Tofino Security | Abterra Technologies | ScadaHacker.com

White Paper

Version 1.0
Published February 22, 2011

How Stuxnet Spreads – A Study of Infection Paths in Best Practice Systems

Contents

Authors

Eric Byres, P. Eng. ISA Fellow
CTO, Byres Security Inc.
eric@byressecurity.com
www.tofinosecurity.com

Andrew Ginter, CISSP
CTO, Abterra Technologies
aginter@abterra.ca
www.abterra.ca

Joel Langill, CEH, CPT, CCNA
CSO, SCADAhacker.com
joel@scadahacker.com
www.scadahacker.com

Executive Summary

The Stuxnet worm is a sophisticated piece of computer malware designed to sabotage industrial processes controlled by Siemens SIMATIC WinCC and PCS 7 control systems. The worm used both known and previously unknown vulnerabilities to install, infect and propagate, and was powerful enough to evade state-of-the-practice security technologies and procedures.

Since its discovery, there has been extensive analysis of Stuxnet's internal workings. What has not been discussed is how the worm might have migrated from the outside world to supposedly isolated and secure industrial control systems (ICS). Understanding the routes that a directed worm takes as it targets an ICS is critical if these vulnerable pathways are to be closed for future worms.

To help address this knowledge gap, this White Paper describes a hypothetical industrial site that follows the high security architecture and best practices defined in vendor documents. It then shows the ways that the Stuxnet worm could make its way through the defenses of the site to take control of the process and cause physical damage.

It is important to note that the analysis presented in this paper is based on a security model that, though it is accepted in industry as a best practice, is often not implemented in practice. System architectures in the real world are typically much less secure than the one presented in this paper.

The paper closes with a discussion of what can be learned from the analysis of pathways in order to prevent infection from future ICS worms. Key findings include the following:

- A modern ICS or SCADA system is highly complex and interconnected, resulting in multiple potential pathways from the outside world to the process controllers.
- Assuming an air-gap between ICS and corporate networks is unrealistic, as information exchanges are essential for process and business operations to function effectively.
- All mechanisms for transfer of electronic information (in any form) to or from an ICS must to be evaluated for security risk. Focusing security efforts on a few obvious pathways (such as USB storage drives or the Enterprise/ICS firewall) is a flawed defense.
- Industry must accept that the complete prevention of ICS infection is probably impossible and that instead of complete prevention, industry must create a security architecture that can respond to the full life cycle of a cyber breach.
- Industry must address the containment of attacks when prevention fails and aggressively segment control networks to limit the consequences of compromise. In particular, securing last-line-of-defense critical systems, such as safety integrated systems (SIS), is essential.
- Combining control and safety functionality in highly integrated ICS equipment exposes systems to common-cause security failures. For critical systems, diversity is important.
- Providing security by simply blocking or allowing entire classes of protocols between manufacturing areas is no longer sufficient. Stuxnet highlights the need for the deep packet inspection (DPI) of key SCADA and ICS protocols.
- The Remote Procedure Call (RPC) protocol is an ideal vector for SCADA and ICS attacks because it is used for so many legitimate purposes in modern control systems.
- Industry should start to include security assessments and testing as part of the system development and periodic maintenance processes in all ICS.
- There is a need to improve the culture of industrial security among both management and technical teams.

If the critical infrastructures of the world are to be safe and secure, then the owners and operators need to recognize that their control systems are now the target of sophisticated attacks. Improved defense-in-depth postures for industrial control systems are needed urgently. Waiting for the next worm may be too late.

Introduction

The Stuxnet worm is a sophisticated piece of computer malware designed to sabotage industrial processes controlled by Siemens SIMATIC WinCC, S7 and PCS 7 control systems. The worm used both known and previously unknown vulnerabilities to spread, and was powerful enough to evade state-of-the-practice security technologies and procedures.

Since the discovery of the Stuxnet worm in July 2010, there has been extensive analysis by Symantec, ESET, Langner and others of the worm's internal workings and the various vulnerabilities it exploits. From the antivirus point of view, this makes perfect sense. Understanding how the worm was designed helps antivirus product vendors make better malware detection software.

What has not been discussed in any depth is how the worm might have migrated from the outside world to a supposedly isolated and secure industrial control system (ICS). To the owners and operators of industrial control systems, this matters. Other worms will follow in Stuxnet's footsteps and understanding the routes that a directed worm takes as it targets an ICS is critical if these vulnerable pathways are to be closed. Only by understanding the full array of threats and pathways into a SCADA or control network can critical processes be made truly secure.

It is easy to imagine a trivial scenario and a corresponding trivial solution:

> **Scenario**: *Joe finds a USB flash drive in the parking lot and brings it into the control room where he plugs it into the PLC programming station.*
>
> **Solution**: *Ban all USB flash drives in the control room.*

While this may be a possibility, it is far more likely that Stuxnet travelled a circuitous path to its final victim. Certainly, the designers of the worm expected it to – they designed at least seven different propagation techniques for Stuxnet to use. Thus, a more realistic analysis of penetration and infection pathways is needed.

This White Paper is intended to address this gap by analyzing a range of potential "infection pathways" in a typical ICS system. Some of these are obvious, but others less so. By shedding light on the multitude of infection pathways, we hope that the designers and operators of industrial facilities can take the appropriate steps to make control systems much more secure from all threats.

Methodology

The first part of the analysis starts with an introduction to the Siemens SIMATIC PCS 7 product line, since this was the target of the Stuxnet worm.

In the second part, we provide an overview of the worm and how it infects a system. We outline how it spreads between computers as it attempts to locate its ultimate victim. Finally, we briefly describe how the worm affects a control system using Siemens SIMATIC products.

In the third part of the paper, we propose a hypothetical "high security site" that is the target of Stuxnet or the next generation of Stuxnet-like worms. The architecture used in the paper assumes this fictitious site is following all the guidance provided in Siemens SIMATIC "Security Concept PCS 7 and WinCC – Basic Document." From a security point of view, this assumption is probably optimistic, as the gap between guidance and reality in the ICS world is often large. However, it is a good model for two reasons – it provides a conservative starting point and it highlights that current "best practices" in ICS security might still have a way to go.

Part four proposes several ways Stuxnet could move from an infected computer of little importance on the corporate network to deep inside the control system. We also look at how the Peer-to-Peer (P2P) and Command and Control (CC) components of Stuxnet could be effective in an otherwise isolated industrial plant.

Finally, we close with a brief analysis of what this means for the security of industrial control systems in the longer term. In particular, we discuss how other "non-Siemens" systems should

consider the vulnerabilities exploited by Stuxnet on a Siemens architecture and prepare for dealing with the next generation worm that could exploit other ICS platforms.

What is the Siemens PCS 7 Industrial Control Systems – A Primer

In order to understand the directed attack Stuxnet performed against Siemens ICS systems, a brief overview of the Siemens SIMATIC PCS 7 architecture is in order.

SIMATIC is a comprehensive term used by Siemens, which includes their complete portfolio of industrial automation solutions ranging from machine vision to distributed I/O systems and programmable controllers. SIMATIC WinCC is a specialized process visualization system that comprises the core Supervisory Control and Data Acquisition System (SCADA). It can be used with Siemens-branded control equipment, such as the S7 line of programmable logic controllers (PLC) or it can be used independently with other control products.

Figure 1: Some Products in the Siemens SIMATIC line, including PLCs, Operator Stations and Engineering Stations

The SIMATIC STEP 7 software environment is used specifically for the programming of the Siemens S7 line of controllers. An integrated solution, composed of S7 PLC's, WinCC visualization software, and STEP 7 configuration software, is then referred to as SIMATIC PCS 7. All computer software components run on Microsoft Windows operating systems, including XP, Server 2003 and Windows 7.

In understanding the SIMATIC PCS 7 system, it is important to separate the functional components that are called “systems” from their platform components that commonly carry names like “stations” or “servers”.

The basis of the SIMATIC PCS 7 control system is divided into three functional components as shown in Figure 2:

- Operator System (OS)
- Automation System (AS)
- Engineering System (ES)

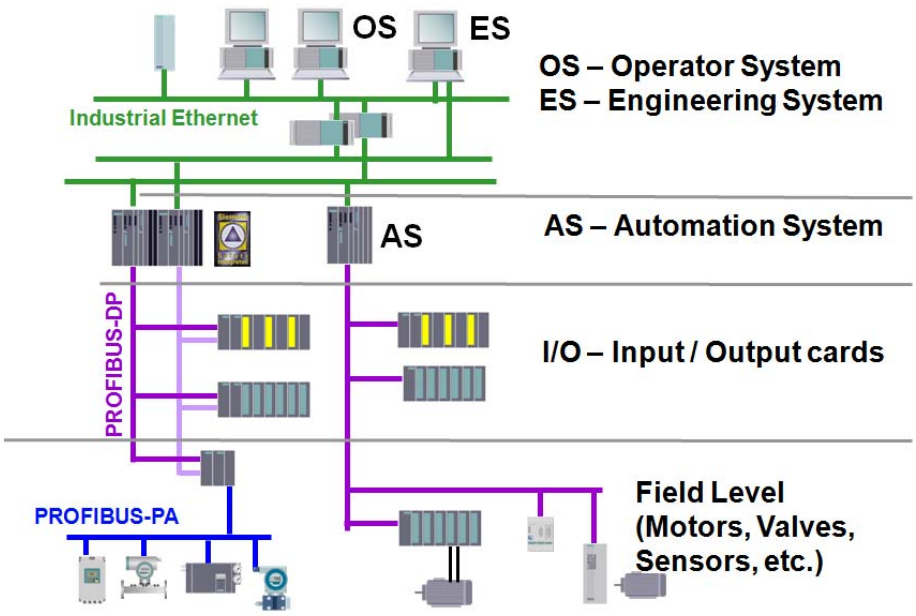

Figure 2: Core Functional Components of the Siemens SIMATIC PCS 7 Control System

The **Operator System (OS)** permits the secure interaction of the operator with the process under control of PCS 7. Operators can monitor the manufacturing process using various visualization techniques to monitor, analyze and manipulate data as necessary. The Operator System architecture is highly flexible, but always consists of a client and server function, which may be implemented on the same or separate physical platforms.

The **Automation System (AS)** is the name given to the class of programmable logic controllers (PLC) used with PCS 7. This includes both the Microbox solution based on a software controller running on a standard computer, and the S7-300 and S7-400 lines of hardware controllers.

The **Engineering System (ES)** consists of software that is responsible for configuring the various PCS 7 system components. The ES is further broken down into the engineering software required to configure either the Operator System (OS) or Automation System (AS), since the OS requires different engineering software for configuration than the AS. The ES allows for configuration and management of the following PCS components and functions:

- Control system hardware including I/O and field devices
- Communication networks
- Automation functionality for continuous and batch processes (Application System engineering via STEP 7 software)
- HMI functionality (Operator System engineering via WinCC software)
- Safety applications (Safety Integrated for Process Automation)
- Diagnostics and asset management functionality
- Batch processes, automated with SIMATIC BATCH
- Material transport, controlled by SIMATIC Route Control

- Cooperation with host CAD/CAE planning tools (import and export of process tags and example solutions)

Since the ES functions are so broad, and cover such a wide range of tasks, Figure 3 below helps clarify the individual components of the ES.

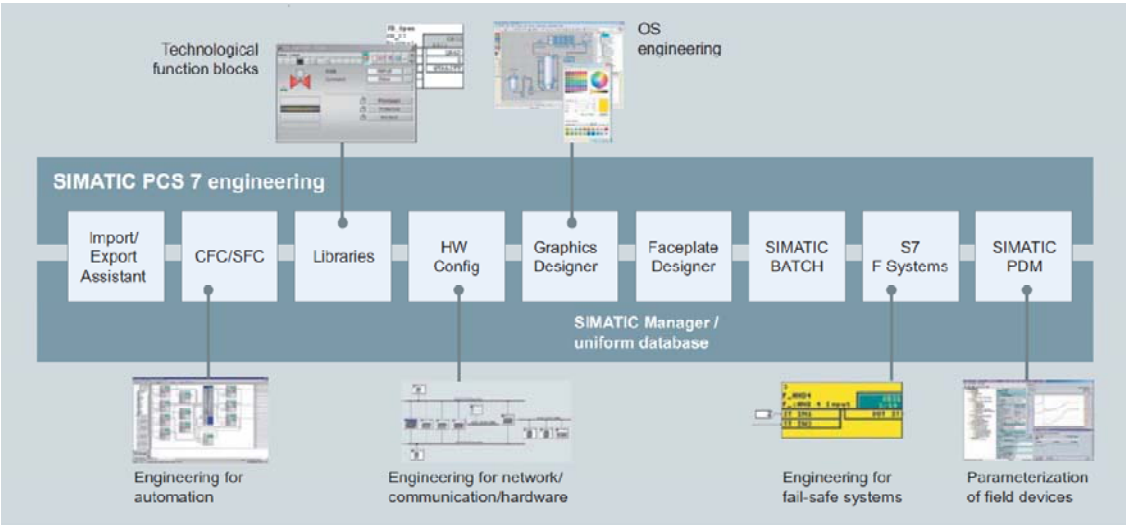

Figure 3: Components of the SIMATIC PCS 7 Engineering System

A few Siemens SIMATIC PCS 7 *software* or *platform* components that are important to note in understanding this paper include the following:

OS Server: The OS Server is one of the two components utilized within the PCS 7 Operator System. The OS Server is used to establish communication with and access basic system level information from the AS components. This includes basic data collection functions for project data, process values, archives, alarms and messages, as well as limited historical trend capabilities. The OS Servers provide all process data to OS Clients. The OS Server can also be used for data collection and archival. However, this data can only be retrieved on OS Clients. OS Servers connect to both the Process Control Network (sometimes called the "terminal bus") and the Control System Network (or "plant bus"). AS controllers are also connected to the Control System Network.

OS Client: The OS Client is the operator terminal that receives data from one or more OS Servers. The OS Server and OS Client may be installed on the same hardware platform in smaller systems, or may be distributed into a true client-server configuration on larger configurations. OS Clients are connected to the Process Control Network.

WinCC Server: The WinCC Server is the second major component comprising the Operator System. It acts as the core server for the Human Machine Interface client/server system, allowing multiple, coordinated HMI client stations to be operated together with process data, archive data, messages, screens and reports. The WinCC Server, like the OS Servers, connects to both the Process Control Network and the Control System Network.

WinCC Client: The WinCC Client is part of the general-purpose WinCC SCADA visualization package used to provide monitoring and control of a particular manufacturing process. When installed with other PCS 7 and OS components, it provides an integrated automation solution incorporating reliable communications, diagnostics functions, and integrated engineering activities. In a typical system, the WinCC client is installed on the same hardware platform as the OS Client, and is connected to the Process Control Network.

Web Navigation Server: The Web Navigation Server provides the capability to monitor

and control the process from external workstations interconnected via an Enterprise Control Network like a company Intranet or even the Internet using standard browser technology. This allows access to the PCS 7 system without the need to install PCS software on the workstations. The Web Navigation Server is installed on a WinCC Server that manages the connection to the PCS 7 system, and allows external access without the clients connecting directly to the PCS 7 system used for real-time control. The Web Navigation Server is connected to the Perimeter Network.

OS Web Server: The OS Web Server provides the ability to access PCS 7 information remotely functioning in a similar fashion to the Web Navigation Server. Unlike the clients using the Web Navigation Server for access to visualization displays of the underlying PCS 7 system, the OS Web Server provides standard Internet access to PCS 7 data functions like process values, archives, alarms and messages, historical trend data, etc. This may include connections from systems such as Manufacturing Execution Systems (MES) or Enterprise Resource Planning (ERP) systems that reside on the Enterprise Control Network. The OS Web Server is connected to the Perimeter Network.

CAS Server: The Central Archive Server (CAS) is used to provide central data management and long-term data archival. This data is then accessible on local PCS 7 OS stations (OS Client, WinCC Client) on the Process Control Network and external workstations on the Enterprise Control Network using a standard Internet browser. The CAS Server is connected to the Perimeter Network.

Engineering Station: An Engineering Station can either be connected to the Process Control Network, or it can reside remotely, where it is referred to as a Support Station. This platform contains all PCS 7 client software components, including the OS Client, WinCC Client, and STEP 7 configuration tools.

What is Stuxnet – A Primer[i]

Stuxnet is a computer worm designed to infect Siemens SIMATIC WinCC and S7 PLC products, either installed as part of a PCS 7 system, or operating on their own. It starts by taking advantage of vulnerabilities in the Windows operating systems and Siemens products. Once it detects a suitable victim, it modifies control logic in specific models of Siemens PLCs. The objective appears to be to sabotage a specific industrial process using two vendors' variable-frequency drive controllers, along with a supervising safety system for the overall process. While there has been much speculation on Stuxnet's intended target, recent information suggests it was Iran's nuclear program and more specifically, its uranium enrichment process.

Stuxnet is capable of infecting both unsupported/legacy and current versions of Windows including Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008 and Windows 7. It also infects the Siemens STEP 7 project files in such a way that it automatically executes when the STEP 7 project is loaded by an uninfected Siemens system.

How Does Stuxnet Spread?

Stuxnet is considered by many to be one of the most complex and well-engineered worms ever seen. It took advantage of at least four zero-day vulnerabilities[ii] and showed considerable sophistication

[i] For a detailed analysis of Stuxnet's internal design, see the Symantec paper "*w32_stuxnet_dossier.pdf*"at http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf

[ii] Zero-day vulnerabilities are defined in this paper as vulnerabilities for which no patch has been issued by the vendor.

in its exploitation of both the Windows platform and the Siemens systems. Some of the important characteristics of the worm are:

- It propagates slowly between sites, typically via USB flash drives and other "removable" media,
- It propagates quickly within a site via multiple network pathways,
- It searches for many vendors' anti-virus technologies on machines being attacked and modifies its behavior to avoid detection,
- It contacts a command and control server on the Internet for instructions and updates,
- It establishes a peer-to-peer network to propagate instructions and updates within a site, even to equipment without direct Internet connectivity,
- It modifies PLC programming logic, causing physical processes to malfunction,
- It hides the modified PLC programs from control engineers and system administrators who are trying to understand why their system has malfunctioned,
- It is signed with certificates stolen from one of two major hardware manufacturers, so that no warnings are raised when the worm is installed, and
- If a particular machine is not the intended target, the worm removes itself from the machine after it has replicated itself to other vulnerable media and machines.

The worm propagates using three completely different mechanisms:

1. Via infected Removable Drives (such as USB flash drives and external portable hard disks);
2. Via Local Area Network communications (such as shared network drives and print spooler services), and
3. Via infected Siemens project files (including both WinCC and STEP 7 files).

Within these three, it uses seven different vulnerability exploitation techniques for spreading to new computers in a system. The worm:

1. Exploits a zero-day vulnerability in Windows Shell handling of LNK files; a vulnerability present in all versions of Windows since at least Windows NT 4.0,
2. Uses several techniques to try to copy itself to accessible network shares and spread from there if at all possible,
3. Copies itself to printer servers using a zero-day vulnerability,
4. Uses an older "Conficker" RPC vulnerability to propagate through unpatched computers,
5. Contacts Siemens WinCC SQLServer database servers and installs itself on those servers via database calls, and
6. Puts copies of itself into Siemens STEP 7 project files to auto-execute whenever the files are loaded.
7. An earlier version of the worm used a variant of the old "autorun.inf" trick to propagate via USB drives.

In addition to the propagation techniques described above, the worm used two zero-day vulnerabilities to escalate privilege on targeted machines. This provided the worm with "system" access privileges so it could copy itself into system processes on compromised machines.

What Does Stuxnet do to Control Systems?

When first installed on a computer with any STEP 7 software installed, Stuxnet attempts to locate Siemens STEP 7 programming stations and infect these. If it succeeds, it replaces the STEP 7 DLL

routines on the programming stations, so that any person viewing a PLC's logic would not see any changes Stuxnet later makes to the PLC. These actions occur on all computers with STEP 7 software installed, irrespective of whether the compromised computers are connected to PLCs.

Stuxnet then looks for specific models of Siemens PLCs (6ES7-315-2 and 6ES7-417). If it is able to connect to one of these two models, it "fingerprints" the PLC by checking for the existence of certain process configurations and strings in the PLC.

If Stuxnet finds what it is looking for in the PLC, it starts one of three sequences to inject different STEP 7 code "payloads" into the PLC. The PLC's PROFIBUS driver is replaced and the main PLC program block (Organizational Block 1) and the primary watchdog block (Organizational Block 35) are significantly modified. As well, depending on which sequence is selected, between 17 and 32 additional function blocks and data blocks are injected into the PLC.

Two of Stuxnet's injected payloads are designed to change the output frequencies of specific Variable Frequency Drives (VFDs) and thus the speed of the motors connected to them, essentially sabotaging an industrial process[iii].

A third payload appears to be designed to control the overall safety system for the centrifuges. This payload takes the inputs coming from the PLC's I/O modules and modifies them so that the PLC safety logic uses incorrect information. The Stuxnet logic then tells the PLC's outputs to do what it wants. This is possibly to prevent a safety system from alarming on or overriding the changes the worm is making to the VFD operations. [iv]

The Target – A High-Security Site

In this part of the analysis, we propose a hypothetical site that is the worm's target. As noted earlier, we assume this site is following all the guidance provided for "high security" sites in Siemens' "Security Concept PCS 7 and WinCC – Basic Document."

It is important to note that the Siemens recommendations for protecting control systems were selected both because the Stuxnet worm specifically targeted Siemens PLCs and because the Siemens recommendations are a good example of existing "best-practice" recommendations. Nothing in this discussion is intended to imply that Siemens control systems are less secure than competing control system solutions. In fact, it is the opinion of the authors that a majority of industrial sites are protected much less thoroughly than is the hypothetical Siemens site described in this paper.

Networks at a High Security Site

According to the Siemens documentation, the high security site is separated into at least four security zones as illustrated in Figure 4:

- The pink "***Enterprise Control Network***" zone is the corporate network, which hosts most business users and business accounting and planning systems, such as Enterprise Resource Planning (ERP) systems. The Enterprise Control Network may itself be separated into additional sub-networks, each with their own protections. Such segmentations and protections are typically established and managed by the corporate IT group.

[iii] The target process is now widely believed to be the centrifuges at Iran's Natanz uranium enrichment facility, but this has not been confirmed.

[iv] Stuxnet experts currently disagree on whether the code path targeting 417 PLCs is actually disabled (blocked by an exception) in Stuxnet. If this is correct, why the author(s) disabled but did not remove the 417 code altogether is unknown.

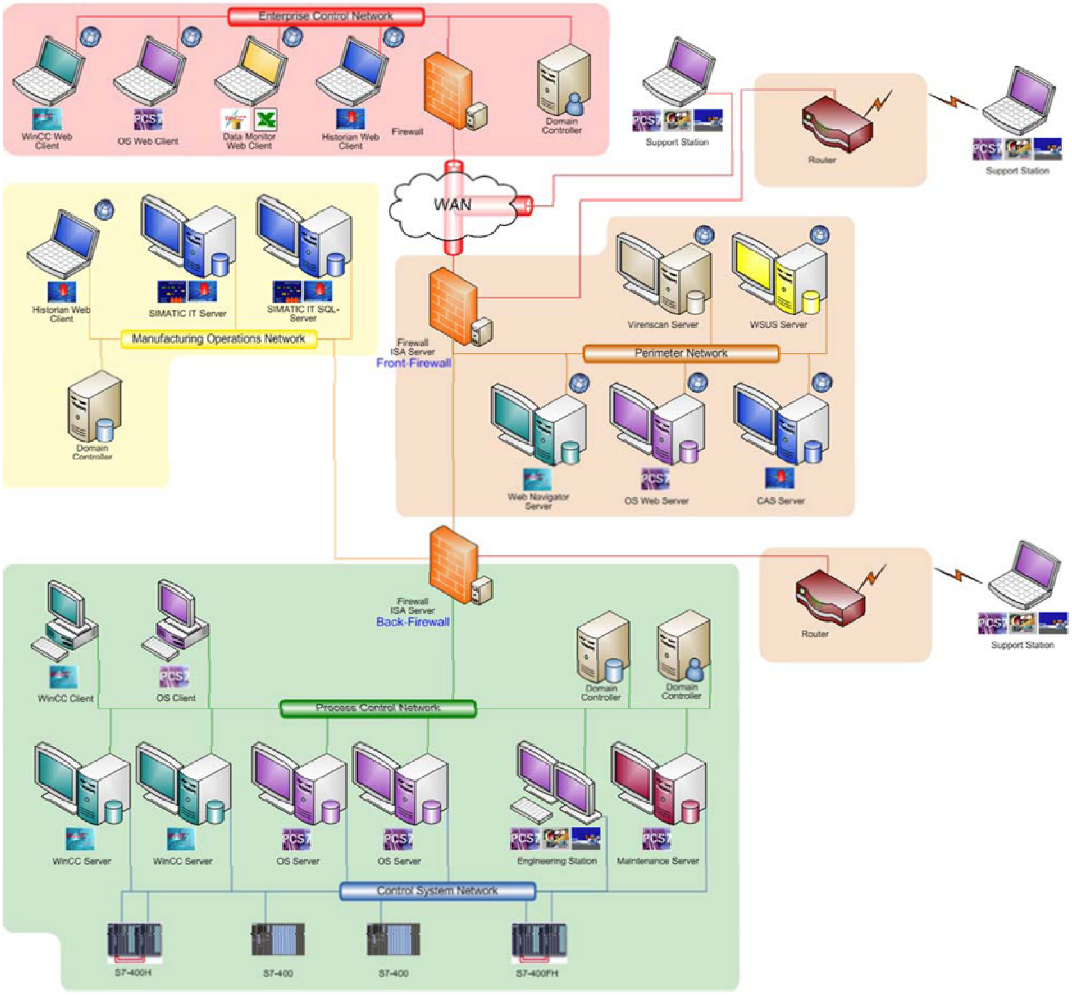

Figure 4: The Hypothetical ICS Network Architecture

- The yellow "***Manufacturing Operations Network***" zone hosts the SIMATIC IT servers, which exchange information between the control system, the ERP system, and other important applications on the Enterprise Control Network.
- The brown "***Perimeter Network***" zone hosts servers that manage equipment in the control system, and servers that provide information to end users on the Enterprise Control Network. This is a common location for servers responsible for providing software patches and updates, including Windows security updates and anti-virus updates. Many of the servers within this zone provide information to end users via web servers and web services. People sometimes refer to this zone as a "demilitarized zone" or DMZ.
- The green security zone hosts two networks: the green "***Process Control Network***" and the blue "***Control System Network***." The Process Control Network hosts the 24x7 plant operators on their Human Machine Interface (HMI) workstations, and is also connected to the WinCC/PCS 7 control system servers. The Control System Network is connected to a number of Programmable Logic Controllers (PLCs) and is also connected to the WinCC/PCS 7 control system servers.

In a large facility, there are frequently multiple "green" zones, one for each control center or operating area. For example, a large chemical plant may have as many as twenty or thirty operating

areas, each with their own SIMATIC PCS 7 system, and each controlling a large portion of the facility with both input and output storage facilities to help decouple operational disturbances between areas. These areas are able to operate independently of other portions of the large facility for some period of time. The facility may have many control rooms and corresponding server rooms, each hosting one or more control centers or operating areas.

The corporate wide area network (WAN) connects sites to one another, and connects different kinds of security zones within sites. Corporate IT manages the various enterprise networks and the corporate firewalls which protect enterprise network segments.

Note that while the Process Control Network and the Control System Network are different networks, they are both in the same security zone. WinCC and PCS 7 control system servers have at least two network interfaces, one for each kind of network. The two networks are separated for performance and technological reasons, not security reasons. In other words, the Control System Network is dedicated to traffic specifically related to "automation" and "control" such as traffic to/from process controllers/PLCs and servers, while traffic on the Process Control Network is utilized for "information" and "display" such as that between HMI's and servers.

Internet Security and Acceleration (ISA) Servers

In the recommended architecture, Microsoft Internet Security and Acceleration (ISA) Servers protect the plant zones from the WAN. They also protect zones from each other. All traffic between security zones passes through an ISA server. Each ISA server hosts a number of functions, such as firewall services, network address translation, web proxies, virus scanning and secure web server publishing.

All of the ISA servers are configured by default to block connections originating in less-trusted networks, such as the corporate WAN. The ISA servers allow connections, such as web services connections, from clients on less-trusted networks to selected servers, such as web servers, in the Perimeter Network.

Servers that receive connections from less-trusted networks are specifically hardened. The ISA servers manage connections to servers in the Perimeter Network, and allow VPN and web connections only for authorized users with legitimate credentials via the WAN.

The ISA servers are also configured to allow machines inside the protected networks to initiate connections "outward" to specific machines and services on less trusted networks. Those connections may pass through the corporate WAN to external servers such as vendor websites on the public Internet. However, connections from protected equipment to arbitrary sites on the Enterprise Control Network or the Internet are not allowed. Just like inbound connections, the outbound connections through the ISA firewalls are "deny by default," with only specific, approved connections to external servers permitted.

It should be mentioned that Windows ISA Server was originally introduced in 2001 to run on the Windows 2000 platform. It was enhanced over the years with new editions released in 2004 and in 2006, with both releases designed for the Windows Server 2003 platform. The Siemens Security Concept document is based on the ISA Server 2006 platform. Today, Microsoft offers the Forefront Threat Management Gateway which was released in 2009 and builds upon the ISA 2006 platform offering new features including support for the Windows Server 2008 and 2008R2 platforms. For additional information on ISA and Forefront TMG, please consult Microsoft's product documentation.

Virtual Private Network Connections

The ISA servers also mediate Virtual Private Network (VPN) connections into protected networks. From time to time, workstations and laptops whose security is managed by third parties are allowed to connect to protected networks through the ISA servers. Such connections are labeled as "support stations" in Figure 4. Support stations are used most commonly for remote engineering activities or vendor support activities. The stations may be at the site, or at a remote corporate site, connected indirectly to the corporate WAN, with their access into corporate networks other than the WAN

mediated by either corporate firewalls or the ISA servers. The vendors may also be at other "non-corporate" remote sites, connecting directly to the ISA servers from quarantine zones served by routers.

When these support stations access protected network zones through an ISA firewall, the firewall authenticates the VPN connection. If the vendor uses WinCC or other process applications that require access to the Process Control Network, the firewall allows a small number of connections, including WinCC and STEP 7 database connections, to protected servers. For broader access to protected networks, the ISA server allows only VPN connections to remote access servers running Microsoft Terminal Services or Remote Desktop Services. These are sometimes referred to as "jump hosts", and are intended to provide isolation between the untrusted hosts, such as support laptops, and the trusted hosts such as the servers and workstations on protected networks.

Host Hardening and Malware Prevention

In addition to the firewall and perimeter protections the ISA servers provide, a variety of host hardening and malware prevention mechanisms are also in place, as specified by the Siemens security architecture. On the Enterprise Control Network, all hosts are part of a comprehensive patch management program that provides automated and managed installation of critical software patches and hot fixes. All hosts have anti-virus and anti-spyware products installed, and signatures for these products are distributed to all hosts immediately upon receipt from the anti-malware vendors.

Hosts have only those applications installed and services enabled that are essential to business functions. Enterprise workstations have access to the open Internet, but all web, ftp and email traffic into the Enterprise Control Network is scanned for spam and malware at the Enterprise Control Network firewall. Select workstations on the Enterprise Control Network have VPN access configured to hosts on the Manufacturing Operations Network and hosts on the Perimeter Network, but no workstations on the Enterprise Control Network have VPN access directly into the Process Control or Control System Networks.

On the Manufacturing Operations Network and the Perimeter Network all hosts are part of the security program implemented at the corporate level. All hosts are current with Siemens patches, Microsoft operating system and application patches, third party application patches, anti-virus and anti-spyware signatures, and all hosts have been reviewed to ensure that only applications and services needed for the correct operation of the host and appropriate network are running. On the Process Control Network and Control System Network, hosts are hardened and are running anti-virus software, but the hosts are not part of the corporate patch management system. Operations manages patches on these critical networks, and subjects new Siemens, Microsoft and third-party patches to a rigorous testing process before approving the patches for deployment on critical system components.

The Microsoft Windows Server Update Services (WSUS) servers manage deployment of approved patches, and such deployment is staged so that if unexpected problems arise when patches are deployed, the affected equipment can be taken offline and rolled back without impacting the overall performance of the control system. In addition, operations manages the anti-virus servers for Process Control and Control System Networks, testing all new signature sets before approving them for deployment, and staging deployment of signature sets just like patch deployment is staged. The WSUS servers also provide management of the deployment of patches allowing users to configure the specific hosts and their timing and sequencing of installation in order to minimize any risk associated with patch rollout. This ensures that equipment that develops unexpected problems because of new patches and signature sets can be taken offline and repaired without affecting the overall performance of the control system.

Compromising the Network

Given the well-secured industrial control system described above, how could a worm like Stuxnet ever penetrate all the way to the PLCs? Yet clearly it did – Siemens reports that it is aware of at least 22 sites that experienced infected control systems and certainly there were other sites, such as sites

with other vendors' products, who would have not reported infections back to Siemens. Suggesting possible answers to this question is the goal of this paper.

For this analysis, assume that the date is May 1, 2010. At that date, the Stuxnet worm had been refined over the course of about 12 months into its mature form, using the shortcut or LNK vulnerability rather than "autorun.inf" to propagate via USB drives. No patches existed for the zero-day vulnerabilities the worm used. No anti-virus signatures existed for the worm. No security researchers knew the worm existed.

With the variety of propagation technologies available to the worm, many scenarios would lead to the state-of-the-practice network described in the previous section to be compromised. The discussion that follows illustrates one way the target ICS could have been infiltrated. At each stage, alternative pathways are also noted.

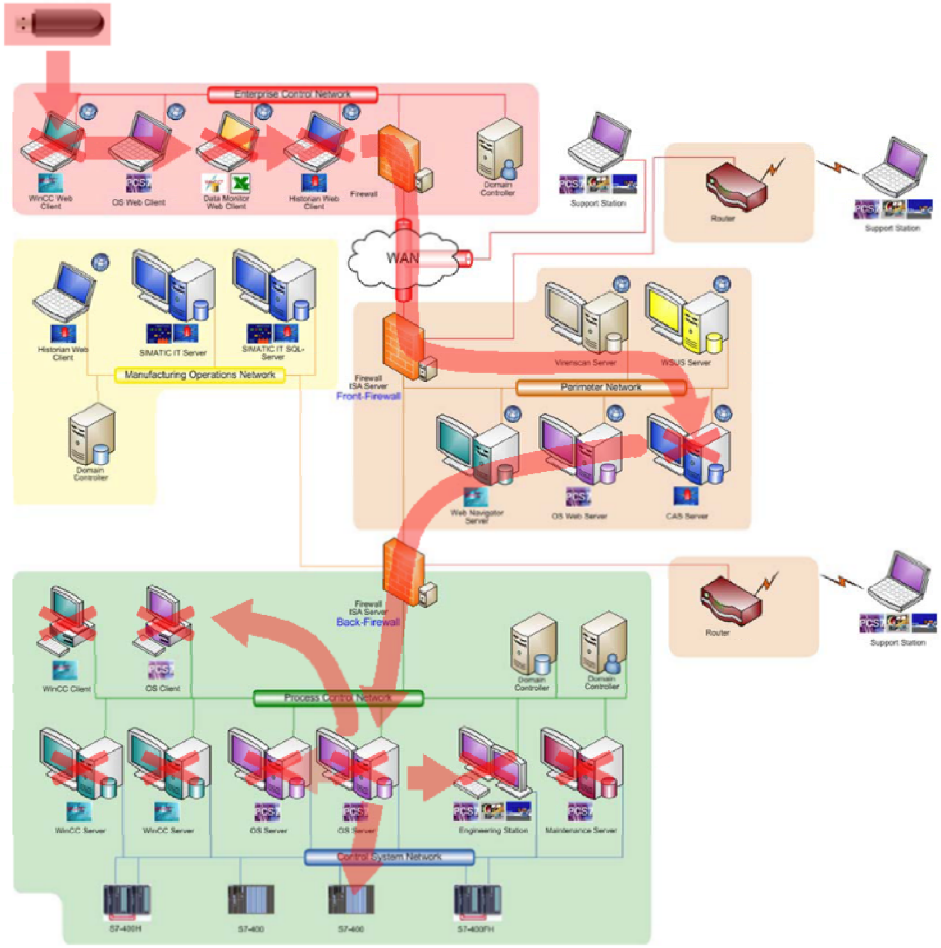

Figure 5: Compromising the Site's Networks

Initial Handoff of the Worm[v]

In our primary scenario, a company employee returns from an off-site visit to a contractor's facility with an infected USB flash drive. The employee has been given the infected drive deliberately by a saboteur employed at the contractor facility.

Alternative pathways: the infected drive may have been simply targeted at the contractor with the assumption that the worm would eventually be transferred to the target site. Most contractor/client relationships are well known in the industry, making selection of a suitable contractor relatively easy.

The initial handoff of the worm to an employee of the target company could also occur at industry tradeshows. Free "branded" USB flash drives are commonly used as give-aways by vendors or as an alternative to CD's for distribution of conference materials. In the past year, one of the authors of this paper was given a "new" USB drive at a major control vendor tradeshow as a gift. The USB drive was infected!

The worm could have also been sent to the organization through a targeted email that contained a special dropper program designed to install Stuxnet. For example, the authors have been able to construct a proof-of-concept dropper for of Stuxnet that is based on an infected PDF.

Infection of Initial Enterprise Computer

Once the employee inserts the infected USB flash drive into his workstation and navigates to the drive using Windows Explorer, the workstation is immediately infected. Anti-virus on the workstation does not generate any alerts, because there are no signatures for the Stuxnet worm at this time. The fact that the workstation is fully patched is of no help, because the LNK vulnerability on shortcut files that the worm uses to infect the machine has no patch at this point in time. Nor do the escalation of privilege vulnerabilities the worm uses to gain system-level access on the workstation. The worm is also able to install what is called "rootkit" software that hides the files used by the worm when browsing the infected flash drive.

Alternative pathways: The initial infection of a computer on the target company network could also occur by the contractor supplying PLC project files that are infected. Due to the nature of contractor/client relationships and the need for continuous collaboration, a variety of project files are freely exchanged between team members. These files not only include the PCS 7 project files that the Stuxnet worm could piggy back on, but also other potentially vulnerable file formats including drawing, spreadsheet, database and PDF files that future worms could exploit. It is unlikely that the transfer of these files can be completely prevented, since many are essential to the engineering design process.

Propagation to other Enterprise Computers

As noted earlier, once on a network, Stuxnet is designed to spread aggressively. Thus within a few hours, the worm would likely spread to printer servers and file servers on the Enterprise Control Network connected directly or indirectly to the compromised workstation.

At this point, the worm might lay dormant, infecting new USB flash drives as they are inserted into compromised equipment, waiting for someone to carry such a flash drive and the worm to a protected network. Alternatively, it may request new instructions from a command and control server – see the section "*Peer-to-Peer Networking*" below. All personnel carrying and using infected flash drives would be unaware that the worm is installed on their drives, because the rootkit hides the worm's files from the user.

Alternative pathways: Some additional alternative paths for infection of the Enterprise Control Network include:

---

[v] Analysis by Symantec indicates that the worm was initially handed off by its developers to at least five separate organizations inside Iran. Severally of these organizations were repeatedly targeted over a period of a year.

- The employee may have attached an "approved" external drive to an infected machine while visiting a contractor and subsequently brought this drive back into the company network.
- The employee may have connected his or her laptop to a compromised network offsite, and thus infected the laptop and then subsequently connected it to the Enterprise Control Network on his or her return.
- A contractor may have visited the site, bringing and using a compromised external drive on the site network.
- A contractor may have visited the site, bringing and using a compromised laptop on the site network.
- A contractor or employee at another facility may have used a file share at this site over the WAN and so compromised the Enterprise Control Network.

Penetrating the Perimeter Network

In our primary scenario, we will assume that one of the workstations on the Enterprise Control Network belongs to an employee who occasionally interacts with the person who manages the historian server on the Perimeter Network. As is commonly done in the industry, the manager has a file share configured on his workstation, as do most employees in that group. The control system team uses the shares on their own workstations to exchange large files with each other over the Enterprise Control Network, rather than exchange the files via the space-limited file servers located on the Enterprise Control Network. Of course, only specific domain accounts are permitted to access these shares.

Stuxnet uses the domain credentials of the user logged into the compromised machine to send a copy of itself to the manager's workstation and activates that copy, compromising that workstation.

In many plants, the historian manager would routinely access the Siemens WinCC Central Archive Server (CAS) historian server from his workstation over a VPN. Typically, the administrator uses both the web interface and Siemens OS Client to the historian to access the CAS server. The web interface provides a view of functionality that the historian exposes to users, and the OS Client allows the administrator to access advanced features of the historian, used primarily for configuration and administration tasks.

Since the manager's workstation is now compromised, the Stuxnet worm contacts the local instance of the SQLServer database "client" on the compromised workstation and discovers the OS Clients' connection to the WinCC database that is installed as part of all CAS servers. The worm contacts the WinCC SQLServer database on the CAS server and propagates to the CAS server on the Perimeter Network through that database connection. The worm installs itself on the CAS server by manipulating both the CAS database contents and stored procedures within the database. The worm now has a foothold on the Perimeter Network.

Alternative pathways: Some alternate paths of infection of the Perimeter Network include:

- At many "real world" sites, the Perimeter Network hosts are not patched routinely. As a result, any VPN connection from a compromised host on the Enterprise Control Network to a host on the Perimeter Network using common Windows RPC communications is at risk. Specifically any host on the Perimeter Network with no patch for the 2008 MS08-067 vulnerability would allow the worm to compromise the Perimeter Network.
- While it does not follow the Siemens security recommendations, it is not unusual for the VPN connections from Enterprise Control Network workstations to the Perimeter Network to not aggressively restrict communications to specific ports and hosts. Often workstations with VPN connections to the Perimeter Network can communicate with any port on any host on the Perimeter Network. In such cases, any Enterprise Control Network workstation with a VPN connection to the Perimeter Network puts at risk every server or workstation on the Perimeter Network with file sharing enabled or a printer connected.

- A contractor or vendor using a remote access mechanism to provide assistance with the support of hosts on the Perimeter Network may remotely access that network from a compromised laptop or workstation. If the contractor can communicate with any exposed file shares or print spoolers on the Perimeter Network, that would permit compromise of those hosts. If the contractor or vendor's workstation can communicate with any unpatched hosts exposing the MS08-067 vulnerability, that channel also permits compromise of hosts on the Perimeter Network.
- While this does not follow the Siemens security recommendations, site administrators on the Enterprise Control Network are known to use file shares to exchange information with servers on the Perimeter Network. Such file shares expose the Perimeter Network to compromise.

Propagation to other Perimeter Network Computers

Once the worm has a foothold in the Perimeter Network, it would attempt to infect any print servers and file servers it could discover. Next, the worm would identify the WinCC software installed on the Web Navigation and CAS Servers, and would likely infect these local databases. It is also possible that if the Web Navigation Server is configured to use Terminal Services for remote access, there could also be STEP 7 software installed on this host, offering the worm the opportunity to install itself inside the STEP 7 project files.

Propagation to Process Control Network and Control System Network

Once the worm takes over the PCS 7 servers in the Perimeter Network, it is then trivial to utilize the network connections that exist to the servers located in the Process Control Network to infect the servers within this zone.

Furthermore, once the STEP 7 project files are infected, it is only a matter of time before an authorized user copies a project file to the Process Control or Control System Networks. In addition, if an administrator were to copy these files to another plant at another site and use the files there, these STEP 7 project files would lead to compromise of that new site by the Stuxnet worm.

In addition, the WinCC Central Archive Server (CAS) on the Perimeter Network has database connections configured through the ISA server, so that the historian server can request historical data from Operator System (OS) Servers on the Process Control Network. The Stuxnet worm can propagate over these connections into these OS Servers and infect all servers on the Process Control Network which expose either print servers, file servers or which have WinCC or STEP 7 software installed on them. STEP 7 is typically installed on engineering stations, while WinCC is common on both operator and engineering stations.

Some of the compromised OS Servers manage connections to the S7 PLCs that control the physical process. The worm connects to those PLCs and modifies the programming in all the PLCs that match the worm's selection criteria. It also installs a special driver on the STEP 7 hosts effectively hiding any modified code from administrators or engineers querying the PLCs, making the worm “invisible” once it is installed on the PLC.

Alternative pathways: Alternative paths for infection of the Process Control and Control System Networks include:

- File shares or print spoolers may be exposed to hosts on the Perimeter Network. Even if a site did not mean to expose such services on the Process Control Network to the Perimeter Network, WinCC components on the Perimeter Network make heavy use of Windows RPC communications to interact with components on the Process Control Network. Print spooling and file sharing use RPC communications. Any path through the ISA firewall that permits RPC communications would permit connections to print spoolers and file shares, regardless of whether such connections were anticipated by personnel designing ISA firewall rules.

For example, if an OPC Classic server (such as OPC Data Access) on the Process Control Network serves information to an application on the Perimeter Network, that connection exposes the RPC communications path since it is the foundation of the OPC Classic protocol.

- Most servers on the Perimeter Network use database connections to servers on the Process Control Network to acquire data for presentation to enterprise users. If any of those servers or workstations becomes compromised, the worm can propagate over that machine's database connection to the Process Control Network.
- PLC programming projects may routinely be carried out on test beds for which security measures are weaker than those applied to production networks. Such test beds may become compromised by removable drives, remote vendors, connections to compromised enterprise hosts or other means. If those infected project files are communicated to hosts on Process Control and Control System Networks, the worm compromises those new hosts.
- A contractor or vendor using a remote access mechanism to provide assistance with the support of hosts on the Process Control Network may remotely access that network from a compromised laptop or workstation. If the contractor can communicate with any exposed file shares or print spoolers on the Process Control Network that would permit compromise of those hosts. If the contractor or vendor's workstation can communicate with any unpatched hosts exposing the MS08-067 vulnerability, that channel also permits compromise of hosts on the Process Control Network.
- Using an infected external drive on any single host on the Process Control Network would compromise that host and the other computers on that network.

Peer-to-Peer Networking

At this point in the scenario, the physical process may or may not immediately malfunction. The Stuxnet worm was designed to contact one of two command and control (C&C) servers over the Internet for new instructions and updates. The worm exchanges information with these servers over the HTTP protocol, on port TCP/80. The payload of communications with those servers is encrypted, but the “envelope” for the communications is plain-text HTTP. None of the contents of the HTTP traffic matches anti-spam or anti-malware rules in corporate Internet firewalls or intrusion monitoring systems (IPS/IDS), and so the traffic to the C&C servers is permitted through to the Internet.

The defense-in-depth posture of the example site however, forbids communication from any ISA-protected network with any machine on the open Internet, outside of a list of specifically authorized machines. The C&C servers are not approved destinations, and direct communication between the infected hosts on the trusted internal control networks and the C&C servers is effectively blocked. Stuxnet works around this defense with a peer-to-peer (P2P) networking capability built into the worm, illustrated in Figure 6. The P2P network uses Windows remote procedure calls (RPC) as its transport – the same protocol used by Windows file sharing, windows print spooling, OPC, and a number of Siemens proprietary data exchange protocols. RPC communications must be enabled within local area networks for the PCS 7 system to function. Thus, all of the infected equipment on the Process Control and Control System Networks are interconnected by the P2P capability.

In this scenario, we will assume that one of the machines on the Process Control Network is used routinely by a control system administrator on the Enterprise Control Network. The administrator connects to the machine through a VPN connection configured to allow only Remote Desktop (RDC) traffic encrypted within the VPN tunnel. This way, a virus or worm on the administrator's machine has minimal opportunity to propagate into the protected network. This administrator, however, routinely prints information from the OS Client machine on the Process Control Network while using the machine remotely. The printer is mapped to the administrator's Enterprise Control Network-connected workstation, and so an RPC connection has been allowed through the ISA firewalls from the OS Client to the administrator's workstation.

Unfortunately, this open RPC connection allows all RPC traffic, including the P2P RPC network that Stuxnet uses. The administrator's workstation, being on the Enterprise Control Network, has no restrictions on connectivity with new sites on the Internet. Since at the proposed time of this scenario (i.e. May 2010), no security researcher has yet discovered Stuxnet or the C&C servers; those server addresses are not included in any list of banned sites on the corporate firewall.

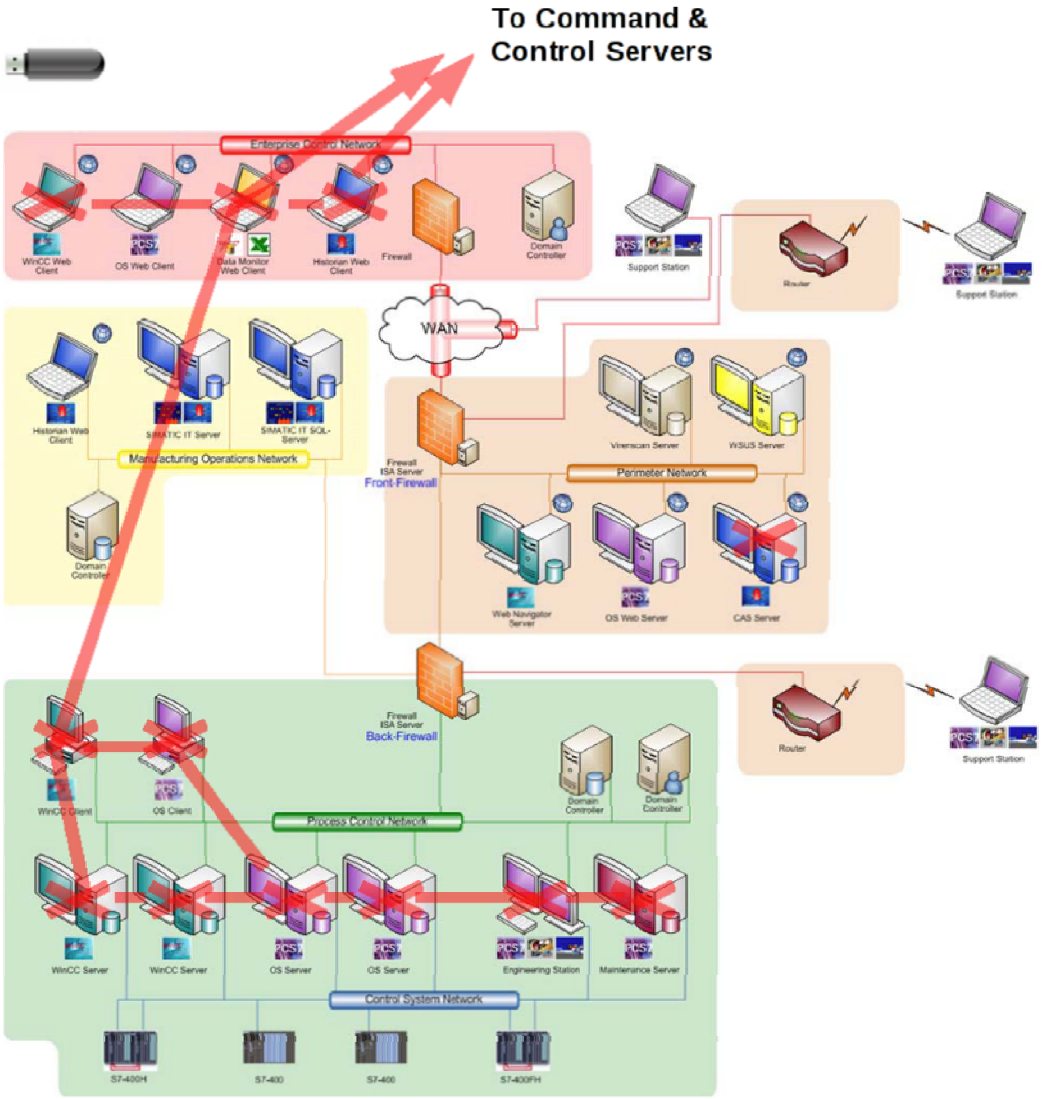

Figure 6: Command and Control Communications

Stuxnet takes over the administrator's workstation using the zero-day print spooler vulnerability, and uses the RPC connection with that workstation to extend the P2P network to the Enterprise Control Network. The P2P network now includes hosts that have contact with the C&C server, and the entire network of compromised machines is put in contact with the Stuxnet authors' command and control servers.

It is important to point out that this path is successful because of the primary difference in philosophy between the “deny by default” policy employed in the configuration of firewalls that interface to trusted control system networks and the “allow outbound by default” policy commonly used in firewalls that connect corporate networks to the Internet.

Some of the capabilities of the C&C servers have been determined through an examination of the Stuxnet worm software, but nothing further has been published about any investigations into those servers. We know the Stuxnet worm's C&C communications and RPC communications software are capable of receiving new versions of the worm and distributing those versions throughout the P2P network. We also know the worm is capable of receiving new executables of any type, including PLC program function blocks, over those communications channels and is capable of executing them locally.

No information is yet available as to what executables, besides new versions of the worm, may have been transmitted to infected sites. This ability to receive and run executables may have assisted in the development of new versions of the worm, and could be used to help propagate the worm through specific target networks. That said, but nothing definitive has been published about how the ability to run arbitrary files was in fact used.

Alternative pathways: Alternative paths of communications with command and control servers include:

- WinCC components on the Perimeter Network make heavy use of Windows RPC communications to interact with components on the Process Control Network. All such communications paths through the ISA firewall, including OPC Classic connections, permit RPC P2P communications as well.
- While not described in the Siemens security recommendations, at many sites administrators on the Enterprise Control Network use file shares to exchange information with servers on Perimeter Network. Paths through the ISA firewall that permit such communications also permit Stuxnet P2P traffic.
- While not described in the Siemens security recommendations, at many sites the VPN connections from Enterprise Control Network workstations to the Perimeter Network do not aggressively restrict communications to specific ports and hosts; most workstations with VPN connections to the Perimeter Network can communicate with any port on any host on the Perimeter Network. In such cases, any compromised host on the Enterprise Control Network with a VPN connection to the Perimeter Network exposes its P2P communications capability to all compromised hosts on the Perimeter Network.
- Even if communications with command and control servers are successfully blocked, any route the original infection either used or could have used can serve as a route through which updates to the worm are propagated. When new versions of the worm are installed on compromised machines, they re-propagate just as the original worm did. This kind of communication path, however, can only be used to update copies of the worm, not to interactively and remotely execute arbitrary files on compromised hosts.

Discussion

If you have managed to stay with the analysis to this point, we congratulate you. One of the key lessons from this analysis is just how complex and interconnected a typical control system is. Potential pathways exist right from the outside world, through the Enterprise Control Network and down to the process controllers.

SCADA/ICS Complexity: Many Roads Lead to Rome

Because of this complexity, Stuxnet had many possible pathways to get to its target process. In Figure 7, we have attempted to summarize some of these pathways in an attack graph or infection data flow diagram. As complicated as this diagram looks, it is certainly incomplete – there are likely many other potential paths this worm (and future worms) might take that we have missed.

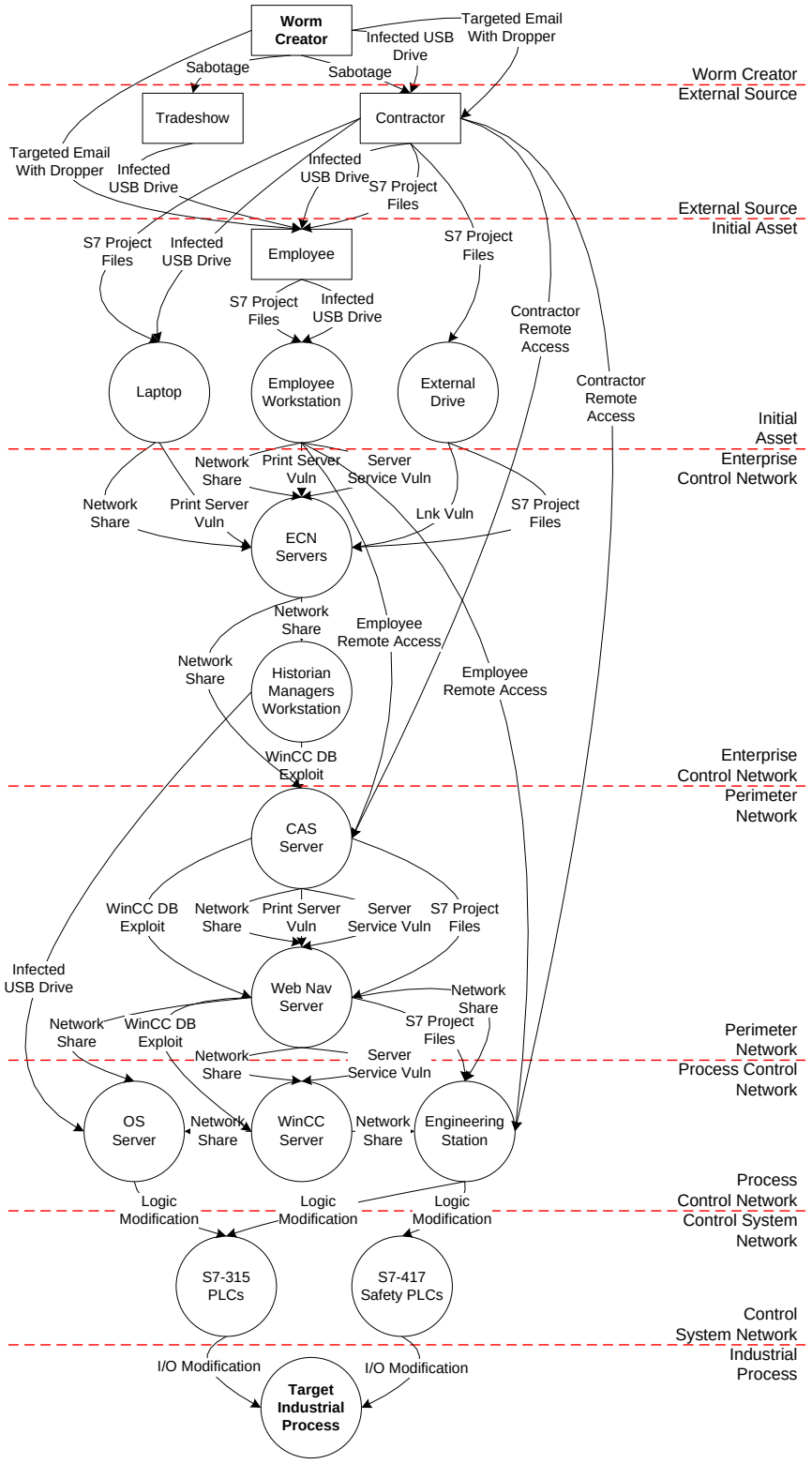

Figure 7: Partial Stuxnet Attack Graph

To make matters worse, some of the stages might be completely bypassed by the worm. For example, the infected USB storage drive might have first compromised one of the Support Stations and so gained direct entrance to Perimeter or Process Control networks[vi]. Alternatively, a PLC programming laptop, used and infected at another site, might have been carried directly into the Control Network and used to program the target PLCs. In these situations, many of the security controls proposed by the Siemens' Security Concept documents would be completely circumvented.

Excessive Focus on the USB Drive as an Attack Vector

What can the SCADA or control engineer learn from such a complicated diagram? First, that focusing on a single path is a flawed defense. For example, the topic discussed most heavily throughout most of the early Stuxnet discourse was the LNK vulnerability in the Windows operation system and how USB drives spread the worm. This was understandable as the novelty, simplicity and zero-day nature of the LNK generated widespread interest in the IT community.

Unfortunately, some companies immediately focused on banning USB drives in their control systems areas, failing to realize it was only one tool in Stuxnet's toolkit of nasty tricks. As the above discussion and diagram illustrate, Stuxnet could have propagated to its target without ever needing to use a USB drive as a pathway, using for example a compact disc (CD) with infected project files to launch the initial infection. Focusing on one path and failing to address the others is a serious security failing.

Furthermore, while it is easy to decree that all USB drives are banned from the plant floor, there are many cases when the drives are the lesser of many security evils. For example, imagine if the network connection to a plant floor device, such as a switch, fails. The maintenance team needs diagnostic data to diagnose the failure. Plugging in a USB drive to download the logs is much safer than plugging in a laptop, but a complete ban on USB drives can force staff to either resort to the less secure option, or do without the diagnostic data and most likely take much longer to diagnose and repair the root problem.

Attack Opportunities using Remote Procedure Calls

The fact that the worm's authors made heavy use of the RPC protocol for both propagation and the P2P network provides important lessons. RPC is an ideal protocol for SCADA and ICS attacks because it is used for so many legitimate purposes in modern control systems. For example, the dominant industrial integration technology, OPC Classic, is based on DCOM and thus requires that RPC traffic be allowed between process areas. Furthermore, control system servers and workstations are routinely configured to share files or printers using the Microsoft RPC/SMB transport between networks. Perhaps most relevant in this example, all Siemens PCS 7 systems make extensive use of a proprietary messaging technology that travels over RPC. Simple blocking of RPC traffic at control systems firewalls would result in a self-induced denial of service for the entire process.

RPC will be a potential pathway for ICS worms for some time to come. The complexity of this protocol and its heavy use in proprietary systems means the opportunities for new zero-day vulnerabilities are significant. Stuxnet's easy paths, such as USB drives, may soon be blocked at many sites, but future worms will have many other paths to choose from – RPC, easily infected project files, and widespread use of hardcoded passwords to mention a few.

Common-Cause Security Failures in ICS

It is also important to consider the fact Stuxnet needed to attack both control and safety functions in the target system in order to be successful. It is not known whether the target system had these functions integrated in the same controller (i.e. the S7-315 PLC) or the S7-315 PLC provided the

[vi] Support Stations connecting via the Back-Firewall will have a trusted connection to the Process Control Network, whereas the Support Stations connecting via the Front-Firewall are typically only granted access to the semi-trusted Perimeter Network.

control functions while the S7-417 PLC provided safety functions. What is known is that the worm was able to use a common protocol and programming system to affect both control and safety functions. This significantly reduced the complexity of the worm and the likelihood of failure.

In the safety industry, Stuxnet would be considered common cause failure mode. An effective mitigation is to ensure that control and safety functions are independent and diverse in mission critical systems. Unfortunately, the current trend in the industry is increasing integration and close coupling of these two functions.

Protecting the Crown Jewels

Is the situation hopeless? We certainly do not think so; we do believe that ICS/SCADA security best practices must improve significantly.

First, the industry needs to accept that the complete prevention of control system infection is probably impossible. Determined worm developers have so many pathways available to them that some assets will be compromised over the life of a system.

Instead of complete prevention, the industry must create a security architecture that can respond to the full life cycle of a cyber breach. One area that needs attention is in the early identification of potential attacks. Currently, there are limited products available that are designed specifically for ICS environments, and in particular, little in terms of inspecting data in transit within control system networks. However, many benefits can be realized from network behavior analysis and existing intrusion detection technologies that use normal traffic patterns to capture anomalies indicating potential threats. These early warning signs can then be integrated with security event monitoring tools capable of reading and analyzing event information from multiple control system hosts further offering insight into the state of the system. Complex alarm annunciation systems are common for plant safety; it is time that these same tools are used to address security issues that can compromise safety.

Next, the industry needs to focus on containment of attacks when prevention fails. For example, assuming that Iran was Stuxnet's target, no matter what its engineering teams did, Stuxnet would have likely infected a number of computers. However, the number of infected systems in Iran (estimated at 60,000) would have been significantly reduced with good zone-based design, such as is suggested by the ANSI/ISA-99 standards. This is an important lesson for all industrial sites anywhere in the world, as the next worm may not be so selective when choosing its victims.

Furthermore, if Stuxnet had been prevented from making that final hop to the Siemens S7 PLCs, particularly the S7-417 PLC that was possibly the safety system, then the actual process would have been safe. This is an important lesson that all SCADA/ICS asset owners should consider – while infected computers are bad, infected safety systems are deadly. The effort spent securing these "last-line-of-defense" critical systems needs to match the seriousness of the consequences if they are breached.

The Need for Better Firewall Granularity and Deployment

The use of firewalls as suggested in the Siemens Security Concepts documents could also be improved. For example, the widely followed "*NISCC Good Practice Guide on Firewall Deployment for SCADA and Process Control Networks*" suggests:

> *An extension to this concept is the idea of using "disjoint" protocols in all PCN-enterprise communications. That is, if a protocol is allowed between the PCN and DMZ then it is explicitly NOT allowed between DMZ and enterprise networks. This design greatly reduces the chance of a worm such as Slammer actually making its way into the PCN/SCADA network since the worm would have to deploy two different exploits over two different protocols.*

Unfortunately, in the security architecture proposed by Siemens (and many other vendors), the same protocols (particularly RPC) are allowed through multiple firewalls and zones. Rules that enforce disjoint protocols probably would not have stopped Stuxnet, as it did have different exploits available, but they could make life much harder for the next worm developer.

Finally, it should be clear that the idea of providing security by simply blocking or allowing entire classes of protocols between areas is no longer sufficient. The fact that RPC was both critical to PCS 7 operations and at the same time a major vector for Stuxnet highlights the need for the deep packet inspection (DPI) of key SCADA and ICS protocols. This type of fine-grained control of network traffic is currently available for a few protocols, such as Modbus TCP and OPC, but DPI for other protocols is also needed.

Weak Industrial Security Culture

It is important to reiterate that the analysis presented in this paper is based on a security model that, though it is accepted in industry as a best practice, is often not implemented in practice. System architectures in the real world are typically much less secure than the one presented in the Siemens Security Concept document.

There are a number of reasons for this. One reason is that there are often inaccurate perceptions about cyber security. For example, senior management at many firms believe their control systems to be completely isolated from the outside world, or they believe that a firewall separates systems securely and therefore their situation is equivalent to being isolated from outside threats.

To date, the perceived risk from external threats has been low, and has not merited more than a cursory understanding by management. This low risk perception has led to most organizations not budgeting sufficient funds or people to protect their control systems from the multiple infection pathways of advanced threats.

For example, most operators today have not sufficiently segmented their control networks to limit the consequences of the occasional infection, do not have early warning infection detection systems in place, and do not include security assessments and testing as part of system development.

Another example; most procurement processes do not include security processes or components in their specifications. Any vendor who includes extra components such as security servers, software and appliances which are not specified and which involve additional costs are at a competitive disadvantage and will often lose their bid. Similarly, ICS vendors investing additional resources in creating a highly secure ICS product are likely to be at a competitive disadvantage in a world largely unaware of the need for security.

In addition to increased capital costs, there are ongoing operational and maintenance costs incurred to ensure that a strong initial security posture is maintained throughout the life of a deployment. Not all ICS management teams have understood the need for these expenses.

The fact that the Siemens SQLServer systems had an embedded password that could not be changed is an excellent example of the conflict between what many ICS customers are willing to pay for and what is needed. This password was available on the Internet as early as 2008 and yet has not been addressed even today. Clearly default passwords that are both unknown to the end user and that cannot be changed even if they are known is not acceptable, so why are they still in the PCS 7 system?

The reason is that the cost of creating a changeable internal password system between the PCS 7 components is expensive, in terms of product modifications and deployment costs. While Siemens will likely release a new version of the PCS 7 that allows modifiable passwords, the bulk of the cost will be on the customers who have to deploy these changes in live and highly distributed ICS systems. Thus, most ICS users have not demanded secure password management from any of their suppliers, with Siemens just an unfortunate example.

In a post-Stuxnet world, vendors, management teams and technical teams need to undertake frank conversations about the risk to their operations of advanced threats, and allocate resources accordingly. In short, they need to work together to improve their industrial security culture and practices.

Looking Forward

Stuxnet is certainly not the last worm of its kind that the SCADA/ICS industry will face. If Stuxnet was successful in damaging its target, whatever that target was, it is wishful thinking not to expect the injured party to respond in kind. Even if Stuxnet was not successful, it is clear that the infrastructure of the developed and developing world is vulnerable to attack by malware as sophisticated as Stuxnet, and that enemies of different countries and cultures now have an example of how to structure their own malware to carry out such attacks. This analysis also demonstrates that control systems are vulnerable not only due to the weaknesses of a particular vendor, but in general with any vendor due to shortcuts or omissions from accepted industry best practices addressing security.

Government agencies are not the only potential threat. Organized crime rings in many geographies have demonstrated amply over the last several years that the skills needed to construct most of the components of the Stuxnet worm are readily available on the black market. Acquiring the remaining PLC programming skills is a matter of identifying the target technologies, purchasing examples of them, and purchasing and attending vendor training in one of the many geographies such training is offered. The payoff would be a powerful new tool for extortion threats against the major infrastructure providers – a style of attack that the banking industry has been dealing with for close to a decade.

Integrating individual components into a single package like Stuxnet is something we have not seen before, but the required skills seem comparable to the skills required to produce any complex software application. Creating another threat like Stuxnet seems straightforward for any organization with sufficient funds and a bit of time. Modifying copies of the Stuxnet worm to target other industrial platforms is also possible and should likely cost far less than writing an entirely new worm.

If the critical infrastructures of the world are to be safe and secure, then the owners and operators need to recognize that their control systems are now the target of sophisticated attacks and need to adjust their security programs accordingly. In particular, security programs need to:

- Consider all possible infection pathways and have strategies for mitigating those pathways, rather than focusing on a single pathway such as USB keys,
- Recognize that no protective security posture is perfect, and take steps to aggressively segment control networks to limit the consequences of compromise,
- Install ICS-appropriate intrusion detection technologies to detect attacks and raise an alarm when equipment is compromised or at risk of compromise,
- Deploy, operate and maintain at maximum effectiveness ICS-appropriate security technologies and practices, including firewalls, antivirus technology, patching systems and whitelisting designed for SCADA/ICS, to make attacks by sophisticated malware much more difficult,
- Look beyond traditional network layer firewalls, towards firewalls that are capable of deep packet inspection of key SCADA and ICS protocols,
- Focus on securing last-line-of-defense critical systems, particularly safety integrated systems (SIS),
- Include security assessments and testing as part of the system development and periodic maintenance processes. Identify and correct potential vulnerabilities, thereby decreasing the likelihood of a successful attack, and
- Work to improve the culture of industrial security amongst management and technical teams.

These changes to improve defense-in-depth postures for industrial control systems are needed urgently. Waiting for the next worm may be too late.

Disclaimers

Siemens control systems and Siemens recommendations for protecting control systems were used extensively in this example both because the Stuxnet worm specifically targeted Siemens control systems, and because the Siemens recommendations are a good example of existing "best-practice" recommendations. This discussion is intended to show how a powerful worm can compromise even well defended systems. Nothing in this discussion is intended to imply that Siemens control systems are less secure than competing control system solutions.

On the contrary, it is the opinion of the authors that a majority of industrial sites are protected much less thoroughly than is the Siemens site described as an example in this paper, and that a similar attack is equally likely on any control system platform. Many of the exploits utilized by Stuxnet would be equally effective on any state-of-the-art platform based on modern computer and control system technologies.

Many of the images in this document are based on images provided in Siemens documentation.

References

For more information about Stuxnet, Siemens PCS7 technology, the Siemens Security Concept and related topics, see the following references:

Siemens Automation

SIMATIC WinCC / SIMATIC PCS 7: Information concerning Stuxnet Malware:

http://support.automation.siemens.com/WW/view/en/43876783

Security concept PCS 7 and WinCC - Basic document:

http://support.automation.siemens.com/ww/view/en/26462131/

Tofino Security White Papers and Application Notes

http://www.tofinosecurity.com/stuxnet-central

Analysis of the Siemens PCS7 "Stuxnet" Malware for Industrial Control System Professionals:

http://www.tofinosecurity.com/professional/siemens-pcs7-wincc-malware

Using Tofino to Control the Spread of the Stuxnet Malware - Application Note:

http://www.tofinosecurity.com/professional/using-tofino-control-stuxnet

Stuxnet Mitigation Matrix - Application Note:

http://www.tofinosecurity.com/professional/stuxnet-mitigation-matrix

Detailed discussion on Stuxnet internals and how the worm works:

http://www.langner.com/en/

http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf

http://www.eset.com/resources/white-papers/Stuxnet_Under_the_Microscope.pdf

Microsoft Security Bulletins

http://www.microsoft.com/technet/security/bulletin/MS08-067.mspx

http://www.microsoft.com/technet/security/bulletin/MS10-046.mspx

http://www.microsoft.com/technet/security/bulletin/MS10-061.mspx

http://www.microsoft.com/technet/security/bulletin/MS10-073.mspx

http://www.microsoft.com/technet/security/bulletin/MS10-092.mspx

Microsoft Security Advisory (2286198)

http://www.microsoft.com/technet/security/advisory/2286198.mspx

http://support.microsoft.com/kb/2286198

http://support.microsoft.com/kb/2347290

US-CERT

http://www.us-cert.gov/control_systems/pdf/ICSA-10-201-01C.pdf

http://www.us-cert.gov/control_systems/pdf/ICSA-10-272-01.pdf

http://www.us-cert.gov/control_systems/pdf/ICSA-10-238-01B.pdf

CVE References

http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-4250

http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-2568

http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-2729

http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-2772