CHAPTER

Hacking Industrial Control Systems

7

INFORMATION IN THIS CHAPTER

- Motives and Consequences
- Common Industrial Targets
- Common Attack Methods
- Examples of Advanced Industrial Cyber Threats
- Attack Trends
- Dealing with an Infection

MOTIVES AND CONSEQUENCES

Industrial networks are responsible for continuous and batch processing and other manufacturing operations of almost every scale, and as a result the successful penetration of a control system network can be used to directly impact those operations. Consequences vary and can range from relatively benign disruptions, such as the interruption of the operation (taking a facility offline), the alteration of an operational process (changing the formula of a chemical process or recipe), to deliberate acts of sabotage that are intended to cause harm. Manipulating the feedback loop of certain processes could, for example, cause pressure within a boiler to build beyond safe operating parameters. Cyber sabotage, on the other hand, can result in environmental damage (oil spill, fire, toxic release, etc.), injury or loss of life, the loss of critical services (blackouts, disruption in fuel supplies, unavailability of vaccines, etc.), or potentially catastrophic explosions.

CONSEQUENCES OF A SUCCESSFUL CYBER INCIDENT

A successful cyber-attack on an ICS can have many undesirable consequences, including

- Delay, block, or alter the intended process, that is, alter the amount of energy produced at an electric generation facility.
- Delay, block, or alter information related to a process, thereby preventing a bulk energy provider from obtaining production metrics that are used in energy trading or other business operations.

FIGURE 7.1 Consequences of a compromised industrial control system.

- Unauthorized changes to instructions or alarm thresholds that could damage, disable or shutdown mechanical equipment, such as generators or substations.
- Inaccurate information sent to operators could either be used to disguise unauthorized changes (see Stuxnet later in this chapter), or cause the operator to initiate inappropriate actions.

The end result could be anything from financial loss to physical safety liabilities, with impacts extending beyond the plant, to the local community, state, and even federal level (see Figure 7.1). Companies can incur penalties for regulatory noncompliance or they may suffer financial impact from lost production hours due to misinformation or denial of service. An incident can impact the ICS in almost any way, from taking a facility offline, disabling or altering safeguards, to life-threatening incidents within the plant—up to and including the release or theft of hazardous materials or direct threats to national security.[1] The possible damages resulting from a cyber incident varies depending upon the type of incident, as shown in Table 7.1.

CYBER SECURITY AND SAFETY

Most industrial networks employ automated safety systems to avoid catastrophic failures. However, many of these safety controls employ the same messaging and control protocols used by the industrial control network's operational processes, and in some cases, such as certain fieldbus implementations, the safety systems are supported directly within the same communications protocols as the operational

Table 7.1 The Potential Impact of Successful Cyber-Attacks

Incident Type	Potential Impact
Change in a system, operating system, or application configuration	Command and control channels introduced into otherwise secure systems Suppression of alarms and reports to hide malicious activity Alteration of expected behavior to produce unwanted and unpredictable results
Change in programmable logic in PLCs, RTUs, or other controllers	Damage to equipment and/or facilities Malfunction of the process (shutdown) Disabling control over a process
Misinformation reported to operators	Inappropriate actions taken in response to misinformation that could result in a change to operational parameters Hiding or obfuscating malicious activity, including the incident itself or injected code
Tampering with safety systems or other controls	Preventing expected operations, fail safes, and other safeguards with potentially damaging consequences
Malicious software (malware) infection	Initiation of additional incident scenarios Production impact resulting from assets taken offline for forensic analysis, cleaning, and/or replacement Assets susceptible to further attacks, information theft, alteration, or infection
Information theft	Leakage of sensitive information such as a recipe or chemical formula
Information alteration	Alteration of sensitive information such as a recipe or chemical formula in order to sabotage or otherwise adversely affect the manufactured product

controls on the same physical media (see Chapter 4, "Industrial Network Protocols," for details and security concerns of industrial control protocols).

NOTE

Critical, risk-based safety operations implemented within the ICS typically follow separate standards regarding the use of programmable logic solvers, field devices, and communication protocols (e.g. IEC 61508/61511, NFPA 85, ISA 84) and how these Safety Instrumented Systems (SIS) can be interfaced and integrated with other ICS components. It is important to realize that not all "safety" controls and interlocks are implemented against these standards, and that it is possible for these systems to share infrastructure (including the controller platform itself) with other ICS systems and components. Regulatory requirements typically require standards-based SIS implementations for safety functions that represent significant unmitigated risk in terms of human health, safety, and environmental impact, and not on production uptime or reliability.

Although safety systems are extremely important, there is the perception that they have been used to downplay the need for heightened security of industrial networks. Research has shown that real consequences can occur in modeled systems. Simulations performed by the Sandia National Laboratories showed that simple man-in-the-middle (MitM) attacks could be used to change values in a control system and that a modest-scale attack on a larger bulk electric system using targeted malware (in this scenario, targeting specific ICS front-end processors) was able to cause significant loss of generation.[2]

The European research team VIKING (Vital Infrastructure, Networks, Information and Control Systems Management) is currently investigating threats of a different sort. The Automatic Generation Control (AGC) system within the electric power network is responsible for adjusting the output of multiple generators on the grid in response to changes in demand. It operates autonomously from human interaction—that is, output actions are based entirely on processing of input states with the logic of the AGC. Rather than breaching a control system through the manipulation of an HMI, VIKING's research attempts to investigate whether the manipulation of input data could alter the normal control loop functions, ultimately causing a disturbance.[3]

TIP

Think of security as separate from safety when establishing a cyber security plan. Do not assume that security leads to safety or that safety leads to security. If an automated safety control is compromised by a cyber-attack (or otherwise disrupted), the necessity of having a strong digital defense against the manipulation of operations becomes even more important. Likewise, a successful safety policy should not rely on the security of the networks used. Both systems will be inherently more reliable by planning for safety and security controls that operate independently of one another. At the same time, safety systems are built around strong process assessments, to protect against identified physical risk conditions. These risk conditions may be the ultimate goal of a cyber-attack, and so safety and security also need to work together within an organization to ensure that cyber defenses are properly implemented.

COMMON INDUSTRIAL TARGETS

Industrial control systems may be comprised of similar components; however, each system is unique in terms of the exact composition, quantity, and criticality of these components. There are, however, some common targets within industrial networks despite these system differences. These include network services, such as Active Directory (directory services) and Identity and Access Management (IAM) servers, which may be shared between business and industrial zones (though the best practice is to not share these services!); engineering workstations, which can be used to exfiltrate, alter or overwrite process logic; operator consoles, which can be used to trick human operators into performing unintended tasks; and of course the industrial applications (SCADA server, historian, asset management, etc.) and protocols (Modbus, DNP3, EtherNet/IPI, etc.) themselves, which can be used to alter, manipulate, blind, or destroy almost any aspect of an ICS. Table 7.2 highlights some of the

Table 7.2 Attack Targets

Target	Possible Attack Vectors	Possible Attack Methods	Possible Consequences
Access control system	- Identification cards - Closed-circuit television (CCTV) - Building management network - Software vendor support portal	- Exploitation of unpatched application (building management system) - RFID spoofing - Network access through unprotected access points - Network pivoting through unregulated network boundaries	- Unauthorized physical access - Lack of (video) detection capabilities - Unauthorized access to additional ICS assets (pivoting)
Analyzers/analyzer management system	- Subcontractor Laptop - Maintenance Remote Access - Plant (analyzer) network	- Exploitation of unpatched application - Network access via insecure access points (analyzer shelters) - Remote Access VPN via stolen or compromised subcontractor laptop - Remote Access VPN via compromise of maintenance vendor site - Insecure implementation of OPC (communication protocol)	- Product quality - spoilage, loss of production, loss of revenue - Reputation - product recall, product reliability
Application servers	- Remote user access (interactive sessions) - Business application integration communication channel - Plant network - Software vendor support portal	- Exploitation of unpatched application - Installation of malware via unvalidated vendor software - Remote access via “interactive” accounts - Database injection - Insecure implementation of OPC (communication protocols)	- Plant upset / shutdown - Credential leakage (control) - Sensitive / confidential information leakage - Unauthorized access to additional ICS assets (pivoting)

(Continued)

Table 7.2 Attack Targets (*cont.*)

Target	Possible Attack Vectors	Possible Attack Methods	Possible Consequences
Asset management system	- Plant Maintenance Software / ERP - Database integration functionality - Mobile devices used for device configuration - Wireless device network - Software vendor support portal	- Exploitation of unpatched application - Installation of malware via unvalidated vendor software - Remote access via “interactive” accounts - Database injection - Installation of malware via mobile devices - Access via insecure wireless infrastructure	- Calibration errors - product quality - Credential leakage (business) - Credential leakage (control) - Unauthorized access to additional business assets like plant maintenance / ERP (pivoting) - Unauthorized access to additional ICS assets (pivoting)
Condition monitoring system	- Subcontractor Laptop - Maintenance Remote Access - Plant (maintenance) network - Software vendor support portal	- Exploitation of unpatched application - Installation of malware via unvalidated vendor software - Network access via unsecure access points (compressor / pump house) - Remote Access VPN via stolen or compromised subcontractor laptop - Remote Access VPN via compromise of maintenance vendor site - Remote access via “interactive” accounts - Database injection - Insecure implementation of OPC (communication protocols)	- Equipment damage / sabotage - Plant upset / shutdown - Unauthorized access to additional ICS assets (pivoting)

Controller (PLC)	- Engineering workstation - Operator HMI - Standalone engineering tools - Rogue device in Control Zone - USB / removable media - Controller network - Controller (device) network	- Engineer / technician misuse - Network exploitation of industrial protocol - known vulnerability - Network exploitation of industrial protocol - known functionality - Network replay attack - Network DoS via communication buffer overload - Direct code / malware injection via USB - Direct access to device via rogue network (local / remote) PC with appropriate tools / software	- Manipulation of controlled process(es) - Controller fault condition - Manipulation / masking of input / output data to / from controller - Plant upset / shutdown - Command-and-control
Data historian	- Business network client - ERP data integration communication channel - Database integration communication channel - Remote user access (interactive session) - Plant network - Software vendor support portal	- Exploitation of unpatched application - Installation of malware via unvalidated vendor software - Remote access via “interactive” accounts - Database injection - Insecure implementation of required communication protocols - Exploitation of unnecessary / excessive openings on perimeter defense (firewall) due to insecure communication infrastructure between applications	- Manipulation of process / batch records - Credential leakage (business) - Credential leakage (control) - Unauthorized access to additional business assets like MES, ERP (pivoting) - Unauthorized access to additional ICS assets (pivoting)

(Continued)

Table 7.2 Attack Targets (*cont.*)

Target	Possible Attack Vectors	Possible Attack Methods	Possible Consequences
Directory services	- Replication services - Print spooler services - File sharing services - Authentication services - Plant network - Software vendor support portal	- Exploitation of unpatched application(s) - Installation of malware via unvalidated vendor software - DNS spoofing - NTP Reflection attack - Exploitation of unnecessary / excessive openings on perimeter defense (firewall) due to replication requirements between servers - Installation of malware on file shares	- Communication disruptions via DNS - Authentication disruptions via NTP - Authentication disruptions via LDAP / Kerberos - Credential leakage - Information leakage - file shares - Malware distribution - Unauthorized access to ALL domain-connected ICS assets (pivoting) - Unauthorized access to business assets (pivoting)
Engineering workstations	- Engineering tools and applications - Non-engineering client applications - USB / Removable media - Elevated privileges (engineer / administrator) - Control network - Software vendor support portal	- Exploitation of unpatched applications - Installation of malware via unvalidated vendor software - Installation of malware via removable media - Installation of malware via keyboard - Exploitation of trusted connections across security perimeters - Authorization to ICS applications without sufficient access control mechanisms	- Plant upset / shutdown - Delay plant startup - Mechanical damage / sabotage - Unauthorized manipulation of operator graphics - inappropriate response to process action - Unauthorized modification of ICS database(s) - Unauthorized modification of critical status / alarms - Unauthorized distribution of faulty firmware - Unauthorized startup / shutdown of ICS devices

			- Process / plant information leakage - ICS design / application credential leakage - Unauthorized modification of ICS access control mechanisms - Unauthorized access to most ICS assets (pivoting / own) - Unauthorized access to business assets (pivoting)
Environmental controls	- HVAC control - HVAC (building management) network - Software vendor support portal	- Exploitation of unpatched application (building management system) - Installation of malware via unvalidated vendor software - Network access through unprotected access points - Network pivoting through unregulated network boundaries	- Disruption of cooling / heating - Equipment failure / shutdown
Fire detection and suppression system	- Fire alarm / evaluation - Fire suppressant system - Building management network - Software vendor support portal	- Exploitation of unpatched application (building management system) - Installation of malware via unvalidated vendor software - Network access through unprotected access points - Network pivoting through unregulated network boundaries	- Unauthorized release of suppressant - Equipment failure / shutdown

(Continued)

Table 7.2 Attack Targets (*cont.*)

Target	Possible Attack Vectors	Possible Attack Methods	Possible Consequences
Master and/or slave devices	- Unauthorized / Unvalidated firmware - Weak communication problems - Insufficient authentication for "write" operations - Control network - Device network	- Distribution of malicious firmware - Exploitation of vulnerable industrial protocols via rogue PC on network (local / remote) - Exploitation of vulnerable industrial protocols via compromised PC on network (local) - Exploitation of industrial protocol functionality via rogue PC on network (local / remote) - Exploitation of industrial protocol functionality via compromised PC on network (local) - Communication buffer overflow via rogue PC on network (local / remote) - Communication buffer overflow via compromised PC on network (local)	- Plant upset / shutdown - Delay plant start - Mechanical damage / sabotage - Inappropriate response to control action - Suppression of critical status / alarms
Operator workstation (HMI)	- Operational applications (HMI) - non-SCADA client applications - USB / Removable media - Elevated privileges (administrator) - Control network - Software vendor support portal	- Exploitation of unpatched applications - Installation of malware via unvalidated vendor software - Installation of malware via removable media - Installation of malware via keyboard - Authorization to ICS HMI functions without sufficient access control mechanisms	- Plant upset / shutdown - Suppression of critical status / alarms - Product quality - Plant / process efficiency - Credential leakage (control) - Plant / operational information leakage - Unauthorized access to ICS assets (pivoting) - Unauthorized access to ICS assets (communication protocols)

Patch management servers	- Software patches / hotfixes - Patch management software - Vendor software support portal - Business network - Plant network - Software vendor support portal	- Insufficient checking of patch "health" before deployment - Alternation of automatic deployment schedule - Installation of malicious software via trusted (supplier) media - Installation of malware via unvalidated vendor software	- Malware distribution server - Unauthorized modification of patch schedule - Credential leakage - Unauthorized access to ICS assets (pivoting)
Perimeter protection (firewall/IPS)	- Trusted connections (Business-to-Control) - Local user account database - Signature / rule updates	- Untested/unverified rules - Exploitation of unnecessary / excessive openings on perimeter defense (firewall) - Insecure office and industrial protocols allowed to cross security perimeter - Reuse of credentials across boundary	- Unauthorized access to business network - Unauthorized access to DMZ network - Unauthorized access to control network - Local credential leakage - Unauthorized modification of rulesets / signatures - Communication disruption across perimeter / boundary
SCADA servers	- Non-SCADA client applications - Application integration communication channels - Data historian - Engineering Workstation - Control network - Software vendor support portal	- Exploitation of unpatched applications - Installation of malware via unvalidated vendor software - Remote access via "interactive" accounts - Installation of malware via removable media - Exploitation of trusted connections within control network - Authorization to ICS applications without sufficient access control mechanisms	- Plant upset / shutdown - Delay plant startup - Mechanical damage / sabotage - Unauthorized manipulation of operator graphics - inappropriate response to process action - Unauthorized modification of ICS database(s) - Unauthorized modification of critical status / alarms - Unauthorized startup / shutdown of ICS devices

(Continued)

Table 7.2 Attack Targets (*cont.*)

Target	Possible Attack Vectors	Possible Attack Methods	Possible Consequences
			- Credential leakage (control) - Plant / operational information leakage - Unauthorized modification of ICS access control mechanisms - Unauthorized access to most ICS assets (pivoting / own) - Unauthorized access to ICS assets (communication protocols) - Unauthorized access to business assets (pivoting)
Safety systems	- Safety engineering tools - Plant / emergency shutdown communication channels (DCS / SCADA) - Control (safety) network - Software vendor support portal	- Exploitation of unpatched applications - Installation of malware via unvalidated vendor software - Installation of malware via removable media - Installation of malware via keyboard - Authorization to ICS applications without sufficient access control mechanisms	- Plant shutdown - Equipment damage / sabotage - Environmental impact - Loss of life - Product quality - Company reputation
Telecommunications systems	- Public key infrastructure - Internet visibility	- Disclosure of private key via external compromise - Exploitation of device "unknowingly" connected to public networks - Network access through unmonitored access points - Network pivoting through unregulated network boundaries	- Credential leakage (control) - Information leakage - Unauthorized remote access - Unauthorized access to ICS assets (pivoting) - Command and control

Uninterruptible power systems (UPS)	- Electrical management network - Vendor / subcontractor maintenance	- Exploitation of unpatched application (building management system) - Installation of malware via unvalidated vendor software - Network access through unprotected access points - Network pivoting through unregulated network boundaries	- Equipment failure / shutdown - Plant upset / shutdown - Credential leakage - Unauthorized access to ICS assets (pivoting)
User – ICS engineer	- Social engineering - Corporate assets - Social engineering - Personal assets - E-mail attachments - File shares	- Introduction of malware through watering hole or spear-phishing attack on business PC - Introduction of malware via malicious email attachment on business PC from trusted source - Introduction of malware on control network via unauthorized / foreign host - Introduction of malware on control network via shared virtual machines - Introduction of malware via inappropriate use of removable media between security zones (home - business - control) - Propagation of malware due to poor segmentation and “full visibility” from EWS - Establishment of C2 via inappropriate control-to-business (outbound) connections	- Process / plant information leakage - ICS design / application credential leakage - Unauthorized access to business assets (pivoting) - Unauthorized access to ICS assets (pivoting / own)

(Continued)

Table 7.2 Attack Targets (*cont.*)

Target	Possible Attack Vectors	Possible Attack Methods	Possible Consequences
		- Exploitation of communication channels resulting from unapproved architecture changes - Exploitation of applications due to unnecessary use of administrative rights - Exploitation of applications due to failure to logout / disconnect when unused	
User – ICS technician	- Social engineering - Corporate assets - Social engineering - Personal assets - E-mail attachments - File shares	- Introduction of malware on control network via connection of unauthorized / foreign host - Introduction of malware on control network via shared virtual machines - Introduction of malware via inappropriate use of removable media between security zones (home - business - control) - Exploitation of applications due to unnecessary use of administrative rights - Network disturbances resulting from connection to networks with poor segmentation	- Plant upset / shutdown - Delay plant startup - Mechanical damage / sabotage - Unauthorized manipulation of operator graphics - inappropriate response to process action - Unauthorized modification of ICS database(s) - Unauthorized modification of status / alarms settings - Unauthorized download of faulty firmware - Unauthorized startup / shutdown of ICS devices - Design information leakage - ICS application credential leakage - Unauthorized access to most ICS assets (pivoting / own)

Users – plant operator	- Keyboard - Removable media - USB - Removable Media - CD / DVD	- Introduction of malware on control network via unauthorized / foreign host - Introduction of malware via inappropriate use of removable media between security zones (home - business - control) - Exploitation of applications due to unnecessary use of administrative rights	- Plant upset / shutdown - Mechanical damage / sabotage - Unauthorized startup/shutdown of mechanical equipment - Process / plant operational information leakage - Credential leakage - Unauthorized access to ICS assets (pivoting) - Unauthorized access to ICS assets (communication protocols)

common targets, how they are likely to be attacked, and what the consequences of such attacks might be.

COMMON ATTACK METHODS

There are many methods of attacking a target, once a target has been identified. MitM, Denial-of-Service (DoS), Replay attacks, and countless more methods all remain very effective in industrial networks. The primary reason for this is a combination of insecure communication protocols, little device-to-device authentication, and delicate communication stacks in embedded devices. If an industrial network can be penetrated and malware deposited (on disk or in memory) anywhere on the network, tools such as Metasploit Meterpreter shell can be used to provide remote access to target systems, install keyloggers or keystroke injectors, enable local audio/video resources, manipulate control bits within industrial protocols, plus many other covert capabilities.

In some cases, the information that is available can be used as reconnaissance for further cyber-attack capability. In many cases, systems can be attacked directly using disclosed exploits, with only basic system knowledge required. If an attack is successful, persistence can often be established, enabling an attacker to gather intelligence over time. In systems that make up a nexus between other systems (such as a control room SCADA server), a persistent presence can also be used to launch secondary attacks against other portions of the industrial network—such as basic control and process control zones that reside within the supervisory zone.

It is important to understand at this point the difference between *compromising* or "owning" a target, and *attacking* a target. There is no formal definition that defines either, but for the purposes of this book, a compromise can be thought of as the ability to exploit a target and perform an *unknown* action (such as running a malicious payload). An attack, on the other hand, can be thought of as causing a target to perform an *undesirable* action. In this case, the device may be performing as designed, yet the ability to attack the device and cause it to perform an action that is not desired by the engineer may lead to negative consequences. Many ICS devices can therefore be attacked via the *exploitation of functionality* versus the *exploitation of vulnerabilities*. In other words, issuing a "shutdown" command to a control device does not represent any particular weakness in the device *per se*. However, if the lack of authentication enables a malicious user to inject a shutdown command (i.e. perform a replay attack), this is a major vulnerability.

MAN-IN-THE-MIDDLE ATTACKS

A man-in-the-middle attack refers to an attack where the attacker goes between communicating devices and snoops the traffic between them. The attacker is actually connecting to both devices, and then relaying traffic between them so that it appears that they are communicating directly, even though they are really communicating through

a third device that is eavesdropping on the interaction. To perform a MitM attack, the attacker must be able to intercept traffic between the two target systems and inject new traffic. If the connection lacks encryption and authentication—as is often the case with industrial protocol traffic—this is a very straightforward process. Where authentication or encryption are used, an MitM attack can still succeed by listening for key exchanges and passing the attacker's key in place of a legitimate key. This attack vector is somewhat complicated in industrial networks because devices can communicate via sessions that are established and remain intact for long periods of time. The attacker would have to first hijack an existing communication session. The biggest challenge to a successful MitM attack is successfully inserting oneself into the message stream, which requires establishing trust. In other words, the attacker needs to convince both sides of the connection that it is the intended recipient. This impersonation can be thwarted with appropriate authentication controls. Many industrial protocols unfortunately authenticate in clear text (if at all), facilitating MitM attacks within the various industrial control systems.

DENIAL-OF-SERVICE ATTACKS

Denial-of-service attacks occur when some malicious event attempts to make a resource unavailable. This is a very broad category of attacks, and can include anything from loss of communications with the device, to inhibiting or crashing particular services within the device (storage, input/output processing, continuous logic processing, etc.). DoS attacks in traditional business systems do not typically result in significant negative consequences if resolved in a timely manner. Access to a web page may be slowed, or email delivery delayed until the problem is resolved. However, while there are rarely physical consequences associated with the interruption of services, a well-targeted DoS could bring very important systems off-line, and could even trigger a shutdown.

Automation systems are deployed to monitor or control a physical process. This process could be controlling the flow of crude oil in a pipeline, converting steam into electricity, or controlling ignition timing in an automobile engine. The inability of a controller such as an SIS to perform its action is commonly called "Loss of Control (LoC)" and typically results in the physical process being placed in a "safe" state—shutdown! This means that even simple disruptions of control functions can quickly translate into physical plant disturbances that can further lead to environmental releases, plant shutdowns, mechanical failure, or other catastrophic events. In the case of the HMI, it is not directly connected to the mechanical equipment; however, in many manufacturing industries, the inability of the HMI to perform its function can lead to "Loss of View (LoV)," which often requires the manufacturing process to be shut down if view of data cannot be restored in a timely manner. In the case of an automobile's ignition control system, if the controller stops performing, the engine stops running!

A hacker typically does not boast of a DoS attack on an Internet-facing website (unless you are part of a hacktivist group), but because a DoS can result in LOV or

LOC, a similar DoS attack on an ICS can lead to far greater consequences: an oil spill, a plant fire and explosion, or spoiled batches of products. Denial of service in industrial environments is much more than an inconvenience, but can lead to significant consequences if not managed accordingly.

REPLAY ATTACKS

Initiating specific process commands into an industrial protocol stream requires an in-depth knowledge of industrial control system operations. It is possible to capture packets and simply replay them to inject a desired process command into the system because most industrial control traffic is transmitted in plain text. When capturing packets in a lab environment, a specific command can be initiated through a console, and the resulting network traffic captured. When these packets are replayed, they will perform the same command. When commands are in clear text, it is simple to find and replace a command from within captured traffic to create custom packets that are crafted to perform specific tasks. If traffic is captured from the field, authentication mechanisms (symmetric encryption, challenge-response, cleartext exchange, etc.) can be captured as well allowing an attacker to authenticate to a device via a replay attack, providing an authorized connection through which additional recorded traffic can be played back. This capability is actually part of many open-source and licensed industrial protocols and is why this can best be referred to as *exploitation of functionality*. If the device is a PLC or other process automation controller, such as the controller functions found in more advanced substation gateways, the behavior of an entire system could be altered. If the target is an IED, specific registers could be overwritten to inject false measurements or readings into a system.

Security researcher Dillon Beresford demonstrated a PLC replay attack at the 2011 Black Hat conference in Las Vegas, NV. The attack began by starting a Siemens SIMATIC STEP 7 engineering console and connecting to a PLC within a lab environment. Various commands were then initiated to the PLC via the STEP 7 console while traffic was being captured. This traffic included a valid STEP 7 to PLC session initiation, allowing the recorded traffic to be played back against any supported PLC to replay those same commands in the field.[4]

Replay attacks are useful because of the command-and-control nature of an ICS. A replay attack can easily render a target system helpless because commands exist to enable or disable security, alarms, and logging features. Industrial protocols also enable the transmission of new programmable code (for device firmware and control logic updates), allowing a replay attack to act as a "dropper" for malicious logic or malware. Researcher Ralph Langner described how simple it could be to write malicious ladder logic at the 2011 Applied Control Systems Cyber Security Conference. He was able to inject a time-bombed logic branch with just 16 bytes of code that was inserted at the front of existing control logic that will place the target PLC into an endless loop—preventing the remaining logic from executing and essentially "bricking" the PLC.[5]

For the subtle manipulation of industrial systems and automation processes, knowledge of specific ICS operations is required. Much of the information needed to attack a PLC can be obtained from the device itself. For example, in Beresford's example, packet replay was used to perform a PLC scan. Using SIMATIC requests to probe a device, Beresford was able to obtain the model, network address, time of day, password, logic files, tag names, data block names, and other details from the targeted PLC.[6]

If the goal is simply to sabotage a system, almost anything can be used to disrupt operations—a simple replay attack to flip the coils in a relay switch is enough to break most processes.[7] In fact, malware designed to flip specific bits could be installed within ICS assets to manipulate or sabotage a given process with little chance of detection. If only read values are manipulated, the device will report false values; if write commands are also manipulated, it would essentially render the protocol functionality useless for that device.

COMPROMISING THE HUMAN–MACHINE INTERFACE

One of the easiest ways to obtain unauthorized command and control of an ICS is to leverage the capabilities of a human–machine interface (HMI) console. Whether an embedded HMI within a control zone, or the centralized command and control capability of DCS, SCADA, EMS or other systems, the most effective way to manipulate those controls is via their console interface. Rather than attacking via the industrial network using MitM or Replay attacks, a known device vulnerability is exploited to install remote access to the console leading to a host *compromise*. One example would be to use the Metasploit framework or similar penetration testing tool to exploit the target system, and then using the Meterpreter shell to install a remote VNC server. Now, the HMI, SCADA, or EMS console is fully visible to and controllable by the attacker. This allows the hacker to directly monitor and control whatever that console is responsible for, remotely. There is no knowledge of industrial protocols needed, no specific experience in ladder logic, or control systems operations—only the ability to interpret a graphical user interface, click buttons, and change values within a console that is typically designed for ease of use.

COMPROMISING THE ENGINEERING WORKSTATION

The vectors used to compromise an Engineering Workstation (EWS) are not much different from those used previously with the HMI. The same vulnerabilities often apply, because the system is managed consistently across all hosts. The same payloads (Meterpreter) can also be used to establish C2 functionality. What is important to consider in this case is the relative value of the logical assets contained on the EWS versus those on the HMI. The HMI does provide bidirectionality read/write capability with the process under control; however, many systems today incorporate role-based access control that may limit the extent of these functions in a distributed architecture consisting of multiple operators and multiple plant areas or units.

The EWS on the other hand, is typically the single host that not only possesses the capability to configure such role-based access control mechanisms, but also the specialized tools needed to directly communicate with, configure, and update the primary control equipment (PLC, BPCS, SIS, IED, etc.). It is also common for the EWS to contain significant amounts of sensitive documentation specific to the ICS design, configuration, and plant operation, making this target a much higher-valued asset than a typical HMI.

BLENDED ATTACKS

Many attacks are more than single exploits against a single vulnerability on a single target. Sophisticated attacks commonly use a blended threat model. According to SearchSecurity, "a blended threat is an exploit that combines elements of multiple types of malware and usually employs multiple attack vectors to increase the severity of damage and the speed of contagion."[8]

In the past, blended attacks typically contained multiple types of malware that were used in succession—a spear phishing attack to access systems behind a firewall that would drop a remote access toolkit (RAT), and then obtain the credentials needed to access the trusted industrial networks, where targets may be compromised or exploited further.

Recently, blended threats have evolved to a much greater degree of complexity. This was first observed with Stuxnet where a single complex and mutating malware framework was deployed that was capable of behaving in multiple ways depending upon its environment. This concept has now been taken even further, with the discovery of Skywiper (also known as Flame), and other complex malware variants.

EXAMPLES OF WEAPONIZED INDUSTRIAL CYBER THREATS

Cyber-attacks against industrial networks were, at one time, purely theoretical. We have now seen real cyber-attacks targeting actual industrial systems. The first documented ICS cyber-attack "in the wild" was Stuxnet discovered in 2010, which was followed shortly by a string of incidents over the next few years. While many high-profile incidents occurred, often targeting the oil industry and countries of the Middle East, Stuxnet remains a strong example of what a modern, weaponized industrial cyber-attack looks like. Stuxnet was very precise, sabotaging specific ICS devices to obtain a specific goal. Shortly after Stuxnet, Shamoon (also DistTrack) and Flame (also called Flamer or Skywiper) surfaced. Shamoon was widely publicized due to its highly destructive nature. Rather than performing a precision attack against target devices, like Stuxnet, Shamoon spread promiscuously and wiped systems clean, incurring huge impact to the computing infrastructure of infected companies. Flame showed signs of being a derivative of Stuxnet, with even greater sophistication. However, the intention of Flame seems to be espionage rather than sabotage or the direct destruction of target systems.

STUXNET

Stuxnet is the poster-child of industrial malware. When discovered, it was the first real example of weaponized computer malware, which began to infect ICSs as early as 2007.[9] Any speculation over the possibility of a targeted cyber-attack against an industrial network has been overruled by this extremely complex and intelligent collection of malware. Stuxnet is a tactical nuclear missile in the cyber war arsenal. It was not just a "shot across the bow," but rather it hit its mark and left behind the proof that extremely complex and sophisticated attacks can and do target industrial networks. The worst-case scenario has now been realized—industrial vulnerabilities have been targeted and exploited by a sophisticated threat actor more commonly called an Advanced Persistent Threat (APT).

Although early versions of Stuxnet were released as early November 2007,[10] widespread discussions about it did not occur until the summer of 2010, after an Industrial Control Systems Cyber Emergency Response Team (ICS-CERT) advisory was issued.[11] Stuxnet was armed with four zero-days in total at its disposal. Stuxnet was able to infect Windows-based computers covering four generations of kernels from Windows 2000 up to and including Windows 7/Server 2008R2. The primary target was a system comprising Siemens SIMATIC WinCC and PCS7 software along with specific models of S7 PLCs utilizing the PROFIBUS protocol to communicate with two specific vendors of variable frequency drives (VFD). These VFDs were used to control the centrifuges used in the process of enriching uranium.[12] (PROFIBUS is the industrial protocol used by Siemens and was covered in Chapter 6, "Industrial Network Protocols".) The subsequent steps taken by the malware depend on what software was installed on the infected host. If the host was not the intended target, the initial infection would load a rootkit that would automatically load the malware at boot and allow it to remain undetected. It then would deploy up to seven different propagation methods to infect other targets. For those methods using removable media, the malware would automatically remove itself after the media infected three new hosts. If the target contained Siemens SIMATIC software, methods existed to exploit default credentials in the SQL Server application allowing the malware to install itself in the WinCC database, or to copy itself into the STEP 7 project file used to program the S7 PLCs. It also had the ability to overwrite a critical driver used to communicate with the S7 PLCs effectively creating a MitM attack allowing the code running in the PLC to be altered without detection by the system users.

Although little was known at first, Siemens effectively responded to the issue, quickly issuing a security advisory, as well as a tool for the detection and removal of Stuxnet. Stuxnet drew the attention of the mass media through the fall of 2010 for being the first threat of its kind—a sophisticated and blended threat that actively targets ICS—and it immediately raised the industry's awareness of advanced threats by illustrating exactly why industrial networks need to dramatically improve their security measures.

Dissecting Stuxnet

Stuxnet is very complex, as can be seen by the Infection Process shown in Figure 7.2. It was used to deliver a payload targeting not only a specific control system, but also

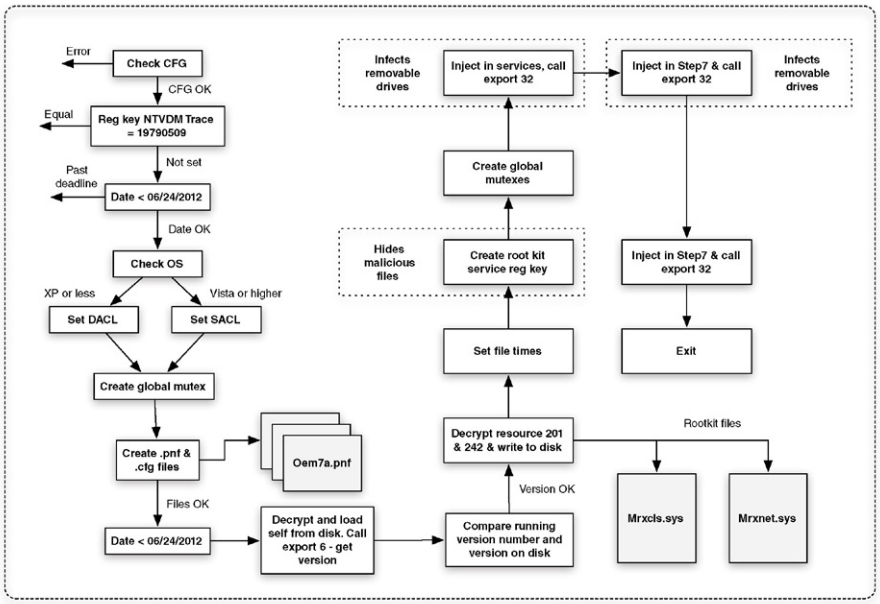

FIGURE 7.2 Stuxnet's infection processes.

a specific configuration of the control system including unique model numbers of PLCs and vendors of field-connected equipment. It is the first rootkit targeting ICS. It can self-update even when cut off from the C2 servers (which is necessary should it find its way into a truly air-gapped system) by enumerating and remembering a complex peer-to-peer network necessary to allow external access. It is able to inject code into the PLCs, and at that point alter the operations of the PLC as well as hide itself by reporting false information back to the HMI. It adapts to its environment. It uses system-level, hard-coded authentication credentials that were publicly disclosed as early as 2008[13] (indications exist that it was disclosed within the Siemens Support portal as early as 2006[14]). It was able to install malicious drivers undetected by Windows through the use of two different legitimate digital certificates manufactured using stolen keys. There is no doubt about it at this time—Stuxnet is an advanced new weapon in the cyber war.

What it Does

The full extent of what Stuxnet is capable of doing is not known at the time of this writing. What we do know is that Stuxnet does the following:[15]

- Infects Windows systems using a variety of zero-day exploits and stolen certificates, and installing a Windows rootkit on compatible machines.
- Attempts to bypass behavior-blocking and host intrusion-protection-based technologies that monitor LoadLibrary calls by using special processes to load any required DLLs, including injection into preexisting trusted processes.

- Typically infects by injecting the entire DLL into another process and only exports additional DLLs as needed.
- Checks to make sure that its host is running a compatible version of Windows, whether or not it is already infected, and checks for installed **Anti-Virus** before attempting to inject its initial payload.
- Spreads laterally through infected networks, using removable media, network connections, print services, WinCC databases, and/or Step 7 project files.
- Looks for target industrial systems (Siemens SIMATIC WinCC/PCS7). When found, it injects itself into an SQL database (WinCC) or project file (Step 7), and replaces a critical communication driver that will facilitate authorized and undetected access to target PLCs.
- Looks for target system configuration (S7-315-2/S7-417 PLC with specific PROFIBUS VFD). When found, it injects code blocks into the target PLCs that can interrupt processes, inject traffic on the Profibus-DP network, and modify the PLC output bits, effectively establishing itself as a hidden rootkit that can inject commands to the target PLCs.
- Uses infected PLCs to watch for specific behaviors by monitoring PROFIBUS.
- If certain frequency controller settings are found, Stuxnet will throttle the frequency settings sabotaging the centrifuge system by slowing down and then speeding up the motors to different rates at different times.
- It includes the capabilities to remove itself from incompatible systems, lay dormant, reinfect cleaned systems, and communicate peer to peer in order to self-update within infected networks.
- It includes a variety of stop execution dates to disable the malware from propagation and operation at predetermined future times.

What we do not know at this point is what the full extent of damage could be from the malicious code that is inserted within the PLC. Subtle changes in **set points** over time could go unnoticed that could cause failures down the line, use the PLC logic to extrude additional details of the control system (such as command lists), or just about anything. Another approach might be to perform man-in-the-middle attacks intercepting invalid process values received from the PLCs and forward to the WinCC HMI bogus values for display making the plant operator unaware of what is actually occurring in the plant. Because Stuxnet has exhibited the capability to hide itself and lie dormant, the end goal is still a mystery.

Lessons Learned

Because Stuxnet is such a sophisticated piece of malware, there is a lot that we can learn from dissecting it and analyzing its behavior. A detailed white paper coauthored by one of the authors of this book has been developed that specifically analyzes Stuxnet in terms of its impact on industrial control systems, and how they are designed and deployed in actual operational environments.[16] How did we detect Stuxnet? It succeeded largely because it was so widespread and infected approximately 100,000 hosts searching for a single target. Had it been deployed more tactically, it might have gone unnoticed—altering PLC logic and then removing itself from the Siemens

Table 7.3 Lessons Learned from Stuxnet

Previous Beliefs	Lessons Learned from Stuxnet
Control systems can be effectively isolated from other networks, eliminating risk of a cyber incident.	Control systems are still subject to human nature: a strong perimeter defense can be bypassed by a curious operator, a USB drive, and poor security awareness.
PLCs and RTUs that do not run modern operating systems lack the necessary attack surface to make them vulnerable.	PLCs can and have been targeted and infected by malware.
Highly specialized devices benefit from "security through obscurity." Because industrial control systems are not readily available, it is impossible to effectively engineer an attack against them	The motivation, intent, and resources are all available to successfully engineer a highly specialized attack against an industrial control system.
Firewalls and Intrusion Detection and Prevention system (IDS/IPS) are sufficient to protect a control system network from attack.	The use of multiple zero-day vulnerabilities to deploy a targeted attack indicates that "blacklist" point defenses, which compare traffic to definitions that indicate "bad" code are no longer sufficient, and "whitelist" defenses should be considered as a catchall defense against unknown exploits.

SIMATIC hosts that were used to inject those PLCs. How will we detect the next one? The truth is that we may not, and the reason is simple—our "barrier-based" methodologies do not work against cyber-attacks that are this well researched and funded. Furthermore, since Stuxnet's propagation mechanisms were all LAN-based, the target host must be assumed on direct or adjacent networks to the initial infection. In other words, the attack originated from inside the targeted organization. They are delivered via zero-days, which means we do not detect them until they have been deployed, and they infect areas of the control system that are difficult to monitor.

So what do we do? We learn from Stuxnet and change our perception and attitude toward industrial network security (see Table 7.3). We adopt a new "need to know" mentality of control system communication. If something is not explicitly defined, approved, and allowed to execute and/or communicate, it is denied. This requires understanding how control system communications work, establishing that "need to know" and "need to use" in the form of well-defined security zones with equally defined perimeters, establishing policies and baselines around those zones, and then implementing cyber security controls and countermeasures to enforce those policies and minimize the risk of a successful cyber-attack.

It can be seen in Table 7.3 that additional security measures need to be considered in order to address new "Stuxnet-class" threats that go beyond the requirements of compliance mandates and current best-practice recommendations. New measures include Layer 7 application session monitoring to discover zero-day threats and to detect covert communications over allowed "overt" channels. They also include more

clearly defined security policies to be used in the adoption of policy-based user, application, and **network whitelisting** to control behavior in and between zones (see Chapter 9, "Establishing Zones and Conduits").

TIP

The axiom "to stop a hacker, you need to think like a hacker" was often used before Stuxnet. This simply meant that in order to successfully defend against a cyber-attack you need to think in terms of someone trying to penetrate your network. This philosophy still has merit, the only difference being that now the "hacker" can be thought of as having a much greater knowledge of deployed ICSs, an understanding of the manufacturing processes, and how the ICS is used to control this environment, along with significantly more resources and motivation. The ISA 62443 family of industry standards provides the ability to address each of these aspects in terms of a **Security Level**. In the post-Stuxnet world, imagine building a digital bunker in the cyber war, rather than simply defending a network, and aim for the best possible defenses against the worst possible attack. In other words, "think like an insider."

SHAMOON/DistTrack

Shamoon, or W32.DistTrack (often shortened to "DistTrack"), possesses both information gathering and destructive capabilities. Shamoon will attempt to propagate to other systems once an initial infection occurs, exfiltrate data from the currently infected system, and then cover its tracks by overwriting files, including the system's master boot record (MBR). The system is then unusable and overwritten data are not recoverable once the MBR is destroyed. The result, Shamoon left a path of inoperable systems in its wake.[17]

Shamoon accomplished this through three primary components:[18]

- Dropper – a modular component responsible for initial infection and network propagation (often through network shares)
- Wiper – a malware component responsible for system file and MBR destruction
- Reporter – a component designed to communicate stolen data and infection information back to the attacker.

Much of the details around Shamoon are protected from disclosure; however, Shamoon reportedly infected business systems of Saudi Aramco (an oil and gas company in the Kingdom of Saudi Arabia) and caused the destruction of at least 30,000 systems. Luckily, this destruction did not spread to industrial network areas, and therefore did not directly impact oil production, refining, transportation, or safety operations.[19]

FLAME/FLAMER/SKYWIPER

Skywiper is an advanced persistent threat that spread actively, targeting Middle Eastern countries, with the majority of infections occurring in Iran. Like Stuxnet, Skywiper (Flame) redefined the complexity of malware in its time. Skywiper had been active for years prior to being discovered also like Stuxnet, mining sensitive

data and returning them to a sophisticated C2 infrastructure consisting of over 80 domain names, and using servers that moved between multiple locations, including Hong Kong, Turkey, Germany, Poland, Malaysia, Latvia, the United Kingdom, and Switzerland.[20]

Over a dozen modules are present within Skywiper, including[21]

- "Flame" – handles AutoRun infection routines (Skywiper is often referred to as Flame because of this package)
- "Gadget" – an update module that allows the malware to evolve, and to accept new modules and payloads
- "Weasel" and "Jimmy" – handle disk and file parsing
- "Telemetry" and "Gator" – handle C2 routines
- "Suicide" – self-termination
- "Frog" – exploit payload to steal passwords
- "Viper" – exploit payload that captures screenshots
- "Munch" – exploit payload that captures network traffic.

Skywiper seems to be focused on espionage rather than sabotage. No modules dedicated to manipulation or sabotage of industrial systems have been detected at the time of this writing. The modular nature of Skywiper would certainly allow the threat to include more damaging modules as needed, no doubt leveraging the "Gadget" update module to further evolve the malware into a directed cyber weapon.

ATTACK TRENDS

Several trends can be discovered in how APT and cyber-attacks are being performed through the analysis of known cyber incidents. These include, but are not limited to, a shift in the initial infection vectors, the quality of the malware being deployed, its behavior, and how it spreads through networks and organizations.

Although threats have been trending "up the stack" for some time with exploits moving away from network-layer and protocol-layer vulnerabilities and more toward application-specific exploits, even more recent trends show signs that these applications are shifting away from the exploitation of Microsoft platform products (i.e. operating system exploitation) toward the almost ubiquitously deployed client-side applications like web browsers (Internet Explorer, Firefox, Safari, Chrome), Adobe Acrobat Reader, and Adobe Flash Player.

Web-based applications are also used heavily both for infections and for C2. The use of social networks, such as Twitter, Facebook, Google groups, and other cloud services, is ideal because they are widely used, highly accessible, and difficult to monitor. Even more interesting is that many users access these services on mobile and portable devices that typically contain no additional security software. Many companies actually embrace social networking for marketing and sales purposes, often to the extent that these services are allowed open access through corporate firewalls. This is further compounded by privacy concerns relating to what corporate

IT is actually allowed to monitor within the social media sessions. Issues around privacy are outside the scope of this book, but it is worth noting that regulations vary widely from country to country, and that the expansion of corporate networks across borders could introduce latent security vulnerabilities that should be accounted for.

The malware itself, of course, is also evolving. There is growing evidence among incident responders and forensics teams of the existence of deterministic malware and the emergence of mutating bots. Stuxnet is a good example again, since it contains robust logic and will operate differently depending upon its environment. Stuxnet spreads, attempts to inject PLC code, communicates via C2, lies dormant, or awakens depending upon changes to its environment.

EVOLVING VULNERABILITIES: THE ADOBE EXPLOITS

Adobe Portable Document Format (PDF) exploits are an example of the shifting attack paradigm from lower-level protocol and operating system exploits to the manipulation of application contents. This shift also allows the attack surface to expand significantly as there are far greater desktops to attack than servers. At a very high level, the exploits utilize the ability within PDFs to call and execute code to perform malicious actions. This occurs by either calling a malicious website or by injecting the code directly within the PDF file. It works like this:

- E-mail from a trusted source contains a compelling message, a properly targeted spear-phishing message. There is a PDF document attached to the e-mail.
- This PDF uses a feature, specified in the PDF format, known as a "Launch action." Security researcher Didier Stevens successfully demonstrated that Launch actions can be exploited and can be used to run an executable embedded within the PDF file itself.[22]
- The malicious PDF also contains an embedded file named Discount_at_Pizza_Barn_Today_Only.pdf, which has been compressed inside the PDF file. This attachment is actually an executable file, and if the PDF is opened and the attachment is allowed to run, it will execute.
- The PDF uses the JavaScript function exportDataObject to save a copy of the attachment to the user's local computer.
- When this PDF is opened in Adobe Reader (JavaScript must be enabled), the exportDataObject function causes a dialog box to be displayed asking the user to "Specify a file to extract to." The default file is the name of the attachment, Discount_at_Pizza_Barn_Today_Only.pdf. The exploit requires that the users' naïveté and/or their confusion regarding a message (which can be customized by the malware author[23]) they do not normally see to cause them to save the file.
- Once the exportDataOject function has completed, the Launch action is run. The Launch action is used to execute the Windows command interpreter (cmd.exe), which searches for the previously saved executable attachment Discount_at_Pizza_Barn_Today_Only.pdf and attempts to execute it.
- A dialogue box will warn users that the command will run only if the user clicks "Open."

This simple and effective hack is readily available in open-source toolkits like Kali Linux[24] and the Social Engineering Toolkit (SET),[25] and has been used to spread known malware, including ZeusBot.[26] Although this attack vector requires user interaction, PDF files are extremely common, and when combined with a quality spear-phishing attempt, this attack can be very effective. Quality is typically measured by how trust is established with the recipient and their likelihood of opening the attachment.

Another researcher chose to infect the benign PDF with another Launch hack that redirected a user to a website, but noted that it could have just as easily been an exploit pack and/or embedded Trojan binary.

There are numerous other Adobe Reader-based vulnerabilities that employ alternate methods to compromise a victim's local computer. Adobes, and other popular client application developers, continue to struggle in keeping up with vulnerability disclosures and the creation of exploit code due to the widespread use and dependence on these applications.

INDUSTRIAL APPLICATION LAYER ATTACKS

Adobe Reader exploits are highly relevant because many computing products—including ICS products—distribute manuals and other reference materials using PDF files and preinstall these on the ICS hosts. What is often the case as well is that the ICS software developers preinstall the Adobe Reader application, which oftentimes remains unpatched through traditional methods because it is not included with other vendor software update and hotfix notices. There are more directly relevant attacks that can occur at the application layer—industrial application attacks.

"Industrial applications" are the applications and protocols that communicate to, from, and between supervisory, control, and process system components. These applications serve specific purposes within the ICS, and by their nature are "vulnerable" because they are designed around control: either *direct* control of processes or devices (e.g. a PLC, RTU or IED), or *indirect* control, via supervisory systems like a DCS or SCADA that are used by human operators to supervise and influence processes or devices.

Unlike typical application layer threats, such as in the case of Adobe Reader, industrial application layer threats do not always require that a specific vulnerability be exploited. This is because these applications are designed for the purpose of influencing industrial control environments. They do not need to be infected with malware in order to gain the control necessary to cause harm, since they can simply be used as they are designed but with malicious intent. By issuing legitimate commands, between authorized systems and in full compliance with protocol specifications, an ICS can be told to perform a function that is outside of the owner's intended purpose and parameters. This method can be thought of as the *exploitation of functionality* and when considered in the context of ICS security, represents a problem that is not typically addressed through traditional IT security controls.

Digital Bond published one example of an industrial application layer attack in 2012 under the project name "Basecamp." The research documented how the EtherNet/IP protocol could be manipulated to control a Rockwell Automation ControlLogix PLC. It should be noted that it was not a ControlLogix vulnerability that was exploited, but the underlying protocol, and as such this exploit is widely applicable due to the prevalence of the EtherNet/IP protocol in ICS supplied by various vendors. A number of attack methods were disclosed, all sharing the common exploitation of EtherNet/IP: [27]

- **Forcing a System Stop**. This attack effectively shuts off the CIP service and renders the device dead by sending a CIP command to the device. This puts the device into a "major recoverable fault" state.[28]
- **Crashing the CPU**. This attack crashes the CPU due to a malformed CIP request, which cannot be effectively handled by the CIP stack. The result is also a "major recoverable fault" state.[29]
- **Dumping device boot code**. This is a CIP function that allows an EtherNet/IP device's boot code to be remotely dumped.[30]
- **Reset Device**. This is a simple misuse of the CIP system reset function. The attack resets the target device.[31]
- **Crash Device**. This attack crashes the target device due to a vulnerability in the device's CIP stack.[32]
- **Flash Update**. CIP, like many industrial protocols, supports writing data to remove devices, including register and relay values, but also files. This attack misuses this capability to write new firmware to the target device.[33]

EtherNet/IP is not the only protocol that can be exploited in this way. In 2013, Adam Crain of Automatak and independent researcher Chris Sistrunk reported a vulnerability with certain implementations of the DNP3 protocol stack, which was found to impact DNP3 master and outstation (slave) devices from a large number of known vendors. The weakness was an input validation vulnerability received from a DNP outstation station that could put the master station into an infinite loop condition.[34] This was not a specific device vulnerability, but a larger vulnerability concerning the implementation of a protocol stack, and because many vendors utilized a common library, it impacted a large number of products from multiple vendors. Of particular concern is that this vulnerability can be exploited via TCP/IP (by someone who has gained logical network access) or serially (by someone who has gained physical access to a DNP3 outstation).

Both of these examples represent weaknesses in protocols that were designed decades ago and are now being faced with new security challenges that were unforeseen at the time of their development. Since these also involve community-led open-source or licensed protocols that are not managed by a single vendor, their deployment can be very wide spread making it difficult to deploy patches and hotfixes that can be implemented in a timely manner. While vulnerabilities of this type are cause for concern, they can typically be mitigated through proper network and system design, and through the implementation of appropriate cyber security controls (which,

hopefully, is why you are reading this book). To put this another way, it is going to be a lot easier and less costly to deploy appropriate security controls to mitigate the risk from these open protocols versus attempting to retrofit and/or replace the affected ICS equipment.

An easy way to look at this is though the ICS devices themselves may be "insecure by design," the overall ICS can be sufficiently secured from cyber threats using a "secure by redesign" approach, rather than a "secure by replacement" one. After all, a "secure" device today could likely have vulnerabilities disclosed in the future that makes it "insecure" at that time. This is why industrial security is always focused on the holistic "system-level" security rather than that of individual ICS components.

ANTISOCIAL NETWORKS: A NEW PLAYGROUND FOR MALWARE

While social networks do not seem to have a lot to do with industrial networks (there should never be open connectivity to the Internet from an industrial zone, and certainly not to social networking sites), it is surprisingly relevant. Social networking sites are increasingly popular, and they can represent a serious risk against industrial networks. How can something as benign as Facebook or Twitter be a threat to an industrial network? Social networking sites are designed to make it easy to find and communicate with people, and people are subject to social engineering exploitation just as networks are subject to protocol and application exploitation.

They are at the most basic level a source of gathering personal information and end user's trust that can be exploited either directly or indirectly. At a more sophisticated level, social networks can be used actively by malware as a C2 channel. Fake accounts posing as "trusted" coworkers or business colleagues can lead to even more information sharing, or provide a means to trick the user into clicking on a link that will take them to a malicious website that will infect the user's computer with malware. That malware could mine additional information, or it could be walked into a "secure" facility to impact an industrial network directly. Even if a company has strict policies on the use of laptops accessing such websites, are these same companies as strict with the laptops used by their vendors and service subcontractors when connected to these same industrial networks? These same vendor/subcontractor computers are commonly connected directly to secure industrial networks. This is why it is equally important to consider the "insider" threats, and not focus entirely on external "outsider" originated attacks.

No direct evidence exists that links the rise in web-based malware and social networking adoption; however, the correlation is strong enough that any good security plan should accommodate social networking, especially in industrial networks. According to Cisco, "Companies in the Pharmaceutical and Chemical vertical were the most at risk for web-based malware encounters, experiencing a heightened risk rating of 543% in 2Q10, up from 400% in 1Q10. Other higher-risk verticals in 2Q10 included Energy, Oil, and Gas (446%), Education (157%), Government (148%), and Transportation and Shipping (146%)."[35]

Apart from being a direct infection vector, social networking sites can be used by more sophisticated attackers to formulate targeted spear-phishing campaigns, such as the "pizza delivery" exercise. Users may post personal information about where they work, what their shift is, who their boss is, and other details that can be used to engineer a social exploitation through no direct fault of the social network operators (most have adequate privacy controls in place). Spear phishing is already a proven tactic, yet it is easier and even more effective when combined with the additional trust associated with social networking communities.

TIP

Security awareness training is an important part of building a strong security plan, but it can also be used to assess current defenses. Conduct this simple experiment to both increase awareness of spear phishing and gauge the effectiveness of existing network security and monitoring capabilities:

1. Create a website using a free hosting service that displays a security awareness banner.
2. For this exercise, create a Google Mail account using the name (modified if necessary) of a group manager, HR director, or the CEO of your company (again, disclosing this activity to that individual in advance and obtaining necessary permissions). Assume the role of an attacker, with no inside knowledge of the company; look for executives who are quoted in press releases, or listed on other public documents. Alternately, use the Social Engineering Toolkit (SET), a tool designed to "perform advanced attacks against the human element," to launch a more thorough social engineering penetration test.
3. Again, play the part of the attacker and use either SET or outside means, such as Jigsaw.com or other business intelligence websites, to build a list of e-mail addresses within the company.
4. Send an e-mail to the group from the fake "executive" account, informing recipients to please read the attached article in preparation for an upcoming meeting.
5. Perform the same experiment on a different group, using an e-mail address originating from a peer (again, obtain necessary permissions). This time, attempt to locate a pizza restaurant local to your corporate offices, using Google map searches or similar means, and send an e-mail with a link to an online coupon for buy-one-get-one-free pizza.

Track your results to see how many people clicked through to the offered URL. Did anyone validate the "from" in the e-mail, reply to it, or question it in any way? Did anyone outside of the target group click through, indicating a forwarded e-mail?

Finally, with the security monitoring tools that are currently in place, is it possible to effectively track the activity? Is it possible to determine who clicked through (without looking at web logs)? Is it possible to detect abnormal patterns or behaviors that could be used to generate signatures, and detect similar phishing in the future?

The best defense against a social network attack continues to be security and situational awareness. Security Awareness helps prevent a socially engineered attack from succeeding by establishing best-practice behaviors among personnel. Situational Awareness helps to detect if and when a successful breach has occurred, where it originated, and where it may have spread to—in order to minimize the damage or impact from the attack and mitigate or remediate any gaps uncovered in security awareness and training.

Social networks can be used as a C2 channel between deployed malware and a remote server. One case of Twitter being used to deliver commands to a bot is the

CAUTION

Always inform appropriate personnel of any security awareness exercise to avoid unintended consequences and/or legal liability, and NEVER perform experiments of this kind using real malware. Even if performed as an exercise, the collection of actual personal or corporate information could violate your employment policy or even state, local, or federal privacy laws.

@upd4t3 channel, first detected in 2009, that uses standard 140-character tweets to link to base64-encoded URLs that deliver infosteeler bots.[36]

This use of social networking as a malicious vector is difficult to detect, as it is not feasible to scour these sites individually for such activity and there is no known way to detect what the C2 commands may look like or where they might be found. Application session analysis on social networking traffic could detect the base64 encoding once a session was initiated in the case of @upd4t3. The easiest way to block this type of activity, of course, is to block access to social networking sites completely from inside industrial networks. The wide adoption of these sites within the enterprise (for legitimate sales, marketing, and even business intelligence purposes) however makes it highly likely that any threat originating from or directly exploiting social networks can and will compromise the business enterprise. Special security considerations must be employed for this reason when evaluating the risk an organization faces from social networking.

Cannibalistic Mutant Underground Malware

More serious than the 1984 New World Pictures film about cannibalistic humanoid underground dwellers, the newest breed of malware is a real threat. It is malware with a mind using conditional logic to direct activity based on its surroundings until it finds itself in the perfect conditions in which it will best accomplish its goal (spread, stay hidden, deploy a weapon, etc.). The goal of Stuxnet was to find a particular ICS by spreading widely through local networks and "sneaker" networks. It then only took secondary infection measures when the target environment (Siemens SIMATIC WinCC/PCS7) was found. It then checked for particular PLC models and versions (Siemens models S7-315-2 and S7-417). Once these models were discovered, it looked for a specific make and model of VFDs (Fararo Paya model KFC750V3 and Vacon NX) before it injected process code into the PLC. If unsuitable targets were infected, it would lay dormant waiting for other hosts to infect.

Malware mutations are also already in use. Stuxnet at a basic level will update itself in the wild (even without a C2 connection), through peer-to-peer checks with other hosts also infected, and if a newer version of Stuxnet bumps into an older version, it updates the older version allowing the infection pool to evolve and upgrade in the wild.[37]

Further mutation behavior involves self-destruction of certain code blocks with self-updates of others, effectively morphing the malware and making it more targeted as well as more difficult to detect. Mutation logic may include checking for the presence of other well-known malware and adjusting its own profile to

utilize similar ports and services knowing that this new profile will go undetected. In other words, malware is getting smarter and at the same time, harder to detect.

DEALING WITH AN INFECTION

Ironically, upon detecting an infection, you may not want to immediately clean the system of infected malware. This is because there may be subsequent levels of infection that exist, yet are dormant and may be activated as a result. There could also be valuable information, such as the infection path used and other compromised hosts as in the case of Stuxnet. A thorough investigation should instead be performed, with the same sophistication as the malware itself.

The first step should be to logically isolate the infected host so that it can no longer cause any harm. Harm to not only other logical assets that may be on the shared network, but also the physical assets that the ICS host may be controlling. Allow the malware to communicate over established C2 channels, but isolate the host from the rest of the network, and remove all access between that host and any sensitive or protected information. A well-established network segmentation philosophy based on common security criteria needs to be deployed in order to effectively isolate infected hosts. This topic is covered further in Chapter 5, "Industrial Network Design and Architecture" and Chapter 9, "Establishing Zones and Conduits." Collect as much forensic detail as possible in the form of system logs, captured network traffic, supplementing where possible with memory analysis data. Important information can be gathered that may result in the successful removal of the infection by effectively sandboxing the infected system.

When you suspect that you are dealing with an infection, approach the situation with diligence and perform a thorough investigation:

- Remember to consider the safe and reliable operation of the manufacturing process as the primary objective. Extra care must be given to ICS components in their operating mode for this reason, and is why it is important to have a documented and rehearsed incident response plan in place.
- Always monitor everything, collecting baseline data, configurations, and firmware for comparison.
- Analyze available logs to help identify scope, infected hosts, propagation vectors, and so on. Logs should be retrieved from as many components on the network as possible, including those that have not been compromised.
- Sandbox and investigate infected systems.
- Be careful to not unnecessarily power-down infected hosts, and valuable information may be resident in volatile memory.
- Analyze memory to find memory-resident rootkits and other threats that may be residing in user memory.
- Clone disk images when possible to preserve as much of the original state as possible for off-line analysis.

- Reverse engineer-detected malware to determine full scope and to identify additional attack vectors and possible propagation.
- Retain all information for disclosure to authorities.

NOTE

Information collected from an infected and sandboxed host may prove valuable to legal authorities, and depending upon the nature of your industrial network, you may be required to report this information to a governing body.

A "bare metal reload" may be necessary where a device is completely erased and reduced to a bare, inoperable state depending on the severity of the infection. The host's hardware must then be reimaged completely. Clean versions of operating systems, applications, and asset firmware should be kept in a safe, clean environment for this reason. This can be accomplished using secure virtual backup environments, or via secure storage on trusted removable media that can then be stored in a locked cabinet, preferably in a separate physical location from the asset archived. It is important to ensure that the images used for system restoration are free and clean of any malware or malicious code that may have triggered the initial incident when using a backup and recovery system.

Free tools, such as Mandiant's Memoryze, shown in Figure 7.3, can help you to perform a deep forensic analysis on infected systems. This can help to determine how deeply infected a system might be by detecting memory-resident rootkits. Memoryze and other forensics tools are available at http://www.mandiant.com. The National Institute of Standards and Technology (NIST) has developed a valuable site containing a forensic tool catalog covering a wide range of common forensic tasks.[38]

TIP

The ability to perform forensics on a compromised system can be an advanced task. To help in this, the National Institute of Standards and Technology has established the Computer Forensics Tool Testing (CFTT) project and offers a "Computer Forensics Tool Catalog." Information can be found at: http://www.cftt.nist.gov.

TIP

If you think you have an infection, you should know that there are security firms that are experienced in investigating and cleaning advanced malware infections. Many such firms further specialize in industrial control networks. Before allowing anyone access to your ICS assets, it is encouraged to request and validate actual system experience—preferably on an ICS similar to yours. These firms can help you deal with infection as well as provide an expert interface between your organization and any governing authorities that may be involved.

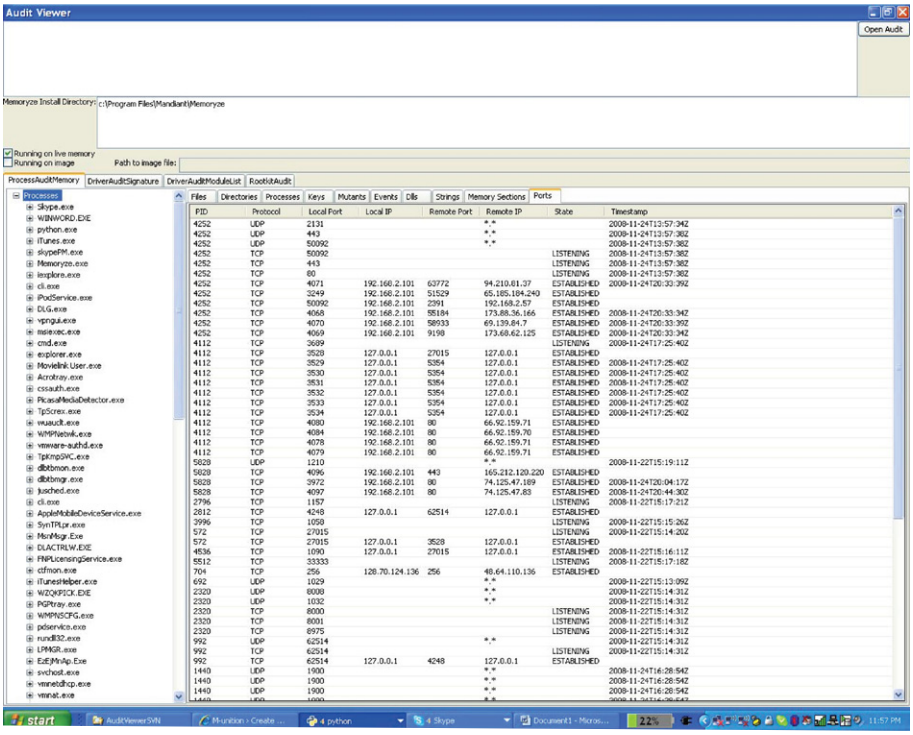

FIGURE 7.3 Mandiant's Memoryze: A memory forensic package.

SUMMARY

Cyber threats are increasing at an alarming rate, making the technologies that everyone now takes for granted the easy criminal path into theft, espionage, and sabotage. Industrial control systems account for less than 1% of the total vulnerabilities listed by the OSVDB, yet the trends associated with ICS cyber-attacks should be alarming. The rate of cyber incidents directly impacting industrial systems has been steadily increasing over the past 30 years according to the Repository of Industrial Security Incidents (RISI).[39] RISI's analysis also reveals that, although malware infections still account for a large number of cyber events (28% in 2013), it has been steadily decreasing over the past five years indicating that ICS users are becoming more aware of the methods to provide malware from affecting ICS architectures. These data also confirm that the vectors involved in ICS cyber events are shifting to more sophisticated mechanisms that are able to avert detection by traditional defenses, pivot through segmented networks, and exploit weaknesses in the underlying design of the ICS architecture.

Anyone who believes that they can prevent 100% of the possible cyber events within a particular system is misinformed and likely to be disappointed. A well-rounded cyber security program is based on a thorough understanding of the

threats that face industrial architectures, and blends security defenses that not only focus on event prevention, but also postbreach detection and forensic capabilities to contain an event and minimize as best as possible the negative consequences to the manufacturing or industrial process that the ICS is designed to control.

ENDNOTES

1. K. Stouffer, J. Falco, K. Scarfone, National Institute of Standards and Technology, Special Publication 800-82 (Final Public Draft), Guide to Industrial Control Systems (ICS) Security, Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology Gaithersburg, MD and Intelligent Systems Division, Manufacturing Engineering Laboratory, National Institute of Standards and Technology Gaithersburg, MD, September 2008.
2. M.J. McDonald, G.N. Conrad, T.C. Service, R.H. Cassidy, SANDIA Report SAND2008-5954, Cyber Effects Analysis Using VCSE Promoting Control System Reliability, Sandia National Laboratories Albuquerque, New Mexico and Livermore, California, September 2008.
3. A. Giani, S. Sastry, K.H. Johansson, H. Sandberg, The VIKING Project: An Initiative on Resilient Control of Power Networks, Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, and School of Electrical Engineering, Royal Institute of Technology (KTH), Berkeley, CA, 2009.
4. Dillon Beresford. Exploiting Siemens SIMATIC S7 PLCs. Prepared for Black Hat USA+2011. Las Vegas, NV. 2011.
5. Ralph Langner. Forensics on a complex cyber attack – lessons learned from Stuxnet. Presentation at the 2011 Applied Control Solutions (ACS) Conference. September 20, 2011. Washington, DC.
6. Dillon Beresford. Exploiting Siemens SIMATIC S7 PLCs. Prepared for Black Hat USA+2011. Las Vegas, NV. 2011.
7. Dillon Beresford. Exploiting Siemens SIMATIC S7 PLCs. Prepared for Black Hat USA+2011. Las Vegas, NV. 2011.
8. SearchSecurity. Definition: Blended Threat. Document from the Internet. Cited Sep 4, 2012. Available from: http://searchsecurity.techtarget.com/definition/blended-threat
9. G. McDonald, L.O. Murchu, S. Doherty, E. Chien, Symantec. Stuxnet 0.5: The Missing Link, Version 1.0, February 26, 2013.
10. Ibid.
11. Industrial Control Systems Cyber Emergency Response Team (ICS-CERT), ICSA-10-238-01—STUXNET MALWARE MITIGATION, Department of Homeland Security, US-CERT, Washington, DC, August 26, 2010.
12. E. Chien, Symantec. Stuxnet: a breakthrough. <http://www.symantec.com/connect/blogs/stuxnet-breakthrough>, November 2010 (cited: November 16, 2010).
13. Open-Source Vulnerability Database (OSVDB). ID 66441: Siemens SIMATIC WinCC SQL Database Default Password. <http://osvdb.org/show/osvdb/66441> (cited: December 20, 2013)
14. WinCC Database Problem. <https://www.automation.siemens.com/forum/guests/PostShow.aspx?PostID=16127> (cited: December 20, 2013)
15. N. Falliere, L.O Murchu, E. Chien, Symantec. W32.Stuxnet Dossier, Version 1.1, October 2010.

16. E. Byres, A. Ginter, J. Langill. "How Stuxnet Spreads - A Study of Infection Paths in Best Practice Systems," Version 1.0, February 22, 2011.
17. ICS-CERT. Joint Security Awareness Report (JSAR-12-241-01B) Shamoon/DistTrack Malware - Update B. Document from the Internet. April 30, 2013. Cited December 22, 2013. Available at: https://ics-cert.us-cert.gov/jsar/JSAR-12-241-01B-0
18. Ibid.
19. Kelly Jackson Higgins. 30,000 Machines Infected In Targeted Attack On Saudi Aramco. Dark Reading. August 2012. Document from the Internet. Cited December 22, 2013. Available at: http://www.darkreading.com/attacks-breaches/30000-machines-infected-in-targeted-atta/240006313
20. Kaspersky Labs. Virus News: Kaspersky Lab Experts Provide In-Depth Analysis of Flame's C&C Infrastructure. Document from the Internet. June 4, 2012. Cited Sep 18, 2012. Available from: http://www.kaspersky.com/about/news/virus/2012/Kaspersky_Lab_Experts_Provide_In_Depth_Analysis_of_Flames_Infrastructure
21. Kaspersky Labs. Virus News: Kaspersky Lab Experts Provide In-Depth Analysis of Flame's C&C Infrastructure. Document from the Internet. June 4, 2012. Cited Sep 18, 2012. Available from: http://www.kaspersky.com/about/news/virus/2012/Kaspersky_Lab_Experts_Provide_In_Depth_Analysis_of_Flames_Infrastructure
22. D. Stevens, Escape from PDF. <http://blog.didierstevens.com/2010/03/29/escape-from-pdf>, March 2010 (cited: November 4, 2010).
23. J. Conway, Sudosecure.net. Worm-Able PDF Clarification. <http://www.sudosecure.net/archives/644>, April 4, 2010 (cited: November 4, 2010).
24. Kali Linux. <http://kali.org>.
25. Social Engineering Framework, Computer based social engineering tools: Social Engineer Toolkit (SET). <http://www.social-engineer.org>.
26. 86 Security Labs, PDF "Launch" Feature Used to Install Zeus. <http://www.m86security.com/labs/traceitem.asp?article=1301>, April 14, 2010 (cited: November 4, 2010).
27. Ruben Santamarta. Attacking ControlLogix. Digital Bond Project Base Camp. 2012.
28. Ibid.
29. Ibid.
30. Ibid.
31. Ibid.
32. Ibid.
33. Ibid.
34. Advisory (ICSA-13-291-01). DNP3 Implementation Vulnerability. ICS-CERT. Original release date: November 21, 2013.
35. Cisco Systems, 2Q10 Global Threat Report, 2010.
36. J. Nazario, Arbor networks. Twitter-based Botnet Command Channel. <http://asert.arbornetworks.com/2009/08/twitter-based-botnet-command-channel>, August 13, 2009 (cited: November 4, 2010).
37. J. Pollet, Red Tiger, Understanding the advanced persistent threat, in: Proc. 2010 SANS European SCADA and Process Control Security Summit, Stockholm, Sweden, October 2010.
38. National Institute of Standards and Technologies (NIST) - Computer Forensic Tools Catalog, < http://www.cftt.nist.gov/tool_catalog/>, <sited: February 20, 2014>.
39. Report "2013 Report on Cyber Security Incidents and Threats Affecting Industrial Control Systems," Repository of Industrial Security Incidents (RISI), Published June 15, 2013.