Fraunhofer SIT

FRAUNHOFER INSTITUTE FOR SECURE INFORMATION TECHNOLOGY

INFILTRATING CRITICAL INFRASTRUCTURES WITH NEXT-GENERATION ATTACKS

W32.STUXNET AS A SHOWCASE THREAT

MARTIN BRUNNER, HANS HOFINGER, CHRISTOPH KRAUSS, CHRISTOPHER ROBLEE, PETER SCHOO, AND SASCHA TODT

12/2010



Fraunhofer SIT

Infiltrating Critical Infrastructures with Next-Generation Attacks

W32.Stuxnet as a Showcase Threat

Version 1.4

Martin Brunner, Hans Hofinger, Christoph Krauß, Christopher Roblee, Peter Schoo, and Sascha Todt

Fraunhofer-Institute for Secure Information Technology SIT Munich

Contact: peter.schoo@sit.fraunhofer.de

December 17, 2010

Abstract

The W32.Stuxnet attack, which first gained widespread media attention in July 2010, highlights the dangers, complexity and technical sophistication of malware targeting society's critical infrastructures. Attacks of its nature have potentially wide-ranging implications for the operation of Industrial Control Systems (ICS), including Supervisory Control And Data Acquisition (SCADA) systems. After reviewing W32.Stuxnet's architecture and functionality, we offer unique insights and best practice recommendations based on our expertise and competencies in the IT security domain. We discuss applicable technical and administrative implications for systems operators as well as lessons learned to prevent future compromises.

This paper is intended for technical and non-technical audiences interested in or responsible for the operation of ICS environments, and who may be vulnerable to attacks exploiting an inadequate IT security posture. We invite readers to contact SIT for site-specific guidance.

NES Research Department at Fraunhofer SIT Munich

The Network Security and Early Warning Systems (NES) research department of the Fraunhofer-Institute for Secure Information Technology SIT Munich[1] is researching and developing solutions for the secure operation of networks and network infrastructure services. One focus of research of this group is IT early warning, which requires significant, in-depth knowledge of IT systems security defences, and consequently up-to-date information on the techniques used by attackers. A core competency of NES is the transfer of IT security domain expertise into other domains, for example Smart Metering, Smart Grid and Wireless Sensor Networks.

[1] http://www.sit.fraunhofer.de/

Contents

1 Introduction **6**

2 Overview of W32.Stuxnet **7**
2.1 Attack vector synopsis 7
2.2 Propagation mechanisms 8
2.2.1 Removable drive propagation 8
2.2.2 Network propagation 9
2.2.3 Propagation via Siemens SIMATIC Step 7 project files . . . 10
2.3 Infection Process 10
2.3.1 Infecting the IT environment 10
2.3.2 Infecting PLCs to manipulate industrial processes 12
2.4 Summary and lessons learned 12

3 Malware Targeting Industrial Control Systems **14**
3.1 Notable characteristics of W32.Stuxnet 14
3.2 Attacker motives 15
3.3 The ICS attack surface 15
3.4 Extent of real-world compromises 16
3.5 Next-generation countermeasures for ICS 16

4 Industrial Control Systems Security Management **19**
4.1 System isolation and vulnerability 19
4.2 Factors inherent in the environment 19
4.2.1 Maintaining compliance through long system life cycles . 20
4.2.2 Market pressure 20
4.3 Human factors 21
4.3.1 Security awareness 21
4.3.2 Staff education 21
4.4 Next-generation event monitoring and information sharing . . . 21

5 Conclusion **23**

6 Fraunhofer SIT Munich Competencies **24**
6.1 Network Security and Early Warning Systems (NES) 24
6.2 Embedded Security and Trusted OS (EMS) 24
6.3 Secure Services and Quality Testing (SST) 24
6.4 Reference Customers 25

Bibliography **25**

1 Introduction

Industrial Control Systems (ICS), also commonly referred to as Process Control Systems (PCS) and Supervisory Control And Data Acquisition (SCADA) systems, are ubiquitous in today's society, managing such critical services as water supply, industrial production, and energy generation and distribution. Cyber attacks on these systems or their connecting IT infrastructures can undermine their safe and reliable operation and potentially expose society as a whole to significant risk. The emergence of malicious software (malware) specifically targeting such systems has increased awareness of the need for appropriate IT security solutions in this field.

Although extensive details on the functionality and propagation of the ICS systems-targeting W32.Stuxnet worm have been disclosed, its true intentions remain unclear. Its unusual specificity and geographic distribution of infections have given rise to many hypotheses as to the attackers' identities and motives. As of this writing, however, such scenarios are mostly speculative.

W32.Stuxnet's unprecedented technical ingenuity and complexity (i. e., exploiting four Windows zero-day vulnerabilities, using stolen private-keys from certificates to authorize device-driver manipulation, and reprogramming industrial processes) has led Dr. Udo Helmbrecht, Executive Director of the European Network and Information Security Agency (ENISA), to declare a "paradigm shift, as Stuxnet is a new class and dimension of malware" [5].

The remainder of this paper is organised as follows: the second chapter provides a technical overview of the W32.Stuxnet attack and describes its varied propagation and infection mechanisms in detail. The third chapter describes the potential impact of ICS-targeting malware and the lessons learned from the W32.Stuxnet incident. The fourth chapter highlights security-relevant issues that operators of such infrastructures should consider when designing or maintaining a security architecture, and provides initial recommendations for operators. The fifth and sixth chapters conclude the paper and review the unique IT security competencies of Fraunhofer SIT Munich.

2 Overview of W32.Stuxnet

According to recent and ongoing investigations by Symantec [6], the first known sample of the W32.Stuxnet worm was observed in June 2009 – more than a year before reaching the attention of the mass media. Since then, W32.Stuxnet has evolved considerably, including driver files signed by trusted parties and an exploit for an additional vulnerability not targeted in previous versions.

We now provide an overview of the key technical aspects of the W32.Stuxnet attack. The information presented within was primarily derived from an ENISA-recommended reference: the *W32.Stuxnet Dossier* version 1.0, issued by Symantec in September 2010. [6]

2.1 Attack vector synopsis

W32.Stuxnet uses a multi-staged attack vector to reach its apparent end-target – Programmable Logic Controllers (PLCs) controlling industrial processes within an ICS system. Figure 2.1 shows the propagation and infection steps: (1) The worm is somehow introduced into the target environment (likely via USB drive, but not currently known), and (2) subsequently infects the first PC from which it (3) propagates over the enterprise network (Section 2.2.2), via USB drives (Section 2.2.1), and through infected PLC programming project files (Section 2.2.3), until it (4) reaches and infects a PLC control PC. By targeting a specific combination of process OS/control software/PLC, W32.Stuxnet modifies the PLC programming software (Section 2.3.1) and (5) finally infects the PLC itself (Section 2.3.2), whose functionality can be modified. This could potentially be used to manipulate physical industrial processes controlled by the PLC.

The spreading process of W32.Stuxnet is roughly divided into two phases, each realized by different worm functionalities, i. e., the *propagation within one or between different networks* (described in Section 2.2) and the *infection process* (see Section 2.3), whereby the latter distinguishes between the infection of the Windows PC used for programming, and that of the PLC itself, which is the interface between the enterprise IT systems and the controlled industrial processes – the worm's apparent end target.

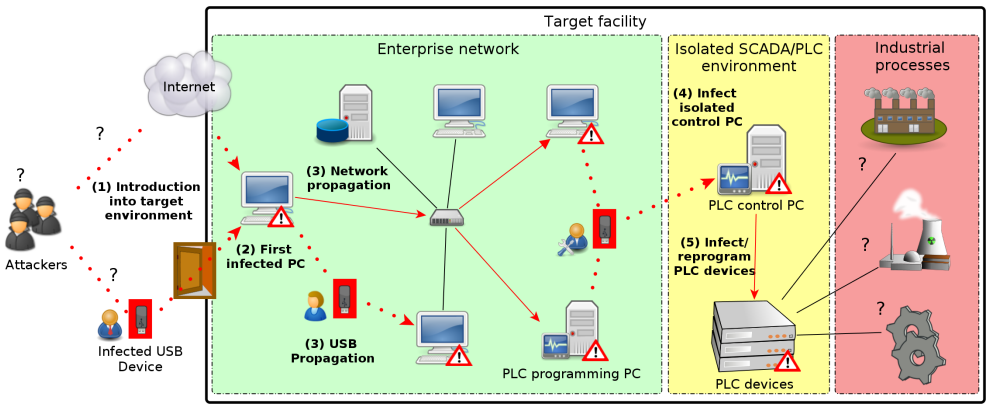


Figure 2.1: W32.Stuxnet attack stages.

2.2 Propagation mechanisms

W32.Stuxnet employs different means of varying sophistication to reach its apparent target: the PLCs of an ICS. Most notable is its mechanism to spread from commodity IT networks to physically-isolated ICSs. As depicted in Figure 2.1 W32.Stuxnet was most likely introduced into the targeted ICS environment(s) through a removable USB storage device, allowing it to breach the perimeter and establish a foothold in the network.

The various propagation techniques employed by W32.Stuxnet illustrate its sly and sophisticated nature, and hint at the level of effort dedicated to its development. This sophistication and mysterious specificity continue to fuel many theories as to W32.Stuxnet's "true" operational objectives, which are outside the scope of this paper.

2.2.1 Removable drive propagation

W32.Stuxnet employs removable USB storage devices (e.g., thumb drives/sticks) to bridge the gap and spread across isolated networks of the ICS and the other (Internet-connected) enterprise PCs of the operator's network. W32.Stuxnet initially relied on the commonly-exploited `autorun.inf` configuration file to execute and spread its infectious payload. Since the requisite Microsoft Windows *AutoPlay* and *AutoRun* features are often disabled for security purposes, however, W32.Stuxnet evolved to exploit the *LNK* zero-day vulnerability (MS10-046 [1]), which allows auto-execution whenever the content of a removable drive (or folder) is simply viewed (i.e., within a folder). W32.Stuxnet installs a rootkit on the infected client PCs to hide its presence on the removable drives. To reduce the risk of detection, it limits the scope of infection by deleting itself from removable drives that have already infected a certain number of other PCs.

[1] http://www.microsoft.com/technet/security/bulletin/ms10-046.mspx

2.2.2 Network propagation

Once W32.Stuxnet has infected a networked PC, it uses several network-based mechanisms to disseminate and to keep itself up-to-date. A simple and stealthy peer-to-peer (P2P) protocol is used to relay W32.Stuxnet updates between infected PCs on the same local area network (LAN), allowing new versions to propagate to systems without Internet connectivity or access to remote command and control (C&C) servers (e.g., in a different security zone).

The four known network-based dissemination mechanisms are summarized below.

Network infection via Siemens SIMATIC WinCC database W32.Stuxnet searches the LAN for a running instance of the WinCC database server (integrated Microsoft SQL Server), and connects to it using a password hard-coded within the WinCC software[2]. Once connected, the server is issued a specially-crafted SQL query, which exploits a vulnerability, ultimately infecting the underlying Windows operating system. The resulting database modifications also allow W32.Stuxnet to re-install itself on the server if necessary.

Propagation via network shares W32.Stuxnet spreads to LAN-accessible network resources using either the local user's Windows credentials, or Windows Management Instrumentation (WMI) to copy itself to the shared resource and execute via a scheduled network job. This propagation vector does not exploit a specific vulnerability, but rather a disregard for the least privilege principle, e.g., running applications as administrator.

Propagation via print spooler zero-day vulnerability (MS10-061) The MS10-061 print spooler vulnerability[3] allows access to the `%System%` folder of a vulnerable (i. e., unpatched) Windows PC on the LAN. W32.Stuxnet exploits this to copy itself into this folder and subsequently execute itself.

Propagation via server service vulnerability (MS08-067) The MS08-067 vulnerability[4] allows arbitrary code execution from SMB clients sending specially-crafted (malformed) path strings, and is widely exploited by the well-publicized Conficker worm [13]. W32.Stuxnet uses this vulnerability to replicate itself to unpatched remote PCs on the LAN after verifying its patch status.

[2] http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2010-2772
[3] http://www.microsoft.com/technet/security/bulletin/MS10-061.mspx
[4] http://www.microsoft.com/technet/security/bulletin/ms08-067.mspx

2.2.3 Propagation via Siemens SIMATIC Step 7 project files

W32.Stuxnet can also hook specific APIs used by the Siemens WinCC SIMATIC manager to open Step 7 project files – the programming software for the targeted PLCs. W32.Stuxnet uses these calls to inject itself into open Step 7 projects, which will infect any PC opening them. The infected project files are furthermore used to update older versions of W32.Stuxnet already installed on the system. Step 7 project files are another mechanism to reach physically isolated (i.e., not connected to any network) PLC control PCs – one step closer to the end target.

2.3 Infection Process

In an ICS, lead and control instructions are loaded from a control PC onto attached PLC devices, which typically control one or more industrial processes. To achieve its end technical goal of infecting specific Siemens SIMATIC PLC devices, W32.Stuxnet must first infect the Windows PC used to program it (running Step 7 control software), which may entail traversing numerous intermediate PCs as "stepping stones" to this control PC. When successful, W32.Stuxnet can tap and modify the communication channel between control PC and PLC, and finally, the programming process itself.

2.3.1 Infecting the IT environment

W32.Stuxnet is initially delivered to a PC via a dropper executable, which wraps a relatively large (approximately 1.5 MB) dll file containing many hidden resources, exports, and two configuration files. The dropper loads the dll into memory and initiates the main W32.Stuxnet bootstrap sequence. The main dll is eventually injected into an existing trusted, or newly-created process to mask its presence. During injection, W32.Stuxnet also scans the Windows registry for indicators of certain anti-virus (AV) programs, whose presence could interfere with the infection process. The injection target itself also depends on the AV or security software installed. For example, W32.Stuxnet injects itself into the trusted Windows *Winlogon.exe* process if McAfee AV is installed.

When initially loaded on a Windows system, the main W32.Stuxnet dll checks certain operating system properties and attempts to gain administrator privileges if it does not already posses them. Administrator rights are necessary for W32.Stuxnet to take full control of the infected PC and complete its operation. To gain these rights, W32.Stuxnet exploits one of two zero-day escalation of privileges vulnerabilities (Task Scheduler[5] or CVE-2010-2743[6]), depending on

[5] http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-3888
[6] http://www.microsoft.com/technet/security/bulletin/ms10-073.mspx

the host's operating system. If not running on a 32-bit version of the following operating systems, W32.Stuxnet simply terminates:

- Windows 2000
- Windows 2003
- Windows XP
- Windows Vista
- Windows Server 2008
- Windows Server 2008 R2
- Windows 7

Once running as a privileged process, W32.Stuxnet initiates its main installation routine, which comprises the following operations:

1. Installation of a load point driver file (`Mrxcls.sys`) that re-launches W32.Stuxnet each time the system is rebooted.
2. Installation of a rootkit driver file (`MrxNet.sys`) to hide and replace (if necessary) malicious files.
3. Injection of functionality into the Windows `services.exe` process to infect removable drives.
4. Injection of functionality into Step 7 process to infect projects (if present).
5. Infection of Step 7 projects via a compromised Step 7 process (if present).

The two (malicious) driver files were digitally signed with a presumably stolen private key of Realtek Semiconductors Corporation. This allows W32.Stuxnet to install the files, masquerading as legitimate Windows drivers, which must be signed by a software publisher trusted by a certification authority (CA), e. g., VeriSign. Otherwise, Windows would either alert the user or refuse to install the driver altogether. After the Realtek certificate was revoked by VeriSign, the driver files were quickly replaced by versions similarly signed by JMicron Technology Corporation.

Once installed on a PC, W32.Stuxnet attempts to contact a remote C&C server on the Internet, providing a (fairly ordinary) backdoor to the attacker, who can use this channel to issue commands and upload arbitrary executables to the infected PC. W32.Stuxnet checks the infected host for the targeted ICS software (Siemens WinCC and SIMATIC Step 7) and reports its presence to the C&C server. This software is used to write programs for the PLCs, and is the next step in the attack vector. If not found on the current host, W32.Stuxnet simply continues its propagation (via LAN and removable drive) through this "intermediate host" in search of an appropriate system.

2.3.2 Infecting PLCs to manipulate industrial processes

As mentioned earlier, the final goal of the W32.Stuxnet attack vector is to reprogram specific SIMATIC PLCs and thereby manipulate the controlled processes. This requires infecting the WinCC and Step 7 PLC control software on a PC physically connected to a PLC device. W32.Stuxnet achieves this by replacing a Step 7 library dll file used for communicating with the attached PLC, allowing the malware to intercept and modify all communications between the controller and the Step 7 control software – effectively a "man-in-the-middle."

Most API/communication calls of the original Step 7 dll remain unchanged, allowing normal operation to continue without revealing the infection, since the hijacked communication calls remain entirely transparent to the PLC operator. W32.Stuxnet then selectively infects at least two specific SIMATIC PLC device families by writing its own compiled code and data blocks (the true W32.Stuxnet payload) into the cyclically-repeated main entry programs, allowing it to partially reprogram the industrial process controlled and monitored by the respective PLC control software. The main payload has now been delivered to target, and the apparent end goal – manipulating industrial processes – achieved.

As mentioned in Section 2.2.1, W32.Stuxnet hides the PLC infection from the control PC with the help of a rootkit functionality, which is incorporated in the modified Step 7 dll file.

The intended physical impact (e.g., disabling pumps, activating turbines) of the injected PLC code is impossible to determine without specifics about the targeted industrial environment and processes. It is very likely that the attackers were in possession of this domain knowledge for at least one victim.

2.4 Summary and lessons learned

ICSs are normally thought to be separated by construction from any external Internet-connection. To bridge this gap and to establish a foothold in its target environment, W32.Stuxnet most likely used a spreading mechanism via portable storage devices (e.g., USB drives). This was already a preferred propagation method many years ago (e.g., floppy disks), when the transport via infected portable media was the only channel to reach a large group of computer systems. Combining these techniques with the abuse of Windows autorun functionalities was also quite common.

Once the gap between the production controlling network, the ICS, and the enterprise network of an industrial operator is bridged, the ICS often exposes an entirely different attack surface than that of a typical Internet-connected system (see Section 3.3).

Unfortunately, the exchange of some data, such as system patches and performance information, between network segments is often unavoidable. To maintain ICS isolation, a viable design would avoid static connections between the two networks entirely, quarantining any initial exploits.

W32.Stuxnet's use of four zero-day exploits and its own limited execution profile suggest that it targets a very specific environment and reveals the sophistication and effort behind this attack. However, at the same time, it relies heavily on breaching the isolation of ICS networks, or on an insider to introduce the exploit into the ICS. Three of the four zero-day vulnerabilities exploited by W32.Stuxnet have meanwhile been resolved by the vendor (i.e., patches available). One of them, the Task scheduler vulnerability for privilege escalation described in Section 2.3.1, remains unpatched at the time of writing, and an exploit is publicly available [7].

The stealthiness of W32.Stuxnet makes it difficult to notice infections on non-target systems. However, an up-to-date AV could have prevented a large number of new infections. For instance, Symantec added a signature for Stuxnet (at that time as "W32.Temphid") on July 13, after which many new infections were still detected (see Figure 5 of [6]). Even if W32.Stuxnet was specifically crafted for a defined target environment, the possibility of side effects and hidden malware functionalities (e.g., backdoor) on infected systems cannot be excluded.

Results from Symantec's recent analysis [8] of W32.Stuxnet's payload (i.e., PLC manipulation), indicate that the attackers required deep insight and insider knowledge about the target's environmental settings to achieve their end goals. It is difficult estimate the level of effort spent to obtain such information, and whether it would be transferable (e.g., to attack additional facilities).

[7] http://www.exploit-db.com/exploits/15589/
[8] http://www.symantec.com/connect/blogs/stuxnet-breakthrough

3 Malware Targeting Industrial Control Systems

The showcase W32.Stuxnet is an extraordinary example of a targeted attack, exhibiting an unprecedented combination of stealth, resilience, and specificity to target individual systems and software in ICS environments. Far from a single exploit, W32.Stuxnet represents a full-scale exploitation and persistence framework – an Advanced Persistent Threat (APT) [1].

This chapter builds on the technical descriptions in chapter 2, initially reviewing W32.Stuxnet's technical highlights, and then outlining the lessons learned in the general context of ICS.

3.1 Notable characteristics of W32.Stuxnet

The W32.Stuxnet worm ensures its resilient and stealthy propagation at multiple levels. Once on a targeted PLC platform, it further ensures that only specific facilities targeted by the attackers are affected. Other infected PCs remain generally unaffected (i. e., no malicious payload), but are instead used as stepping stones to further propagate itself. The creators likely possessed deep insider knowledge of the targeted environment, such as details of its construction, setup, software, and the interaction between all the Siemens SIMATIC components. This is also apparent in the W32.Stuxnet executable files, which contain hundreds of code fragments referring to special functions of the targeted WinCC software. The attack's specificity suggests a collaboration of highly skilled technicians from various disciplines. Additionally, its installed driver files were signed by legitimate, trusted vendors' private keys likely stolen from the vendors themselves.

By involving the user in the attack vector to distribute the malware (e. g., via hand-carried USB drives), W32.Stuxnet performs an "over-the-air" attack to reach offline targets that are not remotely exploitable.

W32.Stuxnet exploits several previously unknown zero-day vulnerabilities, which are very difficult and expensive to procure, and is the first known malware to exploit four different zero-day vulnerabilities. The exploits furthermore cover a broad attack surface - ranging from Windows 2000 to Windows 7 and various software applications. Discovering zero-days requires major investment of both expertise and money, rendering them extremely valuable (commonly hundreds of thousands of dollars) in the underground economy. W32.Stuxnet targets, and

[1] http://www.damballa.com/knowledge/advanced-persistent-threats.php

therefore exposes, several zero-day vulnerabilities at once. We find it remarkable that the attackers were willing to sacrifice their exclusive, highly-valuable knowledge of these vulnerabilities in a single malware campaign. This concentration and sophistication of attack methods integrated in a single piece of malware is unprecedented.

Although developing or acquiring the advanced components necessary for such an attack must have been extremely costly, W32.Stuxnet lacks an apparent profit model. It behaves is entirely unlike a criminal worm, as it neither steals information, nor sends spam, nor incorporates compromised hosts into a botnet, etc.

3.2 Attacker motives

Attackers of ICSs are driven by profoundly different motives than those who target conventional IT environments, such as fraud. Their interests rather lie in violating system integrity and in manipulating processes to perpetrate industrial sabotage. This could range from subtle operational disruptions to the shutdown of entire facilities.

ICSs do not commonly yield attractive information such as passwords, banking credentials or sensitive data, which are among the "classical" targets of malware. Likewise, industrial operating parameters and configuration data, as might be found in programming project files, is not easily transferable to other site installations and therefore lacks value for conventional data thieves seeking a return on their investment. Furthermore, since they typically can't reach the Internet, these systems are not viable as botnet members, which might be used to distribute spam or launch DDoS attacks. Depending on the level of sophistication, infecting and persisting stealthily within an ICS also requires comprehensive reconnaissance of the target.

3.3 The ICS attack surface

As an example of ICS-targeting malware, W32.Stuxnet exposes specific and general security issues in Microsoft Windows, as well as in process control software and equipment. Additionally, it discloses the use of improper or insufficient security practices in this domain, such as [9]:

- Obsolete authentication models
- Insufficient isolation (i.e., adequate security barriers needed when connected to the Internet)
- Misconfiguration (e.g., enabled USB ports on critical systems)
- Static or hard-coded default passwords
- Running old, often unpatched, software stacks for compatibility with legacy systems (e.g., bound to specific software vendor) [17]

These hazards are further aggravated by the growing interconnectivity between ICSs with well-established, but insecure, Internet technologies.

3.4 Extent of real-world compromises

Despite W32.Stuxnet's singular nature, it cannot be assumed to be the first or only attack of its kind – but rather the first *publicly-disclosed*. Its discovery and wide notoriety indicate that it was either not (fully) successful or that the authors intended its exposure.

According to a McAfee Center for Strategic and International Studies study[3], the majority of critical infrastructure providers worldwide report being targeted by "representatives" of foreign governments: more than half have recently experienced a targeted, stealthy attack similar to those dedicated to Google, Adobe, and about thirty other companies. Moreover, almost 90% said their networks had been infected with malware, and more than 70% had been victims of DDoS attacks, vandalism, leakage of sensitive data, insider threats, phishing, or pharming. It can therefore be assumed that targeted attacks do occur semi-regularly, but largely go unnoticed by the public for various reasons.

3.5 Next-generation countermeasures for ICS

Organizations running process control systems face serious challenges securing traditionally proprietary technology. Moreover, common security tools and frameworks lack visibility into the domain of mostly closed-source PLC software. Securing PLCs requires appropriate access so that once sabotaged with malware, it can be detected and mitigated.

Many of these products have been known to be vulnerable for years. According to a Red Tiger Security study [12], the average age of a vulnerability was 311 days for a control system, while some vulnerabilities were more than three years old. According to the McAfee Center for Strategic and International Studies study [3], downtime from an attack can cost critical infrastructure organizations up to eight million dollars per day – and a significant number of those organizations expect a major attack in their industry within the next year (Figure 3.1), while already enduring regular attacks (such as large-scale DDOS) (Figure 3.2).

According to the McAfee study, almost two thirds of operators experienced operational disturbances, some with severe impact on significant functions (Figure 3.3). Traditional defense-in-depth strategies of firewalls and intrusion detection systems can not prevent this type of (multi-staged and -layered) attack. In addition to employing existing ICS security enhancements (e.g., communication protocols [7, 8]), combating attacks like W32.Stuxnet would benefit from collaboration among all involved (perhaps competitive) entities: the process control

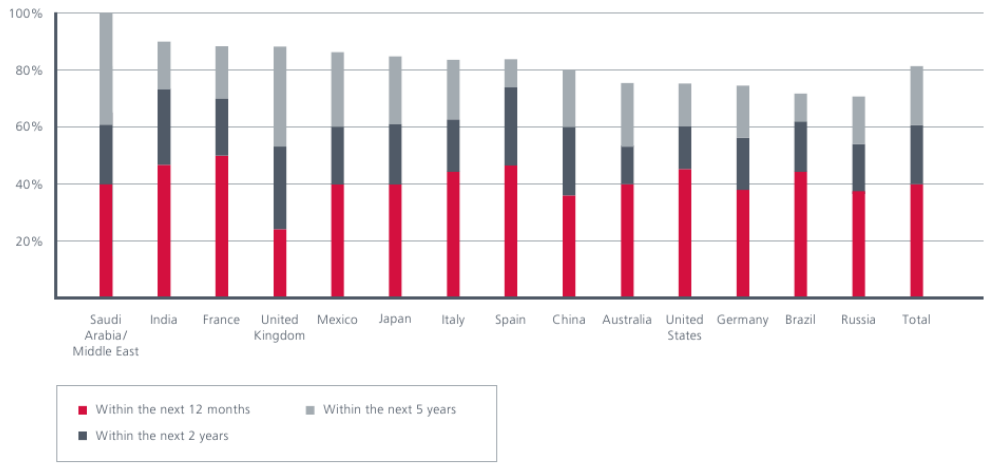


Figure 3.1:
Expectations how long it will take before a major cyber incident happens according to [3].

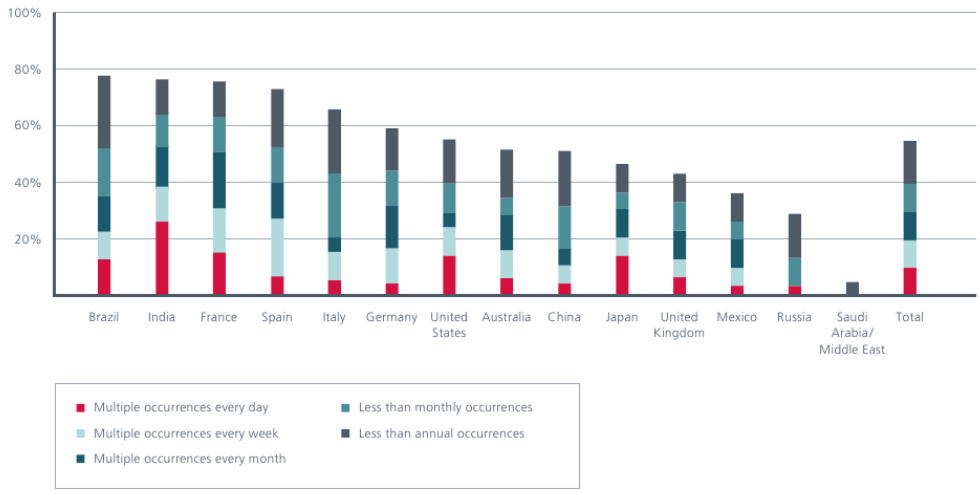


Figure 3.2: Percentage of critical infrastructure operators reporting large-scale DDoS ([3]).

vendors, the operators, and the IT security community. This need for a collaborative approach [2] was already demonstrated during the response to W32.Stuxnet, when various public and private sector entities worked together to address and contain the issue.

Operators are responsible for the safe operation of their own infrastructure, and as threats like W32.Stuxnet continue to emerge and evolve, they will need to develop awareness and implement individual precautions and countermeasures. To ultimately combat widespread attacks, however, collaboration and information sharing will be key to confronting the common security challenges in these environments. Both individualized and collaborative countermeasures are described in Chapter 4.

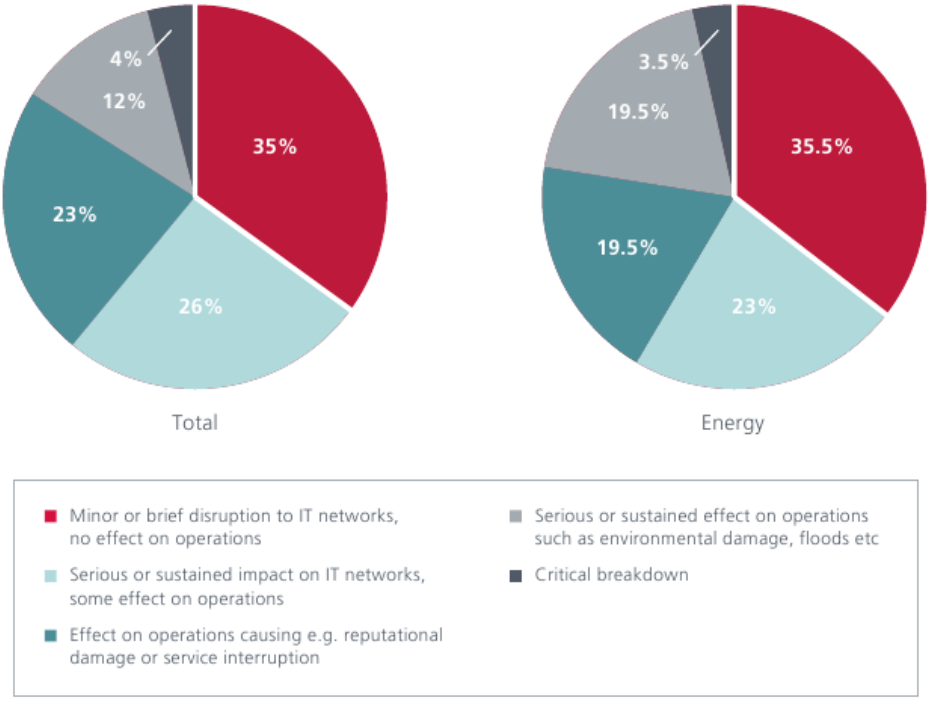


Figure 3.3: Impact of large-scale DDoS against critical infrastructure enterprises [3].

4 Industrial Control Systems Security Management

W32.Stuxnet, albeit the most prominent one, was and will not be the only worm targeting ICS.Based on the identified properties exploited by W32.Stuxnet (see Section 2.3), we discuss general critical properties of ICS in this chapter. Furthermore, we discuss their relevance for operators and which countermeasures should be considered.

4.1 System isolation and vulnerability

Deployed ICSs are highly controlled and their connectivity to the outside world is well defined. It is often assumed that such systems are isolated and not accessible by an adversary. However, as W32.Stuxnet demonstrates, this can be a false assumption. Serious attacks may be enabled as new pathways are exploited that were not previously vulnerable (e.g., USB ports). Although the hardware is proprietary and relatively difficult to exploit ("black box"), its control software often runs on a highly-targeted platform: Microsoft Windows. Sometimes additional attack surfaces are introduced in the process of improving usability (e.g., wireless communication technologies to communicate with remote sensors). An operator should be aware of this issue and consider countermeasures and precautions such as:

- Security evaluation of facilities and networks
- Penetration testing
- Customization of process flows
- Definition of policies
- Introduction of intrusion detection mechanisms (e.g., device specific honeypots)
- System hardening (e.g., hardware lockdown by disabling unnecessary I/O interfaces)

4.2 Factors inherent in the environment

A robust security management strategy must account for inherent domain-specific factors of the industrial process environment, which is described in the following sections.

4.2.1 Maintaining compliance through long system life cycles

Typical ICS deployments have long deployment life cycles, as their return on investment period often exceeds ten years. When the innovation cycle of employed technologies is shortened, the overall system design cannot address vulnerabilities in new technologies. This may also result in some security management aspects being overlooked. Sometimes, systems run unmodified for a long period of time due to the unavailability of updates, regulatory constraints, or an insufficient implementation of best practices. Furthermore, an operator may not be able to change pre-set configurations or hard-coded settings since it could void the warranty or invalidate the vendor's (liability) responsibilities.

There are several standards and guidelines (such as [1, 4, 9, 10, 11, 14, 15, 16]) that might be binding for operators depending on the country and jurisdiction they are operating in. Security-related violations of these standards may lead to liability issues as well as serious impact on business continuity and reputation.

Also, the device technology may age, and security maintenance is usually not foreseen by the manufacturer. An operator should therefore consider:

- Awareness of security management on all levels (system to individual devices)
- Integration of attack detection mechanisms
- Patch management on device level
- Convenient and reliable patch/update management for systems disconnected from commodity networks
- Architectural countermeasures as means of compartmentalization or isolation

Fraunhofer SIT Munich's NES research department can help operators evaluate and implement appropriate concepts and countermeasures to face the security challenges arising from sophisticated, targeted cyber attacks on ICS.

4.2.2 Market pressure

ICSs are an economically attractive market, and as in all technology-oriented markets, the vendor offering solutions first is considered to have a competitive advantage. The paradigm of "be the first" to market is often promoted at the expense of security. An adequate consideration of security mechanisms requires a higher priority from theoutset. As a result, an operator should consider:

- Avoiding security retrofitting
- Influencing the systems' design and development process to incorporate a security model
- Leveraging technologies capable of supporting an existing security framework

4.3 Human factors

For security measures to succeed, they should conform to the facility's operational framework, and must be well understood and accepted by its users. They should also develop emergency response plans.

4.3.1 Security awareness

The propagation of W32.Stuxnet through infected USB devices shows the continued lack of awareness about the dangers involved whenever data is exchanged between ICS and conventional IT networks. Although other malware, such as Conficker, use the same propagation channel, it seems that the lessons learned from incidents in conventional networks have not yet transferred to these specialized systems. Hence, there is the need to train ICS operators to develop a better IT security awareness. Furthermore, operators need to adequately safeguard design and configuration data of their environments, as these were likely used by Stuxnet's authors to target specific end systems (i.e., PLC device models).

Mitigating the overall risk of compromise requires well-defined business processes and security measures. Extra precautions must be taken to avoid the infection of ICS with malware and further threats introduced through other connected networks (directly or indirectly). This risk, however, can never be completely eliminated, so operators should be prepared to react appropriately in the event of a potentially-damaging security incident. These countermeasures include the active development of a crisis management strategy and a business continuity plan.

4.3.2 Staff education

Apart from increasing the awareness for ICS operators, there is a need for special training of staff who work with these systems. Combined with qualified authorisation and auditing mechanisms, this helps mitigate the remaining risk of threats like insider attacks, unintentional misuse, and social engineering attacks. The ISO/IEC 27000 series provides a set of standards for Information Security Management, which supports the risk identification and mitigation process. If an incident is unavoidable, audit data, such as extensive system logs, can help operators analyse the chain of events that led to it and prevent repeat compromises.

4.4 Next-generation event monitoring and information sharing

Reporting security incidents to a higher control instance, and sharing information across operators' administrative domains can facilitate threat identification and

mitigation. Why should technical system borders impede the adoption of new operational improvements? It requires defined processes and information flows to communicate condensed information to single points of contact within an organization or to other stakeholders (such as operators from other plants) who may benefit.

The following factors should be considered:

- Collaboration is cheap: an operator can utilize the resources of others, resulting in a high return on investment
- Sharing information on locally-discovered security issues is not equivalent to an admission of being compromised, which is sensitive information.

Next-generation monitoring also requires consistent policy enforcement, which in turn relies on comprehensive logging and alerting mechanisms. These mechanisms should not be limited to inbound connections, but should cover outbound ones as well, because attackers often need outbound connectivity to complete their tasks (e.g, data exfiltration).

5 Conclusion

The combined technical sophistication and specificity of W32.Stuxnet represent an evolutionary leap in both the development and operation of malware, but only time can tell its full impact on industrial control and conventional IT systems. Since detection signatures and security patches are now widely available, the danger from the specific W32.Stuxnet malware is relatively contained. As we have shown, its virulence is also fundamentally limited in scope as it apparently targets victims running very specific technology stacks. Best practices and an overall security awareness remain paramount to confronting threats of this nature, however.

As with most IT security hazards, there is no single, monolithic solution to fully mitigate W32.Stuxnet in these environments. It targets a highly heterogeneous attack surface comprised of disparate commodity and proprietary computing systems, and demonstrates unprecedented resilience by propagating through both direct and indirect channels.

Generally, a robust patch management policy can sufficiently reduces the vulnerability window to forestall most attacks. In the case of W32.Stuxnet, however, which targeted several unknown (zero-day) vulnerabilities and systems less-frequently updated, many victims were inevitably left unprepared.

Attacks like W32.Stuxnet are best prevented and mitigated through a holistic defense-in-depth information assurance strategy (i.e., not merely at perimeter systems) and ongoing security awareness throughout the system's life cycle.

Fraunhofer SIT Munich can provide expert guidance to operators of IT and ICS environments in evaluating and implementing appropriate security models and protection mechanisms to confront the pervasive security challenges discussed in this paper.

6 Fraunhofer SIT Munich Competencies

The Fraunhofer Institute for Secure Information Technology (SIT) at Munich consists of three research departments and offers competencies in the following areas:

6.1 Network Security and Early Warning Systems (NES)

Due to the growing interconnection of devices concerning ever more parts of life, secure system design as well as the development of methods for malware detection and early warning systems gain relevance in modern networks. The NES research department works on and develops solutions for the secure operation of networks and network infrastructure services, as well as on early warning solutions. Its expertise originating from IT security is often transferred into other domains, such as automotive security and wireless sensor networks.

NES operates a malware collection and analysis laboratory, in which up-to-date information on current attacks is gathered and practical analytics are developed. The laboratory enables safe and thorough evaluations under industrial constraints.

6.2 Embedded Security and Trusted OS (EMS)

The EMS Department researches and develops methods for improving the security of embedded systems, i.e., small electronic devices which usually contain a microprocessor and custom software. In EMS Secure systems are built after evaluation of customers' demands, based on security-enhanced hardware components and innovative software concepts.

The competencies of EMS include both theoretical and practical experience in the analysis of side-channel and fault attacks of embedded systems, and the respective countermeasures. For these activities, EMS has available a dedicated and adequately equipped hardware laboratory.

6.3 Secure Services and Quality Testing (SST)

SST addresses the protective goals of IT security – integrity, availability and confidentiality – within the field of Cloud computing systems. Loss of control as well as a limited availability of the used services are a main concern of Cloud

providers and customers. SST offers concepts, solutions and tools for establishing secure and safe Cloud ecosystems in accordance with Governance-Risk-Compliance (GRC) approach. These competences are complemented with expertise in secure identity management as well as security solutions in distributed systems (Service Oriented Architectures (SOA)).

For practical evaluation the department of SST has established Cloud Computing testing facilities, within which functionality, reliability and interoperability of Cloud systems can be investigated.

6.4 Reference Customers



Bibliography

[1] ANSI/ISA-99.00.01-2007. *Security for Industrial Automation and Control Systems Part 1: Terminology, Concepts, and Models*. International Society of Automation, 2007. 20

[2] E. Bagheri and A.A. Ghorbani. The State of the Art in Critical Infrastructure Protection: a Framework for Convergence. *International Journal of Critical Infrastructures 4(3)*, 2008. 17

[3] S. Baker, S. Waterman, and G. Ivanov. In the crossfire – critical infrastructure in the age of cyber war. Technical report, McAfee, 2009. 16, 17, 18

[4] DNP User Group. *DNP Specification Documents*. DNP User Group, 2010. 20

[5] European Network and Information Security Agency (ENISA). EU Agency analysis of 'stuxnet' malware: a paradigm shift in threats and critical information infrastructure protection. Press Release, Oct 2010. 6

[6] L. Falliere. N.and O'Murchu and E. Chien. W32.stuxnet dossier. White paper version 1.0, Symantec, September 2010. 7, 13

[7] J.H. Graham and S.C. Patel. Security considerations in scada communication protocols. Technical Report TR-ISRL-04-01, Dept. of Computer Engineering and Computer Science,University of Louisville, Sep 2004. 16

[8] J.L. Hieb, J.H. Graham, and S.C. Patel. Cyber security enhancements for scada and dcs systems. Technical Report ISRL-TR-07-02, Dept. of Computer Engineering and Computer Science,University of Louisville, SepOct 2007. 16

[9] A. Hildick-Smith. Security for critical infrastructure scada systems. White paper, SANS Institute, Feb 2005. Version 1.4c. 15, 20

[10] IEC 60870-5-SER. *Telecontrol equipment and systems - Part 5: Transmission protocols*. ISO, Geneva, Switzerland, 2007. 20

[11] IEC 61499. *Function Blocks*. IEC, Geneva, Switzerland, 2005. 20

[12] Red Tiger Security Jonathan Pollet. Electricity for free? the dirty underbelly of scada and smart meters. Blackhat USA, June 2010. 16

[13] Felix Leder and Tillmann Werner. Know your enemy: Containinig conficker. Whitepaper, The Honeynet Project, April 2009. rev.2. 9

[14] National Institute of Standards and Technology. Draft guide to industrial control systems (ics) security. http://csrc.nist.gov/publications/drafts/800-82/draft_sp800-82-fpd.pdf. 20

[15] National Institute of Standards and Technology. Recommended security controls for federal information systems and organizations. http://csrc.nist.gov/publications/nistpubs/800-53-Rev3/sp800-53-rev3-final_updated-errata_05-01-2010.pdf. 20

[16] National Institute of Standards and Technology. System protection profile - industrial control systems. http://www.isd.mel.nist.gov/documents/stouffer/NISTIR_7176.pdf. 20

[17] E.H. Spafford. Remembrances of things pest. *Commun. ACM*, 53:35–37, August 2010. 15

Impressum:

Picture on title page: fotolia (AndreasG), stock.xchng (dlritter)
Layout: Katharina Franz, Fraunhofer SIT