MSc in Computer Networking and Security
Department of Computing
Cork Institute of Technology
Ireland

Risk Mitigation in ICS/SCADA Systems.

How it should be applied within Irish Medium Size Enterprises

Research project of XXXXX XXXXXX
Supervised by Jonathan Sherwin

Date: 11th August 2011

Abstract

The recent events in Iran that lead to the suspension of that countrys Nuclear Power generation program were triggered by a cyber-attack. It is believed that this attack was carried out by one or more nation states. This cyber-attack is more commonly known as the Stuxnet Virus which attaches SCADA systems. Attacks such as this have the potential to cause catastrophic failure of such systems as was outlined in the movie "Die Hard 4.0 (Live Free or Die Hard)".

This research project outlines the use of SCADA systems in the control and acquisition of data from Industrial Control Systems (ICS). It outlines the vulnerabilities faced by these systems and focuses on the threats of cyber-attack that these systems are now vulnerable to. It examines Standards and Procedures that can be applied to make these systems secure and presents a set of recommendations that Medium Size Enterprises (MSE) in Ireland can use to protect ICS/SCADA systems and themselves.

Acknowledgments

I would like to thank my fiancée and soon to be wife, XXXXX XXXXXXXX. Without her patience and support it would not have been possible to complete this project or study for this MSc in Networking and Security.

I would like to thank XXXXXX XXXXXX and XXXXX XXXX from the Department of Electrical Engineering of Cork Institute of Technology. The knowledge that XXXX and XXXXXXX have in the area of Industrial Control Systems (ICS) and Supervisory Control and Data Acquisition (SCADA) was a great help initially to grasp and understand the technologies involved.

I would also like to thank my project supervisor Jonathan Sherwin for all his guidance and direction during this project. I especially appreciate the feedback given on my project progress while he was on holidays.

Finally I want to thank Vincent Ryan, John Creagh and members of the Masters course for their valuable input and guidance.

Table of Contents

Glossary of terms

ACL	Access Control Lists
ADU	Application Data Unit
CIP	Critical Infrastructure Protection
CPU	Central Processing Unit
DHS	Department of Homeland Security
DMZ	De-Militarised Zone
DNP3	Distributed Network Protocol 3
DoS	Denial of Service
GUI	Graphical User Interface
HMI	Human Machine Interface
HVAC	Heating Ventilating and Air Conditioning
ICS	Industrial Control System
ICS-CERT	Industrial Control Systems Cyber Emergency Response Team
IP	Internet Protocol
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
ISA	International Society of Automation
IT	Information Technology
MBAP	MODBUS Application Protocol header
MITM	Man-in-the-Middle
MTU	Master Terminal Unit
NIST	National Institute of Standards and Technology
OLE	Object Linking and Embedding
OPC	OLE Process Control
PC	Personal Computer
PDU	Protocol Data Unit
PLC	Programmable Logic Controller
RTU	Remote Terminal Unit
SCADA	Supervisory Control and Data Acquisition
SSL	Secure Sockets Layer
TLS	Transport Layer Security
US-CERT	US Cyber Emergency Response Team

Introduction

ICS stands for *Industrial Control Systems* which are pieces of computer equipment that are commonly used in industry to mechanical devices such as motors, pumps and actuators. SCADA stands for *Supervisory Control and Data Acquisition* which are systems used to monitor and control industrial plant from a master or central location. These systems are commonly used to acquire actual plant values from the ICS equipment and use this data to:

1. Display this information graphically to human operators, use this information to, to control the ICS equipment. This is known as "manual control".
2. Automatically control the ICS equipment based on the parameters and programs configured on the SCADA control software.

SCADA systems are commonly used to monitor and control equipment of industrial, infrastructure or facility-based processes. While ICS and SCADA systems are very closely linked, SCADA systems cannot exist without ICS systems, however ICS systems can exist without SCADA systems. In the increasingly automated age that we live in, these systems have become more-of-less inseparable. For this reason, I will treat these combined ICS and SCADA systems as one, which I will call an ICS/SCADA system.

The ICS/SCADA systems were initially designed to give greater control and acquisition of data in local centralised sites. They have evolved greatly since this, to a stage where they are now used to monitor and control infrastructure in distributed sites. This was done initially through the use of PSTN modems but is done with the use of IEEE 802.3 Ethernet Networks. These Ethernet Networks have also allowed the use of mobile and hand held devices on control these systems. Although this increased flexibility to interact with these systems has many advantages it presents many challenges.

In the last 10 years there have been ever more sophisticated forms of malware and targeted attacks on IT infrastructure. There has been a realisation of how critical these ICS/SCADA systems now are and how susceptible they are to malicious attack. As far back as 1998, U.S. President Bill Clinton published Presidential Decision Directive 63 (PDD63) setting up the "Critical Infrastructure Protection" (CIP) Program. This put pressure on the critical infrastructure organisations such as electricity, water and gas to protect against the threat of cyber-attack on their systems (including their ICS/SCADA

systems). In Europe there is an equivalent "European Programme for Critical Infrastructure Protection" (EPCIP).

In the past, there have been many widely reported incidents of hacking and disruption to SCADA, as summarised in an article [1]. The most publicised and high profile of these cyber-attacks has been the Stuxnet virus (see Appendix 2). This was specifically designed to infect the SCADA systems of the equipment manufacture Siemens. This virus infected many computers world-wide but lay dormant unless the host PC ran a specific SCADA application. It is believed that this virus targeted the SCADA systems of the Iranian nuclear enrichment facility and the associated nuclear power generating station. From an Irish context, these systems are commonly used in the manufacturing, chemical, transportation, water and water-treatment, power/gas distribution industries. In these industries, the potential consequences of an ICS/SCADA system incident, be it cyber based or otherwise, could lead to: [2]

- Reduction or loss of production at one site or multiple sites simultaneously
- Injury or death of employees
- Injury or death of persons in the community
- Damage to equipment
- Environmental damage
- Violation of regulatory requirements
- Product contamination
- Criminal or Civil legal liabilities
- Loss of proprietary or confidential information
- Damage to brand image or customer confidence

The aim of this project is to research the vulnerabilities that are associated with these ICS/SCADA systems, look at methodologies that can be used to make them more secure and see how these can be applied to Irish organisations.

CHAPTER 1
SCADA Architecture Overview

1.1 Introduction to SCADA

Supervisory Control and Data Acquisition (SCADA) systems are used to monitor and control Industrial Control Systems (ICS) for:

- Industrial manufacturing equipment in factories and other production processes such as oil and gas refining and distribution.
- Critical infrastructure equipment for electricity generation, gas/ power distribution, water supply and treatment facilities and transportation infrastructure controls.
- Buildings maintenance systems equipment for Heating Ventilating and Air Conditioning (HVAC).

At the heart of these ICS are micro-computer systems called Programmable Logic Controllers (PLCs). These PLCs are not based on the common understandings of computer systems. They do not utilise the x86 PC based architecture or the Windows platform. Instead they are like any other micro-computer. They have a Central Processing Unit (CPU) (usually the 80c51 microcontroller) memory in the form of EPROM, RAM and an input/output section for communication with peripherals. These PLCs monitor electrical inputs and make changes to their outputs based on programs that they run.

The programs running on PLCs are not based on any procedural languages like C. They utilise the IEC 61131-3 standard of programming which is a suite of five programming languages which are: Function Block, Ladder, Structured Text, Sequence Function Charts and Instruction List. The IEC 61131-3 programming standard was designed for minimal training requirements as it resembles electrical relay wiring diagrams. These PLCs are commonly programed by either SCADA system Engineers or automation Engineers. They are usually programed with a Windows based programming application running on a PC using a direct connection through a serial, USB or Ethernet interface to the device. Once programmed PLCs carry out the task they are programmed to with very little updating or modification required.

In their basic form, PLCs do not provide for any form of graphical user interface (GUI) and generally take inputs from human operators through hard-wired push button switch interfaces. The technology for these ICS systems was developed in the 1970's and has

proven itself to be a very rugged and effective way to automate and control for what were previously very manual and repetitive, tasks and processes. Later ROBOTS, which is a general term for an electro-mechanical machine which is guided by computer and electronic programs, were developed and introduced into the works place. ROBOTS were increasingly used to do the tasks that were previously done manually. ICS systems helped to reduce production costs by automating tasks, so that little or no input is required from human beings.

In the 1980's, as the introduction of ICS systems became common place, there was an increased demand for more control of these systems and for methods to acquire data from these ICS systems. Ultimately the goal of industry was to automate tasks further and to reduce production costs further. This lead to the development of SCADA technology, which sits on top of ICS systems, to provide the additional functionality sought. Initially like with computer networks, these systems used bus technology to achieve this. At the physical layer these systems used serial RS232/RS485 DCE/DTE connections and later used RS485/RS422 multi-drop methods. For installations where networks of up to 1km are required, with high noise immunity, a 4-20mA current loop was commonly used. Initially manufacturer specific proprietary transmission protocols were used before open-source protocols such as MODBUS, PROFIBUS and DNP were developed (this is covered in more detail in section 1.5).

These networks and communication protocols allowed SCADA systems to acquire data from remote PLCs and sensors and the capability to control these devices such as motor and actuators. These systems rely on a "Command and Control" centre that generally utilises the x86 architecture and Windows operating systems, to run SCADA Control application software as well as a database to store historical data. These "Command and Control" devices are commonly called SCADA Servers.

The first generation of these SCADA systems and networks were installed in local sites for local control. The introduction and use of PSTN modems later brought additional flexibility to these systems. Remote sites could now communicate back to a central control location over the PSTN telecommunications infrastructure which allowed for remote monitoring and control of these sites. PSTN infrastructure also allowed for remote dial-in to these control systems for maintenance and modification by Engineers.

In the late 1990s there was a move away from the dedicated bus technologies and technologies that utilise the Ethernet IEEE 802.3 Networks were introduced to provide communications between SCADA devices. This was mainly done by taking the existing SCADA protocols and encapsulating them into an Ethernet Packet that could then be transmitted across the Ethernet Networks. This proved to be very popular option to users of these SCADA systems as it allowed a very flexible and inexpensive method to install, manage and operate these ICS/SCADA systems. In the last 10 years this increased flexibility has led to SCADA systems being migrated onto corporate business networks and also ported to the internet. This was a vital new tool which allowed managers and decision makers access to critical real time business information. This increases access to information was achieved by merging the automation networks, which SCADA systems run on, with the corporate business network. Migration of automation networks also allow Engineers to monitor, maintain and modify the control processes from remote locations.

1.2 SCADA Components

The fundamental components of ICS/SCADA systems can be broken down into three categories.

1. Master Terminal Units (MTUs).
2. Communications Network.
3. Remote Terminal Units (RTUs).

These and other concepts are detailed in this section. Figure 1.1 details how these components interact with each other.

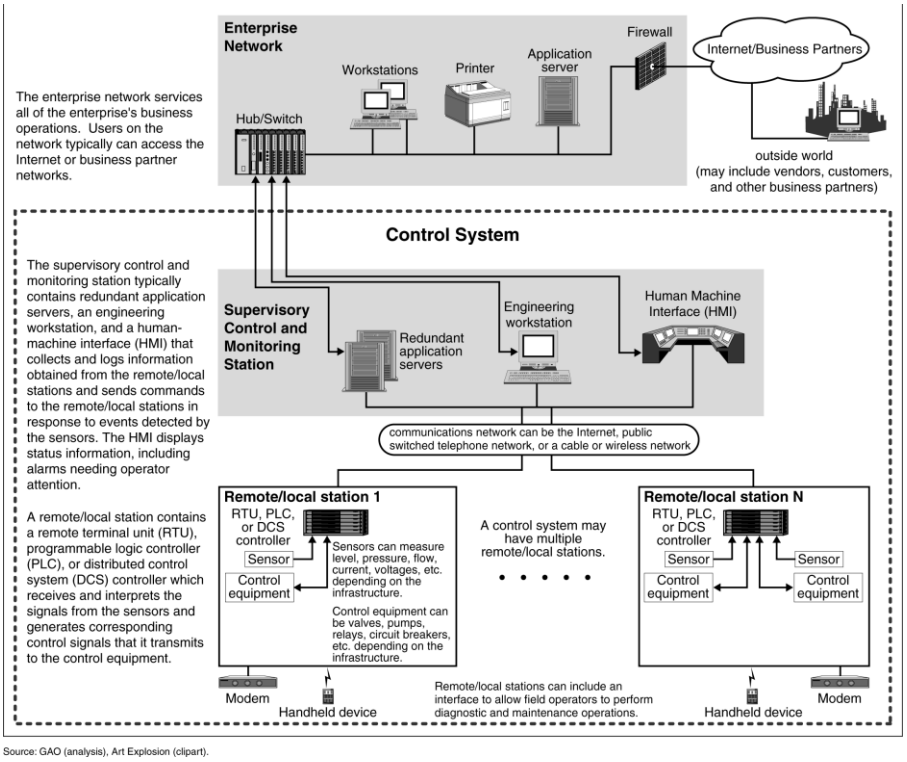


Figure 1.1 SCADA System Components [3]

1.2.1 Master Terminal Unit (MTU)

The MTU is one of the fundamental components of any ICS/SCADA system and can be thought of the processing centre of the SCADA system. It is also known as a SCADA server or a supervisory control and monitoring station. A MTU usually consists of a Windows based workstation running Human Machine Interface (HMI) application software and is usually located in a central site location with network connections to remote controller stations. The MTU issues commands to gather data from remote RTU's, PLCs and other devices. The data collected is either: displayed through the HMI to human operators who control the system, or used to conduct automatic control, through the use of a program on the SCADA server, which has predetermined settings and procedures. Once data is collected it is stored in a database so that various analyses on the systems can be conducted.

1.2.2 Remote Terminal Unit (RTU)

Remote Terminal Units (RTU's) are devices that communicate with the MTU's and also communicate with field instrument sensors and actuators. Sensors convert physical parameters to electrical signals usually in the format of a voltage or current reading. RTUs take the sensor information and packages it into a format that can be sent to the MTU via a communications network. The RTU only sends this instrumentation data to the MTU when it is polled by the MTU to do so.

Actuators are used to control motors, valves and other mechanical devices. The MTU sends control information to the RTU to carry out a specific operation. The RTU interrogates this information and sends corresponding information to an actuator to carry out the required operation. The majority of times this operation involves the opening or closing of an electrical contact to, switch on/off a motors, open/close a valves or some other similar operation.

The latest generation of ICS/SCADA equipment typically have the RTU functionality built into sensors and actuators. This allows for ease of deployment and installation. Figure 1.2.1 shows a graphical representation of the typical interaction between MTU, RTU, Sensors and Actuators.

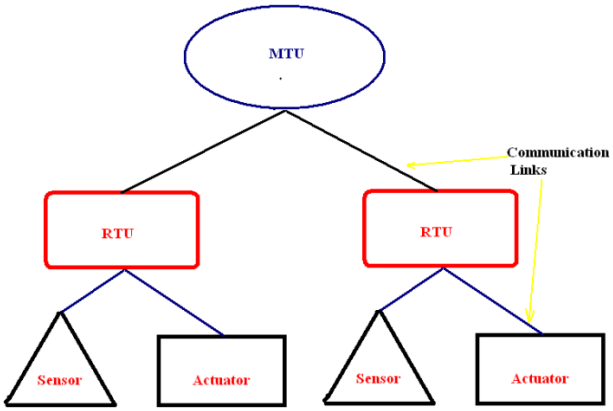


Figure 1.2.1 Block Diagram Representation

1.2.3 Programmable Logic Controllers (PLCs)

PLCs (as already detailed) have been an important part of ICS systems for many years. They form the backbone of the control system in many industrial production and other facilities. In recent years, the introduction of SCADA technology has led to new technology and additional features being add to these devices. PLC devices can now directly connect to communication networks to receive instruction commands from the MTU and to send update information as required by the MTU. This additional control is achieved with the use of new registers within the PLC. These are used by the SCADA system to read and write values to and from the PLC. These registers can be incorporated into the programs running on the PLC so that the operation of the PLC can be influenced by the values in these registers. These registers are single bit register which can be thought of as inputs to the programs being ran by the PLC.

1.2.4 Human Machine Interfaces (HMI)

Human Machine Interfaces (HMI) are run on the MTU and consist of graphical representation that mimics the system being controlled by SCADA system. This makes for ease of recognition with the systems being controlled. Each input and output to the system can be represented graphically on the HMI screen to give status and measurement values from the system in real-time. These inputs and outputs are assigned “tags” during system configuration. These tags are used to tie the values returned by a device to their graphical representation on the HMI.

1.3 ICS/SCADA Functions and Operation

ICS/SCADA systems provide an effective method to monitor and control processes through the acquisition and recording of data in a centralised location and the transmission of control information to remote devices. This activity is achieved through use of tags which in the case of HMIs provide for graphical representations of the status of the equipment being monitored and controlled. These tags can be used to create alarms windows and other GUI based information for manual control by human operators. Or also used for automatic control by making automatic adjustments to the ICS system based on the program ran by the ICS/SCADA system, this will keep the ICS system within thresholds and limits set for it.

These tags can be either "hard" or "soft": a hard tag represents a physical input or output on the PLC or RTU, while a soft tag are results from logic or mathematical operations which are applied to measured values. These tags consist of a tag name, description, value, time, date and other important information. This tag information is often fed into Historian Server, usually a Database Management System, to allow for trending and other analytical auditing.

1.4 SCADA Communication Protocols

SCADA systems started out with the use of manufacturer proprietary communication protocols. This was done to lock-in the manufacturers’ customer base to their products and technology. Typical these SCADA protocols include MODBUS, RP-570 and Conitel which are all vendor specific. Later, the development of open source communication protocols lead to different manufacturers choosing to use different non-compatible and competing protocols. These actions also led to inoperability between devices that were not from the same manufacturers. These newer open protocols were designed to be used by SCADA systems, were designed to work on the bus technologies

initially. Some of these are IEC 60870-5-101 or 104, IEC 61850, Profibus and DNP3. These communication protocols were standardised and recognised by all major SCADA vendors. In more recent times many of these protocols were migrated to operate over TCP/IP so that SCADA systems utilise this functionality. This was done by encapsulating the existing protocols data with headers to enable its transmission across the Ethernet Network. Currently the most commonly used of these protocols are DNP3 WAN/LAN, MODBUS/TCP and PROFINET.

1.4.1 DNP3 WAN/LAN

Distributed Network Protocol 3 (DNP3) was initially a proprietary protocol developed by Harris Controls, before it was made an open standard communications protocol. It was designed to operate over serial networks [4] [5]. It was later modified to use TCP/UDP transport methods and branded DNP3 WAN/LAN. DNP3 has become the industry standard SCADA communications protocol in the electric utility industry and also used widely in industries such as water and waste-water treatment, transportation and the oil and gas industries.

The link layer is responsible of making the physical link between communicating devices reliable. It does this by providing for error detection and duplicate frame detection, see figure 1.4.1. The link layer sends and receives data packets, which are called frames. Sometimes transmission of more than one frame is necessary to transport all of the information from one device to another.

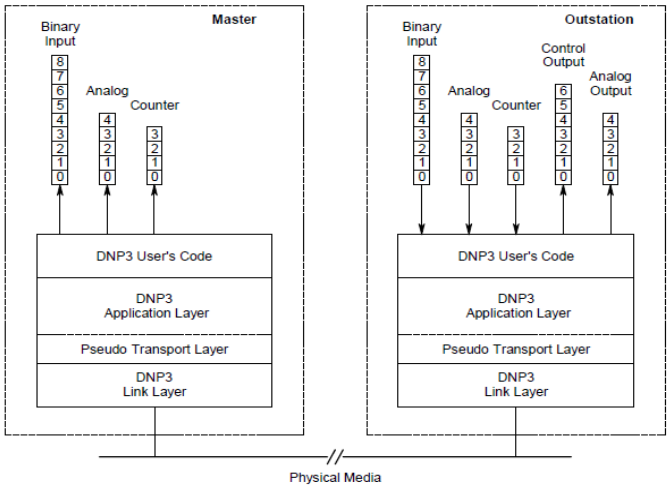


Figure 1.4.1 DNP3 Layers [6]

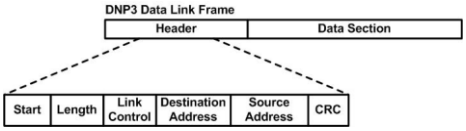


Figure 1.4.2 DNP3 Data Link Frame [4]

A DNP3 frame consists of a header and data section, see figure 1.4.2. The header specifies the frame size, contains data link control information and identifies the

DNP3 source and destination device addresses. The data section is commonly called the payload and contains data passed down from the layers above [6].

1.4.2 MODBUS-TCP

MODBUS-TCP has evolved from a serial protocol called MODBUS. It was developed by an equipment manufacturer called Modicon. Like DNP3 it later became open-source that lead to MODBUS becoming one of the de-facto standard communications protocols in the industry [7]. The MODBUS protocol defines a simple Protocol Data Unit (PDU) independent of the underlying communication layers. Devices communicate using a series of registers (16-bit integers) and coils (bits) which can be integrated or written to by the MTU. It uses a function code instruction that dictates to the device which kind of action to perform [8]. The mapping of MODBUS protocol on specific buses and networks leads to the introduction of an additional field to create an Application Data Unit (ADU). The client that initiates a MODBUS transaction builds the MODBUS ADU.

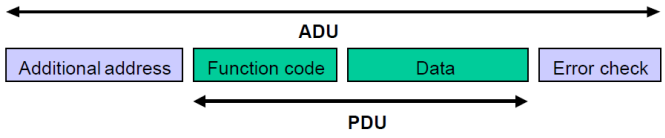


Figure 1.4.3 MODBUS packet [8]

A dedicated header is used on TCP/IP to identify the MODBUS ADU. This is called the MODBUS Application Protocol header (MBAP).

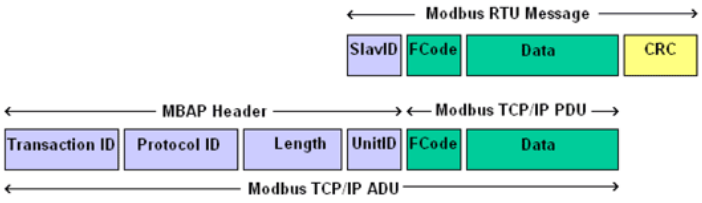


Figure 1.4.4 MODBUS TCP/IP Frame [8]

Initially the MODBUS ADU is encapsulated with TCP headers to enable its transport. All MODBUS/TCP ADU are sent via the registered TCP port 502. Next the Layer 3 IP addresses and Layer 2 MAC addresses are encapsulated as well as a layer 2 checksum (FCS). Please see figure 1.4.5. Figure 1.4.6 shows the signalling conducted during a typical request and transmission of SCADA telemetry and control information across the IEEE 802.3 Ethernet infrastructure.

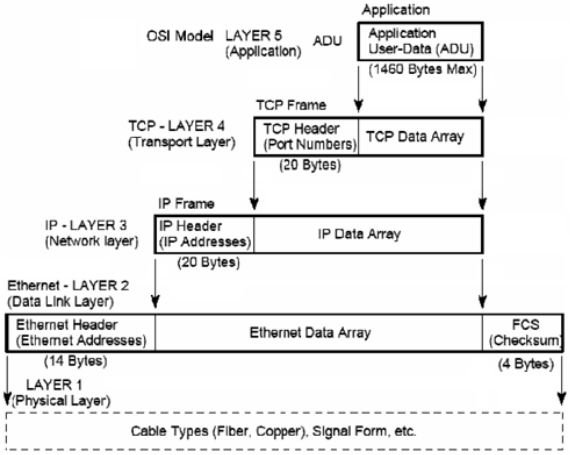


Figure 1.4.5 Construction of the MODBUS TCP/IP Data Packet [8]

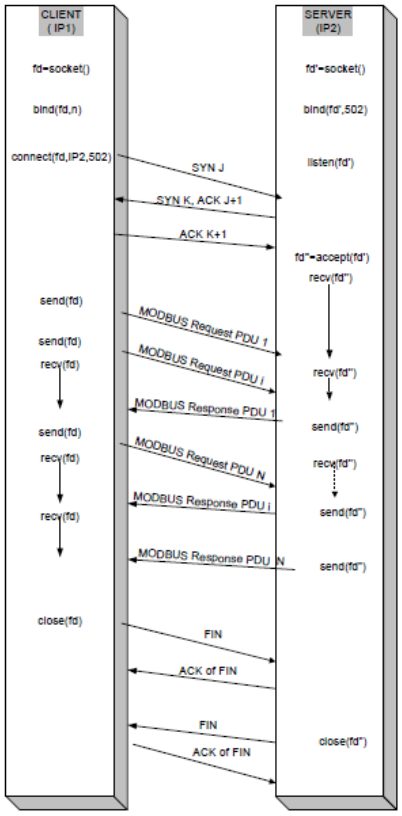


Figure 1.4.6 MODBUS TCP/IP Exchanges [8]

1.4.3 PROFI-NET

PROFINET has evolved from a serial protocol called PROFI-BUS which was also known as Process-Field-Bus [9]. PPROFINET is defined in IEC 61158 & 61784. It uses three different communication channels to exchange data with PLCs and other devices [10]. These are

1. Standard TCP/IP
2. Real Time (RT)
3. Isochronous Real Time (IRT)

RT and IRT are currently not commonly used as they are designed very high speed and applications. More information on these can be found in [10]. The standard TCP/IP channel uses configuration and acyclic read/write operations. Like with DNP3 and MODNUS, the automation data is encapsulated with the TCP, IP and Ethernet headers as shown in Figure 1.4.7 below.

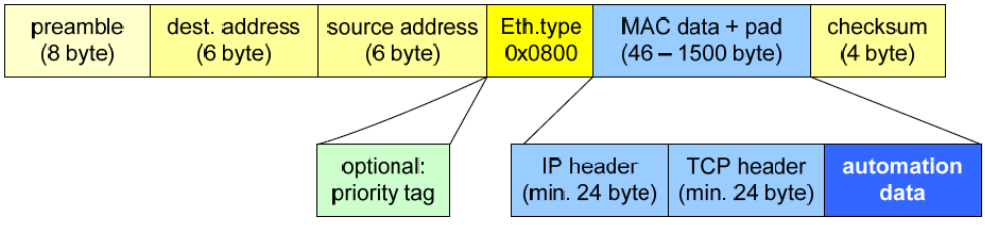


Figure 1.4.7 Standard PROFINET Frame[10]

Within the automation data addressing representation of PROFINET devices is the same as that used by PROFIBUS. This is done in a similar manner to how a PLC physically looks and uses terms such as Racks/Slots/Subslots.

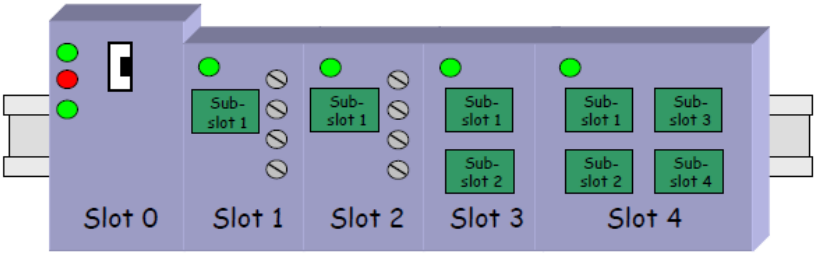


Figure 1.4.9 PROFINET Addressing [10]

Slots designate the physical slot of an I/O module in a modular I/O field device. The configured modules contain one or more sub-slots for data exchange. Within a slot, sub-slots form the actual inputs and outputs to the peripheral devices being controlled. Reading and writing to these sub-slots can be bitwise, byte-wise, or word-wise. The sub-slots can be written to (to control a peripheral device) or read from (to determine the status of the peripheral device). The data content of a sub-slot is always accompanied by status information, from which the validity of the data can be derived.

1.5 OLE Process Control (OPC)

As there are a large number of different SCADA equipment manufacturers using a large variety of technologies, architectures and protocols there is a lot of inter-operability in the automation industry. In the late 1990's a task force was set up among the automation SCADA equipment suppliers in order to develop an open standard which would provide a common basis for software applications and the process control hardware. This standard was called the OLE Process Control (OPC), which provides connectivity using an open standard. It does for the automation industry what printer drivers did for Windows [11] . Its use has led to inter-operability in automation by creating and maintaining an open specification that standardises the acquisition and control of process data from sensors, instruments, PLCs, RTUs, and HMI's from all manufacturers. OPC relies on Remote Procedure Call (RPC) and Distributed Component Object Model (DCOM) and it is now the world's most widely used industrial integration standard.

This standard defines an OPC server as the gateway between the automation equipment and the Client Application Software (SCADA MTU) used to monitor and control the

SCADA system. The OPC servers use Microsoft's Object Linking and Embedding (OLE) technology (also called Component Object Model (COM)) to communicate with clients. This allows for information exchange between software applications and process hardware to be defined. It also allows any OPC client access data from process control devices through an OPC server for that device, rather than having to access the data through a custom interface or driver.[12]

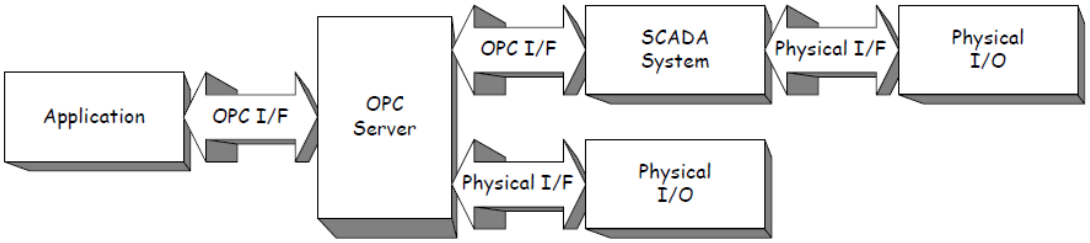


Figure 1.5.1 OPC Operation [12]

1.6 Evolution of SCADA systems

The first generations of SCADA systems used serial communication techniques and were placed on physically separated and dedicated control networks. This provides an "*air gap*" isolation between the control network and the un-trusted networks which were the internal corporate network and the external networks. It allows a level of security that I would describe as "Security by Obscurity". With the recent development of technology to migrate SCADA systems to Ethernet networks and the additional functionality which this brings, it is no longer feasible for these networks to run autonomously. Business Managers require access to the information which the extra functionality provides. This information needs to stream from a variety of different devices and locations, both internal and external to the corporate network. While this increased access to SCADA system data brings many benefits to organisations that use it, it also brings inherent dangers resulting from making SCADA granting network access to a wide range of users. Dangers include malicious software (Malware) and hackers. Finding a balance to meet the needs of accessibility with the need to safeguard their integrity, availability and confidentiality now becomes the challenge.

The US National Institute of Standards and Technology (NIST) (publication FIPS 199 [13] for Information Technology systems) defines the loss of confidentiality as the unauthorised disclosure of information, the loss of integrity as the unauthorised modification or destruction of information and the loss of availability as the disruption of access to the use of information or an information system. When applying these terms to ICS/SCADA systems

- Confidentiality – for market value and not reliability in protecting from unauthorised access.
- Integrity - most important due to impact and needs to assure valid data and control functions.
- Availability – usually addressed with redundancy to ensure system continuity of operations.

1.7 Findings

In summary, ICS/SCADA systems are very complex systems that use very different technology and concepts from typical computer systems. These systems often control industrial processes and critical infrastructure assets that we all rely on. They have evolved greatly, their current stage where by they utilise Ethernet Networks to provide organisations with access to real-time information such as business critical data analysis tools beyond the original scope of SCADA design.This increased functionality added to these systems however raised concerns of availability, integrity and confidentiality of these systems.

CHAPTER 2
SCADA Vulnerabilities and Attacks

This chapter outlines the security issues that are involved with ICS/SCADA systems. It explores the vulnerabilities that these systems can be exploited to that can affect availability, integrity and confidentiality. It examines cyber-attacks and categorises them as well as looking at the sources of these attacks. The risks to ICS/SCADA systems are also examined which is a combination of probability of a security related event and its perceived consequences of a successful attack. It also looks at possible attack scenarios to ICS/SCADA systems.

2.1 Threats to ICS/SCADA systems

The sources of attacks or threats to the ICS/SCADA systems are the same as those faced by all IT networks and infrastructure. The US Department of Justice (DOJ) and Government Accountability Office (GAO) has categorised these threats which are summarised in the Appendix 1. Along with as these threats, IT systems are also susceptible to threats from human error/accidents, unintentional consequences, equipment failure and natural disasters. Recently it has been discovered that ICS/SCADA systems are vulnerable to the same threats as traditional IT system - that is traditional personal and corporate computer networks.

Three well-known and documented attacks on SCADA systems that emphasise there risks are

- In Australia in 2000 an employee who had been dismissed released millions of litres of untreated sewage using a laptop to get access to the SCADA control system from the company car-park. He was arrested, convicted and jailed.
- In August 2003, the Nuclear Regulatory Commission confirmed that in January 2003, the Microsoft SQL Server worm known as “Slammer” had infected a private computer network at the Davis-Besse nuclear power plant in Ohio, USA disabling a safety monitoring system for 5 hours.
- In late 2010, the Stuxnet Virus was the most sophisticated attack to date and used a large variety of techniques to gain access to the SCADA control system. It is speculated that the foreign intelligence services of the USA and Israeli governments colluded to bring down the Uranium enrichment facilities of the Iranian nuclear industry, see Appendix 2.

2.2 Vulnerabilities of ICS/SCADA systems

Modern SCADA systems are vulnerable in a large number of areas as they are quite complex and encompass a large amount of different technologies and architectures. In a recent National Institute of Standards and Technology (NIST) [14] publication a detailed list of vulnerabilities that may be found in typical ICS/SCADA system was provided. I have grouped these into four distinct areas which will be covered individually in the following chapters as well as an examination on possible solutions to overcome these vulnerabilities

1. Policy and Procedure Vulnerabilities (CHAPTER 4)
2. Platform Vulnerabilities (CHAPTER 5)
3. Network Vulnerabilities (CHAPTER 6)
4. Communication Protocol Vulnerabilities (CHAPTER 7)

2.3 Increased risk to SCADA systems.

The level of risk to SCADA systems has increased due to a number of factors

1. The adoption of standardised technologies with known vulnerabilities.
2. The availability of technical information about SCADA systems.
3. The connectivity of SCADA systems to the corporate network that port to the internet.
4. Non implementation of any practices and security technologies.
5. The use of insecure network architectures and remote connections.
6. The lack of Real-Time Monitoring for security issues.

2.4 Misconceptions for ICS/SCADA Security

Three misconceptions about the security of ICS/SCADA systems are as follows: [15]

- The ICS/SCADA system resides on a physically separate, standalone network.
- Connections between ICS/SCADA systems and other corporate networks are protected by strong access controls.
- ICS/SCADA systems require specialised knowledge, making them difficult for network intruders to access and control.

2.5 Possible Incident Scenarios

Attackers with malicious intent might take one or more of the following actions to successfully attack control systems: [3]

- Many ICS/SCADA systems are time critical and need to ensure that data being transmitted in not overly delayed in its transmission. Attackers can trigger delays to systems that will affect availability.
- Disrupt the operation of control systems. This can be done by delaying by or blocking the flow of information through control networks, thereby denying availability of the networks to control system operators.
- Make unauthorised changes such as programme instructions in PLCs, RTUs, or ICS/SCADA controllers, alarm thresholds, or issue unauthorised commands to control equipment. Any such change which could potentially result in damage to equipment (if tolerances are exceeded), premature shutdown of processes (such as prematurely shutting down transmission lines), causing an environmental incident or disabling control equipment.
- Send false information to control system operators either to disguise unauthorised changes or to initiate inappropriate actions by system operators.
- Modify the control system software, producing unpredictable results; and interfere with the operation of safety systems.
- Malicious Software introduced into the system.
- Amend work instructions in order to cause damage to products, equipment or Personnel.

2.6 Current Security posture

In the last 10 years there has been an increased emphasis on protecting the enterprise side of IT networks and the assets connected to them. This has been highlighted as essential by the increasingly sophisticated cyber-attacks now being carried out. Today cyber-attacks is big business and it is a multi-million Euro under-ground industry, which employs a large number of programmers.

Methods of improving protection of the enterprise IT network include reducing the attack surface and layering of security defences. Often referred to as “defence in depth” this model is designed to manage risk with diverse layered defensive strategies. The layered defence give several benefits [16] i.e. if one layer of defence is compromised, another layer of defence, using a different security method, presents an additional

obstacle preventing further penetration. In addition each defensive layer can be optimised to deal with a specific range of threats thus dealing with attacks that come in different format.

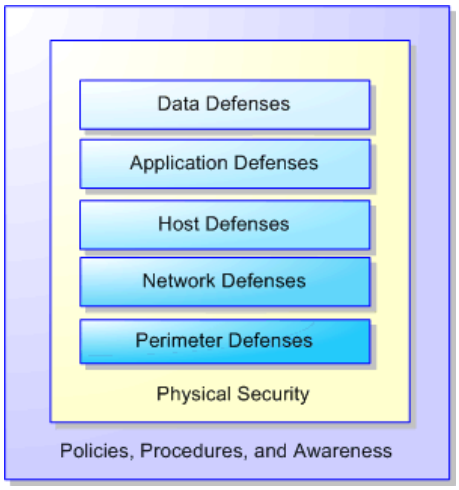


Table 2.6.1 Defence-in-Depth Security model [16]

There has been little focus placed on security of the "hidden" SCADA networks. These networks are generally managed by Automation Engineers and the Central IT team who have had little or no control or knowledge of the systems operation, configuration or maintenance. These distinct teams do not fully understand each others' IT requirements and don't collaborate to secure the ICS/SCADA control networks. While there has been a lot of academic research papers published in the area of security of ICS/SCADA systems none have yet reached the implementation phase. This is mainly due to the huge financial investment needed to implement same. It is only since the setection of the Stuxnet virus in the last year that there has been a fresh look at the security of ICS/SCADA systems by manufacturers and users with a view to implementing security controls on their existing systems.

2.7 Cyber-attacks

The term cyber-attack described many different types of malicious Software attacks which all IT infrastructure are susceptible to. The US DOJ and GAO have categorised the different levels of cyber-attack on IT infrastructures into 11 categories which are outlined in the table 2.1 below

Type of attack	Description
Denial of service	A method of attack from a single source that denies system access to legitimate users by overwhelming the target computer with messages and blocking legitimate traffic. It can prevent a system from being able to exchange data with other systems or use the Internet.
Distributed denial of service	A variant of the denial-of-service attack that uses a coordinated attack from a distributed system of computers rather than from a single source. It often makes use of worms to spread to multiple computers that can then attack the target.
Exploit tools	Publicly available and sophisticated tools that intruders of various skill levels can use to determine vulnerabilities and gain entry into targeted systems.
Logic bombs	A form of sabotage in which a programmer inserts code that causes the program to perform a destructive action when some triggering event occurs, such as terminating the programmer's employment.
Phishing	The creation and use of e-mails and Web sites—designed to look like those of well-known legitimate businesses, financial institutions, and government agencies—in order to deceive Internet users into disclosing their personal data, such as bank and financial account information and passwords. The phishers then take that information and use it for criminal purposes, such as identity theft and fraud.
Sniffer	Synonymous with packet sniffer. A program that intercepts routed data and examines each packet in search of specified information, such as passwords transmitted in clear text.
Trojan horse	A computer program that conceals harmful code. A Trojan horse usually masquerades as a useful program that a user would wish to execute.
Virus	A program that infects computer files, usually executable programs, by inserting a copy of itself into the file. These copies are usually executed when the infected file is loaded into memory, allowing the virus to infect other files. Unlike the computer worm, a virus requires human involvement (usually unwitting) to propagate.
War dialing	Simple programs that dial consecutive telephone numbers looking for modems.
War driving	A method of gaining entry into wireless computer networks using a laptop, antennas, and a wireless network adaptor that involves patrolling locations to gain unauthorized access.
Worm	An independent computer program that reproduces by copying itself from one system to another across a network. Unlike computer viruses, worms do not require human involvement to propagate.

Table 2.1 Cyber-attacks on IT infrastructures [3]

Cyber-attacks allow attackers to gather information on IT systems with the ultimate goal of gaining unauthorised access to them. Once access has been gained, further steps can be taken to disrupt these systems and effect availability, integrity and confidentiality. The Stuxnet Virus proves ICS/SCADA systems are also vulnerable to cyber-attack.

2.8 Findings

In summary, the threats to ICS/SCADA systems are complex but can be thought of as being similar to that of traditional IT infrastructure. As these systems are responsible for the control of critical infrastructure considerable damage can be done in a security breach. The vulnerabilities can be broken down into four areas: Policy and Procedure, Platform, Network and Communication Protocol. It is my belief that in Ireland country organisations are not adequately investing in securing these systems and the security for ICS/SCADA systems is typically 5 to 7 years behind typical IT systems. The "Defence in Depth" model may to be a useful tool to secure ICS/SCADA systems.

CHAPTER 3
Interviews Conducted

As part of this research project interviews were conducted with experienced ICS/SCADA users in the Cork area. These interviews were conducted to gain an understanding of the level of awareness of the vulnerabilities associated with ICS/SCADA systems and controls are being put in place to mitigate risk. These interviews were conducted by telephone which allowed me to adapt my range of questions depending on the knowledge of the individual in this area. I believe that this adaptive approach to gathering of this data was the most beneficial in this study.

3.1 Interview 1

Date: 11th March 2011.
Company: Pfizer Pharmaceutical Ireland.
Interviewee: Paul O'Sullivan.
Role: Lead Infrastructure Security Engineer.

Paul is the project leader on an Infrastructure Security program project which incorporates ICS/SCADA systems. Paul described how the security of infrastructure equipment within Pfizer and the communications between this equipment, is now of high importance within the organisation. As part of this project, Pfizer has introduced the recommendations set out by the International Society of Automation (ISA). He outlined that the ISA have two publications which Pfizer company use as models to secure their automation networks. These are ISA-95 and ISA-99 (which I will examine in Section 8.1) later in the research project.

Pfizer implements layers of security in the form of Intrusion Prevention and Detection Systems (IPDS) and Firewalls between Level 3 and Level 4 layers of the ISA-95 in order control the flow of information between these layers. These devices will provide security to restrict IP traffic between these layers. This will also provide the controls to stop malicious traffic from outside the organisation propagating to the production systems. I will explore further and examine the level of security that models like these provide.

Paul described how use a large variety of vendors for SCADA equipment which use several different communications protocols between devices. When I asked him what level of security they implement on the SCADA communications protocols such as MODBUS (which sit at level 1 and level 2 of the ISA-95 model) Paul stated that Pfizer don't look at specific security within these levels and that they concentrate on securing the Infrastructure between level 4 and level 3 as explained.

I also asked Paul about the current requirements for securing the Infrastructure under regulations set down by organisations such as the US Food and Drug Administration (FDA). Paul stated that there currently were not any regulations in regard to this, but he doesn't see this being the case in the medium term. Paul also said that a lot of smaller companies that have ICS/SCADA systems and have no concept of the security vulnerabilities associated with these systems and no understanding on how to make them more secure.

3.2 Interview 2

Date: 31st March 2011.
Company: Sole-trader.
Interviewee: Eoin Daly.
Role: Automation Engineer

Eoin is an Automation Engineer specialising in the design and maintenance of ICS/SCADA systems and works with many different Pharmaceutical companies in the Cork area. Eoin highlighted to me that within the industrial automation sector there is quite a lot of legacy automation equipment that on which companies rely. The industrial plant and the instrumentation have long life cycles with many pieces in use up to fifteen years. There is no provision for updating these devices in the same way companies currently upgrade the out of date administration or management PC. Most companies have an attitude of "if it's not broken don't fix it" and in order to reduce operating costs. A lot of these SCADA systems are reliant on out-dated unsupported and un-patched operating systems such as Windows NT4.

Eoin highlighted the importance of ICS/SCADA systems in industry with an example of incident where product batch production was put on hold until an investigation on the loss of SCADA communications between a PLC and a SCADA control server was carried out. It was discovered that these servers were running on the out-dated NT4 operating system which had been configured and programmed 10 years previous. The investigation had to identify what went wrong and if the product was affected by this loss of communications. This type of incident can occur up to 4/5 times per year with the potential cost to €20,000 between investigation and loss of production while the batch was put on hold. It was suggested to Eoin that if an attacker with malicious intent were to launch a attach using the ICS/SCADA communications protocol, such as MODBUS/TCP or PROFINET, to swamp a network with packets this could potentially result in the loss of communications between the PLC and SCADA server as in the

case he highlighted. Eoin said that if such an attack were possible it would cause considerable issues in the companies that he supports.

Eoin also raised the fact that these systems and the peripherals that they connect to are often operated in an insecure manner. No password control policy is in existence and often staff use the default passwords that ship with the products. There are usually generic logins with no record when a user logged onto the system. These systems are not protected by any centralised access system such as Active Directory to enforce password policies or remove user rights from these devices as they leave the company. Many of the older legacy systems have no connections to the business enterprise network where policies can be implemented through the integration with domain controllers that provide IP layer security. These systems sit in on Manufacturing Local Area Networks (MLAN), a type of remote VLAN.

Eoin also mentioned decision makers in organisations have requested access to Historian services that can be provided by ICS/SCADA systems to look at historical and real-time manufacturing data. This would involve porting these systems onto the corporate business network. If this is done without adequate security measures it leaves the ICS/SCADA systems very vulnerable.

3.3 Findings

These interviews have highlighted the lack of knowledge within many organisations in Ireland when it comes to the security and the threats posed to ICS/SCADA systems. One of the leading pharmaceutical companies has only recently been given the resources and prioritisation to access the security and make changes to their ICS/SCADA systems. This initiative is company lead rather than industry lead. ICS/SCADA security regulation maybe not far away in industries like the Pharmaceutical given recent attacks like Stuxnet. I believe that it will be some time before the unregulated automated industries will follow the steps taken by Pfizer's. Organisations that use ICS/SCADA systems in this country now need to examine the security aspects of their systems in light of recent attacks. There are many different security standards and best practice recommendations published that will guide organisations in the process of securing these ICS/SCADA systems. Some of these I will explore further in Chapter 8. I will then develop a set of recommendations that organisations in Ireland that rely on ICS/SCADA systems, can implement to protect the organisation and the ICS/SCADA systems on which they rely (see Chapter 10).

CHAPTER 4
ICS/SCADA Policy & Procedure Vulnerabilities

4.1 Security methods and standards

There are many different and proven information security methods and standards which the corporate environment can use to improve the security of Information Technology Data Systems. Standards such as ISO 27000 series of Information Security Management System (ISMS), the National Institute of Standards and Technology (NIST) special publication series 800 recommendations for computer security policies procedures and guidelines are just two options that can be mixed or matched and applied within organisations to prevent data loss. These methods, or combinations of them, will provide the guidelines of what to do and how to do it when it comes to securing Information Services. Knowing what to do and how to do it accounts for 60% of any security solution, the other 40% coming from a solid commitment to a set of attainable goals and complying with the standards and practices that are selected. For example, the ISO 27000 ISMS [17] will provide an organisation with good guidelines around IT security basis for IT security under the following headings

- Risk Assessment
- Security Policies
- Organisation Information Security
- Asset Management
- Human Resource Security
- Physical and Environmental Security
- Communications and Operations Management
- Access Control
- Information system acquisition development and maintenance
- Information Security incident Management
- Business continuity Management
- Compliance

4.2 Introduction to corporate policies and procedures

As outlined in section 2.6 the last 10 years has seen a large amount of focus on securing Information Systems of the enterprise corporate network. This has been achieved using methods such as the “defence in depth” model. As part of this layered approach there is an overall layer of security placed on Information Systems with the use of policies. Once these policies are in place implementation guides/procedures relating to these

systems can be developed to steer users in the proper use of these systems. Policies and procedures are an essential foundation for effective Information security programs. They are the least expensive control that can be put in place but the most difficult to implement. Policies primary function is to communicate to users their roles so that the controls are implemented consistently across an organisation.

In the field of Information Technology Security there are three different types of policy which organisations can use.

- Enterprise information security program policy (EISP)
- Issue specific information security policies (ISSP)
- System-specific information Security policies (SysSPs)

Every security policy needs management buy-in and support to enable its development, implementation and operation. Resources in the form of personal and possibly external training/consulting may be needed during this process.

When implemented successfully security policies provide a well-documented and well-communicated policy that insures no ambiguity amongst users. Staff will then be able to maintain a secure Information System while being aware of the implications of non-adherence to these procedures.

The increased focus on Information System security has led to many organisations developing a large selection of well thought out and effective security policies when it comes to the use of the corporate IT infrastructure. The SANS Institute provides a variety of sample policies which organisations can take free of charge and modify to meet their own specific needs [18]. Some examples of possible policies that a company might develop to protect their Information Systems are as follows

- Acquisition Assessment Policy
- Bluetooth Device Security Policy
- Email Security Policy
- Ethics Policy
- Information Sensitivity Policy
- Internal Lab Security Policy
- Personal Communication Devices and Voicemail Policy
- Risk Assessment Policy

- Technology Equipment Disposal Policy
- Web Application Security Assessment Policy
- Anti-virus update Policy
- Application and OS Security Patching Policy

4.3 ICS/SCADA Policy and Procedure Vulnerabilities

Upon examination of the policies used on corporate networks and infrastructure it becomes clear that they do not extend easily to the ICS/SCADA systems. My findings from the interview process conducted show in a lot of companies' while there are policies and procedures that focus on the corporate Information Systems only, with little or no focus on providing this type of control to the ICS/SCADA infrastructure. The lack of these policies and procedures introduces vulnerabilities to control and automation systems and engineers/users have no guidance on their secure design, implementation and operation. Like with all technologies it is not advisable to use ICS/SCADA systems without sufficient security controls in the form of policies in place. The vulnerabilities associated with this lack of inadequate policies to protect the ICS/SCADA equipment led to [14]

- No formal ICS/SCADA security training and awareness program.
- Inadequate security Architecture and Design.
- Lack of documented security implementation guides/procedures.
- Few or no security audits on the ICS/SCADA to determine the adequacy of controls within the system.
- Lack of sanctions to non-compliance to security fundamentals.
- Lack of ICS/SCADA specific of "disaster recovery" plans.
- Lack of ICS/SCADA specific Configuration Change Management.

4.4 Steps to overcome Policy & Procedure Vulnerabilities.

To control and mitigate the vulnerabilities associated with the lack of security policies for ICS/SCADA systems, organisations need to develop security policies, guides and procedures and implement same. There a number of different policies that should considered. Three such policies are outlined in this section below.

4.4.1 ICS/SCADA Security Patching Policy

Any ICS/SCADA systems are reliant on vulnerable Operating Systems (OS's) such as Linux and Windows. These OS's, (the firmware that devices run as well as the applications software) need regular updating to patch newly identified

vulnerabilities. In many cases these patching updates may require system restarts which can lead to loss of availability of the system and/or downtime. As ICS/SCADA systems have different requirements to the corporate IT Patching Policy companies need to develop a separate ICS/SCADA Security Patching Policy. An implementation processes can then be developed to carry out the requirements set down by this policy. The Patching Policy also needs to address the issues of the testing of updates on a non-production ICS/SCADA system to ensure that no unanticipated side-effects result from updates being applied.

4.4.2 ICS/SCADA Antivirus Update Policy

Often cumbersome procedures need to be carried out for anti-virus updates on ICS/SCADA systems. Organisations should develop an ICS/SCADA Antivirus Update Policy. This policy should mandate the frequency of updates and look to put in place procedure for the distribution of anti-virus signature services locally within the ICS/SCADA systems. This should also address the issue of testing of anti-virus updates on a non-production ICS/SCADA system.

4.4.3 ICS/SCADA Access Policy

As highlighted in Chapter 3, ICS/SCADA Access Policies rarely exist and in many of cases generic logins are used. It is also evident that the access mechanisms to these devices are not maintained and updated as employees depart an organisation. An ICS/SCADA Access Policy will help to address these issues and allow for the development of procedures that can be put in place to reduce the exposure.

4.5 ICS/SCADA Security Program

The process of developing policies and procedures should form part of a much larger ICS/SCADA Security Program. This security program should be consistent with and integrated into the existing IT security experience, programs and practices. The ICS/SCADA Security Programs needs to be tailored to the specific requirements and characteristics of the ICS/SCADA technologies and environment.

Once in place the ICS/SCADA Security Program should be reviewed and updated regularly to reflect any changes in technology, operations, standards, regulations and security needs of the organisation. This also needs to be done to identify new vulnerabilities so that when a system is safe it stays safe.

The important aspects of an ICS/SCADA security program will be examined later in Chapter 8 where I will examine ICS/SCADA Security Standards and Procedures. In Chapter 10 , I will outline a set of recommendations for the creation of an ICS/SCADA Security Program that organisations in Ireland can use to protect their ICS/SCADA systems.

4.6 Findings

The absence of policies and procedures for the secure design, implementation and operation of ISC/SCADA systems leave these systems vulnerable. This can lead to insecure ISC/SCAD architecture and design, insecure personnel guidelines, lack of security audits and lack of disaster recovery planning.

Security policies are the cheapest control but are difficult to implement when it comes to security related issues. Organisations need to develop well thought ICS/SCADA policies as part of an overall ICS/SCADA Security Program which integrates with existing IT Security Policies.

CHAPTER 5
SCADA Platform Vulnerabilities

The vulnerabilities that are associated with the ICS/SCADA platform are broken down into four key areas by the NIST special publication 800-82. This document is used extensively in this chapter to explore these vulnerabilities [14]. These areas are

1. Platform Configuration Vulnerabilities
2. Platform Hardware Vulnerabilities
3. Platform Software Vulnerabilities
4. Platform Malware Protection Vulnerabilities

5.1 Platform Configuration Vulnerabilities

The configuration vulnerabilities of ICS/SCADA systems are quite complex and can cover a wide amount of devices which have to be configured and maintained. In many cases these vulnerabilities are not unique to ICS/SCADA systems and are generic to many Information Systems commonly used by organisations. In this section, I outline a broad selection of these configuration issues which may lead to vulnerabilities.

ICS/SCADA systems typically utilise OS's based on the Linux or Windows platforms. They also utilise either off-the-shelf or bespoke application software. No matter how well written these OS and application software are, they can contain coding bugs. A bug or security hole in computer software can be an error, flaw, mistake, failure, or fault in a computer program that produces an incorrect or unexpected result or causes the software to behave in unintended ways. Typically they cause a program to crash or freeze, leading to a denial of service. They can also enable a malicious user to develop an exploit to bypass access controls in order to obtain unauthorised privileges to systems.

The bugs in OS's and application software that exist were never identified during the testing phase of the software development process. They can be identified by individuals examining the source code or decompiling the OS's and applications software. Once a bug is found the attacker can write and distribute an exploit to take advantage of this security hole. Many of these exploit writers don't make use of these exploits themselves and move onto identifying and exploiting further security holes. These exploits usually become available on the Black-Market where they can be bought or rented to incorporate into malicious attacks.

When an exploit of a security hole has been released it is known as a "Zero-Day". It will take a varying amount of time before the software developers, whose software has the security hole, to become aware of the problem. The developers then set out to write, test, release, distribute and install a patch on systems to fix the security hole in their software. The delay in this process of applying fixes gives attackers a window of opportunity to gain control to either, alter the system resources, gather information or cause disruption. Ultimately the attackers' goal is to make money before the security hole is patched.

There are many well-known vulnerabilities on the Linux and Windows platforms as well as the associated Application Software. Microsoft generally releases urgent patches as soon as the patch is ready but for less urgent patches these are not released until the first Tuesday in each month, commonly called "Patch Tuesday". The Adobe suite of applications is receiving press attention at the moment due to the amount of bugs that are in their applications software. In the case of ad-hoc/bespoke applications, these are rarely patched as these are generally one-off application where no resources are available to update.

The delay in the process of getting a security patch applied to ICS/SCADA systems, is a major issue to ICS/SCADA. If a patch is unavailable, or if available but not applied, there is the potential for major disruption. In the case of the Stuxnet virus, see Appendix 2, it utilised two unpatched previously known vulnerabilities and two new zero-day bugs on the Microsoft Windows platform to gain access to ICS/SCADA systems. A recent news article [19] highlighted that at the application software level there are currently "at least 34 vulnerabilities in widely used SCADA programs sold by four different vendors".

In Chapter 3 it was highlighted that many companies are still using out-dated and un-supported applications and OS. Some organisations are still using Windows NT4, which is not supported by Microsoft and thus no up to date security patches are released for this OS. All ICS/SCADA systems infrastructure such as servers, software applications, network devices such as switches, RTUs and PLCs need regular updating to patch for new vulnerabilities identified. One such vulnerability to PLC devices was identified in a recently issued alert from the US Industrial Control Systems Cyber Emergency Response Team (ICS-CERT) [20]. They had identified weaknesses in the Ethernet Network interface of Siemens PLCs. Certain network conditions led to a stopping of the

control program within the PLCs by placing it in a shut-down state. This security alert lead to Siemens releasing a firmware update, CPU firmware version V 2.0.3, to patch this vulnerability on these systems [21]. Security patches must first be tested on an off-line test environment which is designed to replicate the operation of the systems being patched. Security patches that are applied to systems without firstly being thorough tested can have side-effects that could bring down the ICS/SCADA operations of an organisation.

The default configurations of application programs and OS's can lead to vulnerabilities on ICS/SCADA systems which were not anticipated. Often insecure unnecessary ports are opened and unnecessary services and applications can be running on hosts that can be exploited.

The configuration settings of ICS/SCADA device need to be securely stored and available in the event of accidental, or malicious, configuration changes to these systems. Having these configurations available will also provide for rapid recovery in the event of hardware failure where so fail-over redundancy is built in. This will help to reduce the unavailability period of devices and system interruption. Documentation of the configuration changes should be carried out in a change management process to request, authorise and implement configuration changes in a structured manner.

The storage of ICS/SCADA specific information such as architecture, passwords and configuration information on portable devices should be protected with a layer of security such as encryption. This will help in the event of loss, or theft, of these devices so that no sensitive information is lost. This will need a mobile device policy and procedure to protect against such scenarios.

The lack of passwords on ICS/SCADA systems is an issue that leads to vulnerabilities. Often systems have not been configured with passwords for system login, system power-on or system screen saver. Controls should be in place to mandate the use of passwords, clear definitions on the content of passwords and their period of use. The creation of a policy on the use of passwords will reduce the likelihood of unauthorised access to these systems. Weak passwords can be guessed or algorithms such as dictionary attacks used to gain unauthorised access to devices. The default shipping passwords on equipment should not be used as this provides attackers with a well-known route to access devices. Once strong passwords are in place, they should be kept

confidential by not writing them down, not sharing them, not sending them over un-encrypted communications and by training users on social engineering password recovery techniques.

Often when these systems are configured users are given an inadequate or inappropriate access control rights. Often the default settings give administrator level access which can leave ICS/SCADA systems vulnerable and lead to accidental or un-intention modification to the configuration settings. If the level of access is set too low it could lead to users not being able to carry out certain critical tasks. An access control policy needs to be developed as part of ICS/SCADA security process to mandate to level of access for users of these ICS/SCADA systems.

5.2 Platform Hardware Vulnerabilities

The hardware vulnerabilities of ICS/SCADA systems all relate to specific physical devices associated with the ICS/SCADA systems. In this section I outline a broad selection of these hardware issues which may lead to vulnerabilities on these systems.

Often the physical equipment may not be available for the non-production testing and validation of patches and anti-virus updates. This can lead to security measures being implemented into the production and can lead to unanticipated consequences that effect availability of the production systems.

The physical security of ICS/SCADA hardware devices is a concern where devices are in remote locations which are difficult to secure. Measures need to be taken to protect against physical theft, destruction/damage/interception of data as well as insuring that devices are not, accidentally or otherwise, disconnected from communication networks.
ICS/SCADA systems now provide for remote access by engineers and vendors for support and maintenance purposes. Attackers can often use these remote access techniques to gain unauthorised access to ISC/SCADA systems. Controls need to be put in place to mitigate these risks.

The identification and documentation of ICS/SCADA assets is an important aspect of these networks which needs to be carried out. If all devices are not identified and plans to secure them put in place, back-doors into seemingly secure systems may exist.

The hardware used on ICS/SCADA systems is susceptible to Radio Frequency (RF) and Electro-Magnetic Pulse (EMP) interference. These can cause damage to equipment by

producing damaging current and voltage surges. Sufficient shielding, grounding and suppression techniques are needed to counteract this vulnerability.

Interruptions to these power-supplies will lead to the unavailability of ICS/SCADA systems. Emergency and redundant backup power capability will be needed in organisations to mitigate the risk. The loss of environmental controls such as air-conditioning could lead to overheating of devices that can lead to unavailability of ISC/SCADA systems. Controls are needed to inform of environmental control failure.

Lack of redundancy for ICS/SCADA system devices can be quiet catastrophic and potentially lead to significant loss of production. Controls such as automatically switching redundancy, redundant power-supplies, on devices can help mitigate the risks associated with this vulnerability.

5.3 Platform Software Vulnerabilities

The software vulnerabilities of ICS/SCADA systems generally relate to application software which these ICS/SCADA systems utilise. In this section I outline a broad selection of these software issues which may lead to vulnerabilities on these systems.

The software applications that are used to implement ICS/SCADA systems can be vulnerable to buffer-overflow bugs which can be used by attackers to launch attacks on these systems. When writing data to a program buffer, it may overrun the buffer's boundary and overwrite to adjacent memory. These bugs if known can be triggered by inputs from attackers that are designed to execute malicious code or alter the way the programs operate. This may lead to erratic program behaviour, memory access errors, incorrect results, an application or system crash, or a breach of system security. Buffer-overflows are used by many attackers to launch malicious exploits. Coding techniques such as boundary checking should have been used during software development to limit this type of vulnerability. Once buffer-overflow bugs are identified in application they need to be patched to mitigate the risks associated with them.

In some cases software applications are designed and written with security capability built in but these capabilities may not be enabled with the default instillation processes. Users often incorrectly assume that applications which installed, have a layer of security which protects. The status of security features are usually buried deep within applications and there is no intuitive way of determining if the protections are actually

enabled. This often provides organisations with a false-sense-of-security that they are protected from specific vulnerabilities.

The ICS/SCADA system software applications can be vulnerable to Denial-of Service attacks (DoS) that can delay the operation and functionality of these systems. In extreme cases these attacks block off access to systems completely. They can be launched by forcing the application into the mishandling of undefined, poorly defined, or illegal conditions. These can also be triggered remotely by an attacker with the use of malformed, illegal of unexpected packets to these devices.

Vulnerabilities to these systems can be introduced when unneeded system services are running on the platform by default. These services can either tie-up limited system resources or be exploited to gain unauthorised access to the systems. Reducing the attack surface by disabling these supurphulous services is essential and this also frees up system resources.

Research into findings for ICS/SCADA system vulnerabilities lead to further system vulnerabilities. This is a so called “catch 22” scenario where if problems are not identified they can’t be fixed but this in turn leads to attackers being provided with new attack avenues to explore and exploit. Ethical methods of notifying software developers of vulnerabilities prior to publication of vulnerabilities should be used in this scenario to allow security holes to be patched before they are made common knowledge.

Inadequate application software authentication and access control for the configuration and programming of devices can introduce vulnerabilities these systems. This could provide attackers with the ability to corrupt devices on the ICS/SCADA system.

The absence of Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) leaves these systems vulnerable. These IDS/IPS devices provide an additional layer of security that will help to protect these systems as they detect and prevent attacks on the ICS/SCADA system. ICS/SCADA systems can be left vulnerable to attacks if logs detailing system activity, are not created, maintained or reviewed. Logs need to be stored in a central, non-default, location to protect against attackers hiding their activities while accessing these systems. Logs also need to be configured to automatically trigger notification alerts to inform of incidents that need investigation so that system recovery purposes can take place. If incidents that are triggered by IDS/IPS

or logs are not investigated in a timely manner, vulnerabilities will not be detected and countermeasures will not put in place to counteract them.

5.4 Platform Malware Protection Vulnerabilities

ICS/SCADA systems are vulnerable if Anti-Virus software is not installed on hosts that these systems utilise. This could lead to performance degradation, loss of system availability and the capture, modification and deletion of system data by attackers. Antivirus software will prevent these systems from being infected but it needs to be kept up-to-date with the latest virus definitions, otherwise new attacks won't be detected. Prior to antivirus protection software being applied to ICS/SCADA systems, thorough testing is required to insure that this security protection doesn't adversely affect the operation of the systems being protected.

5.5 Findings

The vulnerabilities associated with ICS/SCADA platforms can be broken down into Configuration, Hardware, Software and Malware Protection. These vulnerabilities cover a wide range of ICS/SCADA systems that organisations, that use them, need to be aware of and put in controls to mitigate the risks associated with these vulnerabilities. Use of standard vulnerable OS, vulnerable application software/firmware, shared and unencrypted passwords, lack of access control and absence of techniques such as IDS and latest anti-virus updated were explored. It was also identified that vulnerabilities other than cyber-attack need to be addressed to provide availability, integrity and confidentiality of ICS/SCADA systems.

CHAPTER 6
ICS/SCADA Network Vulnerabilities

The vulnerabilities that are associated with the ICS/SCADA networks are broken down into six key areas by the NIST special publication 800-82. This document is used extensively in this chapter to explore these vulnerabilities [14]. The vulnerabilities I will explore are

- Network Configuration.
- Network Hardware.
- Network Perimeter.
- Network Monitoring and Logging.
- Communication.
- Wireless Connection.

6.1 Network Configuration Vulnerabilities

Network configurations vulnerabilities, with ICS/SCADA systems, are not unique to these systems and affect a wide range of other systems. Often the network infrastructure equipment has been set up to meet the needs of the corporate business requirements which commonly have security holes and weak network security. If these holes are not identified and rectified they could lead the ICS/SCADA systems being vulnerable to backdoor network attacks. Access Control Lists (ACL) are needed so only administrators have direct access to ICS/SCADA assets and disable all other direct connections to ICS/SCADA devices.

Vulnerabilities are introduced to ICS/SCADA systems by use of default configurations on network equipment. These configurations can have insecure and un-needed open ports and as well as other exploitable services. Poorly configured firewalls rules and router ACL's will also allow unnecessary network traffic that ties up network resources.

Procedures for the storage and backup of network device configurations are required. This will mitigate the vulnerably of accidental, or attacker changes to configurations. Change Management techniques to control configuration settings should also be used.

The use of either inadequate or, no access control techniques on networks may lead to the unauthorised access to these networks devices with administrative privileges. This can allow for monitoring and disruption of the ICS/SCADA traffic.

The transmission of passwords over network infrastructure in plain-text is poor practice. Passwords can be captured by attackers and used to gain unauthorised access to network equipment to monitor and disrupt ICS/SCADA activity. The access passwords on network devices should be changed on a regular basis so that if an attacker becomes aware of a device password, they don't have access to network devices for very long.

6.2 Network Hardware Vulnerabilities

Inadequate physical protection of network hardware will leave the ICS/SCADA systems vulnerable. The physical access to this network equipment should be controlled to prevent damage or disruption to ICS/SCADA services. This will require locating equipment in secure locations and restricting access to those with the correct privileges.

Unsecure USB and PS2 ports on hosts and the network equipment can lead to unauthorised connection of media and keystroke loggers. The information gathered by these can be used to gain unauthorised access to these systems.

The access to network equipment and connections on ICS/SCADA systems should be restricted. Open access could lead to theft and destruction of data and hardware and the unauthorised change of security features. This could lead to unauthorised interception and manipulation of network activity and the disconnection of links or connection of unauthorised links.

The lack of redundant communication links or fail-over protection within networks could cause unavailability issues with ICS/SCADA systems. Lack of redundancy could lead to substantial un-availability of services while replacement equipment is sourced and configured.

6.3 Network Perimeter Vulnerabilities

The segregation of the automation networks and the corporate business networks is an important security control. These networks have vastly different requirements that do not map easily together on a unified network. The corporate network could easily consume network resources therefore delaying and/or disrupting the ICS/SCADA communications. The security perimeter of the automation and corporate networks needs to be clearly defined so that security controls are deployed and configured correctly.

Properly configured firewalls are required between the corporate and the automation networks. Firewalls will help to control the flow of information between these networks so that threat of hacker access and the spread of malware is minimised. They also reduce the threat posed by monitoring techniques and limiting the unauthorised access to the ICS/SCADA systems.

Services such as DHCP and DNS are often placed centrally in organisations corporate IT infrastructure. When the segregated automation network needs access to these services it is often dependant on getting these services from the corporate network. This technique may not meet the availability or reliability needs for the automation network and could lead to interruption of service provided by the ICS/SCADA systems. The provision of these services on the automation network needs to be considered.

6.4 Network Monitoring and Logging Vulnerabilities

Network devices like firewalls and routers produce activity logs. These logs need to be saved to secure a central location so that they can be reviewed if needed to identify attackers' attempts to gain access to these systems. These logs may also prove useful when investigating security incidents carried out by attackers. Failure to carry out regular reviews of these logs for suspicious activity can leave the ICS/SCADA systems open to attack.

Regular security monitoring of network devices is required to identify incidents and problems with security controls that can affect the ICS/SCADA systems. ICS/SCADA systems also need to be monitored for misconfiguration or failures of security features.

6.5 Communication Vulnerabilities

Monitoring for rogue or unauthorised connections to the ICS/SCADA network needs to be carried out on a continual basis. These rogue and unauthorised communications are often via un-patched and unsecure devices that leave backdoor access to the ICS/SCADA systems. Security techniques such as mutual authentication will help to control this vulnerability.

The monitoring of standard communication protocols such as Telnet, FTP and NTS that use plain-text messaging can be easily performed by attackers. This protocol traffic can be captured as they traverse the communications network and be used to gain information on the ICS/SCADA system. This information can be used to mount attacks or gain unauthorised access to these systems.

There are other communication vulnerabilities that fall into this category but I have decided to treat these separately as SCADA Protocol Vulnerabilities in (see Chapter 7).

6.6 Wireless Connection Vulnerabilities

Strong mutual authentication between wireless clients and access points is required to ensure that clients don't connect to the rogue access points of attackers. By using rogue access points, attackers can easily ascertain user credentials and other information to gain access to the ICS/SCADA systems through a Wireless Automation Network. Mutual authentication will help to ensure that attackers do not connect to the Wireless ICS/SCADA Network. Data being transmitted across these wireless networks should also have strong encryption to insure that attackers do not obtain access to it as it crosses the transmission medium.

6.7 Findings

The Network Vulnerabilities associated with the ICS/SCADA systems are split into Configuration, Hardware, Network Perimeter, Monitoring and Logging, Communication and Wireless Connection. In this chapter I have identified vulnerabilities that can be used to disrupt or monitor ICS/SCADA operations and network activity.

Many of the vulnerabilities identified are common to other networking technologies so these vulnerabilities are not new to IT network and security professionals. There are many well documented techniques which can be employed to address these vulnerabilities which can be implemented on ICS/SCADA networks to make them more secure.

CHAPTER 7
ICS/SCADA Protocol Vulnerabilities

This chapter focuses on attacks on the ICS/SCADA protocol specifications. The NIST 800-82 defines the vulnerabilities associated with the ICS/SCADA Protocols as being a mix of Platform Software Vulnerability as covered in section 5.3 and Communication Vulnerabilities covered in section 6.5. There is good case for this approach however I believe that as this is such a large area of potential vulnerability, it should be treated separately and more attention should focus on this side of ICS/SCADA systems. While these protocols are implemented in software, it's not an implementation problem as there are fundamental design problems with these protocols that need to be addressed.

The commonly used communications protocols of MODBUS/TCP, DNP3 and PROFINET, as already described in section 1.5, were designed initially as serial communication protocols. As Ethernet IEEE 802.3 became more available the serial protocols were ported to the IP based protocols.

These serial protocols were designed for use on the dedicated serial links has no requirement to secure the communication between devices. The porting of these protocols did not introduce any additional security features and currently these protocols lack even the most basic security features. As a result these protocols are inadequate to allow for the secure transmission of data across a corporate network or the internet as they provide no facilities for end-to-end authentication, authorisation, integrity, non-repudiation or any methods to encrypt the clear-text ISA/SCADA. Attackers can take advantage of vulnerabilities and misuse the protocols to monitor and disrupt ICS/SCADA systems. This can be done using malicious techniques such as those outlined below: [22]

- Unauthorised Command Execution: The lack of authentication between MTU and RTU can be used by attackers to forge packets and send then directly to the MTU.
- SCADA-DOS: Using the similar principal above, an attacker can forge meaningless SCADA packets and consume the resources of the RTU.
- Man-in-the-Middle attacks: The lack of integrity checks allows attackers to access the production network to implement typical Man-in-the-Middle (MITM) attacks, modifying the legal packets sent by the master.
- Replay-Attacks: The lack of anti-replaying mechanisms allows attackers to reuse captured legitimate SCADA packets.

- Compromised Masters: As non-repudiation mechanisms are not implemented, it is hard to proof the trustworthiness of malicious Masters, which could have been compromised.

Malicious protocol attacks commonly use hacking techniques such as packet interruption, packet interception, packet modification and packet spoofing [23]. See figure 7.1.

- Interruption blocks the transmission of the packets between the source and destination. This is commonly used for Denial-of Service attacks.
- Interception of packets occurs when packets are optioned by the attacker as they traverse the communications media. The packet in not modified, rather a copy taken and commonly used for eavesdropping and replay attacks.
- Modification can occur where the packets are routed firstly, to the attacker where they are modified and then forwarded to the original destination. This technique is commonly used in MITM attacks.
- Fabrication where the attacker crafts a packet and sends it to a destination as if they were sent by a legitimate source. This is commonly used to spoof devices such as sensors and even user identity.

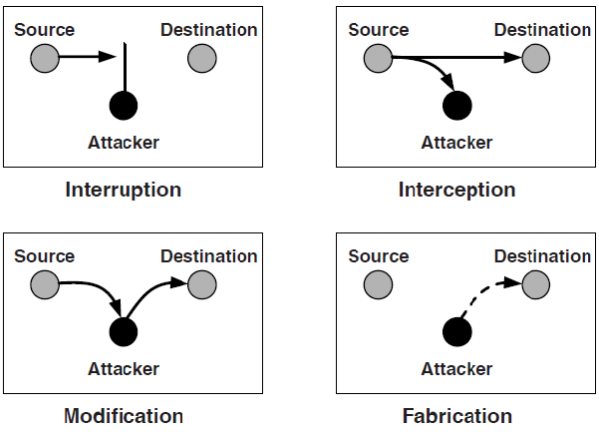


Figure 7.1 Protocol traffic attacks [23]

Jeff Dagle demonstrated the effect of this type of attack during a security conference in 2003. He hacked into a test-bed system and fabricated a packet that tripped an electrical circuit breaker. Once tripped, the breaker sent a message to the SCADA server to inform it of this trip but the message was modified during transmission. The SCADA controller did not know the breaker was open and so no corrective action was taken. This was a simulation of a denial-of-service attack that used message fabrication to send a spoofed message to the circuit breaker and then used packet modification techniques to hide the fact that the circuit breaker was in an unexpected state. [24]

Similar techniques were used in the best known SCADA attack to date. In Iran the Bushehr Nuclear facility uses SCADA systems to control centrifuges that were used during the uranium enrichment process. These centrifuges have a maximum RPM for their safe operation. The SCADA systems controlled this ISC equipment and kept them within the thresholds as set down by the equipment manufacturers.

The Stuxnet Virus used a verity of sophisticated techniques to install itself on SCADA systems without detection, as are outlined in Appendix 2. Consequently the SCADA systems were infected and they sent commands to the centrifuges that pushed them outside safe operation limits by fabricating packets. It then modified the messages being sent back by these centrifuges to the SCADA central control to falsely claim that the system was working normally. When the SCADA system received these modified messages it took no corrective actions to bring these devices back into safe operation guidelines. This led to damage of many of the centrifuges, and made the remainder un-operational due to their operation outside their design specification. It is believed that the Stuxnet Virus has set back the Iranian Nuclear program 3-5 years. These examples clearly demonstrate the weakness of ICS/SCADA systems at the protocol level.

As part of this examination of the ICS/SCADA communications protocol vulnerabilities I will concentrate on four areas.

1. Current research into ICS/SCADA protocol vulnerabilities.
2. Security measures to protect current ICS/SCADA protocols.
3. Current research into addition of security features to existing ICS/SCADA protocols.
4. Current research into new secure ICS/SCADA protocols.

7.1 Current research into ICS/SCADA protocol vulnerabilities

Recently there has been a lot of research carried out in the area of ICS/SCADA protocol vulnerabilities, in attempt to identify weaknesses and enhance security awareness. This research can enable vendors and asset owners to design architectures, configure equipment and operate systems in a manner that addresses the identified vulnerabilities. A piece of research into the vulnerabilities of the MODBUS protocol was presented in a publication in 2008 [25]. Entitled “Attack taxonomies for the Modbus Protocols” it generalised the attacks on ICS/SCADA systems and networks into three categories:

1. Attacks that exploit the communications protocol specification.
2. Attacks the exploit vendor implementations of the MODBUS protocol.

3. Attacks that target the support infrastructure of information Technology, networking and telecommunication assets.

This research focused on attacks that exploit the MODBUS protocol specification. The methodology used by this group of researchers involved analysing the MODBUS protocol specification and identifying weaknesses. Attacks were then formulated to exploit these weaknesses. Each attack was then analysed to assess its ability to intercept, interrupt, modify and fabricate the primary targets (MTU's, RTU's, network/communication paths and messages). This research used packet sniffers and packet injectors to block, modify and fabricate the MODBUS/TCP messages. They categorised the attacks on MODBUS/TCP into Broadcast Message Spoofing, Baseline Response Relay, Direct Slave Control, Modbus Network Scanning, Passive Reconnaissance, Response Delay, Rogue Interloper, Irregular TCP framing, TCP FIN Flood, TCP Pool Exhaustion and TCP RST Flood. This research identified 28 distinct attacks and 113 attack instances where exportation of a MODBUS asset occurred. Table 7.1 shows a summary of the impact of MODBUS/TCP attacks on the target assets assessed in this research. The effects of these attacks range from obtaining network or device configuration data to corrupting outstation devices and seizing control of the master unit. This research concludes by saying that their analysis was not fully comprehensive and the authors believe that many more attacks remain to be discovered.

Impact of Modbus TCP attacks on target assets				
28 distinct attacks (113 instances)	Master	Field device	Network path	Message
Interception		8 obtain field device data	5 obtain network data	3 read message
Interruption	16 DoS master	21 DoS field device	7 DoS network path	12 block message
Modification	8 bad data in master	12 bad data in field device 3 control field device	3 bad traffic	2 bad data in message
Fabrication	4 control process	3 fabricated field device	3 fabricated network path	3 fabricated message

Table 7.1 Impact of Modbus TCP attacks on target assets [25]

In a more recent publication [23] similar techniques were used to analyse the DNP3 protocol. It identified 28 attacks and 91 attack instances where exploitation of DNP3 assets occurred. Both of these pieces of research into the vulnerabilities of protocols, commonly used by ICS/SCADA systems, show that there a large proportion of high-impact attacks, especially those involving the interruption, modification and fabrication of ICS/SCADA assets.

7.2 Security measures to protect current protocols.

Currently there are some a number of options available to organisations to help protect against the vulnerabilities at the protocol level. One possible solution that organisations may implement is to use SSL/TLS or IPSec that wrap the communication with a layer of security and does not make changes to the protocol itself. A study examining the use of these two options when securing SCADA protocols [26] found that SSL/TLS provides fast, standard and economical solution in the short run but it had some limitations. IPSec was more expensive to implement but could make the SCADA protocols more secure.

7.2.1 Secure Sockets Layer / Transport Layer Security

Secure Sockets Layer(SSL)/Transport Layer Security(TLS) protocol is commonly used to protect client/server communications over the Internet from threats of eavesdropping, tampering and message forgery. It allows authentication of sending and receiving devices and provides integrity by using digital signatures and privacy using encryption. It is a well-established protocol and commonly used in web browsers, web servers and other internet systems. It is commonly used to secure internet traffic carried by HTTP to form HTTPS.

TLS was initially implemented on top of any of the Transport Layer protocols, encapsulating the application-specific protocols such as HTTP and FTP. It is primarily used with Transmission Control Protocol (TCP). It is also commonly used to tunnel an entire network stack to create a Virtual Permanent Network (VPN). Detailed information on the SSL components may be found in NIST Special Publication 800-113 entitled “Guide to SSL VPNs”.

Organisations can use open source versions of the SSL/TLS protocol such as “Open SSL” to very quickly and cheaply place a layer of security in their vulnerable ICS/SCADA communications protocols [26].There are however some problems associated with this approach SSL/TLS does not provide for non-repudiation, end-to-end security and has some performance overheads associated with it.

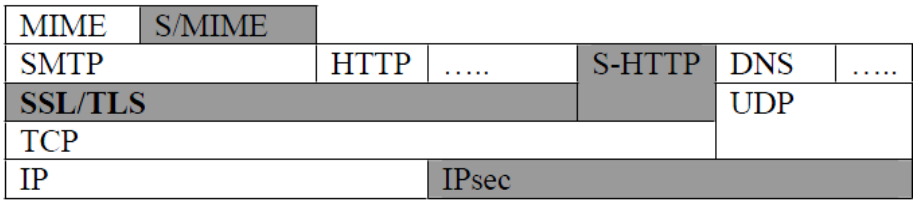


Figure 7.2 Protocol Stack [26]

7.2.2 Internet Protocol Security

Internet Protocol Security (IPsec) is a protocol suite for securing Internet Protocol (IP) communications by authenticating and encrypting each IP packet of a communication session. IPsec also includes protocols for establishing mutual authentication between agents at the beginning of the session and negotiation of cryptographic keys to be used during the session.

IPSec provides many of the same security features as SSL/TLS but as it operates at a lower layer than SSL/TLS (see Figure 7.2), it can secure any TCP or IP while SSL/TLS can only secure TCP traffic. SSL/TLS only prevents arbitrary packets from being inserted into a session and is not able to prevent a connection reset (denial of service attack) as the connection handling is done by a lower level protocol (i.e. TCP). IPSec prevents both arbitrary packets entering a Transport-Layer stream and connection resets because connection management is integrated into the secured Network Layer. IPsec also provides security for any traffic between two hosts [26]. Detailed information on the IPSec components may be found in NIST Special Publication 800-87 entitled "Guide to IPSec VPNs".

The finding of [26] indicate that there are advantages and disadvantages in the use of VPN's which use either the IPSec or SSL security protocols. It suggests that SCADA applications could be enhanced by a range of alternatives from adding authentication/encryption to inherent changes in ways in which the application works. It also proposes that more research is needed to make the protocols themselves more secure which would fundamentally improve the security and reliability of SCADA systems.

7.3 Current research into addition of security features to existing protocols.

At present there is a lot of research which explores communications the potential of improving security of protocols by adding authentication and encryption techniques to the protocols themselves as suggested by [26]. One piece of research is examining the securing the MODBUS/TCP protocol [27] called *"Secure" MODBUS protocol*. This is only at the proposal stage of development which the MODBUS Organisation will need to examine to assess its suitability and further development of the MODBUS protocol. Similar research is being carried out on other protocols to determine solutions available to make them more secure.

7.3.1 Proposed "Secure" MODBUS protocol

Research was carried out to examine the methods of introducing integrity, authentication, non-repudiation and anti-replay into the commonly used MODBUS protocol [27].The aim of this research was to design a "secure" MODBUS protocol which would insure:

- No unauthorised entity is allowed to access the content of a message.
- No unauthorised entity is allowed to modify the content of a message.
- No entity is allowed to impersonate another entity.
- No entity is allowed to negate a performed action.
- No entity is allowed to reuse a captured message to perform an unauthorised action.

Confidentiality requirement for MODBUS messages were not included as confidentiality does not mitigate any of the attack scenarios. In addition confidentiality is generally implemented using encryption, which is expensive and introduces considerable overhead that can impact real-time performance.

The integrity of a Secure Modbus Packet is guaranteed using the hashing function SHA2 which computes a secure digest of the MODBUS packet, which is then transmitted with the packet. On receiving the MODBUS packet, the receiving device computes a new SHA2 digest. If the digest values differ, then the packet is known to have been modified in transit. However this does not prevent an attacker from creating a malicious Modbus packet, computing its SHA2 digest and sending the malicious packet and the digest to the receiver. Authentication is used by RSA-based signature scheme by signing the SHA2 digest with the RSA private key of the sending device. Receiver verifies the authenticity of the packet using the sender's public key. This ensures that the packet was created by a trusted sender and that the packet was not modified on route. This RSA-based signature scheme also provides a non-repudiation mechanism as only the owner of the RSA private key could have sent the packet. The SHA2 hashing and RSA signature schemes do not prevent an attacker from re-using a "sniffed" Modbus packet signed by an authorised sender. The "secure" MODBUS protocol also has a mechanism that enables the receiver to discriminate between a "new packet" and a "used packet." This is accomplished by incorporating a time stamp (TS) in the packet being transmitted (which is used by the receiver to check the "freshness" of the received packet). Network

Time Protocol (NTP) time stamps are used for this to evaluation this freshness with high precision. This will need NTP server in the SCADA architecture to provide a reliable clock for all communicating devices.

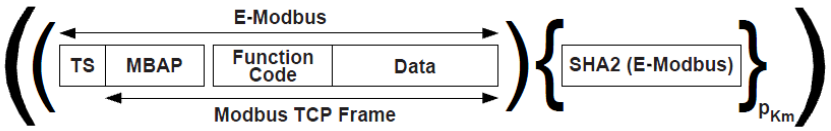


Figure 7.3 "Secure" MODBUS Packet [27]

After evaluation of this proposed protocol using a SCADA test-bed it was found that it provided good security functionality without significant latency to communications of significant cost to implement. However it does not address the scenario where by attackers seize control of a master and send malicious packets or where the sender's private key is compromised. This group is currently looking at methods to address these issues and a filtering unit to identify suspect MODBUS messages using IDS/IPS technology [28] . Further studies are examining the protection of keys and methods to refine signature scheme for improved real-time performance.

7.4 Current research into new secure protocols.

This section outlines some of the work being carried out to develop new ICS/SCADA protocols which are much more security conscious and move away from the limitations of using older serial based protocols to do this.

7.4.1 Secure SCADA Communications Protocol

Secure SCADA Communications Protocol (SSCP) was developed by the Pacific Northwest National Laboratory (PNNL) [29] (a group set up by the US Department of Energy to research the problems in energy, the environment and national security). As SCADA systems are used for many aspects of energy generation and distribution, this group has performed large amounts of research into the security of SCADA systems. SSCP was designed to ensure SCADA system data integrity through message authentication and optional encryption [30] . This protocol marks the original SCADA messages with a unique identifier and authenticator and sends it to the lower layers for delivery. At the receiving device, once the transportation layers are stripped off, it will scan the identifier and validate the message to ensure that the information comes from a trusted source and has not been altered in transit. Unauthenticated commands are logged and reported as errors.

SSCP uses symmetric key cryptography and protects communications by encapsulating the original message within a header and authenticator. To do this the sender and receiving devices use a Diffie-Hellman key exchange to establish a session and generate cryptographic keys for session authentication and encryption. A static encrypted key is used to secure the Diffe-Helman key during session negotiation. The two symmetric authentication session keys provide authenticity of network data. The set of encrypted keys are available for data encryption. A nonce is also used to provide each packet with a unique value. The authentication key is used to calculate a hashed message authentication code (HMAC) using SHA-1alogirithm. The HMAC, message and unique packet value are sent to the receiver who uses the message, the unique packet value and the symmetric key to compute the HMAC again and compares it with the received HMAC. If these values match then the message has been authenticated [31].

7.4.2 SEL Encryption and Authentication Protocol

Currently there are some peripheral products available to help organisations to secure ICS/SCADA communication protocols. Schweitzer Engineering Laboratories (SEL) have developed the SEL Encryption and Authentication Protocol (SEAP) that secures the operator communication channel with strong message encryption and authentication [30]. It allows operators to log into the modules securely as each operator has a static AES encryption key, HMAC authentication key, user name and password. An encryption key provides confidentiality during the session negotiation process. The authentication key provides authentication during the session negotiation process.

This is achieved through the use of a Schweitzer SEL-3045 PCMCIA standalone cryptographic module that resides in the host device to secure data on a particular network. It is designed to protect sensitive SCADA traffic as it crosses the networks.

During the session negotiation process, the user name and password are securely provided to the module in order to authenticate the operator and assign the appropriate access privileges. Session encryption and authentication keys are transported by the module and are used to provide confidentiality and

authenticity of each frame for the remainder of the session. These keys are transported, encrypted and the operator's with 128 bit AES encryption key.

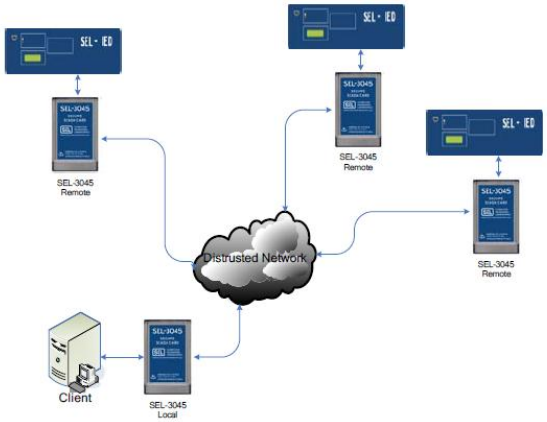


Table 7.3 SEL-3045 Architecture [30]

7.5 Findings

As it currently stands, there is very little security in the ICS/SCADA protocols. This I equate to the first inception of Wireless IEEE 802.11 Ethernet networks. Security was not designed into the WiFi technology until these networks matured and became reliable. The same can be said for the ICS/SCADA protocols. Initially they had no security and developed to a point where the technology is now well mature and widely used. Attention is now beginning to focus on the securing of these protocols. The tunnelling technologies of SSL/TLS and IPSec may provide an interim layer of security to be added to these protocols but in the long term more secure versions of the currently used protocols or completely new secure protocols will need to be developed and implemented.

The legacy and current generation of ICS devices do not provide for the (as-of-yet undefined) security features to make these protocols secure. These devices may need firmware updates or even replacement, in order to incorporate additional security features to these protocols. Initially these devices may be ported to the new secure protocols with the use of devices that bridge between the new security protocols and the current unsecure protocols. In the long term extra functionality on devices may be required through additional memory and processing power to run new security algorithms.

CHAPTER 8
ICS/SCADA Security Standards, Procedures and Research

ICS/SCADA systems control a wide and varied range of equipment from industrial manufacturing equipment, critical infrastructure equipment and building maintenance systems. For each industry and application of these systems there are varying levels of security required. There are both recognised and unknown standards, recommended procedures and publications that will provide guidance to organisations on the securing of their ICS/SCADA systems. Many of these standards and procedures are very similar to each other and cover similar security concepts that organisations can apply. This can lead to confusion as to which suites an organisation's needs. In this chapter I shall examine some of these ICS/SCADA standards and procedures as well explore research currently being done in the area.

8.1 International Society of Automation Security Standards

The International Society of Automation (ISA) is a non-profit organisation that assists organisations solve difficult technical automation problems [32]. They have two security standards that are relevant to SCADA systems which are ISA 95 and ISA 99.

ISA 95 is an international standard for developing an automated interface between enterprise and ICS/SCADA systems [33]. It consists of models and terminology that can be used to determine the information to be exchanged between these two distinctly different systems. The terminology section of this standard provides consistent terminology which is a foundation for supplier and manufacturer communications. The five layer model provides consistent operations models for clarifying application functionality and help define boundaries between the enterprise systems and the ICS/SCADA systems as to how information is to be used.

ISA-95 provides for different levels of distinction between ICS/SCADA network equipment as illustrated in Figure 8.1.1. Level 1 is I/O devices such as valves and instrumentation sensors that communicate through Discrete and Process Device Communication Networks to PLC and/or RTU devices. These PLCs and RTUs sit in Level 2 which also contain the Automation Network and well as the HMI's and SCADA servers. Level 3 in turn contains Manufacturing Operations Systems such as database and historical services which ICS/SCADA systems rely on. These Manufacturing Operations Systems are then connected to the Corporate Business Process Information Network with which Management Systems, which sit at Level 4,

can interact and control. The fifth layer to this model, Level 0, is the actual production process itself and doesn't cover any aspect of the ISC/SCADA system itself but rather the hardware they interact with.

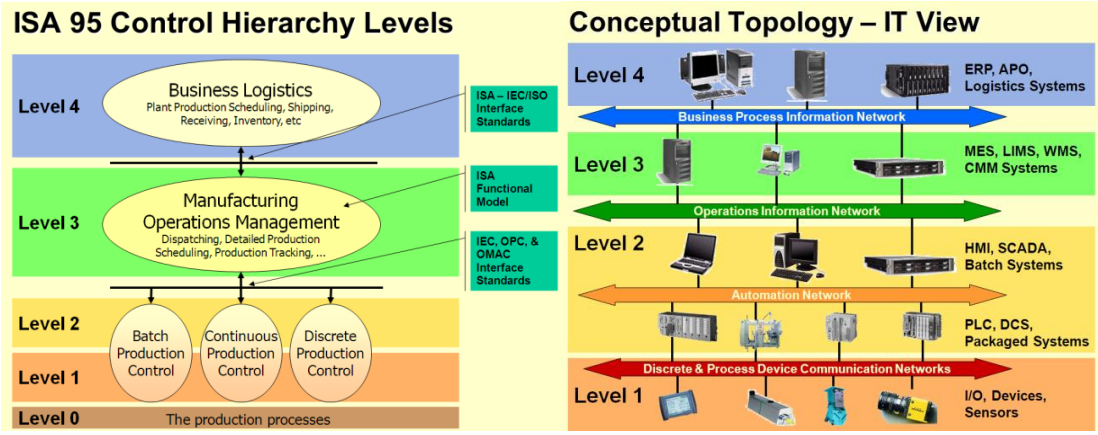


Figure 8.1.1 ISA Models [33]

ISA-99 is series is built on top of the technologies and models of ISA-95. It consists of 14 standards & technical reports that address all aspects of ICS/SCADA security (This is not an open source publication and it is difficult to access). The ISA provide professional training in this standard and certification to individuals and organisations that meet the requirements for the standard. This certification, if attained, will provide assurance to clients that the ICS/SCADA systems are secure. The ISA-99 standard focuses on containing communication within the ICS/SCADA systems to avoid having issues in one area migrate to another. There are currently four different publications for the development of this standard for ICS Security.

Part 1- Security for Industrial Automation and Control Systems (concepts, terminology and models)

Part 2- Establishing an Industrial Automation and Control System Security Program.

Part 3- Operating an Industrial Automation and Control System Security Program

Part 4- Requirements for Industrial Information and Control Systems

ISA99-Part 2 provides guidance for developing a program for the security of ICS/SCADA systems. It details process activities and key elements for establishing a "Cyber Security Management System" (CSMS) for ICS/SCADA systems. The steps involved in implementing this system are: [47]

1. Risk Analysis
2. Addressing the Risks with CSMS
3. Monitoring and Improving the CSMS

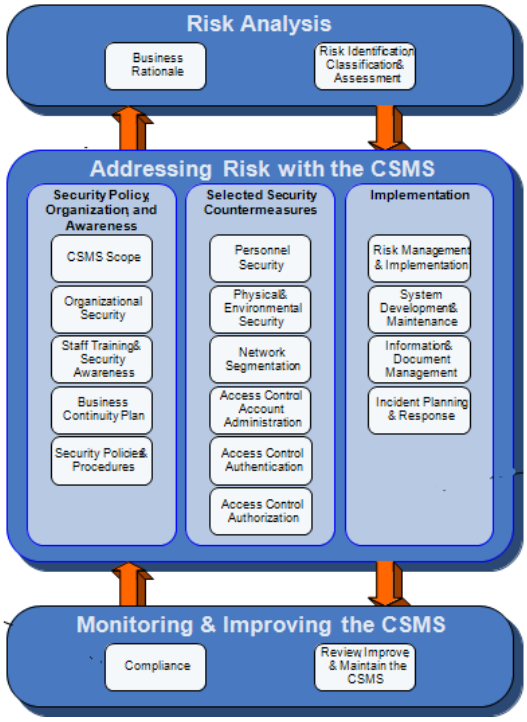


Figure 8.1.2 ISA-99 Part 2 CSMS Implementation[47]

There is also another ISA standard called ISA 100 which looks specifically at the issue of mobile and wireless devices operating in the ICS/SCADA area. Within the ISA standards there is also the possibility for organisations to attain certification to the ISA-99 standard. This will give assurance to organisations and their customers that the ICS/SCADA systems are being secured to internationally recognised standards.

8.2 ISC-CERT

In America there is a team called the United Stated Computer Emergency Readiness Team (US-CERT) whose goal it is to improve US nation's cyber security posture, to coordinate cyber information sharing and to proactively manage cyber risks US while protecting the Constitutional Rights of Americans [34]. Within this team, there is another group who focus exclusively on the security aspects of ICS/SCADA systems. This group is called the Industrial Control Systems Cyber Emergency Response Team (ICS-CERT) and its goals are to: [35]

- Respond to and analyse control systems related incidents
- Conduct vulnerability and malware analysis
- Provide onsite support for incident response and forensic analysis
- Provide situational awareness in the form of actionable intelligence
- Coordinate the responsible disclosure of vulnerabilities/mitigations
- Share and coordinate vulnerability information and threat analysis through information products and alerts one

This group has a series of advisories and reports that can be accessed on their website [36] which will notify organisations of updates in security issues regarding ICS/SCADA systems. The same group also critique publications by other organisations into issues of ICS/SCADA security. This can be found at [37].

8.3 US Department of Homeland Security Recommendations

Since September 11th 2001 concern about the threat of terrorist attacks worldwide has drastically increased. The US government is taking this threat especially seriously. Concerns about cyber-attack from for terrorist groups on the Information Technology Systems that control critical infrastructure have highlighted that ICS/SCADA systems are responsible for controlling the majority of the US vital critical infrastructure such as electricity, oil, gas, water, transportation and chemical. In a publication by The Department of Homeland Security (DHS) entitled *"Recommended Practice: Improving Industrial Control Systems Cybersecurity with Defence-In-Depth Strategies"* [38] the DHS makes recommendations on how to make these systems more secure. They recommend dividing the common ICS/SCADA architecture into clearly defined zones as an effective method to apply multiple layers of defence. To achieve the goal, firewalls, routers with Access Control Lists (ACL), configured switches, static routes and routing tables, and dedicated communications media are all required. The zones defined in this publication are External Zone, Corporate Zone, Manufacturing/Data Zone, Control/Cell Zone and Safety Zone. Figure 8.2.1 illustrates this concept.

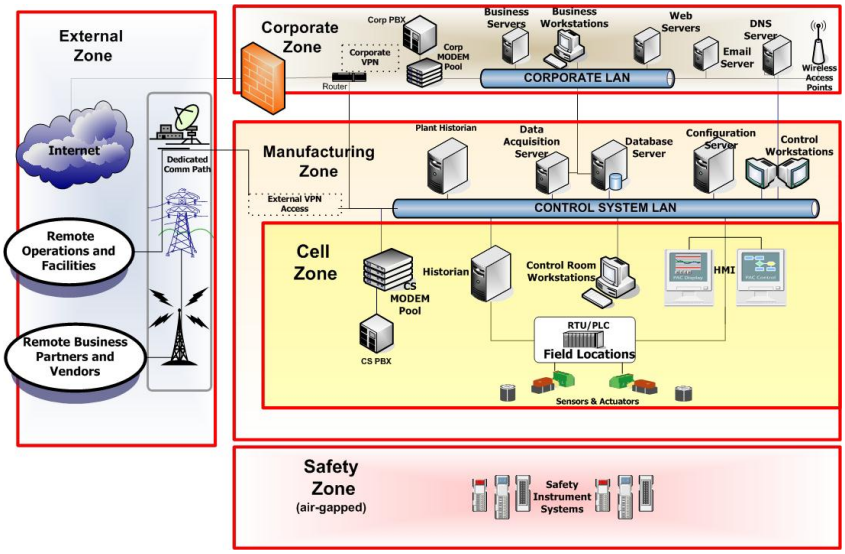


Figure 8.2.1 DHS Recommended Zones [38]

The publication recommends the use of firewalls to provide levels of defence and provide the capability to add tighter and complex rules for communication between the different network segments or zones. Figure 8.2.2 shows the recommendation that firewalls should be used between the:

1. External and Manufacturing Zones.
2. Corporate and Manufacturing Zones.
3. Field Level Device and ICS/SCADA servers (in Cell Zone)

The use of De Militarised Zones (DMZ's) of protection should also be implemented with Firewalls which are sub-networks or VLAN's. These VLAN's will contain and secure an organisation's external services to a larger un-trusted network. This adds an additional layer of security to an organisation's network as external attacker only has access to equipment in the DMZ, rather than any other part of the network. Figure 8.2.2 shows the use of DMZ's behind the firewalls of external links to un-trusted networks.

The publication also recommends the use of Intrusion Detection Systems (IDS) to monitor networks in case of unusual or unauthorised activity. IDS compare network traffic against rule sets as well as against a set of known attack "signatures". Having recognised an attack pattern or any deviation from what has been defined as normal/allowable traffic, the systems will carry out a set of instructions which will include alerting the systems administrator.

Historically many of the leading IDS products when applied to ICS/SCADA systems were ineffective when subjected to attack. Recently however, there has been a lot of research carried out to address this issue [38]. There are now several IDS options available to organisations that want to apply them on their ICS/SCADA systems which can have a significant contribution to managing each security zone. They also help expedite incident response and as logging is also an inherent function.

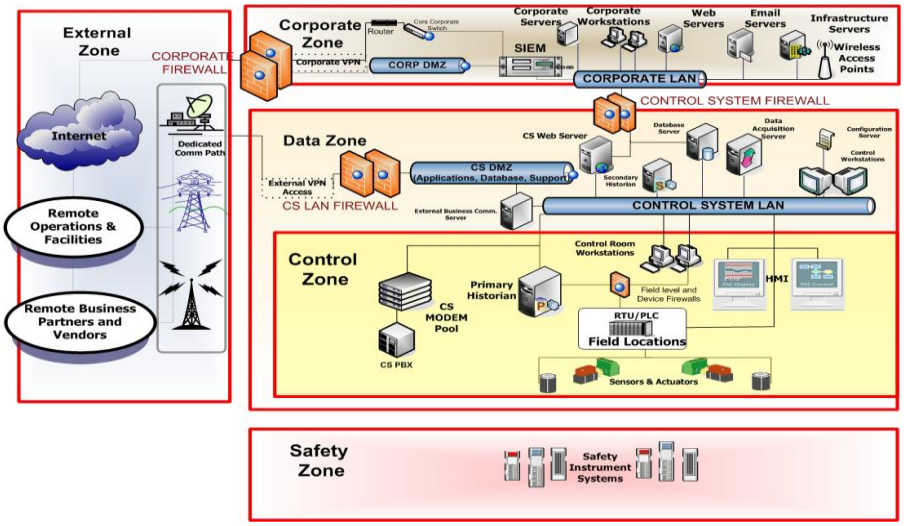


Figure 8.2.2 DHS Defence-In-Depth Architecture [38]

The DHS also recommends the use of policies and procedures such as Log and Event Management, Security Policies, Patch Management Planning and Procedures, Security Training, and Incident Response and Forensics.

The document finishes with five countermeasures to drive cyber-security activates in ICS/SCADA systems.

1. Security policies to control systems network and its individual components, but they should be reviewed periodically.
2. Blocking access to resources and services using perimeter devices with firewalls, proxy servers, host-based firewalls and antivirus software.
3. Detecting malicious activity through monitoring of log files and IDS.
4. Mitigating possible attacks by removal of the vulnerabilities.
5. Fixing core problems by updating, upgrading, or patching the software vulnerability or removing the vulnerable application.

The DHS also has a publication called *"Catalog of Control Systems Security: Recommendations for Standards Developers"* [39].This is a catalogue of practices which various industry bodies have recommended to increase the security ICS/SCADA systems from both physical and cyber-attacks. The recommendations are grouped into 19 categories to be considered and judiciously employed, as appropriate, when reviewing and developing cyber-security standards for control systems. This catalogue is not designed for any specific industry sector and can be used by all sectors to develop a framework needed to produce a sound cyber-security program. An outline of this catalogue can be found in Appendix 5.

8.4 US Department of Energy

The US Department of Energy white paper titled "21 Steps to Improve Cyber Security of SCADA Networks" [40] outlines the essential actions to be taken to improve the protection of SCADA networks. The steps are divided into two categories:

1. Specific actions to improve implementation.
2. Actions to establish essential underlying management processes and policies.

A list of the recommended steps are outlined in Appendix 4.

8.5 NIST Recommendations for ICS/SCADA Security

As already introduced in section 2.2 the National Institute of Standards and Technology (NIST) have a very comprehensive publication "*Guide to Industrial Control Systems (ICS) Security"* which provides guidance for establishing secure ICS/SCADA systems [14]. The publication details an initial overview of ICS/SCADA systems, and examines the vulnerabilities of these systems. The document also outlines the steps required for the development and deployment of a security program to protect these systems. Finally the publication details the aspects of network architecture and at the Security Controls to protect these systems. Appendix 6 gives a list of the items covered in the last two sections.

The network architecture section of this publication recommends the segregation of control and corporate networks and the use of the DHS Defence in Depth model using firewalls, DMZ's and IDS's. For security controls it recommends the use of another NIST document entitled "Recommended Security Controls for Federal Information Systems and Organisations" [14]. This document was designed for use with traditional IT systems but can also be applied to ICS/SCADA systems. It recommends the use of Management Controls using policies and procedures, Operational Controls that are implemented by personnel as opposed to systems and technical controls such as hardware, software or firmware.

8.6 North American Electric Reliability Corporation Standards

The mission of the North American Electric Reliability Corporation's (NERC) is to ensure the reliability of the North American bulk power system. It has a mandate to establish and enforce reliability standards for the bulk-power system. NERC develops and enforces reliability standards as detailed in the Critical Infrastructure Protection (CIP). There are eight specific standards each of which is mandatory for electric power and utility companies listed below:

1. CIP-002-1 Cyber Asset Identification
2. CIP-003-1 Security Management Controls
3. CIP-004-1 Personnel & Training
4. CIP-005-1 Electronic Security Perimeters
5. CIP-006-1 Physical Security
6. CIP-007-1 System Security Management
7. CIP-008-1 Incident Reporting and Response
8. CIP-009-1 Recovery Plan for Critical Cyber Assets

8.7 Manufacturers Recommendations

Each of the manufacturers of ICS/SCADA equipment has guidelines on the implementation and operation of these systems. One such manufacturer that I have examined is Siemens. Siemens have a dedicated section on their website which aims to keep users of their equipment up-to-date with the latest security issues and fixes [41]. Siemens recommend that when planning and implementing efficient industrial security solutions, the following five points must be taken into account:

1. Implementation of an appropriate system-wide security management system with regard to the technology and the engineering and production processes.
2. The network interfaces to office IT and the Internet/Intranet are subject to clear regulations and are monitored accordingly.
3. Protection of PC-based systems (HMI, engineering and PC-based controllers) by means of antivirus software, white-lists, and integral security mechanisms.
4. Protection at the control level with automatically active security functions integrated into the automation and drive components with security functions that have to be activated by the programmer with passwords.
5. Monitoring of all communication with systems for the purpose of detecting intruders, and intelligent subdivision of the network with the help of firewall.

In a recently published document, [42] Siemens outline that the recommendations for technical and organisational measures for secure operation of plant and machinery should be achieved through:

1. Security organisation and guidelines
2. Plant security
3. Plant IT security
    i. Network segmentation
    ii. System hardening
    iii. Patch management
4. Access protection

8.8 Research into Vulnerabilities

Since the identification of the Stuxnet virus there has been an increase in research into the vulnerabilities of ICS/SCADA systems to cyber-attacks. There are many facets to research and I will explore three of them in this section.

Researchers in Iowa State University (ISU) are developing cyber-attacks in order to learn how to fix attacks on ICS/SCADA systems in the future. They are using a SCADA test bed to simulate and study cyber-attacks and to study methods of protecting these systems [43].

A Russian security firm, called Gleg [44] is developing an exploit pack called Agora SCADA +, which they hope will collect all publicly available SCADA vulnerabilities into one exploit pack. Organisations can buy this exploitation pack to test the configurations and vulnerabilities of their SCADA systems. The exploitation pack currently contains 22 modules that incorporate exploits for eleven zero-day vulnerabilities.

On the 21st of March, 2011 an Italian cyber-security researcher called Luigi Auriemma analysed the code of ICS/SCADA software. He managed to identify 34 SCADA vulnerabilities in server-side SCADA server software such as Siemens Tecnomatix FactoryLink, Iconics GENESIS32 and GENESIS64, 7-Technologies IGSS, and DATAC RealWin. These vulnerabilities are posted on his website. [45]

8.9 Findings

From my brief overview of the standards and procedures for ICS/SCADA security, it is evident that many of these are similar in content. With a myriad of potential options it is easy to see why there is confusion within organisations as to which one to implement. These guidelines, best practice standards and procedures need to by studied by organisations looking to protect their ISC/SCADA systems and apply those relevant to their specific requirements and circumstances.

All of the options examined in this chapter recommend separating the ICS/SCADA network from the corporate network, as much as possible due to the complex nature of the network traffic.

The DHS recommendation for a defence-in-depth model to protect ICS/SCADA systems is very detailed and mandates that the segregated corporate and ICS/SCADA networks should only be joined with minimal or a single connection. This connection will need to be provided by a firewall and a DMZ. Servers containing ICS/SCADA information that needs to be accessed from the corporate network should be put on the DMZ network segment. Only systems in the DMZ should be accessible from the

corporate network. For external connections to ICS/SCADA networks only minimum access should be permitted through the firewall, including opening only the ports required for specific communication. External access should also use to concept of the DMZ to protect these systems. It also mandated the use of IDS's to detect suspicious network activity.

The NIST Special Publication 800-82 [14] publication is a very comprehensive document that looks at the security of ICS/SCADA systems. It makes many of its oven recommendations as well as recommending the use of others like the DHS publication. Appendix C contains a brief outline of the current activities addressing ICS/SCADA cyber security. Appendix 3 in this document gives a list of these activities.

CHAPTER 9
Implementing Protocol Security with IPSec Tunnel

As detailed previously it was stated that ICS/SCADA packets are presently transmitted over communications networks in easily readable form. With little difficulty hackers can gain access to this traffic, and therefore gain useful information about target ICS/SCADA systems. This information will not directly inform attackers on what this instrumentation and control traffic is performing. Industrial plant wiring diagrams and other architecture documentation will be needed to gain a full understanding of what is contained within these packets. The potential for attackers to get access to infrastructure information is quite high through various hacking techniques if systems are not hardened as was the case with the Stuxnet Virus. With these two pieces of information an attacker can intercept, interrupt, modify, or fabricate packets to carry out attacks such as unauthorised command execution, SCADA-DOS, Man-in-the-Middle, replay, and compromised master attacks.

9.1 Development of test bed

For the process of capturing and examining ICS/SCADA packets, an ICS/SCADA "test-bed" consisting of several devices was configured. This "test-bed" consisted of a Siemen's S7-1200 PLC programmed and configured with an IP address. Programming was achieved using "ladder logic" and a very simple PLC program was written using the Siemens' STEP7 programming application. Figure 9.1 shows this ladder logic program.

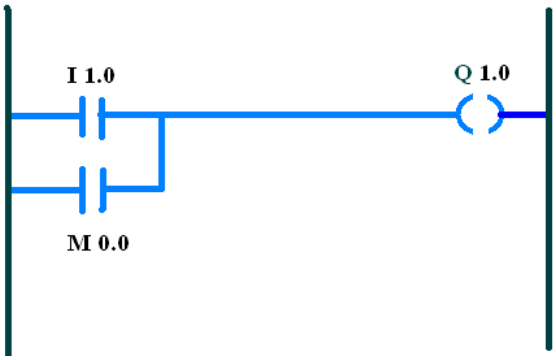


Figure 9.1 PLC Ladder program

The program was set up to examine a physical input labelled I1.0 and also a memory register within the PLC denoted M0.0. If either of these two locations are high, "logic 1" the program within the PLC would turn on physical output called Q1.0. This simple program can be thought of as a Logic "OR" operation.

The changes in the input and output conditions would then to be transmitted to a SCADA server. The SCADA server utilised was the Kepware Technologies KEPServer EX application. The server was configured with IP address of the PLC and configured with "tags" that would monitor the inputs and outputs to be monitored, in this case M0.0 and Q1.0. The hardware configuration can be observed in in figure 9.1.1 below.

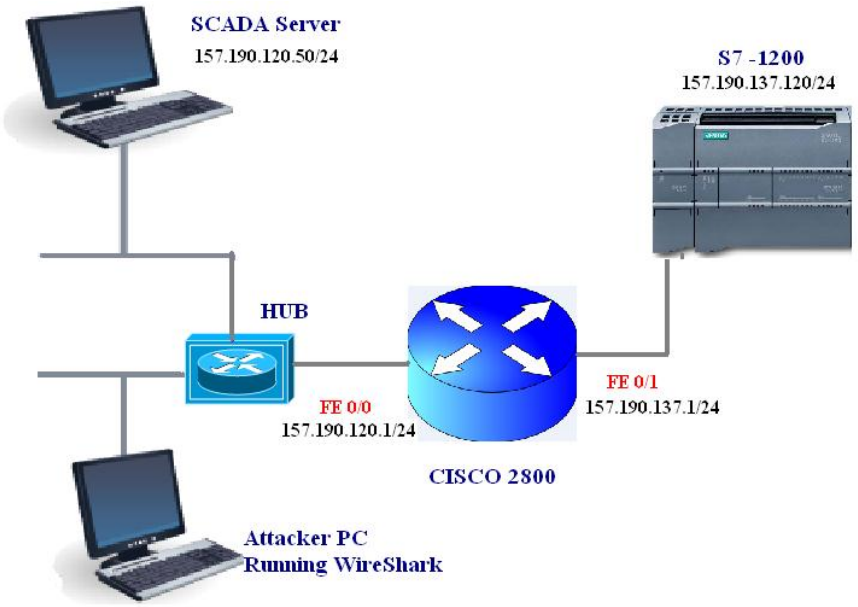
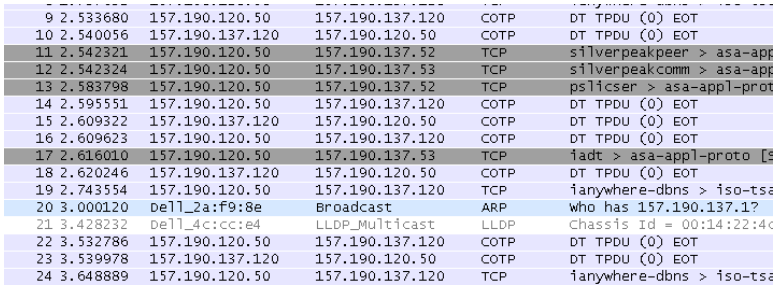


Figure 9.1.1 Implementation Architecture

Using the KEPServer application it was then possible to monitor the status of the output Q1.0 when the input I1.0 was manually toggled form a logic "hi" to "low". KEPServer also allowed for the facility to control the output Q1.0 by writing to memory location M0.0 which the PLC program was programmed to monitor and toggle Q1.0 based on the information in this register. The devices in this configuration use the PROFINET communications protocol as was covered in Section 1.5.3.

9.2 Capturing ICS/SCADA packets

A PC denoted as "Attacker PC" was utilised which was connected between the hub and the router as shown in Figure 9.1.1. This hub re-broadcasted all traffic being sent between the router and the SCADA server and this allowed a potential point for data capture. This hub was used for ease of implementation however a switch with a span port or alternately use of a technique called Content Addressable Memory (CAM) Table Overflow could have been used. Installed on the "Attackers PC" was "Wireshark" which would be used to capture the packets sent between the router and the SCADA server. Once captured this traffic was then analysed to identify ICS/SCADA protocol packets of interest which are shown in figure 9.2.1.



9	2.533680	157.190.120.50	157.190.137.120	COTP	DT TPDU (0) EOT
10	2.540056	157.190.137.120	157.190.120.50	COTP	DT TPDU (0) EOT
11	2.542321	157.190.120.50	157.190.137.52	TCP	silverpeakpeer > asa-app
12	2.542324	157.190.120.50	157.190.137.53	TCP	silverpeakcomm > asa-app
13	2.583798	157.190.120.50	157.190.137.52	TCP	pslicser > asa-appl-prot
14	2.595551	157.190.120.50	157.190.137.120	COTP	DT TPDU (0) EOT
15	2.609322	157.190.137.120	157.190.120.50	COTP	DT TPDU (0) EOT
16	2.609623	157.190.120.50	157.190.137.120	COTP	DT TPDU (0) EOT
17	2.616010	157.190.120.50	157.190.137.53	TCP	iadt > asa-appl-proto [s
18	2.620246	157.190.137.120	157.190.120.50	COTP	DT TPDU (0) EOT
19	2.743554	157.190.120.50	157.190.137.120	TCP	ianywhere-dbns > iso-tsa
20	3.000120	Dell_2a:f9:8e	Broadcast	ARP	Who has 157.190.137.1?
21	3.428232	Dell_4c:cc:e4	LLDP_Multicast	LLDP	Chassis Id = 00:14:22:4c
22	3.532786	157.190.120.50	157.190.137.120	COTP	DT TPDU (0) EOT
23	3.539978	157.190.137.120	157.190.120.50	COTP	DT TPDU (0) EOT
24	3.648889	157.190.120.50	157.190.137.120	TCP	ianywhere-dbns > iso-tsa

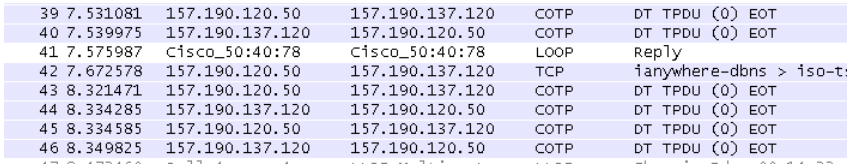
Figure 9.2.1 Initial Wireshark capture.

From the traffic data it was noted that Wireshark recognised the SCADA systems packets as using the Connection Orientated Transport Protocol (COTP). When these packets were examined in more detail it was observed that the destination port of this traffic was port 102.

At the very start of this capture there is a pair of packets transferred between the PLC and the SCADA Server, which can be seen at capture 9 & 10. These are “keep- alives” which are used in order to maintain synchronisation between these devices. The time between these “keep-alives” can be set on the SCADA server. The SCADA server initiates these “keep-alive” and waits for a reply from the PLC.

Captures 14, 15, 16 and 18 are a manual result of an “Asynchronous Write” operation to the PLC from the server. The “Asynchronous Write” was initiated by the author, which placed logic”1” into memory location M0.0 of the PLC memory and turned on the output Q1.0 of the PLC. These 4 packets show

1. The information being sent to the PLC.
2. The PLC replying to Server saying that it got the information.
3. Server requesting an update from the PLC.
4. PLC sending a status update.



39	7.531081	157.190.120.50	157.190.137.120	COTP	DT TPDU (0) EOT
40	7.539975	157.190.137.120	157.190.120.50	COTP	DT TPDU (0) EOT
41	7.575987	Cisco_50:40:78	Cisco_50:40:78	LOOP	Reply
42	7.672578	157.190.120.50	157.190.137.120	TCP	ianywhere-dbns > iso-t
43	8.321471	157.190.120.50	157.190.137.120	COTP	DT TPDU (0) EOT
44	8.334285	157.190.137.120	157.190.120.50	COTP	DT TPDU (0) EOT
45	8.334585	157.190.120.50	157.190.137.120	COTP	DT TPDU (0) EOT
46	8.349825	157.190.137.120	157.190.120.50	COTP	DT TPDU (0) EOT

Figure 9.2.2 Second Wireshark capture

The next step was the reverse of the previous procedure, in that logic “0” was written to memory location M0.0 of the PLC. This had the effect of turning off output Q1.0 of the PLC. This action created the same sequence of events as described previously and are shown in capture 43, 44 ,45 and 46 of figure 9.2.2.

Examining packets 14 and 43, Wireshark indicated that the format of these packets was eth:ip:tcp:tpkt:cotp:data. These protocols, used within these packets could be examined in detail using Wireshark. The protocol information was not of relevance but information within the data fields of these two packet captures was. Figure 9.2.3 below shows some the Wireshark information about capture 43, the data in the packets can be seen at the bottom.

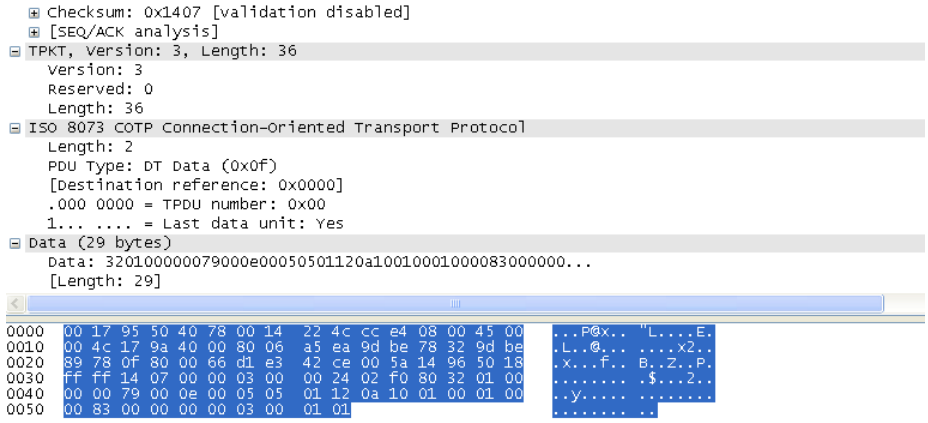


Figure 9.2.3 Data in Packet Capture 43

By comparing the data of four successive writes to memory location M0.0 in the PLC using the technique outlined above, then data changes could be analysed. This analysis was simply performed and the results are shown in figure 9.2.3 below. Highlighted in red are the components of the packets which are changing. The first of these is a sequence number that is used to sequence the packet and the second is the actual bit being written into the memory location M0.0.

Data to M0.0	Parced of data being sent to PLC
1	32:01:00:00:0e:75:00:0e:00:05:05:01:12:0a:10:01:00:01:00:00:83:00:00:00:00:03:00:01:01
0	32:01:00:00:00:79:00:0e:00:05:05:01:12:0a:10:01:00:01:00:00:83:00:00:00:00:03:00:01:00
1	32:01:00:00:00:a9:00:0e:00:05:05:01:12:0a:10:01:00:01:00:00:83:00:00:00:00:03:00:01:01
0	32:01:00:00:00:af:00:0e:00:05:05:01:12:0a:10:01:00:01:00:00:83:00:00:00:00:03:00:01:00

Figure 9.2.4 Comparison of packets that write data to M0.0 of the PLC

Using the information on these captured packets an attacker an easily use this to launch attacks on the protocol level of ICS/SCADA systems as outlined in Chapter 7.

9.3 Implementing protection against protocol vulnerabilities.

A relatively straight forward method that organisations can used to implement a layer of security to mitigate against protocol vulnerabilities is by use of encryption tunnels. This can be implemented easily, often with existing equipment that many organisations commonly have. IPSec tunnels are used to secure protocols, where IP traffic travels along insecure networks, see section 7.2.2. The router, in the test-bed ICS/SCADA network, was configured to act as a VPN concentrator and installed on SCADA server

was VPN Client application software. The router configuration can be seen Appendix 7. Once the tunnel was initialised by launching the VPN client software and connecting to the routers VPN service, all the ICS/SCADA protocol traffic travelled inside this encrypted tunnel to the router. This tunnel could span across several network devices but in this test-bed it is connecting directly to the FE 0/0 interface on the router.

Capturing and examination of the traffic between the router and the SCADA server using Wireshark was still possible but identification on any ICS/SCADA protocol traffic was not. This is because this traffic is now encrypted inside the configured IPsec tunnel. Figure 9.3.1 shows the Wireshark capture of the IPSec traffic which has the ICS/SCADA protocol traffic embedded. There is no indication from this capture that the SCADA server is talking to the PLC or visa-versa. This layer of security mitigates the treat of attacks at the protocol layer of ICS/SCADA systems. It provides a tunnel in which the ICS/SCADA traffic can travel along un-trusted paths securely without the threat of reconnaissance that may lead to an attack. It is a measure that organisations can easily implement with little costs involved.

7	0.385574	157.190.120.1	157.190.120.50	ESP	ESP (SPI=0xb8b87d0c)
8	0.502947	157.190.120.50	157.190.120.1	ESP	ESP (SPI=0xbcda3578)
9	1.364486	157.190.120.50	157.190.120.1	ESP	ESP (SPI=0xbcda3578)
10	1.370663	157.190.137.120	157.190.120.50	TPKT	Continuation
11	1.375926	157.190.120.1	157.190.120.50	ESP	ESP (SPI=0xb8b87d0c)
12	1.385745	157.190.120.1	157.190.120.50	ESP	ESP (SPI=0xb8b87d0c)
13	1.388341	Dell_2a:f9:8e	Broadcast	ARP	Who has 157.190.137.1
14	1.508883	157.190.120.50	157.190.120.1	ESP	ESP (SPI=0xbcda3578)
15	2.363562	157.190.120.50	157.190.120.1	ESP	ESP (SPI=0xbcda3578)
16	2.370969	157.190.137.120	157.190.120.50	TCP	[TCP Keep-Alive] iso-
17	2.380563	157.190.120.1	157.190.120.50	ESP	ESP (SPI=0xb8b87d0c)
18	2.387446	157.190.120.1	157.190.120.50	ESP	ESP (SPI=0xb8b87d0c)
19	2.461487	Cisco_50:40:78	Cisco_50:40:78	LOOP	Reply
20	2.514815	157.190.120.50	157.190.120.1	ESP	ESP (SPI=0xbcda3578)
21	2.888407	Dell_2a:f9:8e	Broadcast	ARP	Who has 157.190.137.1
22	3.118373	157.190.120.50	157.190.120.1	ESP	ESP (SPI=0xbcda3578)
23	3.118393	157.190.120.50	157.190.120.1	ESP	ESP (SPI=0xbcda3578)
24	3.152901	157.190.120.50	157.190.120.1	ESP	ESP (SPI=0xbcda3578)
25	3.185171	157.190.120.50	157.190.120.1	ESP	ESP (SPI=0xbcda3578)
26	3.362625	157.190.120.50	157.190.120.1	ESP	ESP (SPI=0xbcda3578)
27	3.364444	Dell_4c:cc:e4	LLDP_Multicast	LLDP	Chassis Id = 00:14:22
28	3.370633	157.190.137.120	157.190.120.50	TCP	[TCP Keep-Alive] iso-
29	3.375908	157.190.120.1	157.190.120.50	ESP	ESP (SPI=0xb8b87d0c)
30	3.385705	157.190.120.1	157.190.120.50	ESP	ESP (SPI=0xb8b87d0c)
31	3.520722	157.190.120.50	157.190.120.1	ESP	ESP (SPI=0xbcda3578)

Figure 9.3.1 Wireshark Capture of IPSec tunnel traffic.

The addition of security into the network may have had an effect on response times within the SCADA network. Analysis was performed on the latency that may be experienced when using such an IPsec tunnel to protect the ICS/SCADA traffic. This was achieved by measuring the round trip delay of an ICMP packet being sent from the SCADA server to the PLC and back. Firstly 20 ICMP messages were transmitted across the non-VPN communications path, and subsequently 50 ICMP messages were transmitted with the VPN in place. The “PING” command was utilised to perform this test and the results were piped into a text file using the command

ping 157.190.137.120 –n 50 >> c:\capture.txt

Figure 9.3.2 is a graph of the round-trip delay experienced by ICMP packets when carrying out these tests. It shows that there is slight additional latency involved with the use of the VPN tunnel but it does not introduce significant performance degradation.

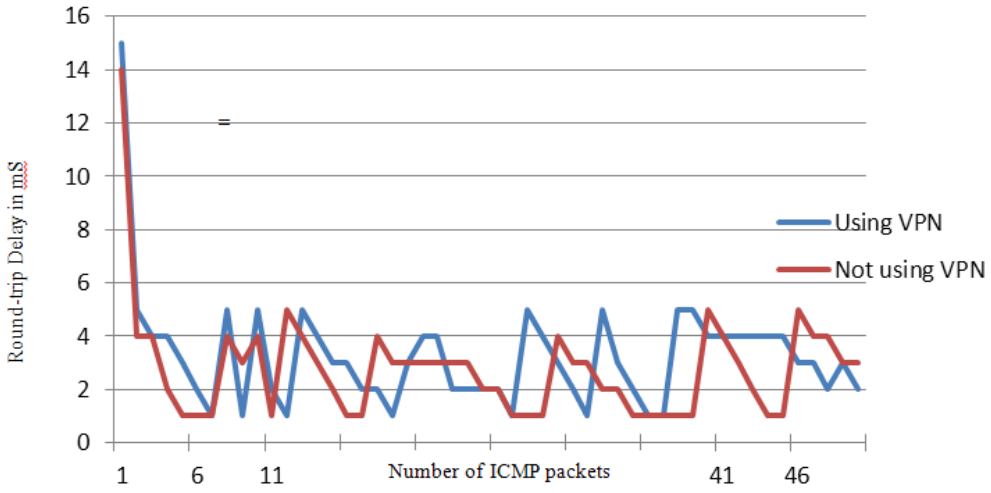


Figure 9.3.2 Analysis of Latency with use of VPN

9.4 Findings

The use of packet capturing applications like Wireshark can be used to intercept and analyse ISC/SCADA protocol traffic. The transportation headers on captured packets can easily be stripped off to get access to the embedded data. This data can be examined by attackers to attain information that can be used to launch an attack on ICS/SCADA systems. The use of IPSec tunnels can provide a layer of security to these unencrypted ICS/SCADA packets. As demonstrated this technique is very easy to implement and often only minimal costs are involved depending on the capability of network equipment. Low cost mini routers may need to be used in cases there existing network equipment does not have the capability required. Organisations can easily and cheaply implement IPSec VPN's to mitigate the risks associated with protocol vulnerability.

This method can be used to secure ICS/SCAD communications between remote sites that utilise the insecure internet. It can also be used for the purposes of remote access for the maintenance of the ICS/SCADA systems. Within the ICS/SCADA systems themselves this technique will provide a very good layer of security to the ICS/SCADA protocols, to overcome the vulnerabilities outlines in Chapter 7. This technique can be utilised in situations there the security of organisations ICS/SCADA communications traffic is critical.

CHAPTER 10

Recommendations

In previous chapters the ICS/SCADA vulnerabilities were outlined as being: Policy and Procedure, Platform, Network and Communication Protocol based. Methodologies that organisations can use to secure against many of these vulnerabilities and an examination of popular security standards and procedures in securing these systems were outlined. Based on this research it is now possible to make recommendations to organisations that use ICS/SCADA systems within Ireland on the steps that they need to take to secure their systems against the vulnerabilities outlines. These recommendations are in the format of a step-by-step approach that organisation should undertake when implementing a secure ICS/SCADA environment. These steps provide a high-order road map that will provide organisations with the basic concepts that it will need to implement a successful ICS/SCADA security mechanism. The actual specific implementation will depend on each individual organisation requirements.

The recommendations in this report concentrate on the threats of cyber-attack and the ICS/SCADA vulnerabilities that enable such attacks. The target organisation for these reccomendations is a Medium Sized Enterprise (MSE). Enterprise Ireland defines an MSE as "*an enterprise that has between 50 employees and 249 employees and has either an annual turnover not exceeding €50m or an annual Balance Sheet total not exceeding €43m*" [46]. These recommendations can also be applied to larger and smaller organisations that operate this equipment.

A typical target MSE operates in the chemical processing industry and utilises an ISC/SCADA system to monitor and control a batch process in one site. The MSE employs a centralised command and control centre which acquires instrumentation data from in the region of 200 sensors and controls in the region of 200 actuator and PLC devices. This company currently has a complex mix of the older serial based ICS/SCADA equipment which has been ported to the newer TCP/IP based networks through the use of bridging devices. The company is gradually migrating to using the newer Ethernet technology being introduced by equipment manufacturers, as new production processes are being designed and implemented. Managers require "read" access to the data generated by SCADA system and also "write" access to batch input data into the batch process.

10.1 Five Steps

The five steps recommended here are a variation on the four key processes for Security Management Function (SMF) developed by Microsoft as part of its Microsoft Operations Framework (MOF) [48]. These steps will deliver the overarching ICS/SCADA security management program for MSE organisation that utilise these systems in Ireland.

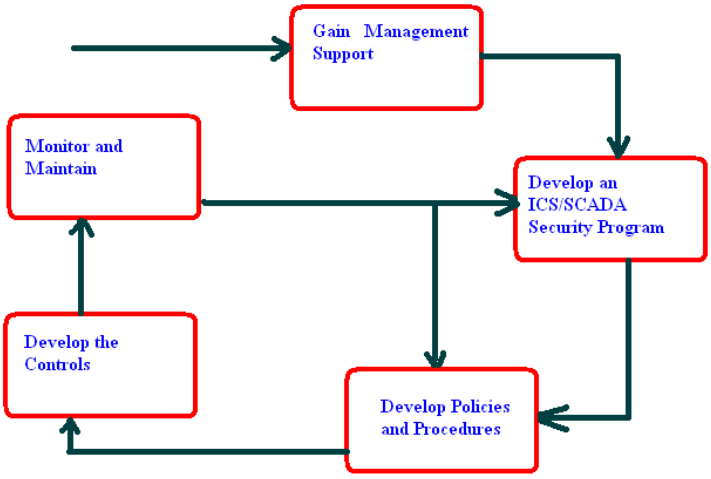


Figure 10.1 Steps to secure ICS/SCADA systems

Step 1-Gain upper management support.

Management buy-in is required as resources in the form of financial, personnel and training will be required to successfully implement this program. As with any investment program, it will need to indicate to management the potential savings that could be made in reducing lost production and unscheduled downtime. Management also need to be informed of the potential consequences of not implementing such a security program. Often it is not sufficient to highlight benefits such as protection against adversarial threats, accidental threats and known natural threats. Benefits such as those listed below are often of more interest to management rather than technical benefits

1. Increased reliability and security of ICS/SCADA systems.
2. Increased Profits.
3. Reduced Costs.
4. Help compliance with legislation and regulation requirements.
5. Help improved Management/Production processes.
6. Help to maintain competitive advantage.

Step 2-Develop an ICS/SCADA security program.

This is the planning phase for an ICS/SCADA security program. This phase should be similar to the planning of the corporate IT security program. The "defence-in-depth" model has proven itself to be an effective method to protect IT systems so I recommend that this should be used when protecting the ICS/SCADA systems. The DHS also recommend the use of this model when it comes to protecting these systems.

Organisations often use IT security standards such as ISO27000 to implement Information Security Management System (ISMS) [49]. For the planning phase of ICS/SCADA security I recommend that organisations should use the steps below which are modifications to the ISO 27000 planning phase for ISMS.

1. Appoint a steering Committee.
2. Identify ICS/SCADA Security Program Scope and Boundaries.
3. Asset Identification and Classification.
4. Identify threats to Assets ISC/SCADA system and assets.
5. Access the exposure or risk of these threats.
6. Monitor the development of the ICS/SCADA Security Program.
7. Develop a project time-line.

To properly address the security of ICS/SCADA systems a cross-functional steering committee is required. This ICS/SCADA security committee should consist of a member of the organisation's IT staff, control engineer, control system operator, network and system security expert, a member of the management staff, and a member of the physical security department. This committee will need to share their varied knowledge and experience and consult with ICS/SCADA vendors, to evaluate and mitigate risk to the ICS/SCADA systems. This committee may not initially have the knowledge and expertise to address the recommendations presented in this document. Training or help form external organisations may be required.

Step 3-Develop Policies and Procedures that will secure these systems.

As was outlined in Chapter 4 the lack of Security policies is a vulnerability in itself when it comes to the security of any ISC/SCADA systems. Policies are often the cheapest security measure to add security to ICS/SCADA systems, but the most difficult to implement. For these reasons I recommend that the process of developing policies

and procedures should be treated as a stand-alone step in the five step set of recommendations being made in this document.

Once the planning stage of the ICS/SCADA is complete (i.e. Step 2) the identified risks to the ISC/SCADA system will need to be addressed. The first step to addressing risk is to put in place a policy that will mitigate this risk. Once polices are in place users of the ICS/SCADA system need to be informed about the policy. The IT and HR Groups will also need to develop training and awareness programs to inform users of their responsibilities and the purposes of the new policies and procedures employed so that they are implemented effectively with constructive criticism encouraged. Users also need to be informed about the implications of non-compliance with the policy. The recommend steps for policy development process are:

1. Identify and access appropriate controls that mitigate the risks.
2. Setup policies and procedures to mitigate and control risk.
3. Assign resources and develop training and awareness program.
4. Communicate the policies and implications of non-compliance to users.

Just some of the policies that an organisation may look to implement to secure ICS/SCADA systems are:

1. ICS/SCADA Security Patching Policy
2. ICS/SCADA Antivirus Update Policy
3. ICS/SCADA Remote Access Policy
4. ISC/SCADA Password Policy

The SANS Institute provides a verity of sample policies can be used by organisations freely as a guide to the creation of policies.

Step 4- Develop the technical controls that need to be put in place to implement the policies that were developed to protect the ICS/SCADA system.

Active controls are needed in the development of policies and implementation of safeguards. Without these controls there is no method of ensuring that the new policies, procedures and methodologies are adhered to. Recommendations on the use of Technical controls are made later in this chapter.

Step 5- Monitor, Audit and Maintain the ICS/SCADA security program.

As with any implementation of new procedures and policies there is a requirement to employ effective checks and monitoring systems to insure that the new systems are

performing in the manner expected. Outputs from this stage should be fed back into the Security Program and Policy/Procedure Phases to make modifications to these which could lead to new controls to be developed. This step will involve the use of systems like IDS, penetration testing tools, vulnerability assessment tools and the examination of system activity logs. It will also involve the process of problem and incident identification which will also feed into the Security Program and Policy/Procedure steps so that controls can be but in place to counteract such problems or incidents.

10.2 Technical Controls

To implement and enforce the policies and procedures developed in an ICS/SCADA security program, technical controls need to be developed and deployed. This is a challenging aspect for an organisation ICS/SCADA security program as it will need to identify and implement the controls needed to protect systems. The remainder of this chapter provides recommendations that will focus on the technical controls that organisations will need to implement to secure the ICS/SCADA systems.

The recommendations made here focus on hardening the perimeter devices such as routers, firewalls, ACL etc. This technique will help overcome SCADA protocol and device vulnerabilities. The control of access by users to these systems is addressed as well as focus on controls to patch security holes and implement anti- virus updates.

10.2.1 Segregate the Networks

The segregation of the ICS/SCADA networks and the corporate business networks is an important security control. These networks have vastly different requirements that do not map easily together on a unified network. Once these networks are segregated, the security aspects of both can be treated separately and appropriate controls put in place. This segregation can be achieved through the use of VLANs which provides an OSI Layer 2 layer of security to these systems.

10.2.2 Segregate the ICS/SCADA Network

Within the ICS/SCADA network further segregation is required to as outlined with the examination of the ISA-95 recommendations in section 8.1. This network should be broken into Operations Information Network, Automation Network and Discrete & Process Communication Network. This step will lead to a better understanding of the technology associated with ISC/SCADA systems and allow for IT and automation specialists to have a clear vision on what's involved in the operation and security of these systems.

10.2.3 Use Firewalls and ACLs

The use of firewalls and ACLs to control the flow of information between the segregated ICS/SCADA network and the corporate network is required. This recommendation comes from the DHS publication that recommends this to provide the capability to add tighter and complex rules for communication between these networks. This will provide for protection from such attacks as communication protocol vulnerabilities that originate from outside these networks. Section 8.3 outlines the DHS recommendations in more detail.NIST SP 800-41 "*Guidelines on Firewalls and Firewall Policy*" will help provide guidance in this area.

10.2.4 Use DMZs

It is recommended that services required by the business processes from outside the ICS/SCADA network should be provided by the use of a DMZ. This adds a layer of security to the ICS/SCADA network as potential attackers only have access to equipment in the DMZ, rather than any other part of the ICS/SCADA network.

10.2.5 Secure Remote Access

The aspects of remote access to the ICS/SCADA systems need to be secured with the use of technologies such as VPN that use IPSec or other encryption techniques. Section 9.3 outlines in detail how this may be achieve very cheaply and easily. Securing of remote access should also be done using the DMZ security techniques covered above.

10.2.6 Use Identification and Authentication

The identification of network users, hosts, applications, services, and resources needs a combination of identification factors or credentials. This will be the basis for which access is given to the ICS/SCADA system. Currently within the IT security sector there are moves towards two factor authentications which vastly improves the security of systems. The ICS/SCADA system need an Identification and Authentication control process through the use of appropriate authentication mechanisms such as passwords, challenge/response, physical token or biometric authentication mechanisms. This will also prove useful for access control and accountability.

10.2.7 Communications Protection

As covered in Chapter 7 ICS/SCADA communications protocols transmit data across the transport medium in plane text which lets them open to all form of protocol based attacks. As Chapter 9 shows VPNs can provide an easy to implement techniques that organisations utilise to encrypt this traffic with little cost or performance overhead. With the recommendations above of securing the parameter with firewalls and ACL to stop the protocol entering or leaving the ICS/SCADA system there may not be requirements for such a control. However if there is a need to add this additional layer of security this will prove to be very effective way to circumvent the vulnerabilities associated with the ICS/SCADA communications protocols.

10.2.8 Patch Management

Mechanism need to be put in place to insure the testing and application of security patches that are released by equipment manufactures and application software providers. Where support from providers is no longer available, this equipment should be retired and invest in more secure equipment should be made. This will be a major step for many organisations that need to get away from the "if it's not broke don't fix it" attitude when it comes to ICS/SCADA systems.

The NIST SP 800-40 *"Creating a Patch and Vulnerability Management Program"* will provide guidance to organisations then setting up this security control process. This will include mechanism to keep up to date with the latest ICS/SCADA system vulnerabilities by examining manufactures websites and other organisations like ICS-CERT as outlined in section 8.2.

10.2.9 Anti-Virus Management

Mechanisms need to be up in place for the testing and application of anti-virus definitions on an on-going basis. Methods to distribute anti-virus definitions locally within the ICS/SCADA system should be used rather than having to put holes in firewalls to access these definitions from vendors online databases should be employed.

10.2.10 Security Control Tools

Implementation of controls such as intrusion detection software, activity log generation, file integrity checking, penetration testing tools, vulnerability assessment tools will help to protect ICS/SCADA systems. These tools can also be used in the Monitoring, Auditing and Maintaining of the ICS/SCADA security program.

10.2.11 Configuration Control

Controls need to be put in place to document and control the configurations of ICS/SCADA systems. This should be done through a Change Management process which is a process where changes to these systems need to be requested, assessed and authorised before they can be applied. This will help in the event of malicious attacks on these systems and reduce potential losses that can result from such a scenario by restoring system after an incident. Change Management Systems Applications can also be used to monitor and detect unauthorised configuration changes on ICS/SCADA systems. This process of configuration control will also provide in input the Regulatory check such as those done by organisations like the FDA.

10.2.12 Physical Security Control

Mechanisms need to be put in place to restrict physical access to ISC/SCADA devices. This needs to be done for two reasons. The first is the threat of physical theft, damage or disconnection of this equipment from the communications medium. The second is the threat of malicious software being loaded intentionally or accidently through the use of portable media storage devices such as CD or USB keys. Physical security of these devices should be addressed and they should be only deployed in secure locations. Controls to disable portable media devices should also be used.

CHAPTER 11
Conclusions

The threats to ICS/SCADA systems can be malicious, accidental or environmental. Currently the highest threat is from malicious cyber-attack that can lead to availability, integrity and confidentiality of these systems being affected. The vulnerabilities associated with these threats have been grouped into 4 categories.

1. Policy and Procedure Vulnerabilities
2. Platform Vulnerabilities
3. Network Vulnerabilities
4. Communication Protocol Vulnerabilities

From my research I have found that many organisations that utilise ICS/SCADA systems are unaware of these vulnerabilities. The danger for these vulnerabilities to cause serious damage to equipment, personnel and the environment are huge. They can even damage the reputation and cause financial loss to organisations.

It is my belief that there is a lack of knowledge within organisations of the potential threats that exist to their ICS/SCADA systems. The reason for this lack of knowledge is down to the fact that ICS/SCADA systems are often viewed in an unusual category within organisations. The ICS/SCADA systems are often created and maintained by Automation Engineers, who limited knowledge of IT networks and security related issues, but yet use the common infrastructure built by IT Engineers. For this reason the security of ICS/SCADA systems is current lagging behind the security measures being applied to the Corporate IT environment, by 5-7 years.

I have examined many of the popular security standards and procedures used to secure these systems. Using this research as a basis I have developed a set of recommendations that a MSE in Ireland should use to secure their ICS/SCADA and traditional IT system. These recommendations took the form of a 5 step approach to implement an ICS/SCADA security program as well as recommendations on the Technical Controls that should be put in place as part of this security program. Many of these techniques have proved themselves to be very effective in protecting of the traditional IT infrastructures and will work just as well when applied correctly to ICS/SCADA systems.

The main focus of these recommendations is to secure the perimeter of both the ICS/SCADA system and the corporate IT infrastructure by using policies and other

controls. More advanced methods of securing ICS/SCADA communication protocol traffic through the use of VPNs was explored however these are generally not required unless there is a specific requirement within the ICS/SCADA system or when the security of remote access to these systems needs to be addressed.

This ICS/SCADA security program that I recommend will have a holistic view of the IT Corporate and Automation networks and will provide for the transfer of information between them in a structured manner and without compromising security in either.

Investment in an ICS/SCADA security program will have the added benefit of reducing system down time, improve system reliability, reducing operation costs, and therefore increase profit.

Bibliography

[1] Frank Dickman, Pipline & Gas Journal Issue November 2009, *" Hacking The Industrial SCADA Network"*. http://pgjonline.com/hacking-industrial-scada-network

[2] Keith Stouffer, National Institute of Standards & Technology (NIST) Proceedings of the ISA Expo, Chicago October 25 – 27, 2005"*NIST Industrial Control System Security Activities*" http://www.isd.mel.nist.gov/documents/stouffer/ISA_Expo.pdf

[3] Robert F. Dacey, Information Security Issues Oct 2003, *"Critical Infrastructure Protection, Challenges in Securing Control Systems"*. http://www.gao.gov/new.items/d04140t.pdf

[4] Samuel East, Jonathan Butts, Mauricio Papa and Sujeet Shenoi, Critical Infrastructure Protection III, IFIP Advances in Information and Communication Technology, Volume 311. ISBN 978-3-642-04797-8. Springer Berlin Heidelberg, pp. 67–81, 2009, "A TAXONOMY OF ATTACKS ON THE DNP3 PROTOCOL" http://scada-dnp3.googlecode.com/files/East2009.pdf

[5] http://www.dnp.org/

[6] Ken Curtis, DNP Users Group, Revision A, 20 March 2005, *"DNP3 Primer"*. http://www.dnp.org/About/DNP3%20Primer%20Rev%20A.pdf

[7] http://www.modbus.com/

[8] Modbus Organization, *"MODBUS MESSAGING ON TCP/IP IMPLEMENTATION GUIDE V1.0b"* October 24, 2006. http://www.modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf

[9] http://www.profibus.com/

[10] PI International, A PI Whitepaper, *"PROFINET AND IT V1.0"*, December 2008 http://www.profibus.com/nc/downloads/downloads/pi-white-paper-profinet-and-it/display/

[11] http://www.opcfoundation.org

[12] OPC Foundation, OPC Task Force, *"OPC Overview Version 1.0"*, October 27, 1998 http://www.fer.hr/_download/repository/OPC_Overview_1.00.pdf

[13] National Institute of Standards and Technology (NIST), FIPS PUB 199, *"Standards for Security Categorization of Federal Information and Information Systems"*, Feburary 2004, http://csrc.nist.gov/publications/fips/fips199/FIPS-PUB-199-final.pdf

[14] Recommendations of the National Institute of Standards and Technology, Special Publication 800-82, *"Guide to Industrial Control Systems (ICS) Security"*, June 2011 http://csrc.nist.gov/publications/nistpubs/800-82/SP800-82-final.pdf

[15] Amanullah, A. Kalam and A. Zayegh, IEEE 2005, *"Network Security Vulnerabilities in SCADA and EMS"* http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1546981

[16] Darek Kominek, Eric Byres *"Improving ICS Security with OPC and Defense in Depth"* April 2011, http://www.controlglobal.com/articles/2011/ICSSecurity1105.html

[17] http://www.27000.org/

[18] http://www.sans.org

[19] Dan Goodin, The Register *"Dozens of exploits released for popular SCADA programs -Giant bullseyes painted on industrial control software"*, March 2011 http://www.theregister.co.uk/2011/03/22/scada_exploits_released/

[20]"*SIEMENS SIMATIC S7-1200 PLC VULNERABILITIES"* 10th June 2011 http://www.us-cert.gov/control_systems/pdf/ICS-ALERT-11-161-01.pdf

[21] *"Security Vulnerabilities in Siemens SIMATIC S7-1200 CPU"* 5th July 2011 http://support.automation.siemens.com/dnl/jM/jMyMjM5OQAA_50428932_Akt/Siemens_Security_Advisory_SSA-625789.pdf

[22] European Network for Security of Control and Real Time Systems *"Taxonomy of Security Solutions for the SCADA Sector"*, Version 1.1, March 2010. http://www.cen.eu/cen/Sectors/Sectors/ISSS/Focus/Documents/D22.pdf

[23] Samuel East, Jonathan Butts, Mauricio Papa and Sujeet Shenoi, Critical Infrastructure Protection III, IFIP AICT 311,IFIP International Federation for Information Processing 2009 *"A Taxonomy of attacks on the DNP3 protocol"*. pp. 67–81, 2009 http://scada-dnp3.googlecode.com/files/East2009.pdf

[24] Mechanical Engineering Magazine December 2003 http://www.memagazine.org/backissues/membersonly/dec02/features/scadavs/scadavs.html

[25] P. Huitsing, R. Chandia, M. Papa and S. Shenoi, *"Attack taxonomies for the Modbus protocols"*, International Journal of Critical Infrastructure Protection, vol. 1, pp. 37–44, 2008. http://www.sciencedirect.com/science/article/pii/S187454820800005X

[26] J. Graham, S. Patel, Dept. of Computer Engineering & Science, University of Louisville, "Security Considerations in SCADA Communication Protocols", September 2004 http://louisville.edu/speed/research/centers-and-labs/isrl/tech_papers/ISRL-04-01.pdf/at_download/file

[27] I. Nai Fovino, A. Carcano, M. Masera A. Trombetta, IFIP International Federation for Information Processing 2009, Critical Infrastructure Protection III, IFIP AICT 311, *"Design and implementation of a secure Modbus protocol"* , pp. 83–96, 2009

[28] Nai Fovino, A. Carcano, M. Masera, Institute for the Protection and Security of the Citizen Joint Research Centre, EU Commission via E. Fermi 1, 21027 Ispra, Italy, 2009, *"A Secure and Survivable Architecture for SCADA Systems"*. http://ieeexplore.ieee.org/Xplore/login.jsp?url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F5211047%2F5211048%2F05211091.pdf%3Farnumber%3D5211091&authDecision=-203

[29] http://www.pnl.gov/

[30] Schweitzer Engineering Laboratories *"SEL-3045 Security Policy"* Version 0.14, http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1488.pdf

[31] M.D. Hadley and K.A. Huston, Pacific Northwest National Laboratory, US Department of Energy *"Secure SCADA Communication Protocol Performance Test Results"* August 2007 http://www.doe.gov/sites/prod/files/oeprod/DocumentsandMedia/SSCP_Test_Results.pdf

[32] http://www.isa.org/

[33] http://www.isa-95.com/

[34] http://www.us-cert.gov/

[35] http://www.us-cert.gov/control_systems/ics-cert/

[36] http://www.us-cert.gov/control_systems/ics-cert/archive.html

[37] http://www.us-cert.gov/control_systems/practices/Recommended_Practices.html

[38] Department of Homeland Security, *"Recommended Practice: Improving Industrial Control Systems Cybersecurity with Defense-In-Depth Strategies"*, October 2009 http://www.us-cert.gov/control_systems/practices/documents/Defense_in_Depth_Oct09.pdf

[39] Department of Homeland Security, *"Catalog of Control Systems Security: Recommendations for Standards Developers",* September 2009
http://www.us-cert.gov/control_systems/pdf/Catalog_of_Control_Systems_Security_Recommendations.pdf

[40] http://www.oe.netl.doe.gov/docs/prepare/21stepsbooklet.pdf

[41] http://www.siemens.com/industrialsecurity

[42] Siemens *"Operational Guidelines for Industrial Security",* Version 1.0, 15-7-2011
http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/Operational_Guidelines_Industrial_Security_v1.0_en_2011-07-15.pdf

[43] "*Team of ISU researchers combats cyber warfare*", 7th January 2011,
http://iowastatedaily.com/news/article_89d0b690-187d-11e0-b624-001cc4c03286.html

[44] http://gleg.net/

[45] http://aluigi.org/

[46] http://www.enterprise-ireland.com/en/About-Us/Our-Clients/SME-Definition.html

[47] William Shaw, Swantech,"*Cyber Security, Wireless Technologies & Overview of ISA SP99*"
http://www.bwisa.org/BWisaLunchMeeting.pptx

[48] http://technet.microsoft.com/en-us/library/cc506049.aspx

[49] http://www.27000.org/

[50] http://www.theatlantic.com/technology/archive/2010/11/the-stuxnet-worm-more-than-30-people-built-it/66156/

[51] http://www.kaspersky.com/news?id=207576183

[52] Blockbuster Worm Aimed for Infrastructure, But No Proof Iran Nukes Were Target By Kim Zetter
http://www.wired.com/threatlevel/2010/09/stuxnet/

[53] Symantec's "W32.Stuxnet Dossier Version 1.2 (November 2010)" by Nicolas Falliere, Liam O Murchu, and Eric Chien
http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf

[54] ESET securities "Stuxnet Under the Microscope" Revision 1.2 by Aleksandr Matrosov, Eugene Rodionov, David Harley, and Juraj Malcho http://www.eset.com/resources/white-papers/Stuxnet_Under_the_Microscope.pdf

[55] http://www.microsoft.com/technet/security/bulletin/ms10-073.mspx

Appendix 1: Threats to IT Infrastructure

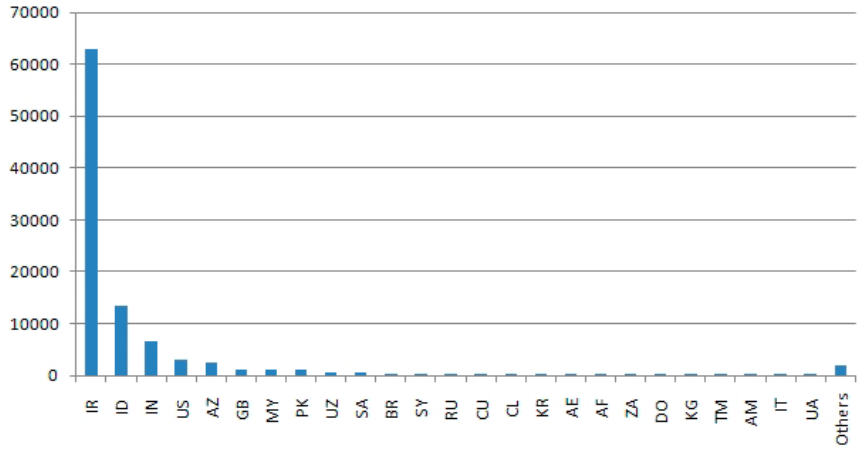
[3]

Threat	Description
Bot-network operators	Bot-network operators are hackers; however, instead of breaking into systems for the challenge or bragging rights, they take over multiple systems in order to coordinate attacks and to distribute phishinga schemes, spam, and malwareb attacks. The services of these networks are sometimes made available on underground markets (e.g., purchasing a denial-of-service attack, servers to relay spam or phishing attacks, etc.).
Criminal groups	Criminal groups seek to attack systems for monetary gain. Specifically, organized crime groups are using spam, phishing, and spyware/malware to commit identity theft and online fraud. International corporate spies and organized crime organizations also pose a threat to the United States through their ability to conduct industrial espionage and large-scale monetary theft and to hire or develop hacker talent.
Foreign intelligence services	Foreign intelligence services use cyber tools as part of their information-gathering and espionage activities. In addition, several nations are aggressively working to develop information warfare doctrine, programs, and capabilities. Such capabilities enable a single entity to have a significant and serious impact by disrupting the supply, communications, and economic infrastructures that support military power—impacts that could affect the daily lives of U.S. citizens across the country.
Hackers	Hackers break into networks for the thrill of the challenge or for bragging rights in the hacker community. While remote cracking once required a fair amount of skill or computer knowledge, hackers can now download attack scripts and protocols from the Internet and launch them against victim sites. Thus, while attack tools have become more sophisticated, they have also become easier to use. According to the Central Intelligence Agency, the large majority of hackers do not have the requisite expertise to threaten difficult targets such as critical U.S. networks. Nevertheless, the worldwide population of hackers poses a relatively high threat of an isolated or brief disruption causing serious damage.
Insiders	The disgruntled organization insider is a principal source of computer crime. Insiders may not need a great deal of knowledge about computer intrusions because their knowledge of a target system often allows them to gain unrestricted access to cause damage to the system or to steal system data. The insider threat also includes outsourcing vendors as well as employees who accidentally introduce malware into systems.
Phishers	Individuals, or small groups, that execute phishing schemes in an attempt to steal identities or information for monetary gain. Phishers may also use spam and spyware/malware to accomplish their objectives.
Spammers	Individuals or organizations that distribute unsolicited e-mail with hidden or false information in order to sell products, conduct phishing schemes, distribute spyware/malware, or attack organizations (i.e., denial ofservice).
Spyware/malware authors	Individuals or organizations with malicious intent carry out attacks against users by producing and distributing spyware and malware. Several destructive computer viruses and worms have harmed files and hard drives, including the Melissa Macro Virus, the Explore.Zip worm, the CIH (Chernobyl) Virus, Nimda, Code Red, Slammer, and Blaster
Terrorists	Terrorists seek to destroy, incapacitate, or exploit critical infrastructures in order to threaten national security, cause mass casualties, weaken the U.S. economy, and damage public morale and confidence. Terrorists may use phishing schemes or spyware/malware in order to generate funds or gather sensitive information.
Industrial spies	Industrial espionage seeks to acquire intellectual property and know-how by clandestine methods

Appendix 2: The Stuxnet Virus

In 2010 was a lot of international press coverage on a new internet worm called Stuxnet. This worm is very complex and has many new features not seen in any type of malicious attack prior to this. The worm targets SCADA systems and is the first ever such an attack on these types of systems. Stuxnet targets one manufacturer of these SCADA systems and initial indications pointed to the fact that the worm was targeting the infrastructure of one country.

The graph below comes from Symantec and shows the source of traffic to the Stuxnet worm command and control (C&C) servers. This shows that that 60% of all infected hosts worldwide, which connected to this C&C server, stem from Iran. With such a large number of instances of Stuxnet in Iran, media organisations speculated that this attack was targeted at one Iranian site, namely the Bushehr Nuclear facility. They also speculated that the USA and Israel as being the likely source of this attack as they have the motive, technology and vast resources needed to launch such a cyber-attack.



Graph 1:Infected Hosts Locations

Software security experts working in this field of malicious software identified evidence that would back up this speculation. They concluded that this attack is unlikely to have been carried out by an individual and they estimate that it would have taken 5 to 6 experts in the region of 6 months to write this attack. In a more recent report by Symantec's Brian Tillett he said that traces of more than 30 programmers have been found in the source code [50]. Within the code of Stuxnet there are many pointers and references to details in Jewish history which may be interpreted as clues to where the attack came from. These may also be put in there to throw investigators off the real source of the attack.

With all this speculation the prime suspects for this attack look to be the USA and Israel with the latter being the most likely. There has been no confirmation or denial of this speculation by either the Americans or Israeli authorities.

The only communication from the Iranian government about Stuxnet came on the 27th September 2010. They confirmed that the Stuxnet cyber-attack was targeted at their industrial facilities and the attack had so far failed to cause any serious damage. They also stated that it was unclear who was behind the attack.

It should also be noted that Stuxnet did not just infect devices in Iran. Stuxnet spread to infect systems in Indonesia, India, the USA and many more.

A Russian security company Kaspersky Labs stated that “Stuxnet was a working and fearsome prototype of a cyber-weapon that will lead to the creation of a new arms race in the world" [51]. It also stated that Stuxnet had been in existence for some time before it was discovered, so it is likely that the attackers has completed their initial attack by the time the worm itself was identified.

Beyond all this hype the fact that the Stuxnet worm ICS/SCADA systems, associated with one manufacturer Siemens is very serious. This is done with a large selection of tools and techniques which can be summarised.

1. Self-replication through removable devices
2. Complex process of injection and hooking of victims.
3. Spreads in LAN’s through Windows Print Spooler Vulnerability
4. Vulnerability in Server Service for Remote Code Execution (MS08-067)
5. Spreads through Server Message Block (SMB).
6. Copies and executes itself through network shares.
7. Copies and executes itself to computers running WinCC database server.
8. Copies itself onto Step 7 projects.
9. Utilised four zero day exploits, two unpatched previously known and two new 0-days.(MS10-046-LNK CVE vulnerability; MS10-061-Windows Print Spooler Vulnerability; MS10-073-Privilege exculpation vulnerability; MS10-XX-unknown)
10. Command and control Interface to allow hackers download and execute code.
11. Windows root-kit that hides malicious code.
12. Anti-Virus and security product evasion techniques.
13. The first ever PLC root-kit to hide modified code on PLCs.
14. Fingerprinting the Siemens PLC to potentially sabotage the system.

Initially Stuxnet infects an organisation via either the external network or infected removable drives. Once a foothold on one organisation device is achieved it begins to search for other devices on the corporate LAN through 0-day vulnerabilities, detailed later. As a lot of SCADA systems may not be connected directly to the organisation LAN through the use of Air-Gaps, Stuxnet relies on removable devices to introduce itself to these systems. These systems are generally un-patched and have no form of Anti-virus protection.

Stuxnet is designed to attack SCADA systems that are protected by an “Air Gap”. These systems do not have network connectivity so Stuxnet will not have the capability to download additional malicious code as part of the installation process. As a result of this Stuxnet is large [52] and comes with the complete functionality to install itself, propagate to new systems via removable device or network and sabotage a SCADA system.

Once Stuxnet finds Windows based systems with the Siemens PLC programming application “STEP 7” installed it uses the application default passwords to command the software to infect any Step 7 projects that may exist on the system. These modifications to code on Programmable Logic Controllers (PLCs), is likely to do considerable damage to processes that are controlled by the SCADA and PLC systems.

Stuxnet has the ability to copy itself to any additional removable drives that connect to infected machines. It also has the ability to hide its associated files if a user browser the memory with applications such as Windows Explorer. The worm also removes itself from an infected

memory key once that key has successfully compromised target machines in an attempt to cover its tracks.

Stuxnet has the ability to update its executables within an organisation through peer-to-peer methods which also come as part of the installation of Stuxnet. It was also found that this peer-to-peer traffic was encrypted to FIPS 140-2 standard [50].

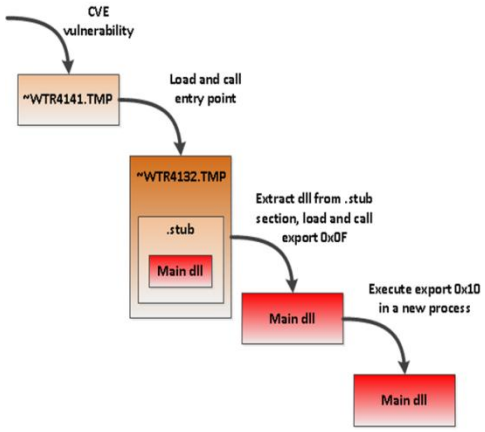
Stuxnet Architecture

As can be seen by from the introduction section of this assessment document the Stuxnet worm is very complex and has many different and new malicious techniques. There is a huge amount of documents on the topic of Stuxnet easily accessible on the web with varying levels of detail. The two documents used are [53] [54]. These two documents have a very high level of technical content. At the heart of Stuxnet is

1. A large .dll file that contains many different exports and resources.
2. Two encrypted configuration blocks.

These two components are contained in a "stub" section of a wrapped dropper file which is called ~WTR4132.TMP. The stub section is integral to the workings of the Stuxnet worm. When Stuxnet executes, the wrapper file extracts the .dll file from its stub section, maps it to memory as a module and calls one of the exports within.

A pointer in the stub section is passed to the export as a parameter. This initial export will in turn extract the .dll file from the stub section again and call another different export. The pointer in original stub section also passed to this export. This occurs constantly as Stuxnet is run so the original stub section is constantly passed around between different processes and functions as a parameter to the main payload so that each layer of the treat will have access to the main .dll and configuration blocks.



Stuxnet uses a technique to call exports from the main .dll file which allows for the execution of templates from its own resources. This is done by populating a template with appropriate data such as which .dll file to load and which export to call. This template is then injected into a new executable and another process to be executed. This new process will use the original main .dll and call the export passed in the crafted template.

Exports:

As mentioned already within the main .dll file there is a large number of Exports. Each of these exports carries out a unique purpose to control the operation of the Stuxnet worm. The table 1 below shows these Exports. A detailed examination of what each of these Export do is detailed

in [53] and [54]. I will detail some of these Exports in more detail below. The main ones that I will cover are Export 16 which is the "Main Installation" and Export 15 "Initial entry point"

DLL Exports	
Export #	Function
1	Infect connected removable drives, starts RPC server
2	Hooks APIs for Step 7 project file infections
4	Calls the removal routine (export 18)
5	Verifies if the threat is installed correctly
6	Verifies version information
7	Calls Export 6
9	Updates itself from infected Step 7 projects
10	Updates itself from infected Step 7 projects
14	Step 7 project file infection routine
15	Initial entry point
16	Main installation
17	Replaces Step 7 DLL
18	Uninstalls Stuxnet
19	Infects removable drives
22	Network propagation routines
24	Check Internet connection
27	RPC Server
28	Command and control routine
29	Command and control routine
31	Updates itself from infected Step 7 projects
32	Same as 1

Table 1: DLL Exports

DLL Resources	
Resource ID	Function
201	MrxNet.sys load driver, signed by Realtek
202	DLL for Step 7 infections
203	CAB file for WinCC infections
205	Data file for Resource 201
207	Autorun version of Stuxnet
208	Step 7 replacement DLL
209	Data file (%windows%\help\winmic.fts)
210	Template PE file used for injection
221	Exploits MS08-067 to spread via SMB.
222	Exploits MS10-061 Print Spooler Vulnerability
231	Internet connection check
240	LNK template file used to build LNK exploit
241	USB Loader DLL –WTR4141.tmp
242	MRxnet.sys rootkit driver
250	Exploits Windows Win32k.sys Local Privilege Escalation (MS10-073)

Table 2: DLL Resources

Resources:

The main .dll file also contains many different Resources that the Exports described above used to control the Stuxnet worm. These Resources can be either

1. Full .dll files, detailed in table 2 above.
2. Template executables.
3. Configuration files
4. Exploit modules

Bypass system protection when loading DLL's:

In any form of Windows malicious attack loading of infected .dll files onto the target system is a fundamental step that needs to be carried out. Stuxnet is no different and when it needs to install itself or any other dll it uses a specially designed behaviour blocking bypassing technique. These behaviour blocking processes commonly monitor the LoadLibary calls that are made from kernel.dll on the system when attempting to load dll's.

Stuxnet calls the LoadLibrary function with a specially crafted filename that does not exist on the system which causes the function to fail. Stuxnet has also hooked Ntdll.dll to monitor for requests to load specifically crafted file names which are mapped to another location specified by the Stuxnet. These files have been previously decrypted and sorted and have the name patterns of KERNEL32.DLL.ASLR.[XXXX] or SHELL32.DLL.ASLR. [XXXX].

The functions that are hooked within the Ntdll.dll are:

- ZwMapViewOfSection
- ZwCreateSection
- ZwOpenFile
- ZwCloseFile
- ZwQueryAttributesFile
- ZwQuerySection

Once a .dll file has been loaded the GetProcAddress function is used to find the address of a specific export within the .dll and that export is called. This can then be used to handing control to a new .dll.

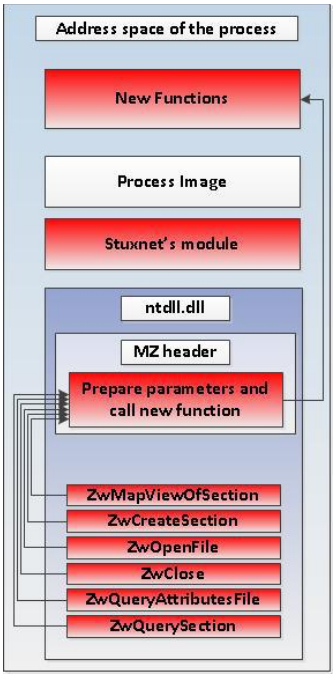


Diagram 1: Process of hooking functions in ntdll.dll

Injection Technique

When an Export is called by Stuxnet it typically injects the entire dll into another process and then calls that particular export. These dll's can be injected into

1. An existing process.
2. A newly created process.
3. A pre-selected trusted process.

When injecting into a pre-selected trusted process Stuxnet can keep the injected code in that process or instruct the trusted process to inject the code into another currently running process. The trusted processes that can by injected into are a set of default Windows processes and a selection of security processes listed below.

- Kaspershy KAV(avp.exe)
- Mcafee (Mcshield.exe)
- AntiVir (avguard.exe)
- BitDefender (bdagent.exe)
- Etrust (UmxCfg.exe)
- F-Secure (fsdfwd.exe)
- Symantec (rtvscan.exe)
- Symantec Common Client (ccSvcHst.exe)
- Eset NOD32 (ekrn.exe)
- Trend Pc-Cillin (tmpproxy.exe)

The registry of the potential victim is searched for indicators of the anti-virus applications running on the system and if found the version number that product is ascertained. This information is then used to determine the process of injection that will be used against the system. If the injection cannot bypass the security process the process fill fail and Stuxnet will not infect that target machine. Once it has been ascertained that injection without detection by AV can be achieved Stuxnet uses one of the target processes listed below to inject itself onto a victim:

- LSASS.exe
- WinLogon.exe
- Svchost.exe
- Installed security product process.

The table below details which one of these processes is to be used to inject the worm, based on the security product installed.

Process Injection	
Security Product Installed	**Injection target**
KAV v1 to v7	LSASS.EXE
KAV v8 to v9	KAV Process
McAfee	Winlogon.exe
AntiVir	Lsass.exe
BitDefender	Lsass.exe
ETrust v5 to v6	Fails to Inject
ETrust (Other)	Lsass.exe
F-Secure	Lsass.exe
Symantec	Lsass.exe
ESET NOD32	Lsass.exe
Trend PC Cillin	Trend Process

Table 3: Injection Processes

Stuxnet also determines which of the two Microsoft privilege escalation vulnerabilities needs to use before running the identified process. Once privilege escalation has been carried out this process can now run in suspended mode.

When Stuxnet initially surface these two vulnerabilities were unknown by Microsoft. These are commonly called "zero-day or 0-day vulnerabilities". On the 12th of October 2010 Microsoft released Security Bulletin MS10-073-"Vulnerabilities in Windows Kernel-Mode Drivers Could Allow Elevation of Privilege" [55] and a patch to fix one of these vulnerabilities. This vulnerability resides in code that calls a function in a function pointer table. The index into the table is not validated properly allowing code to be called outside of the function table. There is currently no patch or no details available from Microsoft on the second vulnerability.

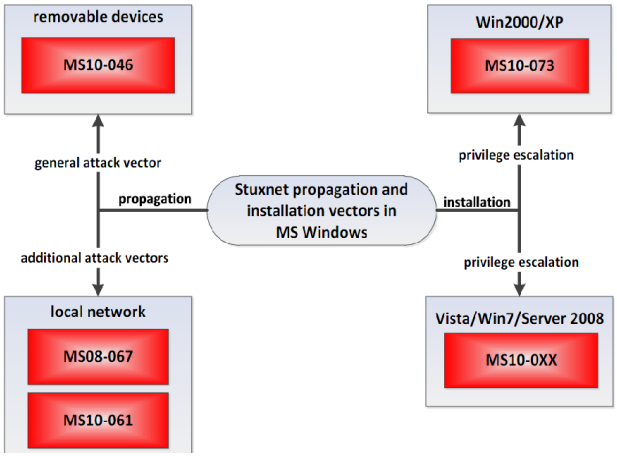


Diagram 2: Vulnerabilities used by Stuxnet to install and propagate.

Stuxnet bypasses security products by extracting a template file and creating a new section called .verif. This section is made large enough so that entry point address of the target process falls within this .verif section. At that entry point address in the template file Stuxnet places a jump to the actual desired entry point to inject code. These bytes are written to the target process and a ResumeThread function is called allowing the process to execute and call the injected code. The stub section of the main dll is also mapped to the memory of the new process with the use of a shared section. When an injected process is run it unpacks the .dll file from the stub section and calls the desired export.

Configuration Data file:

As already mentioned Stuxnet consists of a main dll and configuration data. This configuration data is used to control how Stuxnet acts on a compromised computer. When Stuxnet is installed

on a compromised victim this configuration data is updated and a computer description block is appended to the configuration data. The computer description holds such data as computer name, domain, OS version, infected S7P paths etc. This file is used to conduct checks on the information gathered about the system.

Installation Process

The installation of the Stuxnet worm on a target machine is very complex and is outlined in this section of the document. This process is carried out by the execution of Export 15 initially followed by Export 16.Export 15 is called when the main .dll file is loaded for the first time. This

1. Checks that's its on a compatible version of Windows
2. Checks that the PC's not infected already
3. Elevate the privilege of the current process
4. Checks which anti virus is installed
5. Determines what the best process to inject into

Once this export has completed, Stuxnet has gained administration rights on the victim via a process with admin rights, conducted a detailed checks of OS and AV and has decided which processes the dll will be injected into. Export 16 now called which is the main installer of Stuxent. The steps Export 16 carries out are

1. Checks the date and version number of the victim
2. Decrypts, creates and installs the rootkit files and registry keys,
3. Injects itself into the service.exe process,
4. Injects itself into the Step7 process to infect projects
5. Sets up the global metexes used to communicate between different components
6. Connects to the RPC server

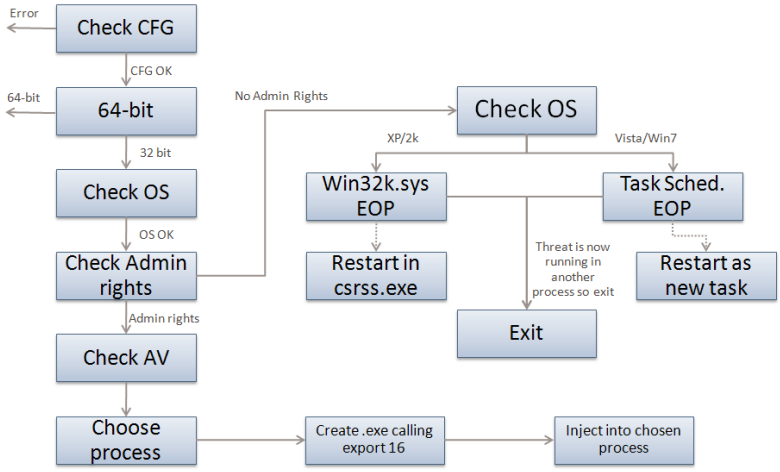


Diagram 3: Flowchart of Export 15

It can be seen from the flow chart above that Export 15

1. Checks the configuration data is up-to-date. This data is stored in two locations so stuxnet checks which is more up-to-date and uses that one.
2. Checks if victim is a 64 bit and if so terminated
3. Checks 0S of victim as Stuxnet can only on specific windows platforms.
4. Checks if it has Admin rights on machine. Will execute one of two zero-day escalations, already explained, if required. This will result in the main dll file running a new process within the csrss.exe or as new task, with admin rights

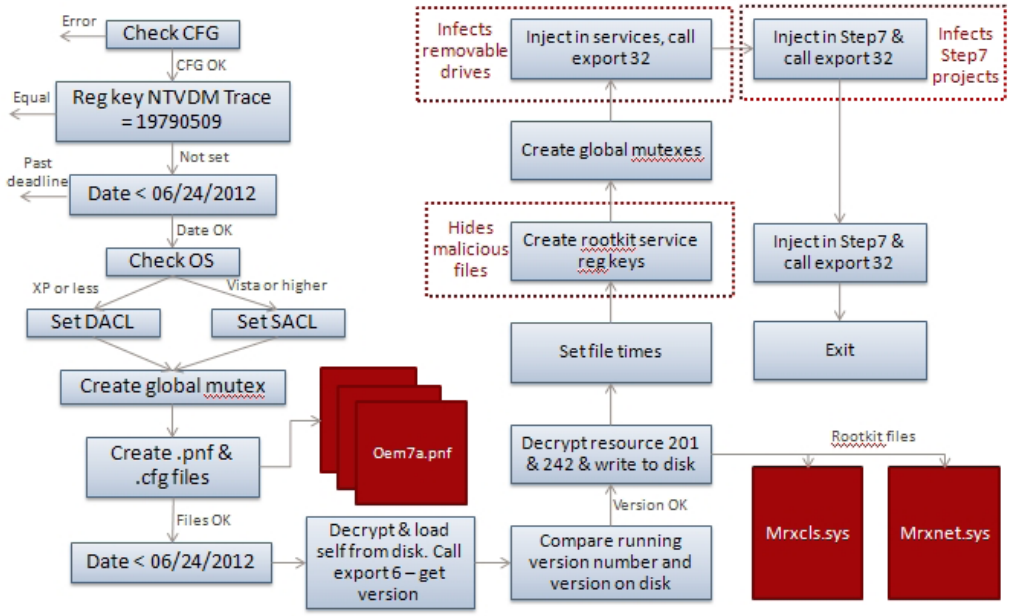


Diagram 4: Flowchart of Export 16

It can be seen from Diagram 4 above that Export 16

1. Checks the configuration data is valid then
2. Checks the value "NTVDM TRACE" a registry key. If this value is 19790509 the threat will exit. This value appears to be a date 9/5/1979 which is a date a prominent member of the Jewish Iranian Community was executed.
3. It reads the date from its configuration data file which is always 24th June 2012. If current date is greater than that date, infection will not occur.
4. Stuxnet uses global "mutexes" to communicate and share resources between different components. It creates these by using SetSecurityDescriptorDacl (SetDACL) for XP or less and SetSecurityDescriptorSacl API for Vista of greater (SetSACL)
5. Creates 3 encrypted files from the .stub which are encrypted and written to disk
    - Main Stuxnet payload .dll saved as Oem7a.pnf
    - 90 byte file copied to %SystemDrive%\inf\mdmeric3.PNF
    - Configuration data file copied to %SystemDrive%\inf\mdmcpq3.PNF

    A log file also copied to %SystemDrive%\inf\oem6C.PNF as part of this step
6. Another date check, this time with date in configuration file just created.
7. Checks if current version is newer than any existing encrypted version on disk. (Export 6 used to return the version number)
8. Extract, decode and write two files from resource section to disk. These are the Rootkit files that are read for resources 201 and 242 and are written to disk with the names "Mrxnet.sys" and Mrxcls.sys". One file is a load point (explained below) and the other is used to hide malicious files on victim, replace the Stuxnet files if removed. The file time on these files match the times of other files in the system directory to help avoid suspicion. It also creates the registry entries to load these files as services that will run on Windows startup.
9. Once the rootkit is established Export 16 creates some more global mutexes to signal that installation was successful.
10. Control is passed to two other exports to continue installation and infection routines.
    - Injects the payload .dll file into the services.exe process and calls export 32 which is responsible for infecting newly connected removable drives and for starting the RPC server.

- Injects payload .dll file into the Step7 process S7tgtopx.exe and calls Export 2 which infects Step7 project files

Now that all the spreading and payload routines have been activated, execution of Stuxnet continues via the 2 injections and via the driver files and services that were created.

Load Point:

As already mentioned above, Export 16 uses Resource 242 to create the load point file MrxCls.sys driver. This driver allows Stuxnet to be executed every time an infected system boots and thus act as the main-load point for the worm. The driver is registered as a boot start service, creating a registry key to inject and execute copies of Stuxnet into specific processes. The driver comes as part of the payload of the worm and is digitally signed, by one of two stolen code signing keys. These keys are from two different Taiwanese certification companies, JMicron and Realtek. Speculation on how these were stolen is well documented but there is no evidence on how this was done. Verisign has since revoked these two signed certificates.

The driver also contains an encrypted block of data which when decrypted contains a registry keys and registry value pairs which are a list of pairs of target names and module to inject. These are

1. services.exe (to be injected with %Windir%\inf\oem7A.PNF)
2. S7tgtopx.exe (to be injected with %Windir%\inf\oem7A.PNF)
3. CCProjectMgr.exe (to be injected with %Windir%\inf\oem7A.PNF)
4. explorer.exe (to be injected with %Windir%\inf\oem7m.PNF)

Service.exe, S7tgtopx.exe and CCProjectMgr.exe will be injected with oem7a.pnf which is a copy of the main Stuxnet dll. Explorer.exe is injected with oem7m.pnf.

Command and Control:

Up to this point I have covered how Stuxnet installs itself, drops its files and gathers information on the system it has infected. When all this is done the command and control server kicks in and sends some basic information about the compromised victim to the attacker via port 80 HTTP. For this two webservers were used

1. http://www.mypremierfutbol.com/
2. http://www.todaysfutbol.com

These are servers in Malaysia and Denmark that have since been redirected to prevent attackers controlling compromised computers. Stuxnet had the capability to update itself with new commands and controls from these servers but there has no evidence that this took place before the redirection took place.

Victims system data such as IP address, domain name, version of OS and much more is gathered by Export 28 . Export 29 can then be used to send this data as a payload to one of the two target server. The process used to do this is iexporer.exe. If iexplorer.exe does not exist on the victim a browser process is created and injected. Before contacting the Command and Control servers, Export 29 first needs to tests that network connectivity can be achieved by testing connections to www.windowsupdate.com or www.msm.com.

Once this test pass has passed the network packet data is built and the payload is sent to one of the URL's. This data is sent in such a way to bypass corporate firewall blocking rules. The Stuxnet server processes the received data from the victim and may send a response. Depending on the command data revived from the server the payload module will either be loaded in the current process or into a separate process via Remote Procedure Call (RPC). This gives Stuxnet a backdoor facility to upload and run any code to a victim. This could have allowed attackers to download and execute additional tools and deliver updates versions of Stuxnet. However this process was minimised with the redirection of the URL for controlling the attack.

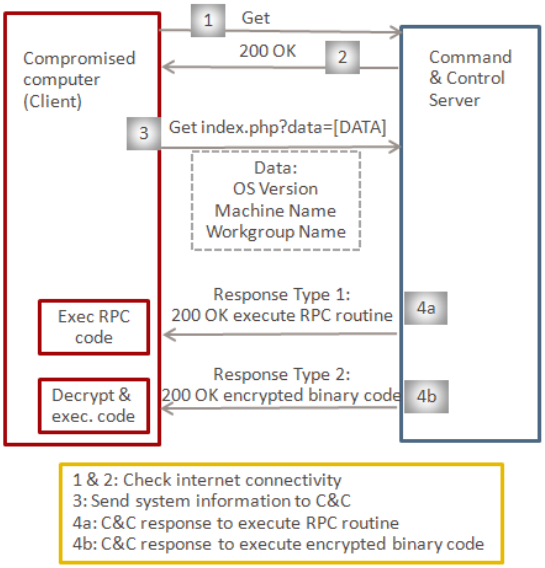


Diagram 5: Flowchart of command and command process

Windows Rootkit Functionality:

As already mentioned Stuxnet uses Export 16 to extract and creates a driver file called MrxNet.sys, which has one of two forget digital signature. This driver scans file system driver objects

1. \filesystem\ntfs
2. \filesystem\fastfat
3. \filesystem\cdfs

A new device object is created by Stuxnet and attached to the device chain of each of the devices managed by the above driver objects. By inserting these objects Stuxnet will be able to intercept I/O Request Packet (IRP) to these devices. MrxNet.sys also registers to a file system registration call-back routine to hook newly created file system objects.

With this interception technique the MrxNet.sys driver can monitor the "directory control" IRP such as those sent to a device when a user browses a directory and requests a list of files it contains. It then filters out two types of files from the response from these requests. (***.LNK** and **~wtr****.tmp**). These filters hide files used by Stuxnet and are present on infected removable devices. These files typically are

Copy of Copy of Copy of Copy of Shortcut to.lnk

- Copy of Copy of Copy of Shortcut to.lnk
- Copy of Copy of Shortcut to.lnk
- Copy of Shortcut to.lnk
- ~wtr4132.tmp
- ~wtr4141.tmp

Propagation Methods

Stuxnet can propagate itself by infecting removable drives or copying itself over a network using a verity of different means. It can also propagate itself to Step 7 projects which auto-execute when a project is opened.

Network propagation is conducted with the use of Export 22 within Stuxnet. This Export has five different methods to infect a remote host. These are

1. Peer to Peer communication and updates
2. Infects machines with Siemens WinCC application software via a hard-coded database server password
3. Propagating through network shares
4. Propagating through the MS10-061 Print Spooler Zero-Day Vulnerability
5. Propagating through the MS08-067 Windows Server Service Vulnerability

Removable drive propagation is used to spread to and from removable devices. It is the main method used to infect non-networked Windows machines that are protected by an "Air-Gap" as already described.
This method of propagation uses two different methods to spread the worm. The first method uses a vulnerability that allows for auto-execution when viewing removable drives. The second method uses the autorun.inf file.

When a removable drive is inserted into a machine that is infected with Stuxnet some checks are performed before copying the worm propagation files to it

1. The drive was not just infected
2. The infection is less than 21 days old
3. The drive has at least 5MB of free space
4. The drive has at least 3 files

Once these conditions have been met the following files are copied to the removable device.

- ~wtr4132.tmp(contains main dll in the stub section derive from Resource 210)
- ~wtr4141.tmp (loads ~wtr4141.tmp and built from Resource 241)
- Copy of Shortcut to.lnk
- Copy of Copy of Shortcut to.lnk
- Copy of Copy of Copy of Shortcut to.lnk
- Copy of Copy of Copy of Copy of Shortcut to.lnk

The .lnk files contain an exploit, known as the LNK Vulnerability (CVE-2010-2568) (MS10-046), that will automatically execute the ~wtr4141.tmp file when simply viewing the folder. The ~wtr4141.tmp file will then load the ~wtr4132.tmp and hide the presence of both files that are on the removable drive. This is done as the Windows Rootkit defined above has not yet been installed to hide these files. Hiding the files id carried out by hooking certain API's in the kernal.dll and ntdll.dll. It replaces the code of these functions with its own code to hide the malicious files on the removable device.

The ~wtr4132.tmp file is extracted into memory and Export 15 is called to carry the steps outlined in the installation process described above. The running of ~wtr4132.tmp involves

loading a .dll file, techniques explained above in "Bypassing system protection when loading DLL's" is used for this process.

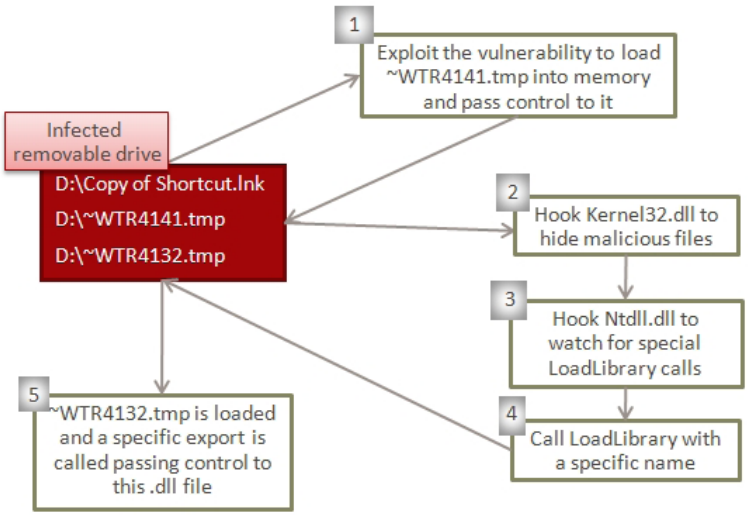


Diagram 6: USB Execution Flow

AutoRun.Inf was used in earlier versions of Stuxnet which did not have the LNK 0-day exploit. This is a configuration file placed on removable drives that instruct Windows to automatically execute a file on the drive when it is inserted to a machine. Stuxnet used this technique to get the autorun command to install the malicious components and hide that anything suspicious has occurred.

The stuxnet also has the capability of deleting itself form the removable drive after it has infected 3 computers in an attempt to cover its tracks.

The Propagation to Step 7 project files is done by opening project files inside the s7tgtopx.exe process which is the WinCC manager to manage projects and files. The Import Address Tables of the following DLLs are modified:

- In s7apromx.dll, mfc42.dll, and msvcrt.dll, CreateFileA is replaced to point to "CreateFileA_hook".
- In ccprojectmgr.exe, StgOpenStorage is replaced to point to "StgOpenStorage_hook".

These hooks give control to the Stuxnet worm to insert malicious code into these Step 7 projects that allows a path to a project as an input and infect the project causing Stuxnet to execute when the project is loaded. If this project it later copied to a machine not infected with Stuxnet and the project is opened that machine becomes a victim.

Modifying PLCs

The Stuxnet worm looks for markers to indicate that either the Siemens WinCC or Step 7 applications are on the infected system. If these applications are not the worm will attempt to propagate using the techniques covered.

At this point it important to remember that the goal of Stuxnet is to infect specific SCADA systems and mainly the Siemens Programmable Logic Controllers (PLCs). These PLCs are autonomous devices which when loaded with STL or SCL programs, by a windows PCs, function by themselves to execute, control and monitor industrial processes.

These PLC to PC communication utilise s7otbxdx.dll. Stuxnet replaces the legitimate version of this file with its own version which is the PLC rootkit, so that it can

1. Monitor PLC blocks being written to and from the PLC
2. Infect a PLC `by inserting its own blocks or infecting existing blocks of code
3. Mask the fact that the PLC is infected.

The Step 7 application running on the Windows machine calls different routines in the s7otbxdx.dll file. Stuxnet renames the existing file to s7otbxsx.dll and puts it's own malicious copy onto the system. Stuxnet can now intercept any call that is made to access the PLC by the Step 7 application. The Malicious is quiet small and does not have all of the functions of the original renamed .dll. Any entry not in the malicious version is forwarded to the renamed s7otbxsx.dll for processing as if there was no malicious action. However there are 16 exports within the malicious file that are not forwarded. These exports are mainly in the form of read, write and enumerate blocks of code on the PLC. Using these exports Stuxnet will be able to modify data sent to and returned to a PLC from the Step 7 application. This will take place without raising suspicion and can also be done in such a way to hide the malicious code on the infected PLC. Diagram 7 below helps to show this process.

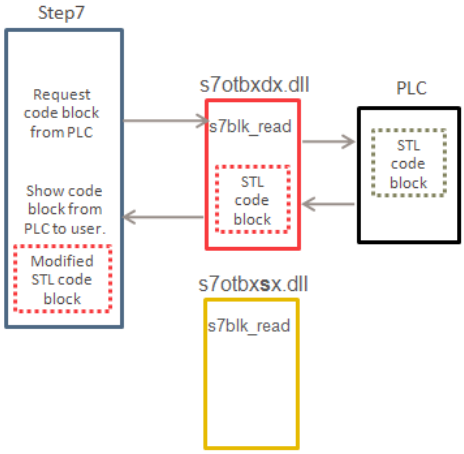


Diagram 7: Infected communication

Stuxnet has three different injection techniques depending on the characteristics of the targeted system. Two of these are very similar sequence of tasks called "Sequence A and B" the third is called "Sequence C" (Page 35 [53]). Different sequences are used to infect different series of products in the Siemens PLC family range.

The injection process consists of inserting code blocks and data blocks into the PLC to alter its behaviour. The PLC rootkit in the malicious copy of s7otbxdx.dll is responsible for Stuxnet to continue to exist undetected by

1. Reading requests for its own malicious code blocks on PLC
2. Reading requests from infected blocks on PLC
3. Writing requests that could overwrite Stuxnet code in PLC

The injected code into these PLC devices is generic but the process which these PLC devices operate in, are not generic. Each PLC device is configured with bespoke hardware and software designs with many different ways to connect motors and other devices. The Stuxnet PLC injection code is only makes very small changes to the programs on them. This malicious attack looks to be more as a proof of concept attack or a form of warning to government organisations of the capability that this type of attack could have.

Appendix 3: List of Current Activities in ICS/SCADA Security

- American Gas Association (AGA) 12
- American Petroleum Institute (API) Standard 1164
- Control Systems Security Centre - INL
- Chemical Sector Cyber Security Program
- Chemical Industry Data eXchange (CIDX)
- DHS Control Systems Security Program (CSSP)
- DHS CSSP Recommended Practices
- Electrical Power Research Institute
- Institute of Electrical and Electronics Engineers, Inc (IEEE)
- Institute for Information Infrastructure Protection (I3P)
- International Electro technical Commission (IEC) Technical Committees 65 and 57
- ISA100 Wireless Systems for Automation
- ISA99 Industrial Automation and Control System Security Standards
- ISO/IEC 27002:2005 Security Techniques - Code of Practice for Information Security Management
- ISO/IEC 27001:2005 Information technology – Security techniques – Information security management systems – Requirements
- International Council on Large Electric Systems (CIGRE)
- LOGIIC – Linking the Oil and Gas Industry to Improve Cyber Security
- National SCADA Test Bed (NSTB)
- NIST 800 Series Security Guidelines
- NIST Industrial Control System Security Project
- North American Electric Reliability Corporation (NERC)
- SCADA and Control Systems Procurement Project
- Smart Grid Interoperability Panel (SGIP) Cyber Security Working Group (CSWG)

Appendix 4: US Energy Department "21 Steps to Improve Cyber Security of SCADA Network"

Specific actions to be taken to increase the security of SCADA networks

1. Identify all connections to SCADA networks
2. Disconnect unnecessary connections to the SCADA network
3. Evaluate and strengthen the security of any remaining connections to the SCADA network
4. Harden SCADA networks by removing or disabling unnecessary services
5. Do not rely on proprietary protocols to protect your system
6. Implement the security features provided by device and system vendors
7. Establish strong controls over any medium that is used as a backdoor into the SCADA network
8. Implement internal and external intrusion detection systems and establish 24-hour-a-day incident monitoring
9. Perform technical audits of SCADA devices and networks, and any other connected networks, to identify security concerns
10. Conduct physical security surveys and assess all remote sites connected to the SCADA network to evaluate their security
11. Establish SCADA "Red Teams" to identify and evaluate possible attack scenarios

Management Actions

12. Clearly define cyber security roles, responsibilities, and authorities for managers, system administrators, and users
13. Document network architecture and identify systems that serve critical functions or contain sensitive information that require additional levels of protection
14. Establish a rigorous, ongoing risk management process
15. Establish a network protection strategy based on the principle of defense-in-depth
16. Clearly identify cyber security requirements
17. Establish effective configuration management processes
18. Conduct routine self-assessments
19. Establish system backups and disaster recovery plans
20. Senior organizational leadership should establish expectations for cyber security performance and hold individuals accountable for their performance
21. Establish policies and conduct training to minimize the likelihood that organizational personnel will inadvertently disclose sensitive information regarding SCADA system design, operations, or security controls.

Appendix 5: Department of Homeland Security Catalogue

1. Security Policy
2. Organizational Security
3. Personnel Security
4. Physical and Environmental Security
5. System and Services Acquisition
6. Configuration Management
7. Strategic Planning
8. System and Communication Protection
9. Information and Document Management
10. System Development and Maintenance
11. Security Awareness and Training
12. Incident Response
13. Media Protection
14. System and Information Integrity
15. Access Control
16. Audit and Accountability
17. Monitoring and Reviewing Control System Security Policy
18. Risk Management and Assessment
19. Security Program Management

Appendix 6:NIST SP 800-82

Network Architecture

1. Firewalls
2. Logically Separated Control Network
3. Network Segregation
    - Dual-Homed Computer/Dual Network Interface Cards (NIC)
    - Firewall between Corporate Network and Control Network
    - Firewall and Router between Corporate Network and Control Network
    - Firewall with DMZ between Corporate Network and Control Network
    - Paired Firewalls between Corporate Network and Control Network
    - Network Segregation Summary
4. Recommended Defense-in-Depth Architecture
5. General Firewall Policies for ICS
6. Recommended Firewall Rules for Specific Services
    - Domain Name System (DNS)
    - Hypertext Transfer Protocol (HTTP)
    - FTP and Trivial File Transfer Protocol (TFTP)
    - Telnet
    - Simple Mail Transfer Protocol (SMTP)
    - Simple Network Management Protocol (SNMP)
    - Distributed Component Object Model (DCOM)
    - SCADA and Industrial Protocols
7. Network Address Translation (NAT)
8. Specific ICS Firewall Issues
    - Data Historians
    - Remote Support Access
    - Multicast Traffic
9. Single Points of Failure
10. Redundancy and Fault Tolerance
11. Preventing Man-in-the-Middle Attacks

NIST ICS/SCADA Security Controls

1. Management Controls
    - Security Assessment and Authorization
    - Security Planning
    - Risk Assessment
    - System and Services Acquisition
    - Program Management
2. Operational Controls
    - Personnel Security
    - Physical and Environmental Protection
    - Contingency Planning
    - Configuration Management
    - Maintenance
    - System and Information Integrity Monitoring
    - Media Protection
    - Incident Response
    - Security Awareness and Training
3. Technical Controls
    - Identification and Authentication
    - Access Control
    - Audit and Accountability
    - System and Communications Protection

Appendix 7: Router Configuration

```
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R2
!
boot-start-marker
boot-end-marker
!
aaa new-model
!
aaa authentication login default local
aaa authentication login VPNAUTH local
aaa authorization network VPNAUTH local
!
aaa session-id common
!
resource policy
!
memory-size iomem 10
ip subnet-zero
!
ip cef
no ip dhcp use vrf connected
!
no ip ips deny-action ips-interface
!
no ftp-server write-enable
!
username cisco password 0 cisco
username john privilege 15 password 0 scada
!
crypto isakmp policy 10
 encr aes 256
 authentication pre-share
 group 2
crypto isakmp keepalive 30 5
no crypto isakmp ccm
crypto isakmp xauth timeout 30
!
crypto isakmp client configuration group scadagroup
 key scada
 pool VPNPOOL
 netmask 255.255.255.0
!
crypto ipsec transform-set MYTRANS esp-aes 256 esp-sha-hmac
!
crypto dynamic-map mymap 10
 set transform-set MYTRANS
```

```
 reverse-route
!
crypto map staticmap client authentication list VPNAUTH
crypto map staticmap isakmp authorization list VPNAUTH
crypto map staticmap client configuration address respond
crypto map staticmap 10 ipsec-isakmp dynamic mymap
!
interface FastEthernet0/0
 ip address 157.190.120.1 255.255.255.0
 duplex auto
 speed auto
 crypto map staticmap
!
interface FastEthernet0/1
 ip address 157.190.137.1 255.255.255.0
 duplex auto
 speed auto
!
interface BRI0/0/0
 no ip address
 shutdown
!
interface Serial0/1/0
 no ip address
 shutdown
 no fair-queue
 clockrate 125000
!
interface Serial0/1/1
 no ip address
 shutdown
!
ip local pool VPNPOOL 172.16.2.100 172.16.2.150
ip classless
!
ip http server
no ip http secure-server
!
access-list 10 permit 172.16.0.0 0.0.255.255
!
control-plane
!
line con 0
line aux 0
line vty 0 4
 logging synchronous
!
scheduler allocate 20000 1000
```