspersky_Lab_Research_Proves_that_Stuxnet_and_Flame_Developers_are_Connected

Razvan, Bogdan. (2010). *Win32.Worm.Stuxnet.A*. Retrieved 20th Feb 2015 from http://www.bitdefender.com/free-virus-removal/#Win32.Worm.Stuxnet.A

S. Karnouskos. (2011). Stuxnet Worm Impact on Industrial Cyber-Physical System Security. In:37th Annual Conference of the IEEE Industrial Electronics Society (IECON 2011), Melbourne, Australia. Retrieved 20 Feb 2015.

Siemens. (2010). *SIMATIC WinCC / SIMATIC PCS 7: Information concerning Malware / Virus / Trojan*. Retrieved on 20th Feb 2015 from support.automation.siemens.com/WW/llisapi.dll?func=ll&objid=43876783&nodeid0=10805583&caller=view&lang=en&siteid=cseus&aktprim=0&objaction=csopen&extranet=standard&viewreg=WW#Recommended_procedure%200408

Stevens, Marc. (2012). *CWI Cryptanalist Discovers New Cryptographic Attack Variant In Flame Spy Malware*. Retrieved on 17th Feb 2015 from www.cwi.nl/news/2012/cwi-cryptanalist-discovers-new-cryptographic-attack-variant-in-flame-spy-malware