410.2
Field Devices and Controllers

SANS

THE MOST TRUSTED SOURCE FOR INFORMATION SECURITY TRAINING, CERTIFICATION, AND RESEARCH | sans.org

Copyright © 2019, Justin Searle. All rights reserved to Justin Searle and/or SANS Institute.

PLEASE READ THE TERMS AND CONDITIONS OF THIS COURSEWARE LICENSE AGREEMENT ("CLA") CAREFULLY BEFORE USING ANY OF THE COURSEWARE ASSOCIATED WITH THE SANS COURSE. THIS IS A LEGAL AND ENFORCEABLE CONTRACT BETWEEN YOU (THE "USER") AND THE SANS INSTITUTE FOR THE COURSEWARE. YOU AGREE THAT THIS AGREEMENT IS ENFORCEABLE LIKE ANY WRITTEN NEGOTIATED AGREEMENT SIGNED BY YOU.

With the CLA, the SANS Institute hereby grants User a personal, non-exclusive license to use the Courseware subject to the terms of this agreement. Courseware includes all printed materials, including course books and lab workbooks, as well as any digital or other media, virtual machines, and/or data sets distributed by the SANS Institute to the User for use in the SANS class associated with the Courseware. User agrees that the CLA is the complete and exclusive statement of agreement between The SANS Institute and you and that this CLA supersedes any oral or written proposal, agreement or other communication relating to the subject matter of this CLA.

BY ACCEPTING THIS COURSEWARE, YOU AGREE TO BE BOUND BY THE TERMS OF THIS CLA. BY ACCEPTING THIS SOFTWARE, YOU AGREE THAT ANY BREACH OF THE TERMS OF THIS CLA MAY CAUSE IRREPARABLE HARM AND SIGNIFICANT INJURY TO THE SANS INSTITUTE, AND THAT THE SANS INSTITUTE MAY ENFORCE THESE PROVISIONS BY INJUNCTION (WITHOUT THE NECESSITY OF POSTING BOND), SPECIFIC PERFORMANCE, OR OTHER EQUITABLE RELIEF.

If you do not agree, you may return the Courseware to the SANS Institute for a full refund, if applicable.

User may not copy, reproduce, re-publish, distribute, display, modify or create derivative works based upon all or any portion of the Courseware, in any medium whether printed, electronic or otherwise, for any purpose, without the express prior written consent of the SANS Institute. Additionally, User may not sell, rent, lease, trade, or otherwise transfer the Courseware in any way, shape, or form without the express written consent of the SANS Institute.

If any provision of this CLA is declared unenforceable in any jurisdiction, then such provision shall be deemed to be severable from this CLA and shall not affect the remainder thereof. An amendment or addendum to this CLA may accompany this courseware.

SANS acknowledges that any and all software and/or tools, graphics, images, tables, charts or graphs presented in this courseware are the sole property of their respective trademark/registered/copyright owners, including:

AirDrop, AirPort, AirPort Time Capsule, Apple, Apple Remote Desktop, Apple TV, App Nap, Back to My Mac, Boot Camp, Cocoa, FaceTime, FileVault, Finder, FireWire, FireWire logo, iCal, iChat, iLife, iMac, iMessage, iPad, iPad Air, iPad Mini, iPhone, iPhoto, iPod, iPod classic, iPod shuffle, iPod nano, iPod touch, iTunes, iTunes logo, iWork, Keychain, Keynote, Mac, Mac Logo, MacBook, MacBook Air, MacBook Pro, Macintosh, Mac OS, Mac Pro, Numbers, OS X, Pages, Passbook, Retina, Safari, Siri, Spaces, Spotlight, There's an app for that, Time Capsule, Time Machine, Touch ID, Xcode, Xserve, App Store, and iCloud are registered trademarks of Apple Inc.

PMP and PMBOK are registered marks of PMI.

SOF-ELK® is a registered trademark of Lewes Technology Consulting, LLC. Used with permission.

SIFT® is a registered trademark of Harbingers, LLC. Used with permission.

Governing Law: This Agreement shall be governed by the laws of the State of Maryland, USA.

ICS410.2 ICS/SCADA Security Essentials

SANS

Field Devices and Controllers

© 2019 Justin Searle | All Rights Reserved | Version E01_01

The **SANS ICS410**, ICS/SCADA Security Essentials course, was developed by a collection of experts whose diverse work experiences, knowledge, and skills truly blend together to cover the very specific content areas for this course.

Justin Searle is the Director of ICS Security at InGuardians, specializing in ICS security architecture design and penetration testing. Justin led the Smart Grid Security Architecture group in the creation of NIST Interagency Report 7628 and has played key roles in the Advanced Security Acceleration Project for the Smart Grid (ASAP-SG), National Electric Sector Cybersecurity Organization Resources (NESCOR), and Smart Grid Interoperability Panel (SGIP). Justin has taught courses in hacking techniques, forensics, networking, and intrusion detection for multiple universities, corporations, and security conferences. He is currently a Senior Instructor for the SANS Institute. In addition to electric power industry conferences, Justin frequently presents at top international security conferences such as Black Hat, DEFCON, OWASP, Nullcon, and AusCERT. Justin co-leads prominent open source projects including the The Control Thing Platform, Samurai Web Testing Framework (SamuraiWTF), Samurai Security Testing Framework for Utilities (SamuraiSTFU), Yokoso!, and Laudanum. Justin has an MBA in International Technology and is a CISSP and SANS GIAC certified Incident Handler (GCIH), Intrusion Analyst (GCIA), Web Application Penetration Tester (GWAPT), and GIAC Industrial Control Security Professional (GICSP).

Dr. Eric Cole is an industry-recognized security expert with over 20 years of hands-on experience. Dr. Cole has experience in information technology with a focus on helping customers focus on the right areas of security by building out a dynamic defense. Dr. Cole has a master's degree in computer science from NYIT and a doctorate from Pace University with a concentration in information security. He served as CTO of McAfee and Chief Scientist for Lockheed Martin. Dr. Cole is the author of several books, including *Advanced Persistent Threat*, *Hackers Beware*, *Hiding in Plain Sight*, *Network Security Bible,* 2nd Edition, and *Insider Threat*. Eric is the inventor of over 20 patents and is a researcher, writer, and speaker. He is also a member of the Commission on Cyber Security for the 44th President and several executive advisory boards. Dr. Cole is the founder and an executive leader at Secure Anchor Consulting, where he provides leading-edge cybersecurity consulting services, expert witness work, and leads research and development initiatives to advance the state-of-the-art in information systems security. Dr. Cole is actively involved with the SANS Technology Institute (STI). He is a SANS faculty Fellow who works with students, teaches, and develops and maintains courseware.

Eric Cornelius is the Director of Critical Infrastructure and Industrial Control Systems (ICS) at Cylance, Inc. He is responsible for the thought leadership, architecture, and consulting implementations for the company. His leadership keeps organizations safe, secure, and resilient against advanced attackers. Previously, Eric served as the Deputy Director and Chief Technical Analyst for the Control Systems Security Program at the US Department of Homeland Security. As an active researcher in the field of cybersecurity since 2002, Eric supported many "boots-on-the-ground" engagements involving penetration testing, forensics, and malware analysis. Through these engagements, he aided multiple government, military, and private sector organizations in protecting their networks and industrial control systems. In addition to his years of technical leadership, Eric literally wrote the book on incident response in the ICS arena. Eric's extensive knowledge of critical infrastructure and those who attack it will be brought to bear at Cylance as he leads a team of experts in securing America's critical systems.

Contributing Authors

Michael Assante is currently the SANS project lead for Industrial Control System (ICS) and Supervisory Control and Data Acquisition (SCADA) security. He served as Vice President and Chief Security Officer of the North American Electric Reliability Corporation (NERC), where he oversaw industry-wide implementation of cybersecurity standards across the continent. Prior to joining NERC, Michael held a number of high-level positions at Idaho National Labs and he served as Vice President and Chief Security Officer for American Electric Power. His work in ICS security has been widely recognized and he was selected by his peers as the winner of *Information Security Magazine's* security leadership award for his efforts as a strategic thinker. The RSA 2005 Conference awarded him its outstanding achievement award in the practice of security within an organization. He has testified before the US Senate and House and was an initial member of the Commission on Cyber Security for the 44th Presidency. Prior to his career in security, Michael served in various naval intelligence and information warfare roles, and he developed and gave presentations on the latest technology and security threats to the Chairman of the Joint Chiefs of Staff, Director of the National Security Agency, and other leading government officials. In 1997, he was honored as a Naval Intelligence Officer of the Year.

Tim Conway is currently the Technical Director of ICS and SCADA programs at SANS. He is responsible for developing, reviewing, and implementing technical components of the SANS ICS and SCADA product offerings. He was formerly the Director of CIP Compliance and Operations Technology at Northern Indiana Public Service Company (NIPSCO) where he was responsible for Operations Technology, NERC CIP Compliance, and the NERC training environments for the operations departments within NIPSCO Electric. Tim was previously an EMS Computer Systems Engineer at NIPSCO for eight years, with responsibility over the control system servers and the supporting network infrastructure. He previously served as the Chair of the RFC CIPC, is the current Chair of the NERC CIP Interpretation Drafting Team, a current member of the NESCO advisory board, the current Chair of the NERC CIPC GridEx 2013 Working Group, and the current Chair of the NBISE Smart Grid Cyber Security panel.

© 2019, Justin Searle

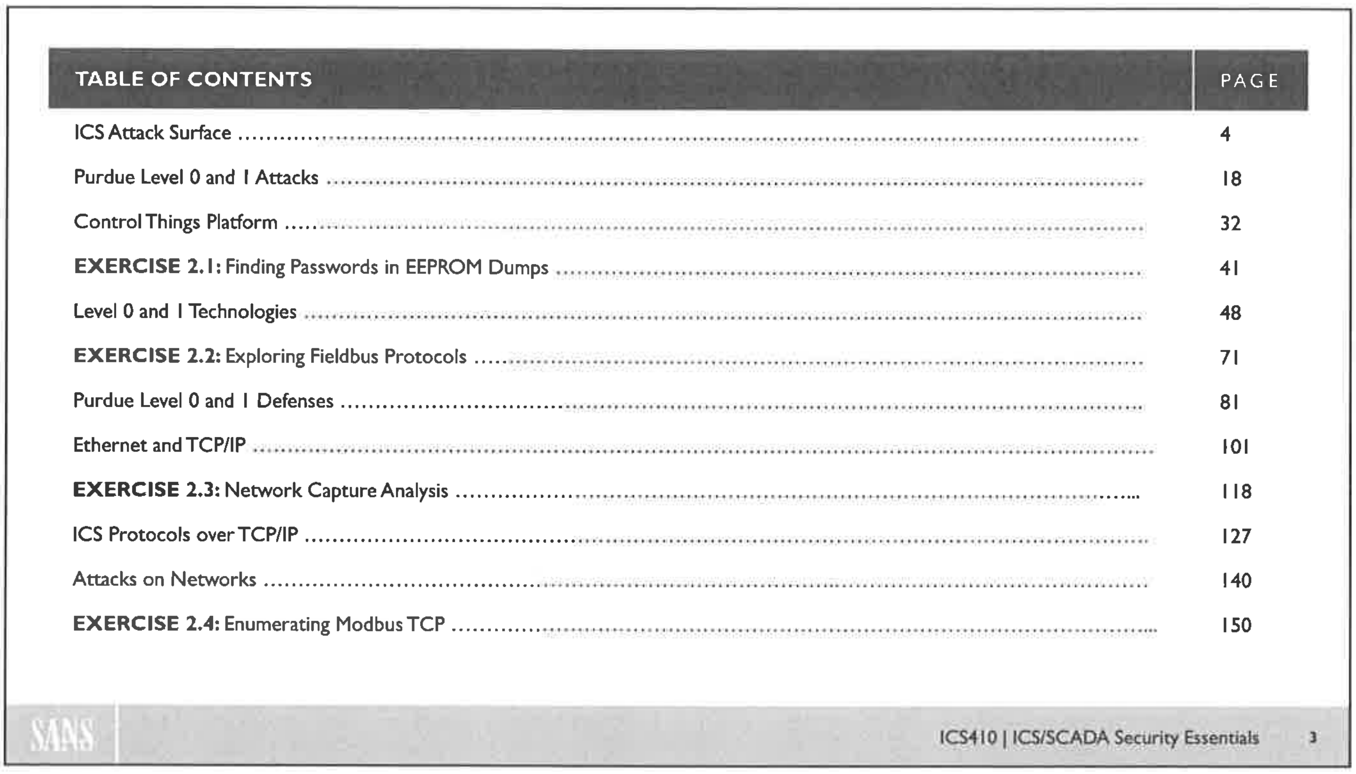


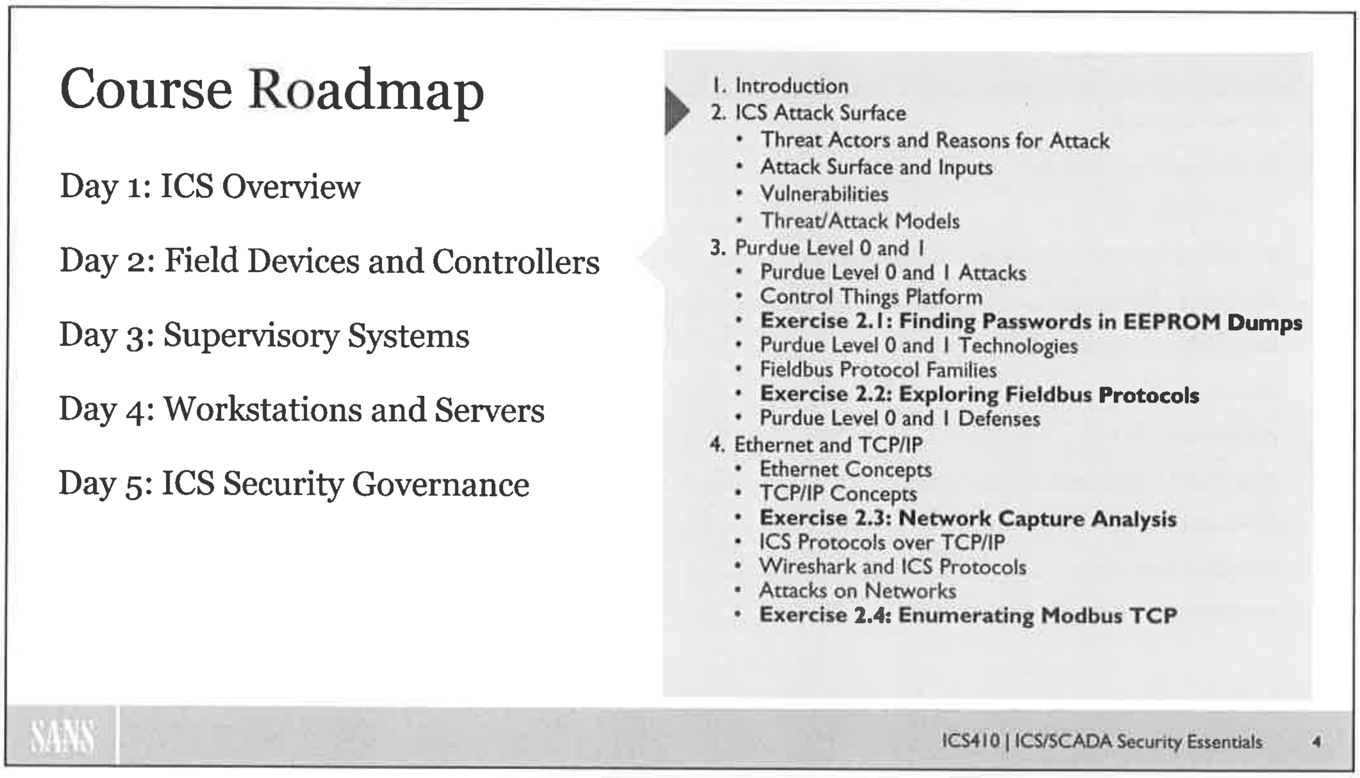
TABLE OF CONTENTS

PAGE

SANS

ICS410 | ICS/SCADA Security Essentials 3

This page intentionally left blank.



Course Roadmap

Day 1: ICS Overview

Day 2: Field Devices and Controllers

Day 3: Supervisory Systems

Day 4: Workstations and Servers

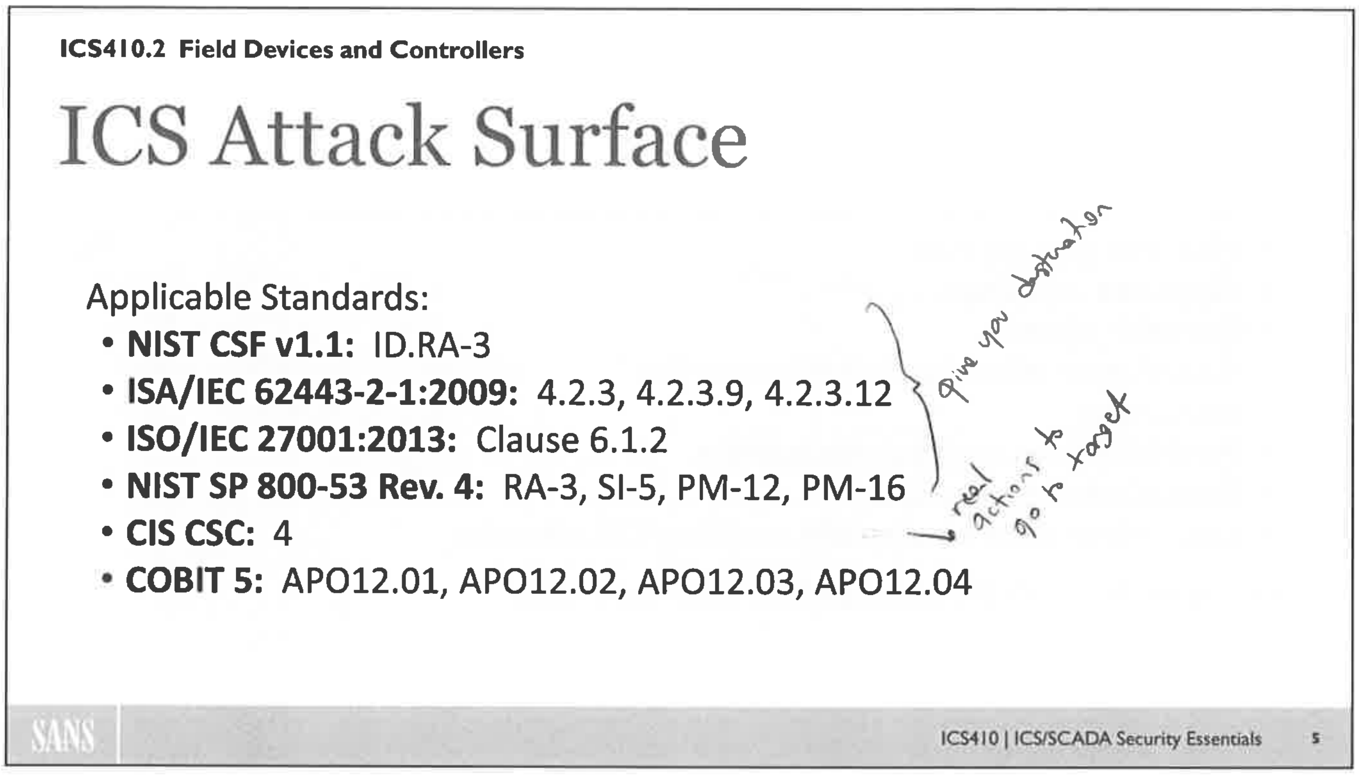
Day 5: ICS Security Governance

SANS

ICS410 | ICS/SCADA Security Essentials 4

This page intentionally left blank.

© 2019, Justin Searle



ICS410.2 Field Devices and Controllers

ICS Attack Surface

Applicable Standards:

- **NIST CSF v1.1:** ID.RA-3
- **ISA/IEC 62443-2-1:2009:** 4.2.3, 4.2.3.9, 4.2.3.12
- **ISO/IEC 27001:2013:** Clause 6.1.2
- **NIST SP 800-53 Rev. 4:** RA-3, SI-5, PM-12, PM-16
- **CIS CSC:** 4
- **COBIT 5:** APO12.01, APO12.02, APO12.03, APO12.04

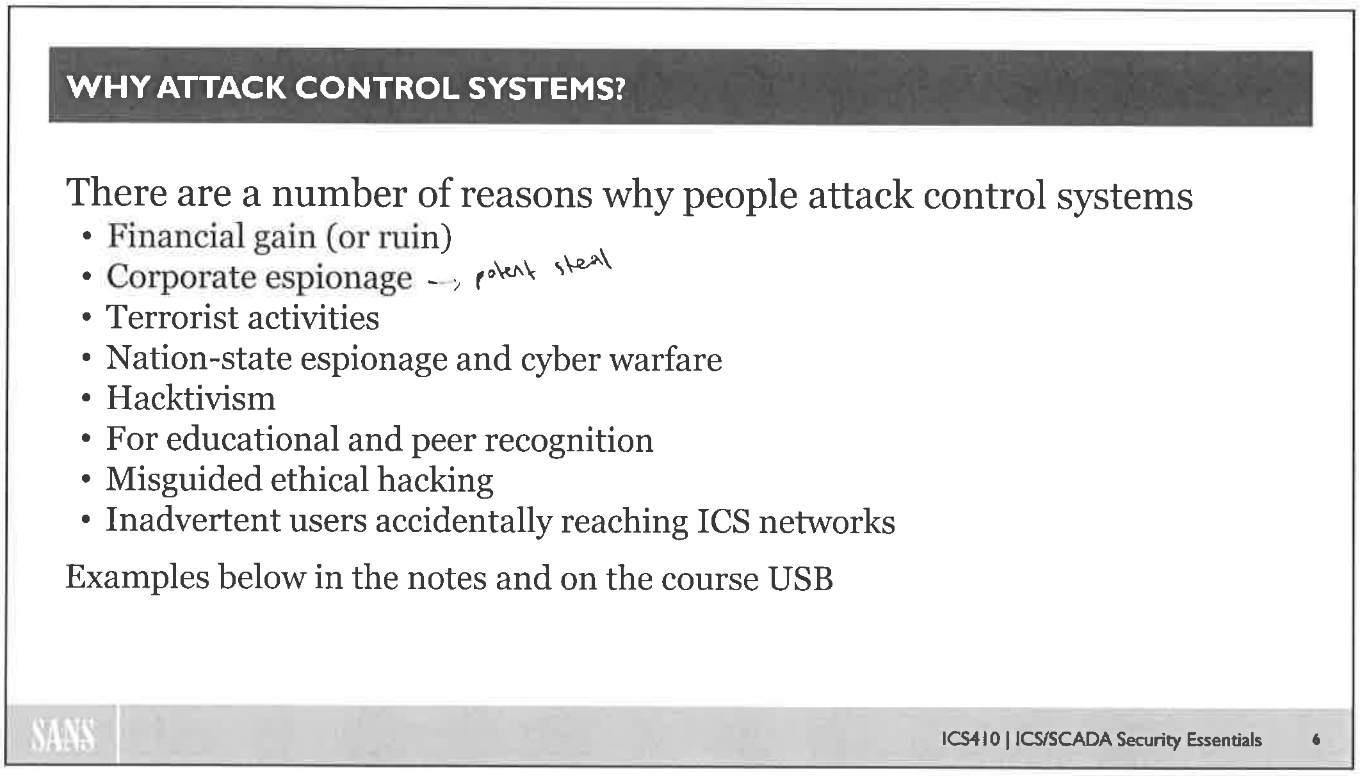
give you destination

real actors to go to target

SANS ICS410 | ICS/SCADA Security Essentials 5

This page intentionally left blank.

1) CSF = Cyber Sec. Framework



WHY ATTACK CONTROL SYSTEMS?

There are a number of reasons why people attack control systems

- Financial gain (or ruin)
- Corporate espionage → patent steal
- Terrorist activities
- Nation-state espionage and cyber warfare
- Hacktivism
- For educational and peer recognition
- Misguided ethical hacking
- Inadvertent users accidentally reaching ICS networks

Examples below in the notes and on the course USB

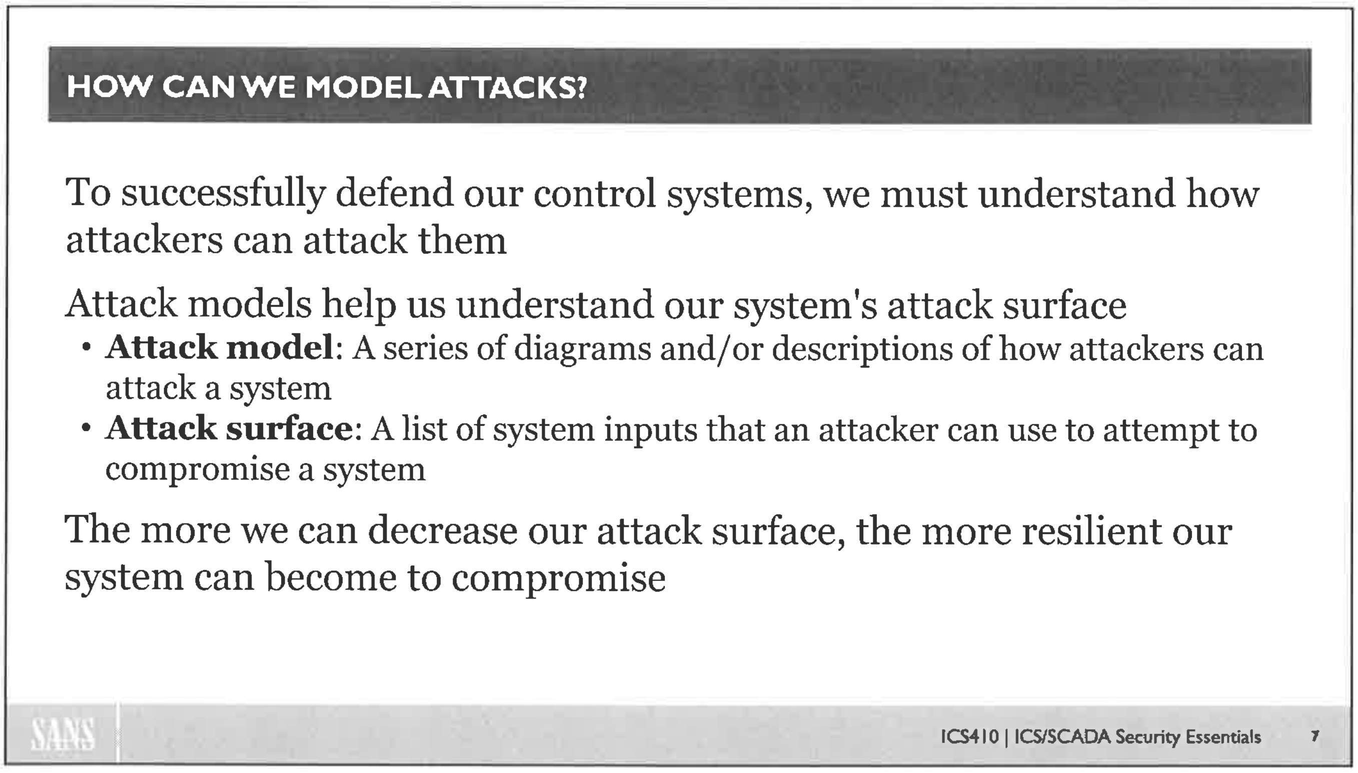
SANS ICS410 | ICS/SCADA Security Essentials 6

In today's age, our industrial control systems make up the majority of most nations' critical infrastructure. Where it isn't critical infrastructure, our industrial control systems often make up the backbone of each nation's manufacturing effort and play a critical part of each nation's gross wealth. Because of this, there are plenty of reasons for malicious actors to attack our industrial control systems. These attacks can be launched by individuals or organizations for the purpose of financial gain (or ruin), corporate espionage, hacktivism, or terrorist activities. Foreign countries are also of grave concern because many nations are now actively involved in cyber espionage and cyber warfare. The last group of people who must be considered are the countless number of individuals who like trying to attack organizations to test themselves, enhance their education, and gain recognition from their peers. This latter group may sometimes cite "ethical hacking" as their goal with claims and sometimes real intent to share vulnerabilities with the compromised company; however, hacking without permission is never ethical.

Examples of Attacks:
https://www.fireeye.com/blog/threat-research/2017/12/attackers-deploy-new-ics-attack-framework-triton.html
https://www.securityweek.com/attackers-alter-water-treatment-systems-utility-hack-report
https://www.securityweek.com/ransomware-attack-disrupts-san-francisco-rail-system
https://www.bbc.co.uk/news/technology-30575104

Also check out the white papers and defense use cases on your ICS410 USB drive

© 2019, Justin Searle



HOW CAN WE MODEL ATTACKS?

To successfully defend our control systems, we must understand how attackers can attack them

Attack models help us understand our system's attack surface

- **Attack model**: A series of diagrams and/or descriptions of how attackers can attack a system
- **Attack surface**: A list of system inputs that an attacker can use to attempt to compromise a system

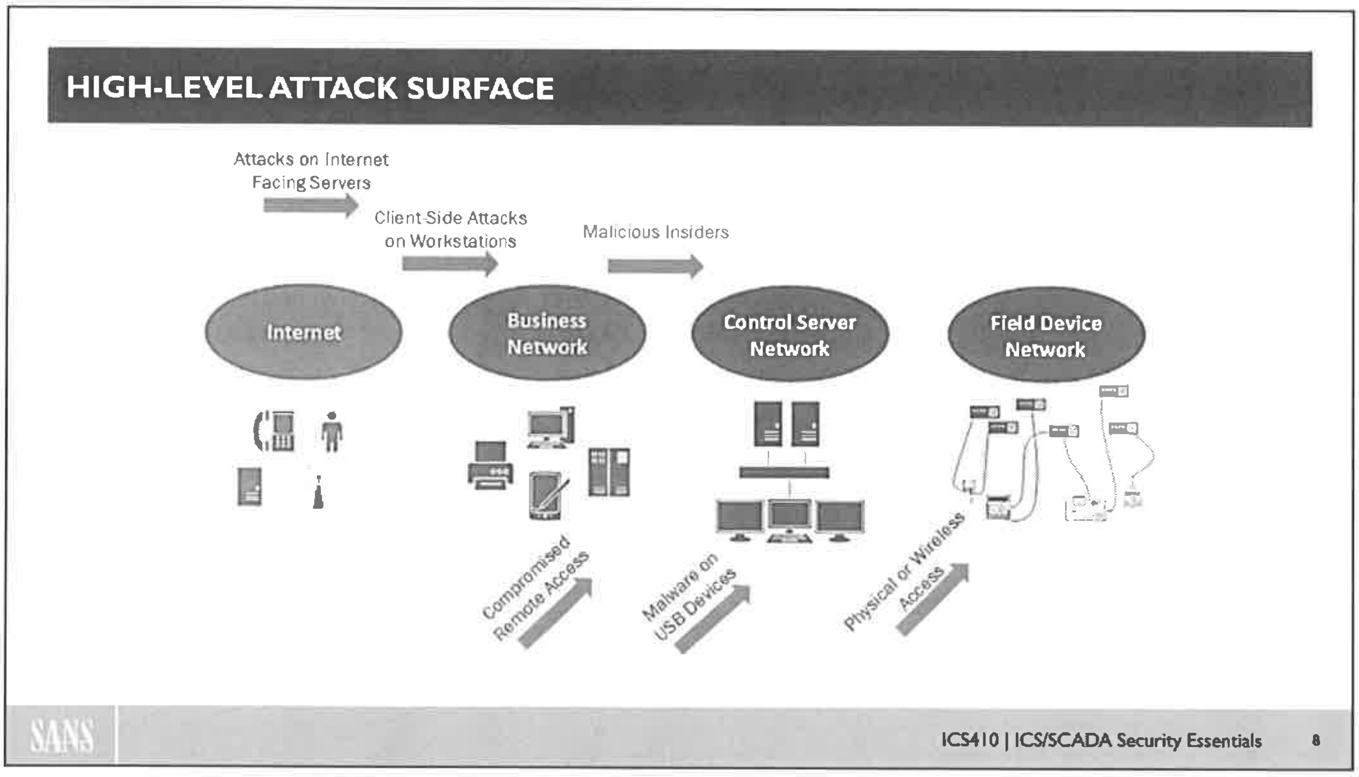
The more we can decrease our attack surface, the more resilient our system can become to compromise

SANS

ICS410 | ICS/SCADA Security Essentials 7

Many industries utilize modeling or simulation tools to study an environment and develop a better understanding. To successfully defend our control systems, we must understand how attackers can attack them. Common approaches utilize attack models in an effort to better understand the system's attack surface.

An *attack model* is a series of diagrams and/or descriptions of how attackers can attack a system through a variety of attack surfaces. *Attack surfaces* are a developed list of system inputs that an attacker can use to attempt to compromise a system. The more we can decrease our attack surface, the more resilient our system can become to compromise. The use of these tools can help us understand system weaknesses and interdependencies, and they can help organizations identify the areas of their systems that may need the most attention.



HIGH-LEVEL ATTACK SURFACE

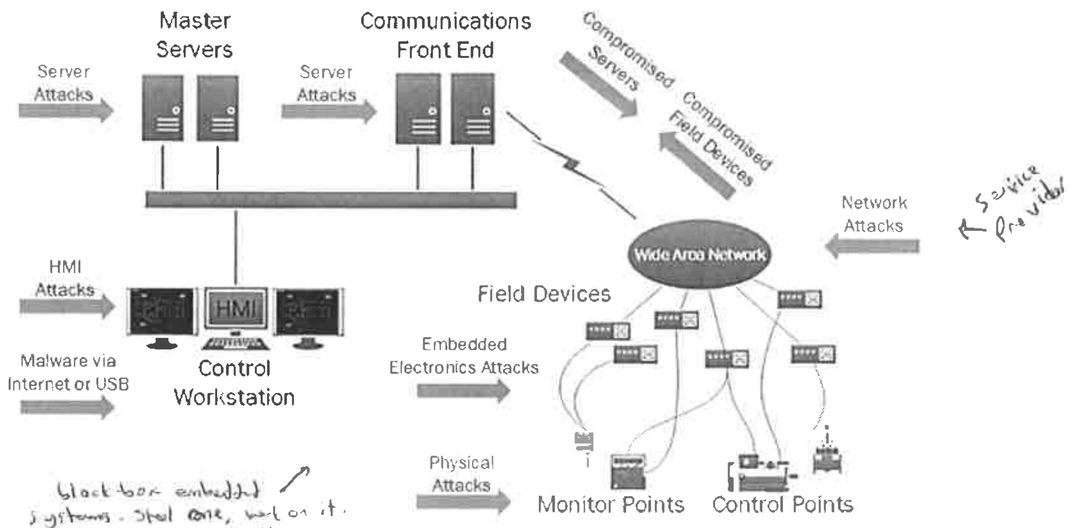
Attacks on Internet Facing Servers
Client-Side Attacks on Workstations
Malicious Insiders
Internet
Business Network
Control Server Network
Field Device Network
Compromised Remote Access
Malware on USB Devices
Physical or Wireless Access
SANS
ICS410 | ICS/SCADA Security Essentials
8

When we discuss an ICS asset owner's attack surface, it is best to start at the high level. Some of the most likely attacks initially come from the internet, either through compromised servers in the company's DMZ or through compromised workstations via client-side attacks. Some of our attacks can originate from inside our companies by malicious insiders or employees who mean to cause harm to the company for a variety of different reasons. Attackers who have come from the internet and have gained a foothold in the corporate network can also be considered malicious insiders, especially considering the research that shows that attackers in our networks aren't discovered for 12–18 months after their initial compromise. Once an attacker is on the business network, it is usually only a matter of time and effort for him to find a way into the control network to compromise servers and to manipulate field devices.

We should also remember that most of our control field networks are also exposed to attackers through a number of different avenues. Attackers can compromise our remote access connections to control networks or have malicious software like Stuxnet carried in on removable storage devices (such as USB sticks or cell phones). Because many of our field networks do not have personnel stationed there, attackers can physically compromise those devices and gain access to the field networks.

© 2019, Justin Searle

CONTROL NETWORK ATTACK SURFACE

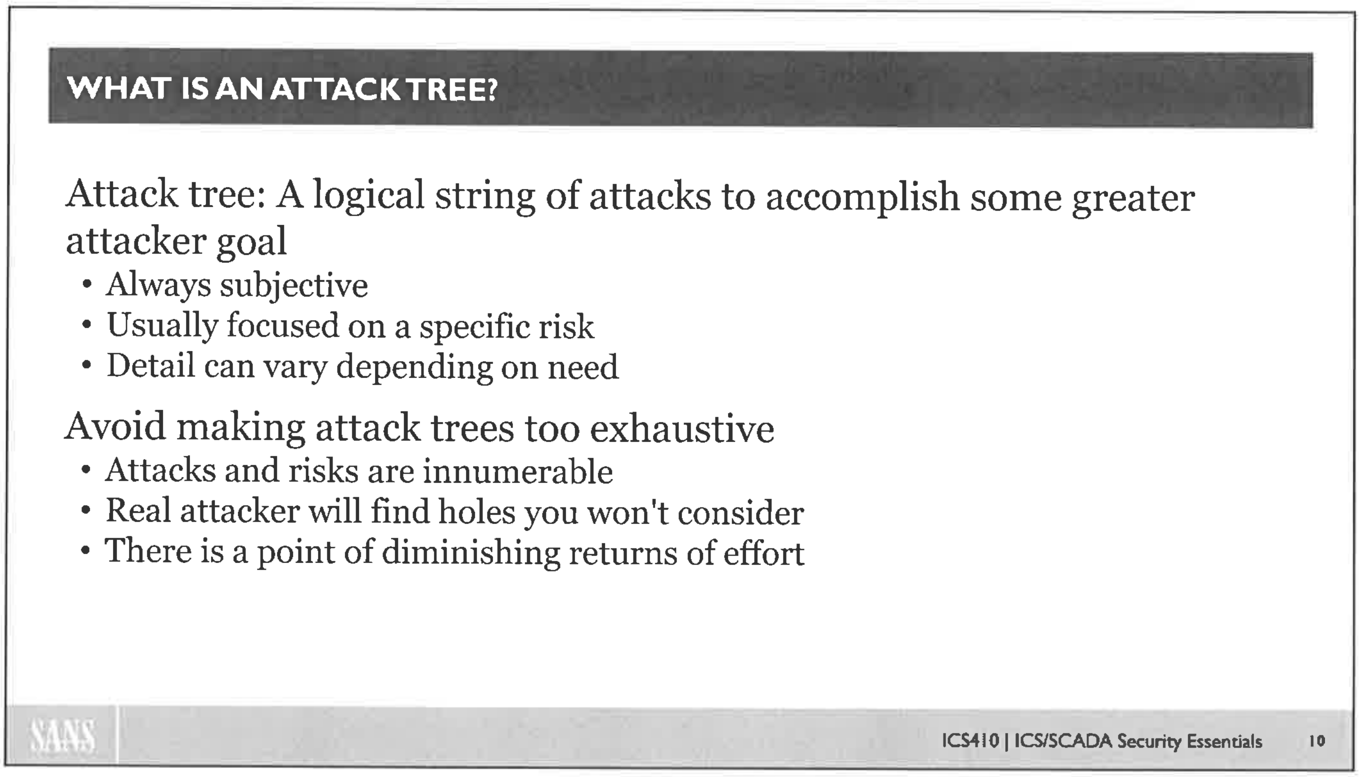


← service provider

black box embedded systems. Steal one, work on it. change code and put it back.

SANS ICS410 | ICS/SCADA Security Essentials 9

Now let's look at the control network itself. Once an attacker has gained a foothold in the control network, he can attempt to compromise any device there. Each device and system in a control network has different technologies that present their own unique attack surfaces. Master servers often run commodity server operating systems, such as Windows, Unix, and Linux, all of which have a large number of known exploits available that attackers can use if those servers aren't patched. Many master servers and HMI devices have web-based applications that often contain flaws that attackers can leverage to bypass authentication, run their own code, and even gain system-level access. Almost all devices and systems can communicate on some network medium, be it a serial connection or a TCP/IP connection. The network protocols used to communicate between these devices may not have security controls on them or may have easily defeated controls that can allow attackers to control our systems. And finally, attackers may find physical access to our field devices to either directly manipulate the processes being controlled, or gain access to the embedded electronics in those field devices to defeat any security control implemented on those field devices. From those field networks, attackers can attempt to compromise other field devices on the network or go for the gold by attacking the control servers themselves.



WHAT IS AN ATTACK TREE?

Attack tree: A logical string of attacks to accomplish some greater attacker goal

- Always subjective
- Usually focused on a specific risk
- Detail can vary depending on need

Avoid making attack trees too exhaustive

- Attacks and risks are innumerable
- Real attacker will find holes you won't consider
- There is a point of diminishing returns of effort

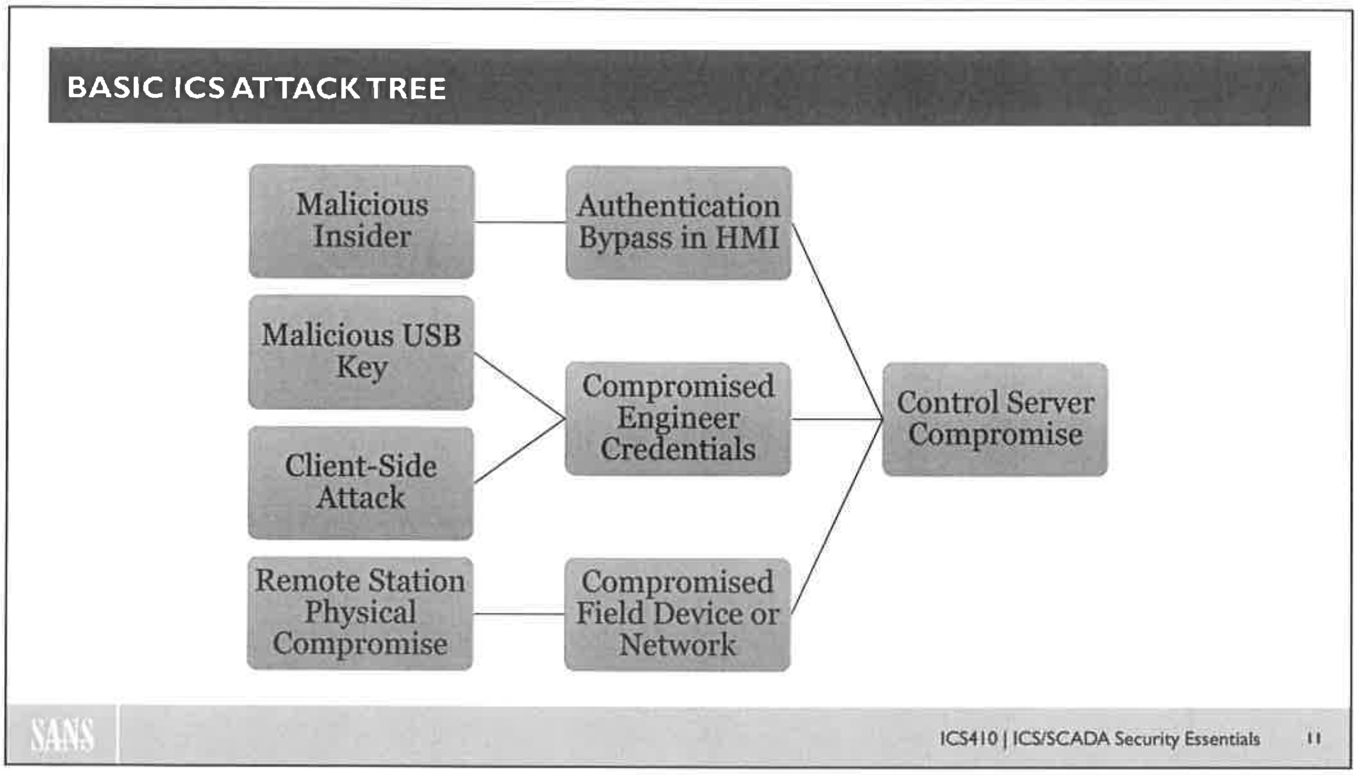
SANS ICS410 | ICS/SCADA Security Essentials 10

An attack tree is a logical way to string multiple attacks together to accomplish some greater attacker goal. This helps us understand how an attacker can use multiple vulnerabilities together to cause greater harm to the organization or its control systems. We can use attack trees to help us understand where we may need additional controls or to justify the need to address a specific weakness.

Because we are hypothesizing what an attacker could do once inside our network, attack trees are always subjective based on the individuals involved in creating them and usually focused on a specific risk. They can be built from the top down (starting with the ultimate end goal of the attacker) or from the bottom up (starting with the initial footholds an attacker obtains). When building attack trees, it's best to keep them simple with minimal steps so you don't get lost in detail or waste time that could be better spent finding and fixing the vulnerabilities.

Efforts to exhaustively model all attacks and all risks often provide diminishing returns because attacks and risks are often innumerable. We can never foresee everything an attacker will do once inside the network; however, we should at least attempt to understand the basic paths the attacker might take.

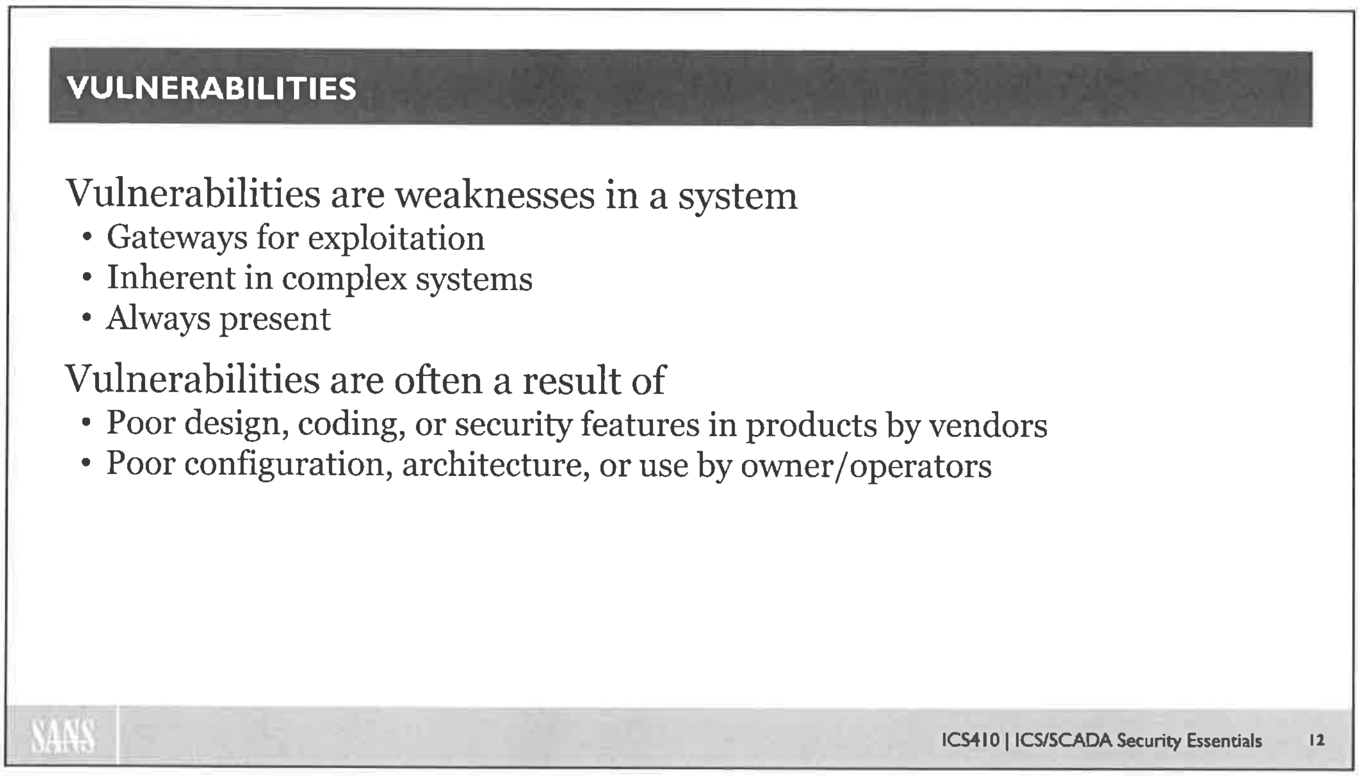
© 2019, Justin Searle



BASIC ICS ATTACK TREE

Malicious Insider
Authentication Bypass in HMI
Malicious USB Key
Compromised Engineer Credentials
Client-Side Attack
Control Server Compromise
Remote Station Physical Compromise
Compromised Field Device or Network

This is an example of a simple attack tree. The attacker's end goal in this tree is the compromise of a control server. To get to this point, we've identified four initial footholds the attacker could gain and at least one intermediate for each foothold before the attacker can gain control of the control server. In real life, this attack tree would show greater detail. However, one will always be able to add intermediate steps and greater detail. The goal is to go deep enough to accomplish your immediate needs for the attack tree and resist the urge to go any deeper. One of the most successful ways to keep the detail in check is to determine a total amount of time to spend building the tree, such as two hours. Stop once you hit that time limit.



VULNERABILITIES

Vulnerabilities are weaknesses in a system
- Gateways for exploitation
- Inherent in complex systems
- Always present

Vulnerabilities are often a result of
- Poor design, coding, or security features in products by vendors
- Poor configuration, architecture, or use by owner/operators

SANS ICS410 | ICS/SCADA Security Essentials 12

Vulnerabilities

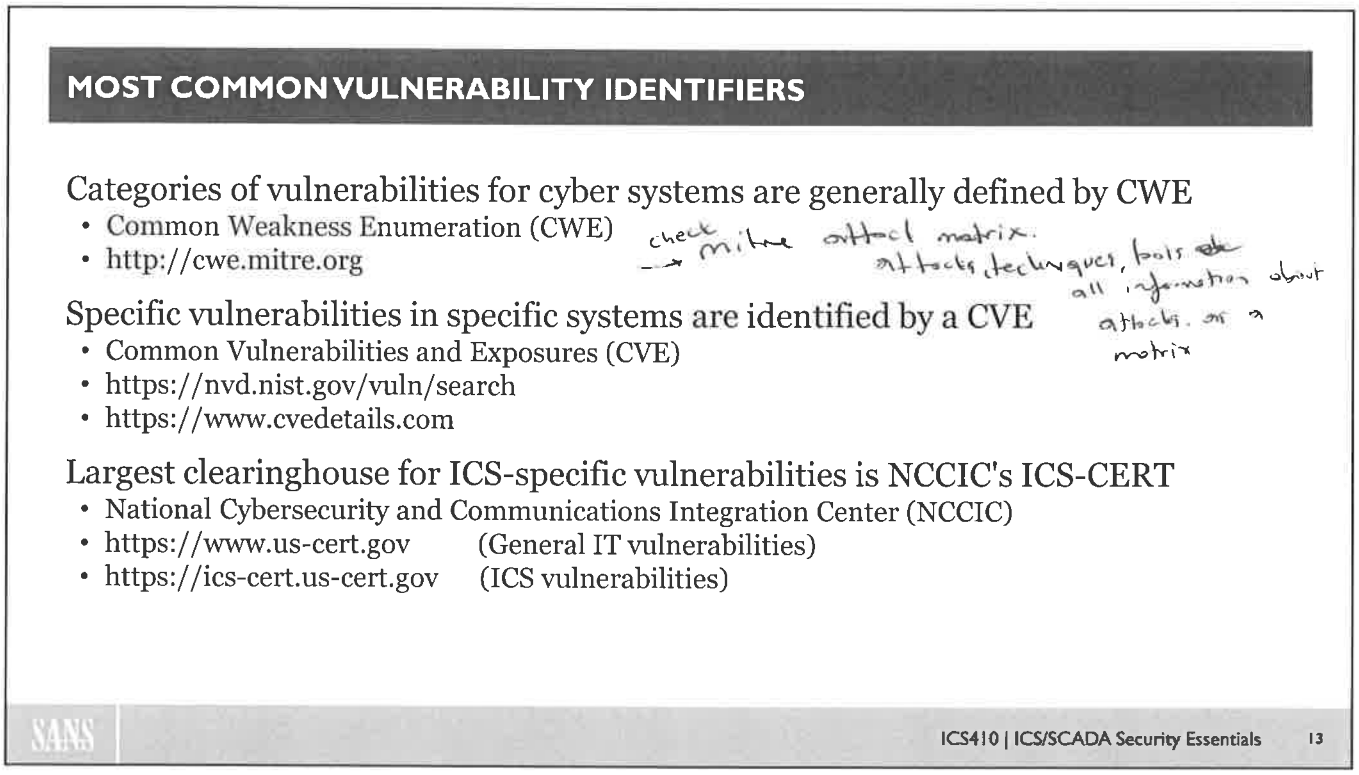
Introduction
In security terms, a *vulnerability* is a weakness in your systems or processes that could be exploited by a threat. However, simply having a vulnerability by itself is not necessarily a bad thing. It is only when the vulnerability is coupled with a threat that the danger starts to set in.

Example
Suppose you like to leave the doors and windows to your house unlocked at night. If you live in the middle of the woods, far away from anyone else, this may not be a bad thing. In this case, the vulnerability of having no locks is present, but there really isn't any threat to take advantage of that vulnerability.

Now suppose you move to a big city full of crime. In fact, this city has the highest burglary rate of any city in the country. If you continue your practice of leaving the doors and windows unlocked, you have exactly the same vulnerability as you had before. However, in the city, the threat is much higher. Thus, your overall danger and risk are much greater. Similar vulnerabilities exist on your computer systems.

Vulnerabilities
Vulnerabilities are the gateways by which threats are manifested. Without vulnerabilities, threats do not pose a risk to the organization. Of course, vulnerabilities do not have to exist solely in *software flaws*. Vulnerabilities can be incorrect configurations, poor physical security, poor hiring practices, etc. When we couple vulnerabilities with threats, we introduce risks to an organization. A zero-day (0-day) attack is an exploit that is not publicly available, and the vendor is usually not aware of the flaw. As you can imagine, these are the most dangerous; however, malware heuristic or behavior analysis of suspicious files and applications can provide some degree of defense.

© 2019, Justin Searle



MOST COMMON VULNERABILITY IDENTIFIERS

Categories of vulnerabilities for cyber systems are generally defined by CWE
- Common Weakness Enumeration (CWE)
- http://cwe.mitre.org

check → mitre attack matrix.
attacks, techniques, tools ~~etc~~
all information about attacks on a matrix

Specific vulnerabilities in specific systems are identified by a CVE
- Common Vulnerabilities and Exposures (CVE)
- https://nvd.nist.gov/vuln/search
- https://www.cvedetails.com

Largest clearinghouse for ICS-specific vulnerabilities is NCCIC's ICS-CERT
- National Cybersecurity and Communications Integration Center (NCCIC)
- https://www.us-cert.gov (General IT vulnerabilities)
- https://ics-cert.us-cert.gov (ICS vulnerabilities)

SANS ICS410 | ICS/SCADA Security Essentials 13

Although traditional security vulnerability lists are still directly applicable to control systems due to the master server's use of operating systems like Windows and Linux, the ICS community has additional vulnerability reporting sites that focus on ICS-specific vulnerabilities.

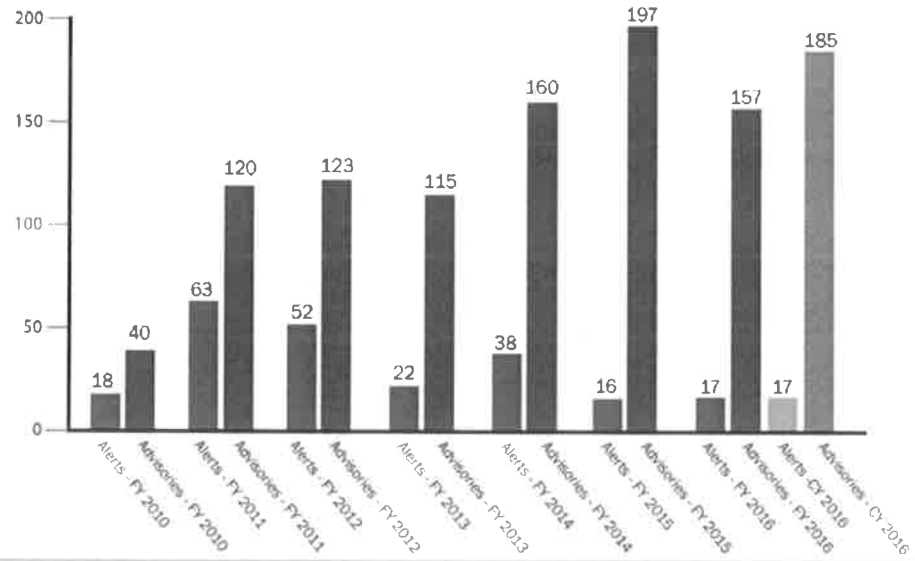
The DHS National Cybersecurity and Communications Integration Center (NCCIC) runs one of the greatest resources for current information. Their ICS-CERT program produces periodic reports on the latest vulnerabilities affecting ICS products and summarizes information about recent ICS security breaches. https://ics-cert.us-cert.gov/

The Common Weakness Enumeration (CWE) is more generally IT-focused. Its database security weaknesses and weakness classifications are useful to the ICS community, and are often referred to by the sources above. CWE's purpose is to provide a common terminology for vulnerabilities, and thus, it doesn't contain vulnerabilities for specific products, but instead, it contains explanations for different vulnerability categories.
http://cwe.mitre.org/data/slices/2000.html

For one of the largest databases for vulnerabilities in specific products, you could consult NIST's National Vulnerability Database (NVD) or the much more user-friendly CVE details that provide an alternate interface to the NVD.

ICS-CERT 2016 – ADVISORIES AND ALERTS ISSUED

CY 2016 based on
255 tickets opened
162 tickets closed
Advisories based on closed tickets...



SANS ICS410 | ICS/SCADA Security Essentials 14

From the 2016 *ICS-CERT Annual Vulnerability Coordination Report*:

"ICS-CERT releases alerts and advisories to notify the ICS community about vulnerabilities that threaten the Nation's Critical Infrastructure (CI). Alerts and advisories provide actionable information about known vulnerabilities, threats, and mitigations. This information helps asset owners and operators understand how attackers might compromise their ICS and how to take action to protect their ICS."

"ICS-CERT alerts provide timely notification to CI owners and operators about publicly known threats that have the potential to affect ICS. ICS-CERT typically releases alerts soon after the identification of publicly available vulnerability information or exploits. Alerts also provide baseline mitigations to reduce the risk of exploitation."

"ICS-CERT advisories provide information about security vulnerabilities in products used in CI and typically contain vendor recommended mitigations or compensating controls."

This table summarizes the number of alerts and advisories for fiscal year (FY) and calendar year (CY) 2016.

Reference:
https://goo.gl/kxKVz6

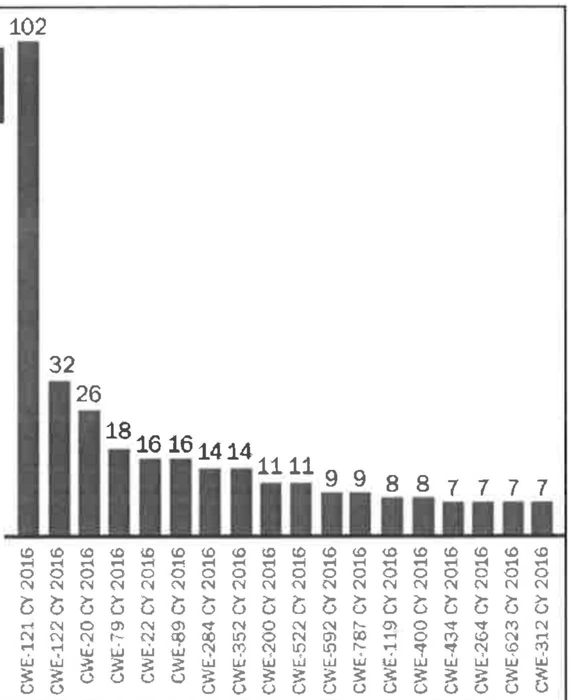
ICS-CERT 2016 – BY CWE

CWE-121: Stack-based Buffer Overflow

CWE-122: Heap-based Buffer Overflow

CWE-20: Improper Input Validation

CWE-79: Cross-site Scripting



From the 2016 *ICS-CERT Annual Vulnerability Coordination Report*:

"ICS-CERT categorizes and assesses the impact of validated vulnerabilities by assigning Common Weakness Enumeration (CWE) numbers and CVSS scores. In the following subsections, we provide the metrics associated with CWE and CVSS score assignments for validated vulnerabilities closed in CY and FY 2016."

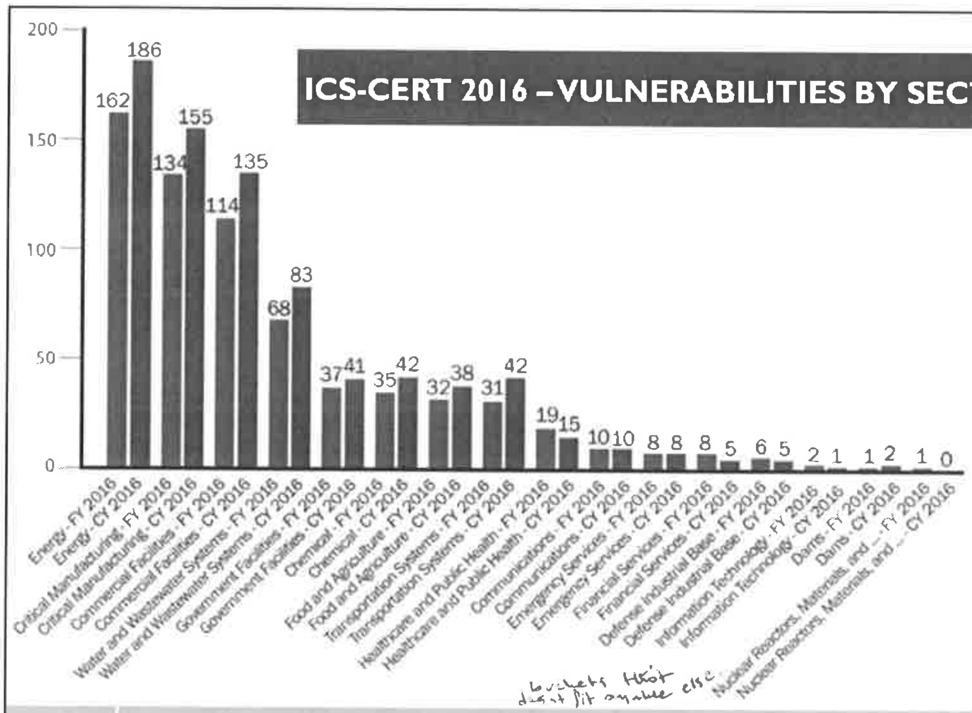
"In FY 2016, ICS-CERT categorized and assigned 70 CWE numbers to 394 validated vulnerabilities. The four most frequently occurring CWEs were CWE-121: Stack-based Buffer Overflow; CWE-20: Improper Input Validation; CWE-79: Cross-site Scripting; and CWE-122: Heap-based Buffer Overflow."

"In CY 2016, ICS-CERT categorized and assigned 85 CWE numbers to 439 validated vulnerabilities. The four most frequently occurring CWEs were CWE-121: Stack-based Buffer Overflow; CWE-122: Heap-based Buffer Overflow; CWE-20: Improper Input Validation; and CWE-79: Cross-site Scripting."

Reference:
https://goo.gl/kxKVz6

Stack base is easier than heap based. Stack based are so many because we know this from early 90s.

ICS-CERT 2016 – VULNERABILITIES BY SECTOR



buckets that don't fit anywhere else

← We are not reporting. That's the reason for numbers

SANS ICS410 | ICS/SCADA Security Essentials 16

This graph shows the number of vulnerabilities coordinated by ICS-CERT by each Critical Infrastructure (CI) sector in which the product is used.

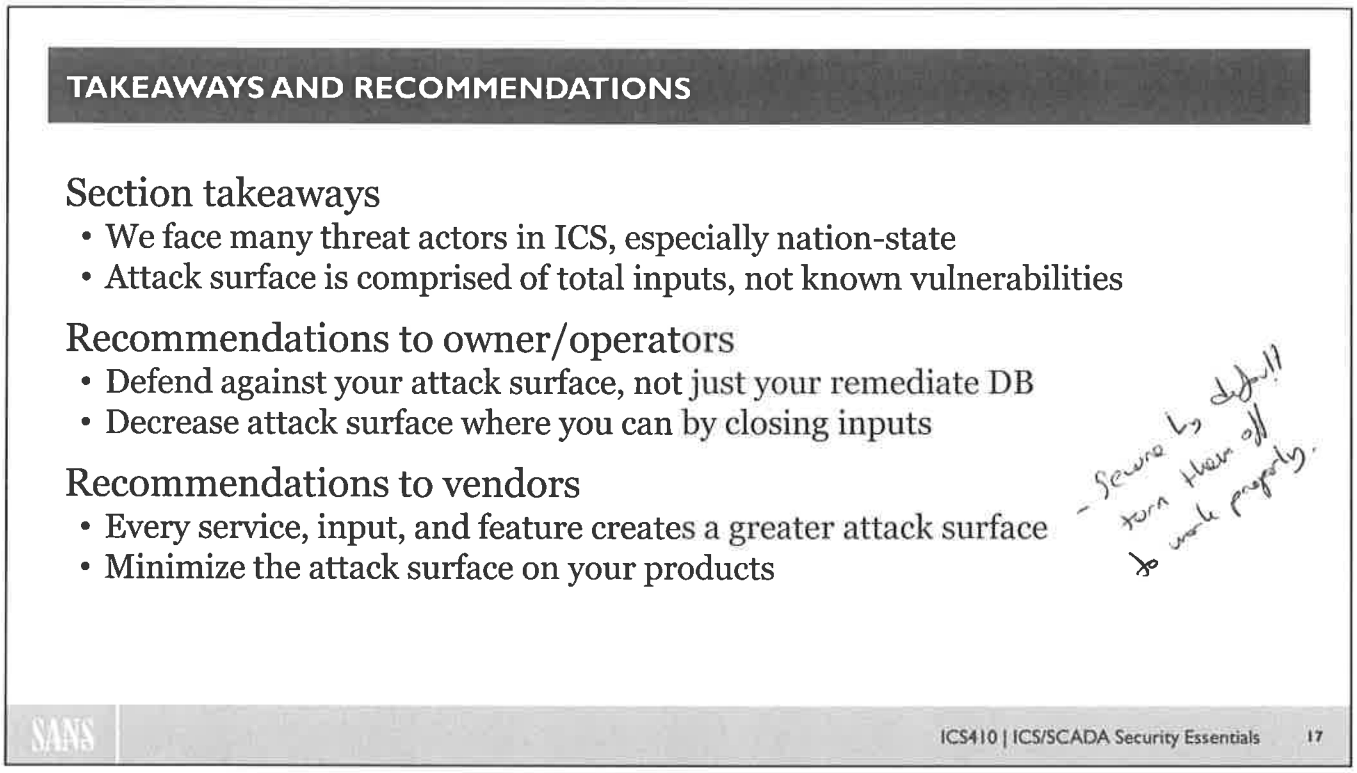
From the 2016 *ICS-CERT Annual Vulnerability Coordination Report*:

"In FY and CY 2016, ICS-CERT released 157 and 185 advisories, respectively, and 17 alerts during both reporting periods. The number of vulnerabilities reported to ICS-CERT in FY and CY 2016 were 2,282 and 2,328 vulnerabilities, respectively. The Vulnerability team performed a more detailed analysis on a subset of the reported vulnerabilities. Cyber-security researchers reported 98.6 percent of the vulnerabilities reported to ICS-CERT and product vendors self-reported the remaining 1.4 percent. The most frequently occurring vulnerability types encountered by ICS-CERT were Stack-based Buffer Overflow, Improper Input Validation, Cross-site Scripting, and Heap-based Buffer Overflow vulnerabilities. The average CVSS score for the vulnerabilities assessed by ICS-CERT was 7.8 out of 10. In FY and CY 2016, 71 and 73.8 percent of the vulnerabilities, respectively, have CVSS scores of seven and above. A CVSS score of seven or above indicates that these vulnerabilities, if exploited, have the potential to have a high or critical impact."

"In FY and CY 2016, ICS-CERT coordinated product vulnerabilities with product vendors that provided product fixes for 363 and 392 vulnerabilities, respectively, which correspond to product fixes for 92.1 percent and 89.3 percent of the vulnerabilities reported to ICS-CERT. The majority of the vulnerabilities coordinated by ICS-CERT in 2016 were most commonly associated with the Energy, Critical Manufacturing, Commercial Facilities, Water and Wastewater Systems Sectors."

Reference:
https://goo.gl/kxKVz6

© 2019, Justin Searle



TAKEAWAYS AND RECOMMENDATIONS

Section takeaways

- We face many threat actors in ICS, especially nation-state
- Attack surface is comprised of total inputs, not known vulnerabilities

Recommendations to owner/operators

- Defend against your attack surface, not just your remediate DB
- Decrease attack surface where you can by closing inputs

Recommendations to vendors

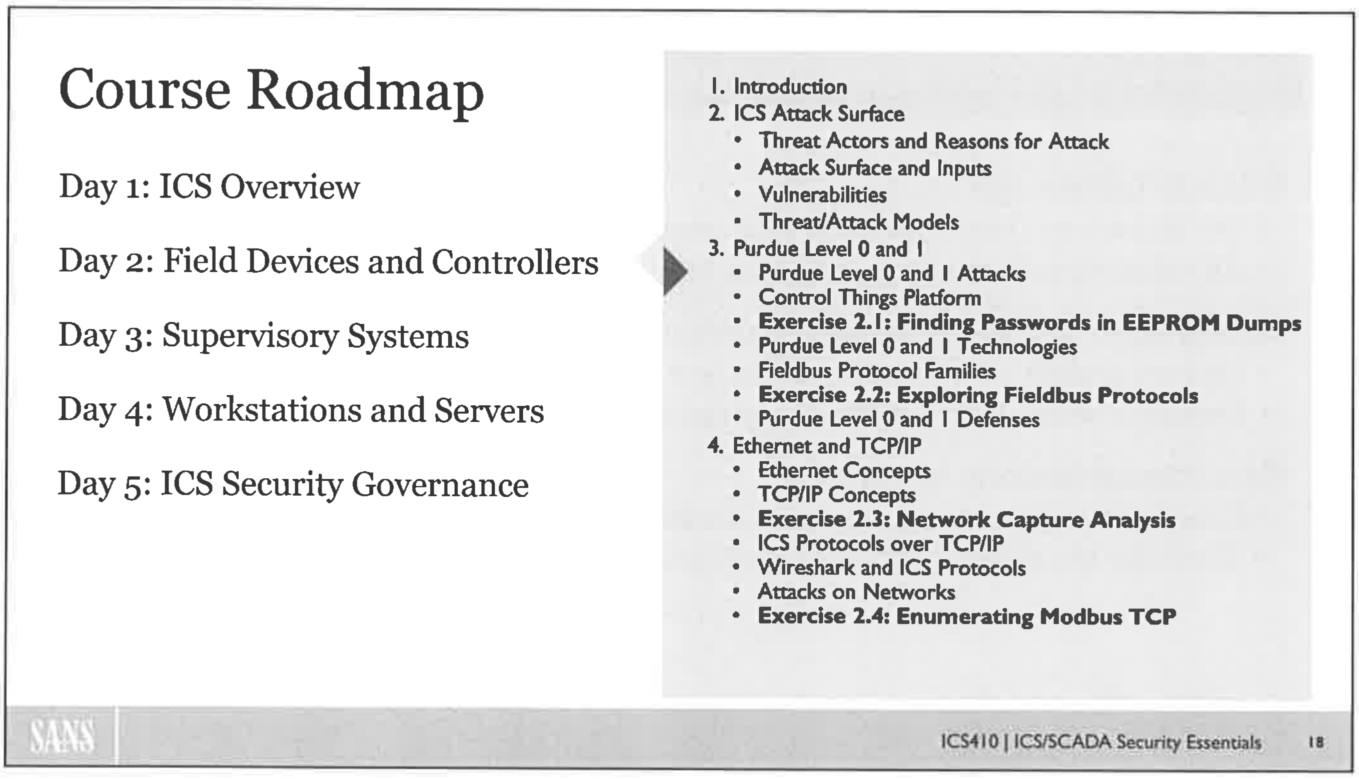
- Every service, input, and feature creates a greater attack surface
- Minimize the attack surface on your products

- Secure by default!!
turn them off
to work properly.

SANS

ICS410 | ICS/SCADA Security Essentials 17

This page intentionally left blank.



Course Roadmap

Day 1: ICS Overview

Day 2: Field Devices and Controllers

Day 3: Supervisory Systems

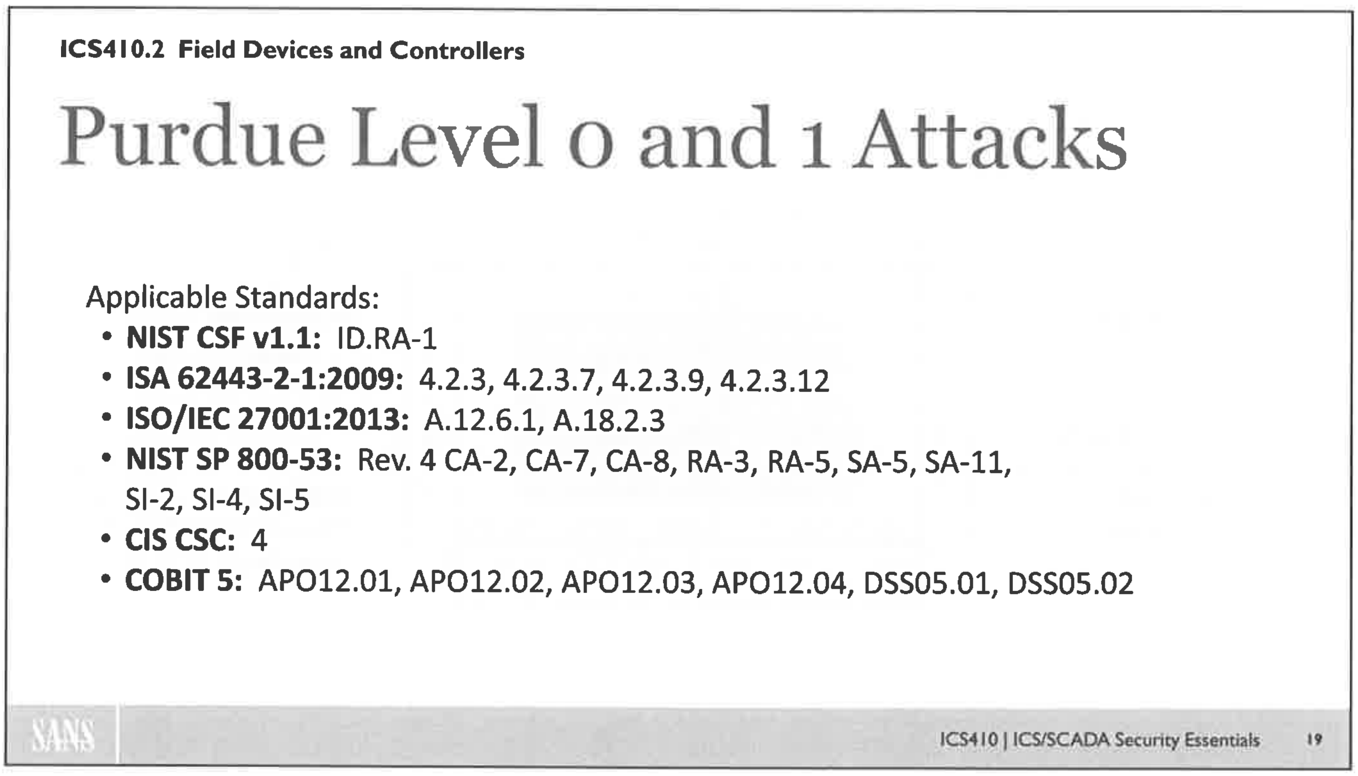
Day 4: Workstations and Servers

Day 5: ICS Security Governance

SANS ICS410 | ICS/SCADA Security Essentials 18

This page intentionally left blank.

© 2019, Justin Searle



ICS410.2 Field Devices and Controllers

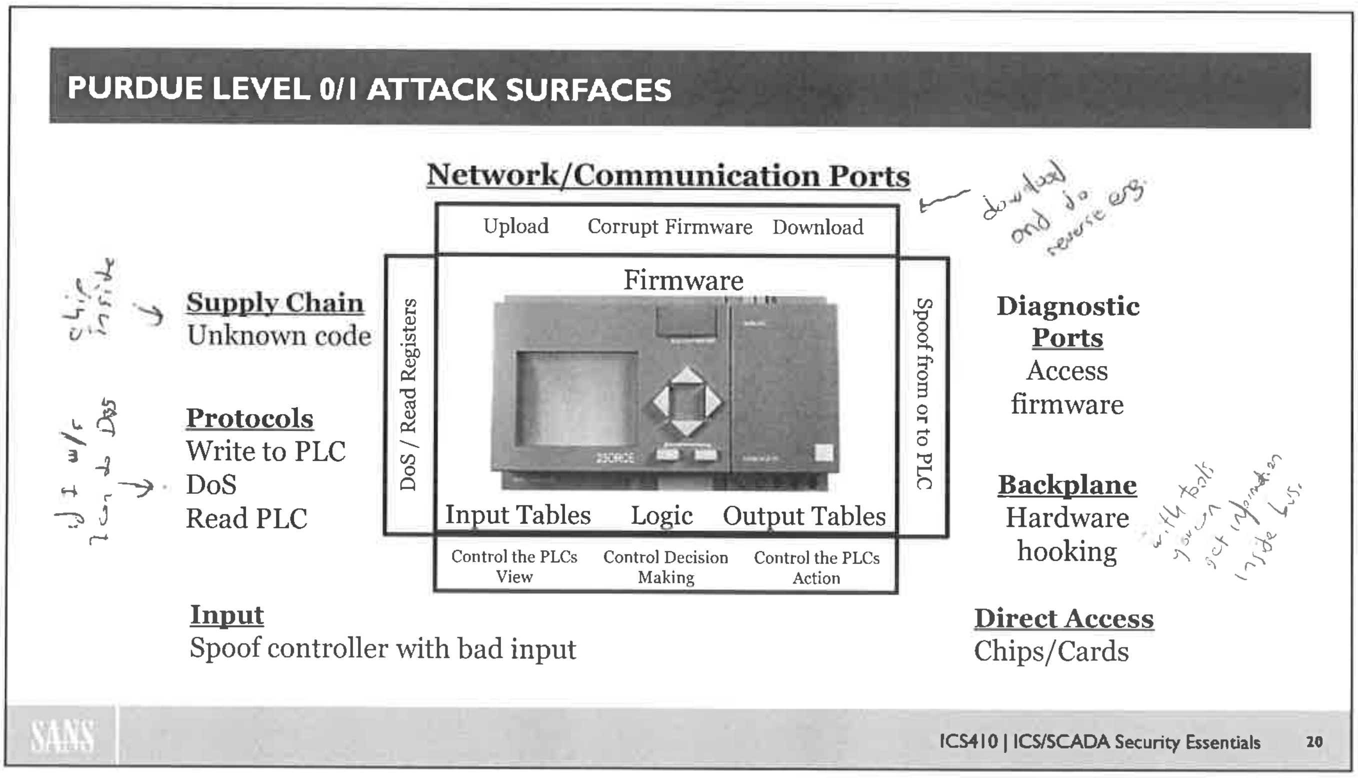
Purdue Level 0 and 1 Attacks

Applicable Standards:

- **NIST CSF v1.1:** ID.RA-1
- **ISA 62443-2-1:2009:** 4.2.3, 4.2.3.7, 4.2.3.9, 4.2.3.12
- **ISO/IEC 27001:2013:** A.12.6.1, A.18.2.3
- **NIST SP 800-53:** Rev. 4 CA-2, CA-7, CA-8, RA-3, RA-5, SA-5, SA-11, SI-2, SI-4, SI-5
- **CIS CSC:** 4
- **COBIT 5:** APO12.01, APO12.02, APO12.03, APO12.04, DSS05.01, DSS05.02

SANS ICS410 | ICS/SCADA Security Essentials 19

This page intentionally left blank.



PURDUE LEVEL 0/1 ATTACK SURFACES

Controllers (PLCs, RTUs, IEDS, and so on) possess numerous attack surfaces. We should consider all the components and major functions of a controller to develop a holistic protection strategy. The controller consists of a central processing unit (CPU), memory (tables and logic), input interfacing, and output interfacing. Memory in the system is generally of two types: ROM and RAM. The ROM contains the program information that allows the CPU to interpret and act on the logic or program stored in the RAM. As the PLC scans and executes logic, it reads and stores values to memory (registers or tables). The values may also be used and referenced by the application program. Newer PLCs are now available with electrically erasable programmable read-only memory (EEPROM), which does not require a battery (to preserve what is stored in the RAM). There are input units and output units and ports to connect to the controller. You will also find a clock, power supply, and typically a battery.

Attack surfaces include:

- Physical components, such as chips, cards, ports, or smart cards (includes communication ports, card slots, and diagnostic ports)—for example, an Ethernet port or an RS-232 serial port. Direct access to the device and its ports provides attackers many options. Attackers can inject malicious code into controllers through physical access of engineering serial ports or by memory update devices such as USB memory sticks, Secure Digital (SD) cards, or CompactFlash (CF) cards. Backdoors and unprotected interfaces (used during development for testing, development, monitoring, or maintenance purposes) are deployed in production equipment. Many devices lack access control and authentication mechanisms to engineering and console ports. Attackers have been able to remove coverings and directly access boards and chips. Physical security protections can be limited or subverted for devices in the field or on the plant floor.

- Firmware/RTOS: Vulnerabilities in the real-time operating system can allow an attacker to access/read data and write to memory. Vulnerabilities can include hardcoded accounts, weak authentication, cleartext FTP and Telnet services, debugging code (assists reverse engineers), and unauthenticated read/writes.

© 2019, Justin Searle

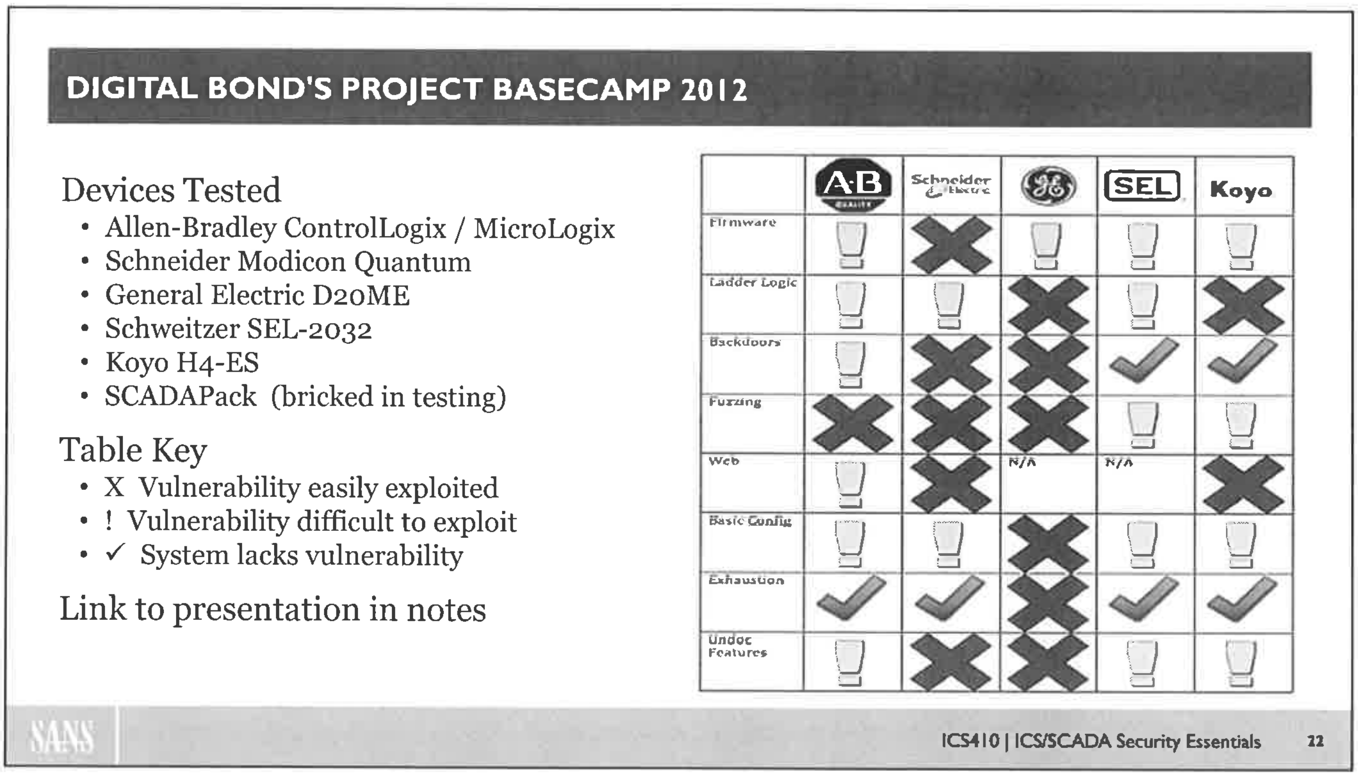
The memory holds the program logic (function blocks, ladder logic, and so on), Input Tables/registers, and Output Tables/registers. Many RTOSs lack software and information integrity mechanisms. Some implementations use an Ethernet Module on the backplane to implement TCP stacks which may rely upon Linux or a custom stack. Lack of adequate equipment disposal may allow an attacker to acquire and reverse engineer equipment.

- Applications (system software) can carry a number of vulnerabilities. Newer PLCs have included web servers to display controller information and provide a management interface. This can add a number of web-based vulnerabilities to the controller.

- Communication includes the protocols used to connect up to the application server, HMIs, and Engineering Workstation, or down to the instruments. Many industrial protocols have few security protections (some have added authentication mechanisms and limited integrity features, such as DNP3v5) and the majority use cleartext communications. Protocol vulnerabilities can be numerous, such as inadequate data source authentication. The control network can be vulnerable to network compromise from an attached business network. Lack of or inadequate network segregation such as using virtual local area networks (VLANs) for security or using the same network services for business operations and control systems can allow attackers access to engineering workstations.

- Supply chain compromises can allow attackers direct access to chips and cards prior to delivery of the controller. This provides an opportunity to add code and undocumented features.

- People. The final attack surface involves the people who interact with the plant floor and field devices. PLC programmers, electricians, field technicians, and process control engineers can put systems at risk through a lack of awareness and poor security behaviors. The insider threat does not need to be a co-opted individual; the more likely is an unwitting person who places an untrusted USB into a device.



DIGITAL BOND'S PROJECT BASECAMP 2012

Devices Tested

- Allen-Bradley ControlLogix / MicroLogix
- Schneider Modicon Quantum
- General Electric D20ME
- Schweitzer SEL-2032
- Koyo H4-ES
- SCADAPack (bricked in testing)

Table Key

- X Vulnerability easily exploited
- ! Vulnerability difficult to exploit
- ✓ System lacks vulnerability

Link to presentation in notes

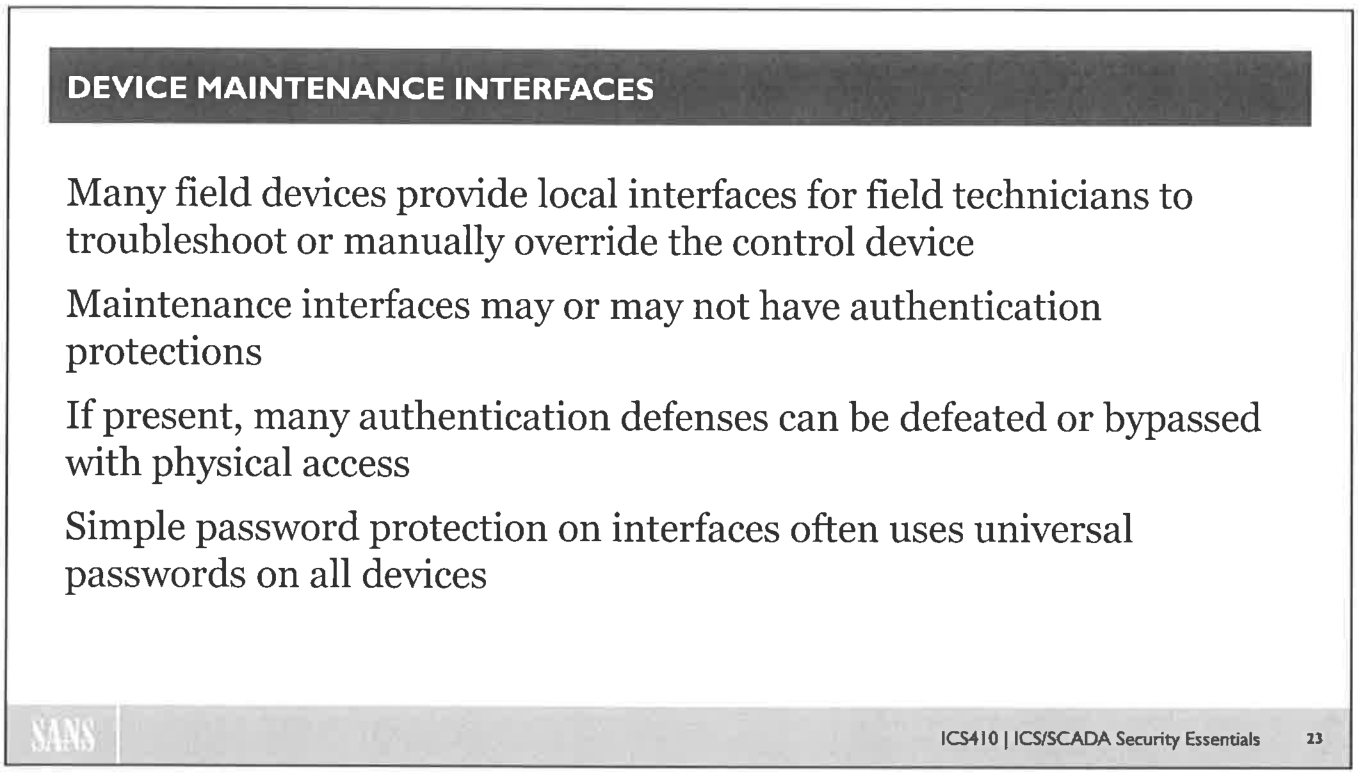
	A·B	Schneider Electric	GE	SEL	Koyo
Firmware	!	X	!	!	!
Ladder Logic	!	!	X	!	X
Backdoors	!	X	X	✓	✓
Fuzzing	X	X	X	!	!
Web	!	X	N/A	N/A	X
Basic Config	!	!	X	!	!
Exhaustion	✓	✓	X	✓	✓
Undoc Features	!	X	X	!	!

SANS ICS410 | ICS/SCADA Security Essentials 22

This research was not exhaustive as the researchers involved worked with available controllers and a firmware/system software version. The results of their "look" provide a good summary of the security vulnerabilities commonly found in controllers. The researchers attempted to discover all the vulnerabilities they could across several areas (firmware, fuzzing, program changes, existing backdoors, web-based vulnerabilities in management interfaces, undocumented features, exhaustion of resources, and configuration weaknesses). Each researcher spent a specific amount of time conducting research. Digital Bond provides a succinct summary: "In the words of Reid Wightman, 'it was a bloodbath.' As everyone expected, the PLCs crashed, had typical vulnerabilities such as overflows and XSS, and had product features that could be used against the device."

Reference:
Full presentation from S4 on YouTube
https://www.youtube.com/watch?v=dtadMIN3CCc

© 2019, Justin Searle



DEVICE MAINTENANCE INTERFACES

Many field devices provide local interfaces for field technicians to troubleshoot or manually override the control device

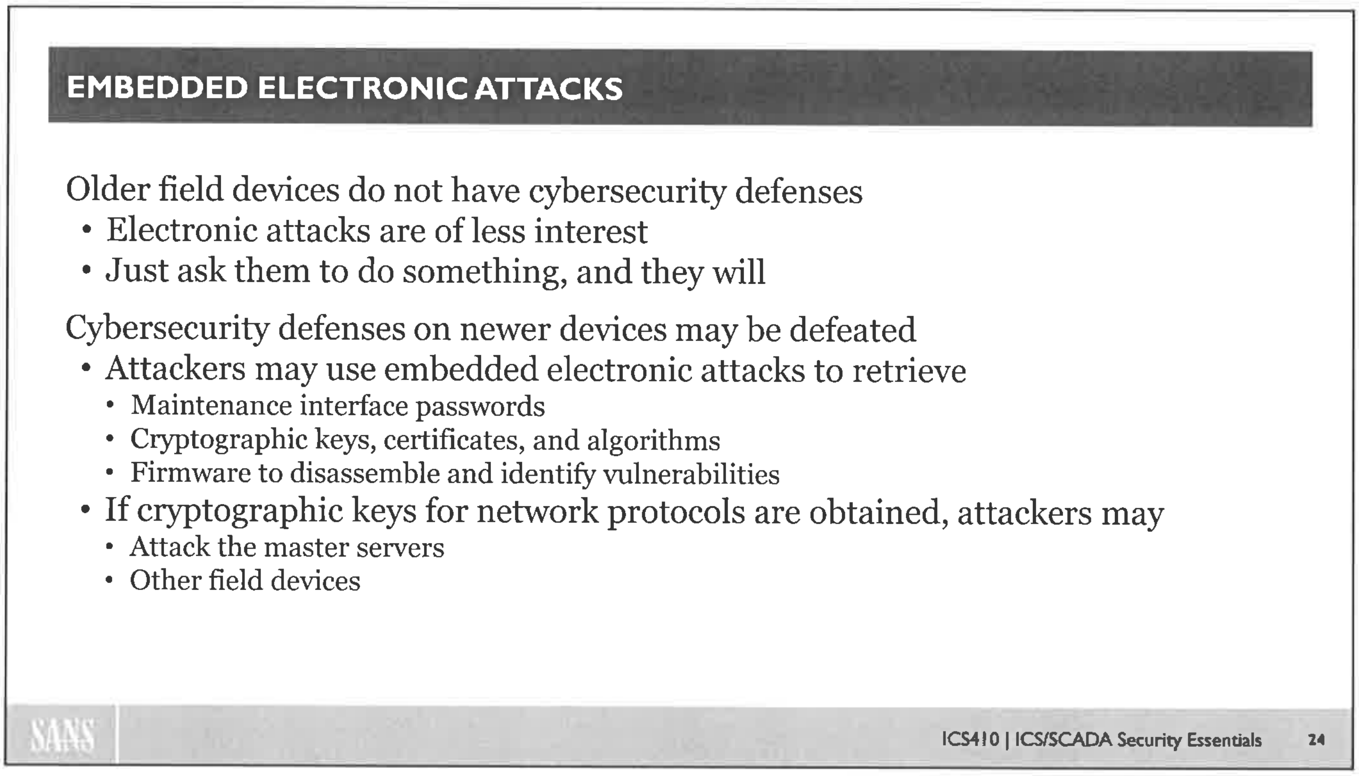
Maintenance interfaces may or may not have authentication protections

If present, many authentication defenses can be defeated or bypassed with physical access

Simple password protection on interfaces often uses universal passwords on all devices

SANS ICS410 | ICS/SCADA Security Essentials 23

Many field devices such as RTUs, PLCs, Relays, and IEDs provide local interfaces for field technicians to troubleshoot or manually override the control device. These maintenance interfaces may or may not have authentication protections. If they don't, then attackers can just manually control the system. However, if authentication protections are present, attackers may be able to defeat or bypass these protections with physical access using the attacks we discussed previously, or using hardware-based attacks, which we discuss in the upcoming slides. Simple password protection on maintenance interfaces often gets set to universal passwords that engineers can use on all devices, making them easier to remember.



EMBEDDED ELECTRONIC ATTACKS

Older field devices do not have cybersecurity defenses
- Electronic attacks are of less interest
- Just ask them to do something, and they will

Cybersecurity defenses on newer devices may be defeated
- Attackers may use embedded electronic attacks to retrieve
  - Maintenance interface passwords
  - Cryptographic keys, certificates, and algorithms
  - Firmware to disassemble and identify vulnerabilities
- If cryptographic keys for network protocols are obtained, attackers may
  - Attack the master servers
  - Other field devices

SANS ICS410 | ICS/SCADA Security Essentials 24

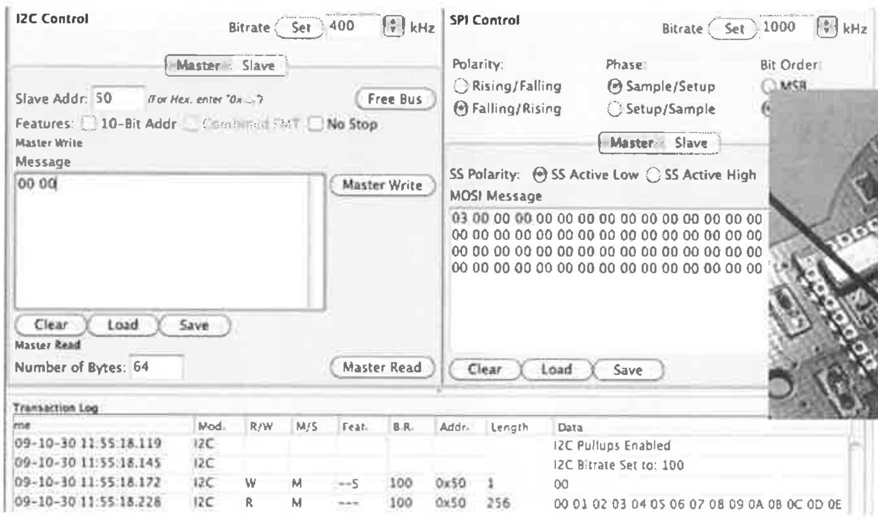
Older field devices that don't have any localized security protections such as passwords or encrypted network communications are a known weakness that has often already been identified. However, newer field devices that have security controls created to protect them from attack become more interesting to us, as those security defenses might be defeated by embedded electronic attacks. These newer devices often have more advanced managing and remote capabilities built into them with the assumption that the security controls make those features safer to use.

Examples of attacks that attackers might perform on these newer devices may be to retrieve cryptographic keys and firmware from the local device storage or memory and use that information to defeat controls and access other systems without the need to open them up. Another example would be an attacker pulling the cryptographic keys for network protocols from the device, which may allow the attacker to use these to generate their own attack traffic to fuzz and exploit the master servers or other field devices.

© 2019, Justin Searle

poisoned cryptographie by firmware...

DUMPING DATA AT REST FROM EEPROMS



Slave Addr: 50 (For Hex. enter "0x...")

Features: ☐ 10-Bit Addr ☐ Combined FMT ☐ No Stop

Message: 00 00

Number of Bytes: 64

Polarity: ○ Rising/Falling [✓ Falling/Rising]

Phase: [✓ Sample/Setup] ○ Setup/Sample

Bit Order: ○ MSB

SS Polarity: [✓ SS Active Low] ○ SS Active High

MOSI Message

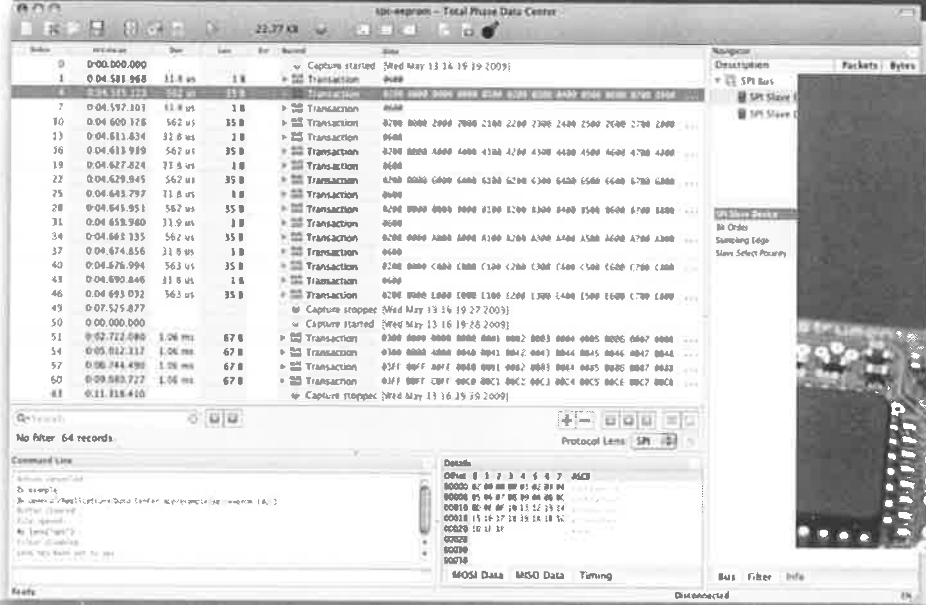
```
03 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

```
me                       Mod.  R/W  M/S  Feat.  B.R.  Addr.  Length  Data
09-10-30 11:55:18.119    I2C                                         I2C Pullups Enabled
09-10-30 11:55:18.145    I2C                                         I2C Bitrate Set to: 100
09-10-30 11:55:18.172    I2C   W    M    --S    100   0x50   1       00
09-10-30 11:55:18.226    I2C   R    M    ---    100   0x50   256     00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E
```

Look at the picture of the electronic circuit board above. Attackers can use special hardware to connect directly to data storage chips on the circuit boards, such as EEPROMs and flash chips, and dump their contents. The application in the screenshot above is an example of a commercial tool from Total Phase for its Aardvark hardware, which does just this. Cheaper solutions are available for attackers to purchase for around $35 and include the Bus Pirate and the GoodFET, both of which can perform these same attacks, but use open source hardware and software to do so.

→ download tool.
run the device.
Capture bus.
take data.
steal

BUS SNOOPING DATA IN MOTION

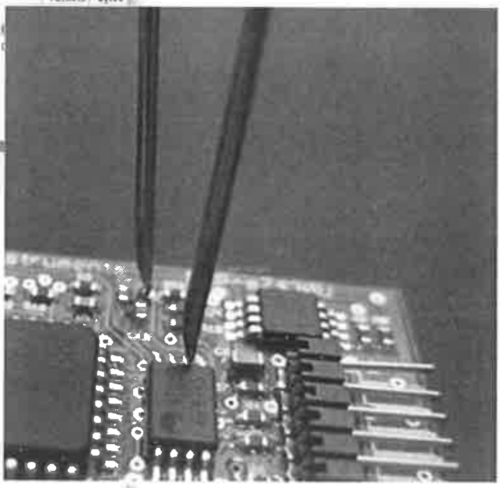


spi-eeprom – Total Phase Data Center

```
 0   0:00.000.000                      Capture started  [Wed May 13 16:19:19 2009]
 1   0:04.581.968   11.8 us   1 B      Transaction      0600
 4   [illegible]    562 us   35 B      Transaction      [illegible]
 7   0:04.597.101   11.8 us   1 B      Transaction      0600
10   0:04.600.126   562 us   35 B      Transaction      0200 0000 2000 2000 2100 2200 2300 2400 2500 2600 2700 2800 ...
13   0:04.611.634   11.8 us   1 B      Transaction      0600
16   0:04.613.939   562 us   35 B      Transaction      0200 0000 4000 4000 4100 4200 4300 4400 4500 4600 4700 4800 ...
19   0:04.627.824   11.8 us   1 B      Transaction      0600
22   0:04.629.945   562 us   35 B      Transaction      0200 0000 6000 6000 6100 6200 6300 6400 6500 6600 6700 6800 ...
25   0:04.643.797   11.8 us   1 B      Transaction      0600
28   0:04.645.951   562 us   35 B      Transaction      0200 0000 8000 8000 8100 8200 8300 8400 8500 8600 8700 8800 ...
31   0:04.659.960   11.9 us   1 B      Transaction      0600
34   0:04.663.135   562 us   35 B      Transaction      0200 0000 A000 A000 A100 A200 A300 A400 A500 A600 A700 A800 ...
37   0:04.674.856   11.8 us   1 B      Transaction      0600
40   0:04.676.994   563 us   35 B      Transaction      0200 0000 C000 C000 C100 C200 C300 C400 C500 C600 C700 C800 ...
43   0:04.690.846   11.8 us   1 B      Transaction      0600
46   0:04.693.032   563 us   35 B      Transaction      0200 0000 E000 E000 E100 E200 E300 E400 E500 E600 E700 E800 ...
49   0:07.525.877                      Capture stopped  [Wed May 13 16:19:27 2009]
50   0:00.000.000                      Capture started  [Wed May 13 16:19:28 2009]
51   [illegible]    1.06 ms  67 B      Transaction      0300 0000 0000 0000 0001 0002 0003 0004 0005 0006 0007 0008 ...
54   [illegible]    1.06 ms  67 B      Transaction      0300 0000 4000 0040 0041 0042 0043 0044 0045 0046 0047 0048 ...
57   [illegible]    1.06 ms  67 B      Transaction      03FF 00FF 80FF 0080 0081 0082 0083 0084 0085 0086 0087 0088 ...
60   [illegible]    1.06 ms  67 B      Transaction      03FF 00FF C0FF 00C0 00C1 00C2 00C3 00C4 00C5 00C6 00C7 00C8 ...
63   0:11.318.410                      Capture stopped  [Wed May 13 16:19:39 2009]
```

No filter 64 records

Protocol Lens: SPI



This circuit board picture is similar in that the attacker is using special hardware to connect to circuit board components; however, this technique is slightly different. Instead of targeting individual chips, the attacker is performing bus sniffing, which is capturing communications between various chips while they are in normal operation. This is a great way for the attacker to capture the firmware as it is transferred from a read-protected flash chip to the microcontroller. Attackers can often gain network cryptographic keys in this way as the microcontroller sends the key to the RF chip or even the frequency hopping algorithm that the microcontroller sends to the RF chip. The application above is Total Phase's bus analysis tool for their Beagle hardware. Total Phase also makes a hardware device called the Aardvark. Once again, the open source community has alternatives for this, which include the Bus Pirate, GoodFET, GreatFET, and others.

Reference:
http://www.totalphase.com/

© 2019, Justin Searle

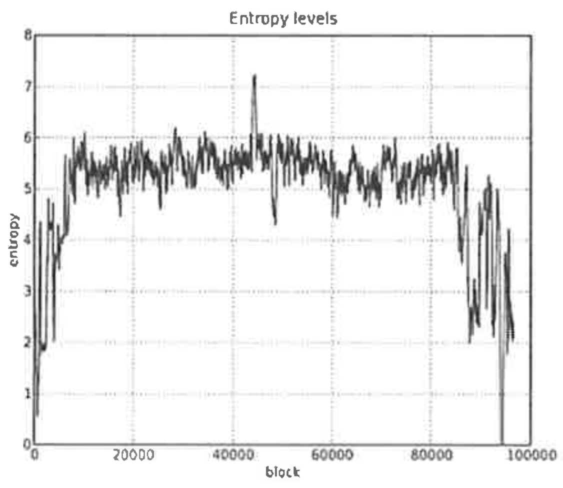
USING ENTROPY TO FIND KEYS

Asymmetric keys have high entropy (very random)

RAM and Flash are filled with non-random data

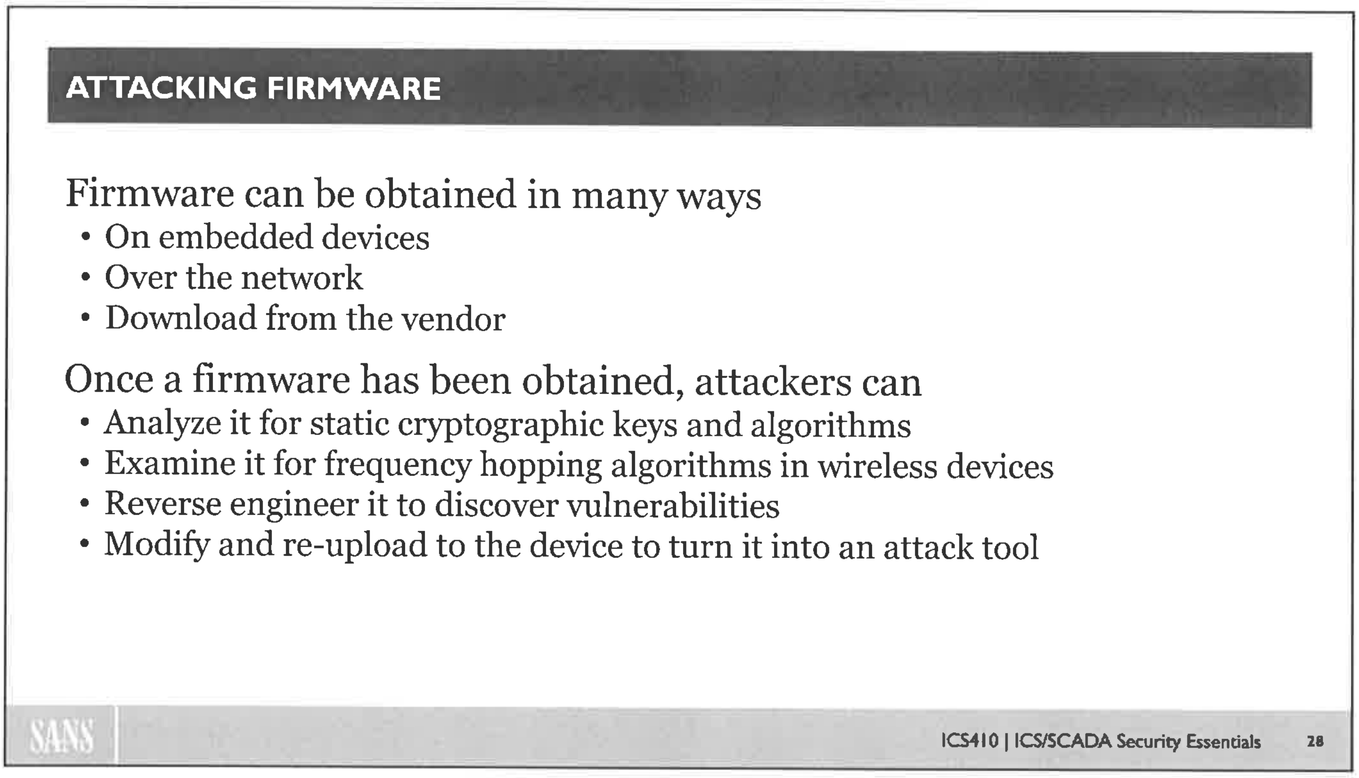
Graphing entropy of Flash reveals a spike in randomness

This spike is the location of the asymmetric key in Flash



Once an attacker has obtained data from a storage device on the circuit board or from sniffing a bus, the attacker needs to analyze that data for what he has found. The first step is usually looking at it manually with hexdump tools like xxd or strings, but the attacker can also do entropy tests on the dataset to identify groups of bytes that appear more random than others. This is possible because asymmetric keys and computer-generated symmetric keys are pseudo-randomly created.

Look at the graph above. The spike you see in the middle is a cryptographic key hidden in a large dataset that was identified using a tool like entropy-graph. Once the attacker knows which bytes have greater levels of entropy, he then needs to figure out exactly where the key begins and ends by determining the key length and educated guesses at where the key begins and ends.


ATTACKING FIRMWARE

Firmware can be obtained in many ways
- On embedded devices
- Over the network
- Download from the vendor

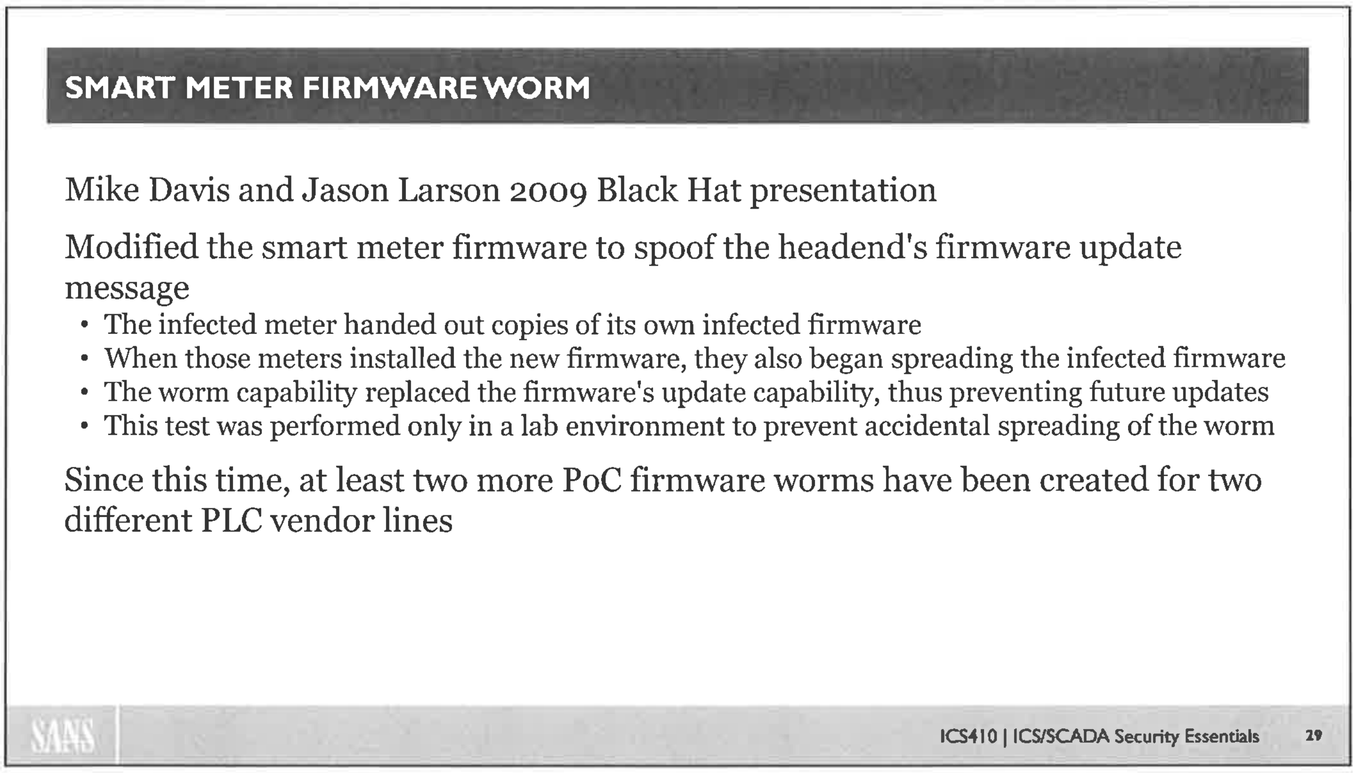
Once a firmware has been obtained, attackers can
- Analyze it for static cryptographic keys and algorithms
- Examine it for frequency hopping algorithms in wireless devices
- Reverse engineer it to discover vulnerabilities
- Modify and re-upload to the device to turn it into an attack tool

SANS ICS410 | ICS/SCADA Security Essentials 28

Firmware may be vulnerable by gaining access and reverse engineering so that an attacker can modify the code or by simply corrupting a firmware load. Attackers can gain access to your firmware by physically connecting to a device, capturing a firmware upload over the network, finding the file on the engineering workstation, or by downloading firmware from the supplier. Once a firmware has been obtained from the device or from the network communications, attackers can analyze it for static cryptographic keys, cryptographic algorithms, or frequency hopping algorithms. Attackers can also reverse engineer it to discover vulnerabilities that are remotely exploitable or attempt to modify the firmware and re-upload it to the device to turn it into an attack tool.

It is important to deny physical access to attackers. Direct access to communication ports or the device can result in modifying logic, writes to memory, retrieving an encryption key, and so on. It is also necessary to keep attackers from network paths as many protocols have weak authentication and are susceptible to injection attacks. We should review the attack surfaces for embedded systems.

© 2019, Justin Searle



SMART METER FIRMWARE WORM

Mike Davis and Jason Larson 2009 Black Hat presentation

Modified the smart meter firmware to spoof the headend's firmware update message

- The infected meter handed out copies of its own infected firmware
- When those meters installed the new firmware, they also began spreading the infected firmware
- The worm capability replaced the firmware's update capability, thus preventing future updates
- This test was performed only in a lab environment to prevent accidental spreading of the worm

Since this time, at least two more PoC firmware worms have been created for two different PLC vendor lines

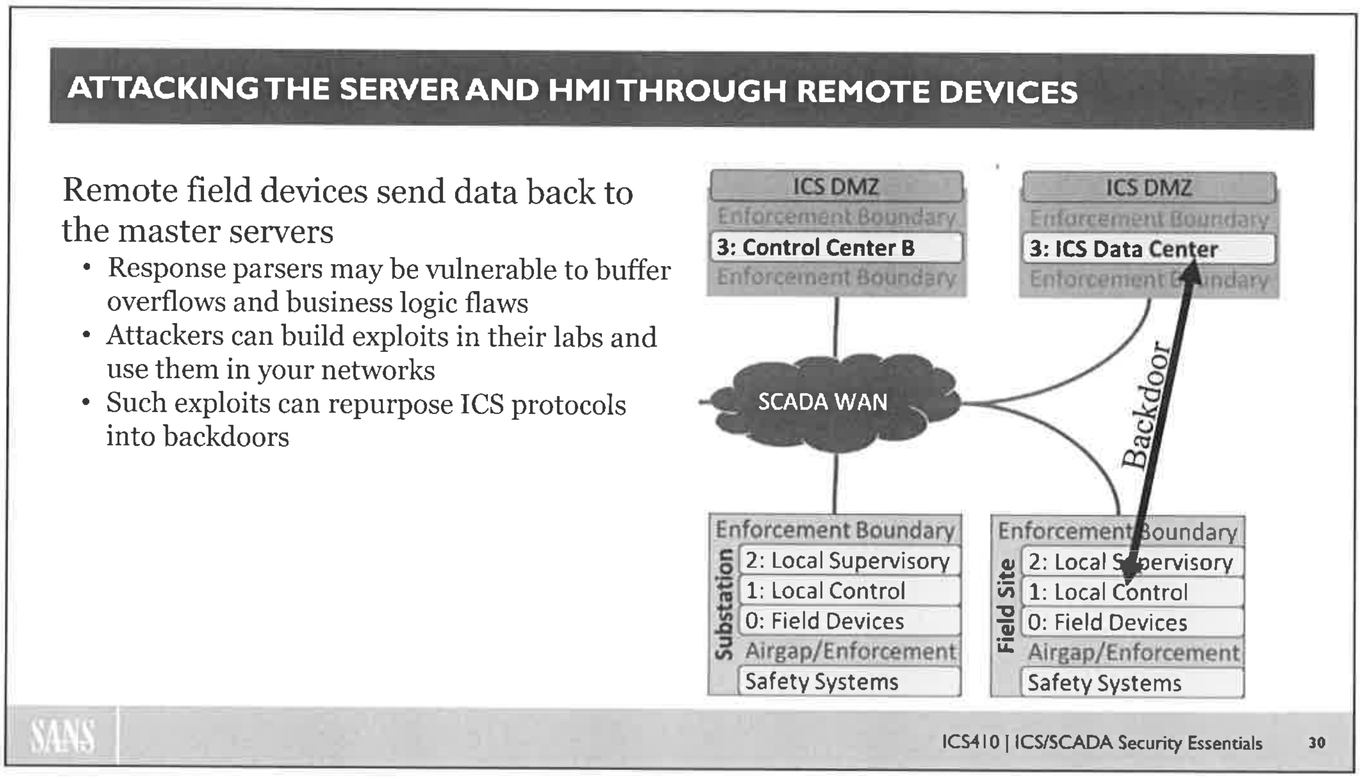
SANS ICS410 | ICS/SCADA Security Essentials 29

An example of firmware modification was demonstrated by Mike Davis and Jason Larson with a proof-of-concept (PoC) firmware worm for smart meters at Black Hat 2009.

After Mike obtained a copy of the firmware from the meter and reverse engineered it, he then modified the smart meter firmware to spoof the headend's firmware update message. He placed this firmware on one of his meters and powered it up. The infected meter followed Mike's custom code and handed out copies of its own infected firmware to the other meters. When those meters saw the update request from what they thought was the headend (but was really Mike's infected meter), they obediently installed the new firmware. After they rebooted, they also began running Mike's custom code by spreading the infected firmware to even more meters.

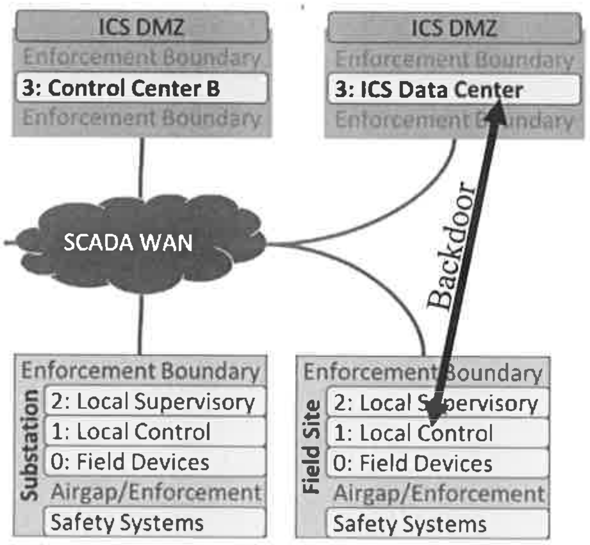
My favorite part about this worm is the fact that Mike needed to remove some of the firmware functionality to make room for his new custom worm-spreading code. The code he chose to remove for this space was the firmware's update capability, thus preventing infected meters from ever receiving future updates, which in the real world would result in a field tech having to deploy and replace each meter by hand. Mike's test was performed only in a lab environment to prevent accidental spreading of the worm, but was an excellent proof of concept to validate the possibility of such a worm.

Reference:
http://www.blackhat.com/presentations/bh-usa-09/MDAVIS/BHUSA09-Davis-AMI-SLIDES.pdf



ATTACKING THE SERVER AND HMI THROUGH REMOTE DEVICES

Remote field devices send data back to the master servers
- Response parsers may be vulnerable to buffer overflows and business logic flaws
- Attackers can build exploits in their labs and use them in your networks
- Such exploits can repurpose ICS protocols into backdoors

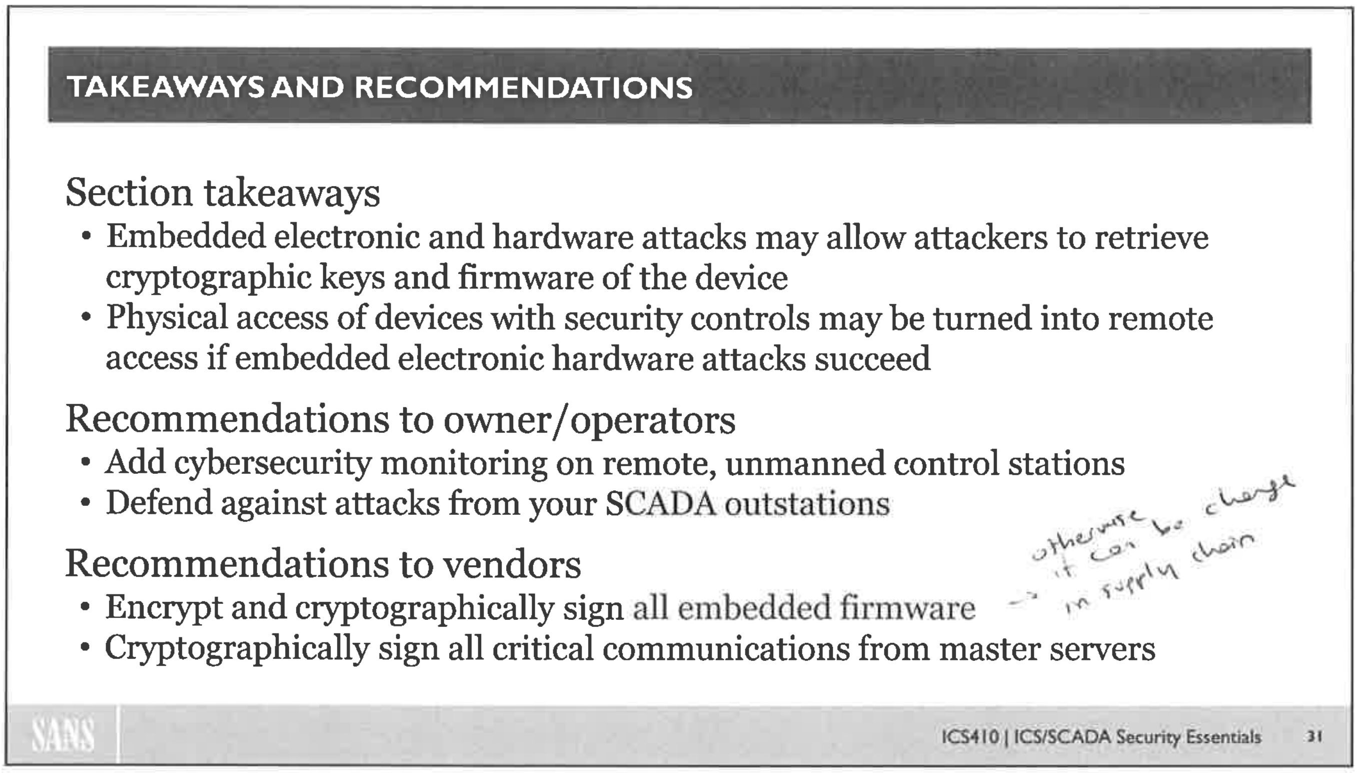


The final attack we wish to mention is the possibility of an attacker attacking the master server from the field. Remember, remote field devices continually send data back to the master servers. These inputs can be susceptible to vulnerabilities such as buffer overflows and business logic flaws, just like any of the network protocol vulnerabilities we previously discussed. Once an attacker has gained a foothold in your field network, attackers can leverage that connectivity to fuzz for vulnerabilities in these inputs and attempt to exploit them. For instance, an attacker in the field may be able to send malicious data to the master server, causing a buffer overflow, and achieve arbitrary remote code execution. These attacks are entirely possible and can be some of the most damaging because, as we discussed at the beginning of the day, compromise of the master server usually means compromise of the whole control system.

Compromising controllers and RTUs can position an attacker to conduct an "upstream" attack to perform a Denial of Service or execute code on the communication gateway or front-end processor (FEP) to gain access to the control network. Even serial communications can be exploited to conduct upstream attacks that can range from knocking over your FEP/communication gateway (highly noticeable) to eloquently allowing the execution of code. A remote device (possibly less protected) can put your larger ICS at risk.

Work with your ICS supplier to select configurations for the FEP/Gateway, which can reduce the potential compromises of an upstream attack.

© 2019, Justin Searle



TAKEAWAYS AND RECOMMENDATIONS

Section takeaways

- Embedded electronic and hardware attacks may allow attackers to retrieve cryptographic keys and firmware of the device
- Physical access of devices with security controls may be turned into remote access if embedded electronic hardware attacks succeed

Recommendations to owner/operators

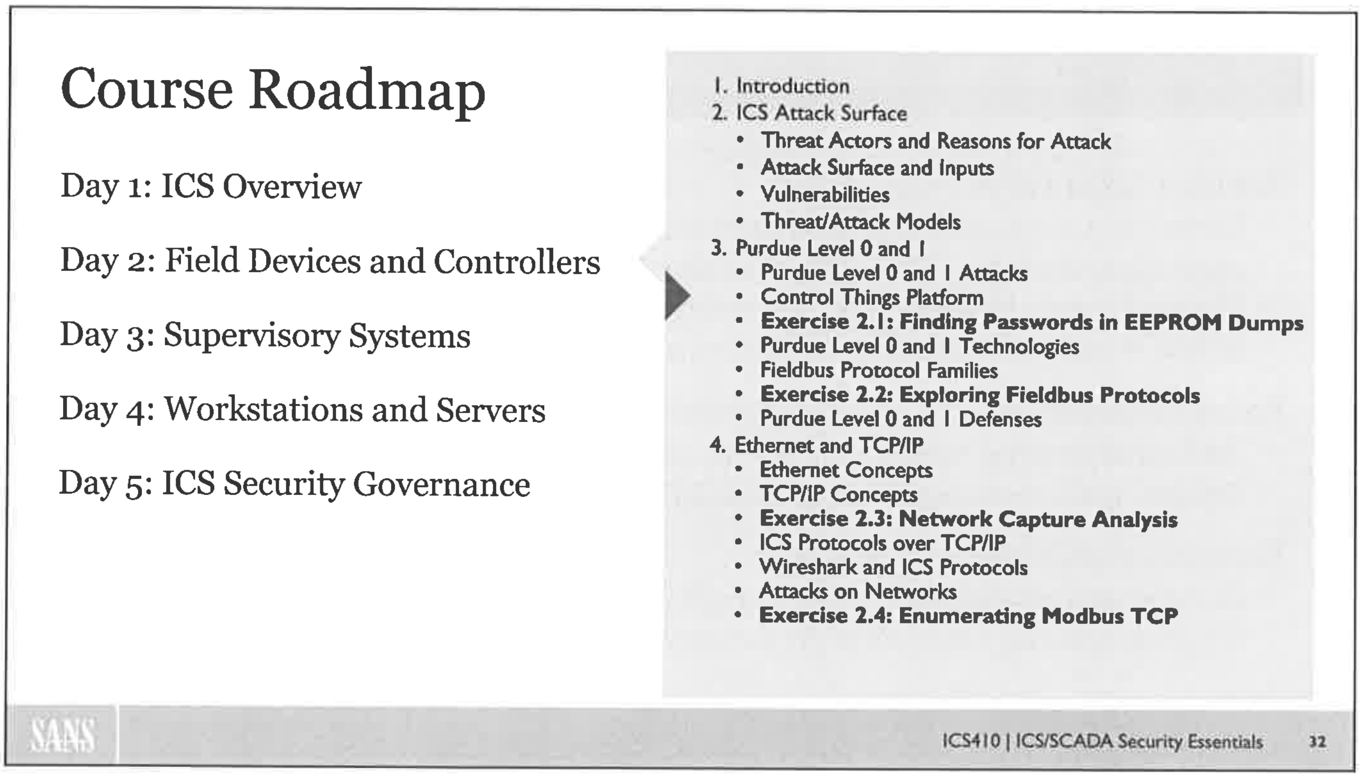
- Add cybersecurity monitoring on remote, unmanned control stations
- Defend against attacks from your SCADA outstations

Recommendations to vendors

- Encrypt and cryptographically sign all embedded firmware → otherwise it can be change in supply chain
- Cryptographically sign all critical communications from master servers

SANS | ICS410 | ICS/SCADA Security Essentials 31

This page intentionally left blank.



Course Roadmap

Day 1: ICS Overview

Day 2: Field Devices and Controllers

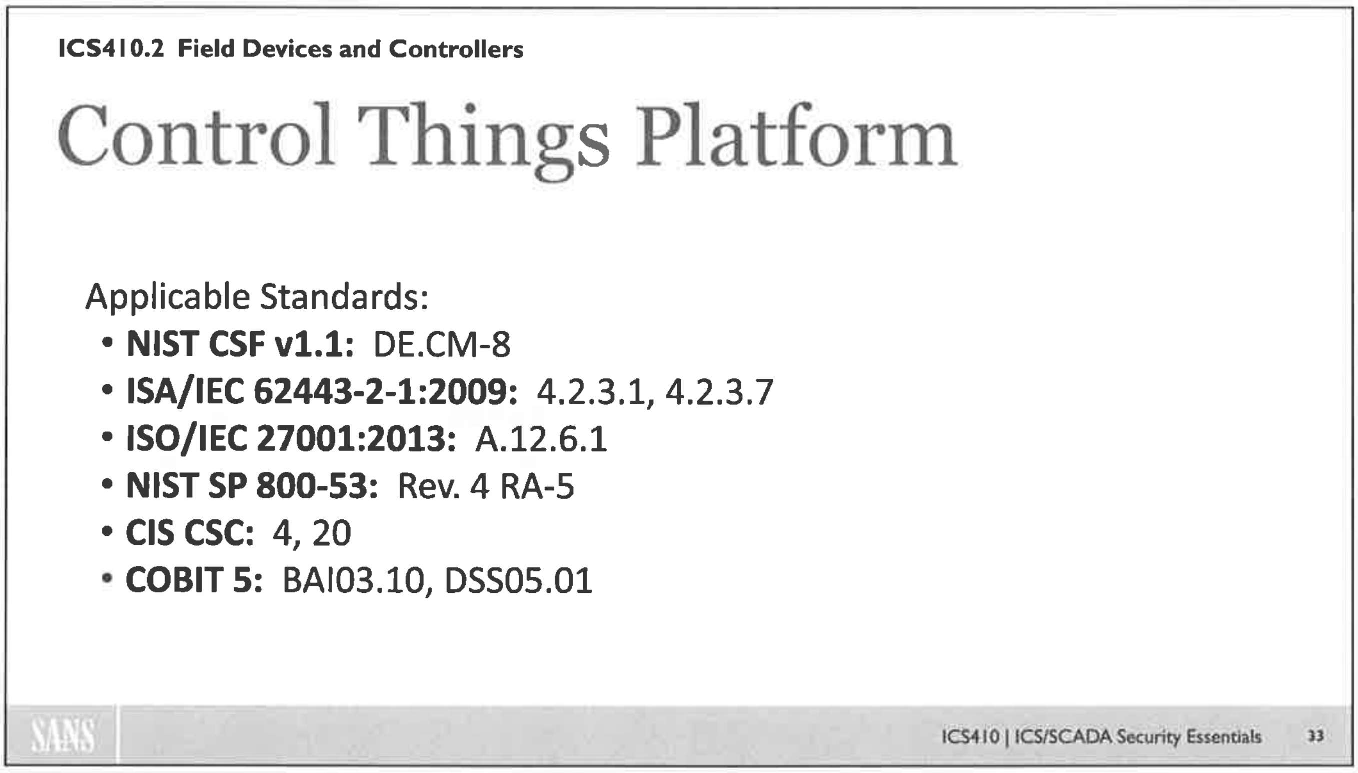
Day 3: Supervisory Systems

Day 4: Workstations and Servers

Day 5: ICS Security Governance

This page intentionally left blank.

© 2019, Justin Searle

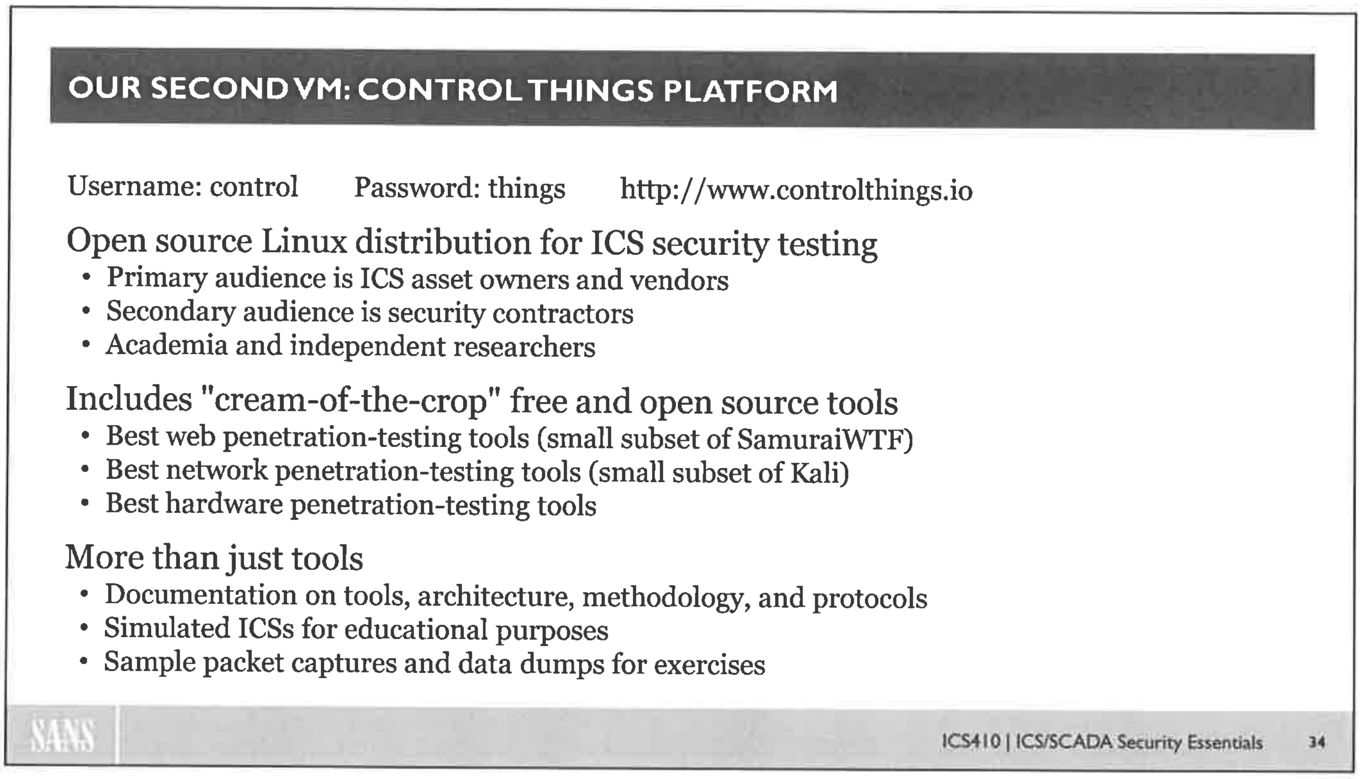


Control Things Platform

Applicable Standards:

- **NIST CSF v1.1:** DE.CM-8
- **ISA/IEC 62443-2-1:2009:** 4.2.3.1, 4.2.3.7
- **ISO/IEC 27001:2013:** A.12.6.1
- **NIST SP 800-53:** Rev. 4 RA-5
- **CIS CSC:** 4, 20
- **COBIT 5:** BAI03.10, DSS05.01

This page intentionally left blank.



OUR SECOND VM: CONTROL THINGS PLATFORM

Username: control Password: things http://www.controlthings.io

Open source Linux distribution for ICS security testing
- Primary audience is ICS asset owners and vendors
- Secondary audience is security contractors
- Academia and independent researchers

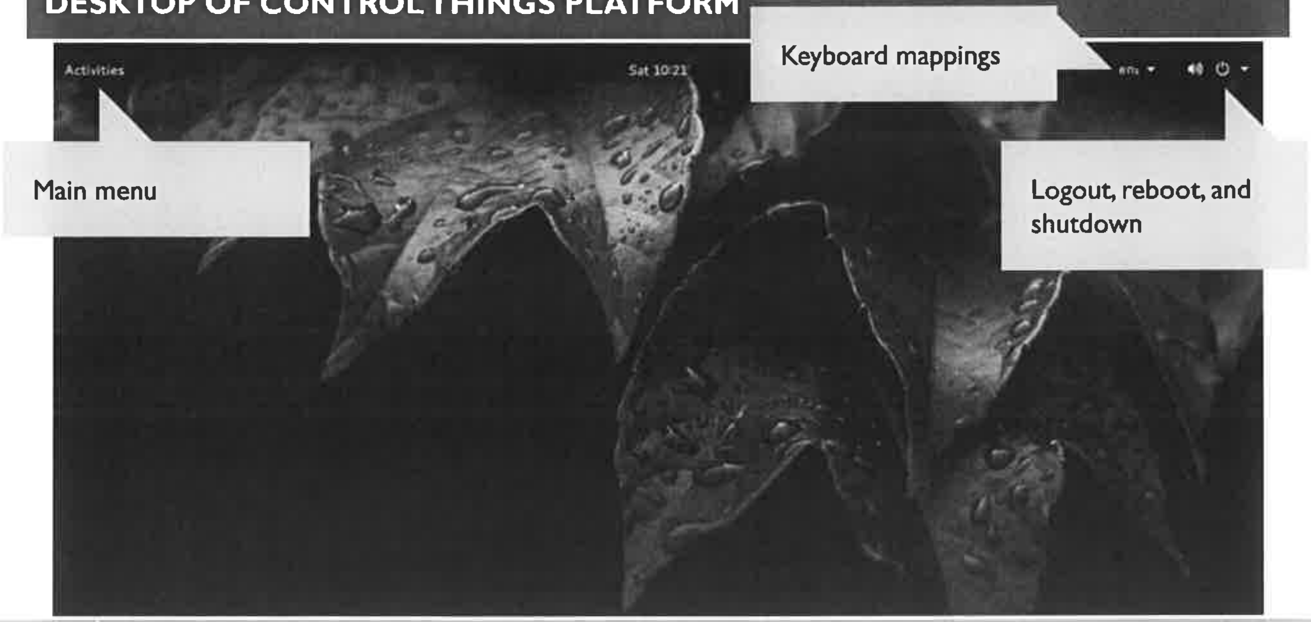
Includes "cream-of-the-crop" free and open source tools
- Best web penetration-testing tools (small subset of SamuraiWTF)
- Best network penetration-testing tools (small subset of Kali)
- Best hardware penetration-testing tools

More than just tools
- Documentation on tools, architecture, methodology, and protocols
- Simulated ICSs for educational purposes
- Sample packet captures and data dumps for exercises

SANS ICS410 | ICS/SCADA Security Essentials 34

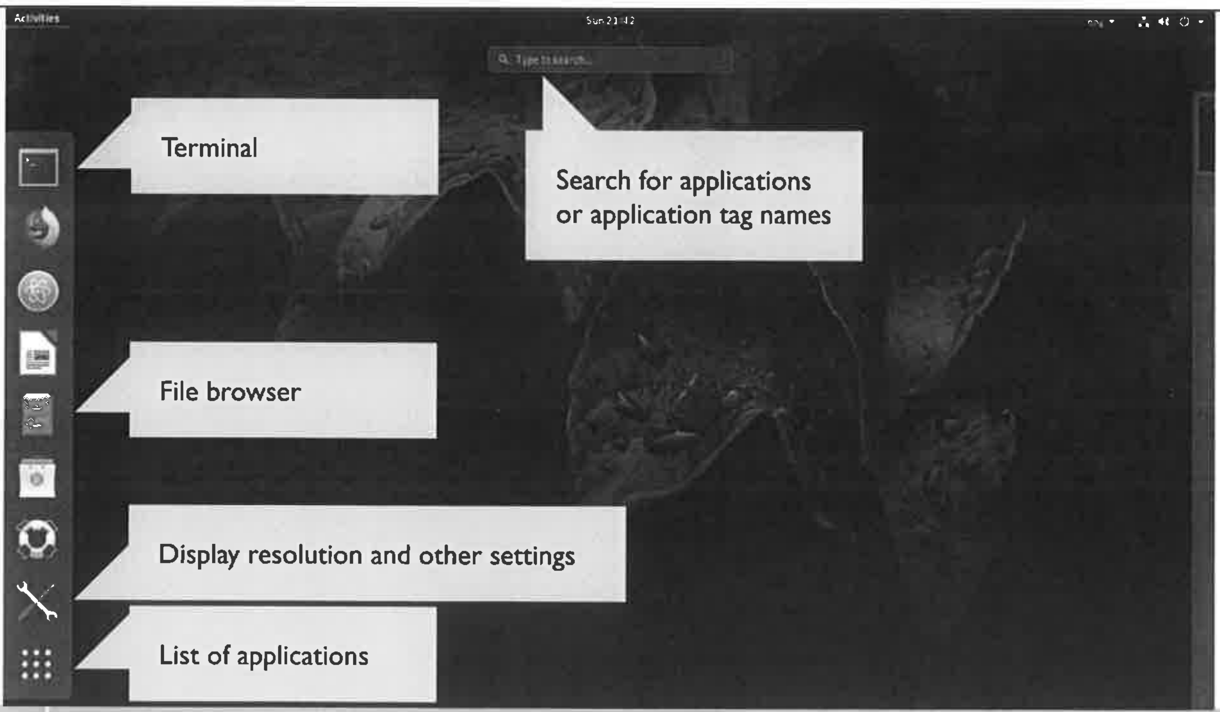
This page intentionally left blank.

DESKTOP OF CONTROL THINGS PLATFORM



This is the home page of the Control Things Platform. There are a lot of security testing tools, resources, and applications hidden in the main menu (click on Activities in the upper left-hand corner). We'll explore some of these in future labs.

If Firefox prompts you to "Reset Firefox to its default state," please do not do so. This will remove all of your plugins needed for testing and to complete future labs in this course.

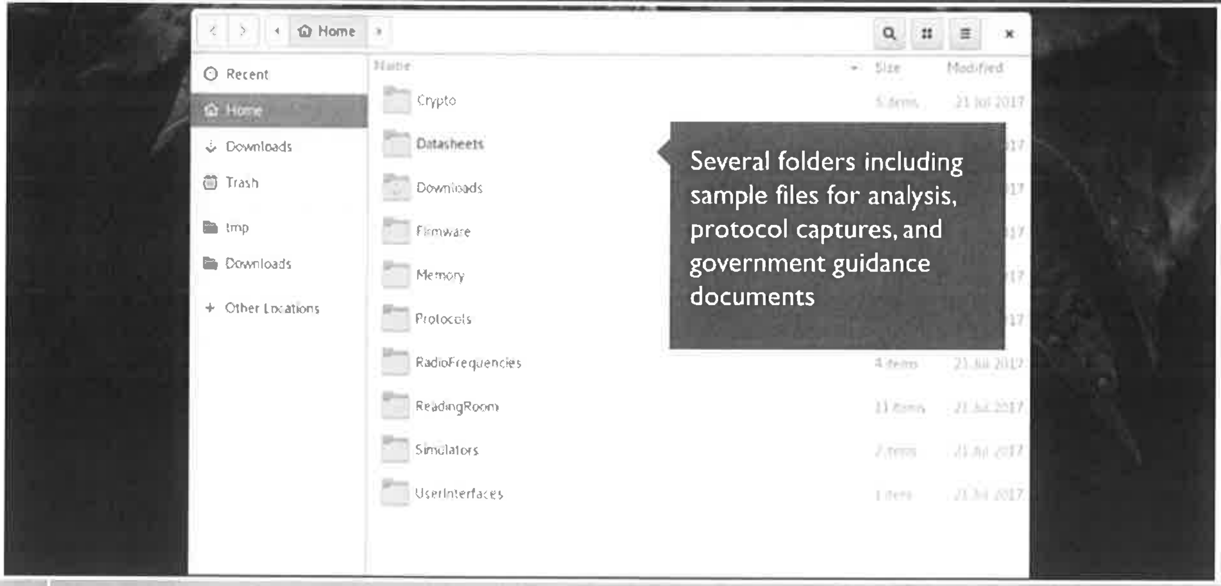

Activities
Sun 21:42
Type to search...
Terminal
Search for applications or application tag names
File browser
Display resolution and other settings
List of applications

ICS410 | ICS/SCADA Security Essentials 36

This page intentionally left blank.

© 2019, Justin Searle

SAMPLE FILES AND GUIDANCE DOCUMENTATION

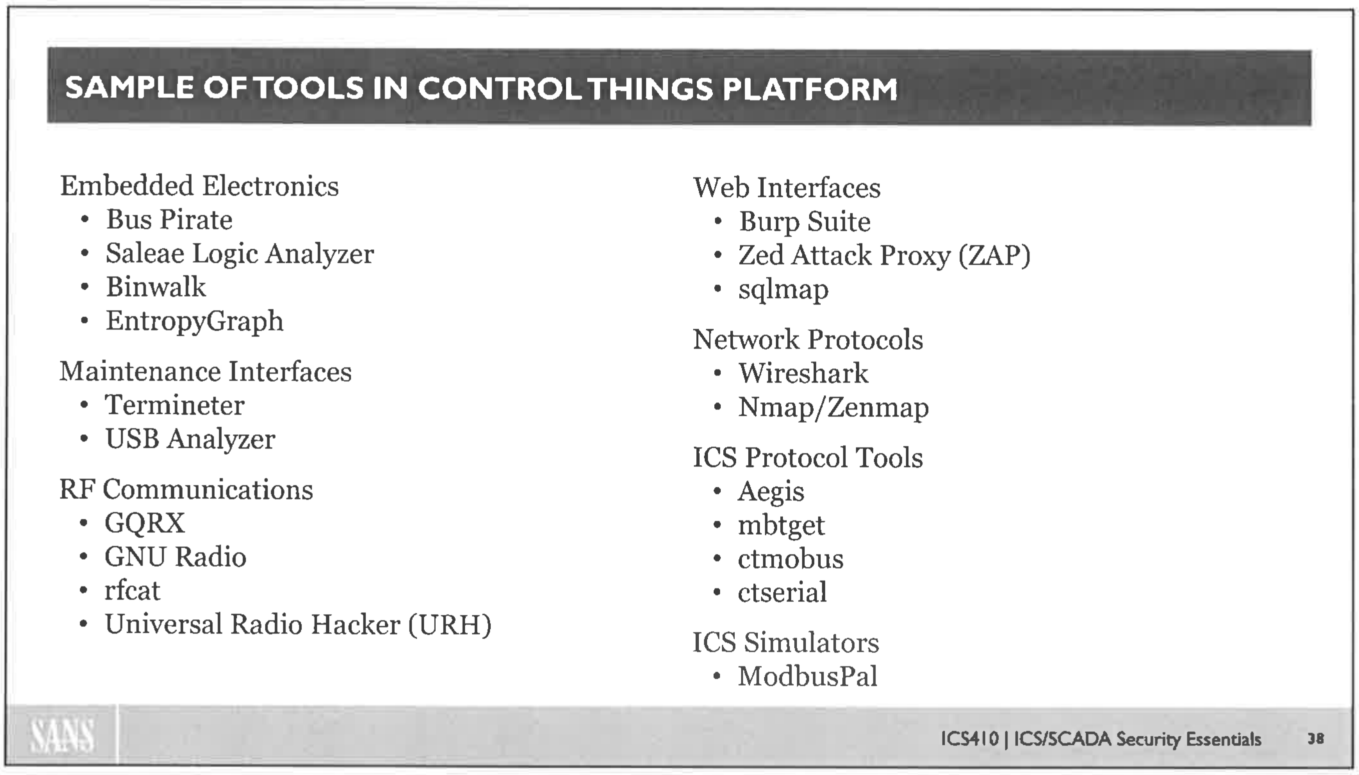


Several folders including sample files for analysis, protocol captures, and government guidance documents

ICS410 | ICS/SCADA Security Essentials 37

Open the file manager and it will open the control user's home folder, or /home/control. In the Reading-Room, you will find many freely available documents discussing various security topics related to industrial control system security. A few documents you may want to explore include the NESCOR Guide to Penetration Testing, which describes a methodology for penetration testing and what this distribution is based on, and the NIST SP800-82, which is the "Guide to Industrial Control Systems (ICS) Security." A copy of the NIST Cybersecurity Framework is also included, which we will discuss on Day 5.

In the other folders, you will find several datasets to practice and learn security assessment techniques. We will use some of these datasets later in this class when we explore the techniques attackers use to exploit our control systems. Having tools relevant to ICS environments is great, but, better yet, the Control Things distribution provides datasets and capture files from ICS environments that you can perform analysis on. We will examine some of these files later in the course.



SAMPLE OF TOOLS IN CONTROL THINGS PLATFORM

Embedded Electronics
- Bus Pirate
- Saleae Logic Analyzer
- Binwalk
- EntropyGraph

Maintenance Interfaces
- Termineter
- USB Analyzer

RF Communications
- GQRX
- GNU Radio
- rfcat
- Universal Radio Hacker (URH)

Web Interfaces
- Burp Suite
- Zed Attack Proxy (ZAP)
- sqlmap

Network Protocols
- Wireshark
- Nmap/Zenmap

ICS Protocol Tools
- Aegis
- mbtget
- ctmobus
- ctserial

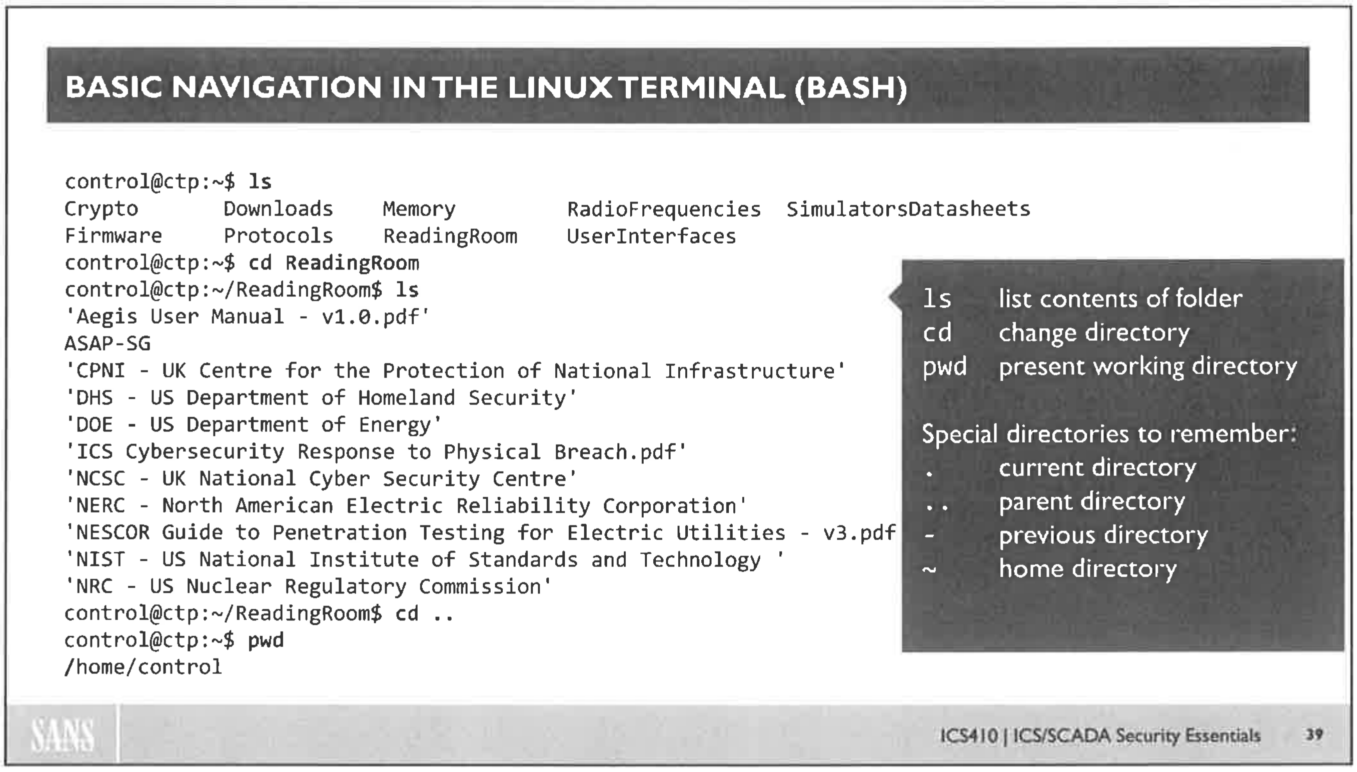
ICS Simulators
- ModbusPal

SANS ICS410 | ICS/SCADA Security Essentials 38

A number of ICS-relevant tools and simulators are available in the Control Things Platform. We will use some of these tools throughout this course; however, many others are covered in other courses, such as the hosted course "Assessing and Exploiting Control Systems" that is usually offered at the various SANS ICS Summits and other non-SANS venues.

Bus Pirate: Dump EEPROMs and Flash chips and capture circuit board communications.
Saleae Logic Analyzer: Capture and analyze circuit board communications.
Binwalk: Analyze binary files and datasets.
EntropyGraph: Discover randomness of blocks of data in a larger dataset.
Termineter: Interface with C12.18 opticoupler port on energy meters.
USB Analyzer: Analyze captured USB traffic.
GQRX: RF spectrum analyzer and traffic capture.
GNU Radio: Capture, analyze, and create proprietary RF traffic.
rfcat: Capture, analyze, and create proprietary RF traffic.
Universal Radio Hacker (URH): Capture, analyze, and create proprietary RF traffic.
Burp Suite: Capture, analyze, and create HTTP traffic.
Zed Attack Proxy (ZAP): Capture, analyze, and create HTTP traffic.
sqlmap: Discover and exploit SQL injection vulnerabilities in web applications.
Wireshark: Capture and analyze network traffic.
Aegis: Fuzz DNP3 traffic.
mbtget: Enumerate Modbus traffic.
ctmobus: Capture, analyze, and create Modbus traffic (serial and TCP).
ctserial: Capture, analyze, and create raw serial traffic.
Nmap/Zenmap: Portscan and map TCP/IP networks.
ModbusPal: This is modus simulation software.

© 2019, Justin Searle



BASIC NAVIGATION IN THE LINUX TERMINAL (BASH)

```
control@ctp:~$ ls
Crypto       Downloads    Memory        RadioFrequencies  SimulatorsDatasheets
Firmware     Protocols    ReadingRoom   UserInterfaces
control@ctp:~$ cd ReadingRoom
control@ctp:~/ReadingRoom$ ls
'Aegis User Manual - v1.0.pdf'
ASAP-SG
'CPNI - UK Centre for the Protection of National Infrastructure'
'DHS - US Department of Homeland Security'
'DOE - US Department of Energy'
'ICS Cybersecurity Response to Physical Breach.pdf'
'NCSC - UK National Cyber Security Centre'
'NERC - North American Electric Reliability Corporation'
'NESCOR Guide to Penetration Testing for Electric Utilities - v3.pdf
'NIST - US National Institute of Standards and Technology '
'NRC - US Nuclear Regulatory Commission'
control@ctp:~/ReadingRoom$ cd ..
control@ctp:~$ pwd
/home/control
```

ls	list contents of folder
cd	change directory
pwd	present working directory

Special directories to remember:

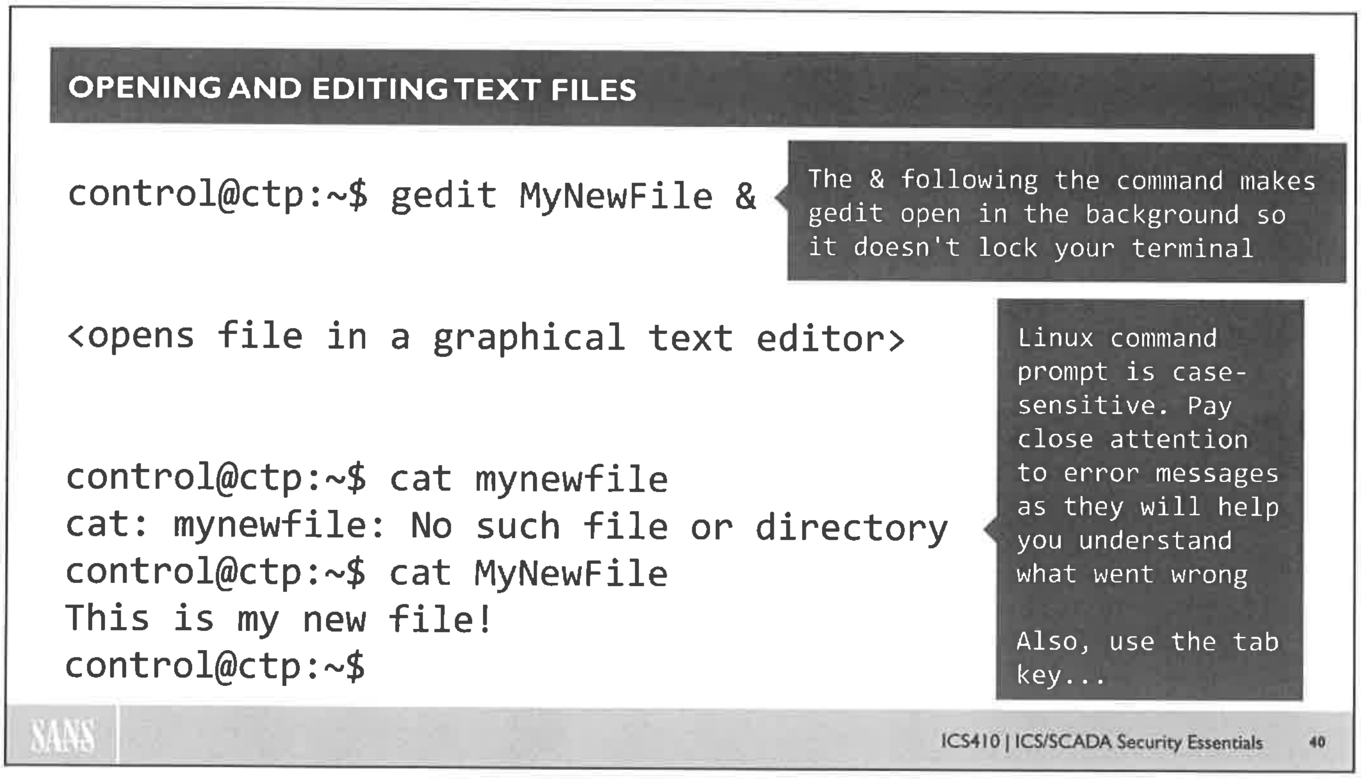
.	current directory
..	parent directory
-	previous directory
~	home directory

SANS ICS410 | ICS/SCADA Security Essentials 39

Now open a terminal by clicking on the "Terminal" icon (black square icon) on the bottom of the window. This opens up a Linux terminal where you can enter commands. Try the command in bold above. Here is what those commands do:

- `ls` stands for "list." Used to see the contents of a directory/folder.
- `cd` stands for "change directory." Used to move from one directory/folder to another.
- `cat` stands for "concatenate." Used to print the contents of a text file to the screen.
- `cd ..` The two periods represent a special directory that tells "cd" to move one directory/folder higher.
- `pwd` stands for "present working directory." Used to find out which directory you are currently in.

As a side note, every time you open a terminal application in Linux (like Terminal), you will always be in your current user's home directory, which, in our case, is `/home/control` because we are logged in as the control user.



OPENING AND EDITING TEXT FILES

```
control@ctp:~$ gedit MyNewFile &
```

The & following the command makes gedit open in the background so it doesn't lock your terminal

```
<opens file in a graphical text editor>


control@ctp:~$ cat mynewfile
cat: mynewfile: No such file or directory
control@ctp:~$ cat MyNewFile
This is my new file!
control@ctp:~$
```

Linux command prompt is case-sensitive. Pay close attention to error messages as they will help you understand what went wrong

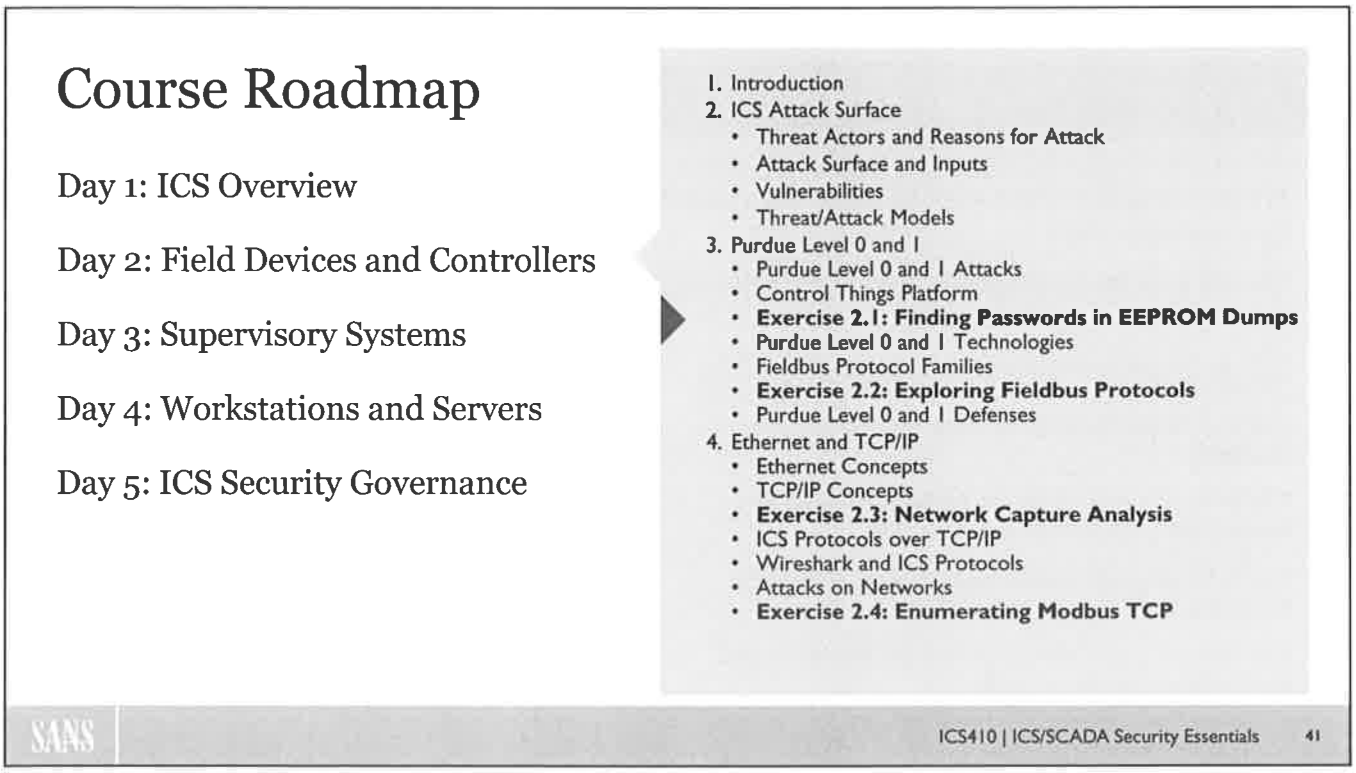
Also, use the tab key...

ICS410 | ICS/SCADA Security Essentials 40

Instead of using the cat command, you can also open up a graphical text editor like Kate. You can do that by using the icon on the main desktop or just by using the `kate` command in the terminal. Try the commands above to create a file called "`MyNewFile`" with that exact capitalization and save it to your home folder. Once you create that folder, close Kate and try reading the file you just created with the `cat` command.

Notice the output above where the first `cat` command fails because we didn't capitalize the right letters. In Linux, everything is case-sensitive, so be careful how you name your files.

© 2019, Justin Searle



Course Roadmap

Day 1: ICS Overview

Day 2: Field Devices and Controllers

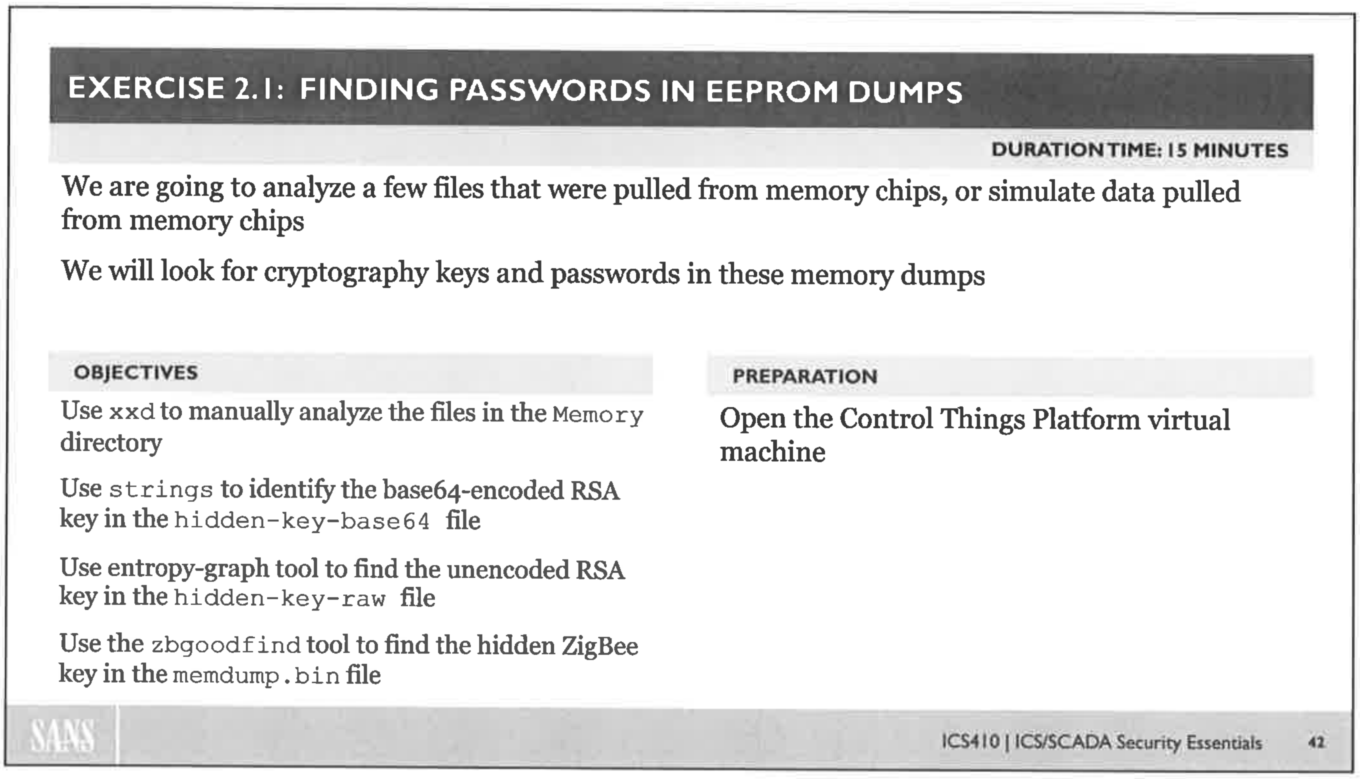
Day 3: Supervisory Systems

Day 4: Workstations and Servers

Day 5: ICS Security Governance

SANS

This page intentionally left blank.



EXERCISE 2.1: FINDING PASSWORDS IN EEPROM DUMPS

DURATION TIME: 15 MINUTES

We are going to analyze a few files that were pulled from memory chips, or simulate data pulled from memory chips

We will look for cryptography keys and passwords in these memory dumps

OBJECTIVES

Use `xxd` to manually analyze the files in the `Memory` directory

Use `strings` to identify the base64-encoded RSA key in the `hidden-key-base64` file

Use entropy-graph tool to find the unencoded RSA key in the `hidden-key-raw` file

Use the `zbgoodfind` tool to find the hidden ZigBee key in the `memdump.bin` file

PREPARATION

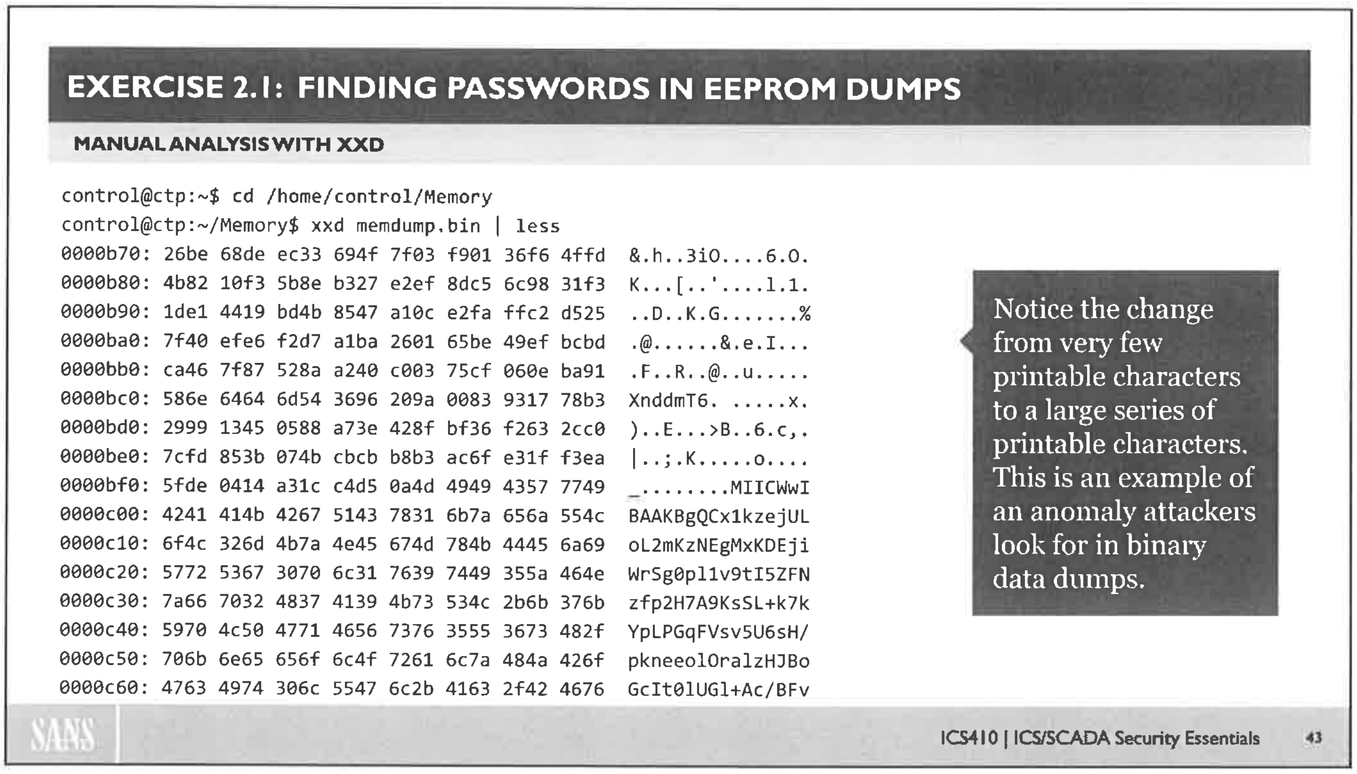
Open the Control Things Platform virtual machine

SANS

ICS410 | ICS/SCADA Security Essentials 42

This page intentionally left blank.

© 2019, Justin Searle



EXERCISE 2.1: FINDING PASSWORDS IN EEPROM DUMPS

MANUAL ANALYSIS WITH XXD

```
control@ctp:~$ cd /home/control/Memory
control@ctp:~/Memory$ xxd memdump.bin | less
0000b70: 26be 68de ec33 694f 7f03 f901 36f6 4ffd  &.h..3iO....6.O.
0000b80: 4b82 10f3 5b8e b327 e2ef 8dc5 6c98 31f3  K...[..'....l.1.
0000b90: 1de1 4419 bd4b 8547 a10c e2fa ffc2 d525  ..D..K.G.......%
0000ba0: 7f40 efe6 f2d7 a1ba 2601 65be 49ef bcbd  .@......&.e.I...
0000bb0: ca46 7f87 528a a240 c003 75cf 060e ba91  .F..R..@..u.....
0000bc0: 586e 6464 6d54 3696 209a 0083 9317 78b3  XnddmT6. .....x.
0000bd0: 2999 1345 0588 a73e 428f bf36 f263 2cc0  )..E...>B..6.c,.
0000be0: 7cfd 853b 074b cbcb b8b3 ac6f e31f f3ea  |..;.K.....o....
0000bf0: 5fde 0414 a31c c4d5 0a4d 4949 4357 7749  _........MIICWwI
0000c00: 4241 414b 4267 5143 7831 6b7a 656a 554c  BAAKBgQCx1kzejUL
0000c10: 6f4c 326d 4b7a 4e45 674d 784b 4445 6a69  oL2mKzNEgMxKDEji
0000c20: 5772 5367 3070 6c31 7639 7449 355a 464e  WrSg0pl1v9tI5ZFN
0000c30: 7a66 7032 4837 4139 4b73 534c 2b6b 376b  zfp2H7A9KsSL+k7k
0000c40: 5970 4c50 4771 4656 7376 3555 3673 482f  YpLPGqFVsv5U6sH/
0000c50: 706b 6e65 656f 6c4f 7261 6c7a 484a 426f  pkneeolOralzHJBo
0000c60: 4763 4974 306c 5547 6c2b 4163 2f42 4676  GcIt0lUGl+Ac/BFv
```

Notice the change from very few printable characters to a large series of printable characters. This is an example of an anomaly attackers look for in binary data dumps.

SANS

ICS410 | ICS/SCADA Security Essentials 43

The first thing we'll do to analyze these files is to use a hex dump tool to manually look at the dataset. In the Control Things Platform virtual machine, the **xxd** tool can be used to do this. To make it even easier, pipe the output of **xxd** to **less** so you can use your arrow keys to scroll up and down.

→ Hex Dump tool

STEP 0 – Click on the "Activities" link in the upper left corner of the screen to get to the main menu, and then click on the "Terminal" icon that looks like a black box on the bottom left. If you don't see a Terminal box on the left, just search for "Terminal" in the search box at the top.

STEP 1 – Type the following command to get to the right directory:

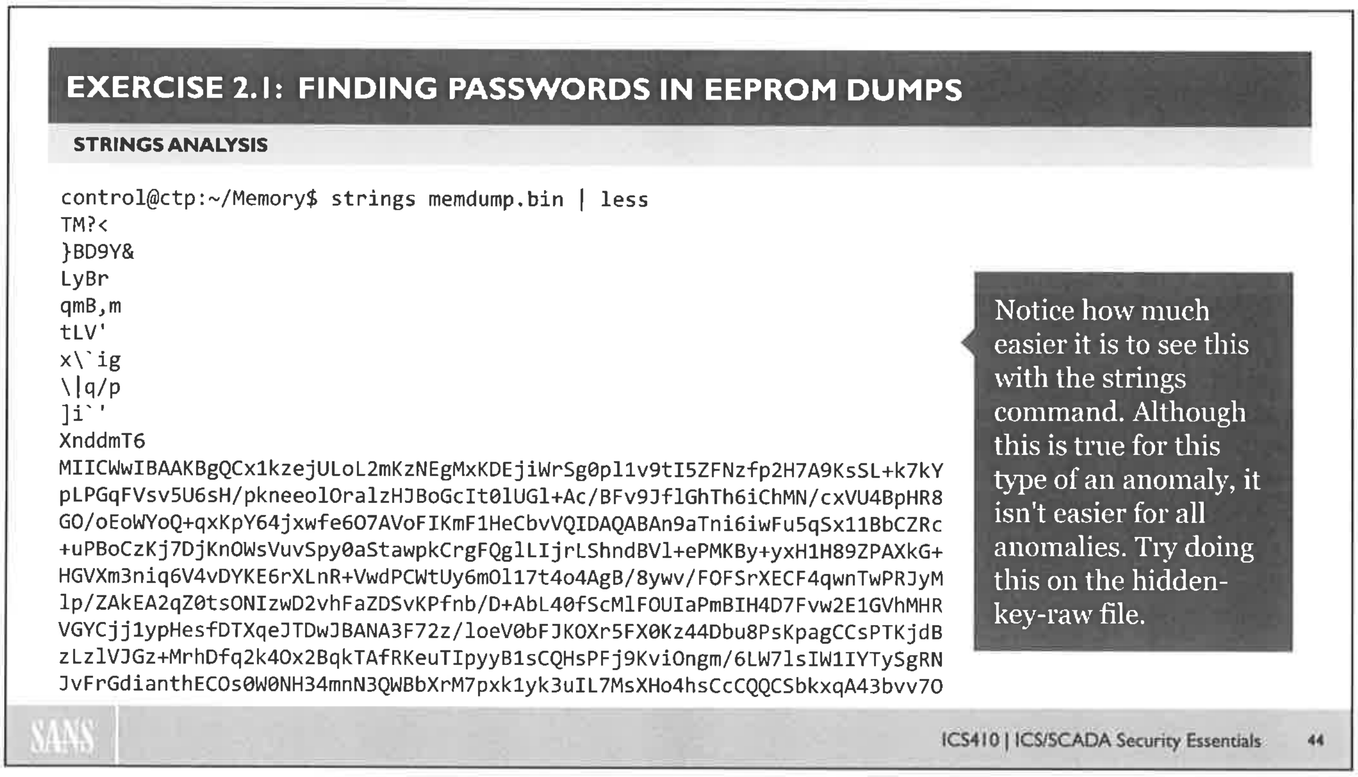
```
cd /home/control/Memory
```

STEP 2 – Type the following command. You can use the up/down arrow keys to scroll through the file and type "q" to exit when finished.

```
xxd memdump.bin | less
```

Once you have the files open, just scroll down looking for anomalies in the data. In this case, the change from lots of non-printable characters (show with all the periods) to no non-printable characters is quite noticeable. With closer inspection, you should be able to tell that this large block of printable characters is base64-encoded, a very common technique for crypto key storage. Check out all the files in this directory.

When you are done looking at this output, type "q" to exit.



EXERCISE 2.1: FINDING PASSWORDS IN EEPROM DUMPS

STRINGS ANALYSIS

```
control@ctp:~/Memory$ strings memdump.bin | less
TM?<
}BD9Y&
LyBr
qmB,m
tLV'
x\`ig
\|q/p
]i`'
XnddmT6
MIICWwIBAAKBgQCx1kzejULoL2mKzNEgMxKDEjiWrSg0pl1v9tI5ZFNzfp2H7A9KsSL+k7kY
pLPGqFVsv5U6sH/pkneeolOralzHJBoGcIt0lUGl+Ac/BFv9JflGhTh6iChMN/cxVU4BpHR8
GO/oEoWYoQ+qxKpY64jxwfe6O7AVoFIKmF1HeCbvVQIDAQABAn9aTni6iwFu5qSx11BbCZRc
+uPBoCzKj7DjKnOWsVuvSpy0aStawpkCrgFQglLIjrLShndBVl+ePMKBy+yxH1H89ZPAXkG+
HGVXm3niq6V4vDYKE6rXLnR+VwdPCWtUy6mOl17t4o4AgB/8ywv/FOFSrXECF4qwnTwPRJyM
lp/ZAkEA2qZ0tsONIzwD2vhFaZDSvKPfnb/D+AbL40fScMlFOUIaPmBIH4D7Fvw2E1GVhMHR
VGYCjj1ypHesfDTXqeJTDwJBANA3F72z/loeV0bFJKOXr5FX0Kz44Dbu8PsKpagCCsPTKjdB
zLzlVJGz+MrhDfq2k4Ox2BqkTAfRKeuTIpyyB1sCQHsPFj9KviOngm/6LW7lsIW1IYTySgRN
JvFrGdianthECOs0W0NH34mnN3QWBbXrM7pxk1yk3uIL7MsXHo4hsCcCQQCSbkxqA43bvv7O
```

Notice how much easier it is to see this with the strings command. Although this is true for this type of an anomaly, it isn't easier for all anomalies. Try doing this on the hidden-key-raw file.

SANS ICS410 | ICS/SCADA Security Essentials 44

Another tool we can use to manually analyze our binary datasets is the **strings** command. This allows you to focus in only on the blocks of printable characters while ignoring all the non-printable characters. While you miss some anomalies by not looking at the non-printable characters, you sometimes see things here that you don't with **xxd**. Another benefit of **strings** is its support of multiple encoding styles.

STEP 3 – Try the following commands. Some will come up with printable strings, but most will not. You never know how your data is encoded, so it doesn't hurt to check all the encoding styles available. The following encoding options to **strings** decodes the file as ASCII, 16-bit unicode, and 32-bit unicode. The lowercase version uses big endian and the uppercase uses little endian. If text appears, analyze it to see if it seems meaningful in any way. If you have time, try this on the other sample files in this directory.

```
strings memdump.bin | less
strings -n 12 memdump.bin
strings -n 12 memdump.bin | base64 -d
strings -es memdump.bin | less
strings -eS memdump.bin | less
strings -eb memdump.bin | less
strings -eB memdump.bin | less
strings -el memdump.bin | less
strings -eL memdump.bin | less
```

at least 12 byte

→ RSA key (59:42)

find data and decode.

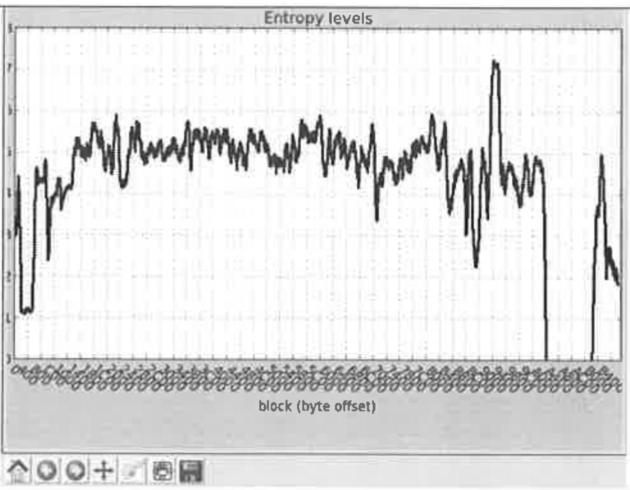
For more information on endianness, see https://en.wikipedia.org/wiki/Endianness

© 2019, Justin Searle

EXERCISE 2.1: FINDING PASSWORDS IN EEPROM DUMPS

ENTROPY ANALYSIS

```
control@ctp:~/Memory$ entropy-graph hidden-key-raw
```



increased randomness. means crypto key.

With the hidden-key-raw file, it's much easier to find the cryptography key by doing an entropy analysis of smaller blocks of data in the larger dataset. The spike starting at byte offset 9000 is the hidden crypto key.

Check out the **hidden-key-raw** files. Using **xxd** and **strings** on this file doesn't help us find the hidden crypto key because it looks the same as its surrounding data. However, if we use a tool like **entropy-graph** to look for blocks of highly random data in the dataset, it becomes trivial to find the crypto key. Try this approach on some of the other files in this directory.

STEP 4 – Type the following command to have entropy-graph build an entropy graph of the hidden-key-raw file:

```
entropy-graph hidden-key-raw
```

You can also try running binwalk to see if it can automatically find the key with:

```
binwalk hidden-key-raw
```

Binwalk can also do a quick but simple entropy analysis like entropy-graph with:

```
binwalk -EB hidden-key-raw
```

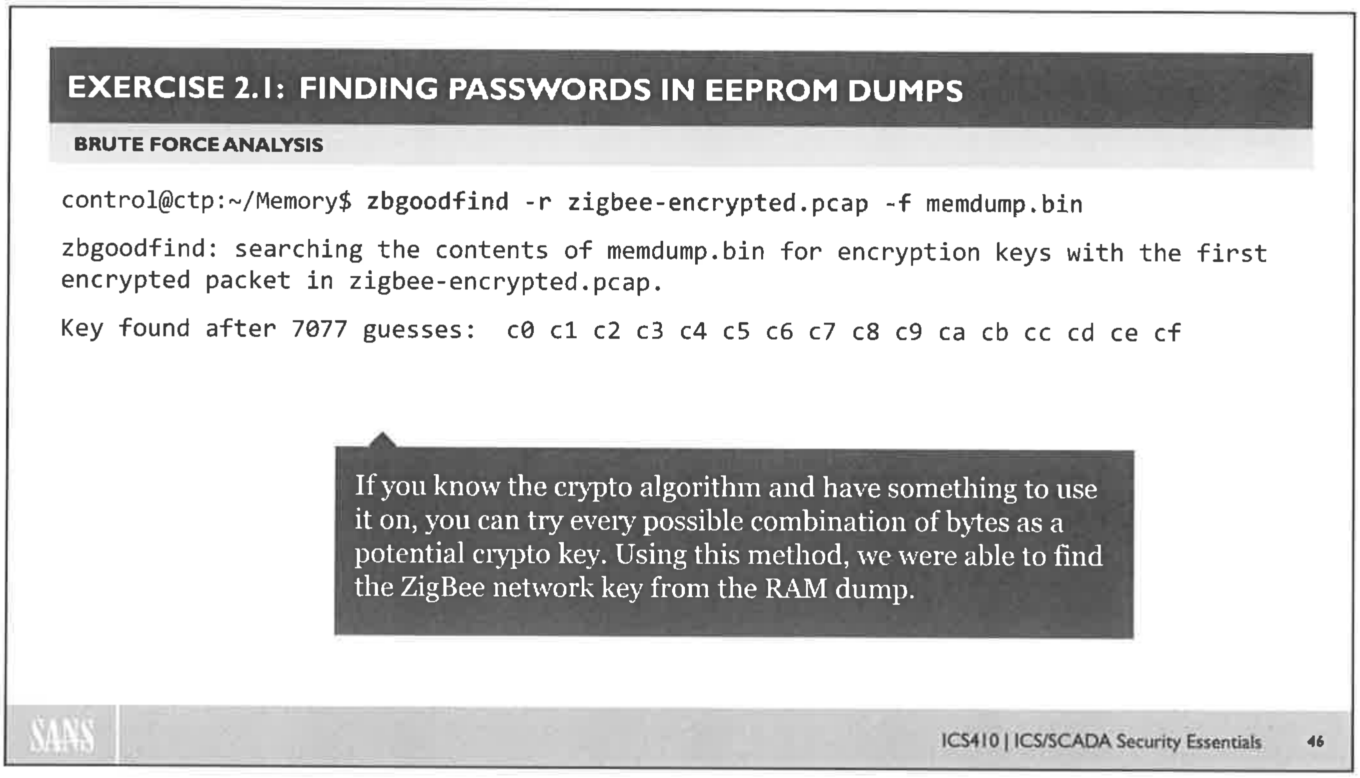
$\longrightarrow$ it can give you predictive analysis over graph. it points the correct place to look at.

It is harder but still possible to use the entropy method if the key is encoded and not in raw form:

```
entropy-graph hidden-key-base64
binwalk -EB hidden-key-base64
```

why do we encode:
in order to eliminate specific character like EOL/0/Null.
64 possible characters are used.
it finishes with = / == equal signs.

© 2019, Justin Searle



EXERCISE 2.1: FINDING PASSWORDS IN EEPROM DUMPS

BRUTE FORCE ANALYSIS

```
control@ctp:~/Memory$ zbgoodfind -r zigbee-encrypted.pcap -f memdump.bin
zbgoodfind: searching the contents of memdump.bin for encryption keys with the first
encrypted packet in zigbee-encrypted.pcap.
Key found after 7077 guesses:  c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 ca cb cc cd ce cf
```

If you know the crypto algorithm and have something to use it on, you can try every possible combination of bytes as a potential crypto key. Using this method, we were able to find the ZigBee network key from the RAM dump.

SANS ICS410 | ICS/SCADA Security Essentials 46

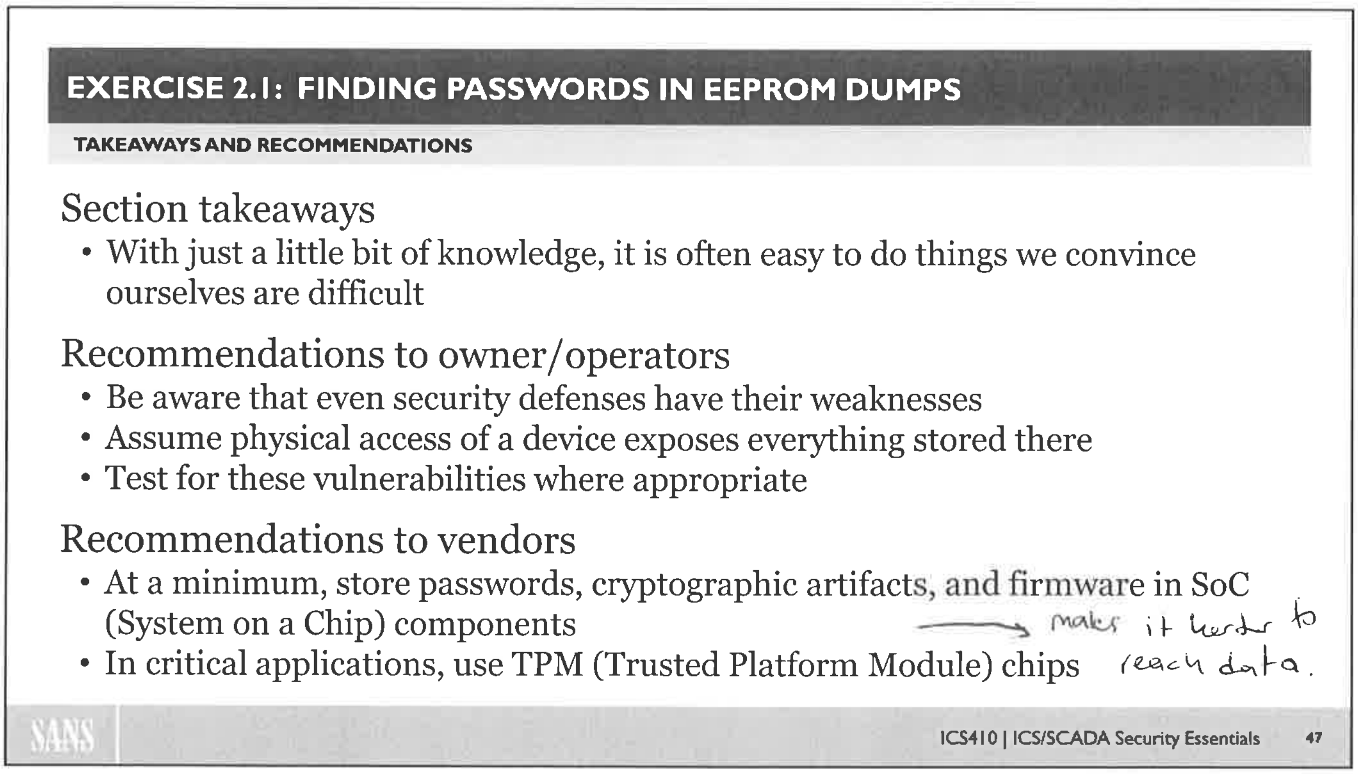
The final method we can use to find crypto keys in large datasets is using brute force techniques. If you know the crypto algorithm and have something to use it on, you can try every possible combination of bytes as a potential crypto key. Using this method, we were able to find the ZigBee network key from the RAM dump. This will work only on the **memdump.bin** file because it is the only file that contains the ZigBee password for encrypted ZigBee packets in the **zigbee-encrypted.pcap** file. In the ICS market, there are only two tools publicly available to perform this type of test. One is this zbgoodfind tool for the ZigBee wireless protocol. The other is a tool named OptiGuard, which is used to test c12.18 passwords on smart meters.

STEP 5 – Type the following command to find the ZigBee key captured in the RAM dump:

```
zbgoodfind -r zigbee-encrypted.pcap -f memdump.bin
```

trying to find zigbeekey.

© 2019, Justin Searle



EXERCISE 2.1: FINDING PASSWORDS IN EEPROM DUMPS

TAKEAWAYS AND RECOMMENDATIONS

Section takeaways

- With just a little bit of knowledge, it is often easy to do things we convince ourselves are difficult

Recommendations to owner/operators

- Be aware that even security defenses have their weaknesses
- Assume physical access of a device exposes everything stored there
- Test for these vulnerabilities where appropriate

Recommendations to vendors

- At a minimum, store passwords, cryptographic artifacts, and firmware in SoC (System on a Chip) components → makes it harder to reach data.
- In critical applications, use TPM (Trusted Platform Module) chips

SANS ICS410 | ICS/SCADA Security Essentials 47

Once an attacker has been able to retrieve binary data from EEPROM/flash dumping and bus snooping, he can use various techniques to find crypto keys and firmware images.

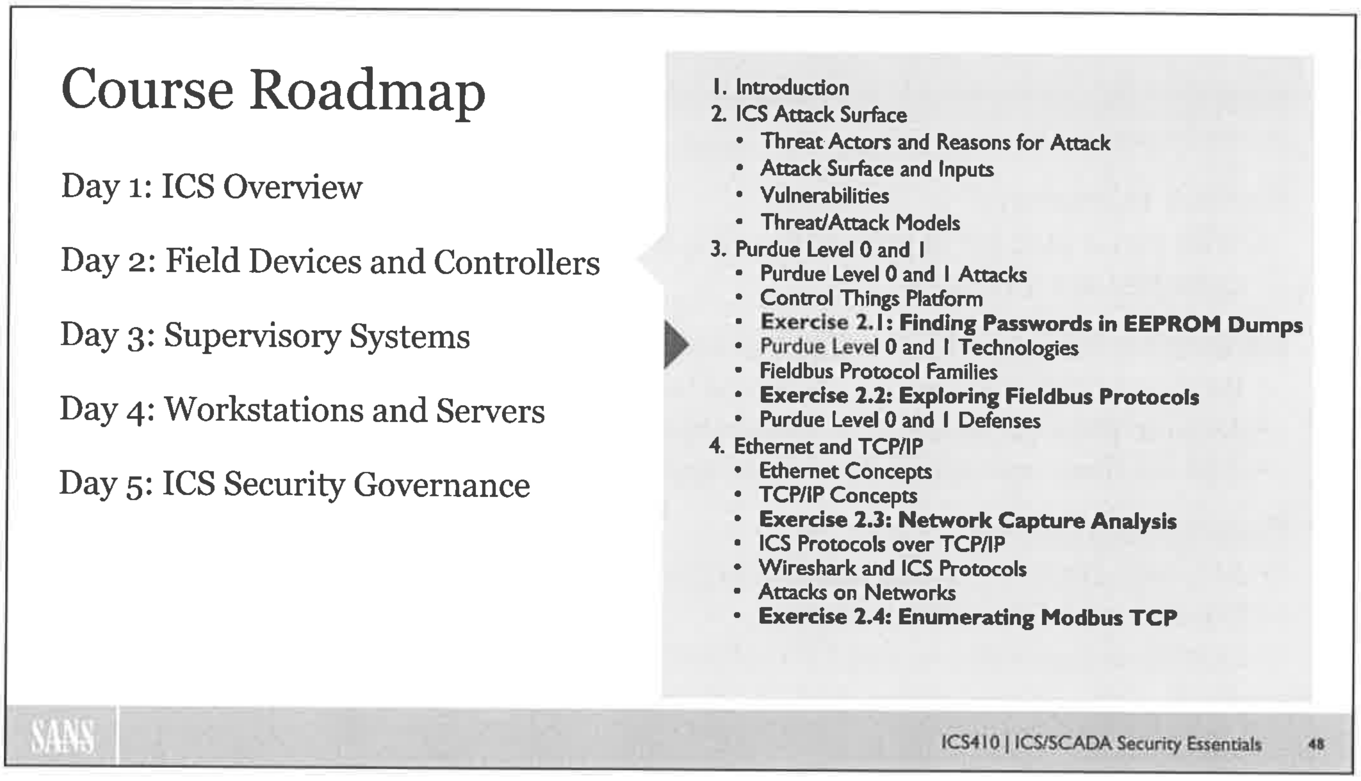
The primary method of discovery is to look for anomalies in the data that signify crypto keys, such as high levels of entropy.

We can manually look for human-chosen keys.

We can use brute force techniques if we know the algorithm and have an encrypted sample.

- Physical access
- take the device.
- Copy the memory dump
- analyze dump.
- find keys.
- decrypt the communication

means if crypto key is same for all field devices, comprimise of one is comprimise → all. means,



Course Roadmap

Day 1: ICS Overview

Day 2: Field Devices and Controllers

Day 3: Supervisory Systems

Day 4: Workstations and Servers

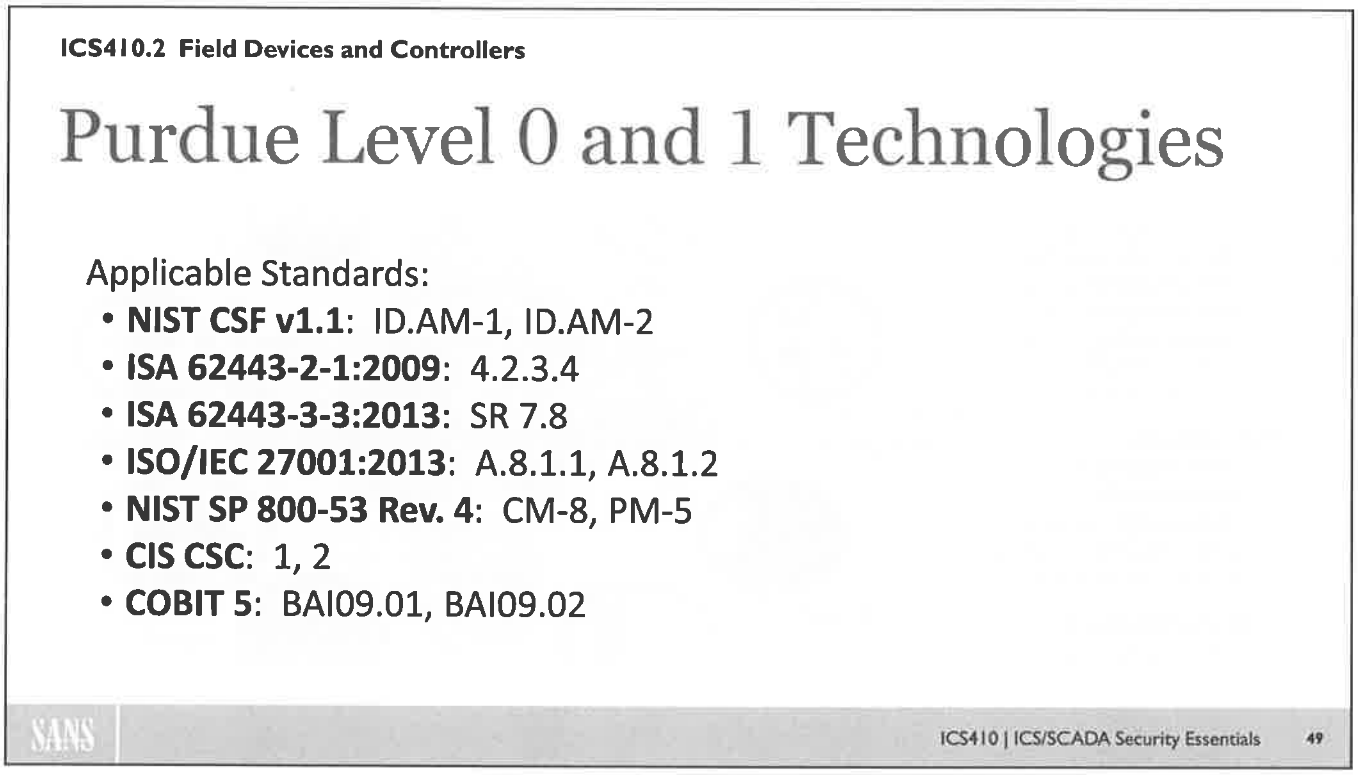
Day 5: ICS Security Governance

SANS

ICS410 | ICS/SCADA Security Essentials 48

This page intentionally left blank.

© 2019, Justin Searle



ICS410.2 Field Devices and Controllers

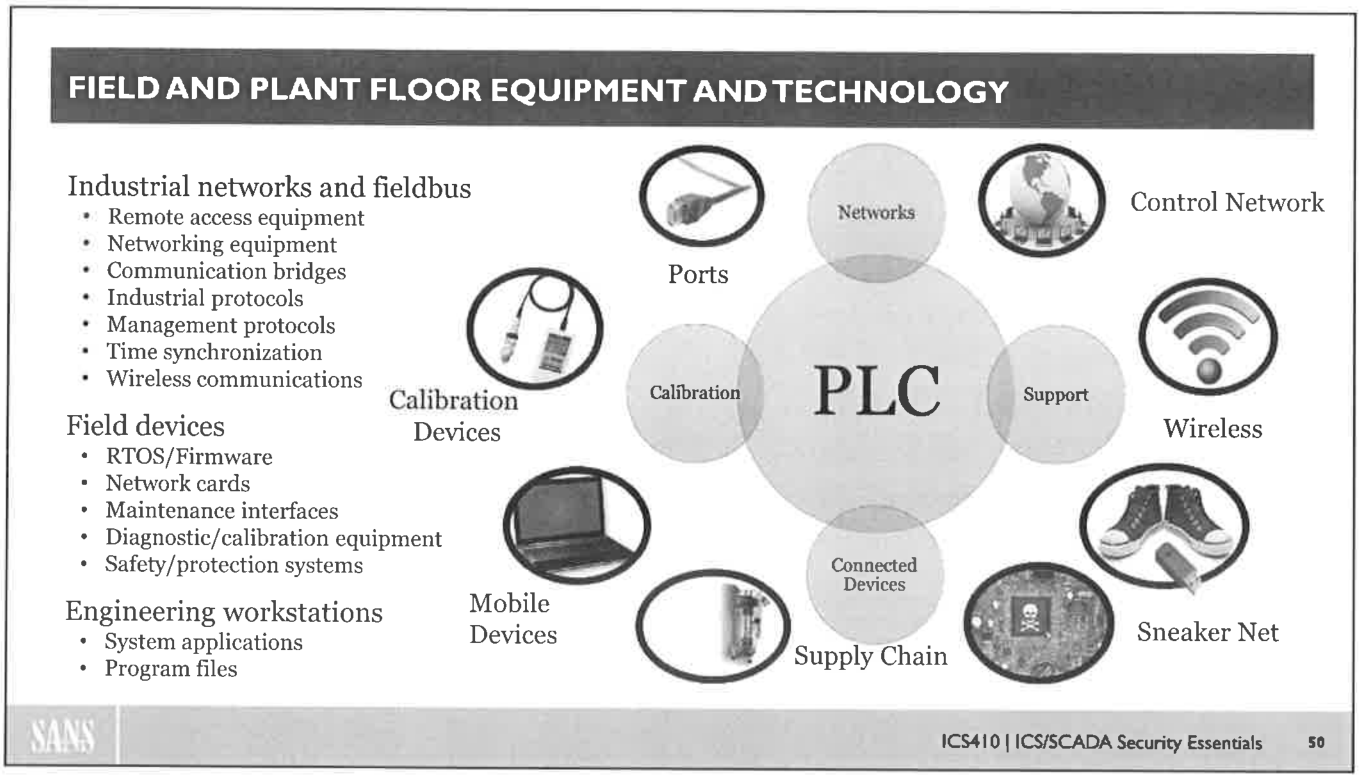
Purdue Level 0 and 1 Technologies

Applicable Standards:

- **NIST CSF v1.1**: ID.AM-1, ID.AM-2
- **ISA 62443-2-1:2009**: 4.2.3.4
- **ISA 62443-3-3:2013**: SR 7.8
- **ISO/IEC 27001:2013**: A.8.1.1, A.8.1.2
- **NIST SP 800-53 Rev. 4**: CM-8, PM-5
- **CIS CSC**: 1, 2
- **COBIT 5**: BAI09.01, BAI09.02

SANS ICS410 | ICS/SCADA Security Essentials 49

This page intentionally left blank.



FIELD AND PLANT FLOOR EQUIPMENT AND TECHNOLOGY

Industrial networks and fieldbus
- Remote access equipment
- Networking equipment
- Communication bridges
- Industrial protocols
- Management protocols
- Time synchronization
- Wireless communications

Field devices
- RTOS/Firmware
- Network cards
- Maintenance interfaces
- Diagnostic/calibration equipment
- Safety/protection systems

Engineering workstations
- System applications
- Program files

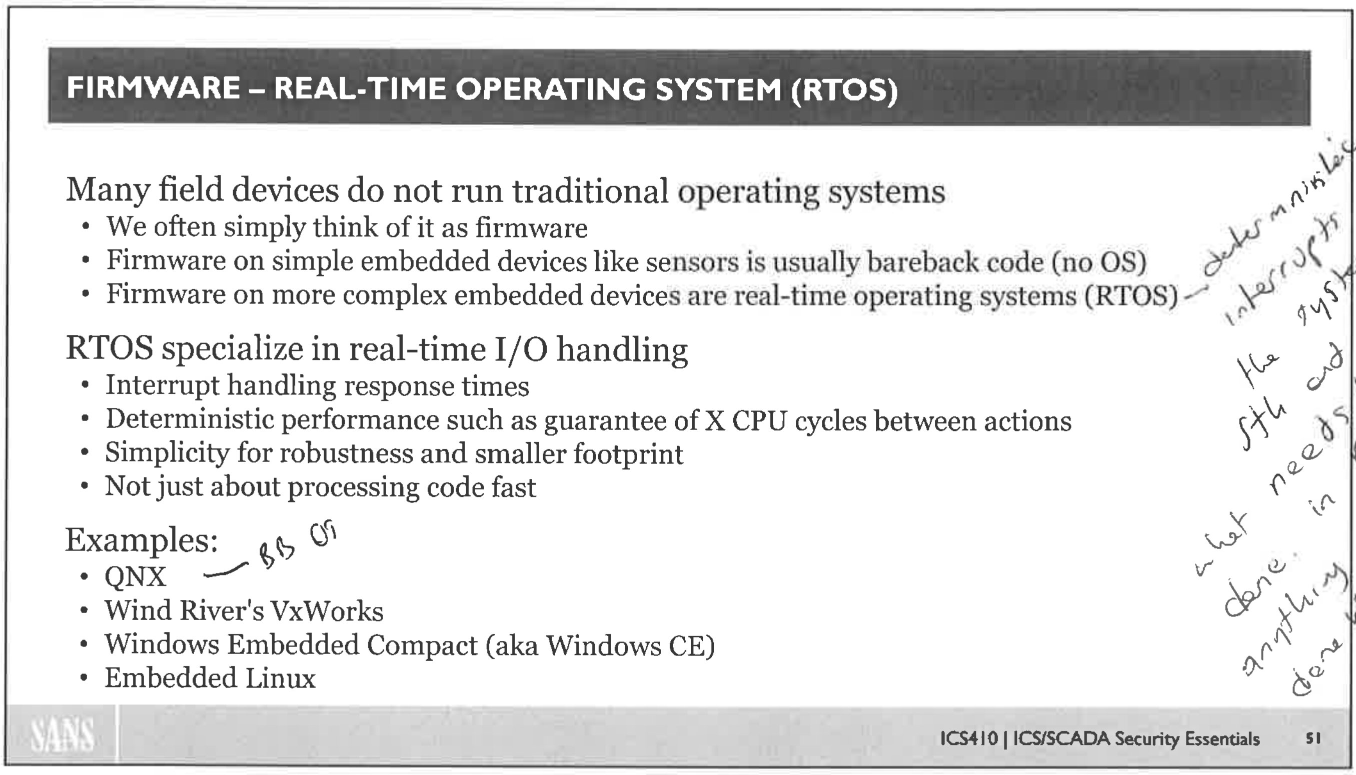
Field and plant floor equipment to be protected can encompass many types of components and data.

Security at this level (closer to the plant floor or field location) can be difficult for an end user. The resources available are often restrained, many security features are incomplete or basic, operational drivers can override security approaches, and the number of access paths is greater than one would think. Security at this level is important as an attacker can achieve significant impacts such as mal-operating the process, making changes to setpoints, damaging ICS components, damaging physical equipment by suppressing the safety system and protections, causing loss of view, blocking control, spoofing operators, and modifying or even spoofing input to logic.

Controllers and field devices are vulnerable via physical access, network access, and remote access. There are many connected (or temporarily connected) devices, infrastructures, and subsystems that interface with these devices that can affect their security. The slide provides a list of things that must be protected or considered, such as networking equipment, protocols, system applications, engineering workstations, controllers/embedded systems, field devices, and equipment that connects to controllers or field devices.

Finally, understanding how operators and field/plant floor maintenance personnel interact with controllers and field devices is also important. One needs to consider how work is performed, such as how systems are calibrated or how firmware updates may be planned and completed.

© 2019, Justin Searle



FIRMWARE – REAL-TIME OPERATING SYSTEM (RTOS)

Many field devices do not run traditional operating systems

- We often simply think of it as firmware
- Firmware on simple embedded devices like sensors is usually bareback code (no OS)
- Firmware on more complex embedded devices are real-time operating systems (RTOS) — deterministic. In OS interrupts manages the system step, the sth and does what needs to be done. in RTOS anything can be done when just you need.

RTOS specialize in real-time I/O handling

- Interrupt handling response times
- Deterministic performance such as guarantee of X CPU cycles between actions
- Simplicity for robustness and smaller footprint
- Not just about processing code fast

Examples:

- QNX — BB OS
- Wind River's VxWorks
- Windows Embedded Compact (aka Windows CE)
- Embedded Linux

SANS ICS410 | ICS/SCADA Security Essentials 51

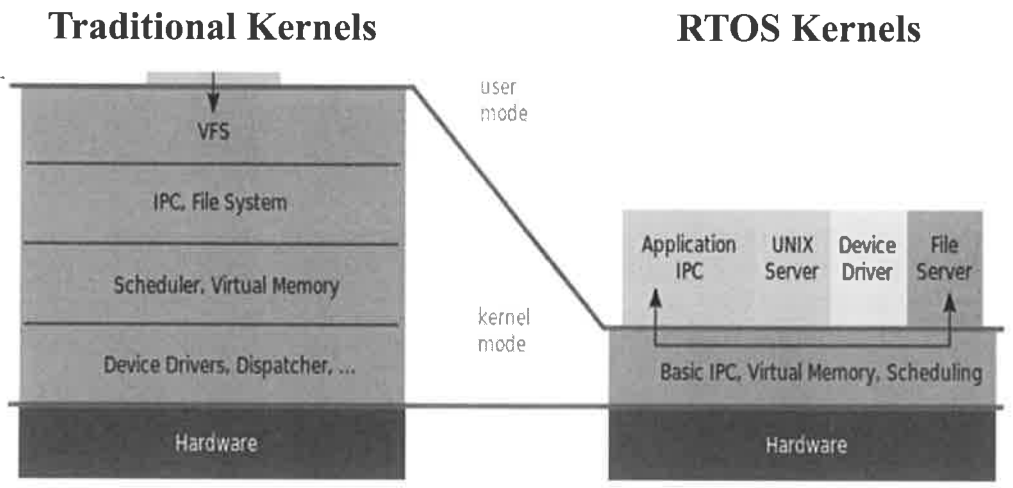
Embedded systems are systems designed for specific tasks and tailored to support those tasks. For instance, simple sensors have a specialized board with basic collection (often analog inputs), possibly digital inputs (buttons), and calibration capabilities. These simple sensors often have real-time computing requirements, but only need limited code to perform their actions, and therefore often use bareback code, or C-like programs installed directly on the microcontroller without any operating system. This keeps hardware costs low for production and code complexity low for supportability and reliability.

However, in more complex embedded devices, vendors can simplify development by building functionality on top of an operating system, eliminating the code they have to write such as TCP/IP communications stacks and parallel process scheduling. An RTOS is a real-time operating system that runs on an embedded device providing such capabilities without all the overhead of a traditional operating system we use on our computers. All that really means is that a specialized computer is running programs with a specific scheduling paradigm. A real-time operating system is so named because of the processing requirements. This means a different scheduling paradigm than your typical desktop computer. A process hogging a needed resource may not be acceptable. It's not uncommon for an RTOS to have a watchdog capability to reboot in the event of CPU lock or other system delay. To illustrate the idea, let's take a look at an RTOS scheduling algorithm, round robin. If three processes are running on a system, each is given equal time slices on the CPU. The processes are executed in a round-robin order, one after the other, and when a time slice has run out, the process is preempted until the next time around the cycle. If each process needed to perform its critical section every second, then no process could execute for over 333ms.

Additionally, hardware interrupts may be handled differently. The interrupt handlers are kept to a minimum—just enough to acknowledge the interrupt, disable future interrupts of that type, and alert any processes that there is work to do. This is possible because, in RTOS systems, there is unlikely to be separation of kernel space and user space. Thus, a process has greater system call efficiency and the ability to disable interrupts or otherwise manage the RTOS environment.

Exploits against embedded systems take on a different form than traditional computers. They require specialization to the specific device or environment, memory architecture, scheduling system, and other specifics. On the other hand, finding a vulnerability in an embedded system is more likely to gain an attacker full control of the system without the need for privilege escalation or other chaining of vulnerabilities.

MICROKERNEL VS. MONOLITHIC



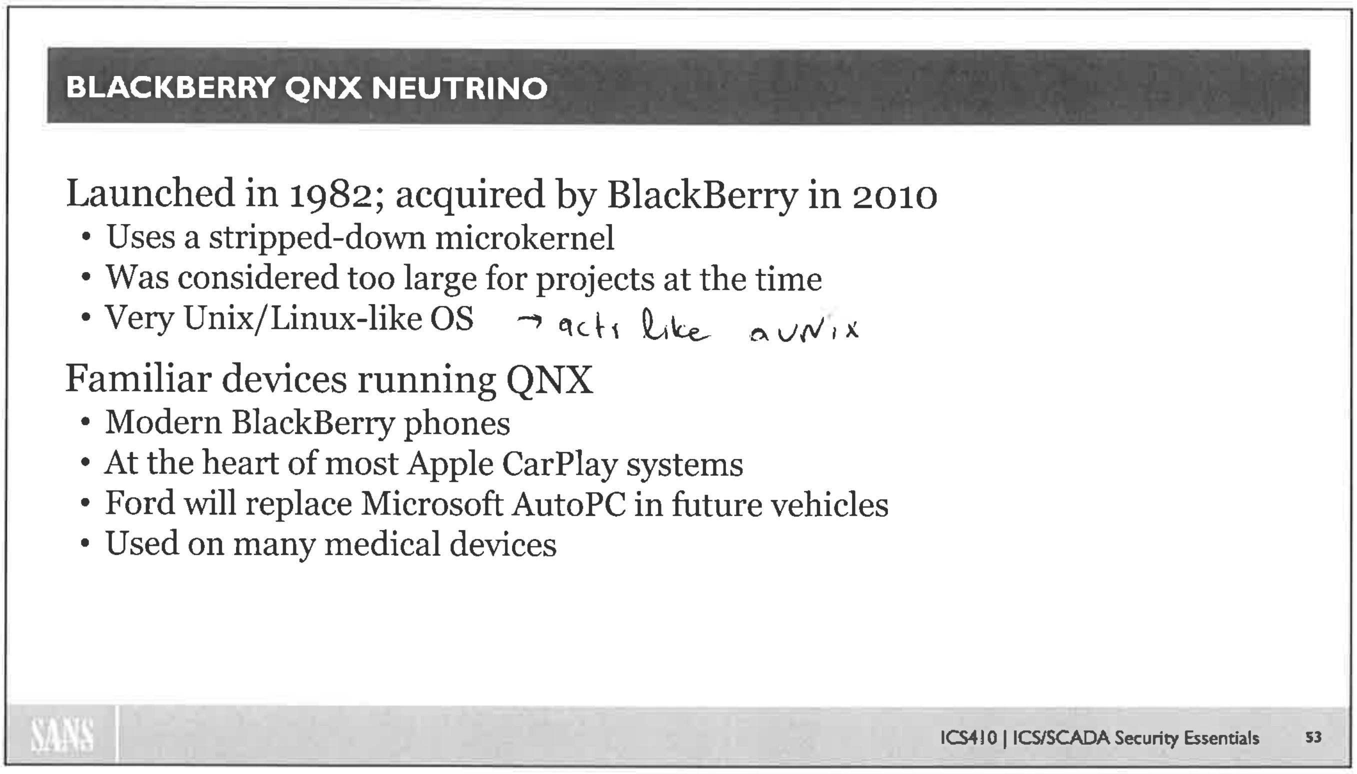
User prgrm needs to go all thru layers.

Not required to run real-time processes, but a common approach used by many RTOSs

This diagram illustrates the differences between a microkernel and a monolithic kernel. In this example, the microkernel is responsible for the functions of basic interprocess communication (IPC), virtual memory, and scheduling (which is different than QNX in the IPC area). All other typical kernel operations are userland programs that run and are responsible for things like file management.

Source: http://en.wikipedia.org/wiki/Microkernel

© 2019, Justin Searle



BLACKBERRY QNX NEUTRINO

Launched in 1982; acquired by BlackBerry in 2010

- Uses a stripped-down microkernel
- Was considered too large for projects at the time
- Very Unix/Linux-like OS → acts like a UNix

Familiar devices running QNX

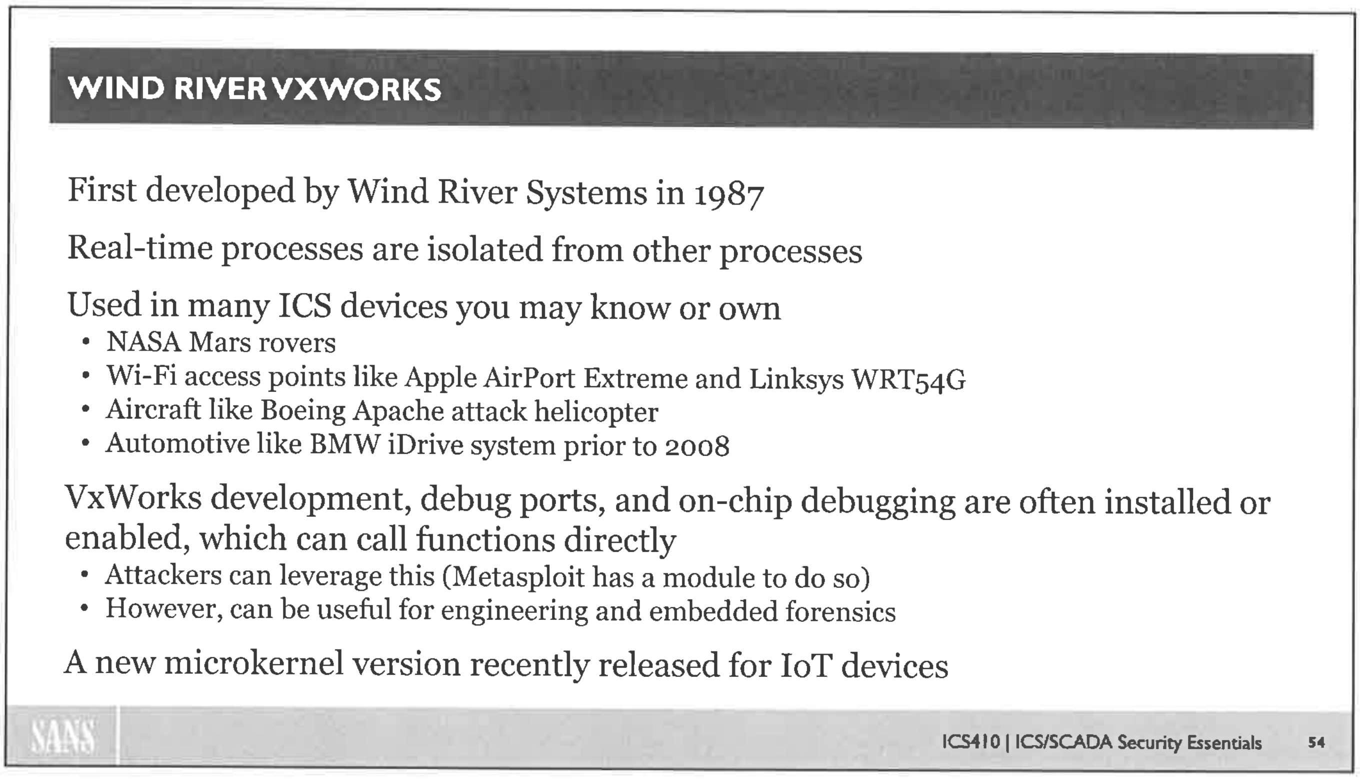
- Modern BlackBerry phones
- At the heart of most Apple CarPlay systems
- Ford will replace Microsoft AutoPC in future vehicles
- Used on many medical devices

SANS ICS410 | ICS/SCADA Security Essentials 53

QNX is an RTOS developed as a microkernel RTOS. A microkernel is different from the kernel you are most familiar with (a monolithic kernel) in ways that the name suggests. The QNX kernel is responsible for only three things: CPU scheduling (with real-time scheduling requirements), interrupts, and interprocess communication. The rest of the system exists as user processes.

QNX interprocess communication (IPC) differs in an interesting way from a standard *nix kernel. The messages are passed by copying from one program's memory space into another. That itself isn't very interesting, but what is interesting is how it interacts with the CPU scheduler. Remember that we're running in an RTOS environment here, so scheduling is on a different paradigm.

In QNX, when an IPC message is sent, if the receiving program is currently waiting for such a message, control of the CPU is immediately passed to the receiving process along with the message **without** moving through the Scheduler. This allows a sending program to receive an immediate response without getting preempted and missing its opportunity to run on the CPU.



WIND RIVER VXWORKS

First developed by Wind River Systems in 1987

Real-time processes are isolated from other processes

Used in many ICS devices you may know or own

- NASA Mars rovers
- Wi-Fi access points like Apple AirPort Extreme and Linksys WRT54G
- Aircraft like Boeing Apache attack helicopter
- Automotive like BMW iDrive system prior to 2008

VxWorks development, debug ports, and on-chip debugging are often installed or enabled, which can call functions directly

- Attackers can leverage this (Metasploit has a module to do so)
- However, can be useful for engineering and embedded forensics

A new microkernel version recently released for IoT devices

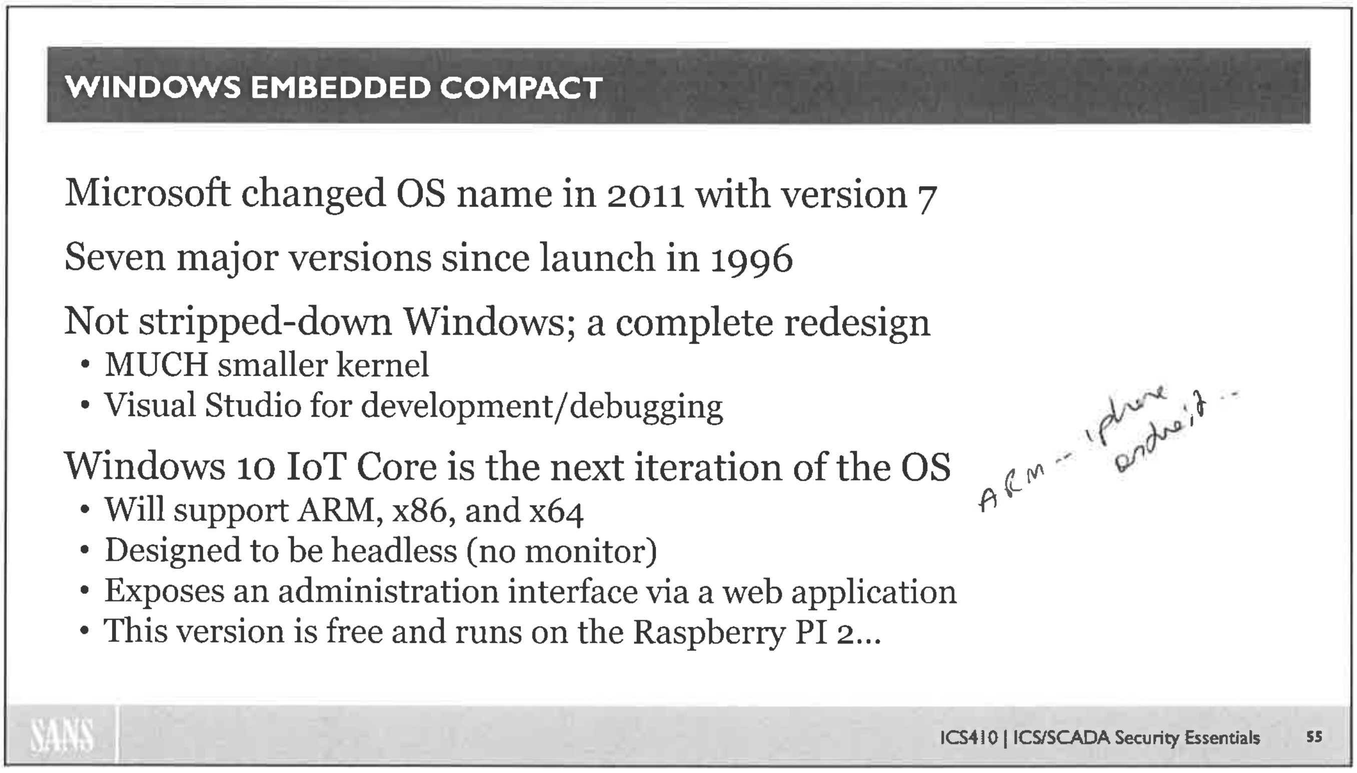
SANS ICS410 | ICS/SCADA Security Essentials 54

VxWorks is an embedded RTOS used in a variety of devices. We briefly talk about VxWorks to bring up a couple points. First, the memory model of the OS is interesting. Kernel and user space processes all have isolated memory segments. Second, VxWorks is commonly left with debug tools still installed on production systems. This is both helpful and a risk. From a defense perspective, having debug access and control on the system allows forensic collection of potentially important data related to a security incident or breach. On the other hand, it can be used by an attacker. Metasploit modules take advantage of this and were added by HD Moore over two years ago.

This commit adds functional exploits for retrieving the Apple AirPort Extreme password through the VxWorks debugger service and for setting the auto-answer flag to true for the D-Link i2Eye video conferencing system.

For more information: https://en.wikipedia.org/wiki/VxWorks

© 2019, Justin Searle



WINDOWS EMBEDDED COMPACT

Microsoft changed OS name in 2011 with version 7

Seven major versions since launch in 1996

Not stripped-down Windows; a complete redesign
- MUCH smaller kernel
- Visual Studio for development/debugging

Windows 10 IoT Core is the next iteration of the OS
- Will support ARM, x86, and x64
- Designed to be headless (no monitor)
- Exposes an administration interface via a web application
- This version is free and runs on the Raspberry PI 2...

ARM -- iphone android...

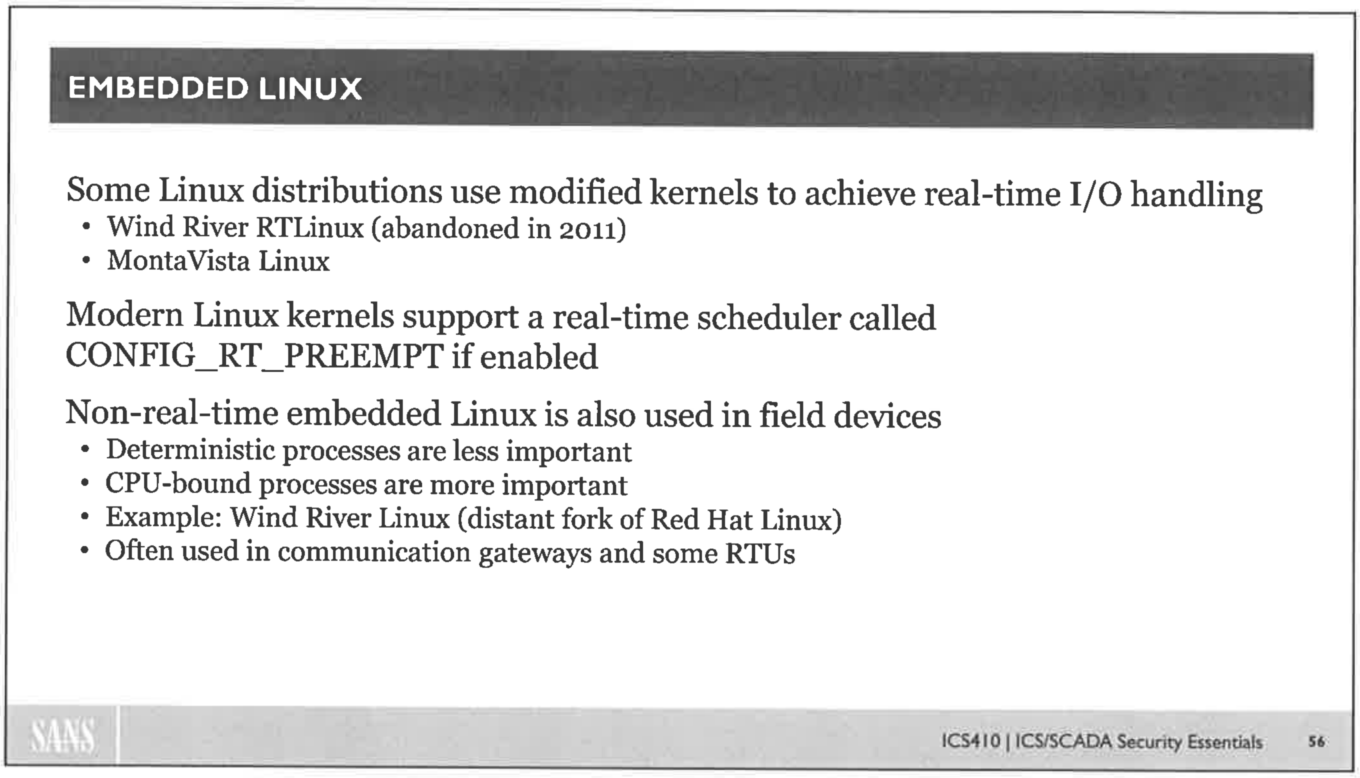
SANS ICS410 | ICS/SCADA Security Essentials 55

Windows Embedded Compact (previously called Windows CE) is Microsoft's stab at an RTOS. It is an entirely separate product with some of the code released publicly. It's similar to the other RTOSs mentioned, with the differences being things you would expect.

Development for Windows Embedded Compact is done in Visual Studio and the resulting projects can run on a multitude of devices and architectures. With modern versions of the RTOS, Microsoft has added additional components and functionality including common Windows software such as Internet Explorer.

Images for Windows 10 IoT Core for Raspberry PI can be found here:

https://www.raspberrypi.org/downloads



EMBEDDED LINUX

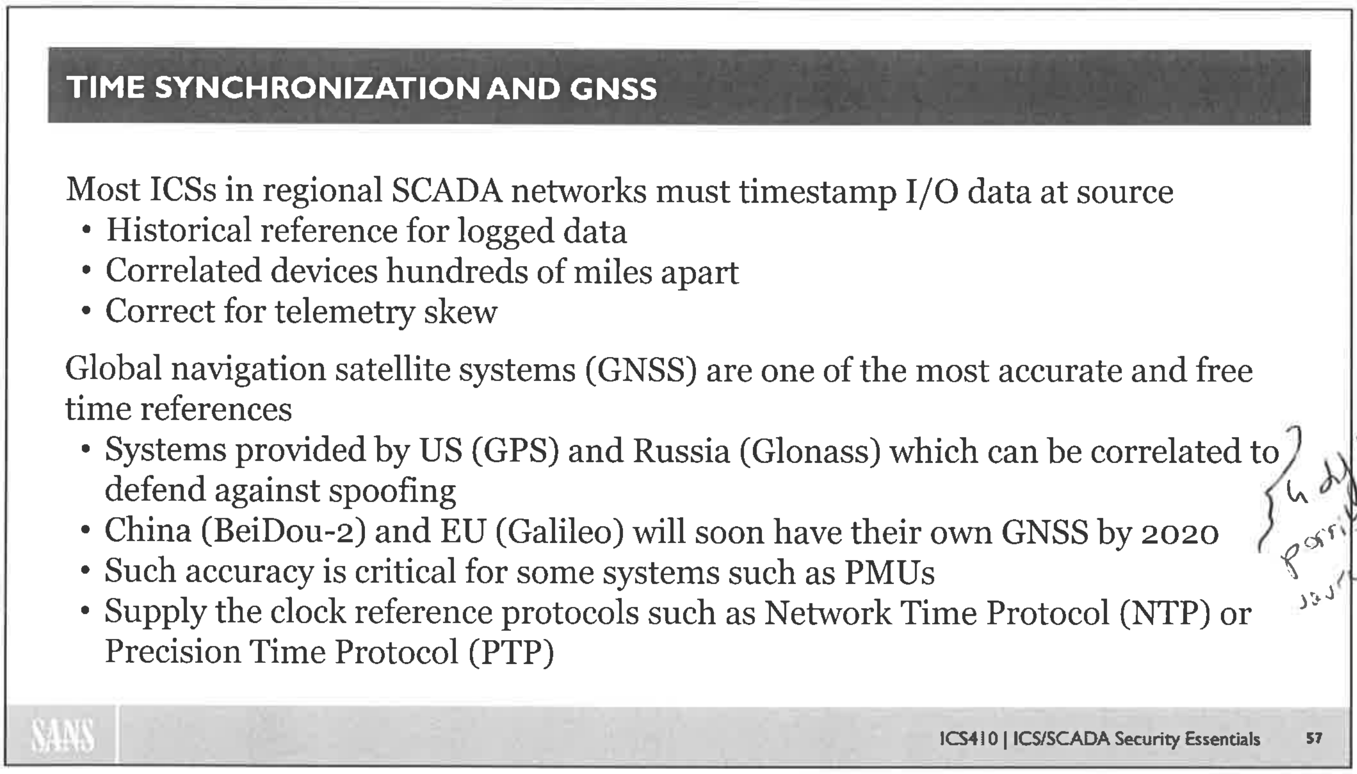
Some Linux distributions use modified kernels to achieve real-time I/O handling
- Wind River RTLinux (abandoned in 2011)
- MontaVista Linux

Modern Linux kernels support a real-time scheduler called CONFIG_RT_PREEMPT if enabled

Non-real-time embedded Linux is also used in field devices
- Deterministic processes are less important
- CPU-bound processes are more important
- Example: Wind River Linux (distant fork of Red Hat Linux)
- Often used in communication gateways and some RTUs

SANS ICS410 | ICS/SCADA Security Essentials 56

This page intentionally left blank.



TIME SYNCHRONIZATION AND GNSS

Most ICSs in regional SCADA networks must timestamp I/O data at source
- Historical reference for logged data
- Correlated devices hundreds of miles apart
- Correct for telemetry skew

Global navigation satellite systems (GNSS) are one of the most accurate and free time references
- Systems provided by US (GPS) and Russia (Glonass) which can be correlated to defend against spoofing
- China (BeiDou-2) and EU (Galileo) will soon have their own GNSS by 2020
- Such accuracy is critical for some systems such as PMUs
- Supply the clock reference protocols such as Network Time Protocol (NTP) or Precision Time Protocol (PTP)

4 diff possible sources.

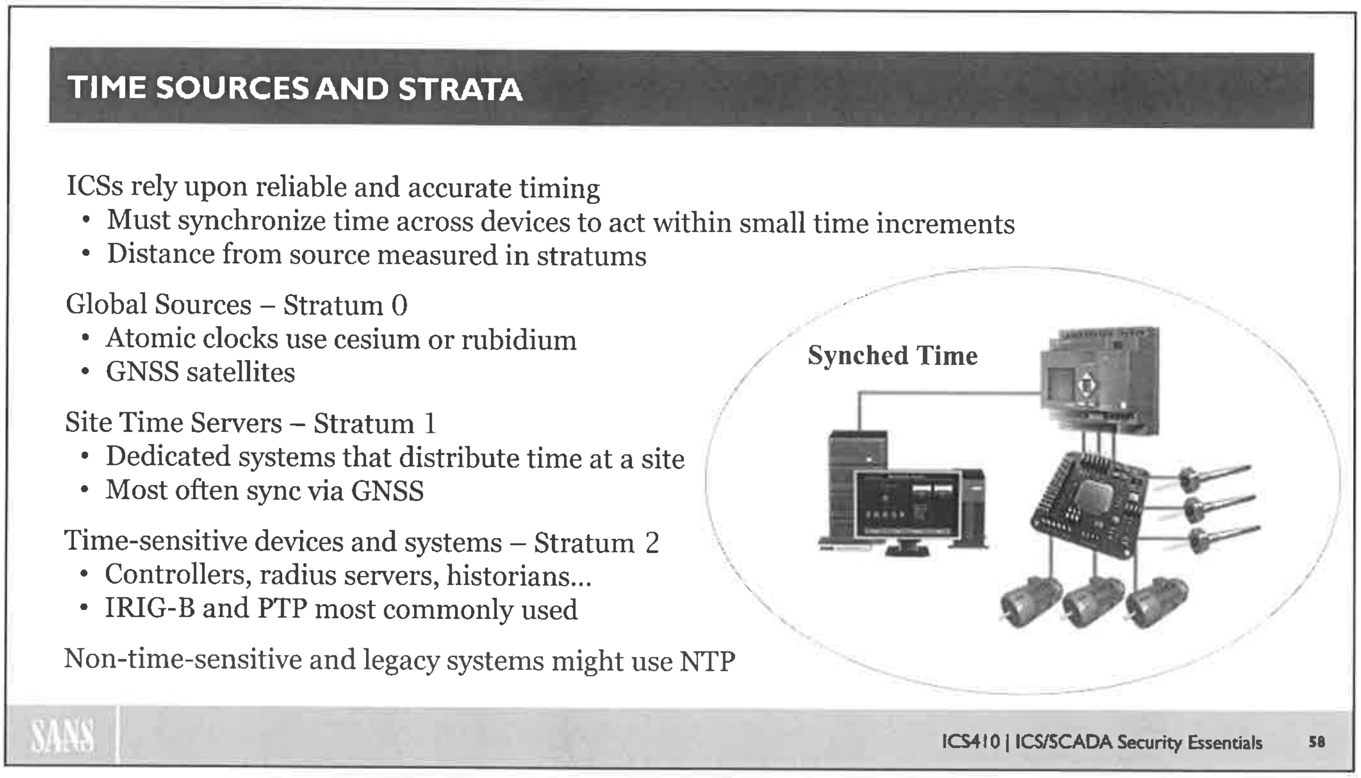
SANS ICS410 | ICS/SCADA Security Essentials 57

Time synchronization is used to timestamp events and data logged by an RTU based on an authoritative time source. Time synchronization becomes extremely important in highly distributed SCADA systems, such as the electric grid where data may need to be correlated between devices hundreds of miles away from each other and from the control center.

Global navigation satellite systems (GNSS) are widely recognized as one of the most accurate, free, and reliable time references available in remote areas. GNSS systems are provided by the US, Russia, China, and the European Union. In fact, some ICS devices reference more than one satellite system (for instance, the US and Russia) and correlate times between the two as a method to thwart GPS spoofing. These GPS sources can also provide reliable synchronized time to distributed systems that may supply the clock reference for a time synchronization system, such as Network Time Protocol (NTP), Precision Time Protocol (PTP), or a GPS receiver, which may also be directly connected to an RTU.

References:
https://en.wikipedia.org/wiki/Satellite_navigation
https://en.wikipedia.org/wiki/Global_Positioning_System
https://en.wikipedia.org/wiki/GLONASS
https://en.wikipedia.org/wiki/Galileo_(satellite_navigation)
https://en.wikipedia.org/wiki/BeiDou_Navigation_Satellite_System#BeiDou-2

Iran stole US drone by sending spoofed GPS signals and fooled it to land and crash.



TIME SOURCES AND STRATA

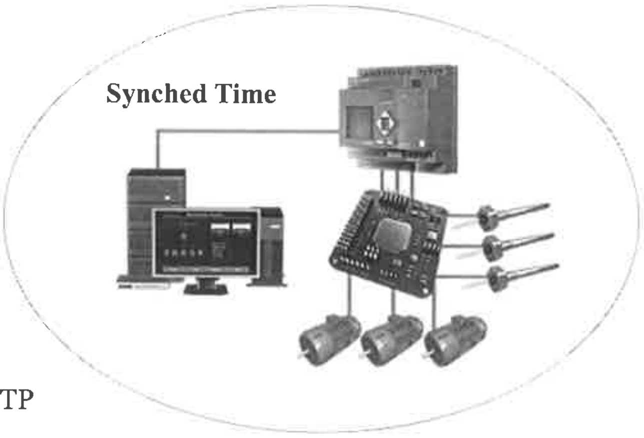
ICSs rely upon reliable and accurate timing
- Must synchronize time across devices to act within small time increments
- Distance from source measured in stratums

Global Sources – Stratum 0
- Atomic clocks use cesium or rubidium
- GNSS satellites

Site Time Servers – Stratum 1
- Dedicated systems that distribute time at a site
- Most often sync via GNSS

Time-sensitive devices and systems – Stratum 2
- Controllers, radius servers, historians...
- IRIG-B and PTP most commonly used

Non-time-sensitive and legacy systems might use NTP



SANS ICS410 | ICS/SCADA Security Essentials 58

ICSs require time services to ensure the time is accurate and synchronized throughout the system, and to ensure there is a basis for sharing timestamped information. Time service is composed of sources, services/protocol, and connectivity. Time sources provide an accurate timing input and need to be available. The most common true time source is GPS. Time servers (providing time services) can be multipurpose network servers, dedicated network servers, or dedicated devices (such as a GPS module in a PMU). A dedicated time server simply provides an accurate time. The most common protocol used for time signals is the venerable Network Time Protocol (NTP) and the newer Precision Time Protocol (PTP); however, some systems use other vendor-specific protocols.

The term "stratum" is used to provide a relative measure of closeness to a central or high-quality time server or reference time source. The labeling system uses a scale that goes from 0 (more accurate and closest) to 15 (less accurate or most distant). A stratum 0 may be a hardware device with a highly accurate time source that is attached to the computer or device's main processing board. ICSs have requirements by application and technology for a certain level of stratum. (They can reject strata that are too high.)

***Warning: Stratum levels are not a measure of reliability or direct quality measure. Some Stratum 2 sources are preferred over some Stratum 1 sources.**

Stratum 0: These are high-precision, time-keeping devices such as atomic clocks that use cesium or rubidium, GPS clocks, or other types of radio clocks. These are referred to as "reference clocks." The US Naval Observatory is an example of an organization that maintains a referenced time signal for US government applications and use.

Stratum 1: These are computing devices that are synchronized to within microseconds of an attached Stratum 0 device.
Stratum 2: These are computing devices that are synchronized over a network to Stratum 1 servers.
Stratum 3: These are computing devices that are synchronized over a network.
Stratum 15: These have the largest acceptable "distance" with a time server.
Stratum 16: This is used to indicate NO accurate time or to indicate that it is unsynchronized to a reference or updated time signal.

© 2019, Justin Searle

There is a hierarchy of time servers available for applications and public use. The first-level time servers are primarily intended to act as source time servers for second-level time servers. The first-level time servers may also be capable of providing mission-critical time services. Some first-level time servers may have a restricted access policy. There are second-level servers available for more consumer or wide (public) use.

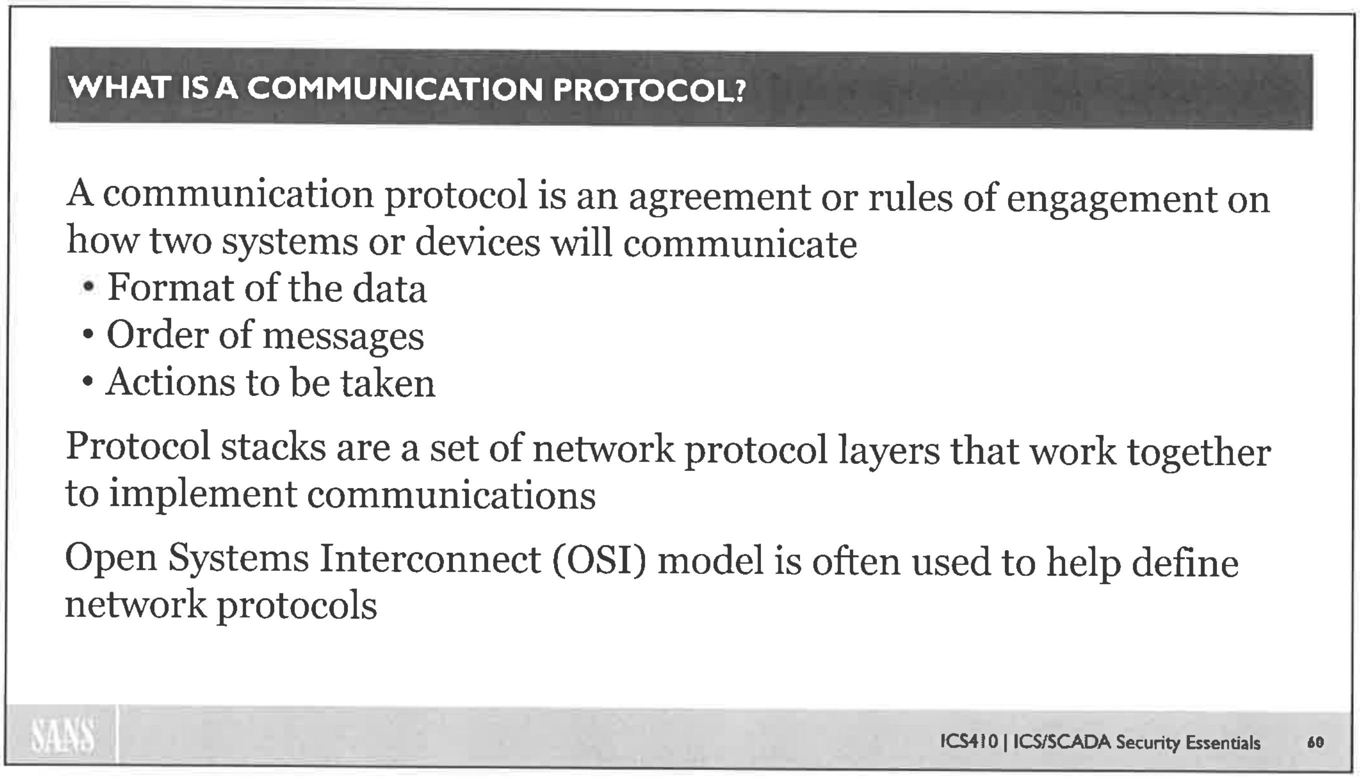
The United States Department of Commerce through the National Institute of Standards (NIST) provides Internet Time Service (ITS). This is a shared system as one central point authority managing multiple servers; it provides time signals as a service to many subscribers. Security experts have identified an attack against time services as a concern for global internet-based attacks. NTP servers are susceptible to man-in-the-middle attacks unless packets are cryptographically signed. (This is usually what is meant when a time server is advertised as having security features.) The use of cryptography can be difficult if your links are very small or with high volume and heavy processing. Internet-based or network-provided time signals can be vulnerable to Denial-of-Service (DoS) attacks.

The global address *time.nist.gov* is resolved to all the available server addresses in a round-robin sequence to equalize the load across all the NIST servers.

References:
http://tf.nist.gov/tf-cgi/servers.cgi

For the list of stratum one-time servers, visit:
http://support.ntp.org/bin/view/Servers/StratumOneTimeServers

For the list of stratum two-time servers, visit:
https://support.ntp.org/bin/view/Servers/StratumTwoTimeServers



WHAT IS A COMMUNICATION PROTOCOL?

A communication protocol is an agreement or rules of engagement on how two systems or devices will communicate
- Format of the data
- Order of messages
- Actions to be taken

Protocol stacks are a set of network protocol layers that work together to implement communications

Open Systems Interconnect (OSI) model is often used to help define network protocols

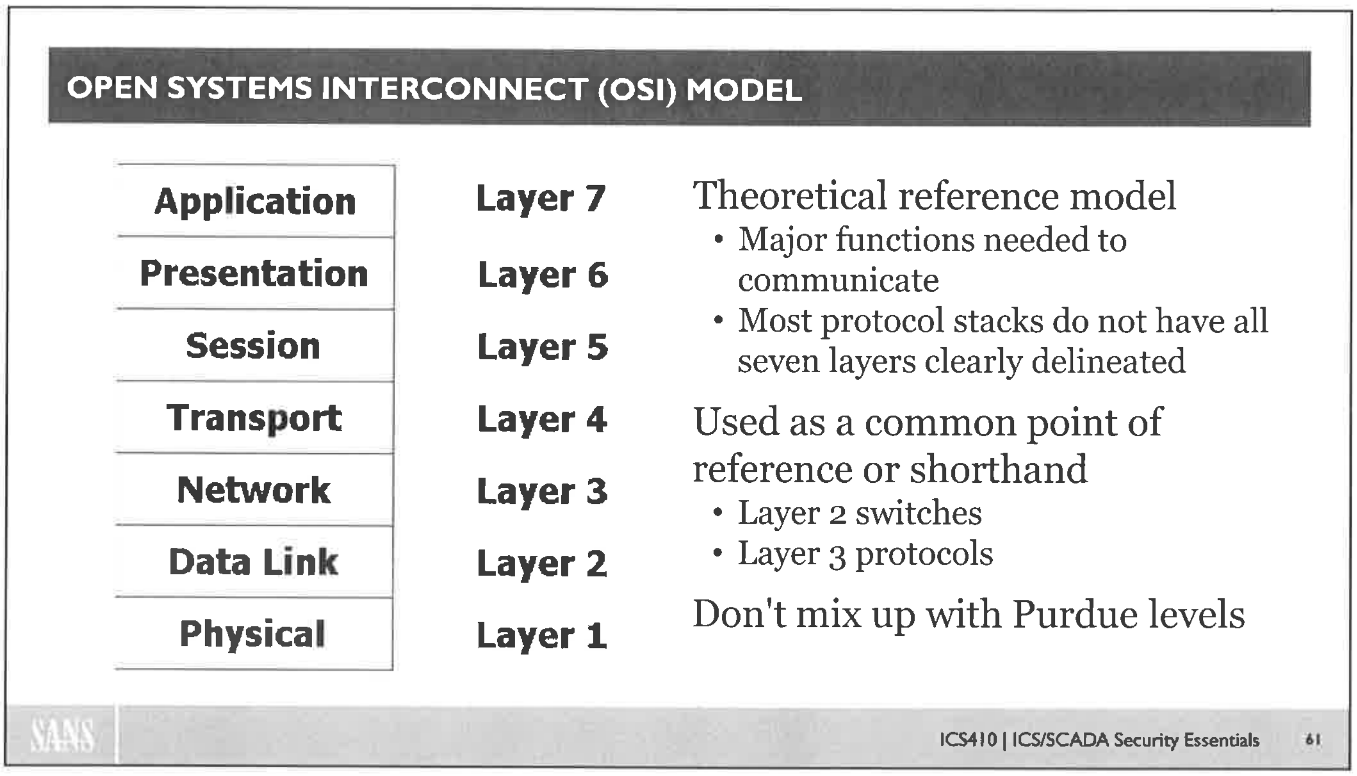
SANS ICS410 | ICS/SCADA Security Essentials 60

In the broadest sense, a protocol is nothing more than an agreement of how different entities will act and react in certain circumstances. A communications/network protocol establishes an agreement between network entities, such as hosts and servers, on how they will communicate. When protocols are worked out in advance, they are effective and efficient. If any participant breaks the protocol, the communication gets confused or can break down altogether. You have probably been in a situation in which you had "interoperable" hardware or software products that were based on the same standards but were not actually compatible. Odds are, one or both of those products deviated from the standard and implemented the protocol differently. This is why we require strict conformance to standard protocols.

There are three basic purposes for communications protocols:
- To standardize the format of a communication
- To specify the order or timing of communication
- To allow all parties to determine the meaning of a communication

As long as both sides of the communication use the same protocol and implement it properly, communication will be successful.

If two computers want to communicate, they need to follow a specific set of protocols for communications to succeed. There are a number of protocols involved. Some protocols concern themselves with breaking up a transmission into smaller bunches of data called packets. Some make sure that each packet has the proper information in the proper locations. Others describe how information is copied from your computer to the network cable. Still others ensure that packets all get to the right place in the proper order. Even with a transaction as seemingly simple as fetching a webpage, a number of protocols are required to allow the communication to succeed. In computer communications, these layered protocols are referred to as a protocol stack.



OPEN SYSTEMS INTERCONNECT (OSI) MODEL

Application	Layer 7
Presentation	Layer 6
Session	Layer 5
Transport	Layer 4
Network	Layer 3
Data Link	Layer 2
Physical	Layer 1

Theoretical reference model
- Major functions needed to communicate
- Most protocol stacks do not have all seven layers clearly delineated

Used as a common point of reference or shorthand
- Layer 2 switches
- Layer 3 protocols

Don't mix up with Purdue levels

ICS410 | ICS/SCADA Security Essentials 61

The standard reference model for protocol stacks is the International Standards Organization's (ISO) Open Systems Interconnect (OSI) model. The OSI model divides network communications into seven layers.

The physical layer handles transmission across the physical media. This includes such things as electrical pulses on wires, light pulses on fiber, connection specifications between the interface hardware and the network cable, and voltage regulation.

The data link layer connects the physical part of the network (cables and electrical signals) with the abstract part (packets and data streams).

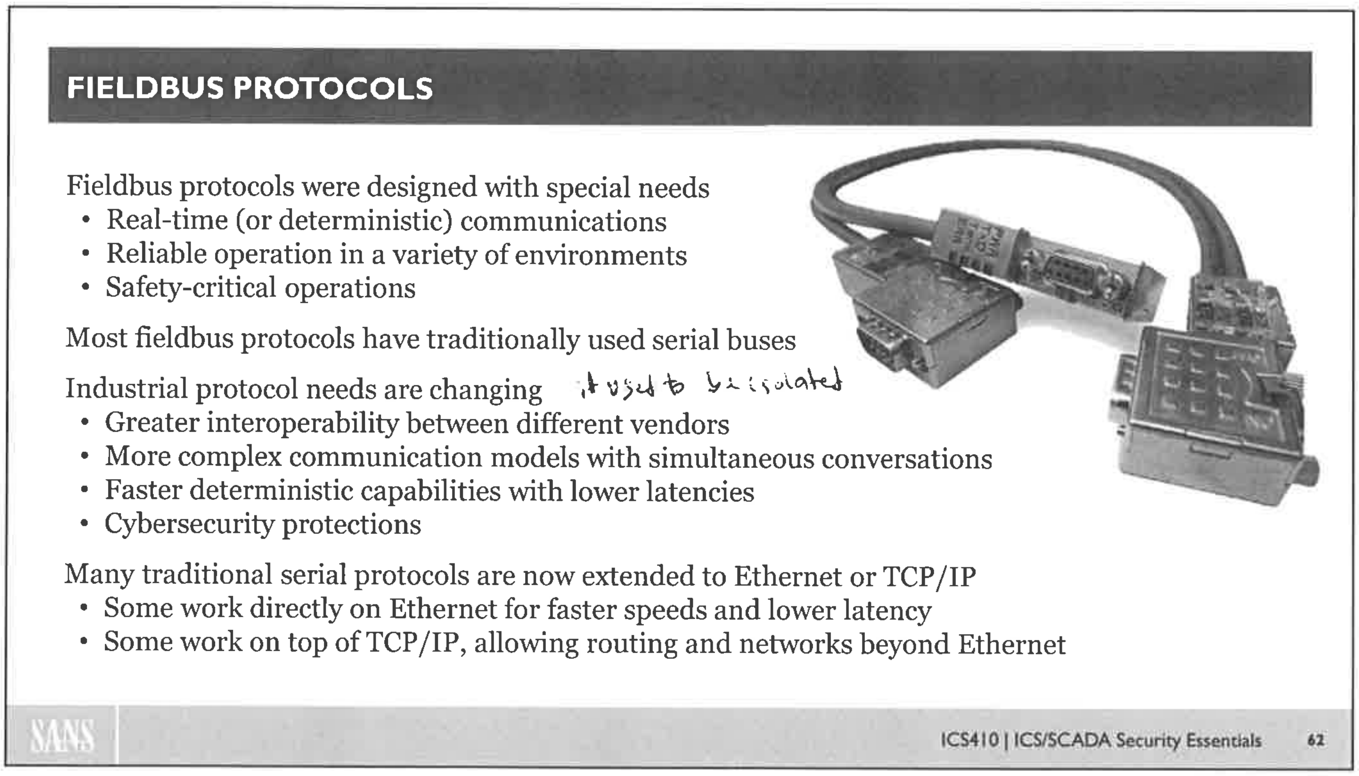
The network layer handles the network address scheme and connectivity of multiple network segments. It describes how systems on different network segments find and communicate with each other.

The transport layer interacts with your data and prepares it to be transmitted across the network. It is this layer that ensures reliable connectivity from end to end. The transport layer also handles the sequencing of packets in a transmission.

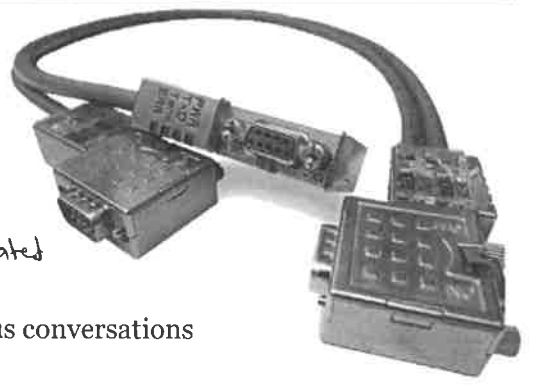
The session layer handles the establishment and maintenance of connections between systems. It negotiates the connection, sets it up, maintains it, and makes sure that information exchanged across the connection is in sync on both sides.

The presentation layer makes sure that the data sent from one side of the connection is received in a format that is useful to the other side. For example, if the sender compresses the data prior to transmission, the presentation layer on the receiving end would have to decompress it before the receiver could use it.

The application layer interacts with the application to determine which network services will be required. When a program requires access to the network, the application layer manages requests from the program to the other layers down the stack.



FIELDBUS PROTOCOLS



Fieldbus protocols were designed with special needs
- Real-time (or deterministic) communications
- Reliable operation in a variety of environments
- Safety-critical operations

Most fieldbus protocols have traditionally used serial buses

Industrial protocol needs are changing it used to be isolated
- Greater interoperability between different vendors
- More complex communication models with simultaneous conversations
- Faster deterministic capabilities with lower latencies
- Cybersecurity protections

Many traditional serial protocols are now extended to Ethernet or TCP/IP
- Some work directly on Ethernet for faster speeds and lower latency
- Some work on top of TCP/IP, allowing routing and networks beyond Ethernet

SANS ICS410 | ICS/SCADA Security Essentials 62

Some Ethernet-inspired and several serial protocols were wrapped or designed to leverage the Ethernet infrastructure to manage connections between devices (hosts, PLCs, instruments, actuators, CNC machines, and more). Simply put, an industrial protocol such as Modbus is designed to send its frames over an Ethernet network using the TCP/IP protocol. Several industrial protocols have been made routable this way to include Modbus TCP, ProfiNet, EtherNet/IP, DNP3, POWERLINK, EtherCAT, and more. These Ethernet-capable protocols can all route across the same infrastructure, but if you need them to talk to each other, you will need to implement a bridging device, such as a protocol converter, for example.

The automation industry has been embracing Ethernet infrastructure to achieve greater performance and productivity through the following:

- Increasing the accessibility of information
- Increasing system availability
- Lowering support cost
- Achieving faster designs and implementations and greater flexibility
- Enabling remote monitoring and optimization services

Connectedness is a powerful driver. Common Ethernet offers increased connectedness, enabling the following advancements:

1. Machine-to-machine and machine-to-cloud communications using wireless transport layers and public infrastructure for further connection of mobile devices and remote hosts. (GE describes this effort as going "man-to-machine.")
2. Connected data to support enhanced analytics and easier integration into business enterprise systems.
3. Cloud-based services (remote monitoring) and optimization. Ethernet can serve as the backbone for the Industrial Internet concept and opening up the Internet of Things. This debate was sharpened during the US smart grid standards development process.

Image Source:
http://blog.idxonline.com/2013/10/profibus-connectors.html

© 2019, Justin Searle

Ethernet benefits include:

- Ease of connecting devices and hosts (simple)
- Enables simultaneous communications with multiple hosts
- Widely understood and supported (many of the protocols are "open" and include toolkits and more)
- Achieves logical and functional segmentation
- Use of SNMP to monitor and manage network devices
- Can establish VLANs and networked zones
- Implements QoS to prioritize network traffic
- Better integration and support for wireless
- Many standard Ethernet tools require standard chipsets; not tied to specific vendor equipment (interoperability)
- Can be addressed over the internet (both good and bad...)
- Lower cost

History and Expanding Use over Time
Ethernet was first used to connect automation systems to company networks for the purposes of more easily sharing data. The non-deterministic nature of Ethernet limited its penetration into the control systems. EtherNet/IP (IP stands for Industrial Protocol, not Internet Protocol) was designed to overcome some of those challenges to better align with the needs of industrial automation and control. EtherNet/IP was used to connect isolated pockets of automation from batch, continuous process, discrete, safety, and drive control from one plant into a TCP networked system. The use of industrial Ethernet moved topologies from silos to converged tiered designs. The plant network was able to embrace segmentation based on function, locations, and security domains while being able to more easily integrate into operational support and business systems. Several major vendors are moving toward fully converged road maps by using unmodified Ethernet. The final penetration is the use of EtherNet/IP down to the actual sensor or smart device. Ethernet-based networks can be found connecting controllers to variable-speed drives, actuators, flow meters, and so on.

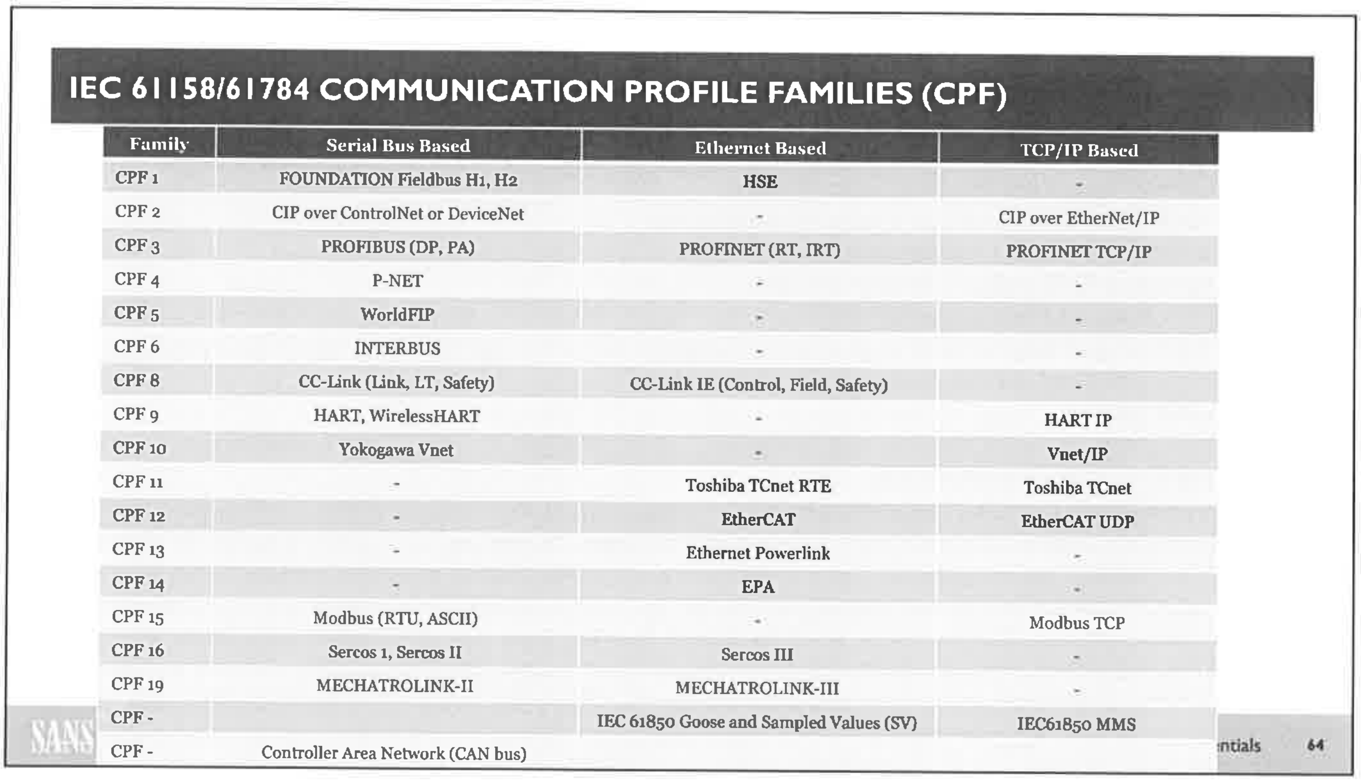
Ethernet devices are made for specific use in industrial environments and have features to make them intrinsically safe and rugged enough to meet the demands of the plant environment. Advances are allowing for linear and ring topologies to achieve greater efficiencies with multiple devices on one line or in banks. As discussed on Day 1, common standards such as IEEE's 802.3, IP, Common Industrial Protocol (CIP), Modbus, ProfiNet, and more can be designed in reference architectures that take advantage of the ability to segment, enforce security, and architect remote access. The frameworks are established in the Purdue Reference Model for Control Hierarchy, ISA-95 Enterprise-Control System Integration, and ISA-99's Control Systems Security.

Industrial Ethernet features include data rates that can be defined by the control engineer to best meet the needs of the process. The performance depends on the network and the hardware. (High-speed Ethernet switches can guarantee performance to achieve low, millisecond speeds.) For example, EtherNet/IP uses a Requested Packet Interval Rate (RPI) and can achieve 5 ms to 10 seconds.

Security and Ethernet
Cybersecurity researchers have been testing Ethernet-capable protocol implementations. There has been a growing concern as ICS devices that are IP-enabled are finding their way onto the internet in large numbers and are more integrated into enterprise business networks. Vulnerabilities can be found in the underlying protocols (session application layers) and in their specific implementations of the communication interfaces. Many of the IP stacks also have issues. Use of modern fuzzing tools has been uncovering a number of errors to be addressed by vendors.

The use of Internet Protocol at the lower layers does enable the use of IP-designed security technologies such as IPsec for IPv6 (industrial Ethernet suppliers have not announced plans to move to IPv6) and supports technologies such as SSL, TLS, and SSH.

References:
Network Infrastructure for EtherNet/IP™: Introduction and Considerations:
http://www.odva.org/Portals/0/Library/Publications_Numbered/PUB00035R0_Infrastructure_Guide.pdf
http://www.modbus.org/faq.php

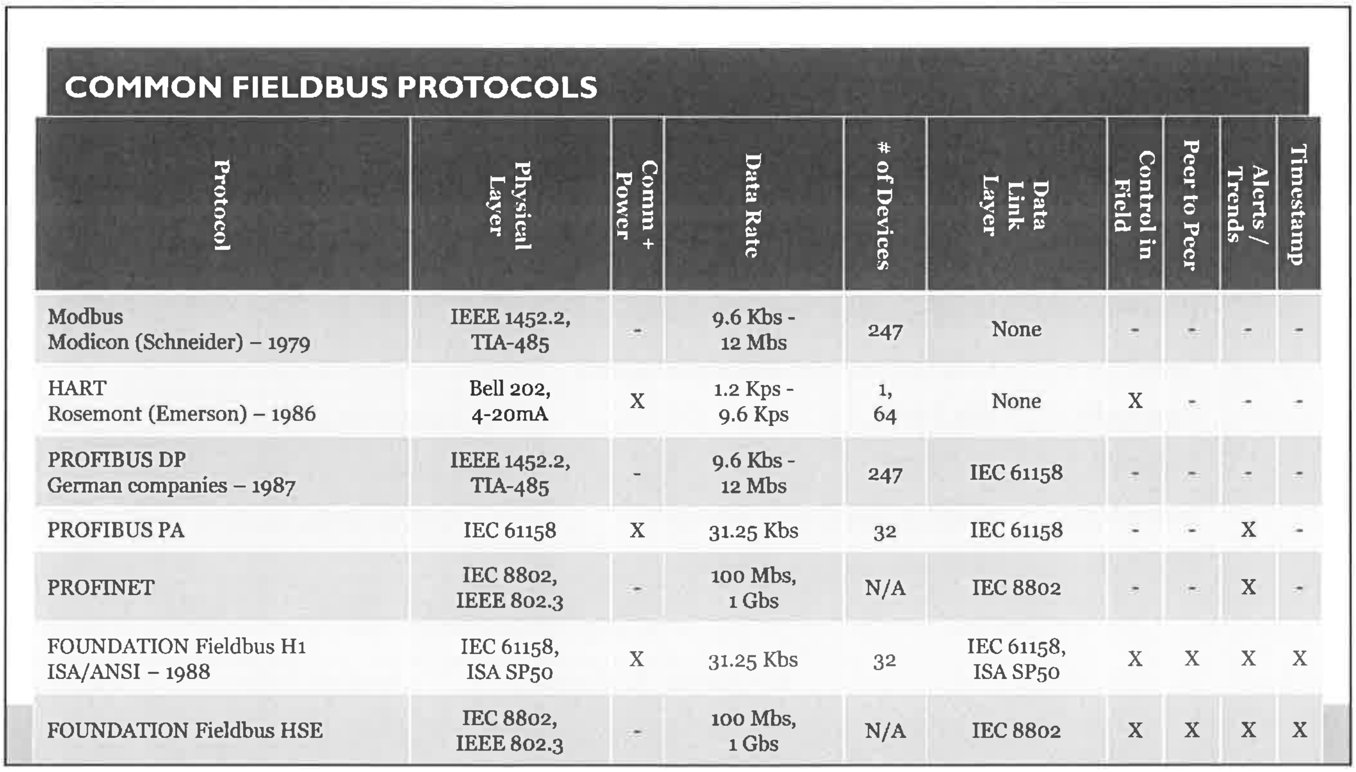


IEC 61158/61784 COMMUNICATION PROFILE FAMILIES (CPF)

Family	Serial Bus Based	Ethernet Based	TCP/IP Based
CPF 1	FOUNDATION Fieldbus H1, H2	HSE	-
CPF 2	CIP over ControlNet or DeviceNet	-	CIP over EtherNet/IP
CPF 3	PROFIBUS (DP, PA)	PROFINET (RT, IRT)	PROFINET TCP/IP
CPF 4	P-NET	-	-
CPF 5	WorldFIP	-	-
CPF 6	INTERBUS	-	-
CPF 8	CC-Link (Link, LT, Safety)	CC-Link IE (Control, Field, Safety)	-
CPF 9	HART, WirelessHART	-	HART IP
CPF 10	Yokogawa Vnet	-	Vnet/IP
CPF 11	-	Toshiba TCnet RTE	Toshiba TCnet
CPF 12	-	EtherCAT	EtherCAT UDP
CPF 13	-	Ethernet Powerlink	-
CPF 14	-	EPA	-
CPF 15	Modbus (RTU, ASCII)	-	Modbus TCP
CPF 16	Sercos 1, Sercos II	Sercos III	-
CPF 19	MECHATROLINK-II	MECHATROLINK-III	-
CPF -		IEC 61850 Goose and Sampled Values (SV)	IEC61850 MMS
CPF -	Controller Area Network (CAN bus)		

References:
FOUNDATION Fieldbus – https://en.wikipedia.org/wiki/Foundation_Fieldbus
CIP (**C**ommon **I**ndustrial **P**rotocol) – https://en.wikipedia.org/wiki/Common_Industrial_Protocol
PROFIBUS and PROFINET – https://en.wikipedia.org/wiki/Profibus
P-NET (**P**rocess **NET**work) – http://www.p-net.org/fieldbus/fieldbus.html
FIP (**F**actory **I**nstrumentation **P**rotocol) – https://en.wikipedia.org/wiki/Factory_Instrumentation_Protocol
INTERBUS – https://en.wikipedia.org/wiki/INTERBUS
CC-Link (**C**ontrol and **C**ommunication) – https://en.wikipedia.org/wiki/CC-Link_Industrial_Networks
HART (**H**ighway **A**ddressable **R**emote **T**ransducer) – https://en.wikipedia.org/wiki/Highway_Addressable_Remote_Transducer_Protocol
Yokogawa Vnet – https://web-material3.yokogawa.com/rd-tr-r00039-005.pdf
Toshiba TCnet – https://www.toshiba.co.jp/sis/en/seigyo/tcnet/tcnet.htm
EtherCAT – https://en.wikipedia.org/wiki/EtherCAT
Ethernet Powerlink – https://en.wikipedia.org/wiki/Ethernet_Powerlink
EPA (**E**thernet for **P**lant **A**utomation) – http://cse.zju.edu.cn/english/redir.php?catalog_id=31001
Modbus – https://en.wikipedia.org/wiki/Modbus
Sercos (**SE**rial **R**eal-time **Co**mmunication **S**ystem) – https://en.wikipedia.org/wiki/SERCOS_interface
MECHATROLINK – https://en.wikipedia.org/wiki/MECHATROLINK
CAN bus – https://en.wikipedia.org/wiki/CAN_bus
IEC 61850 – https://en.wikipedia.org/wiki/IEC_61850

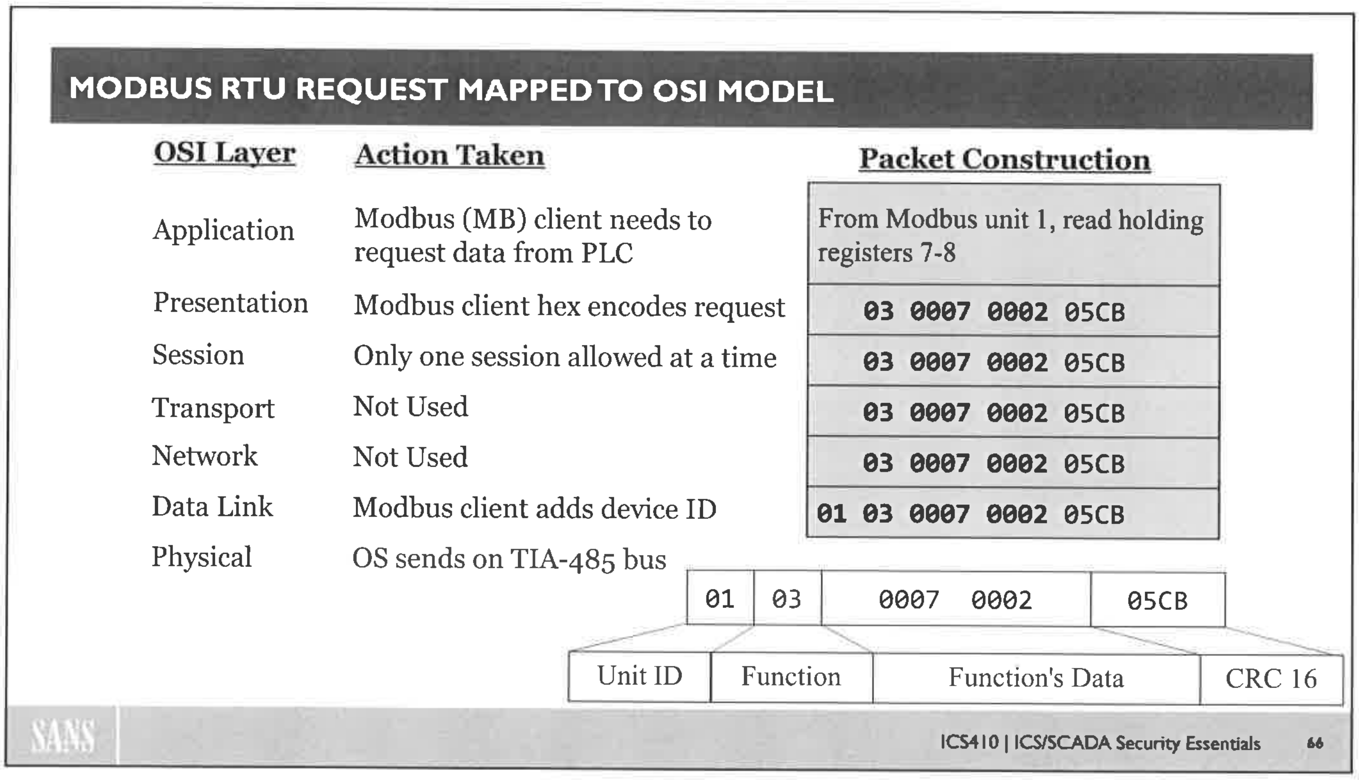
© 2019, Justin Searle



COMMON FIELDBUS PROTOCOLS

Protocol	Physical Layer	Comm + Power	Data Rate	# of Devices	Data Link Layer	Control in Field	Peer to Peer	Alerts / Trends	Timestamp
Modbus Modicon (Schneider) – 1979	IEEE 1452.2, TIA-485	-	9.6 Kbs - 12 Mbs	247	None	-	-	-	-
HART Rosemont (Emerson) – 1986	Bell 202, 4-20mA	X	1.2 Kps - 9.6 Kps	1, 64	None	X	-	-	-
PROFIBUS DP German companies – 1987	IEEE 1452.2, TIA-485	-	9.6 Kbs - 12 Mbs	247	IEC 61158	-	-	-	-
PROFIBUS PA	IEC 61158	X	31.25 Kbs	32	IEC 61158	-	-	X	-
PROFINET	IEC 8802, IEEE 802.3	-	100 Mbs, 1 Gbs	N/A	IEC 8802	-	-	X	-
FOUNDATION Fieldbus H1 ISA/ANSI – 1988	IEC 61158, ISA SP50	X	31.25 Kbs	32	IEC 61158, ISA SP50	X	X	X	X
FOUNDATION Fieldbus HSE	IEC 8802, IEEE 802.3	-	100 Mbs, 1 Gbs	N/A	IEC 8802	X	X	X	X

This page intentionally left blank.



MODBUS RTU REQUEST MAPPED TO OSI MODEL

OSI Layer	Action Taken	Packet Construction
Application	Modbus (MB) client needs to request data from PLC	From Modbus unit 1, read holding registers 7-8
Presentation	Modbus client hex encodes request	`03 0007 0002 05CB`
Session	Only one session allowed at a time	`03 0007 0002 05CB`
Transport	Not Used	`03 0007 0002 05CB`
Network	Not Used	`03 0007 0002 05CB`
Data Link	Modbus client adds device ID	`01 03 0007 0002 05CB`
Physical	OS sends on TIA-485 bus	

01	03	0007 0002	05CB
Unit ID	Function	Function's Data	CRC 16

SANS

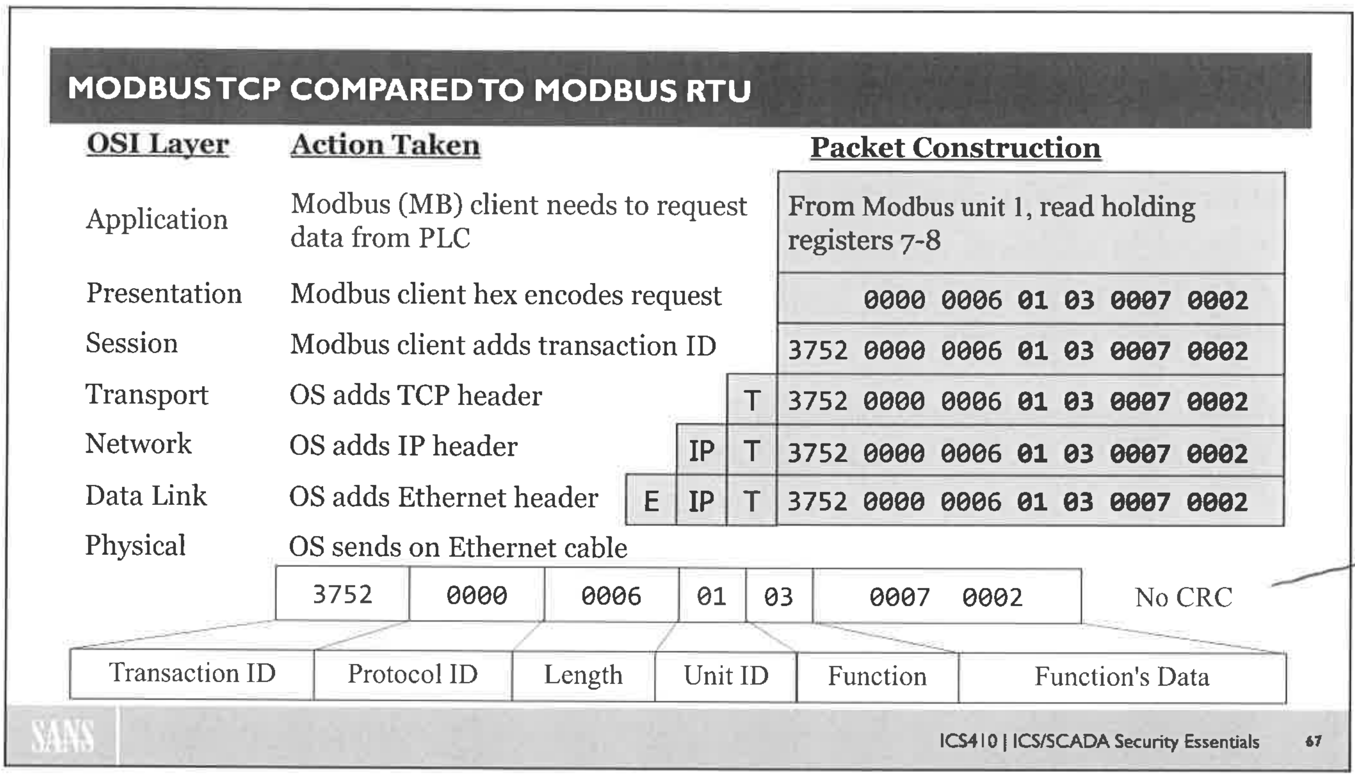
ICS410 | ICS/SCADA Security Essentials 66

It is important to understand how a packet is generated as it moves through the OSI stack. Ultimately, each layer on the sender needs to communicate with the same layer on the receiving computer. However, they cannot directly talk because you must go down the stack, across the network, and back up the stack on the receiving system. The way this is accomplished is by having each layer add a header as you go down the stack on the sender and each system remove a header on the receiving system as it goes up the stack. By performing this function, the header that is created by a given layer on the sender is received by the corresponding layer on the receiving system.

You have probably seen examples of how a TCP/IP packet maps to the OSI model, in fact the following slide shows how Modbus TCP (the TCP/IP version of modbus) maps to the OSI model. However first, lets look how a serial packet maps to the OSI model. In the example above, I have mapped a Modbus RTU serial frame to the OSI model. Notice how there aren't any headers added in the middle layers. Many serial protocols are monolithic, meaning a single protocol standard takes responsibility for all layers of the OSI model, using only the features that protocol needs.

The diagram at the bottom shows you a protocol decode of the Modbus RTU payload.

CRC only for detection
corruption



MODBUS TCP COMPARED TO MODBUS RTU

OSI Layer	Action Taken	Packet Construction
Application	Modbus (MB) client needs to request data from PLC	From Modbus unit 1, read holding registers 7-8
Presentation	Modbus client hex encodes request	`0000 0006 01 03 0007 0002`
Session	Modbus client adds transaction ID	`3752 0000 0006 01 03 0007 0002`
Transport	OS adds TCP header	T `3752 0000 0006 01 03 0007 0002`
Network	OS adds IP header	IP T `3752 0000 0006 01 03 0007 0002`
Data Link	OS adds Ethernet header	E IP T `3752 0000 0006 01 03 0007 0002`
Physical	OS sends on Ethernet cable	

3752	0000	0006	01	03	0007 0002	No CRC
Transaction ID	Protocol ID	Length	Unit ID	Function	Function's Data	

TCP He. does CRC

ICS410 | ICS/SCADA Security Essentials 67

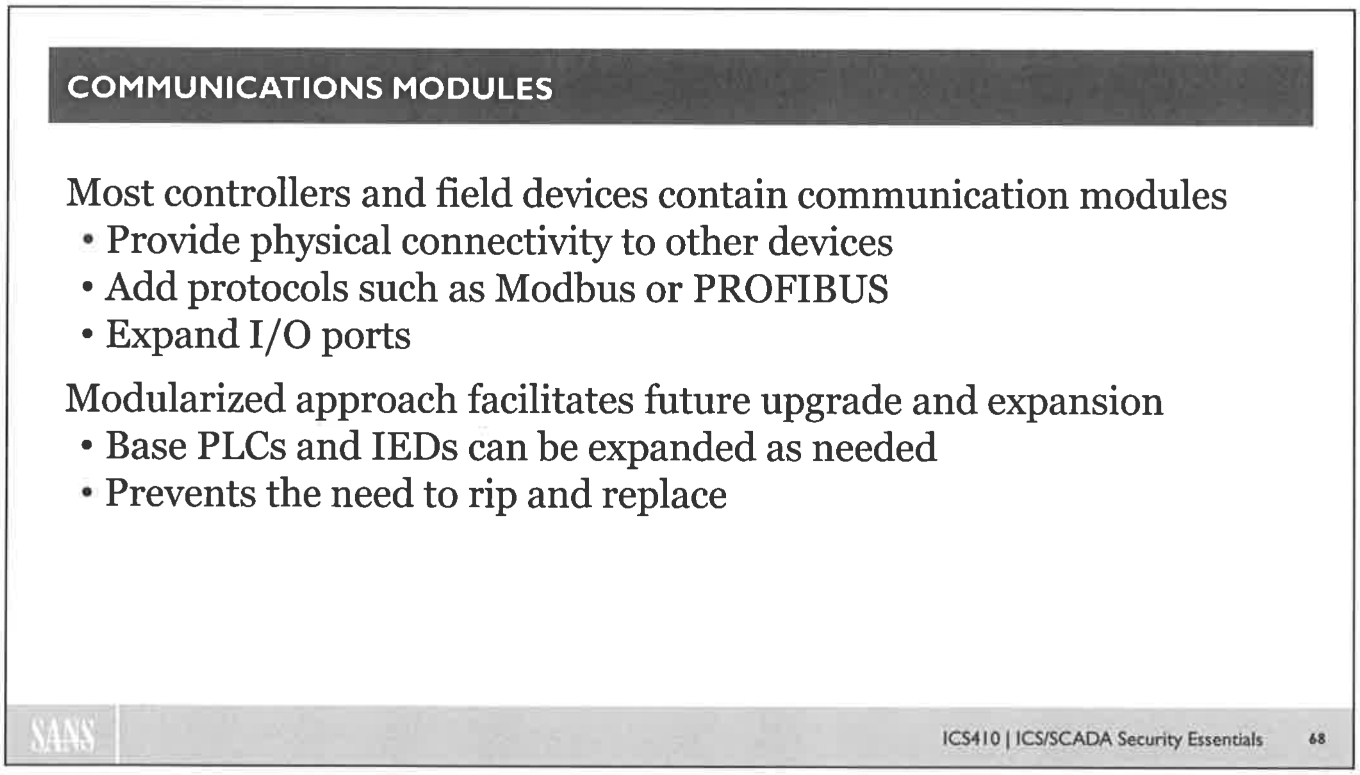
In comparison to Modbus RTU, Modbus TCP adds a few more features when being implementing on top of TCP/IP, namely the ability to support variations of the protocol and the ability to have multiple conversations at the same time with the same unit. To start, the application layer takes information from the application itself. In this case, a Modbus TCP master server will send a "read holding registers 7-8" message to a PLC, the same request we saw in the previous example with Modbus RTU. The Modbus vendor's software handles the first few layers of the OSI model, then gives the properly formatted Modbus message to the operating system's TCP/IP stack.

The operating system's TCP/IP stack takes the packet and adds a transport header to it. The header has all the information that the transport layer on the other side of the connection needs to determine what to do with the packet. After the transport header is put on the packet, it is given to the network layer.

The network layer puts another header in front of the packet. Like the transport layer before it, this header gives information for the internet layer on the other end. After this header is attached, the packet is sent to the data link layer. As you probably have guessed by now, the data link layer will put its own header on the packet. This header will assist the routers and gateways between the two machines in sending the packet along its way. After this final header is placed on the packet, it is put on the wire and sent to its final destination.

What happens when the PLC receives the data? The operation starts again, but this time in reverse. The OS begins by stripping off the Ethernet and TCP/IP headers its counterpart put on the packet in the first place and then passes the rest of the packet up to the Modbus server on the device. This process of stripping off one layer's headers and passing the rest of the packet up to the next higher layer is known as decapsulation.

The diagram at the bottom shows the changes Modbus TCP payloads made to the original Modbus RTU serial protocol it is based on. The sections in red are the changes, the sections in black are the same as the original, with some minor variations on reserved values in the Unit ID and Function values.



COMMUNICATIONS MODULES

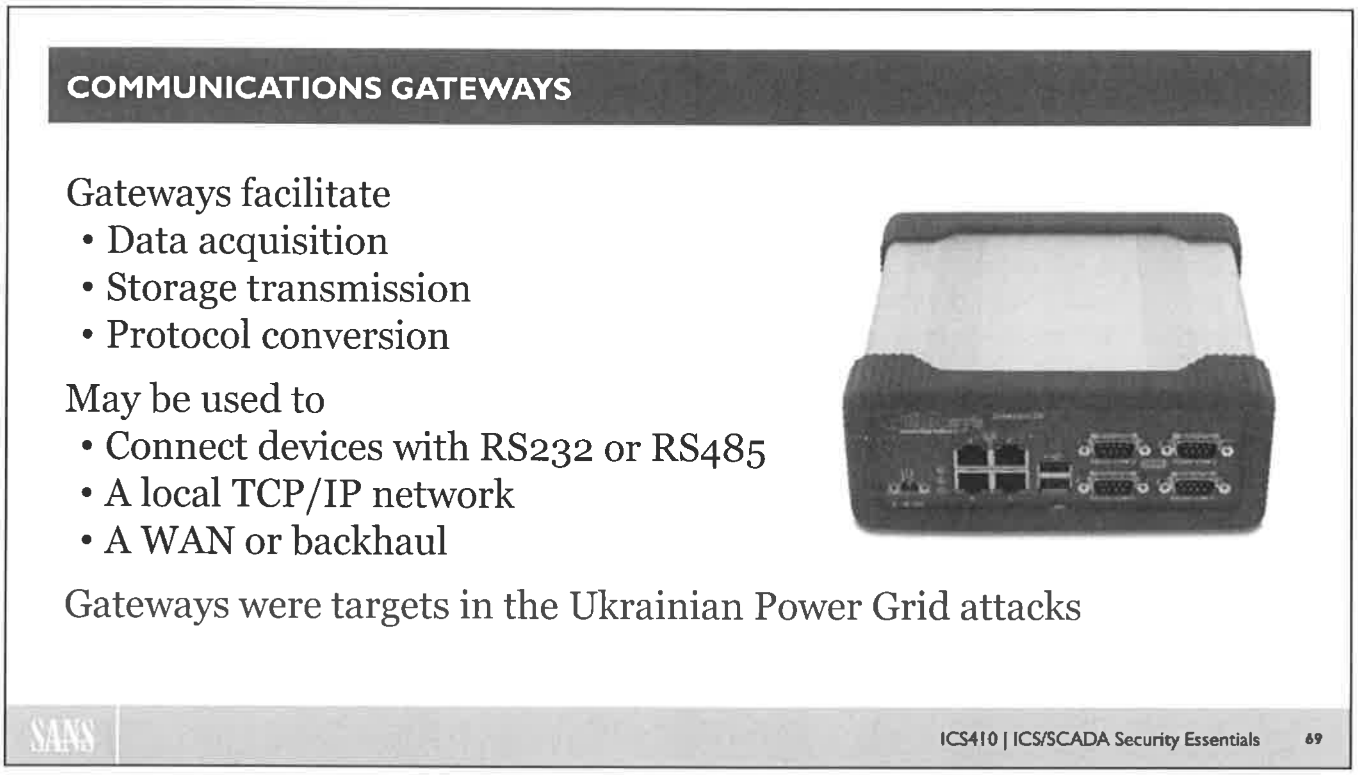
Most controllers and field devices contain communication modules
- Provide physical connectivity to other devices
- Add protocols such as Modbus or PROFIBUS
- Expand I/O ports

Modularized approach facilitates future upgrade and expansion
- Base PLCs and IEDs can be expanded as needed
- Prevents the need to rip and replace

SANS ICS410 | ICS/SCADA Security Essentials 68

Field device communication modules connect the FEP into networks of intelligent electronic devices (IEDs) such as Modbus or PROFIBUS. These communication modules are separate from the FEP, as the type and quantity of interfaces required will differ greatly from one site to another. This modularized approach also allows end users to upgrade or expand their ICS without having to initially purchase PLCs sized for the worst-case scenario.

© 2019, Justin Searle

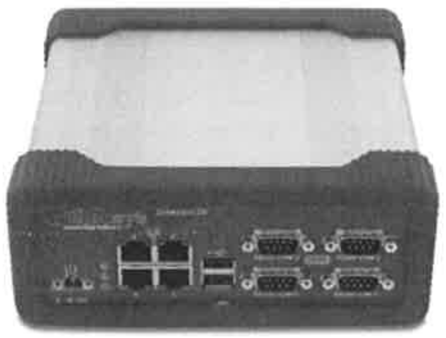


COMMUNICATIONS GATEWAYS

Gateways facilitate
- Data acquisition
- Storage transmission
- Protocol conversion

May be used to
- Connect devices with RS232 or RS485
- A local TCP/IP network
- A WAN or backhaul

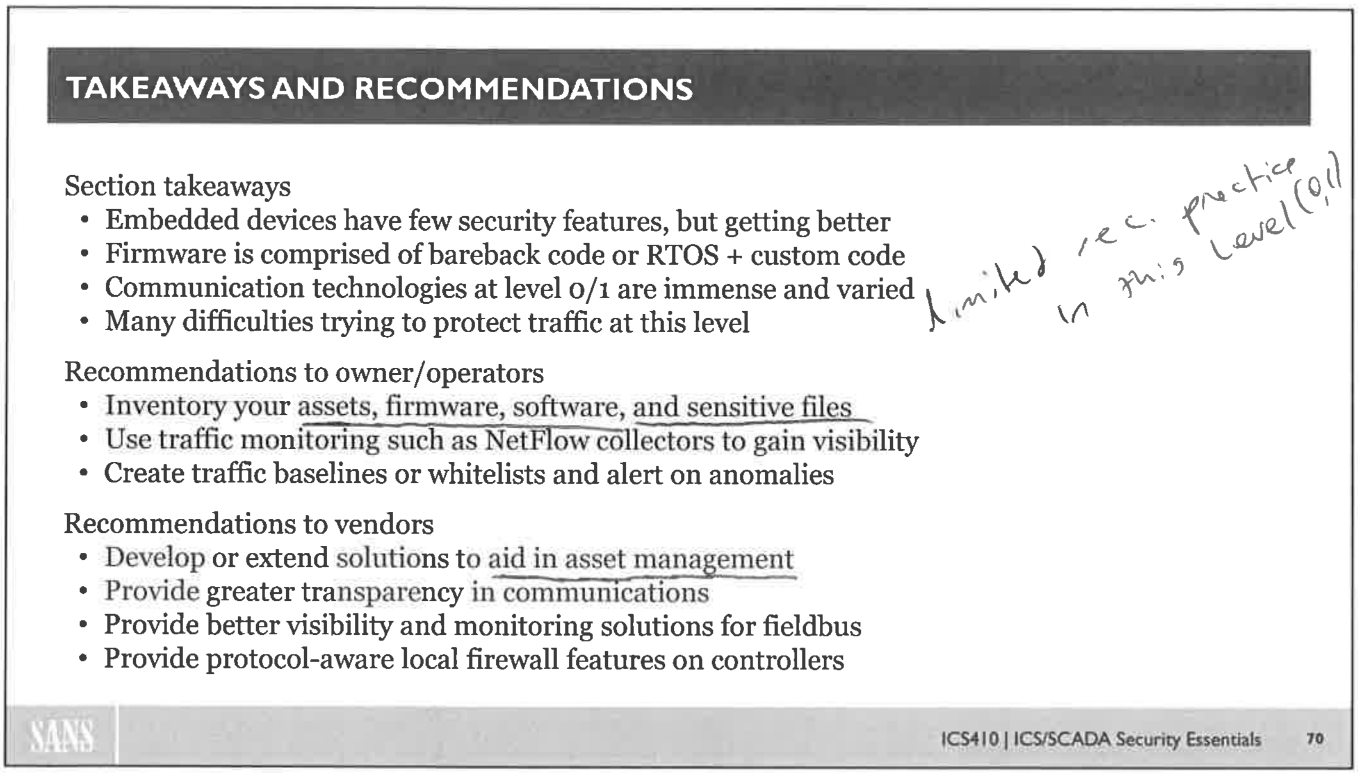
Gateways were targets in the Ukrainian Power Grid attacks



SANS ICS410 | ICS/SCADA Security Essentials 69

A communications gateway facilitates data acquisition, storage, transmission, and protocol conversion throughout the industrial data network. These devices may be used to connect IEDs that have only RS232 or RS485 to a TCP/IP network. They may also be used where communications need to be backhauled via cellular or other wireless networks, such as mesh wireless or licensed radio networks.

Communications gateways are widely used to apply timestamping to data and logs for locally connected RTUs, allowing for several RTUs to be connected to a single communications gateway and share the same time source.



TAKEAWAYS AND RECOMMENDATIONS

Section takeaways
- Embedded devices have few security features, but getting better
- Firmware is comprised of bareback code or RTOS + custom code
- Communication technologies at level 0/1 are immense and varied
- Many difficulties trying to protect traffic at this level

limited rec. practice in this Level (0,1)

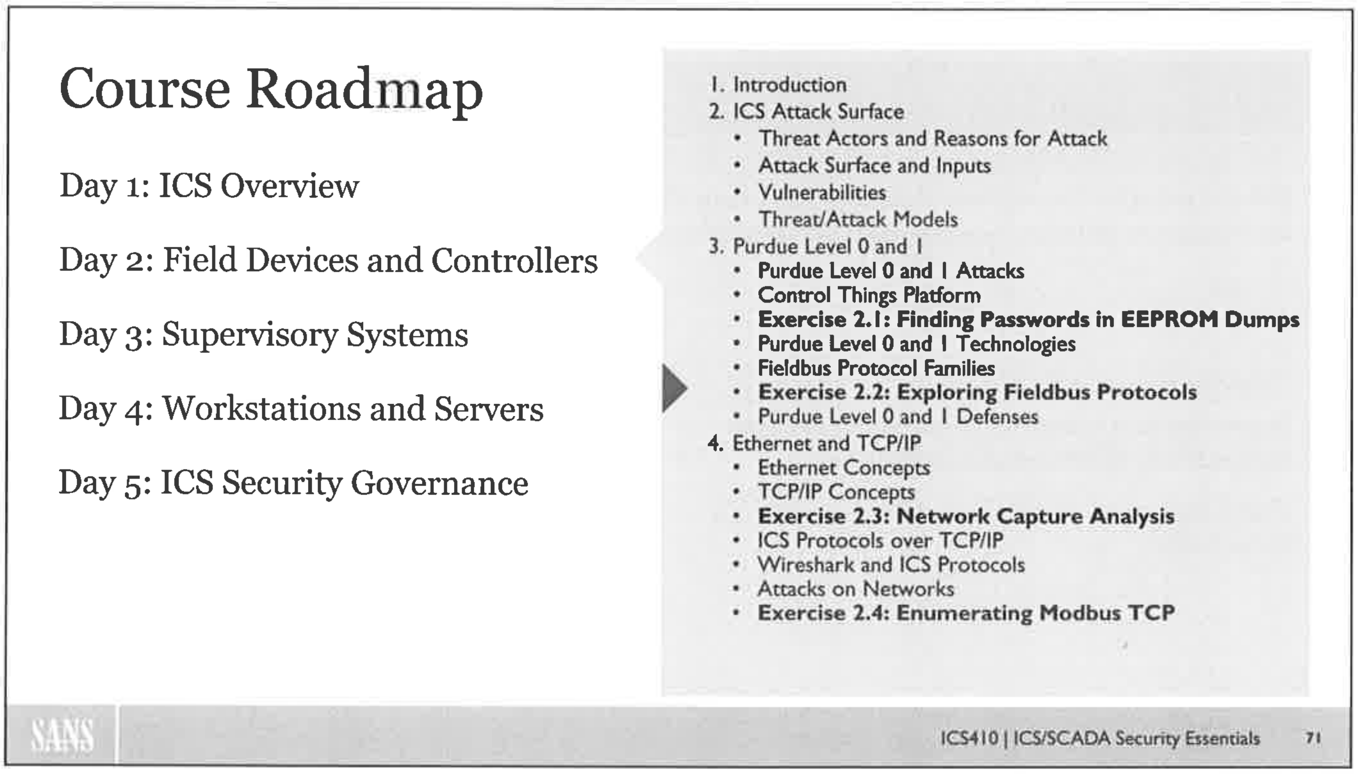
Recommendations to owner/operators
- Inventory your assets, firmware, software, and sensitive files
- Use traffic monitoring such as NetFlow collectors to gain visibility
- Create traffic baselines or whitelists and alert on anomalies

Recommendations to vendors
- Develop or extend solutions to aid in asset management
- Provide greater transparency in communications
- Provide better visibility and monitoring solutions for fieldbus
- Provide protocol-aware local firewall features on controllers

SANS ICS410 | ICS/SCADA Security Essentials 70

This page intentionally left blank.

© 2019, Justin Searle


Course Roadmap

Day 1: ICS Overview

Day 2: Field Devices and Controllers

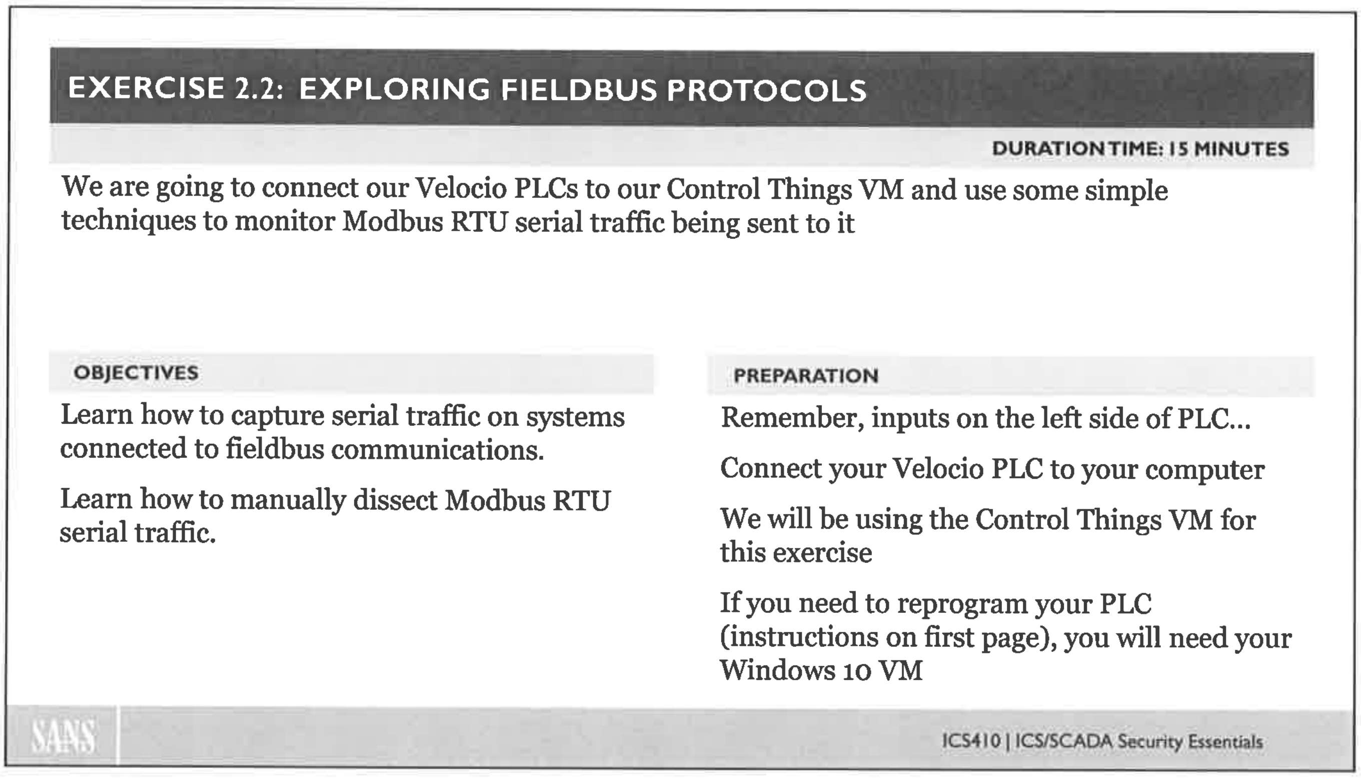
Day 3: Supervisory Systems

Day 4: Workstations and Servers

Day 5: ICS Security Governance

SANS ICS410 | ICS/SCADA Security Essentials 71

This page intentionally left blank.



EXERCISE 2.2: EXPLORING FIELDBUS PROTOCOLS

DURATION TIME: 15 MINUTES

We are going to connect our Velocio PLCs to our Control Things VM and use some simple techniques to monitor Modbus RTU serial traffic being sent to it

OBJECTIVES

Learn how to capture serial traffic on systems connected to fieldbus communications.

Learn how to manually dissect Modbus RTU serial traffic.

PREPARATION

Remember, inputs on the left side of PLC...

Connect your Velocio PLC to your computer

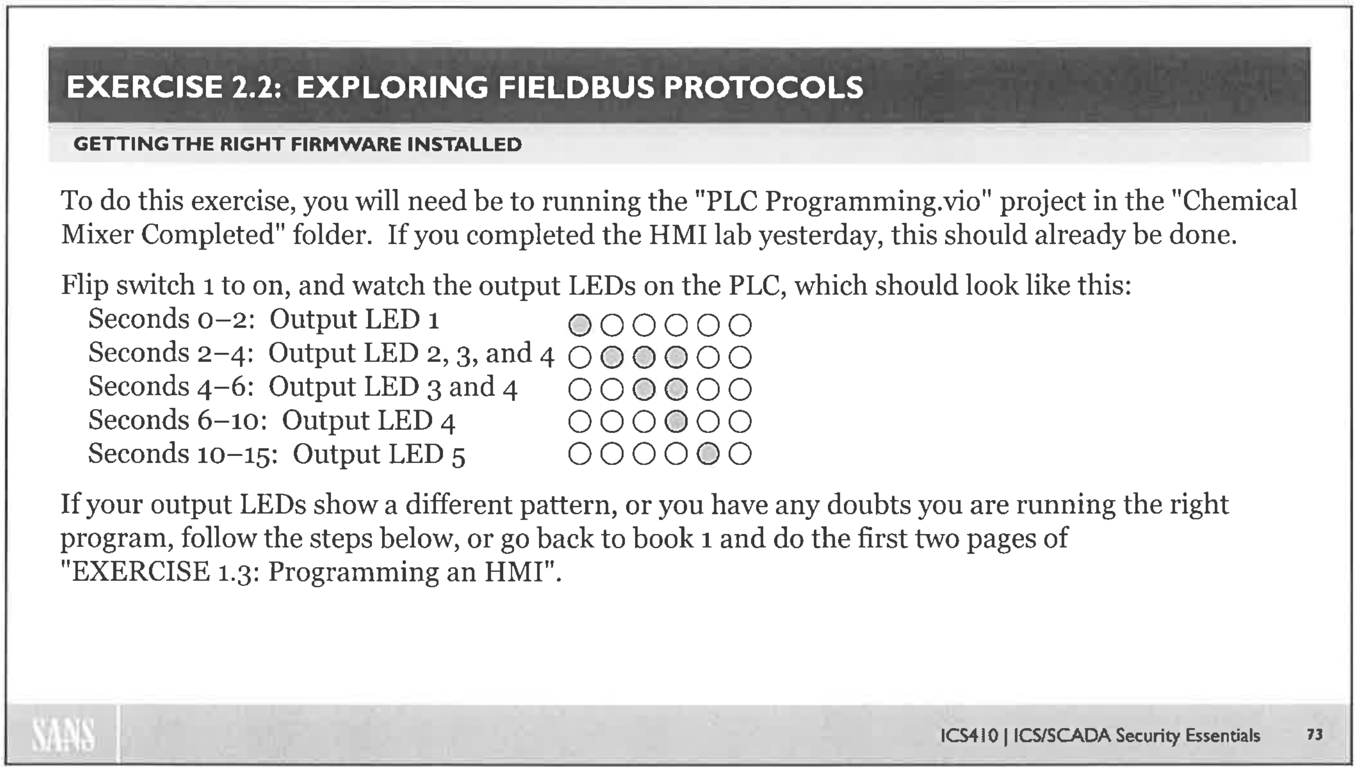
We will be using the Control Things VM for this exercise

If you need to reprogram your PLC (instructions on first page), you will need your Windows 10 VM

ICS410 | ICS/SCADA Security Essentials

This page intentionally left blank.

© 2019, Justin Searle



EXERCISE 2.2: EXPLORING FIELDBUS PROTOCOLS

GETTING THE RIGHT FIRMWARE INSTALLED

To do this exercise, you will need be to running the "PLC Programming.vio" project in the "Chemical Mixer Completed" folder. If you completed the HMI lab yesterday, this should already be done.

Flip switch 1 to on, and watch the output LEDs on the PLC, which should look like this:

Seconds 0–2: Output LED 1	● ○ ○ ○ ○ ○
Seconds 2–4: Output LED 2, 3, and 4	○ ● ● ● ○ ○
Seconds 4–6: Output LED 3 and 4	○ ○ ● ● ○ ○
Seconds 6–10: Output LED 4	○ ○ ○ ● ○ ○
Seconds 10–15: Output LED 5	○ ○ ○ ○ ● ○

If your output LEDs show a different pattern, or you have any doubts you are running the right program, follow the steps below, or go back to book 1 and do the first two pages of "EXERCISE 1.3: Programming an HMI".

SANS ICS410 | ICS/SCADA Security Essentials 73

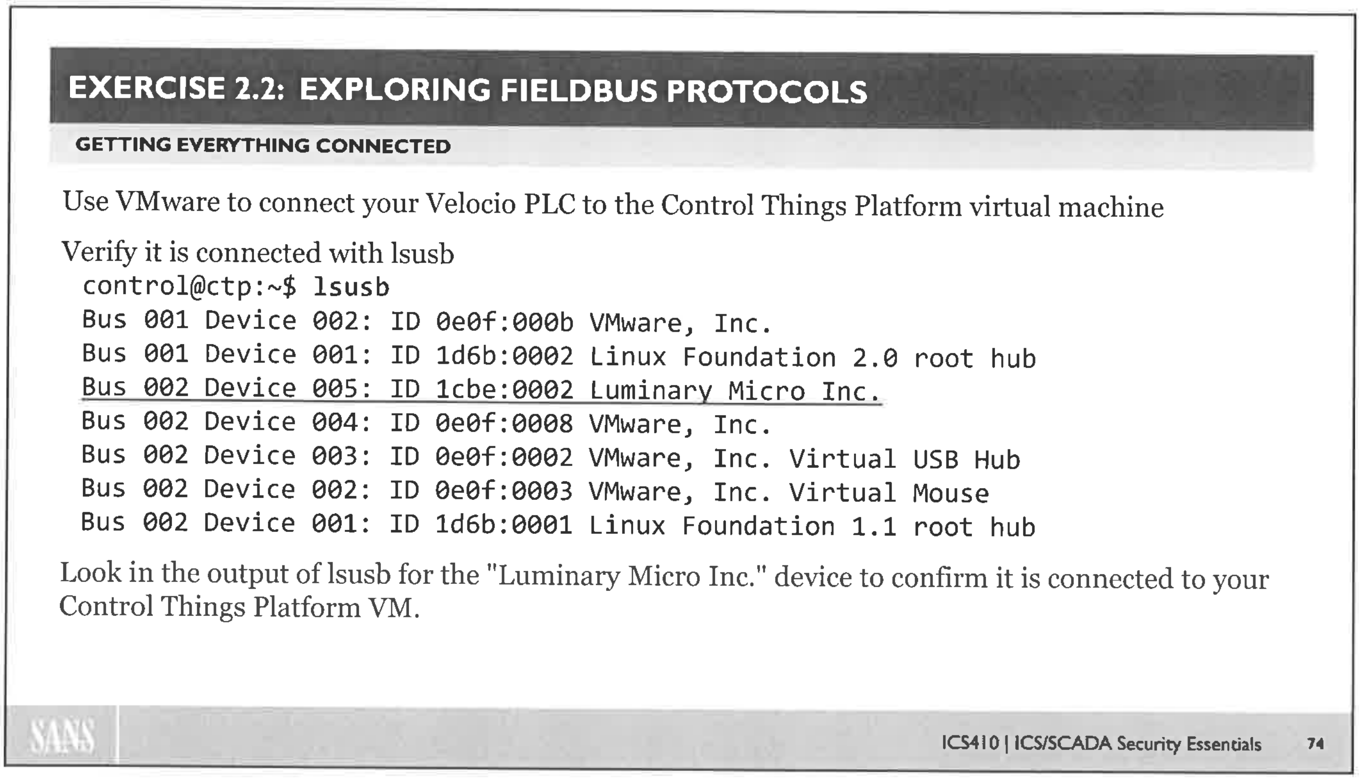
To do this exercise, you will need be to running the "PLC Programming.vio" project in the "Chemical Mixer Completed" folder. If you completed the HMI lab yesterday, this should already be done. If you are not sure if you completed that exercise correctly, or believe you are using the project you created yesterday instead of the "Chemical Mixer Completed" project provided to you, please follow these steps to reprogram your PLC. This one has all tags remotely writable so our HMI can change their values. It also has Modbus RTU enabled for a later lab. The program you want to program to the PLC is the "Chemical Mixer Completed" folder in the "Velocio PLC" folder on your desktop.

- Start your Windows 10 VM.
- Open the "Velocio PLC" folder on the desktop of your Windows 10 VM.
- Open the "Chemical Mixer Completed" folder.
- Open the "PLC Programming" file that has the Velocio logo.
- Answer "Yes" to the prompt about vBuilder modifying your computer.

Velocio needs to be able to install drivers to access your PLC and program your PLC, so it needs a bit more permission than most programs; thus, you will see this prompt every time you start Velocio. Now we need to prepare our PLC to connect to our computers.

- Turn off all the switches on the input switchboard (away from the numbers).
- Connect your input switchboard on the **LEFT SIDE** of your PLC.
- Use the USB cable to connect your PLC to the computer.
- Make sure VMware connects the PLC to your Windows 10 VM.
- Program the PLC by clicking on the Velocio icon in vBuilder.
- Run the program by clicking the Play button.
- On the PLC, flip switch 1 to on, and watch the output LEDs start to cycle on the PLC.

If you have any problems programming the PLC, go back to EXERCISE 1.2 "Programming an HMI" in Book 1 of this course for more detailed instructions on programming your PLC.



EXERCISE 2.2: EXPLORING FIELDBUS PROTOCOLS

GETTING EVERYTHING CONNECTED

Use VMware to connect your Velocio PLC to the Control Things Platform virtual machine

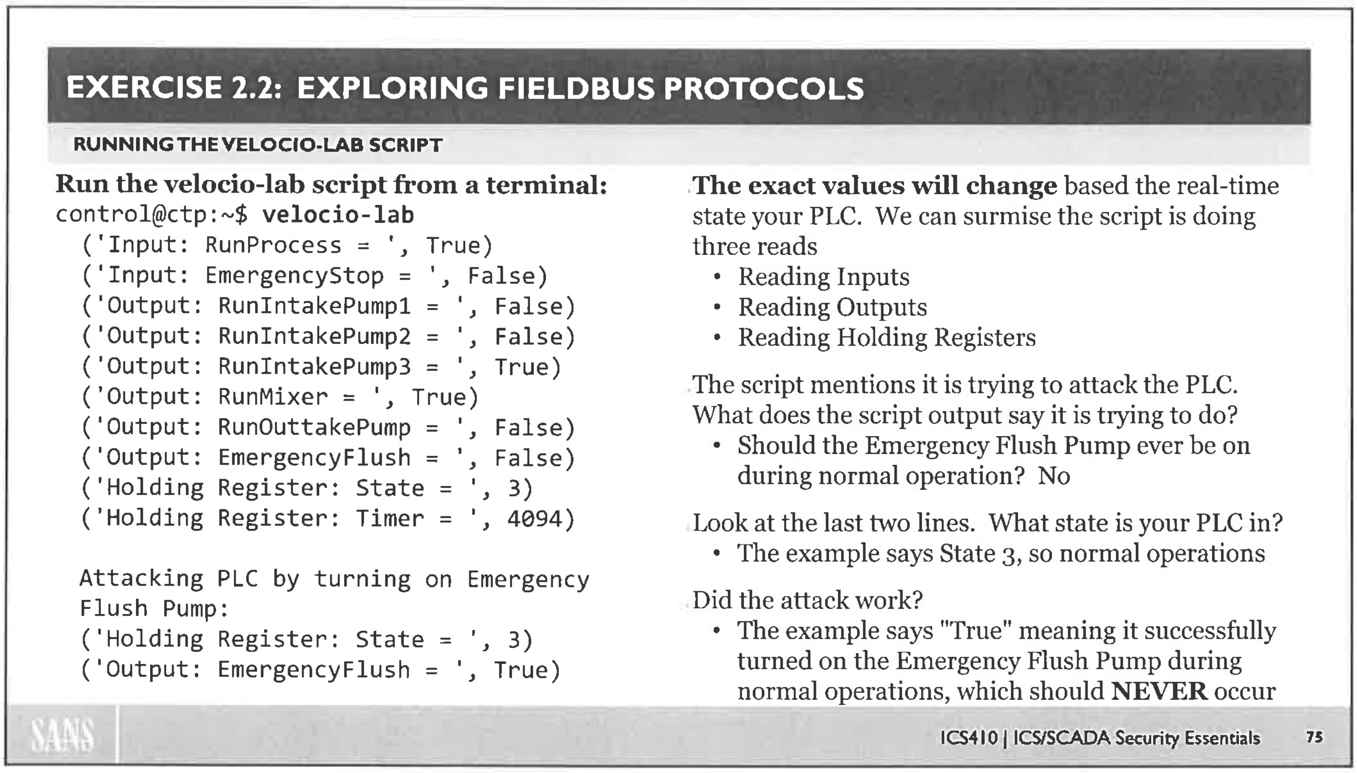
Verify it is connected with lsusb

```
control@ctp:~$ lsusb
Bus 001 Device 002: ID 0e0f:000b VMware, Inc.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 005: ID 1cbe:0002 Luminary Micro Inc.
Bus 002 Device 004: ID 0e0f:0008 VMware, Inc.
Bus 002 Device 003: ID 0e0f:0002 VMware, Inc. Virtual USB Hub
Bus 002 Device 002: ID 0e0f:0003 VMware, Inc. Virtual Mouse
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
```

Look in the output of lsusb for the "Luminary Micro Inc." device to confirm it is connected to your Control Things Platform VM.

Depending on the VMware software and version of that software you are running, there will be several different methods to connect your USB devices to your virtual machines. You should have been able to figure this out during yesterday's exercises, but if not, feel free to ask your instructor, facilitator, or SME. It is usually under one of the menu items in the VMware software, often under the USB or removable devices section. You will need to connect the "Luminary Micro" USB device to your Control Things Platform virtual machine.

Once you get your PLC connected to Control Things, open a terminal and run lsusb to verify Control Things can see your PLC. If you see a line that says "Luminary Micro Inc." in your output, you are successfully connected. Look at the example in the slide above. You will also need to know which USB Bus your PLC is connected to later in this exercise. This is also in the lsbusb command output. In the example above, you can see that the PLC is device 005 on Bus 002. Write that bus number down for later use.



EXERCISE 2.2: EXPLORING FIELDBUS PROTOCOLS

RUNNING THE VELOCIO-LAB SCRIPT

Run the velocio-lab script from a terminal:

```
control@ctp:~$ velocio-lab
  ('Input: RunProcess = ', True)
  ('Input: EmergencyStop = ', False)
  ('Output: RunIntakePump1 = ', False)
  ('Output: RunIntakePump2 = ', False)
  ('Output: RunIntakePump3 = ', True)
  ('Output: RunMixer = ', True)
  ('Output: RunOuttakePump = ', False)
  ('Output: EmergencyFlush = ', False)
  ('Holding Register: State = ', 3)
  ('Holding Register: Timer = ', 4094)

  Attacking PLC by turning on Emergency
  Flush Pump:
  ('Holding Register: State = ', 3)
  ('Output: EmergencyFlush = ', True)
```

The exact values will change based the real-time state your PLC. We can surmise the script is doing three reads

- Reading Inputs
- Reading Outputs
- Reading Holding Registers

The script mentions it is trying to attack the PLC. What does the script output say it is trying to do?

- Should the Emergency Flush Pump ever be on during normal operation? No

Look at the last two lines. What state is your PLC in?

- The example says State 3, so normal operations

Did the attack work?

- The example says "True" meaning it successfully turned on the Emergency Flush Pump during normal operations, which should **NEVER** occur

SANS ICS410 | ICS/SCADA Security Essentials 75

I have created a script that interfaces with the PLC and runs some commands. The script is called velocio-lab. Run this from the terminal and read through the output to understand what it is doing.

Run the velocio-lab script from a terminal to have it communicate with your PLC

```
control@ctp:~$ velocio-lab
('Input: RunProcess = ', True)
('Input: EmergencyStop = ', False)
('Output: RunIntakePump1 = ', False)
('Output: RunIntakePump2 = ', False)
('Output: RunIntakePump3 = ', True)
('Output: RunMixer = ', True)
('Output: RunOuttakePump = ', False)
('Output: EmergencyFlush = ', False)
('Holding Register: State = ', 3)
('Holding Register: Timer = ', 4094)

Attacking PLC by turning on Emergency Flush Pump:
('Holding Register: State = ', 3)
('Output: EmergencyFlush = ', True)
```

The exact values will change based on what state your PLC is in at the time. We can surmise the script is doing three reads: Reading Inputs, Reading Outputs, Reading Holding Registers

The script mentions it is trying to attack the PLC. What does it say it is trying to do? Should the Emergency Flush Pump ever be on during normal operation? No.

Look at the last two lines. What state is yours currently in? Example says 3. Did the attack work? Example says "True" meaning it successfully turned on the Emergency Flush Pump during normal operations. That isn't good.

EXERCISE 2.2: EXPLORING FIELDBUS PROTOCOLS

CAPTURING USB TRAFFIC WITH WIRESHARK

Now let's capture the raw serial traffic with Wireshark

First we must load the usbmon kernel driver so we can sniff USB traffic

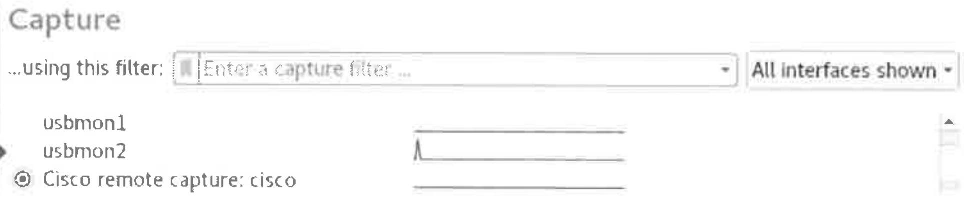
```
control@ctp:~$ sudo modprobe usbmon
[sudo] password for control: things
```

(you will not see what you type here)

Do you remember what USB bus your PLC is connected to? If not, run lsusb again...

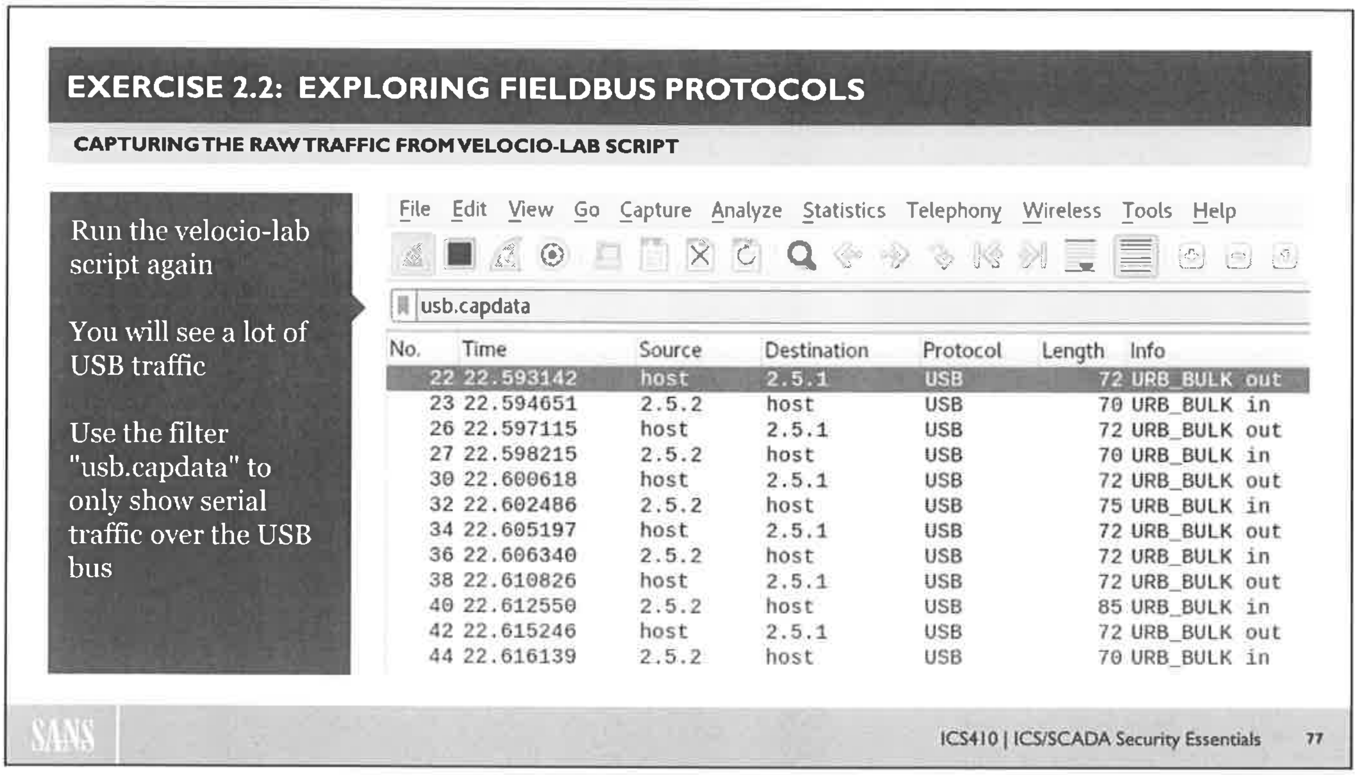
Now click on the Activities menu, search for Wireshark, and open it

Select the correct USB bus number. Mine was 002, so I'll choose usbmon2



The easiest way to capture serial and fieldbus communications is to use a tool like Wireshark on a computer participating in that communication. Wireshark and similar tools do this by capturing data entering and exiting the communication card's driver, which in the case of serial communication and fieldbus communications, is almost always represented as a simple serial device. In Windows, this would be one of the COM ports, and in Linux, it could be any number of device files. However, in Linux, if the device is connected via USB, we can use the usbmon kernel module to sniff the serial data over USB.

Look at the examples above to load the usbmon driver, start Wireshark, and select the right usbmon device to capture.



EXERCISE 2.2: EXPLORING FIELDBUS PROTOCOLS

CAPTURING THE RAW TRAFFIC FROM VELOCIO-LAB SCRIPT

Run the velocio-lab script again

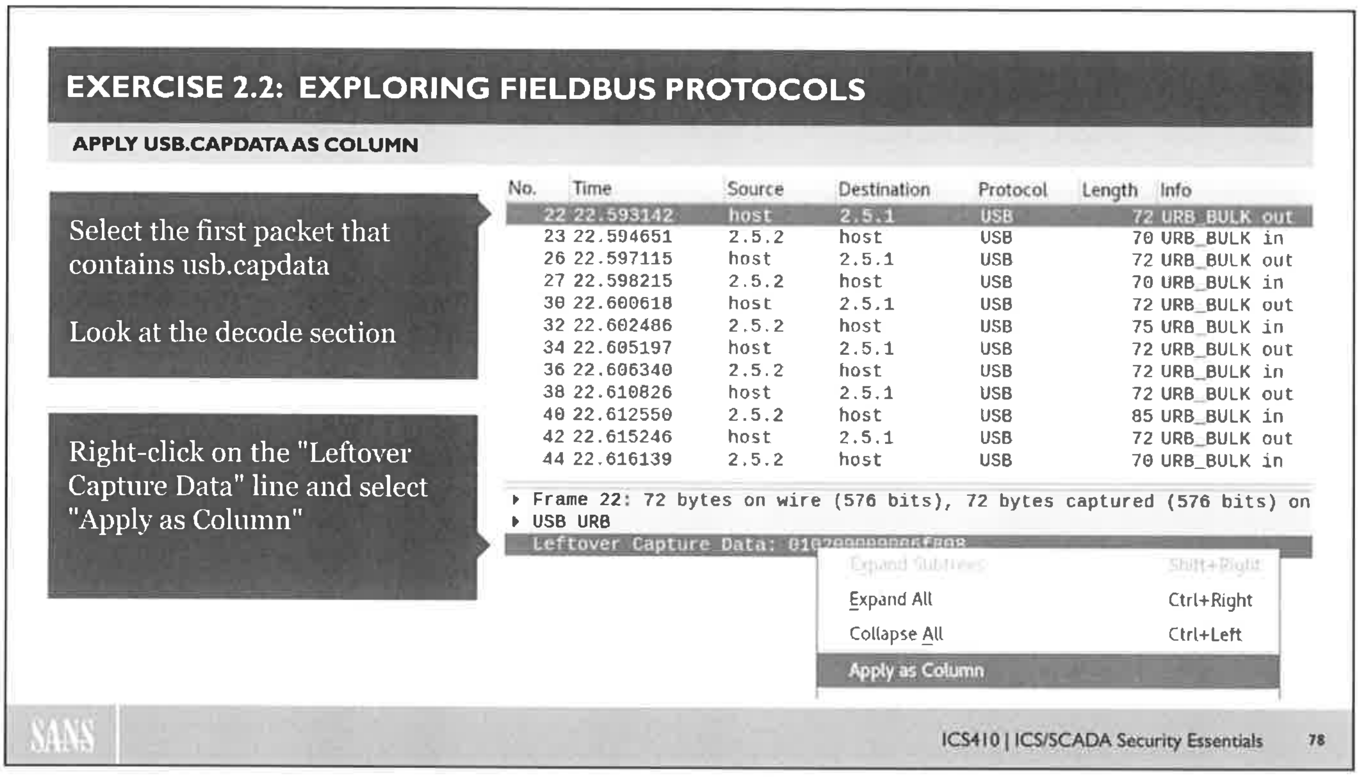
You will see a lot of USB traffic

Use the filter "usb.capdata" to only show serial traffic over the USB bus

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
usb.capdata

```
No.  Time        Source  Destination  Protocol  Length  Info
22   22.593142   host    2.5.1        USB       72      URB_BULK out
23   22.594651   2.5.2   host         USB       70      URB_BULK in
26   22.597115   host    2.5.1        USB       72      URB_BULK out
27   22.598215   2.5.2   host         USB       70      URB_BULK in
30   22.600618   host    2.5.1        USB       72      URB_BULK out
32   22.602486   2.5.2   host         USB       75      URB_BULK in
34   22.605197   host    2.5.1        USB       72      URB_BULK out
36   22.606340   2.5.2   host         USB       72      URB_BULK in
38   22.610826   host    2.5.1        USB       72      URB_BULK out
40   22.612550   2.5.2   host         USB       85      URB_BULK in
42   22.615246   host    2.5.1        USB       72      URB_BULK out
44   22.616139   2.5.2   host         USB       70      URB_BULK in
```

Now that we are capturing USB traffic with Wireshark, run the velocio-lab script again from a terminal. You should see a lot of data appear in Wireshark. You can use the filter "usb.capdata" to only show traffic carrying serial communications we are interested in. See the screenshot above for an example of this filter and the traffic generated by the velocio-lab script.



EXERCISE 2.2: EXPLORING FIELDBUS PROTOCOLS

APPLY USB.CAPDATA AS COLUMN

Select the first packet that contains usb.capdata

Look at the decode section

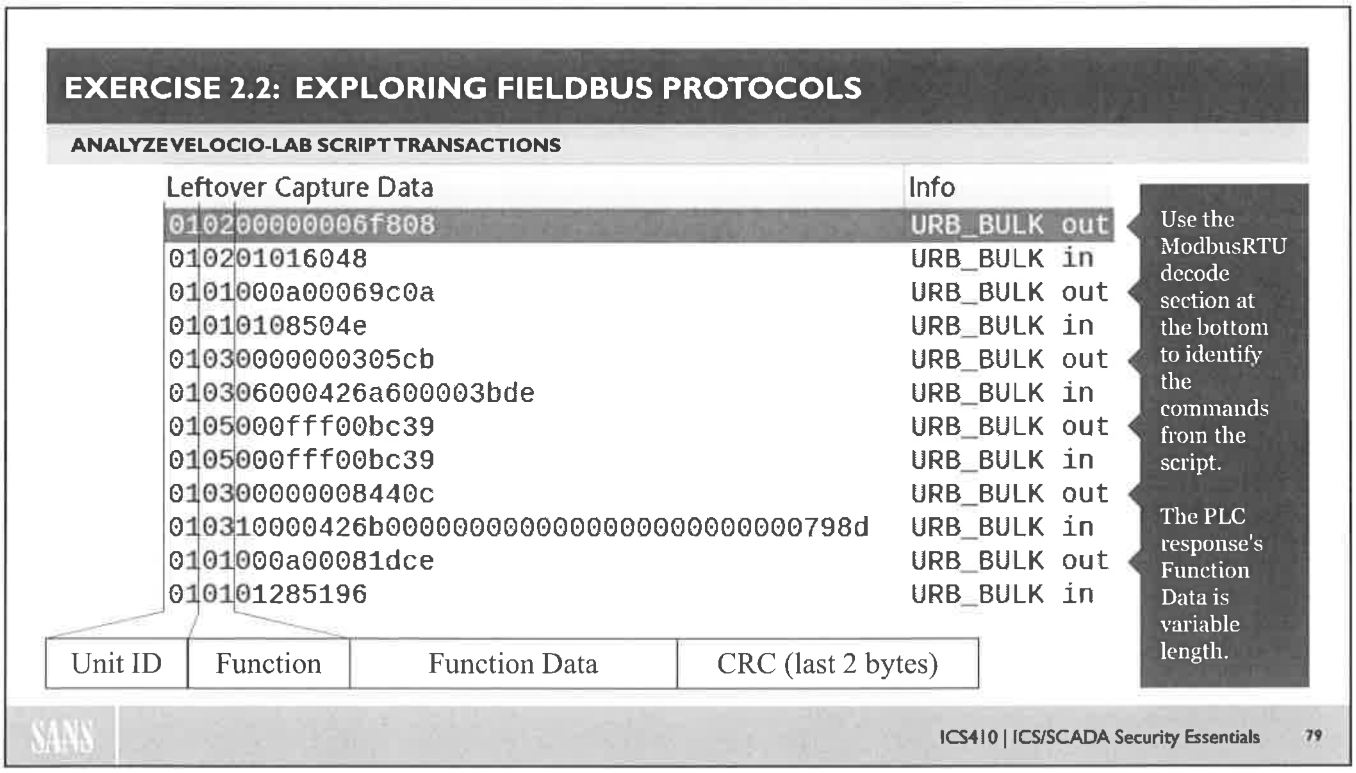
Right-click on the "Leftover Capture Data" line and select "Apply as Column"

```
No.  Time           Source   Destination   Protocol   Length  Info
  22 22.593142      host     2.5.1         USB            72 URB_BULK out
  23 22.594651      2.5.2    host          USB            70 URB_BULK in
  26 22.597115      host     2.5.1         USB            72 URB_BULK out
  27 22.598215      2.5.2    host          USB            70 URB_BULK in
  30 22.600618      host     2.5.1         USB            72 URB_BULK out
  32 22.602486      2.5.2    host          USB            75 URB_BULK in
  34 22.605197      host     2.5.1         USB            72 URB_BULK out
  36 22.606340      2.5.2    host          USB            72 URB_BULK in
  38 22.610826      host     2.5.1         USB            72 URB_BULK out
  40 22.612550      2.5.2    host          USB            85 URB_BULK in
  42 22.615246      host     2.5.1         USB            72 URB_BULK out
  44 22.616139      2.5.2    host          USB            70 URB_BULK in

▸ Frame 22: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on
▸ USB URB
  Leftover Capture Data: 01070000000006f808
```

Take a closer look at the first capture in your filtered view. In the decode section below the list of packets, you can look at the detailed packet breakdown. Notice that every one of these packets has a line called "Leftover Capture Data." This is the serial traffic being pushed through the USB URB channel. We can have Wireshark show this data for each packet in the main packet view by adding it as a column. Right-click on that line and select "Apply as Column."

If you were not aware of this trick in Wireshark, take note. This is by far the easiest method for analyzing what is occurring across a sequence of packets in Wireshark, and can be done with just about any line in the details section.



EXERCISE 2.2: EXPLORING FIELDBUS PROTOCOLS

ANALYZEVELOCIO-LAB SCRIPTTRANSACTIONS

```
Leftover Capture Data                              Info
0102000000006f808                                  URB_BULK out
0102010160 48                                      URB_BULK in
0101000a00069c0a                                   URB_BULK out
010101 08504e                                      URB_BULK in
0103000000000305cb                                 URB_BULK out
010306000426a600003bde                             URB_BULK in
0105000fff00bc39                                   URB_BULK out
0105000fff00bc39                                   URB_BULK in
010300000008440c                                   URB_BULK out
01031000042 6b000000000000000000000000000798d      URB_BULK in
0101000a00081dce                                   URB_BULK out
010101285196                                       URB_BULK in
```

Unit ID	Function	Function Data	CRC (last 2 bytes)

Use the ModbusRTU decode section at the bottom to identify the commands from the script.

The PLC response's Function Data is variable length.

Above is an extraction of the raw Modbus RTU communication between the velocio-lab script and the PLC. If you use the "Info" column, you can see which packets are requests going "out" from the velocio-lab script, and which packets are responses coming back "in" from the PLC. The Modbus RTU breakdown I show above is only applicable for the requests going out. The Responses coming back in from the PLC have a variable length "Function Data" section, so they do not line up cleanly above, but you can manually figure it out as the "Function Data" section is everything between the "Function" section and "CRC" section, which is always the last four characters (or two bytes).

Here is a list of the six most common functions for Modbus:

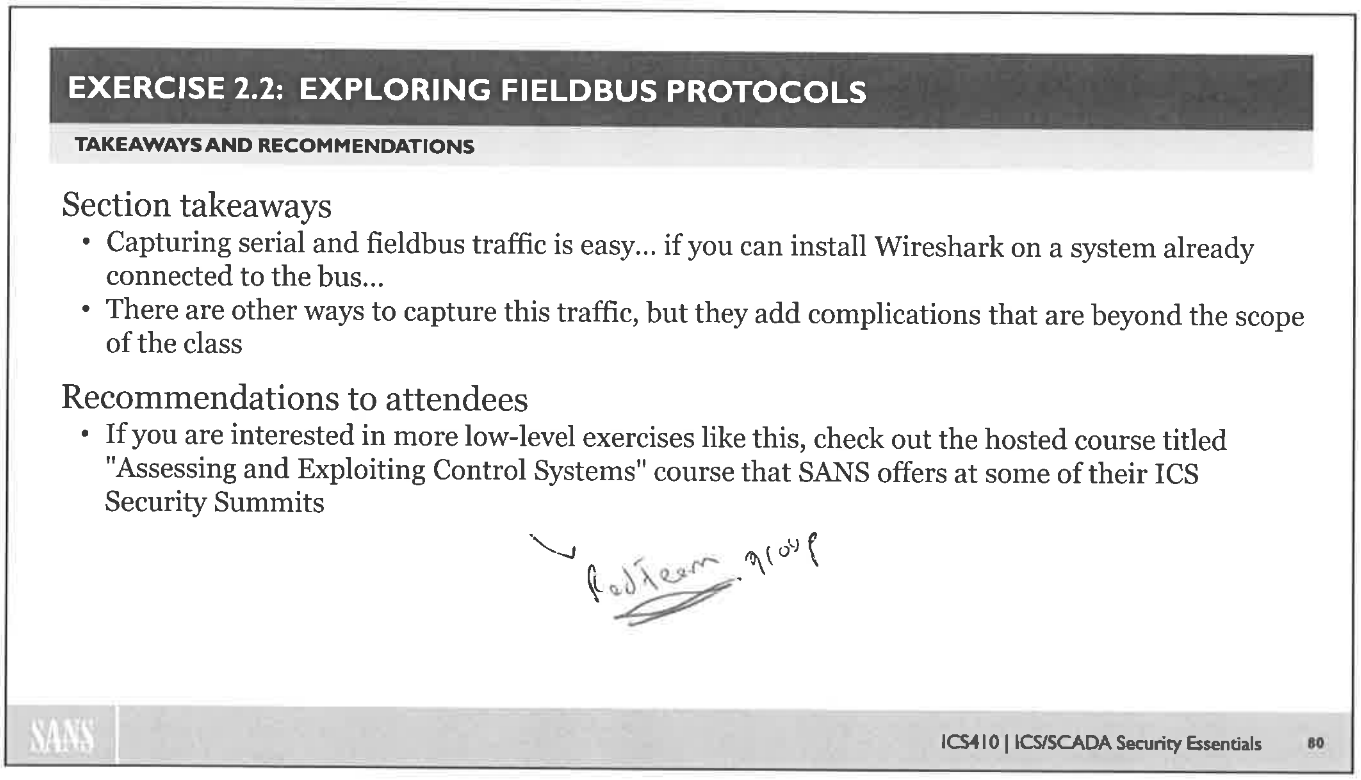
01 – read coils (aka digital outputs)
02 – read discrete inputs (aka digital inputs)
03 – read holding registers (aka analog outputs and internal registers)
04 – read input registers (aka analog inputs)
05 – write single coils (aka digital outputs)
06 – write single registers (aka digital outputs)

Using those function definitions, try to manually decode each read-and-write request above. Remember there is a response for each request listed below:

From unit 1, read digital inputs, starting at address 0, 6 addresses (basically all your inputs)
From unit 1, read digital outputs, starting at address 10 (000a in hex), 6 addresses (basically all your outputs)
From unit 1, read internal registers, starting at address 0, read 3 addresses
From unit 1, write digital outputs, at address 15 (000f in hex), turn on (ff00 is True/On/1 and 0000 is False/Off/0)
From unit 1, read internal registers, starting at address 0, 8 addresses
From unit 1, read digital outputs, starting at address 10, 8 addresses

If you have extra time, see if you can decode the responses from the PLC, which is beyond the scope of this lab. Also, for those familiar with Python, or those who are interested in learning it, check out the code behind the velocio-lab script:

```
control@ctp:~$ cat /usr/local/bin/velocio-lab
```



EXERCISE 2.2: EXPLORING FIELDBUS PROTOCOLS

TAKEAWAYS AND RECOMMENDATIONS

Section takeaways

- Capturing serial and fieldbus traffic is easy... if you can install Wireshark on a system already connected to the bus...
- There are other ways to capture this traffic, but they add complications that are beyond the scope of the class

Recommendations to attendees

- If you are interested in more low-level exercises like this, check out the hosted course titled "Assessing and Exploiting Control Systems" course that SANS offers at some of their ICS Security Summits

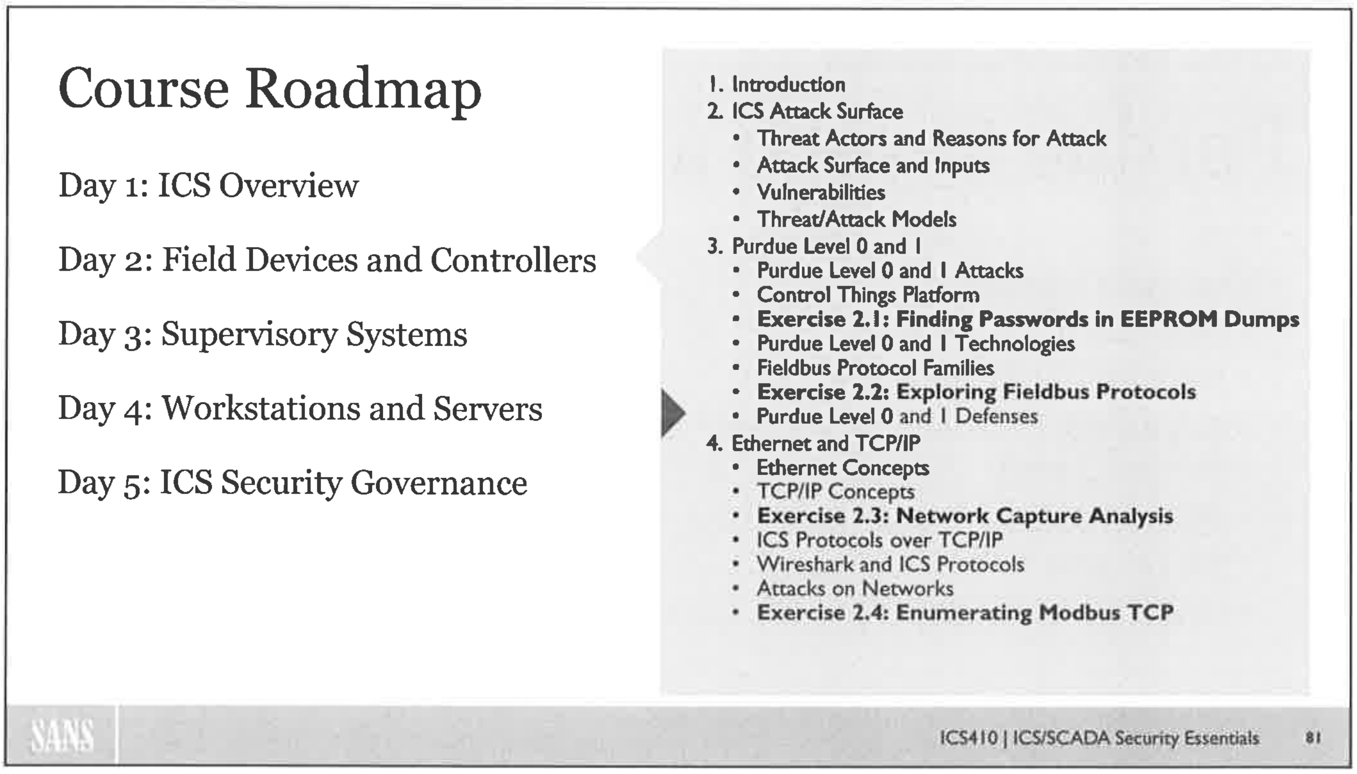
Redteam group

SANS

ICS410 | ICS/SCADA Security Essentials 80

This page intentionally left blank.

© 2019, Justin Searle



Course Roadmap

Day 1: ICS Overview

Day 2: Field Devices and Controllers

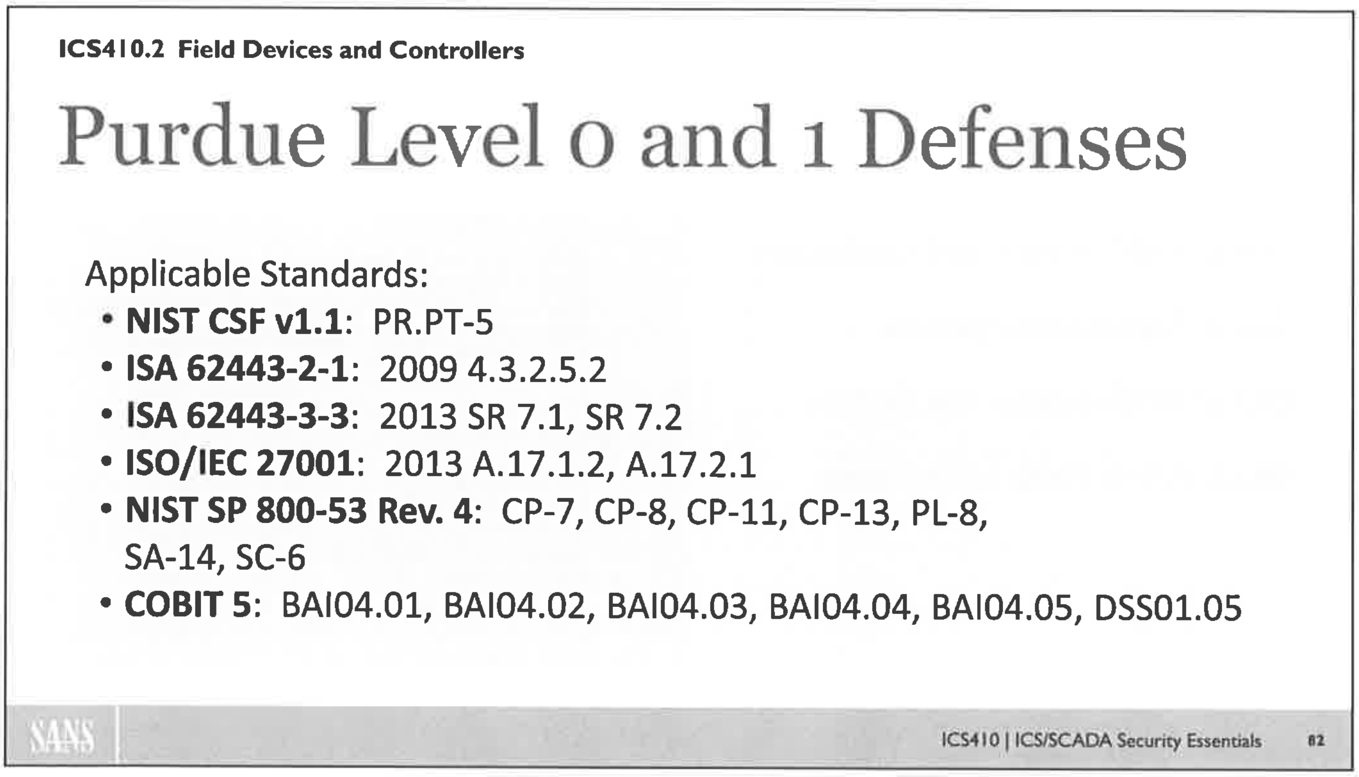
Day 3: Supervisory Systems

Day 4: Workstations and Servers

Day 5: ICS Security Governance

SANS ICS410 | ICS/SCADA Security Essentials 81

This page intentionally left blank.

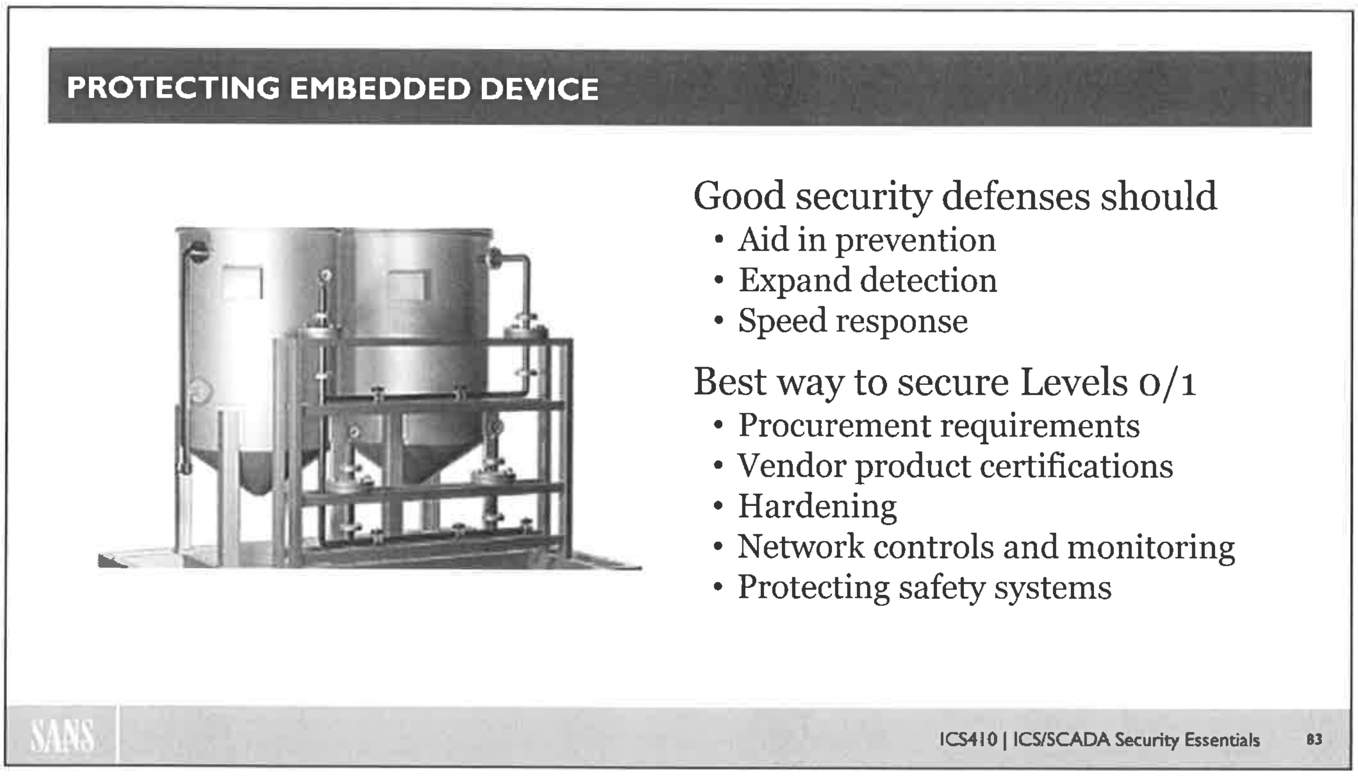


Purdue Level 0 and 1 Defenses

Applicable Standards:

- **NIST CSF v1.1**: PR.PT-5
- **ISA 62443-2-1**: 2009 4.3.2.5.2
- **ISA 62443-3-3**: 2013 SR 7.1, SR 7.2
- **ISO/IEC 27001**: 2013 A.17.1.2, A.17.2.1
- **NIST SP 800-53 Rev. 4**: CP-7, CP-8, CP-11, CP-13, PL-8, SA-14, SC-6
- **COBIT 5**: BAI04.01, BAI04.02, BAI04.03, BAI04.04, BAI04.05, DSS01.05

This page intentionally left blank.



PROTECTING EMBEDDED DEVICE

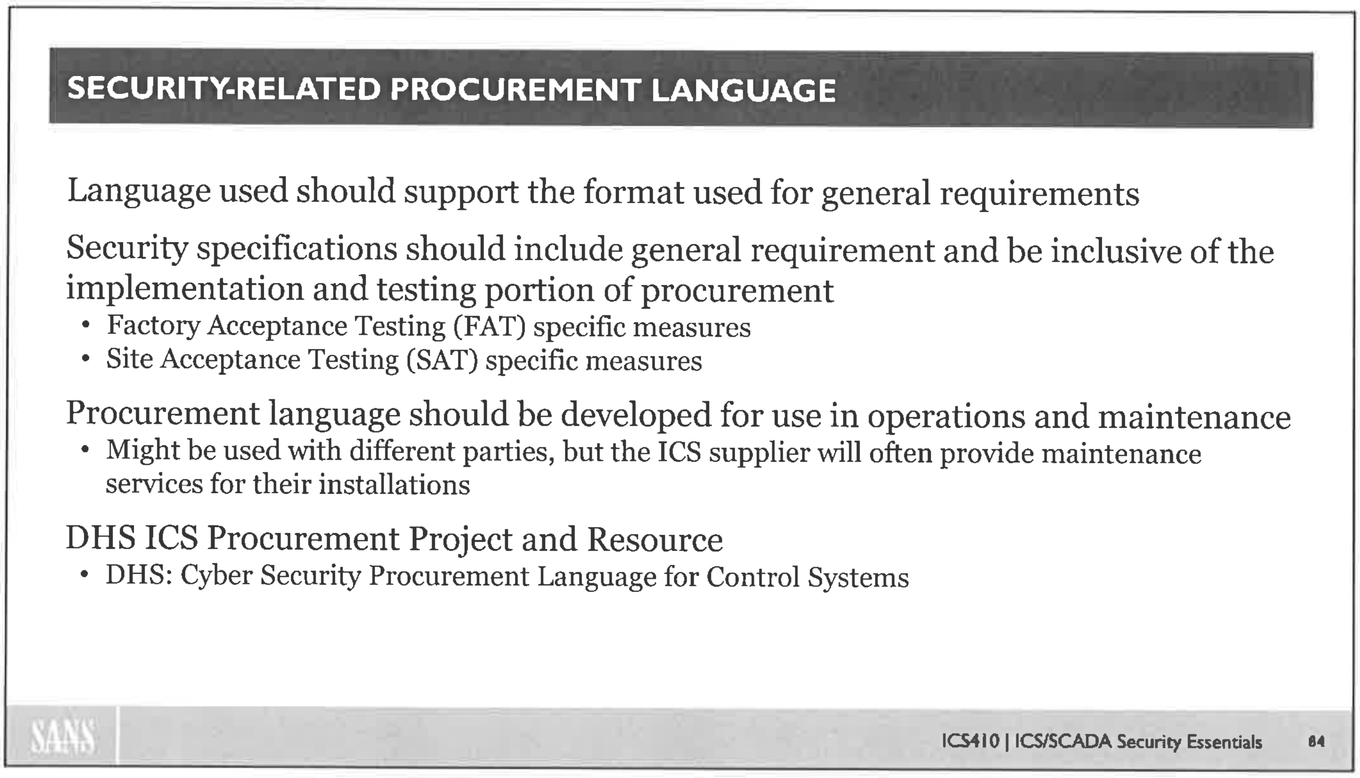
Good security defenses should
- Aid in prevention
- Expand detection
- Speed response

Best way to secure Levels 0/1
- Procurement requirements
- Vendor product certifications
- Hardening
- Network controls and monitoring
- Protecting safety systems

SANS ICS410 | ICS/SCADA Security Essentials 83

Several security mitigations and good security practices can aid in prevention, detection, and response for embedded systems (for example, controllers/PLCs). We can organize these efforts by following the life cycle from procurement, testing, operations, and maintenance to decommissioning.

prevention is ideal
detection is essential



SECURITY-RELATED PROCUREMENT LANGUAGE

Language used should support the format used for general requirements

Security specifications should include general requirement and be inclusive of the implementation and testing portion of procurement
- Factory Acceptance Testing (FAT) specific measures
- Site Acceptance Testing (SAT) specific measures

Procurement language should be developed for use in operations and maintenance
- Might be used with different parties, but the ICS supplier will often provide maintenance services for their installations

DHS ICS Procurement Project and Resource
- DHS: Cyber Security Procurement Language for Control Systems

ICS410 | ICS/SCADA Security Essentials 84

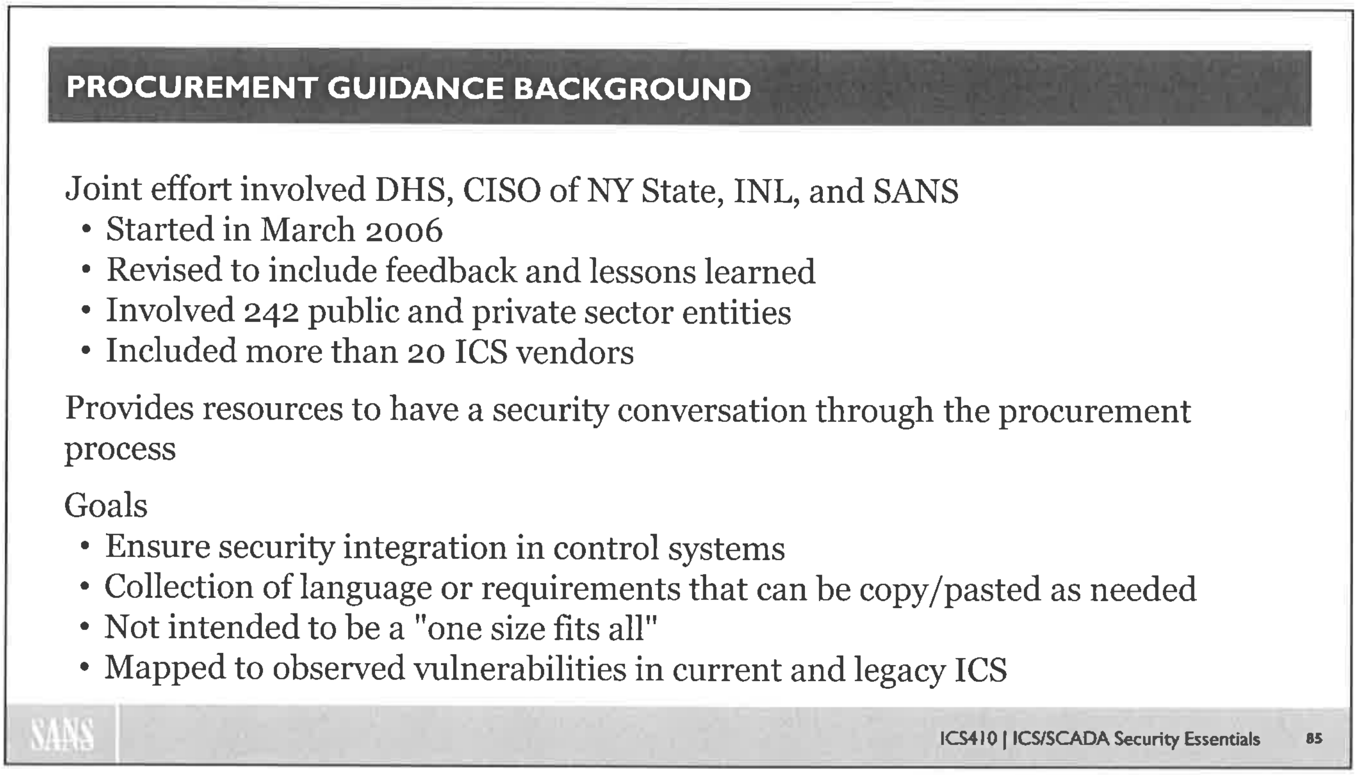
The procurement process has the primary purpose of identifying a customer's needs and matching appropriate solutions against requirements and within the price point for the buyer. Security can be specified as a requirement, but it will be competing with other requirements, such as features, and is certainly an element of the overall price calculus.

Security requirements were hard to introduce, and technical specialists struggled to form the requirements into language used in the procurement process. In developing the resource, procurement specialists were involved to help with language that fit well into this process. Requirements identified in the resource included measures that could be used to verify the requirement had been satisfied. These measures go beyond installation through system operations and maintenance.

References:
Procurement Language Documents: https://ics-cert.us-cert.gov/sites/default/files/documents/Procurement_Language_Rev4_100809_S508C.pdf
http://www.energy.gov/sites/prod/files/2014/04/f15/CybersecProcurementLanguage-EnergyDeliverySystems_040714_fin.pdf

ICS-CERT resources: http://ics-cert.us-cert.gov/Information-Products

© 2019, Justin Searle



PROCUREMENT GUIDANCE BACKGROUND

Joint effort involved DHS, CISO of NY State, INL, and SANS

- Started in March 2006
- Revised to include feedback and lessons learned
- Involved 242 public and private sector entities
- Included more than 20 ICS vendors

Provides resources to have a security conversation through the procurement process

Goals

- Ensure security integration in control systems
- Collection of language or requirements that can be copy/pasted as needed
- Not intended to be a "one size fits all"
- Mapped to observed vulnerabilities in current and legacy ICS

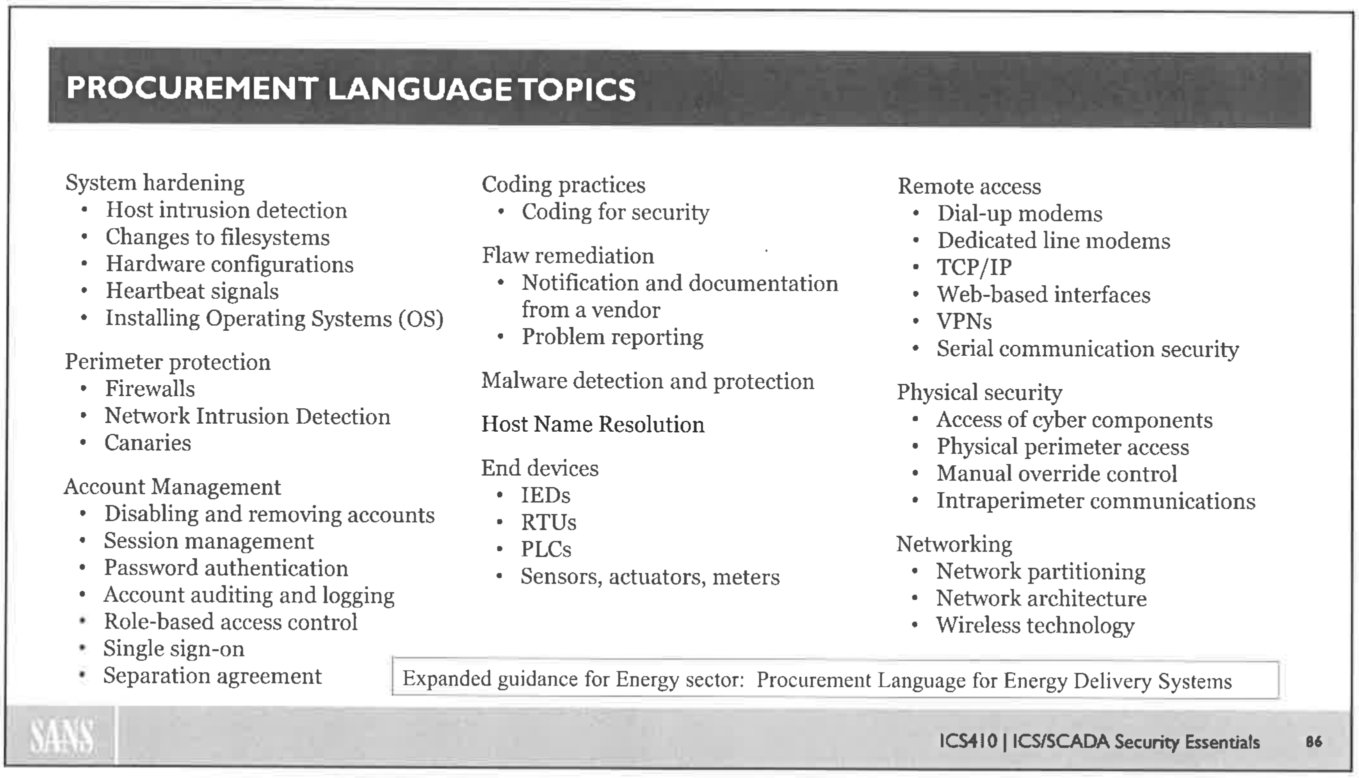
SANS ICS410 | ICS/SCADA Security Essentials 85

The Procurement Project spanned three years and was the result of a joint and global collaboration that included public and private sector organizations. The effort included hundreds of stakeholders and brought together multiple disciplines to develop requirement language.

The result took observed ICS vulnerabilities and developed appropriate requirement language and measures to support future procurements and the development of maintenance agreements.

The document is designed to provide a useful "toolkit" to reduce control systems' cyber risk by asking technology providers, through the procurement cycle, to assist in managing known vulnerabilities and weaknesses by delivering more secure systems.

The toolkit is designed as a resource to start the process. Guidance is provided over a number of security domains and observed vulnerabilities.



PROCUREMENT LANGUAGE TOPICS

System hardening
- Host intrusion detection
- Changes to filesystems
- Hardware configurations
- Heartbeat signals
- Installing Operating Systems (OS)

Perimeter protection
- Firewalls
- Network Intrusion Detection
- Canaries

Account Management
- Disabling and removing accounts
- Session management
- Password authentication
- Account auditing and logging
- Role-based access control
- Single sign-on
- Separation agreement

Coding practices
- Coding for security

Flaw remediation
- Notification and documentation from a vendor
- Problem reporting

Malware detection and protection

Host Name Resolution

End devices
- IEDs
- RTUs
- PLCs
- Sensors, actuators, meters

Remote access
- Dial-up modems
- Dedicated line modems
- TCP/IP
- Web-based interfaces
- VPNs
- Serial communication security

Physical security
- Access of cyber components
- Physical perimeter access
- Manual override control
- Intraperimeter communications

Networking
- Network partitioning
- Network architecture
- Wireless technology

Expanded guidance for Energy sector: Procurement Language for Energy Delivery Systems

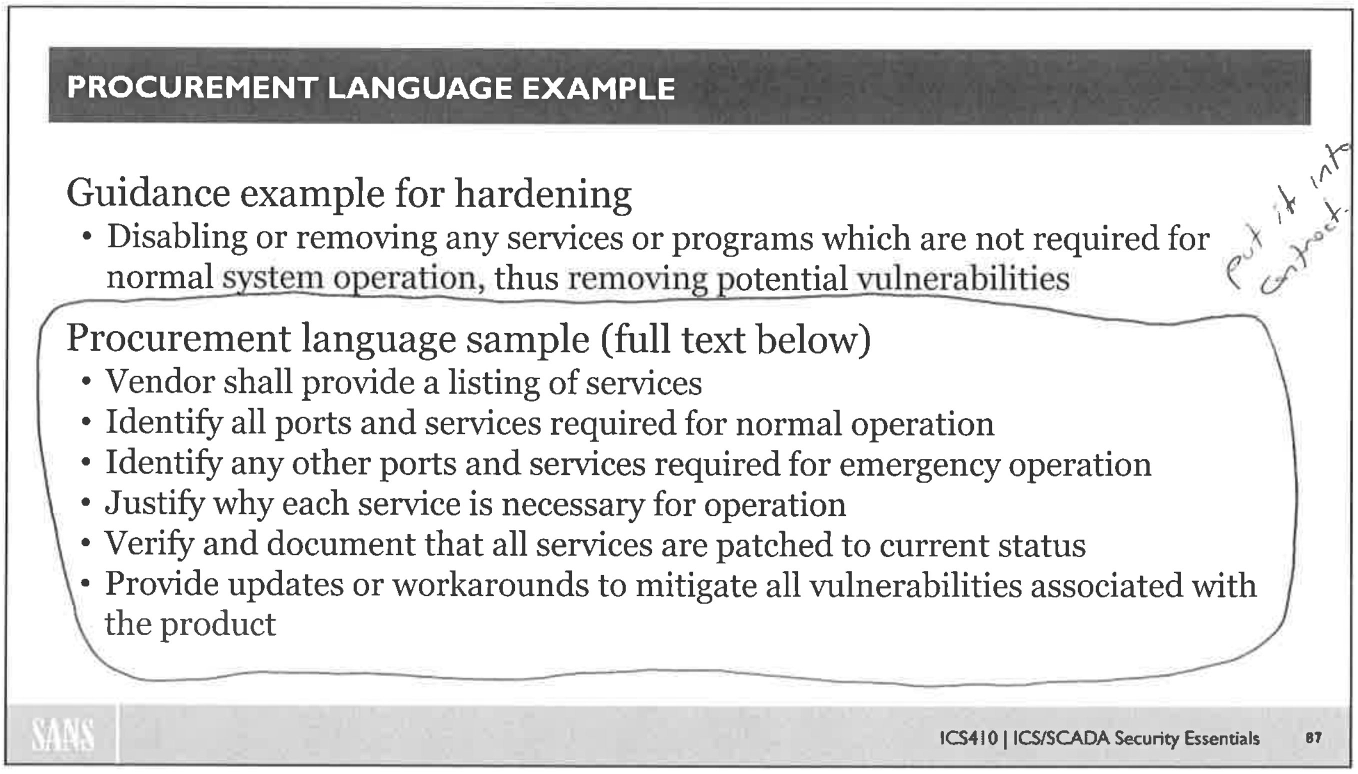
SANS ICS410 | ICS/SCADA Security Essentials 86

As you can see, there are a large number of topics covered in the procurement language toolkit that can aid any business that has ICS.

There is expanded guidance available to the Energy sector in a document called the "Procurement Language for Energy Delivery Systems". This was created by Energy Sector Control Systems Working Group (ESCSWG) and released April 2014. This provides additional verbiage for systems that are unique to the Energy sector that were not covered in the original procurement language toolkit, and updated verbiage to a few items that were originally covered. It has verbiage for the following categories of systems:

- General Cybersecurity Language
- Supplier Life Cycle Security Program
- Intrusion Detection
- Wireless Technologies
- Cryptographic System Management

© 2019, Justin Searle



PROCUREMENT LANGUAGE EXAMPLE

Guidance example for hardening
- Disabling or removing any services or programs which are not required for normal system operation, thus removing potential vulnerabilities

put it into contract.

Procurement language sample (full text below)
- Vendor shall provide a listing of services
- Identify all ports and services required for normal operation
- Identify any other ports and services required for emergency operation
- Justify why each service is necessary for operation
- Verify and document that all services are patched to current status
- Provide updates or workarounds to mitigate all vulnerabilities associated with the product

SANS ICS410 | ICS/SCADA Security Essentials 87

The following text is from the hardening section of the Procurement Language Toolkit.

"Unused services in a host operating system that are left enabled are possible entry points for exploits on the network and are generally not monitored because these services are not used. Only the services used for control systems operation and maintenance shall be enabled to limit possible entry points."

"Often, networked devices ship with a variety of services enabled and default operating system programs/utilities pre-installed. These range from system diagnostics to chat programs, several of which have well-known vulnerabilities. Various attacks have been crafted to exploit these services to obtain information leading to compromise the system."

"Any program that offers a network service that "listens" on specific addresses for connection requests can potentially be vulnerable to an attack. On a Transmission Control Protocol (TCP)/Internet Protocol (IP) network, these addresses are a combination of IP address and TCP or User Datagram Protocol (UDP) ports. A recommended hardening activity is simply disabling or removing any services or programs that are not required for normal system operation, thus removing potential vulnerabilities."

"Port scans are the normal method of ensuring the existence of required services and absence of unneeded services. A port scan shall be run before the FAT with a representative, fully functional system configuration. All input/output (I/O) ports need to be scanned for UDP and TCP. The scan needs to be run before the FAT and again prior to the SAT. Port scans can rarely be used on production systems. In most cases, scanners will disrupt operations."

Procurement Language
"Post-contract award, the vendor shall provide documentation detailing all applications, utilities, system services, scripts, configuration files, databases, and all other software required and the appropriate configurations, including revisions and/or patch levels for each of the computer systems associated with the control system."

"The vendor shall provide a listing of services required for any computer system running control system applications or required to interface the control system applications. The listing shall include all ports and services required for normal operation, as well as any other ports and services required for emergency operation. The listing shall also include an explanation or cross reference to justify why each service is necessary for operation."

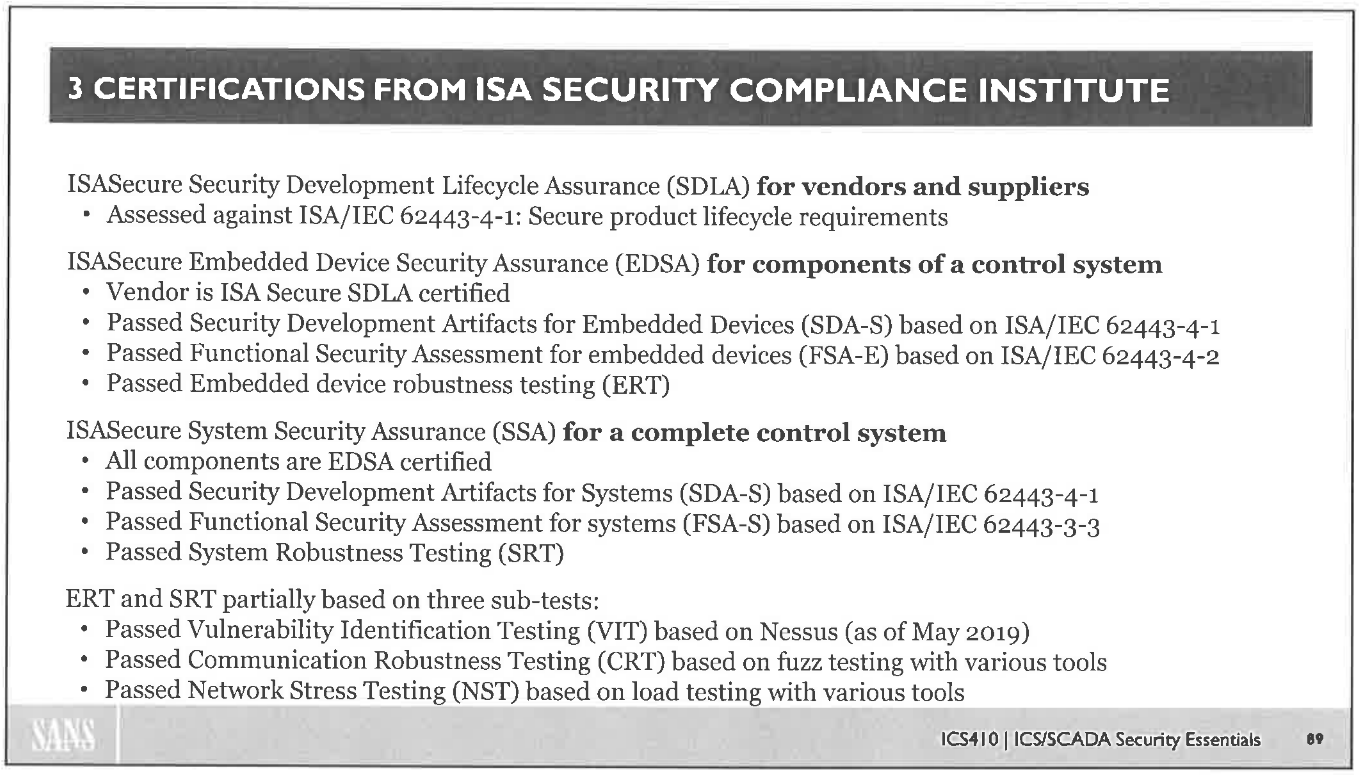
"The vendor shall verify and provide documentation that all services are patched to current status."

"The vendor shall provide, within a prenegotiated period, appropriate software and service updates and/or workarounds to mitigate all vulnerabilities associated with the product and to maintain the established level of system security."

References:
Procurement Language Document: https://ics-cert.us-cert.gov/sites/default/files/documents/Procurement_Language_Rev4_100809_S508C.pdf

http://www.energy.gov/sites/prod/files/2014/04/f15/CybersecProcurementLanguage-EnergyDeliverySystems_040714_fin.pdf

© 2019, Justin Searle



3 CERTIFICATIONS FROM ISA SECURITY COMPLIANCE INSTITUTE

ISASecure Security Development Lifecycle Assurance (SDLA) **for vendors and suppliers**
- Assessed against ISA/IEC 62443-4-1: Secure product lifecycle requirements

ISASecure Embedded Device Security Assurance (EDSA) **for components of a control system**
- Vendor is ISA Secure SDLA certified
- Passed Security Development Artifacts for Embedded Devices (SDA-S) based on ISA/IEC 62443-4-1
- Passed Functional Security Assessment for embedded devices (FSA-E) based on ISA/IEC 62443-4-2
- Passed Embedded device robustness testing (ERT)

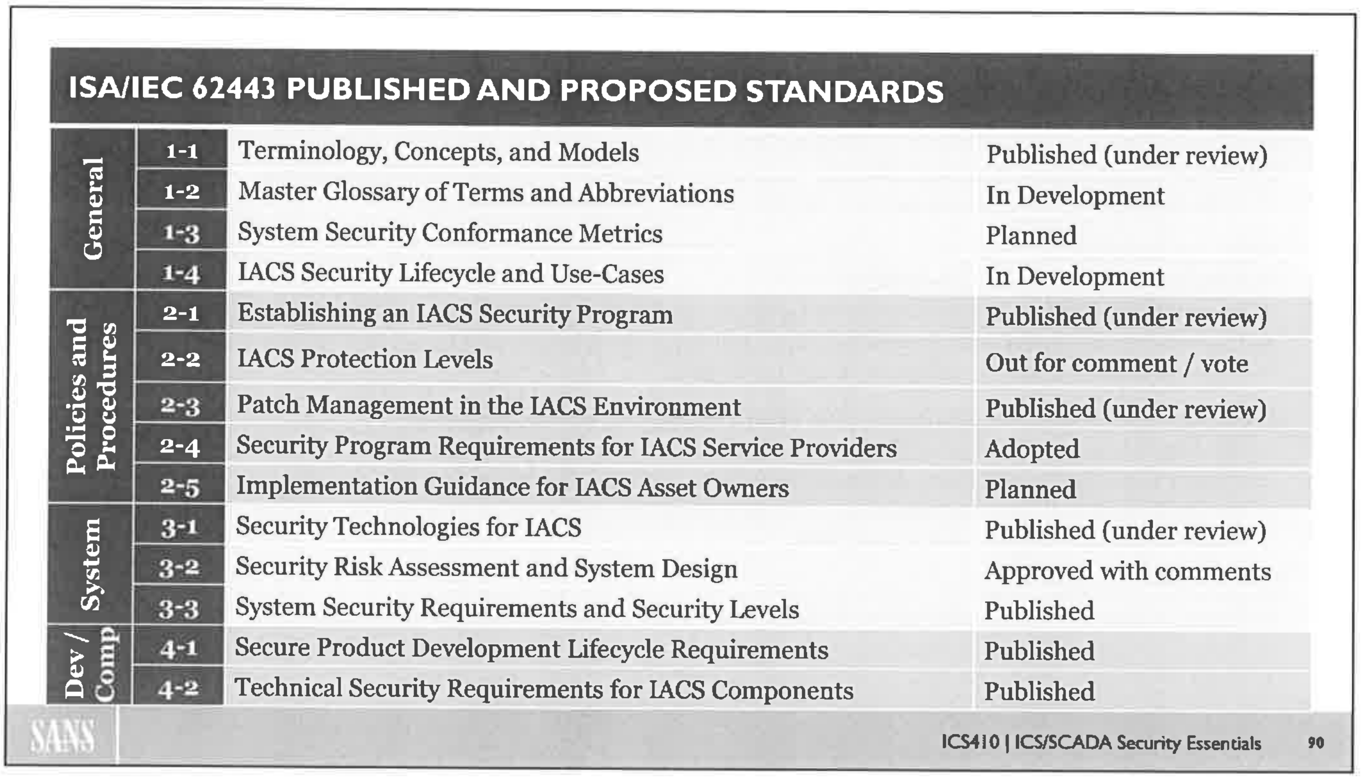
ISASecure System Security Assurance (SSA) **for a complete control system**
- All components are EDSA certified
- Passed Security Development Artifacts for Systems (SDA-S) based on ISA/IEC 62443-4-1
- Passed Functional Security Assessment for systems (FSA-S) based on ISA/IEC 62443-3-3
- Passed System Robustness Testing (SRT)

ERT and SRT partially based on three sub-tests:
- Passed Vulnerability Identification Testing (VIT) based on Nessus (as of May 2019)
- Passed Communication Robustness Testing (CRT) based on fuzz testing with various tools
- Passed Network Stress Testing (NST) based on load testing with various tools

SANS ICS410 | ICS/SCADA Security Essentials 89

The ISA Security Compliance Institute (ISCI) has developed compliance test specifications for ISA99 and other control system security standards. They have also created an ANSI-accredited certification program called ISASecure for the certification of industrial automation devices such as PLCs and field components found in DCSs and PCSs.

References:
http://www.isasecure.org/en-US/
https://isasecure.org/en-US/Certification/IEC-62443-SDLA-Certification-(1)
https://isasecure.org/en-US/Certification/IEC-62443-EDSA-Certification
https://isasecure.org/en-US/Certification/IEC-62443-SSA-Certification
https://www.isasecure.org/en-US/Test-Tools
Recognized CRT test tools, http://isasecure.org/en-US/Test-Tools/Recognized-CRT-Test-Tools
Recognized VIT test tools, http://isasecure.org/en-US/Test-Tools/Vulnerability-Identification-Testing-Tool



ISA/IEC 62443 PUBLISHED AND PROPOSED STANDARDS

General	1-1	Terminology, Concepts, and Models	Published (under review)
	1-2	Master Glossary of Terms and Abbreviations	In Development
	1-3	System Security Conformance Metrics	Planned
	1-4	IACS Security Lifecycle and Use-Cases	In Development
Policies and Procedures	2-1	Establishing an IACS Security Program	Published (under review)
	2-2	IACS Protection Levels	Out for comment / vote
	2-3	Patch Management in the IACS Environment	Published (under review)
	2-4	Security Program Requirements for IACS Service Providers	Adopted
	2-5	Implementation Guidance for IACS Asset Owners	Planned
System	3-1	Security Technologies for IACS	Published (under review)
	3-2	Security Risk Assessment and System Design	Approved with comments
	3-3	System Security Requirements and Security Levels	Published
Dev / Comp	4-1	Secure Product Development Lifecycle Requirements	Published
	4-2	Technical Security Requirements for IACS Components	Published

SANS

ICS410 | ICS/SCADA Security Essentials 90

Industrial Automation and Control Systems (IACS) is the ISA's preferred term for ICS.

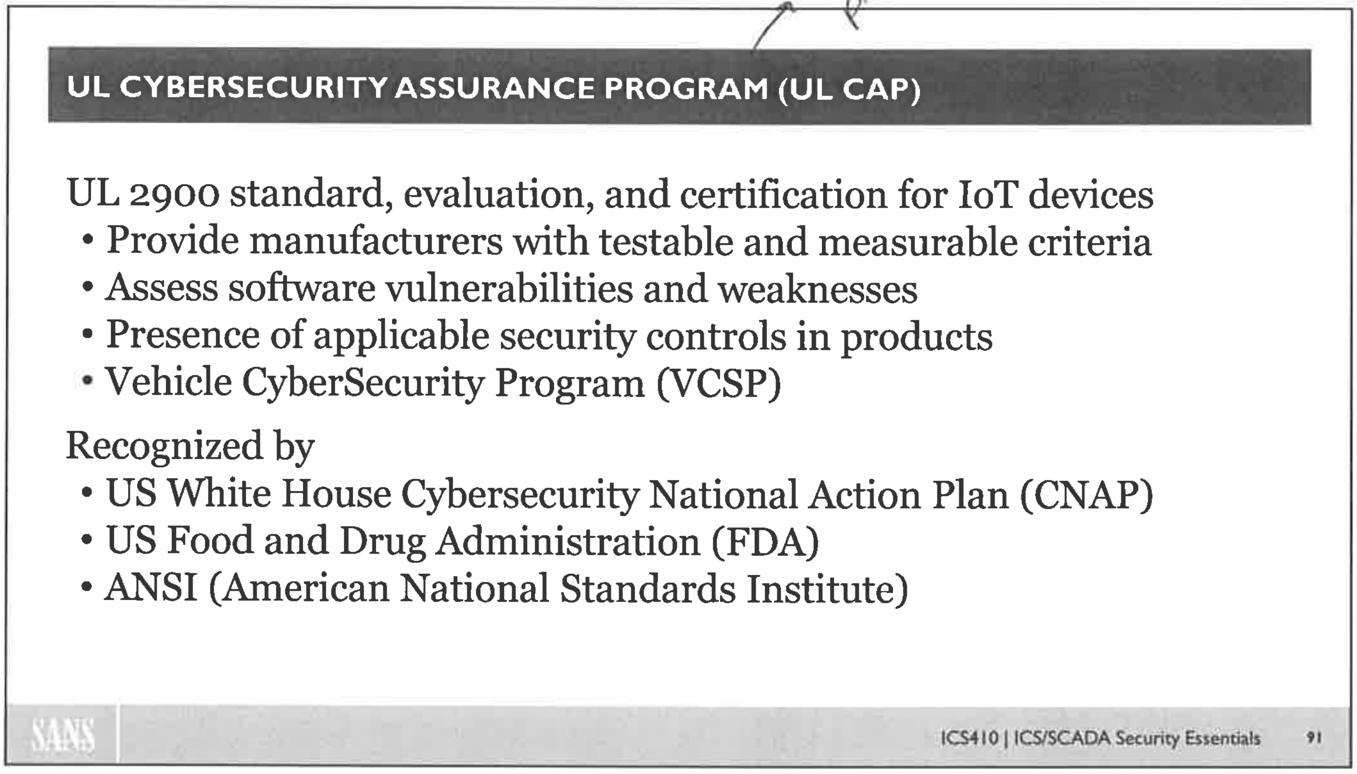
Buying a PDF copy all the published ISA/IEC 62443 standards can be a very expensive experience with each selling for $200-350 for a single copy for a single person. However by becoming an student member of ISA for ~$10/yr or professional member of ISA for ~$130/yr, you can gain browser-only access to all of the ISA standards for free, including ISA/IED 62443. Once you pay your annual membership fee, visit the following link and search for "62443" to access the standards:

https://www.isa.org/standards-and-publications/isa-standards/member-access-to-standards/

Reference:
https://www.isa.org/isa99/

© 2019, Justin Searle

assess your product independently.

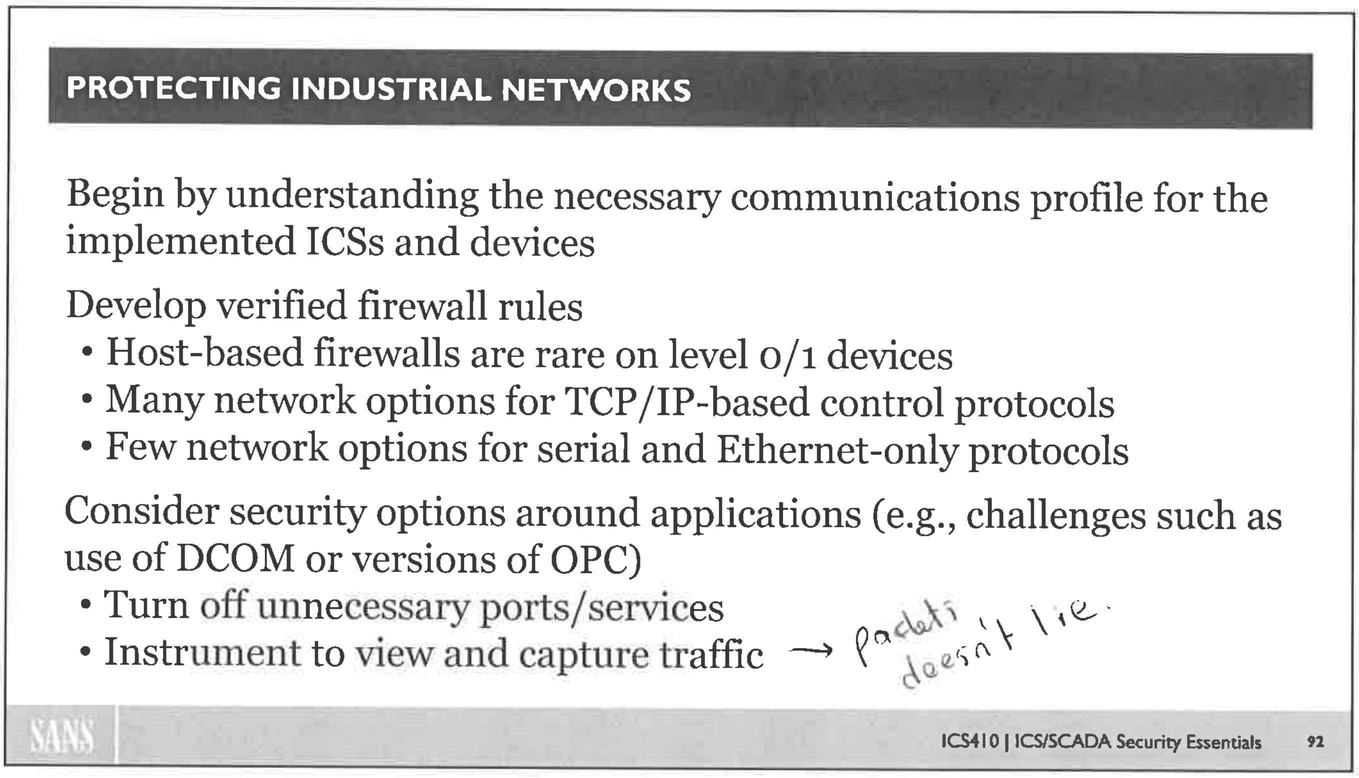


UL CYBERSECURITY ASSURANCE PROGRAM (UL CAP)

UL 2900 standard, evaluation, and certification for IoT devices
- Provide manufacturers with testable and measurable criteria
- Assess software vulnerabilities and weaknesses
- Presence of applicable security controls in products
- Vehicle CyberSecurity Program (VCSP)

Recognized by
- US White House Cybersecurity National Action Plan (CNAP)
- US Food and Drug Administration (FDA)
- ANSI (American National Standards Institute)

© 2019, Justin Searle



PROTECTING INDUSTRIAL NETWORKS

Begin by understanding the necessary communications profile for the implemented ICSs and devices

Develop verified firewall rules
- Host-based firewalls are rare on level 0/1 devices
- Many network options for TCP/IP-based control protocols
- Few network options for serial and Ethernet-only protocols

Consider security options around applications (e.g., challenges such as use of DCOM or versions of OPC)
- Turn off unnecessary ports/services
- Instrument to view and capture traffic $\rightarrow$ packets doesn't lie.

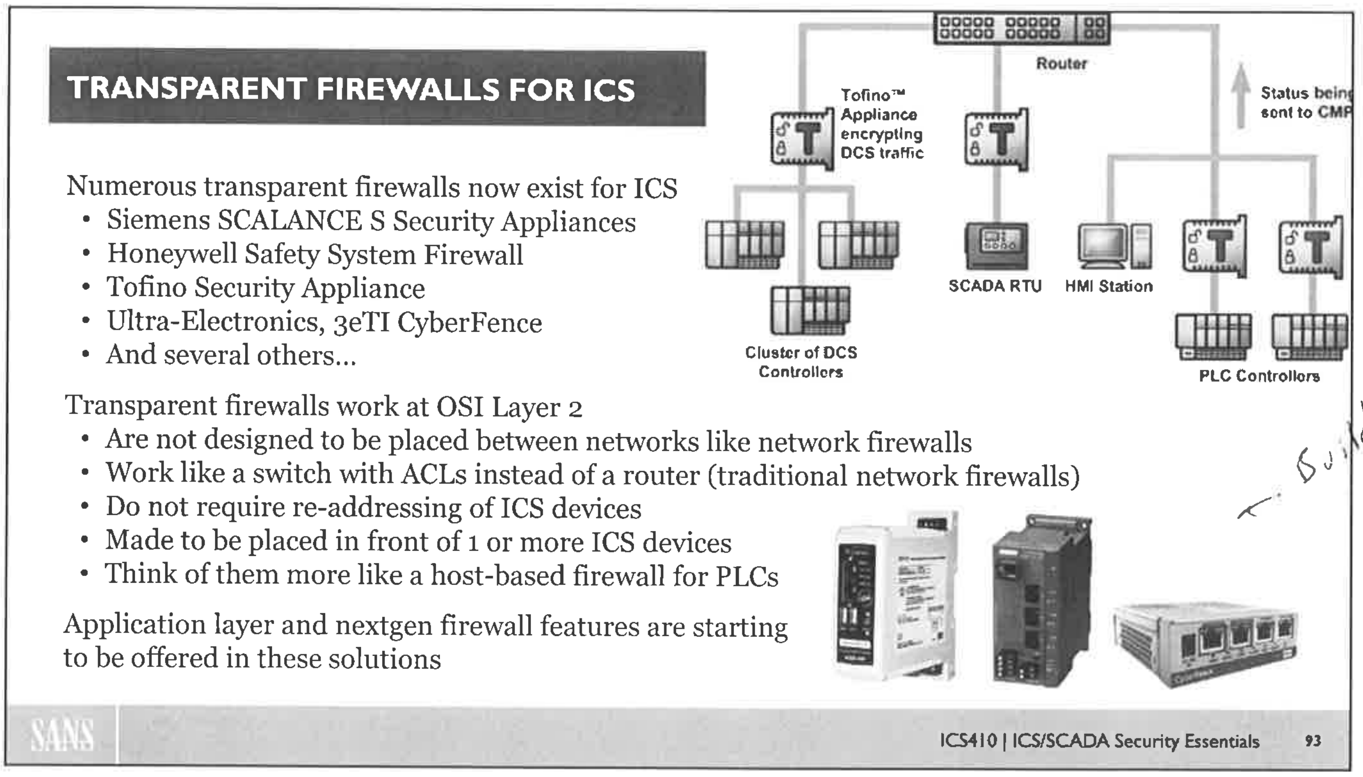
SANS ICS410 | ICS/SCADA Security Essentials 92

Protecting the network and the attached hosts begins by fully understanding the necessary communications between devices (ports and services), protocols in use, and communication profiles associated with normal operations and off-normal operations. (For example, a safety system is triggered.) Several tools can be used to baseline or profile the actual communications of an implemented system (SAT testing or after commissioning). This should be supported by detailed documentation from your ICS supplier as to the necessary ports and services and protocols in use. Your goal is to design the most restrictive segmentation and enforcement strategy that does not impede the functionality of the system. Firewall rules can be designed, tested, and validated. The use of DCOM and certain versions of OPC can limit your ability to develop firewall rules for hosts/networks.

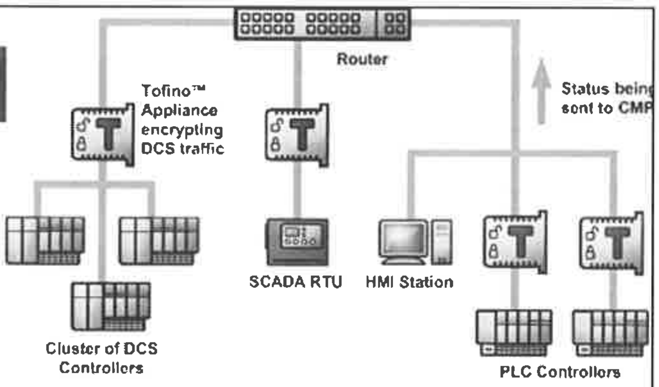
Also, make sure the networking components (switches, hubs, routers, and access points) are assessed, configurations are managed, and protections are implemented. Switch port security measures can help prevent unauthorized devices from connecting to networks and can be further used to allow specific MAC addresses to communicate on specific ports only. It is important to consider how you might instrument the network to view and capture traffic and packets to support troubleshooting or to investigate suspicious activity. Not all switches will support port mirroring or switch port spanning, and in those situations, you may need to implement a network tap and a capture station with a promiscuous mode network card to gain visibility to all communications.

It is important to consider how you might instrument the network to view and capture traffic and packets to support troubleshooting or to investigate suspicious activity. Network monitoring can include the ability to capture full packets (captures all Ethernet/IP activity that is available at the capture location including payload and header) or implement devices such as intrusion detection systems with specific signatures for the protocols in use. In cases where full network packet captures are not available, you may consider system monitoring. Basic system monitoring is commonly performed with Simple Network Management Protocol, or SNMP, and will work with most devices. However, in those cases where implementation challenges exist or if legacy devices do not support SNMP, you can also consider implementing asset availability monitoring through ICMP.

© 2019, Justin Searle



TRANSPARENT FIREWALLS FOR ICS


Router
Tofino™ Appliance encrypting DCS traffic
Status bein
sent to CMP
SCADA RTU
HMI Station
Cluster of DCS Controllers
PLC Controllers

Numerous transparent firewalls now exist for ICS
- Siemens SCALANCE S Security Appliances
- Honeywell Safety System Firewall
- Tofino Security Appliance
- Ultra-Electronics, 3eTI CyberFence
- And several others...

Transparent firewalls work at OSI Layer 2
- Are not designed to be placed between networks like network firewalls
- Work like a switch with ACLs instead of a router (traditional network firewalls)
- Do not require re-addressing of ICS devices
- Made to be placed in front of 1 or more ICS devices
- Think of them more like a host-based firewall for PLCs

Build for ICS.

Application layer and nextgen firewall features are starting to be offered in these solutions

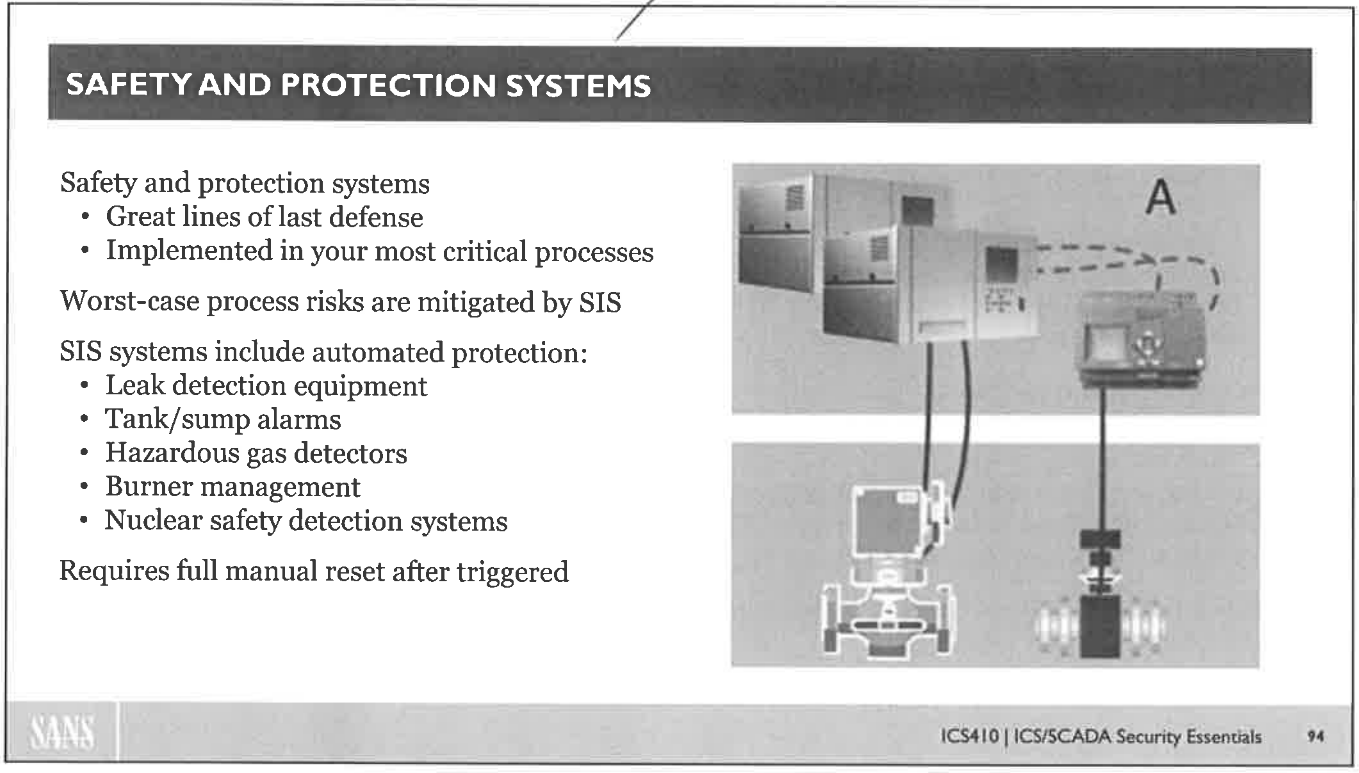
There have been several inline firewalls designed by ICS manufacturers and third parties to enforce filters that are customized to allow only required communications based on the design and application of the control system equipment. Fixed configuration firewalls are designed for specific protocols (for example, allow Modbus, SNMP traps, and more) and signature rules (for example, rate limiting to avoid DoS of communication module) that conform to the suggested use and management of a specific product or control system component. The prepackaged rule sets make it easy for control engineers to address some of the vulnerabilities that may impact ICS devices. They can also enforce security policy by denying the use of default passwords, for example. ICS manufacturer-supplied firewalls are seen as "trusted" solutions as they are tested in the factory, designed to work specifically for the control system family in use, and are easy to install. These firewalls are used in the lower levels of an ICS architecture (Level 2 and below) to filter traffic to controllers and safety system PLCs. The need to place the firewall inline with the device to be protected results in costly deployments for complex processes requiring multiple controllers. Some manufacturers have developed rule-set generators, so customers can create rules for these devices based on their use and device management strategies, configurations, and industrial applications. Many of the firewalls will log remotely (with logs being available via USB port for offloading).

Examples of a fixed configuration firewall are the SCALANCE S602 and the Honeywell Safety System Firewall, which are designed to enforce read-only communication and to filter for valid SIS uses. Another example is the Triconex Tofino Firewall. It works with the Communication Module's on-board User Access Security Model to enforce read-only communications to safeguard safety systems that are connected to the DCS. Many of these devices can include VPN modules. These types of protocol and application-specific firewalls are important tools to enforce appropriate segmentation as more vendors supply converged/integrated control and safety systems. They are also ready made for industrial environments and easy mounting in cabinets and marshaling cabinets.

The picture shows where a specialized ICS firewall like the Tofino Security Appliance can be placed inline with controllers (PLCs, RTUs, and SIS controllers).

Image Source:
http://www.tofinosecurity.com/products/tofino-security-appliance-features

they could kill people or save environment



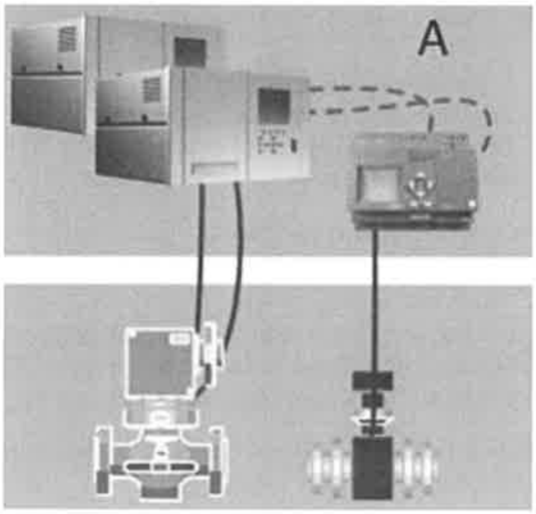
SAFETY AND PROTECTION SYSTEMS

Safety and protection systems
- Great lines of last defense
- Implemented in your most critical processes

Worst-case process risks are mitigated by SIS

SIS systems include automated protection:
- Leak detection equipment
- Tank/sump alarms
- Hazardous gas detectors
- Burner management
- Nuclear safety detection systems

Requires full manual reset after triggered



A *Safety Instrumented System (SIS)* consists of devices or systems that monitor for or remediate any situation that may impact on the plant's safety or personnel safety. SIS systems should be designed and implemented as separate databases and devices from the larger DCS.

Safety *monitoring* instruments or devices include leak detection equipment, tank/sump alarms, hazardous gas detectors, simple emergency stop buttons, burner management, and nuclear safety detection systems. These instruments may be automated in nature, or they can be manual, such as an emergency stop button.

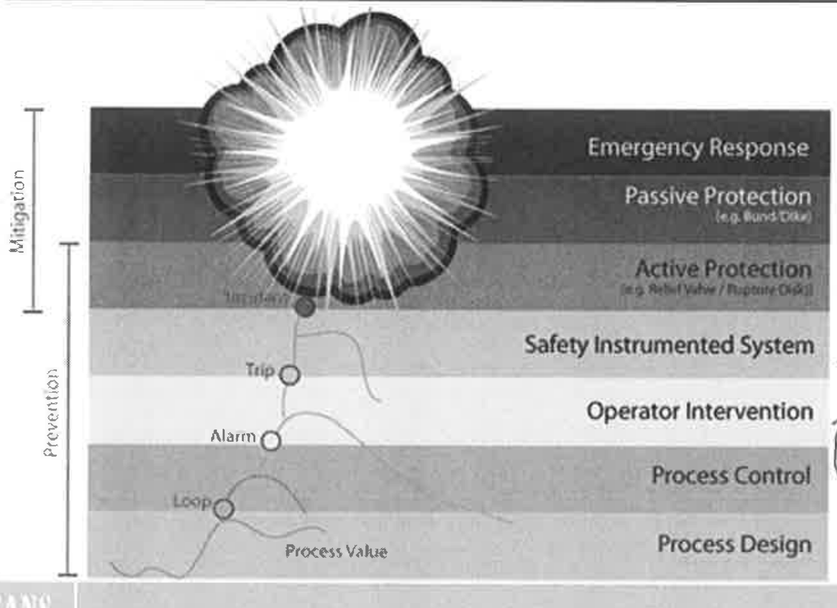
Safety *remediation* instruments or devices include relief valves and emergency shutdown equipment.

- fail safe systems

-

© 2019, Justin Searle

TYPES OF CONTROL SYSTEMS – SIS



SIS performs specific control functions to failsafe or maintain the safe operations

Interested in learning what can go wrong with ICS processes? Check out the videos at the following link:

https://www.csb.gov/videos/

if operator is not good enoug

SIS operates.

Critical processes have been common since the beginning of the Industrial Age. One of the more well-known critical processes is the operation of a steam boiler. Critical parts of the process included the lighting of the burners, controlling the level of water in the drum, and controlling the steam pressure.

A SIS consists of an engineered set of hardware and software controls that are specially used on critical process systems. A critical process system can be identified a system that, once running and an operational problem occurs, may need to be put into a safe state to avoid adverse safety, health, and environmental (SH&E) consequences.

A SIS is engineered to perform specific control functions to failsafe or maintain the safe operation of a process when unacceptable or dangerous conditions occur. SISs must be independent of all other control systems that control the same equipment in order to ensure SIS functionality is not compromised. A SIS is composed of the same types of control elements as any other control system; however, all of the control elements in a SIS are dedicated solely to the proper functioning of the SIS.

The specific control functions performed by a SIS are called Safety Instrumented Functions (SIFs). They are implemented as part of an overall risk reduction strategy that is intended to eliminate the likelihood of a previously identified SH&E event that could range from minor equipment damage up to an event involving an uncontrolled catastrophic release of energy and/or materials.

A safe state is a process condition, whether the process is operating or shut down, such that a hazardous SH&E event cannot occur. The safe state must be achieved in a timely manner or within the "process safety time."

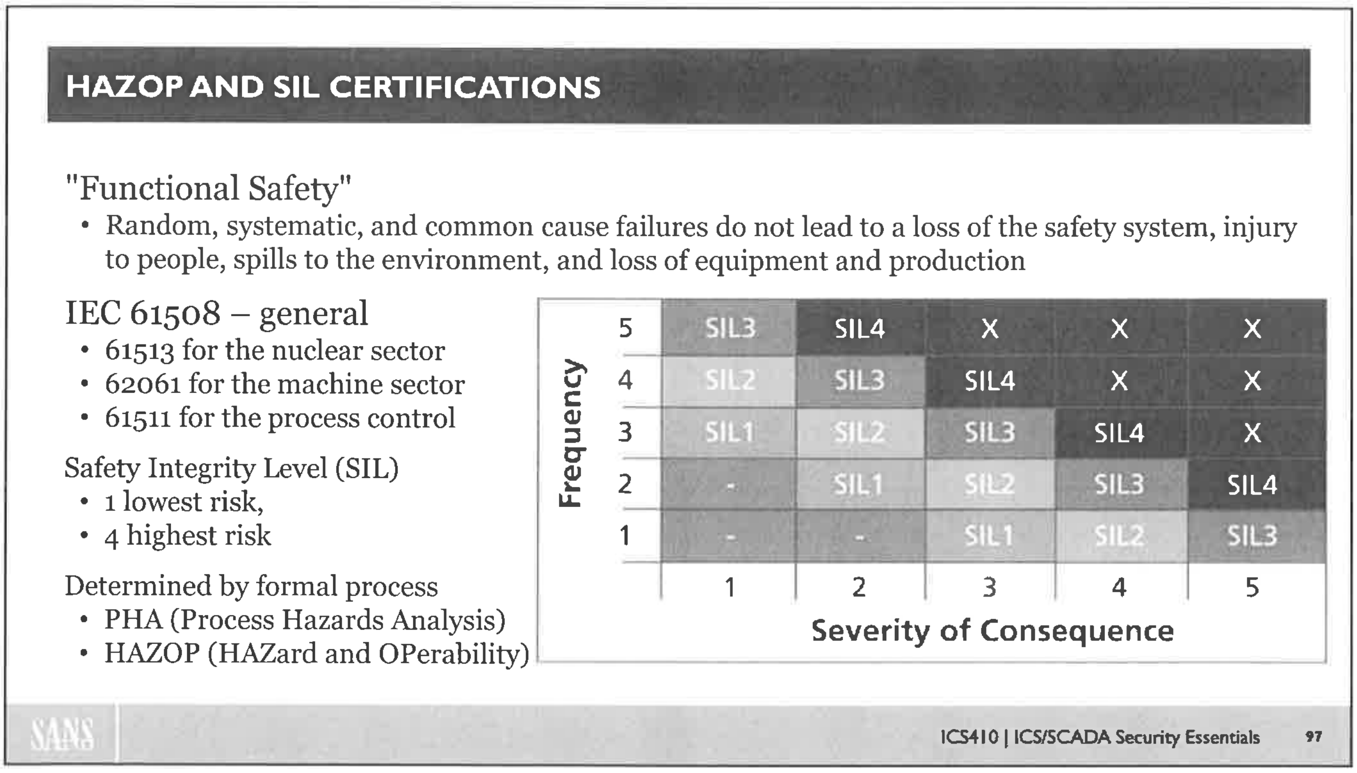
© 2019, Justin Searle

A SIS must have sensors capable of detecting abnormal operating conditions. A logic solver is required to receive the sensor input signals, make appropriate decisions based on the nature of the signals, and change its outputs according to user-defined logic. The logic solver can use electrical, electronic, or programmable electronic equipment, such as relays, trip amplifiers, or programmable logic controllers. Next, the change of the logic solver outputs results in the final elements taking action on the process to bring it to a safe state. Support systems, such as power, instrument air, and communications are generally required for SIS operation. The support systems should be designed to provide the required integrity and reliability.

International standard IEC 61511 was published in 2003 to provide guidance to end users about the application of Safety Instrumented Systems in process industries. This standard is based on IEC 61508, a generic standard for design, construction, and operation of electrical, electronic, and programmable electronic systems. Other industry sectors can also have standards that are based on IEC 61508, such as IEC 62061 (for machinery systems), IEC 62425 (for railway signaling systems), IEC 61513 (for nuclear systems), and ISO 26262 (for road vehicles). ISO 26262 is currently a draft international standard.

Reference:
http://www.exida.com/Alarm-Management

© 2019, Justin Searle



HAZOP AND SIL CERTIFICATIONS

"Functional Safety"

- Random, systematic, and common cause failures do not lead to a loss of the safety system, injury to people, spills to the environment, and loss of equipment and production

IEC 61508 – general

- 61513 for the nuclear sector
- 62061 for the machine sector
- 61511 for the process control

Safety Integrity Level (SIL)

- 1 lowest risk,
- 4 highest risk

Determined by formal process

- PHA (Process Hazards Analysis)
- HAZOP (HAZard and OPerability)

Frequency	1	2	3	4	5
5	SIL3	SIL4	X	X	X
4	SIL2	SIL3	SIL4	X	X
3	SIL1	SIL2	SIL3	SIL4	X
2	-	SIL1	SIL2	SIL3	SIL4
1	-	-	SIL1	SIL2	SIL3

Severity of Consequence

SANS ICS410 | ICS/SCADA Security Essentials 97

"Functional Safety" defines protection against hazards caused by the incorrect functioning of components or systems. A safety system is functionally safe if random, systematic, and common cause failures do not lead to a loss of the safety system and do not result in injury to people, spills to the environment, and loss of equipment and production. Safety is protection against all hazards, but "functional safety" is protection against hazards caused by incorrect function.

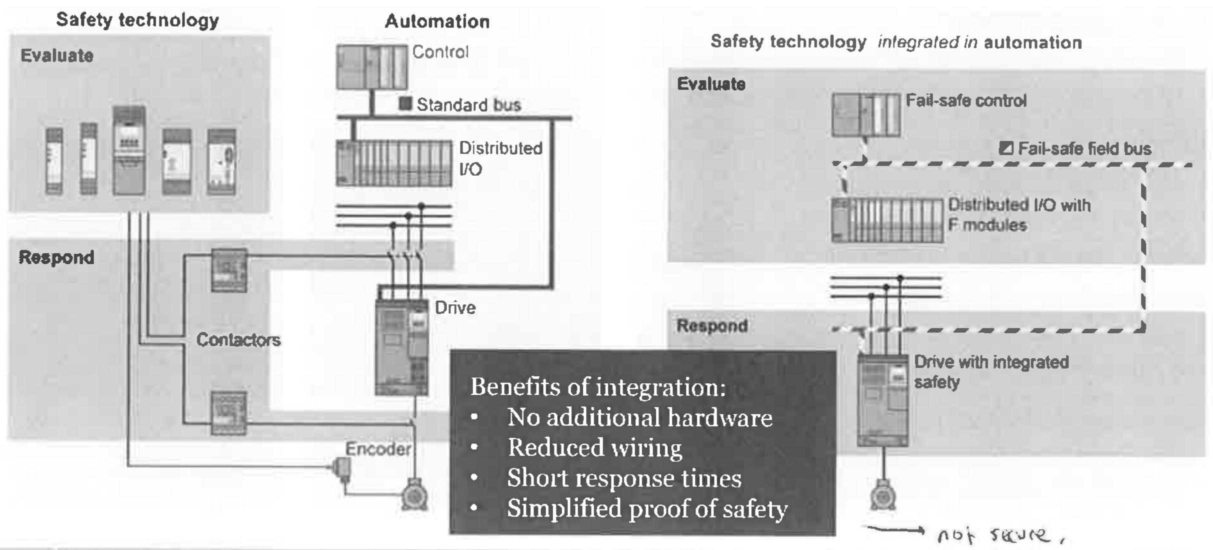
A formal process of hazard identification is performed by the project team engineers and other experts at the completion of the engineering design phase of each section of the process, known as a unit of operation. This team performs a systematic, rigorous, procedural review of each point of possible hazard, or "node," in the completed engineering design. This review and its resulting documentation are called a HAZOP (HAZard and OPerability) study. Safety Integrity Levels (SILs) are defined for the SISs. Based on HAZOP study recommendations and the SIL rating of the SISs, the engineering, the BPCS, and the SIS designs for each unit operation can be finalized.

- SIL ratings from 1 (lowest risk) to 4 (highest risk)
- HAZOP measures the probability of failure on demand calculations and failures per hour calculations

PHA (Process Hazards Analysis) results and other consequence/process risk analysis tools are a great place to identify priorities for protection. Understanding critical equipment or dangerous (where extrinsic danger exists) parts of a process can provide important security misuse cases that need to be fully considered. (For example, the PHA considers questions such as, "What if a particular valve fails to close and shows the wrong position?") The PHA/HAZOP also provides insight into the safeguards and safety controls that have been implemented (for example, use of isolation techniques, leak detection systems, and suppression systems). The central IEC standard is IEC 61508 with various applications and industries have their own as noted in the slide above. The certification process includes management of functional safety, hardware assessments, software assessments, and testing. All top-tier manufacturers get products certified.

Image Source:
https://www.crossco.com/blog/determining-safety-integrity-levels-sil-your-process-application

TRADITIONAL SAFETY SYSTEM VS. INTEGRATED SAFETY SYSTEM



Benefits of integration:
- No additional hardware
- Reduced wiring
- Short response times
- Simplified proof of safety

→ not secure,

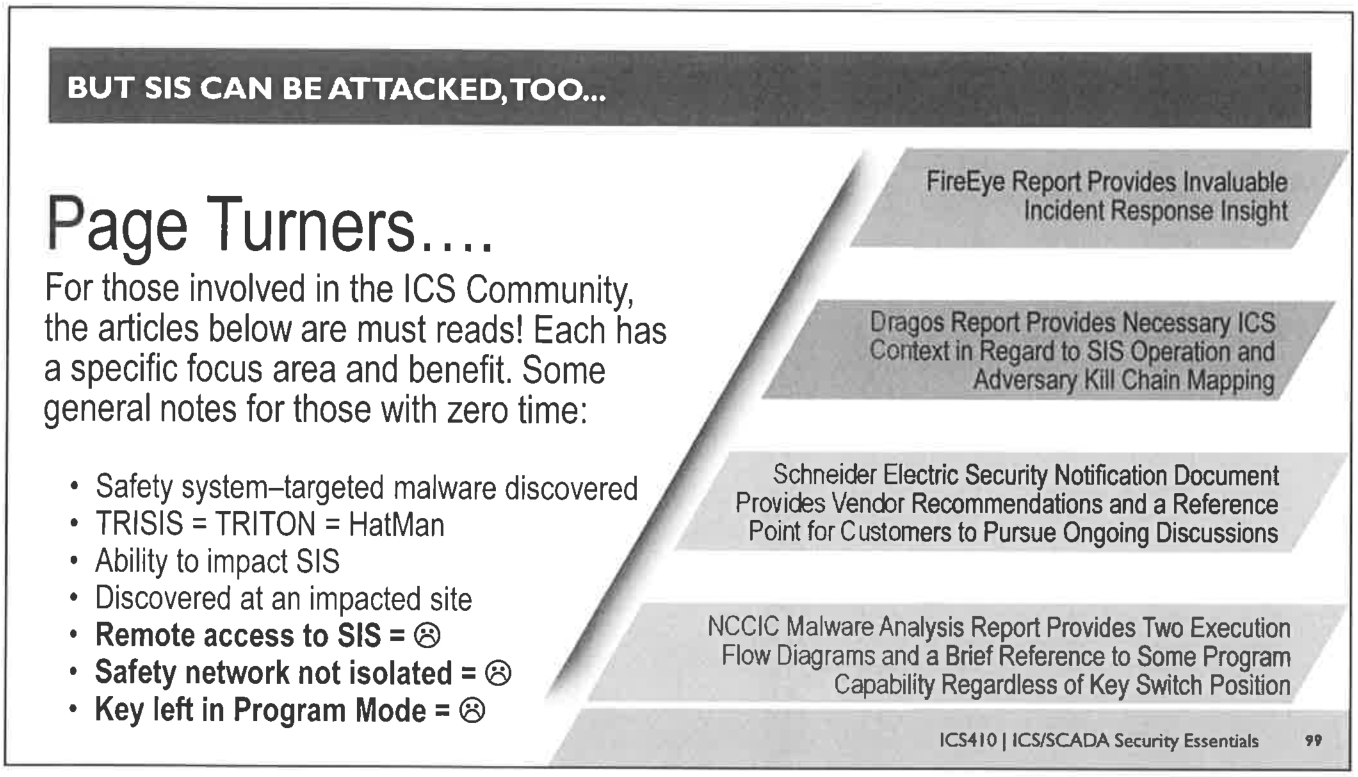
"The safety circuit used to have its own set of logic around the machine and equipment. The movement of integrated safety is bringing safety right into the devices and equipment," Craig Nelson, product marketing manager at Siemens Industry US, told DesignNews. "Everything is on the same system now. It becomes a lot more economical, with more safety, more diagnostics, increased productivity, more functionality. We couldn't do that the traditional way."

"A lot of it comes down to the economics. The company is protecting its reputation. They want the higher level of safety that comes with integrated safety," Nelson said. "It's more economical to put safety in, and it's more economical to maintain safety."

References:
Siemens – SINAMICS Overview
https://goo.gl/TbrEk9

DesignNews – "Integrated Safety Breaks the Cost-Savings Barrier at Plants"
https://goo.gl/fro2KS

© 2019, Justin Searle



BUT SIS CAN BE ATTACKED, TOO...

Page Turners....

For those involved in the ICS Community, the articles below are must reads! Each has a specific focus area and benefit. Some general notes for those with zero time:

- Safety system–targeted malware discovered
- TRISIS = TRITON = HatMan
- Ability to impact SIS
- Discovered at an impacted site
- **Remote access to SIS = ☹**
- **Safety network not isolated = ☹**
- **Key left in Program Mode = ☹**

FireEye Report Provides Invaluable Incident Response Insight

Dragos Report Provides Necessary ICS Context in Regard to SIS Operation and Adversary Kill Chain Mapping

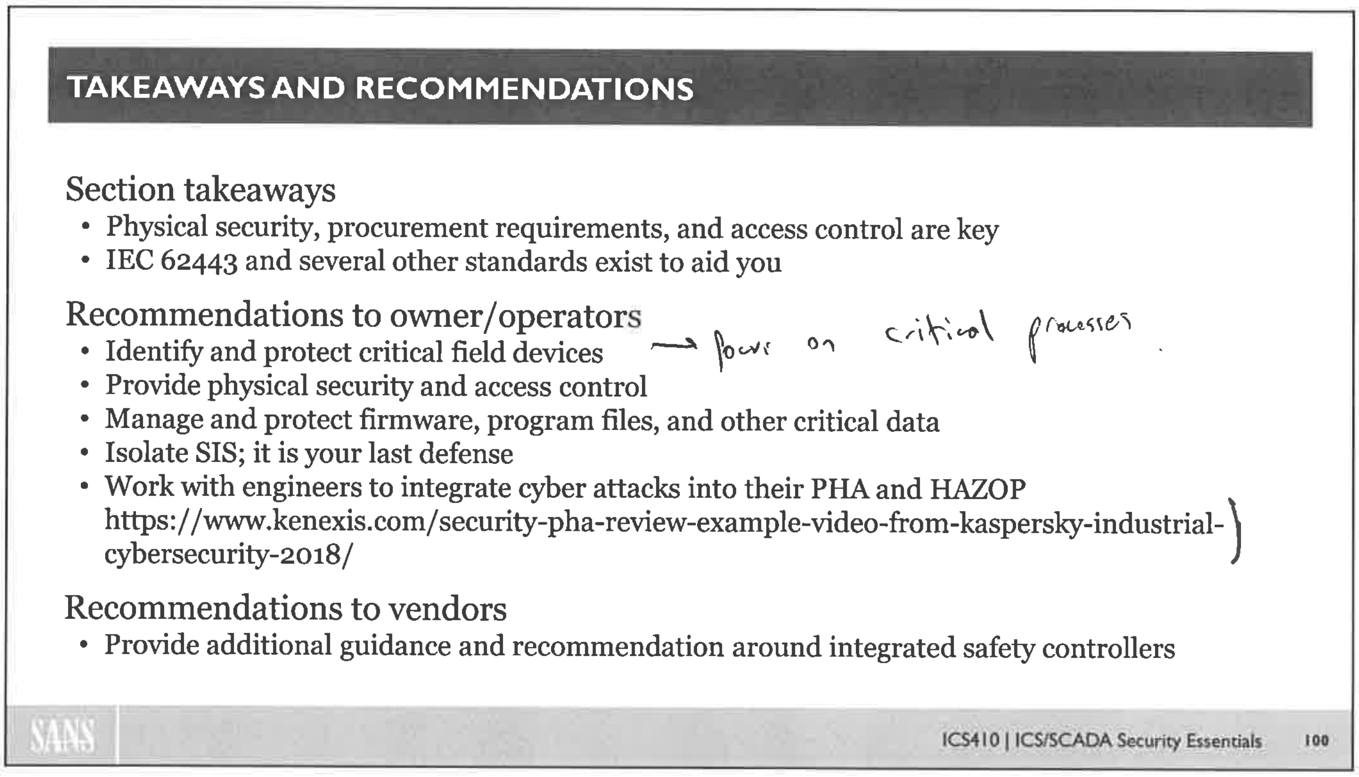
Schneider Electric Security Notification Document Provides Vendor Recommendations and a Reference Point for Customers to Pursue Ongoing Discussions

NCCIC Malware Analysis Report Provides Two Execution Flow Diagrams and a Brief Reference to Some Program Capability Regardless of Key Switch Position

ICS410 | ICS/SCADA Security Essentials 99

Read

References:
https://www.fireeye.com/blog/threat-research/2017/12/attackers-deploy-new-ics-attack-framework-triton.html
https://dragos.com/blog/trisis/TRISIS-01.pdf
https://www.schneider-electric.com/en/download/document/SEVD-2017-347-01/
ICS-CERT Analysis Report - http://bit.ly/2IRjapN



TAKEAWAYS AND RECOMMENDATIONS

Section takeaways

- Physical security, procurement requirements, and access control are key
- IEC 62443 and several other standards exist to aid you

Recommendations to owner/operators → focus on critical processes.

- Identify and protect critical field devices
- Provide physical security and access control
- Manage and protect firmware, program files, and other critical data
- Isolate SIS; it is your last defense
- Work with engineers to integrate cyber attacks into their PHA and HAZOP https://www.kenexis.com/security-pha-review-example-video-from-kaspersky-industrial-cybersecurity-2018/ }

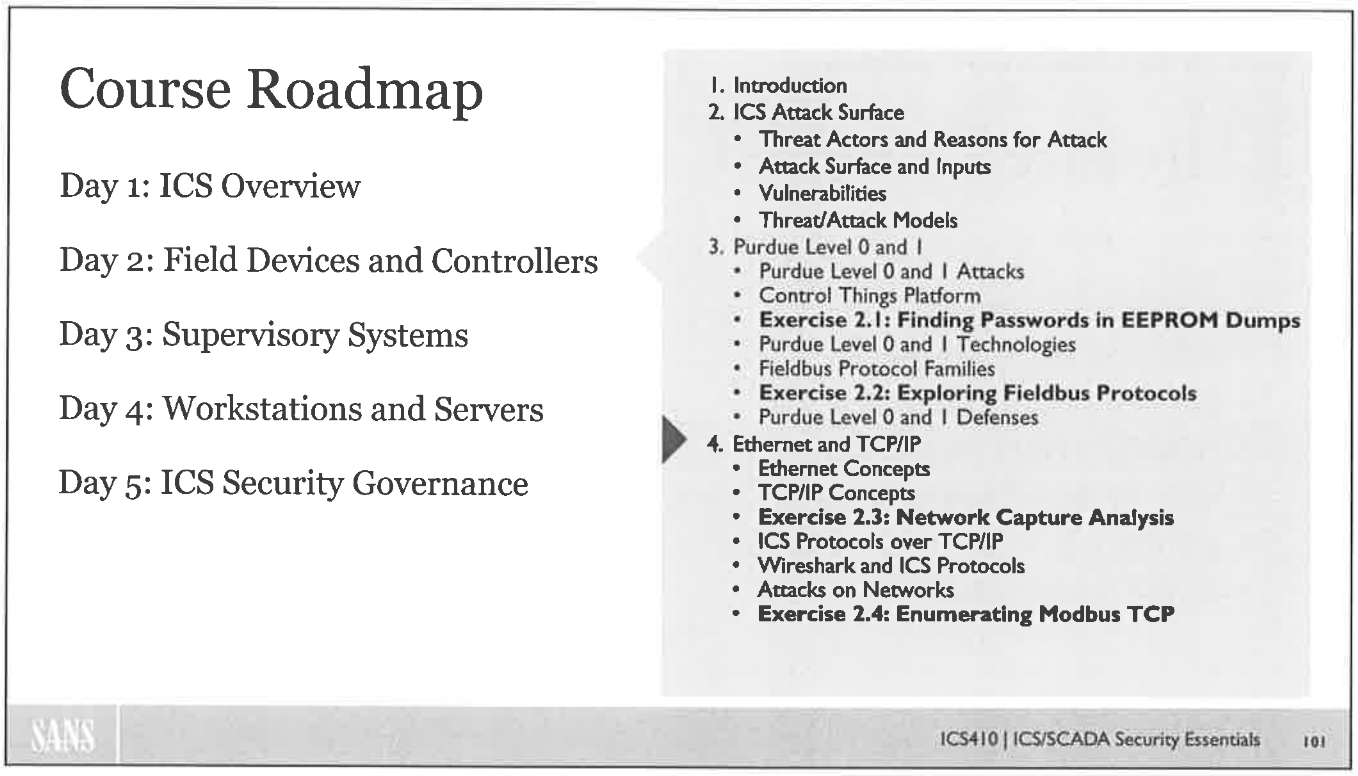
Recommendations to vendors

- Provide additional guidance and recommendation around integrated safety controllers

SANS ICS410 | ICS/SCADA Security Essentials 100

This page intentionally left blank.

© 2019, Justin Searle



Course Roadmap

Day 1: ICS Overview

Day 2: Field Devices and Controllers

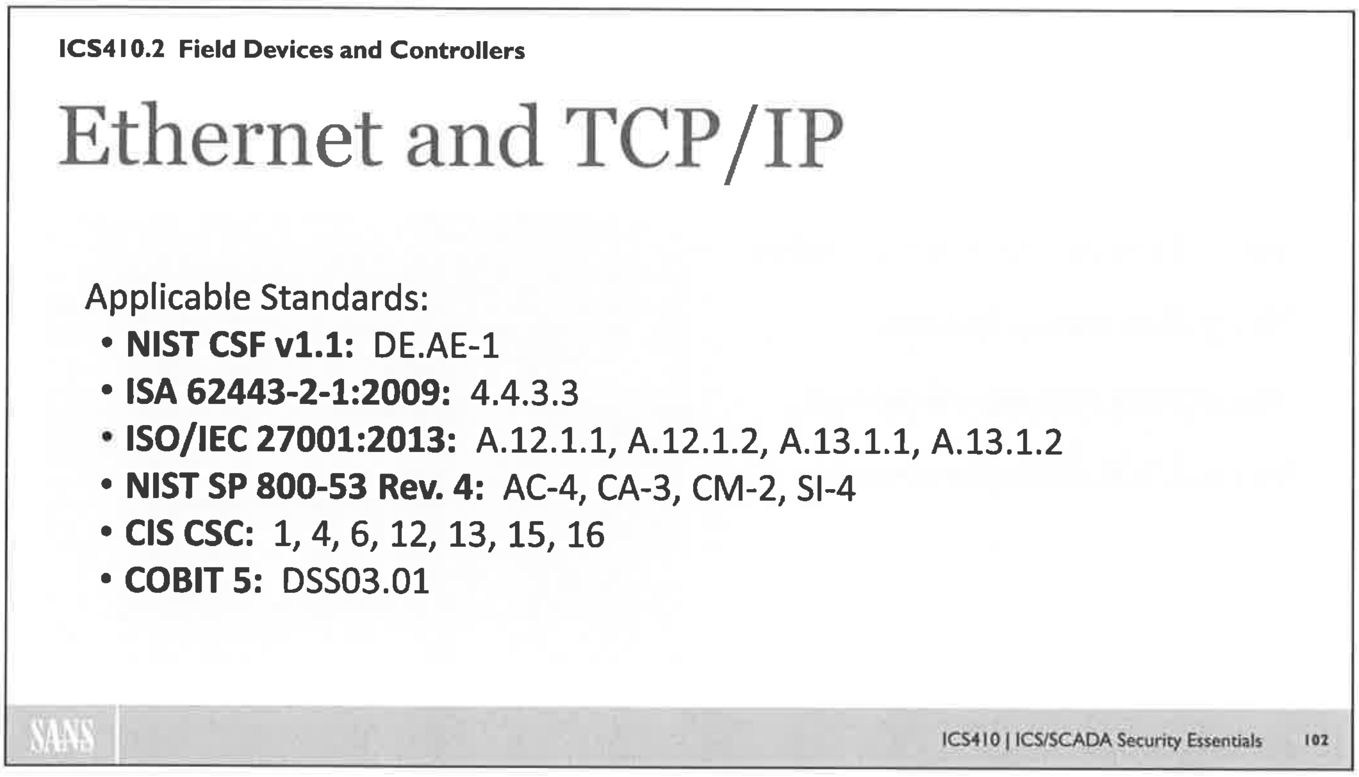
Day 3: Supervisory Systems

Day 4: Workstations and Servers

Day 5: ICS Security Governance

SANS

This page intentionally left blank.


ICS410.2 Field Devices and Controllers

Ethernet and TCP/IP

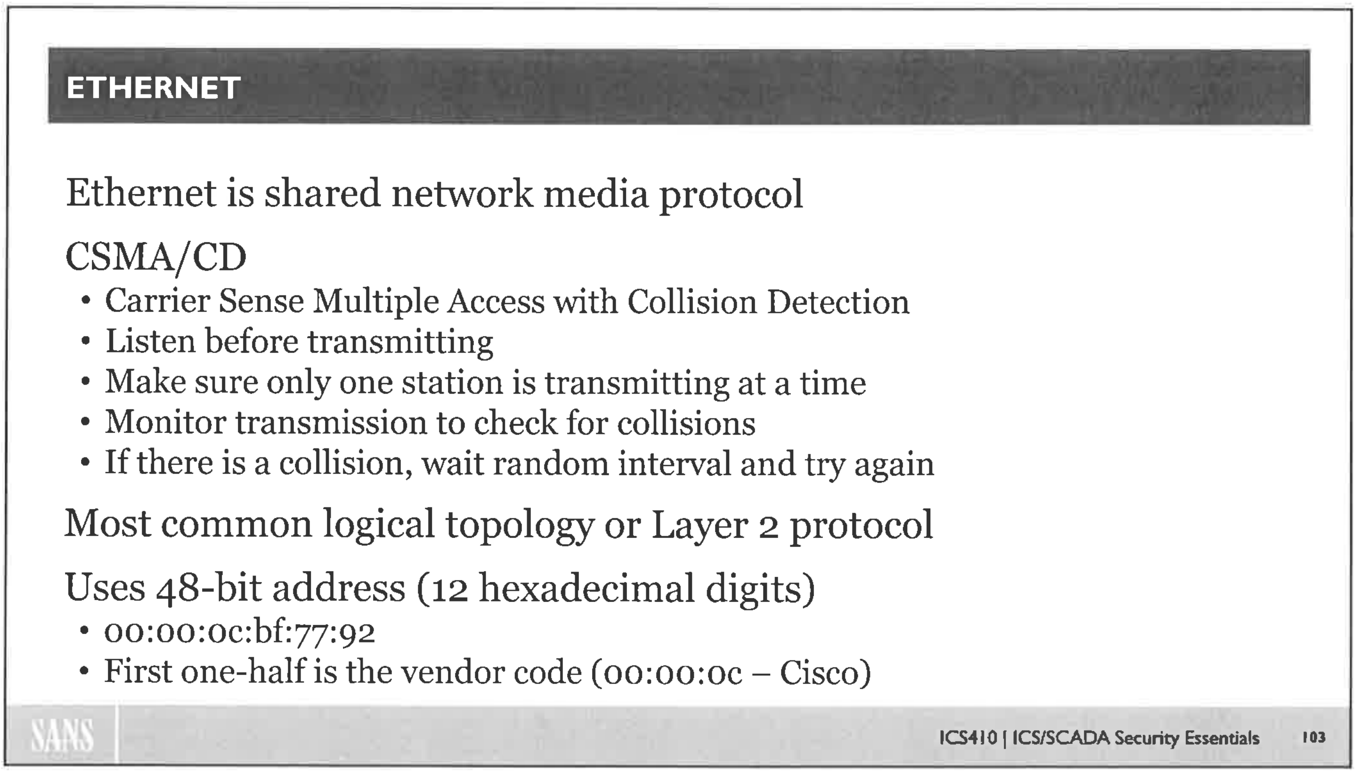
Applicable Standards:

- **NIST CSF v1.1:** DE.AE-1
- **ISA 62443-2-1:2009:** 4.4.3.3
- **ISO/IEC 27001:2013:** A.12.1.1, A.12.1.2, A.13.1.1, A.13.1.2
- **NIST SP 800-53 Rev. 4:** AC-4, CA-3, CM-2, SI-4
- **CIS CSC:** 1, 4, 6, 12, 13, 15, 16
- **COBIT 5:** DSS03.01

SANS ICS410 | ICS/SCADA Security Essentials 102

This page intentionally left blank.

© 2019, Justin Searle



ETHERNET

Ethernet is shared network media protocol

CSMA/CD

- Carrier Sense Multiple Access with Collision Detection
- Listen before transmitting
- Make sure only one station is transmitting at a time
- Monitor transmission to check for collisions
- If there is a collision, wait random interval and try again

Most common logical topology or Layer 2 protocol

Uses 48-bit address (12 hexadecimal digits)

- 00:00:0c:bf:77:92
- First one-half is the vendor code (00:00:0c – Cisco)

SANS ICS410 | ICS/SCADA Security Essentials 103

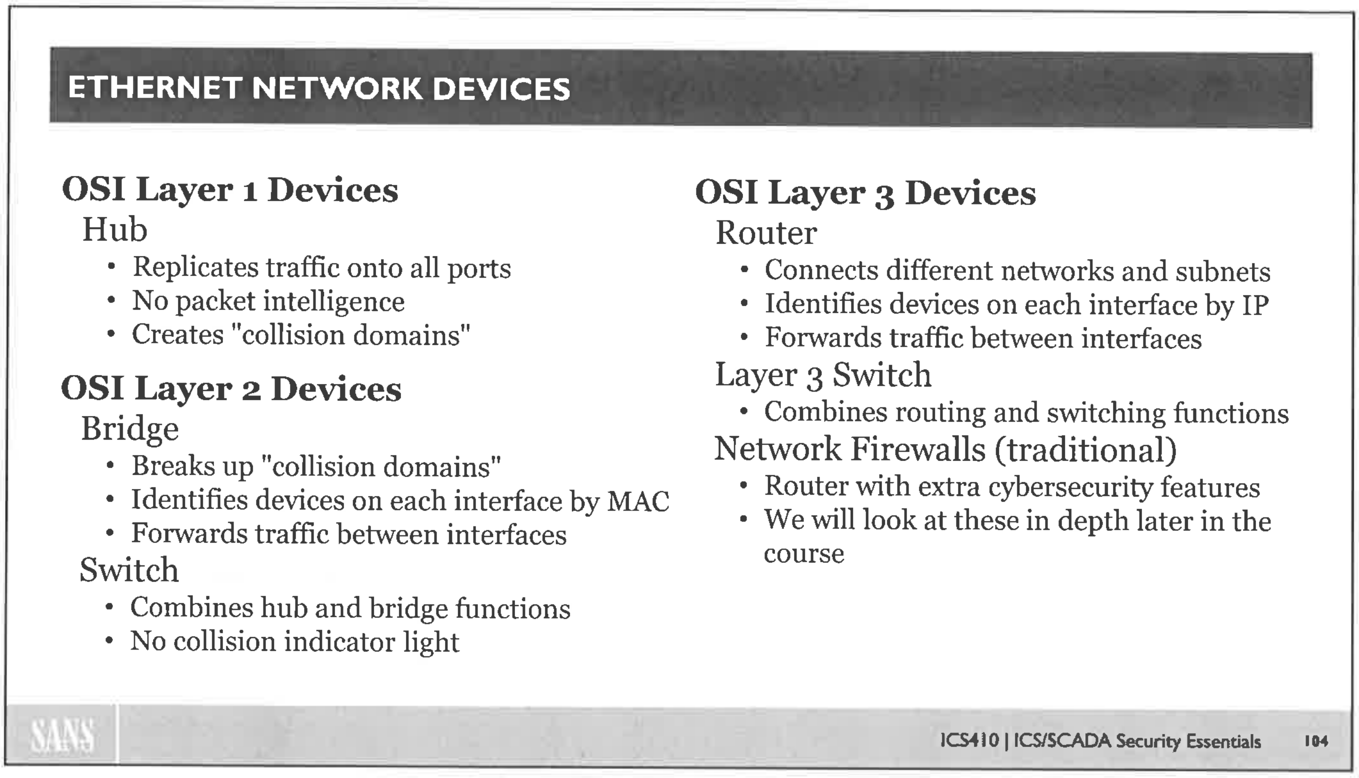
Ethernet is by far the most popular media access protocol (Layer 2 protocol) currently used on LANs. In fact, it is nearly ubiquitous for networking other than on the backbone itself. A chunk of data transmitted by Ethernet over the wire is called a frame. On an Ethernet network, only a single node should be transmitting a frame at a time. If multiple systems are transmitting simultaneously, a collision occurs that can cause both signals to fail and require the systems to retransmit their frames.

To keep the number of collisions to a minimum, a system is required to check whether anyone else is already transmitting before placing a frame on the wire. If another system's signal is already on the wire, the system is expected to wait, according to the algorithm designed, to give each node a fair shot at using the network. If the line is clear, the system generates a signal and monitors the transmission to make sure that no collision occurred. These properties are summarized under Ethernet's designation as a Carrier Sense Multiple Access/Collision Detection (CSMA/CD) protocol.

Ethernet specifications actually define more than just protocols for sending signals over the wire. Other properties include cabling requirements for transferring data at desired rates and the maximum length of the wire segment. In addition, Ethernet standards specify which physical topology should be used for a particular type of Ethernet communication.

A MAC address typically is written as 12 hexadecimal digits grouped in pairs (bytes): 00-00-0c-34-17-a3 is an example of a typical MAC address. The first 24 bits of the address contain a vendor code, and the second one-half of the address is a unique number assigned by the vendor. MAC addresses generally are burned into NICs during the manufacturing process. The Cisco vendor code, for example, is 00-00-0c, and Sun Microsystems' vendor code is 08-00-20.

Reference:
http://www.ieee802.org/3/


ETHERNET NETWORK DEVICES

OSI Layer 1 Devices

Hub

- Replicates traffic onto all ports
- No packet intelligence
- Creates "collision domains"

OSI Layer 2 Devices

Bridge

- Breaks up "collision domains"
- Identifies devices on each interface by MAC
- Forwards traffic between interfaces

Switch

- Combines hub and bridge functions
- No collision indicator light

OSI Layer 3 Devices

Router

- Connects different networks and subnets
- Identifies devices on each interface by IP
- Forwards traffic between interfaces

Layer 3 Switch

- Combines routing and switching functions

Network Firewalls (traditional)

- Router with extra cybersecurity features
- We will look at these in depth later in the course

SANS ICS410 | ICS/SCADA Security Essentials 104

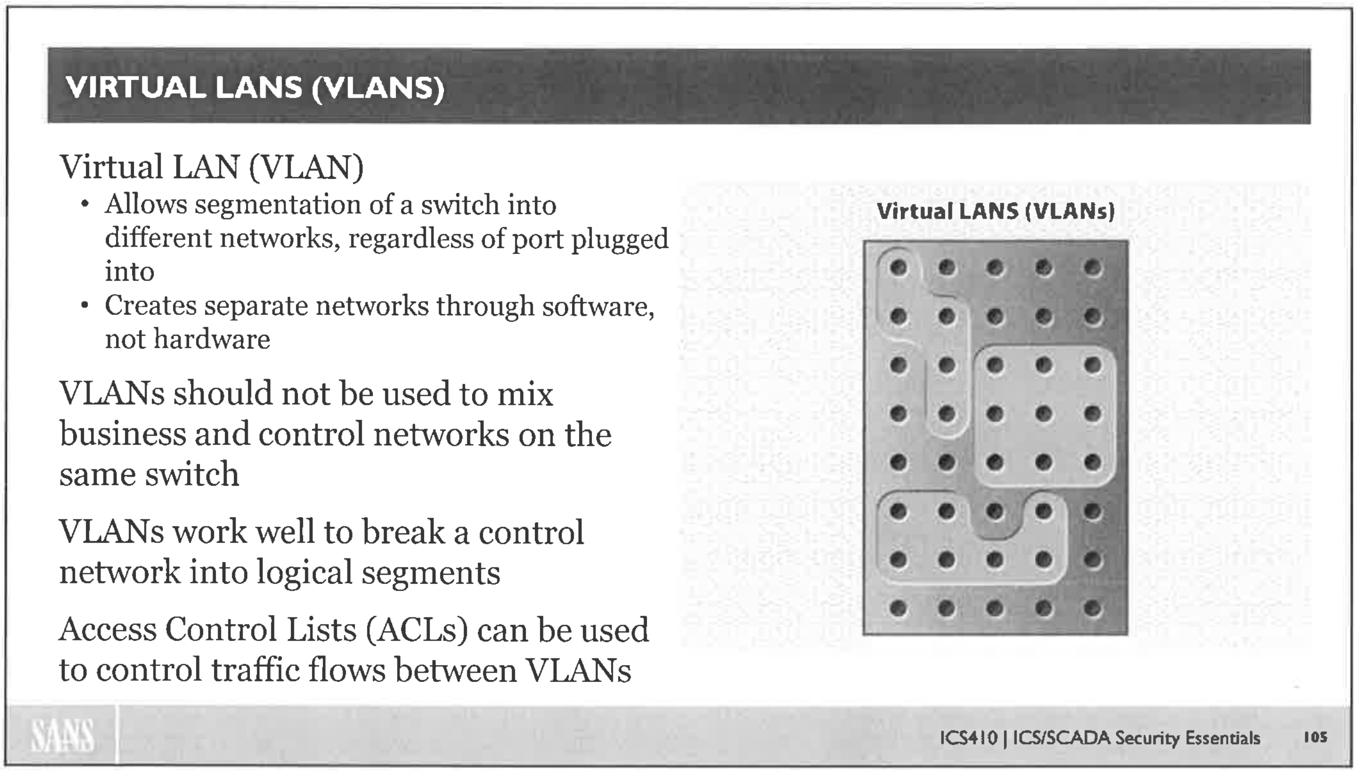
A hub operates by "repeating" data that it receives on one port to its other ports. As a result, a data frame transmitted by one system is retransmitted to all other systems connected to the hub. A classic hub does not have traffic-monitoring capabilities and cannot control which ports should or should not receive the frame, forming a large collision domain.

A bridge is used to connect two physical segments of a network in much the same way as an over-the-water bridge connects two sections of a road. When a bridge receives a data frame on one of its ports, it makes a decision whether the data should be sent to the other port. This functionality allows a bridge to automatically control the flow of data between network segments that it connects. To decide when to replicate the frames from one port to another, the bridge learns which systems reside on which network segment. It accomplishes this by automatically recording the MAC addresses of frames that pass through it to construct a table that maps MAC addresses to network segments.

A network switch combines the functionality of a hub and a bridge into a single device. If you think of a switch as a bridge with more than two ports, you will get the idea. Like a hub, a switch can retransmit data to multiple ports. In addition, an Ethernet switch keeps track of MAC addresses attached to each of its ports, which grants it the traffic control capabilities of a bridge. By monitoring and controlling traffic between its ports, a switch will direct only a data frame to the system or network segment for which it is destined, narrowing each port to its own collision domain. Sniffing becomes ineffective with switches.

Routers are often considered to be perimeter devices because they interconnect logical networks. A switch or a bridge, however, connects physical segments that reside on the same logical network. Much of the internet relies on routers for determining which paths packets should take to get from one network to another. Similar to a switch or a bridge, a router makes decisions regarding where to direct data that passes through it. However, whereas a switch makes its decisions by tracking MAC addresses, a router operates on a higher layer (Layer 3) by looking at IP addresses when forwarding packets.

Unlike a switch or a bridge, which transmits traffic to unknown destinations, a router drops traffic if it does not know where to send it. Routers need to be explicitly configured with information that defines paths for directing traffic to all reachable networks. The router stores this information in a routing table, which it uses to make packet-forwarding decisions. Routers drop all local MAC-level broadcast traffic by default, thereby contributing to their effectiveness at isolating network traffic.



VIRTUAL LANS (VLANS)

Virtual LAN (VLAN)

- Allows segmentation of a switch into different networks, regardless of port plugged into
- Creates separate networks through software, not hardware

VLANs should not be used to mix business and control networks on the same switch

VLANs work well to break a control network into logical segments

Access Control Lists (ACLs) can be used to control traffic flows between VLANs

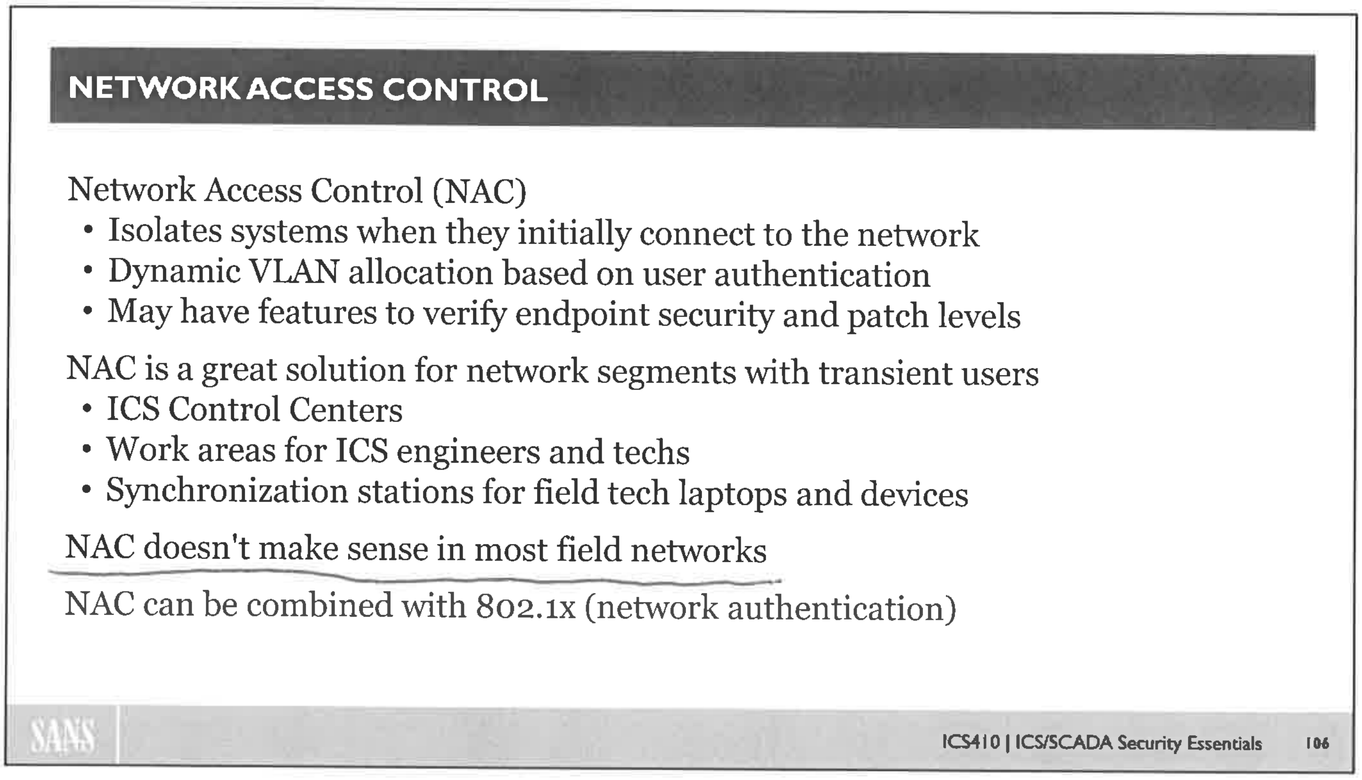
Virtual LANS (VLANs)

ICS410 | ICS/SCADA Security Essentials 105

A virtual LAN (VLAN) allows you to take a large physical switch and, regardless of where systems are plugged in, segment them into different networks based on function or access required. For example, if 100 employees and servers are all plugged into the same switch, you can limit the access or visibility by placing them on separate segments. Least privilege states that you give someone the least access she needs to do her job.

VLANs should not be used to mix business and control networks on the same switch. VLANs are logical network controls and can be misconfigured or fall to network attacks, thus, different physical switches and VLAN schemes should be used separately by business and control networks.

VLANs work well in ICS because they allow you to break a control network into logical segments. For instance, you can use a single switch in a remote station yet still keep traffic from different station systems separate if desired. VLANs also allow you to create different security zones in control centers, separating workstations, servers, printers, and weather/news polling systems into separate segments. In all these cases, Access Control Lists (ACLs) can be used to control traffic flows between VLANs as long as this does not have a negative impact on your required traffic latency.



NETWORK ACCESS CONTROL

Network Access Control (NAC)
- Isolates systems when they initially connect to the network
- Dynamic VLAN allocation based on user authentication
- May have features to verify endpoint security and patch levels

NAC is a great solution for network segments with transient users
- ICS Control Centers
- Work areas for ICS engineers and techs
- Synchronization stations for field tech laptops and devices

NAC doesn't make sense in most field networks

NAC can be combined with 802.1x (network authentication)

SANS ICS410 | ICS/SCADA Security Essentials 106

Network Access Control (NAC) allows systems to be placed on isolated VLANs until they have been scanned and properly patched, thus limiting their exposure to infecting other systems. A key motto of security: Prevention is ideal, but detection is a must. There is no way to completely prevent an attack from occurring. However, if a single system becomes infected, you can prevent it from connecting to critical networks through NAC and stop it from spreading to a large number of hosts through VLANs. Although NAC and VLANs both have value by themselves, together they provide a robust measure of protection for end-user networks like field technician subnets.

NAC is a good solution for network segments where workstations and laptops are used, such as in ICS Control Centers, work areas for ICS engineers and techs, and synchronization stations for field tech laptops and devices. This provides an extra security layer to protect against unapproved devices connecting to these network interfaces or communicating on the control network. However, NAC in its current offerings does not usually make sense for most field networks.

© 2019, Justin Searle

ETHERNET IN ICS

Several ICS serial protocols were converted to take advantage of Ethernet infrastructures

- Some designed to use TCP/IP, which we'll discuss later
- Others designed to sit directly on top of Ethernet

Common Ethernet-only based ICS protocols

- Foundation Fieldbus High Speed Ethernet (HSE)
- PROFINET (RT, IRT)
- CC-Link IE (Control, Field, Safety)
- EtherCAT
- MECHATROLINK-III
- IEC 61850 Goose and Sampled Values (SV)

Benefits of these process/fieldbus protocols

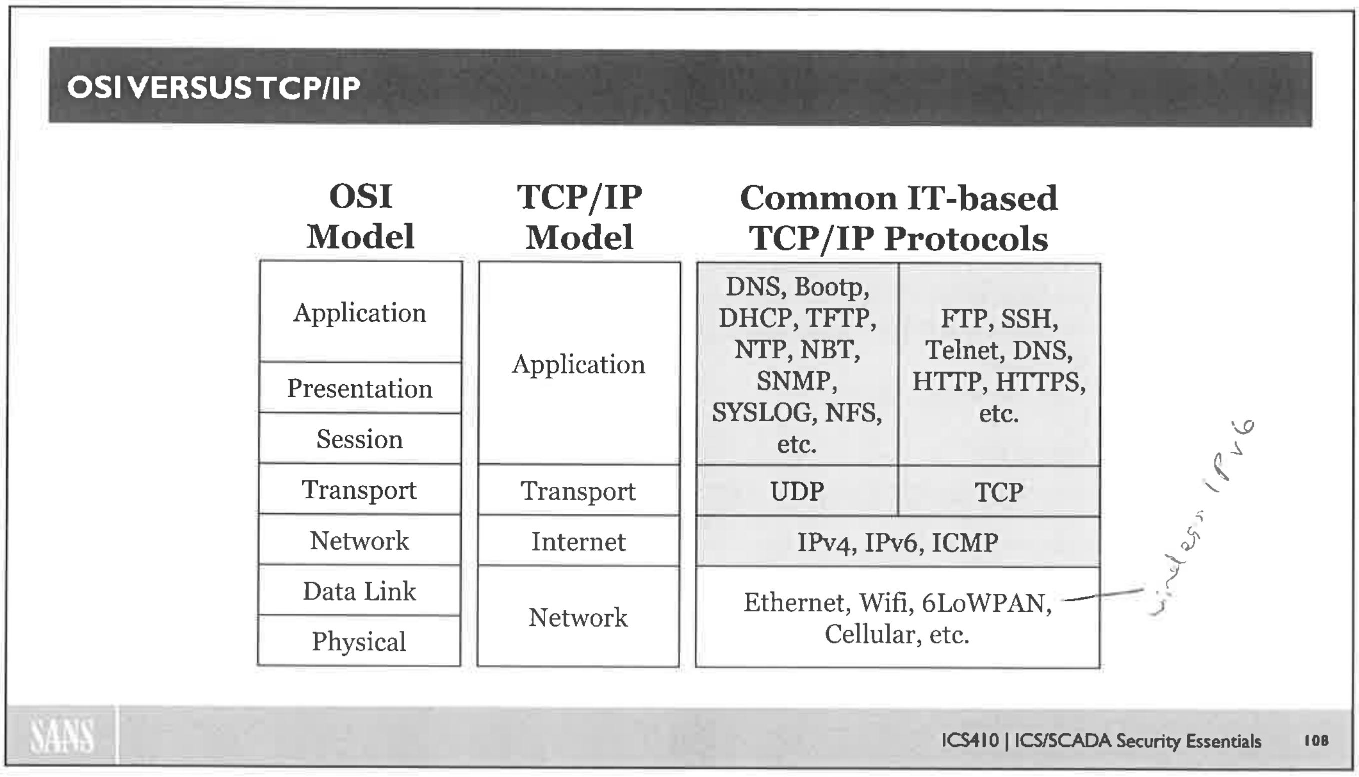
- No TCP/IP overhead
- Low latency (take advantage of Gigabit speeds)
- Nonroutable (boon for security)

The change from electromechanical to digital has brought with it greater reliability and availability along with efficiency gains. The world moved from point-to-point wiring to fieldbus approaches. Many ICS Vendors now implement hybrid Ethernet or fieldbus networks to connect with equipment and ICS components and share information between control systems. Many unique protocols were developed through the early years of automation. It is much more common now for vendors to implement open protocols or converge on the use of a smaller number of widely adopted protocols.

ICS applications carry specific requirements and possess complexities that account for the development of various industrial protocols. Example requirements include:

- Real-time operations as actions and state changes are detected and acted upon in a specific time frame.
- Deterministic operations as actions need to occur in a predetermined sequence.
- Reliable operations as levels of redundancy are required to account for failures to achieve system availability goals.
- Safety as the system must be able to detect and respond to state changes or conditions to maintain the safety of the process.
- Ruggedized connectors and cabling as the environment may include high temperatures, dust, and more.
- Remote accessibility as some control components may be difficult to access and may require minimal human touch.
- Secure operations as some applications are beginning to demand a certain level of security.

This can be seen in the IEC 61850 standard, which provides mapping to a number of protocols for substation automation capability.



OSI VERSUS TCP/IP

OSI Model	TCP/IP Model	Common IT-based TCP/IP Protocols	
Application	Application	DNS, Bootp, DHCP, TFTP, NTP, NBT, SNMP, SYSLOG, NFS, etc.	FTP, SSH, Telnet, DNS, HTTP, HTTPS, etc.
Presentation			
Session			
Transport	Transport	UDP	TCP
Network	Internet	IPv4, IPv6, ICMP	
Data Link	Network	Ethernet, Wifi, 6LoWPAN, Cellular, etc.	
Physical			

wireless IPv6

SANS ICS410 | ICS/SCADA Security Essentials 108

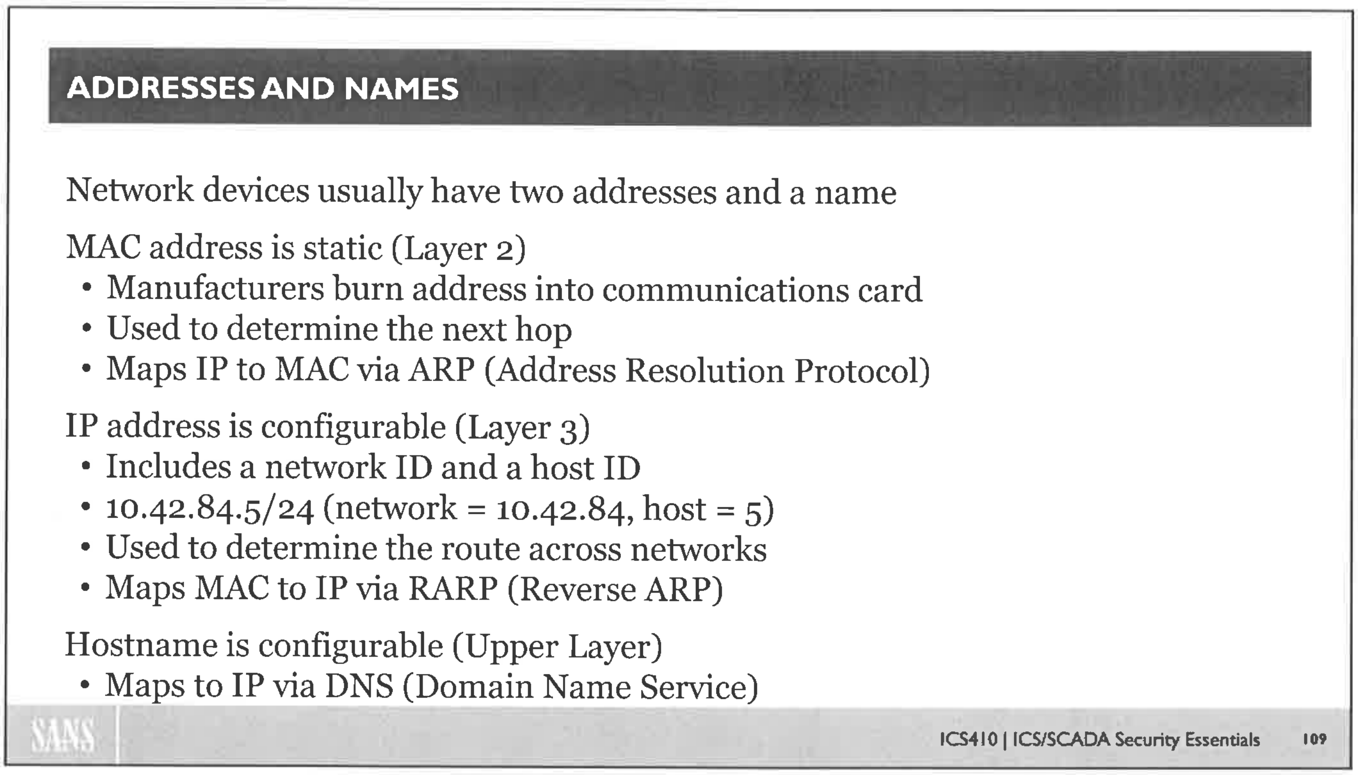
Many people ask which protocol stack is better, OSI or TCP/IP? The answer is both. In practice, both are utilized and need to be understood by the student. When talking about the protocols and referencing the layer a protocol operates at, OSI is utilized. For example, we always say that routing is a Layer 3 function. However, when we actually implement the protocols, we use the layered functionality of the TCP/IP model.

The TCP/IP stack has only four layers: The network layer, the internet layer, the transport layer, and the application layer. Even though the stack has only four layers as compared to the seven-layer OSI model, it still performs the same functions. It just means that because there are fewer layers, each layer has to do a little more work.

As you can see, the OSI model is more granular. The OSI model splits apart some functionality that was combined in the TCP/IP model. The network layer in the TCP/IP model is composed of both the physical and the data link layers in the OSI model, while the application layer in TCP/IP encompasses the application, presentation, and session layers of OSI. The OSI model is more detailed because it was designed to support protocols other than just TCP/IP. By creating more layers, the designers made it easier to break down the functionality of each protocol and build more specific interfaces and linkages between the layers.

Even though each model breaks down the functionality a bit differently, you should realize that no matter which model you use, it must perform all the functions required to take a piece of application data, place it into a packet, put that packet on the wire, and deliver it safely and efficiently to its destination.

© 2019, Justin Searle



ADDRESSES AND NAMES

Network devices usually have two addresses and a name

MAC address is static (Layer 2)
- Manufacturers burn address into communications card
- Used to determine the next hop
- Maps IP to MAC via ARP (Address Resolution Protocol)

IP address is configurable (Layer 3)
- Includes a network ID and a host ID
- 10.42.84.5/24 (network = 10.42.84, host = 5)
- Used to determine the route across networks
- Maps MAC to IP via RARP (Reverse ARP)

Hostname is configurable (Upper Layer)
- Maps to IP via DNS (Domain Name Service)

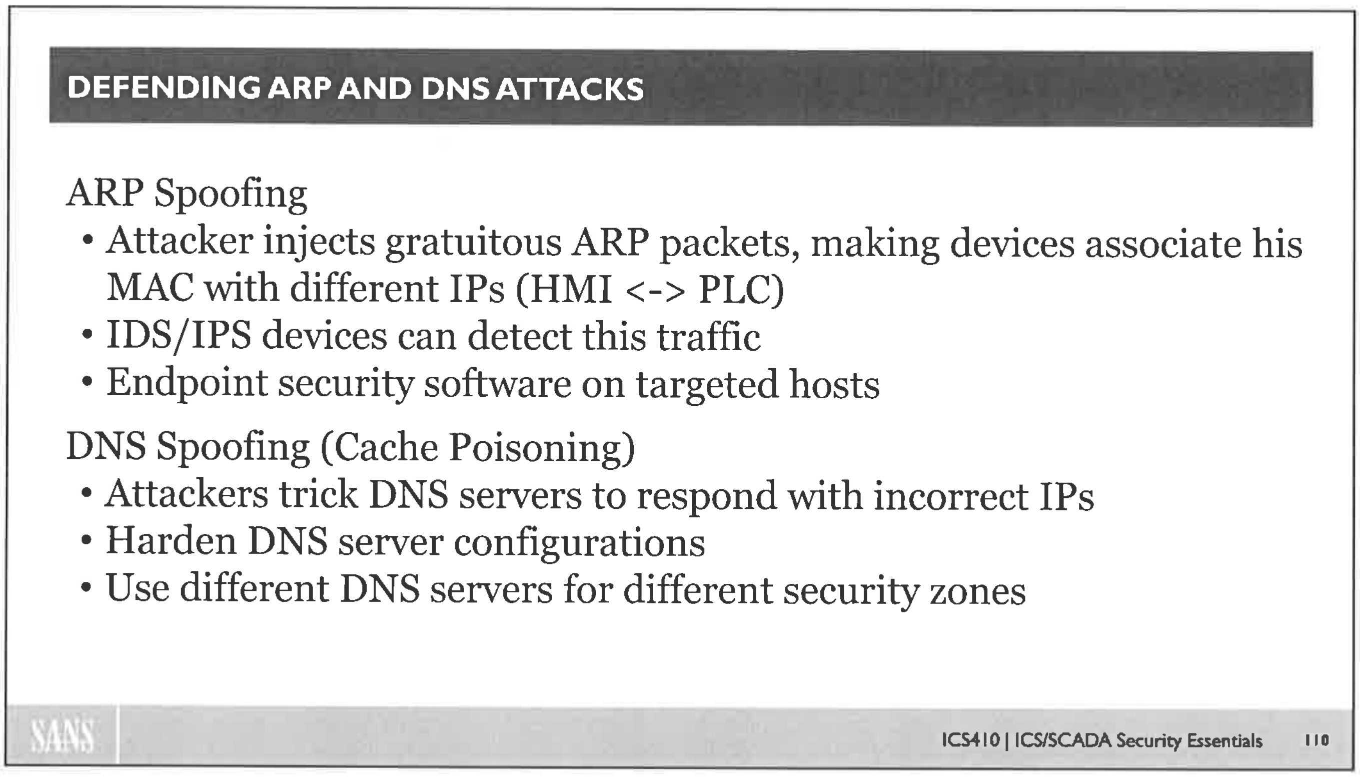
SANS ICS410 | ICS/SCADA Security Essentials 109

It is important to remember that network interfaces have two addresses associated with them, namely a hardware address and a software address. Most devices also have a hostname associated with them, and in the case of systems on the internet, that hostname is a fully qualified domain name or FQDN.

A hardware address is the data link layer (Layer 2) address associated with the network interface. If the network is a frame relay network, for example, the hardware address is a 10-bit data link connection identifier (DLCI). If the host is attached to an IEEE LAN (such as 802.3/Ethernet), the hardware address is a 48-bit media access control (MAC) address. MAC addresses are uniquely allocated to each network interface card (NIC) and are not meant to change; however, if needed, MAC addresses can still be changed through software, driver, or OS configurations.

A software address, such as an IP address, is the network layer protocol address. This address will be specific to the network layer protocol and actual network to which the host is attached. If the computer supports multiple network layer protocols, the NIC will have multiple software addresses. An IP address, as discussed earlier, is 32 bits, or 4 bytes in length and is usually written in dotted-decimal format, such as 10.5.10.37. An IP address is hierarchical for routing purposes; the first part is the network identifier (NET_ID) and the second part is the host identifier (HOST_ID). Historically, IP used classful addresses, where the Class A, B, or C NET_ID was 8, 16, or 24-bits long, respectively. Today, classless addressing should be assumed, where a variable-length subnet mask or CIDR notation indicates the number of bits in the NET_ID.

On a final note, recall that every IP interface has an IP address and a subnet mask. Each IP device is also configured with the IP addresses of the name servers and default gateway (that is, the router to use if the device does have reason to send the packet elsewhere).



DEFENDING ARP AND DNS ATTACKS

ARP Spoofing

- Attacker injects gratuitous ARP packets, making devices associate his MAC with different IPs (HMI <-> PLC)
- IDS/IPS devices can detect this traffic
- Endpoint security software on targeted hosts

DNS Spoofing (Cache Poisoning)

- Attackers trick DNS servers to respond with incorrect IPs
- Harden DNS server configurations
- Use different DNS servers for different security zones

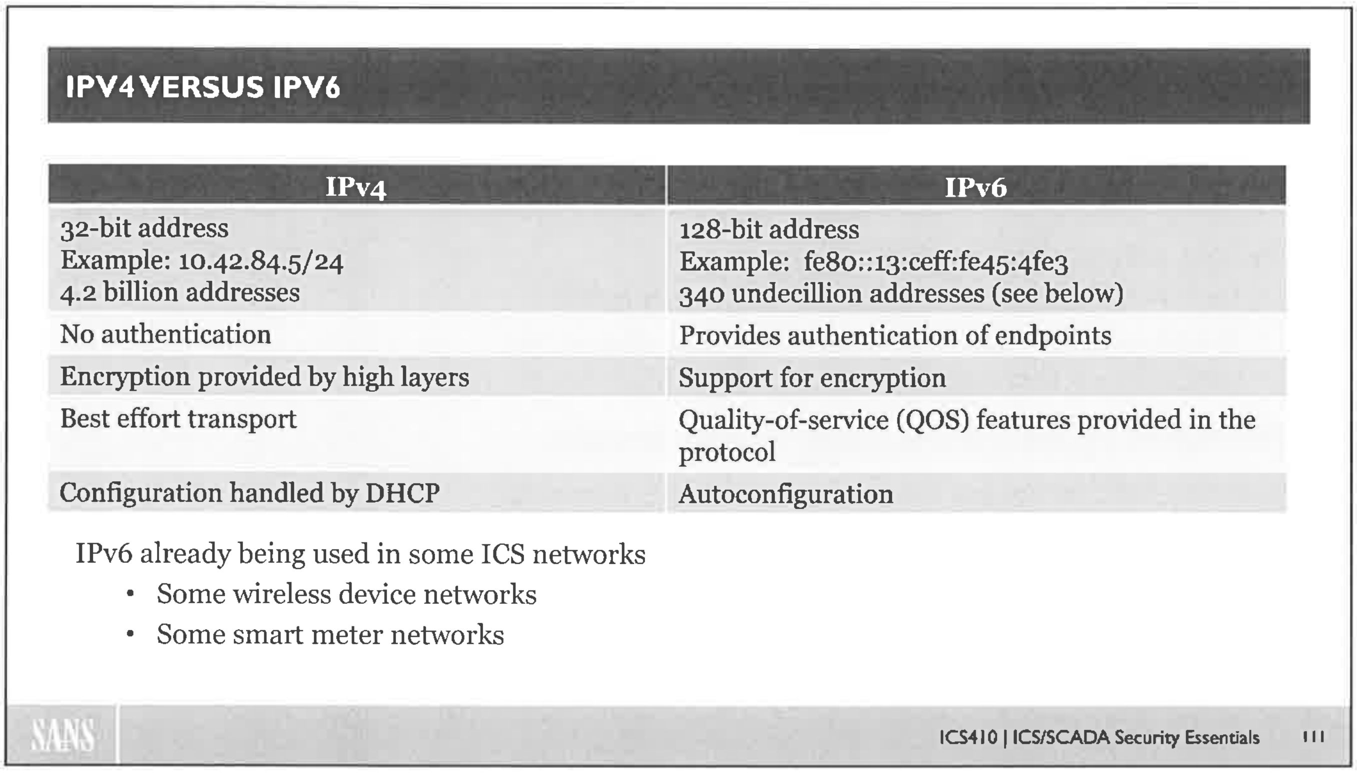
SANS ICS410 | ICS/SCADA Security Essentials 110

ARP Spoofing

Using software like Ettercap, attackers can flood networks with gratuitous ARP packets targeting specific systems, making those systems associate the attacker's MAC with different IPs. For instance, an attacker might want to convince an HMI that the attacker is its PLC and convince the PLC that the attacker is its HMI, thus allowing the attacker to see and modify traffic between the two devices. There are two good ways to defend against this. For instance, you can use an IDS/IPS on your network to watch for this gratuitous ARP traffic, or you can install endpoint security software on your hosts to detect when that host is being targeted. This latter approach is more difficult for ICS networks because many of our devices cannot have endpoint security solutions installed on them. For the Linux and UNIX machine in your network, a great open source utility called arpwatch works well.

DNS Spoofing

Say you want to perform a DNS cache poisoning attack to trick the server into responding with a false IP address, or even more directly pretend to be another computer and spoof its IP address, perhaps to give an incorrect DNS answer. You would have to be in the communication path, so you could sniff the packet and keep it from reaching the real DNS server. Depending on the specifics of the attack, an IDS/IPS might recognize this attack, but your best defenses are hardening your DNS servers to limit who and what they trust and architecting your DNS infrastructure in a secure manner by using different DNS servers in each security zone.



IPV4 VERSUS IPV6

IPv4	IPv6
32-bit address Example: 10.42.84.5/24 4.2 billion addresses	128-bit address Example: fe80::13:ceff:fe45:4fe3 340 undecillion addresses (see below)
No authentication	Provides authentication of endpoints
Encryption provided by high layers	Support for encryption
Best effort transport	Quality-of-service (QOS) features provided in the protocol
Configuration handled by DHCP	Autoconfiguration

IPv6 already being used in some ICS networks

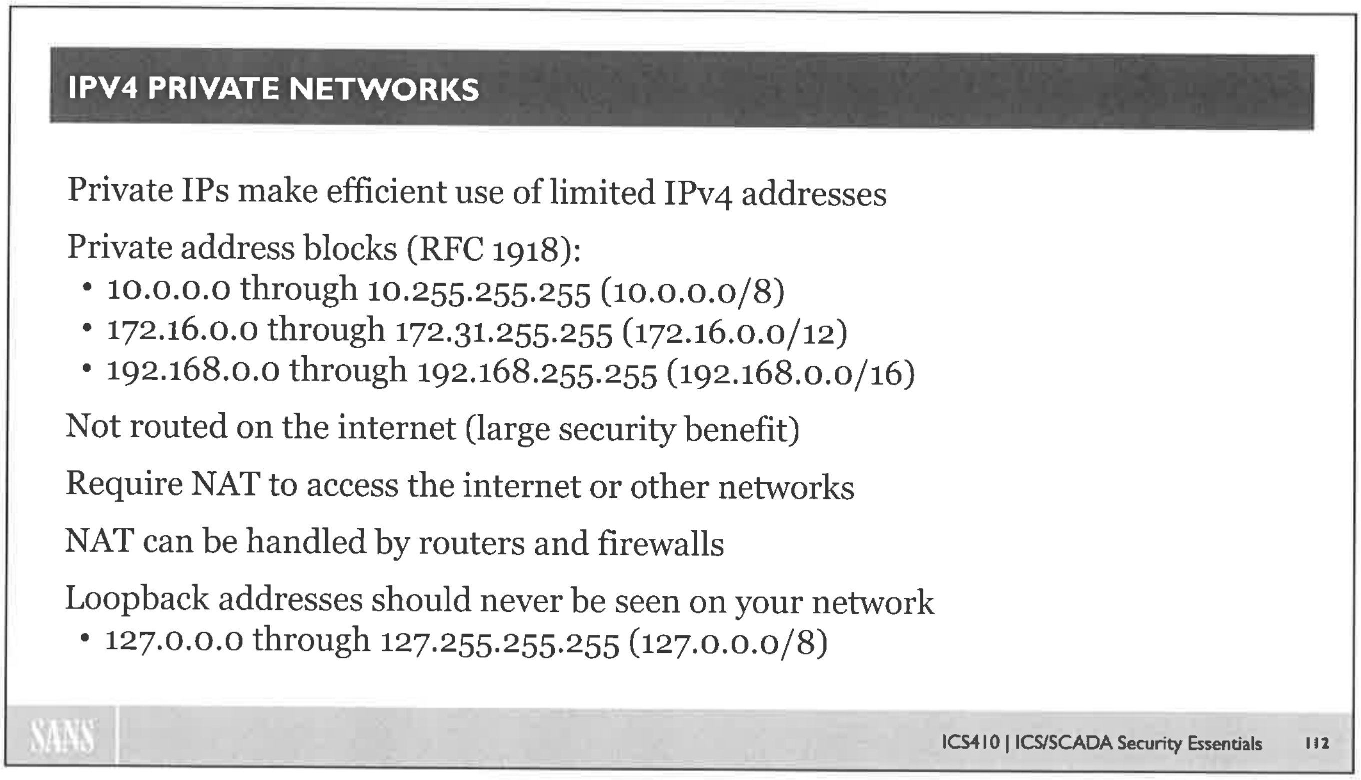
- Some wireless device networks
- Some smart meter networks

SANS ICS410 | ICS/SCADA Security Essentials 111

With IPv6, consideration was made for authentication, security of the communication, and quality-of-service features. The IPv6 protocol was designed to meet these growth demands, expanding the address size from 32 bits to 128 bits. A 128-bit address is approximately 340 undecillion addresses or 340,282,366,920,938,463,463,374,607,431,768,211,456. The sheer volume of available IP addresses provides for more flexible deployment of address space on the internet. For example, ISPs will be able to geographically assign IPv6 prefixes to different parts of the world, allowing for the simplified routing of traffic on the internet.

IPv6 incorporates aspects of IPSec to provide authentication of endpoints and encryption of the packets in transport. Also included in IPv6 are quality-of-service (QOS) features that will permit real-time sensitive applications such as VoIP and interactive media to take priority over less critical packet streams. A key feature of IPv6 is the expansion of address space, permitting route aggregation on core internet routers through geographic address space allocation, improving delegation and management of addresses to organizations and ISPs alike, as well as providing hierarchical distribution of address space that makes troubleshooting and internet routing simpler. Another valuable feature of IPv6 is support for addressing auto configuration. Anyone who has been responsible for manually assigning IP addresses to hosts understands that this is a problematic and cumbersome process. With 128 bits of address space, it becomes possible to use the globally unique MAC addresses on all network cards as IP addresses. In this way, administrators can simply introduce a new node to an IPv6 network without manually specifying an IP address; the IP address is configured automatically, based on the local MAC address and advertisement information from the default gateway on the network.

Another significant change in the IPv6 protocol is the use of a fixed IP header. Although the IPv4 header could expand to include additional information such as strict or loose source routing, the IPv6 protocol has a fixed header length of 40 bytes. To accommodate additional flexibility in the protocol, IPv6 introduces a next header field that indicates what the embedded protocol contained in the packet payload is. This is similar to IPv4's embedded protocol field, but unlike this field, the next protocol can include multiple embedded protocol fields, one right after another. Currently, supported IPv6 next header protocols include the encapsulating security protocol (ESP) and authentication header protocol (AH) for IPSec, the destination options header to specify processing options at the destination system, and upper-layer protocols such as UDP, TCP, and ICMP.



IPV4 PRIVATE NETWORKS

Private IPs make efficient use of limited IPv4 addresses

Private address blocks (RFC 1918):

- 10.0.0.0 through 10.255.255.255 (10.0.0.0/8)
- 172.16.0.0 through 172.31.255.255 (172.16.0.0/12)
- 192.168.0.0 through 192.168.255.255 (192.168.0.0/16)

Not routed on the internet (large security benefit)

Require NAT to access the internet or other networks

NAT can be handled by routers and firewalls

Loopback addresses should never be seen on your network

- 127.0.0.0 through 127.255.255.255 (127.0.0.0/8)

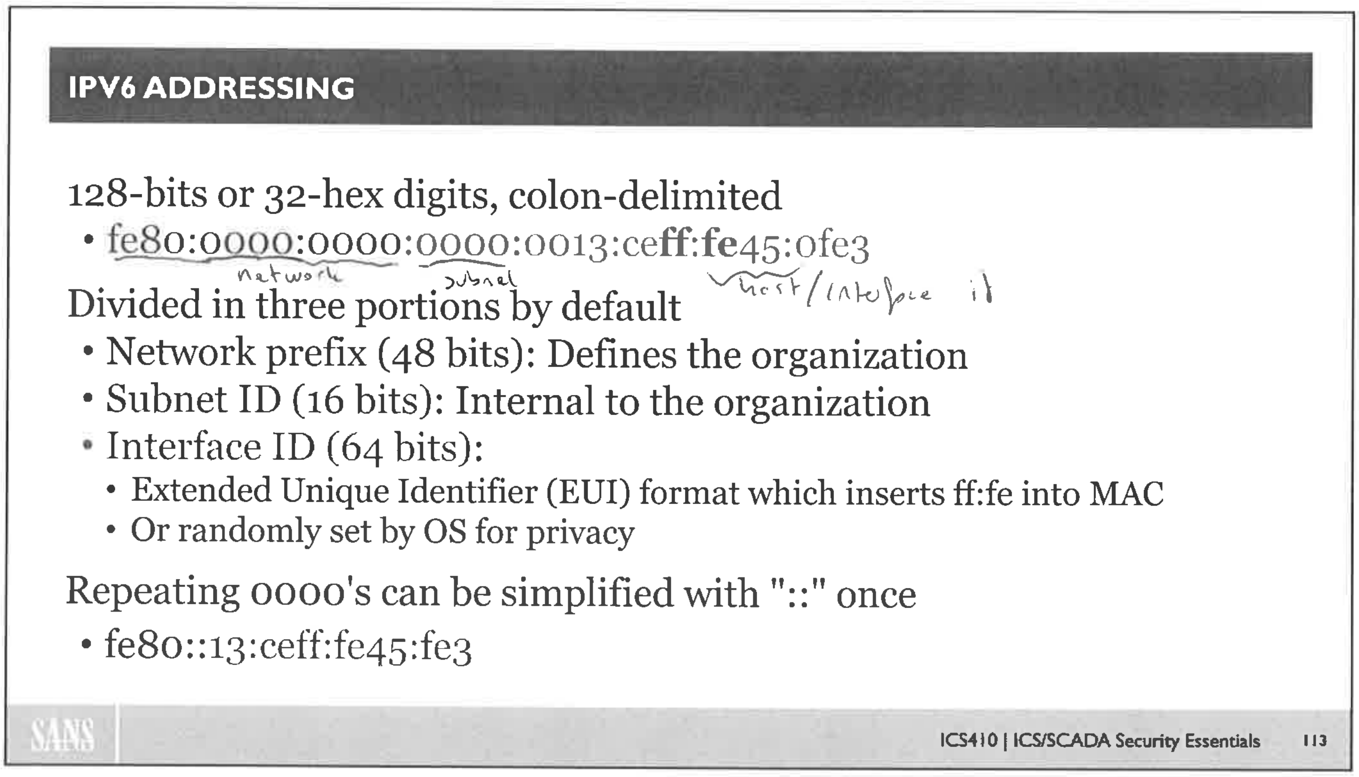
ICS410 | ICS/SCADA Security Essentials 112

Not every host capable of accessing the internet has a direct connection. These days, computers are (or should be!) behind firewalls of some sort. These devices can use Network Address Translation (NAT) so that the IP addresses in use on the internal LAN are automatically mapped to a different IP address or set of IP addresses when they traverse the firewall and go out to the internet. If no one on the internet can ever see these addresses, why should an organization bother to request an address block from its ISP? Even more to the point, why should the ISP waste addresses by allocating them to a customer when these addresses never will be routed over the internet?

It turns out that the answer to each of these questions is, "They do not have to." The Internet Assigned Numbers Authority (IANA), the ultimate authority for IP address assignments, has designated three sets of private address blocks that never can be routed over the internet and therefore are free for anyone to use as they wish within their own networks. Because these addresses cannot traverse the internet, it does not matter if 2, 5, or 10,000 different sites pick the same address to use on their internal networks. As long as the traffic is translated to routable IP addresses before it goes out onto the internet, the actual internal network numbers used do not matter a bit. To get traffic to these private IP addresses from public internet hosts, a router or firewall must perform a Network Address Translation function to associate internal and external IP address. This can be done using Dynamic NAT where internal IPs are automatically mapped to external IPs, or it can be done with static NAT where an administrator makes permanent the association between an internal and external IP.

In addition to these private network numbers, there is another special class of non-routable IP addresses: 127.0.0.0/8. Loopback addresses are never routed over any network, not even a local LAN segment. They are used only as pseudo IP addresses that always refer to the local host. In other words, every host responds to traffic addressed to 127.0.0.0/8, but only if the packets came from the same local host. Loopback addresses are often used by services that must contact other services running on the same machine. Note, however, any time you see a 127.0.0.0/8 packet on the network, either something is improperly configured, or someone is trying to attack you.

© 2019, Justin Searle



IPV6 ADDRESSING

128-bits or 32-hex digits, colon-delimited

- fe80:0000:0000:0000:0013:ceff:fe45:0fe3

network subnet host/interface id

Divided in three portions by default

- Network prefix (48 bits): Defines the organization
- Subnet ID (16 bits): Internal to the organization
- Interface ID (64 bits):
  - Extended Unique Identifier (EUI) format which inserts ff:fe into MAC
  - Or randomly set by OS for privacy

Repeating 0000's can be simplified with "::" once

- fe80::13:ceff:fe45:fe3

SANS ICS410 | ICS/SCADA Security Essentials 113

In IPv6, IP addresses are represented using hexadecimal notation, with values separated by colons instead of dots. For the foreseeable future, many IPv6 addresses will include strings of four repeating 0s ("0000"), which can be condensed to a double colon to represent one or more groups of four zeros.

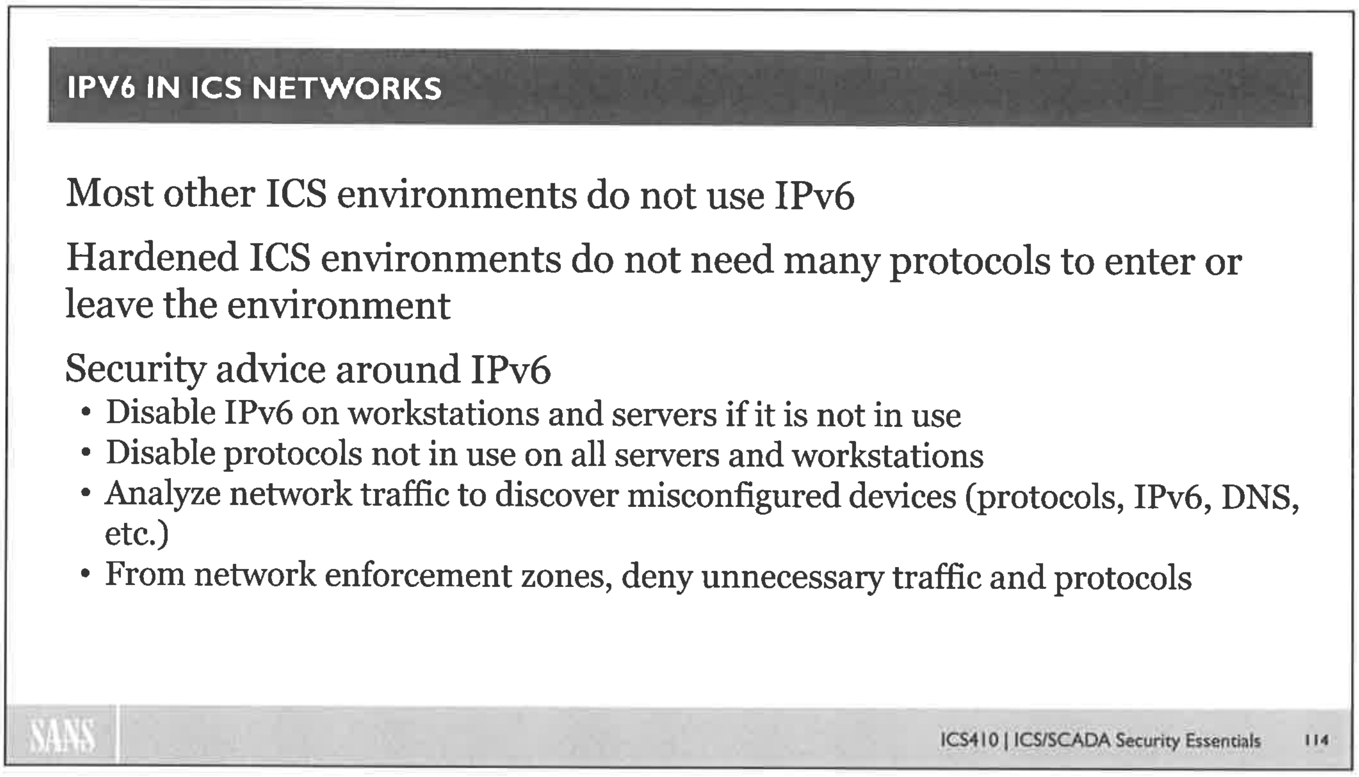
IPv6 addresses are broken up into three major sections:

Network Prefix: The network prefix is represented in the first 48 bits (6 bytes) of the IPv6 address. This is the address portion that is allocated to organizations that need to address IPv6 clients or to preserve other network functionality. Some fixed network prefix allocations include "fe80::" for local network use; "ff00::" for multicast traffic; "2001::" for large ISP interdomain routing; and "2002::" for IPv6-to-IPv4 gateway networks.

Subnet ID: The subnet ID is configured according to the addressing needs of the organization. For flat IPv6 networks, this value will usually be "0000" but can be any value selected by the organization that has been given the network prefix.

Interface Identification: The interface identification section uniquely identifies the IPv6 node. With IPv6 auto-configuration, the MAC address of the client populates the interface identification portion of the IPv6 address. Because a MAC address is a 48-bit value and the interface identification portion of the IPv6 address is 64 bits, the MAC addresses are expanded to fill the space by converting it to the Extended Unique Identifier (EUI) format specified by the IEEE. The EUI expansion takes the first three octets of the MAC address, appends the constant value "ff:fe", and then appends the last 3 bytes of the MAC address to form the interface identification portion of the IPv6 address.

For example, if a small, flat network uses private addressing, it would most likely use fe80:0000:0000 as the network prefix. If the network adapter of the host has a MAC address of 00:13:ce:45:4f:e3, by converting the MAC to EUI-64 notation, the interface identifier would become 0013:ceff:fe45:4fe3. Putting it all together, IPv6 auto-configuration would assign an IPv6 address of fe80:0000:0000:0013:ceff:fe45:4fe3. By convention in IPv6, groups of consecutive zeros can be replaced with "::", thus, permitting us to write the address as fe80::13:ceff:fe45:4fe3.



IPV6 IN ICS NETWORKS

Most other ICS environments do not use IPv6

Hardened ICS environments do not need many protocols to enter or leave the environment

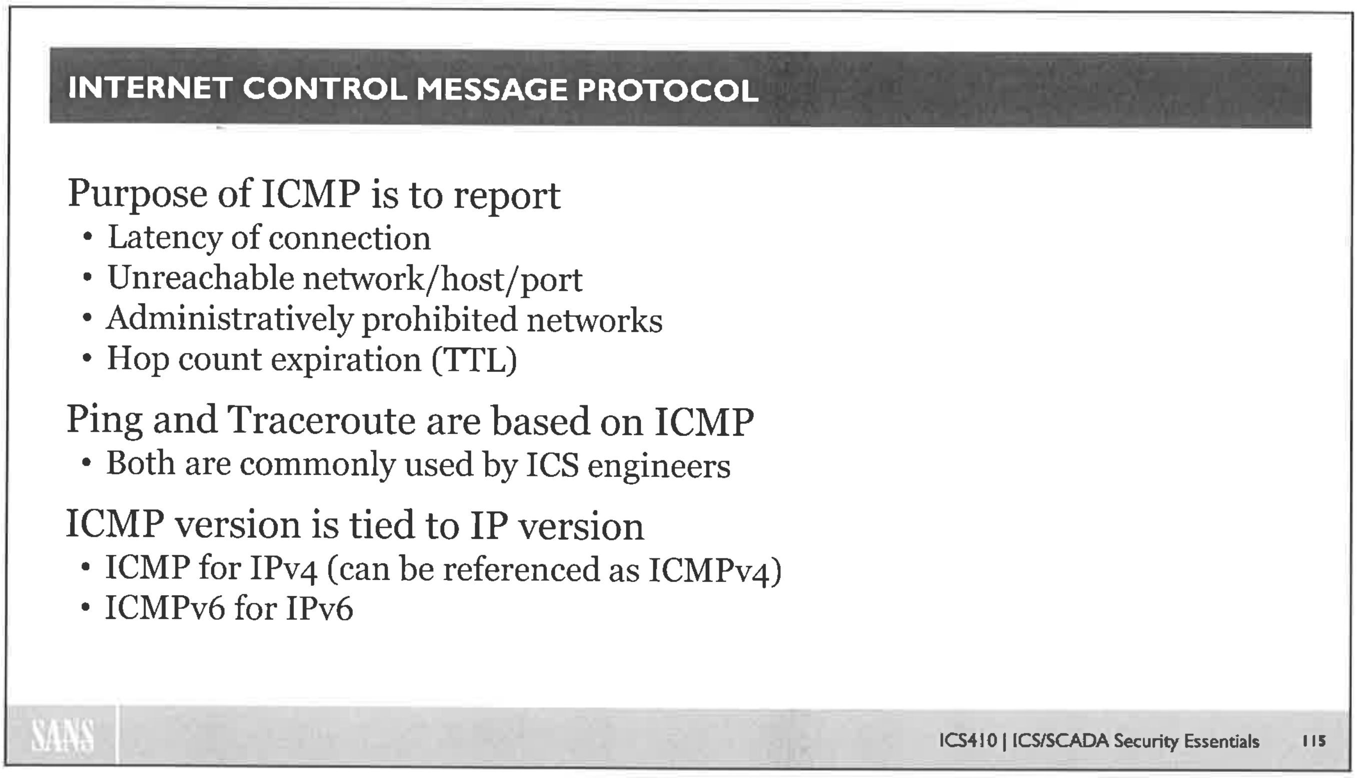
Security advice around IPv6

- Disable IPv6 on workstations and servers if it is not in use
- Disable protocols not in use on all servers and workstations
- Analyze network traffic to discover misconfigured devices (protocols, IPv6, DNS, etc.)
- From network enforcement zones, deny unnecessary traffic and protocols

SANS ICS410 | ICS/SCADA Security Essentials 114

Most ICS environments do not make use of IPv6 and most hardened ICS environments do not need many protocols to enter or leave the environment. Disable IPv6 on workstations and servers if it is not in use. Disable protocols not in use on all servers and workstations. Analyze network traffic to discover misconfigured devices (protocols, IPv6, DNS, and so on). From network enforcement zones, deny unnecessary traffic and protocols.

© 2019, Justin Searle



INTERNET CONTROL MESSAGE PROTOCOL

Purpose of ICMP is to report
- Latency of connection
- Unreachable network/host/port
- Administratively prohibited networks
- Hop count expiration (TTL)

Ping and Traceroute are based on ICMP
- Both are commonly used by ICS engineers

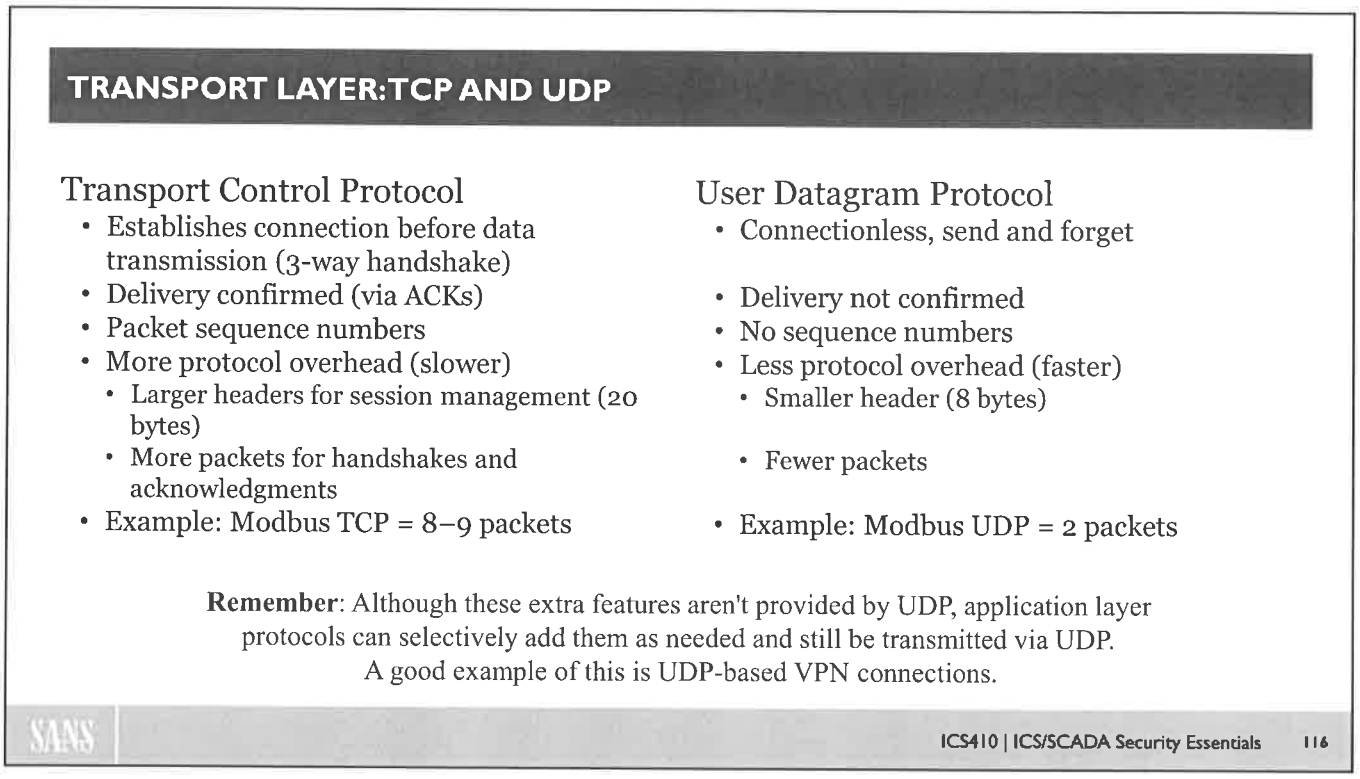
ICMP version is tied to IP version
- ICMP for IPv4 (can be referenced as ICMPv4)
- ICMPv6 for IPv6

SANS ICS410 | ICS/SCADA Security Essentials 115

The third (and final) protocol we discuss in this module is the Internet Control Message Protocol (ICMP). We mentioned this briefly in the last module, but this is a good time to cover it in more detail. ICMP is a network layer protocol, unlike TCP and UDP, which is part of the transport layer. As such, ICMP actually is a peer of IP, even though it is still encapsulated in an IP packet. In fact, IP, TCP, and UDP all rely on ICMP to provide information about network conditions, as well as for status and error messages pertaining to their transmissions.

ICMP is a simple protocol. It is datagram-based, like IP and UDP. Most ICMP transactions require only one or two packets. ICMP packets have only three header fields, fewer fields even than UDP, and one of them is just a checksum!

ICMP is an important protocol because it carries critical and fundamental information about the state of a network and error conditions that occur. It is not meant as a protocol to be used for the transmission of data; rather it is designed for error reporting and network-based troubleshooting techniques. Many of the other protocols rely upon ICMP to perform functions and communicate error conditions.



TRANSPORT LAYER: TCP AND UDP

Transport Control Protocol

- Establishes connection before data transmission (3-way handshake)
- Delivery confirmed (via ACKs)
- Packet sequence numbers
- More protocol overhead (slower)
  - Larger headers for session management (20 bytes)
  - More packets for handshakes and acknowledgments
- Example: Modbus TCP = 8–9 packets

User Datagram Protocol

- Connectionless, send and forget
- Delivery not confirmed
- No sequence numbers
- Less protocol overhead (faster)
  - Smaller header (8 bytes)
  - Fewer packets
- Example: Modbus UDP = 2 packets

Remember: Although these extra features aren't provided by UDP, application layer protocols can selectively add them as needed and still be transmitted via UDP. A good example of this is UDP-based VPN connections.

ICS410 | ICS/SCADA Security Essentials 116

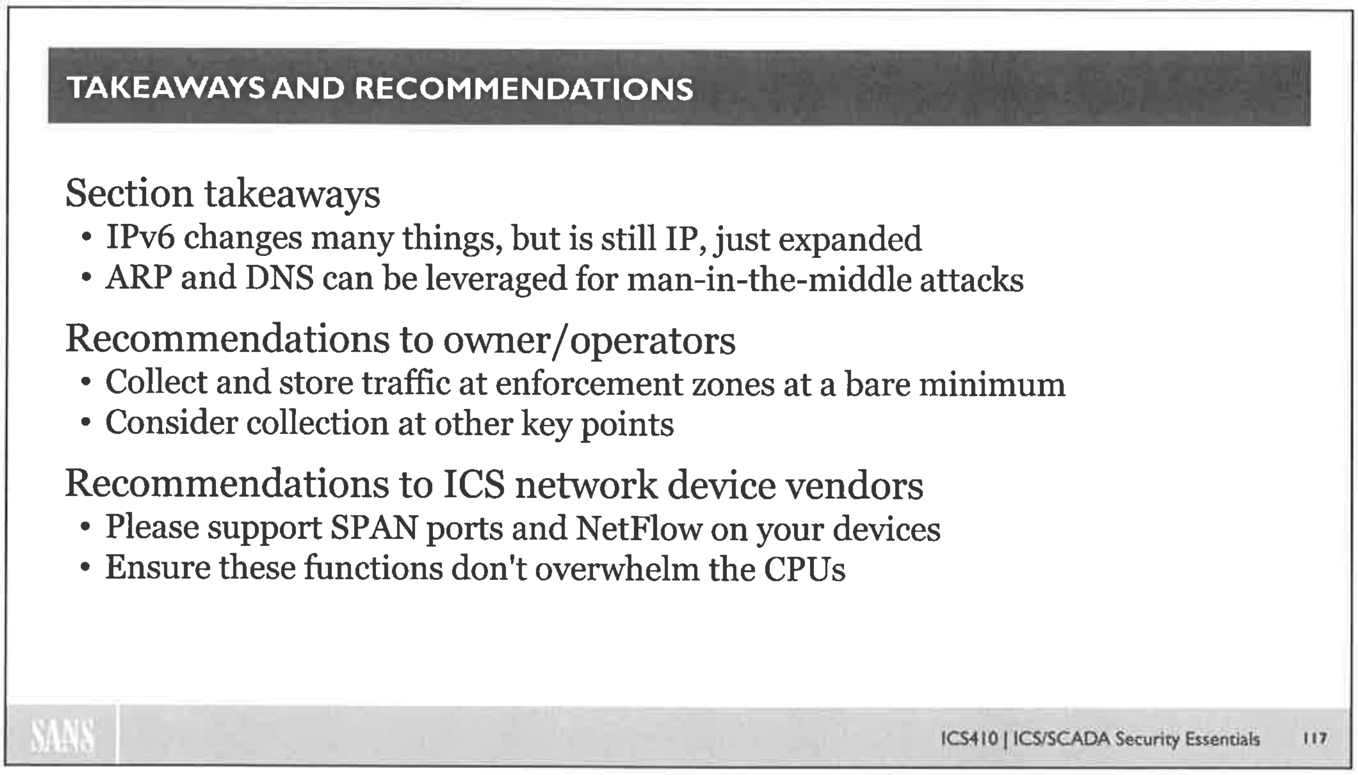
We hope by now you have a much better idea of how much work TCP puts into making sure your connections are efficient and your delivery is confirmed. Having confirmed or guaranteed delivery means a trade-off between raw speed and reliable communications, but for most applications, especially those designed for the internet, it is probably well worth it. Confirmed delivery is easier to program for, because TCP takes care of a lot of the details for you. UDP, however, is good for real-time communication, where guaranteed delivery offers little benefit, and it provides less overhead.

For TCP, the client initiates the 3-way handshake by sending a SYN to signal a request for a connection to the server. Then, if the server is up and offering the desired service, it can accept the incoming connection and respond to the SYN. The response consists of both an acknowledgment of the client's initial connection request (the ACK flag is set) and a connection request of its own (the SYN flag is set), together in a single packet (a SYN-ACK). Finally, after the client receives the SYN-ACK, it sends a final ACK to the server. After the server receives the ACK, the 3-way handshake is complete, and the connection has been established. The two servers can then exchange data.

When the time comes to close a TCP connection, each end of the connection must be closed separately. Assuming that the PC wants to close the connection first, the process starts when the PC sends a FIN packet to the server. The FIN portion indicates to the server that the PC wants to close the connection (continuing with the sequence count it has been using with the server). The server responds by sending an ACK to the PC that is acknowledging the FIN that the PC sent. Next, the server sends a FIN packet to the PC to close its side of the connection. Depending on the operating system, this is often done in a single packet. Finally, the PC sends an ACK to the server to acknowledge the FIN. When abruptly closing a connection between two machines, either side can send a single RST to do so. Finally, the numbers in parentheses are the sequence numbers that are sent along with each packet.

The benefit of using TCP is that the application layer doesn't have to worry about delivery confirmation. It is incorporated into the protocol. However, we should remember that while delivery confirmation isn't provided natively by UDP, this functionality CAN still be provided by the application protocol and still be transmitted via UDP. A good example of this is UDP-based VPN connections. If a UDP frame is lost, then the VPN application layer can detect this and request that the frame be re-sent.

© 2019, Justin Searle



TAKEAWAYS AND RECOMMENDATIONS

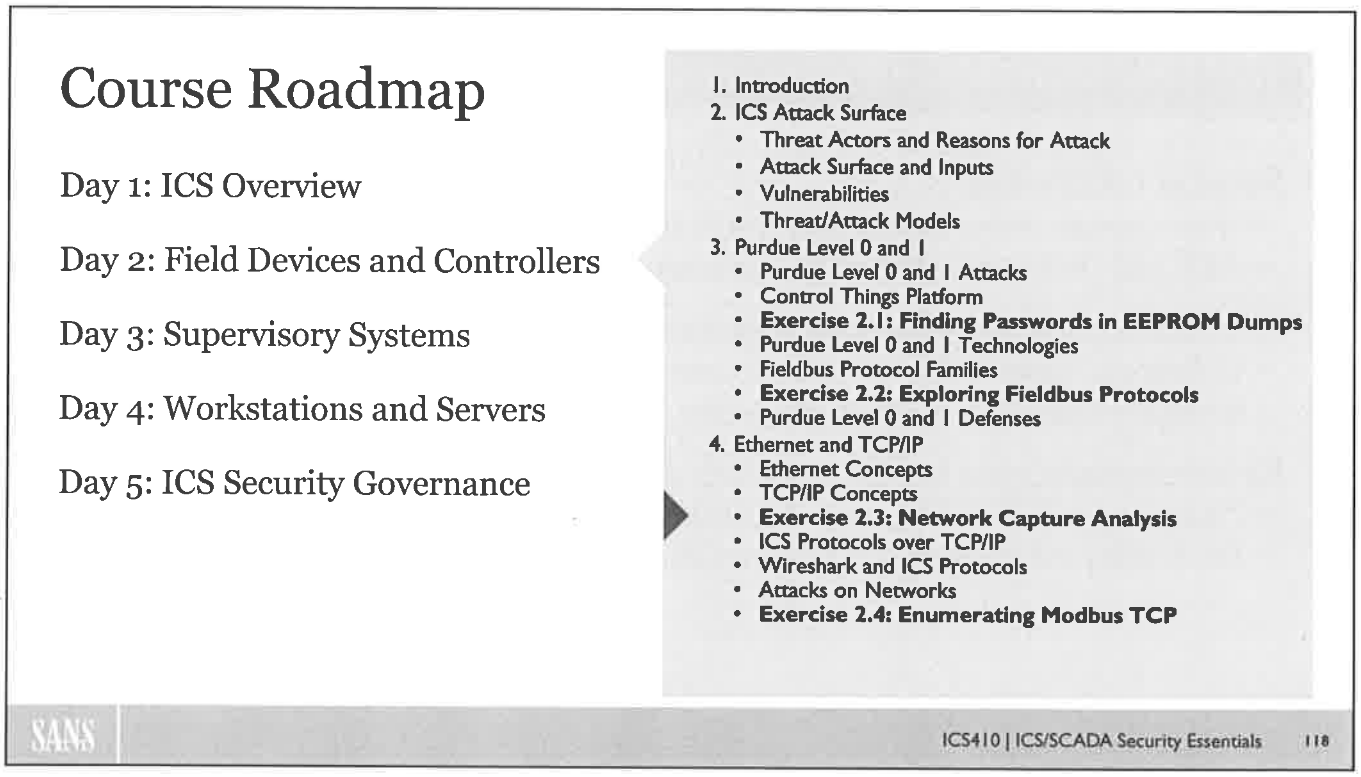
Section takeaways
- IPv6 changes many things, but is still IP, just expanded
- ARP and DNS can be leveraged for man-in-the-middle attacks

Recommendations to owner/operators
- Collect and store traffic at enforcement zones at a bare minimum
- Consider collection at other key points

Recommendations to ICS network device vendors
- Please support SPAN ports and NetFlow on your devices
- Ensure these functions don't overwhelm the CPUs

SANS ICS410 | ICS/SCADA Security Essentials 117

This page intentionally left blank.



Course Roadmap

Day 1: ICS Overview

Day 2: Field Devices and Controllers

Day 3: Supervisory Systems

Day 4: Workstations and Servers

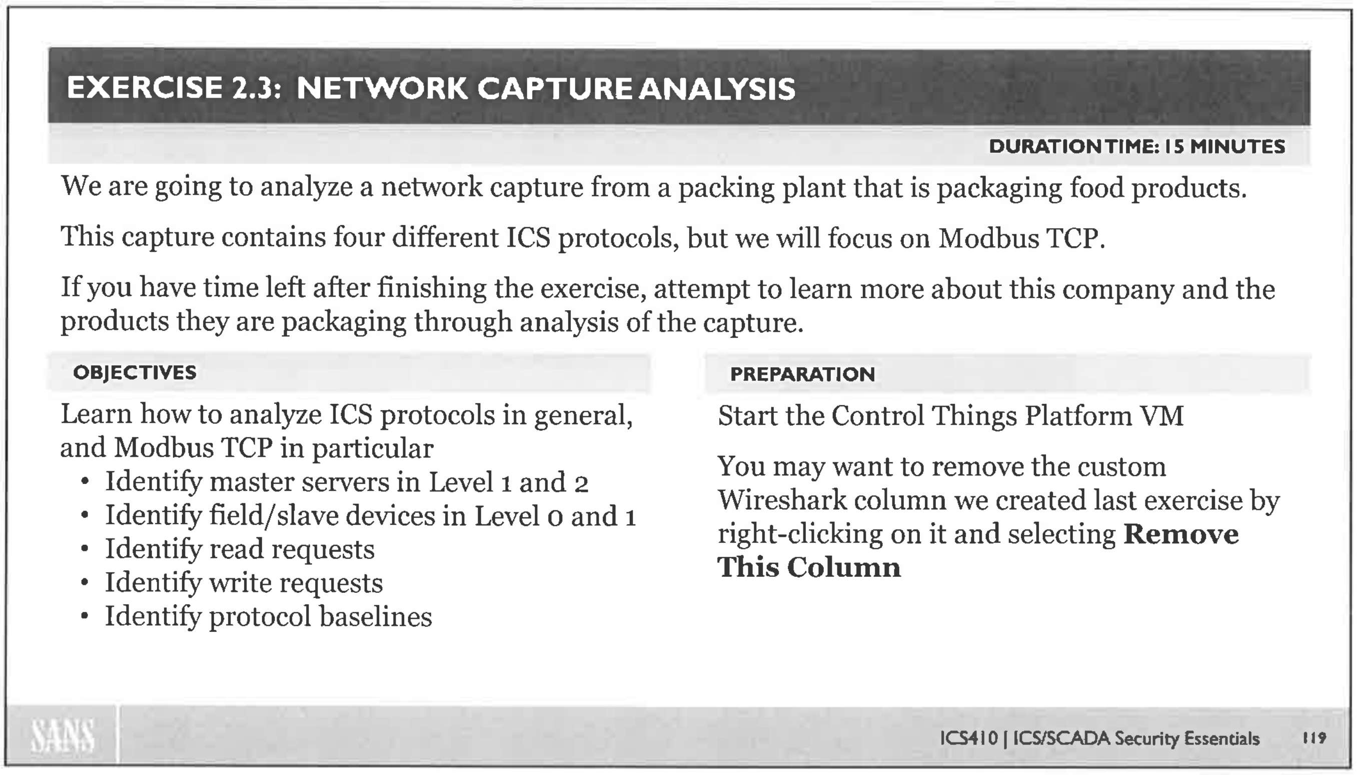
Day 5: ICS Security Governance

SANS

ICS410 | ICS/SCADA Security Essentials 118

This page intentionally left blank.

© 2019, Justin Searle



EXERCISE 2.3: NETWORK CAPTURE ANALYSIS

DURATION TIME: 15 MINUTES

We are going to analyze a network capture from a packing plant that is packaging food products.

This capture contains four different ICS protocols, but we will focus on Modbus TCP.

If you have time left after finishing the exercise, attempt to learn more about this company and the products they are packaging through analysis of the capture.

OBJECTIVES

Learn how to analyze ICS protocols in general, and Modbus TCP in particular

- Identify master servers in Level 1 and 2
- Identify field/slave devices in Level 0 and 1
- Identify read requests
- Identify write requests
- Identify protocol baselines

PREPARATION

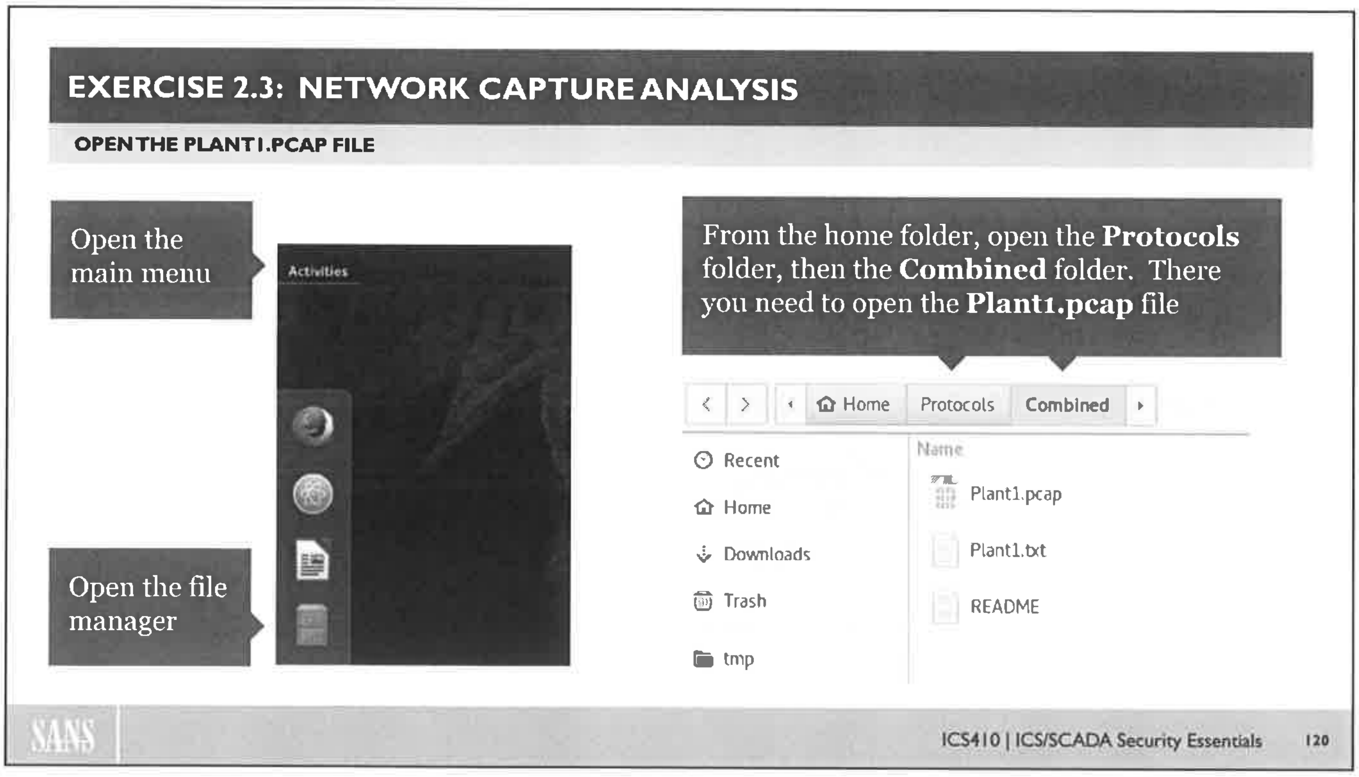
Start the Control Things Platform VM

You may want to remove the custom Wireshark column we created last exercise by right-clicking on it and selecting **Remove This Column**

ICS410 | ICS/SCADA Security Essentials 119

For this exercise, we will be looking at a network capture file of an ICS network. This capture file contains some Modbus traffic, but it also contains other protocols as well. The goal of this exercise is to learn how to use Wireshark's features to extract details about specific traffic patterns for specific protocols. To do this, we'll be trying to answer the following questions about the Modbus TCP traffic in this capture:

Which IP address is the master on?
How many field devices is the master talking to?
Is the master writing any data to the devices?
Is the small traffic spike in the middle of the network capture related to Modbus?



EXERCISE 2.3: NETWORK CAPTURE ANALYSIS

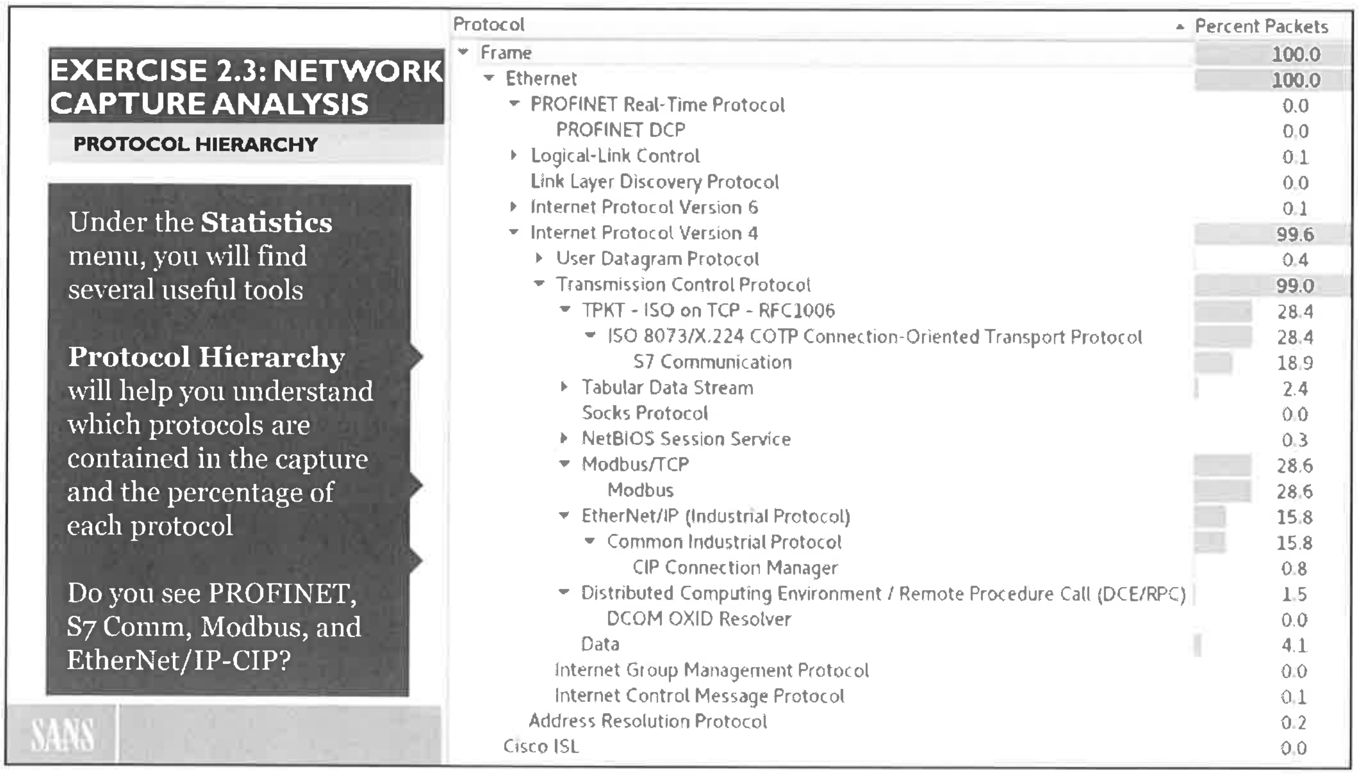
OPEN THE PLANT1.PCAP FILE

Open the main menu

Open the file manager

From the home folder, open the **Protocols** folder, then the **Combined** folder. There you need to open the **Plant1.pcap** file

Open the main menu by clicking on the **Activities** link in the upper left-hand corner. On the left side, you will see an icon of a blue filing cabinet. That is the file manager. Click on it to open it. This places you into your home directory. In your home directory, you will see various folders. Open the **Protocols** folder. There you will find several folders named after the protocols they contain. Open the folder named **Combined**. In this folder, you will find a file called **Plant1.pcap**, which you will need to open. This should automatically open it in Wireshark.



EXERCISE 2.3: NETWORK CAPTURE ANALYSIS

PROTOCOL HIERARCHY

Under the **Statistics** menu, you will find several useful tools

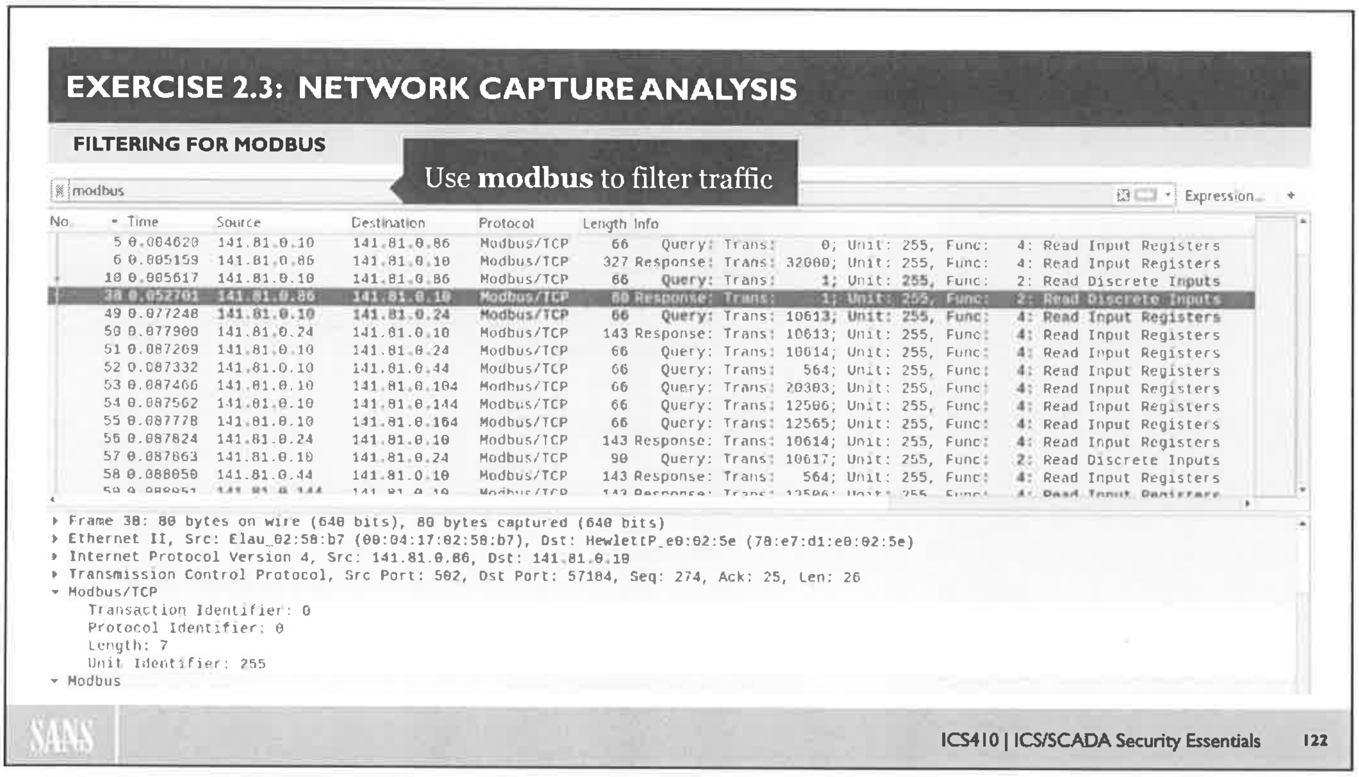
Protocol Hierarchy will help you understand which protocols are contained in the capture and the percentage of each protocol

Do you see PROFINET, S7 Comm, Modbus, and EtherNet/IP-CIP?

```
Protocol                                                                          Percent Packets
▾ Frame                                                                                   100.0
  ▾ Ethernet                                                                              100.0
    ▾ PROFINET Real-Time Protocol                                                           0.0
        PROFINET DCP                                                                        0.0
    ▸ Logical-Link Control                                                                  0.1
      Link Layer Discovery Protocol                                                         0.0
    ▸ Internet Protocol Version 6                                                           0.1
    ▾ Internet Protocol Version 4                                                          99.6
      ▸ User Datagram Protocol                                                              0.4
      ▾ Transmission Control Protocol                                                      99.0
        ▾ TPKT - ISO on TCP - RFC1006                                                      28.4
          ▾ ISO 8073/X.224 COTP Connection-Oriented Transport Protocol                     28.4
              S7 Communication                                                             18.9
        ▸ Tabular Data Stream                                                               2.4
          Socks Protocol                                                                    0.0
        ▸ NetBIOS Session Service                                                           0.3
        ▾ Modbus/TCP                                                                       28.6
            Modbus                                                                         28.6
        ▾ EtherNet/IP (Industrial Protocol)                                                15.8
          ▾ Common Industrial Protocol                                                     15.8
              CIP Connection Manager                                                        0.8
        ▾ Distributed Computing Environment / Remote Procedure Call (DCE/RPC)               1.5
            DCOM OXID Resolver                                                              0.0
          Data                                                                              4.1
        Internet Group Management Protocol                                                  0.0
        Internet Control Message Protocol                                                   0.1
      Address Resolution Protocol                                                           0.2
    Cisco ISL                                                                               0.0
```

Under the **Statistics** menu, you will find several useful tools, some of which we'll explore in this exercise. The first one we'll look at, **Protocol Hierarchy**, will help you understand which protocols are contained in the capture and the percentage of each protocol. You can see these statistics on the right, broken into many different measurements. As for the protocols, they are organized in a collapsible hierarchy that makes it easy to hide/show what you are most interested in.

Do you see PROFINET, S7 Comm, Modbus, and EtherNet/IP-CIP? Over 50% of this capture consists of control traffic, with Modbus, S7 Comm (this is Siemens' proprietary protocol for their S7 family of controllers), and EtherNet/IP-CIP. We are going to focus on the Modbus traffic, but if you have time, feel free to dig into these other protocols.



EXERCISE 2.3: NETWORK CAPTURE ANALYSIS

FILTERING FOR MODBUS

Use **modbus** to filter traffic

```
No.   Time      Source        Destination    Protocol    Length Info
    5 0.004620  141.81.0.10   141.81.0.86    Modbus/TCP    66     Query: Trans:     0; Unit: 255, Func:   4: Read Input Registers
    6 0.005159  141.81.0.86   141.81.0.10    Modbus/TCP   327 Response: Trans: 32000; Unit: 255, Func:   4: Read Input Registers
   10 0.005617  141.81.0.10   141.81.0.86    Modbus/TCP    66     Query: Trans:     1; Unit: 255, Func:   2: Read Discrete Inputs
   38 0.052701  141.81.0.86   141.81.0.10    Modbus/TCP    80 Response: Trans:     1; Unit: 255, Func:   2: Read Discrete Inputs
   49 0.077248  141.81.0.10   141.81.0.24    Modbus/TCP    66     Query: Trans: 10613; Unit: 255, Func:   4: Read Input Registers
   50 0.077900  141.81.0.24   141.81.0.10    Modbus/TCP   143 Response: Trans: 10613; Unit: 255, Func:   4: Read Input Registers
   51 0.087269  141.81.0.10   141.81.0.24    Modbus/TCP    66     Query: Trans: 10614; Unit: 255, Func:   4: Read Input Registers
   52 0.087332  141.81.0.10   141.81.0.44    Modbus/TCP    66     Query: Trans:   564; Unit: 255, Func:   4: Read Input Registers
   53 0.087466  141.81.0.10   141.81.0.104   Modbus/TCP    66     Query: Trans: 20303; Unit: 255, Func:   4: Read Input Registers
   54 0.087562  141.81.0.10   141.81.0.144   Modbus/TCP    66     Query: Trans: 12506; Unit: 255, Func:   4: Read Input Registers
   55 0.087778  141.81.0.10   141.81.0.164   Modbus/TCP    66     Query: Trans: 12565; Unit: 255, Func:   4: Read Input Registers
   56 0.087824  141.81.0.24   141.81.0.10    Modbus/TCP   143 Response: Trans: 10614; Unit: 255, Func:   4: Read Input Registers
   57 0.087863  141.81.0.10   141.81.0.24    Modbus/TCP    90     Query: Trans: 10617; Unit: 255, Func:   2: Read Discrete Inputs
   58 0.088050  141.81.0.44   141.81.0.10    Modbus/TCP   143 Response: Trans:   564; Unit: 255, Func:   4: Read Input Registers
   59 0.088051  141.81.0.144  141.81.0.10    Modbus/TCP   143 Response: Trans: 12506; Unit: 255, Func:   4: Read Input Registers
```

```
▸ Frame 38: 80 bytes on wire (640 bits), 80 bytes captured (640 bits)
▸ Ethernet II, Src: Elau_02:58:b7 (00:04:17:02:58:b7), Dst: HewlettP_e0:02:5e (78:e7:d1:e0:02:5e)
▸ Internet Protocol Version 4, Src: 141.81.0.86, Dst: 141.81.0.10
▸ Transmission Control Protocol, Src Port: 502, Dst Port: 57184, Seq: 274, Ack: 25, Len: 26
▾ Modbus/TCP
    Transaction Identifier: 0
    Protocol Identifier: 0
    Length: 7
    Unit Identifier: 255
▾ Modbus
```

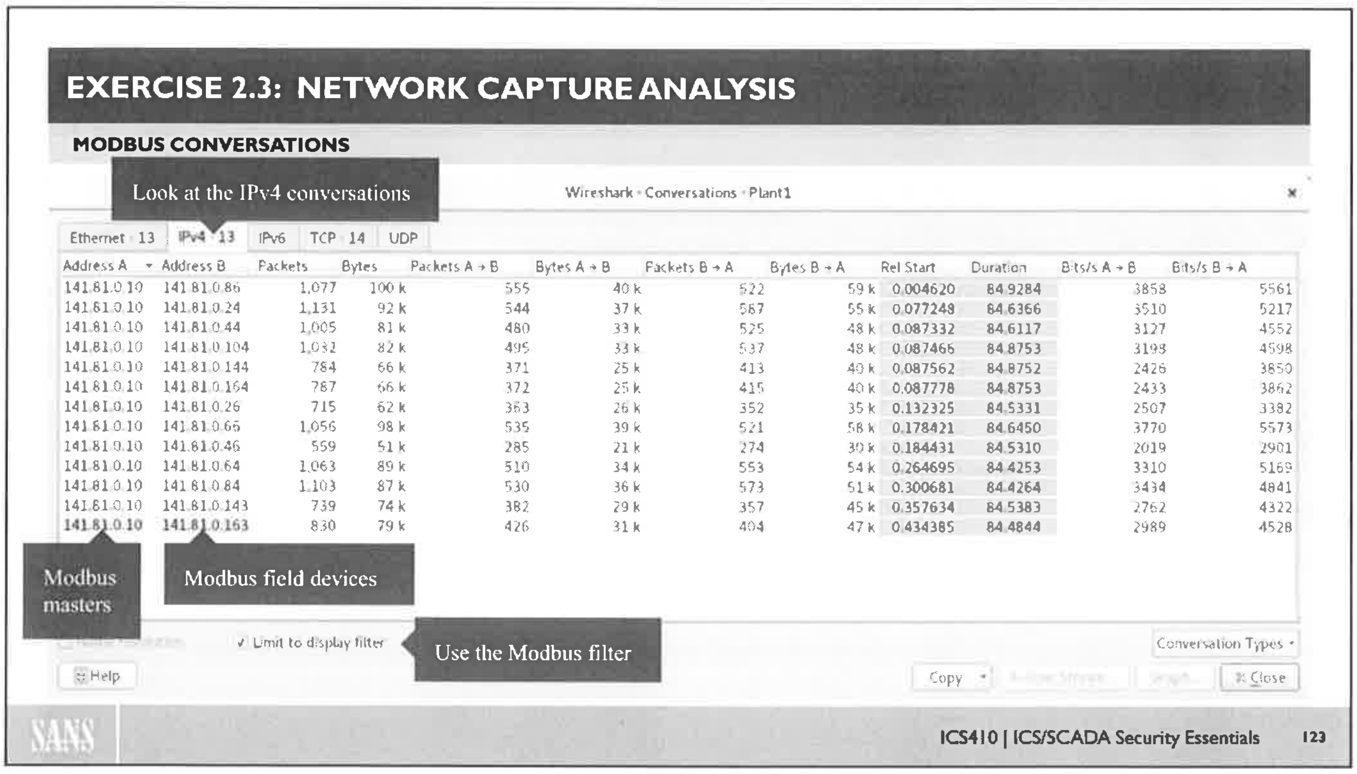
After Wireshark opens the pcap file, you should see a list of network packets. If you have never used Wireshark before, you will see a menu and tool icons at the top of the window. Below those icons is the filter bar. When you get to know Wireshark's filter language, you can easily type your own filters to have Wireshark show you subsets of the packet captures. If you are not familiar with the filter language, or not quite sure which filter options are available for a specific network protocol, you can click the expression button to open up a new window, which will help you find and build the filters you can use. Below the filter bar is the most important part of Wireshark, the table where they show you the details of all the network packets you have captured. You can click any of these to select a packet, which changes the two bottom portions of the windows to show you details for that packet. The first detail window will show you a human-readable breakdown of each of the protocol headers and their components. Below that is the raw packet shown in both hexadecimal and ASCII.

Because we are focusing on the Modbus TCP protocol in this exercise, the first thing we want to do is to build a filter to show only the Modbus TCP packet captures. You can do this by entering **modbus** into the filter input box.

This filter tells Wireshark that we want to see only TCP packets that are either going to TCP port 502 or coming from TCP port 502. This is the port that Modbus TCP uses for its communications. Modbus field devices listen on this port for commands from the Modbus master. Masters will send commands such as reads and writes from their IP address on a randomly chosen TCP port to the Modbus field device's IP address on port 502. When the field device receives this command, the field device replies with a packet from their IP address on TCP port 502 to the master's IP address and the same randomly chosen TCP port used in the original request.

Look through the packets to see this bidirectional communication. For instance, packet 5 is the request from the Modbus master who is on IP 141.81.0.10 using TCP port 57184 and is talking to the Modbus field device at 141.81.0.86 who is listening on his standard Modbus port of TCP 502. Packet 6 is the response from the field device sending the information back to the same IP and port it received the request from.

© 2019, Justin Searle



EXERCISE 2.3: NETWORK CAPTURE ANALYSIS

MODBUS CONVERSATIONS

Wireshark · Conversations · Plant1

Ethernet · 13 | IPv4 · 13 | IPv6 | TCP · 14 | UDP

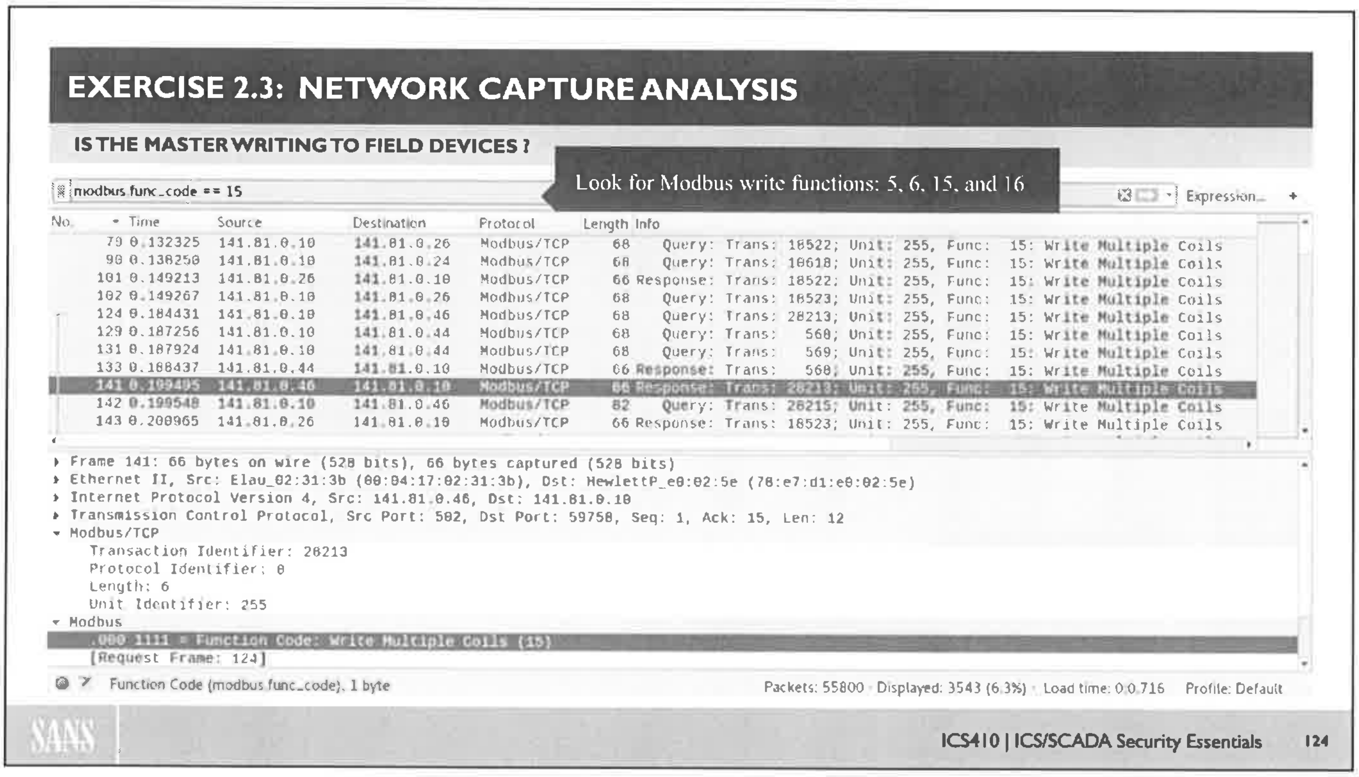
Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
141.81.0.10	141.81.0.86	1,077	100 k	555	40 k	522	59 k	0.004620	84.9284	3858	5561
141.81.0.10	141.81.0.24	1,131	92 k	544	37 k	567	55 k	0.077248	84.6366	3510	5217
141.81.0.10	141.81.0.44	1,005	81 k	480	33 k	525	48 k	0.087332	84.6117	3127	4552
141.81.0.10	141.81.0.104	1,032	82 k	495	33 k	537	48 k	0.087466	84.8753	3193	4598
141.81.0.10	141.81.0.144	784	66 k	371	25 k	413	40 k	0.087562	84.8752	2426	3850
141.81.0.10	141.81.0.164	787	66 k	372	25 k	415	40 k	0.087778	84.8753	2433	3862
141.81.0.10	141.81.0.26	715	62 k	363	26 k	352	35 k	0.132325	84.5331	2507	3382
141.81.0.10	141.81.0.65	1,056	98 k	535	39 k	521	58 k	0.178421	84.6450	3770	5573
141.81.0.10	141.81.0.46	559	51 k	285	21 k	274	30 k	0.184431	84.5310	2019	2901
141.81.0.10	141.81.0.64	1,063	89 k	510	34 k	553	54 k	0.264695	84.4253	3310	5169
141.81.0.10	141.81.0.84	1,103	87 k	530	36 k	573	51 k	0.300681	84.4264	3434	4841
141.81.0.10	141.81.0.143	739	74 k	382	29 k	357	45 k	0.357634	84.5383	2762	4322
141.81.0.10	141.81.0.163	830	79 k	426	31 k	404	47 k	0.434385	84.4844	2989	4528

SANS ICS410 | ICS/SCADA Security Essentials 123

Now to answer the question: "How many Modbus masters and Modbus field devices are there?" You can scroll through all the packet and count by hand, but this will take a lot of time and is prone to mistakes due to the large size of this capture. The easiest way to do this is to use Wireshark's **Conversations** tool, which you can find under the **Statistics** menu. Open the statistics window and go the **IPv4** tab. This shows all the IP conversations in this capture. Notice that by default the conversations window doesn't use the filter we have in the main window. This is easy enough to fix. Click the **Limit to display filter** at the bottom of the window so we are looking only at Modbus TCP traffic.

This view makes it easy to see how many masters and field devices we have in this network. There is one line per conversation between master and field device. The **Address A** device is the one that initialized the conversation and the **Address B** device is the one that is making the replies. Here we can see that in every instance, IP address **141.81.0.10** is initializing the conversation. This must be the Modbus master because the Modbus TCP protocol doesn't have an instance where the field device can initiate its own conversations. Since we don't see any other IP addresses in the **Address A** column, there must be only one Modbus master in this network. As for field devices, you can see there are 13 different conversations, so there must be 13 different field devices that the Modbus master is talking to.

When doing incident response or security assessments, this window will greatly help you build a baseline for normal traffic in your network and help you determine at any point if there is something unusual going on with the Modbus traffic.



EXERCISE 2.3: NETWORK CAPTURE ANALYSIS

IS THE MASTER WRITING TO FIELD DEVICES ?

Look for Modbus write functions: 5, 6, 15, and 16

```
modbus.func_code == 15

No.    Time      Source       Destination  Protocol   Length Info
  79 0.132325  141.81.0.10  141.81.0.26  Modbus/TCP   68    Query: Trans: 18522; Unit: 255, Func:  15: Write Multiple Coils
  98 0.138250  141.81.0.10  141.81.0.24  Modbus/TCP   68    Query: Trans: 18618; Unit: 255, Func:  15: Write Multiple Coils
 101 0.149213  141.81.0.26  141.81.0.10  Modbus/TCP   66 Response: Trans: 18522; Unit: 255, Func:  15: Write Multiple Coils
 102 0.149267  141.81.0.10  141.81.0.26  Modbus/TCP   68    Query: Trans: 18523; Unit: 255, Func:  15: Write Multiple Coils
 124 0.184431  141.81.0.10  141.81.0.46  Modbus/TCP   68    Query: Trans: 28213; Unit: 255, Func:  15: Write Multiple Coils
 129 0.187256  141.81.0.10  141.81.0.44  Modbus/TCP   68    Query: Trans:   568; Unit: 255, Func:  15: Write Multiple Coils
 131 0.187924  141.81.0.10  141.81.0.44  Modbus/TCP   68    Query: Trans:   569; Unit: 255, Func:  15: Write Multiple Coils
 133 0.188437  141.81.0.44  141.81.0.10  Modbus/TCP   66 Response: Trans:   568; Unit: 255, Func:  15: Write Multiple Coils
 141 0.199495  141.81.0.46  141.81.0.10  Modbus/TCP   66 Response: Trans: 28213; Unit: 255, Func:  15: Write Multiple Coils
 142 0.199548  141.81.0.10  141.81.0.46  Modbus/TCP   82    Query: Trans: 28215; Unit: 255, Func:  15: Write Multiple Coils
 143 0.200965  141.81.0.26  141.81.0.10  Modbus/TCP   66 Response: Trans: 18523; Unit: 255, Func:  15: Write Multiple Coils

▸ Frame 141: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
▸ Ethernet II, Src: Elau_02:31:3b (00:04:17:02:31:3b), Dst: HewlettP_e0:02:5e (78:e7:d1:e0:02:5e)
▸ Internet Protocol Version 4, Src: 141.81.0.46, Dst: 141.81.0.10
▸ Transmission Control Protocol, Src Port: 502, Dst Port: 59758, Seq: 1, Ack: 15, Len: 12
▾ Modbus/TCP
    Transaction Identifier: 28213
    Protocol Identifier: 0
    Length: 6
    Unit Identifier: 255
▾ Modbus
    .000 1111 = Function Code: Write Multiple Coils (15)
    [Request Frame: 124]

Function Code (modbus.func_code), 1 byte        Packets: 55800 · Displayed: 3543 (6.3%) · Load time: 0:0.716    Profile: Default
```

If you want to see if 141.81.0.10 is sending any command to modify or control the process on the field devices, you can search for write commands. Modbus has four data write commands.

```
5 - Write Single Coil
6 - Write Single Register
15 - Write Multiple Coils
16 - Write Multiple Registers
```

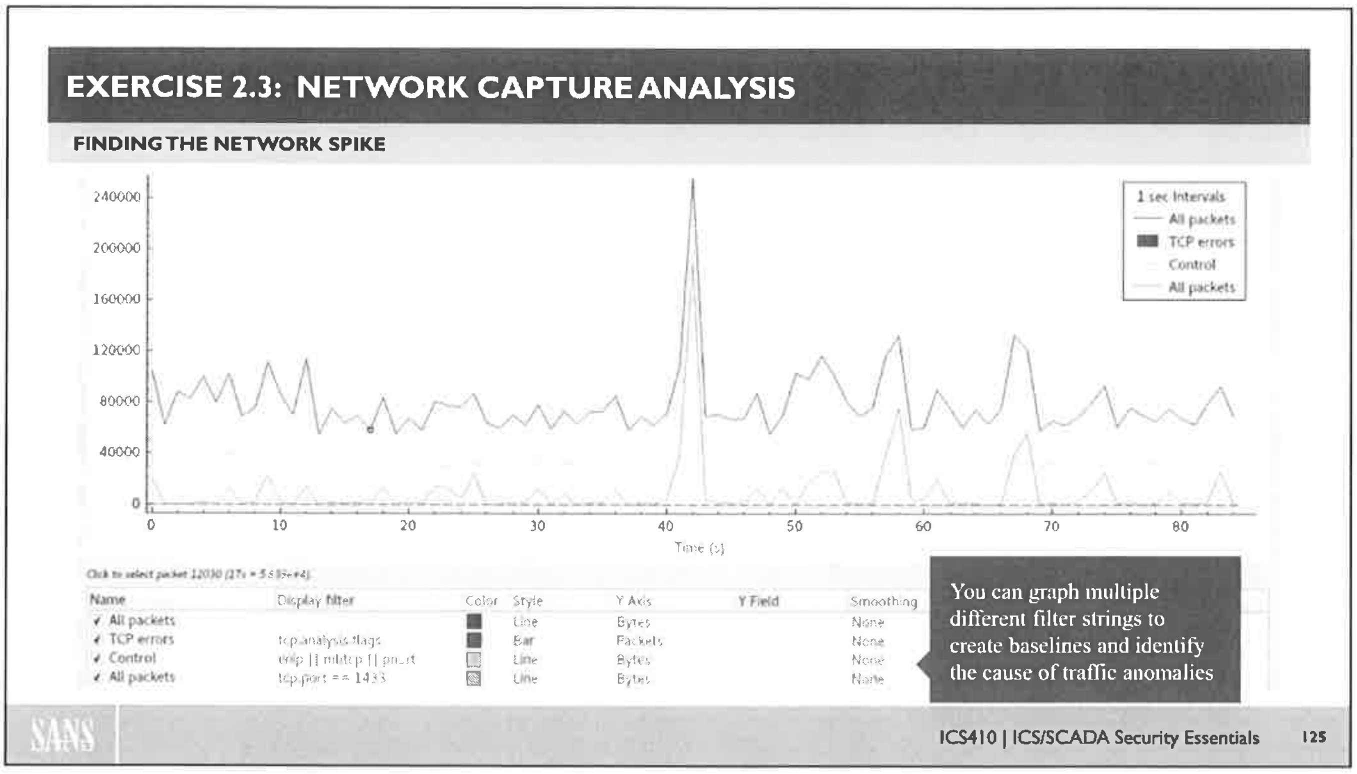
You can filter based on these commands with Wireshark's **modbus.func_code** variable. If you didn't know this filter string, you can learn it by looking through the Modbus filters in the Expression Builder (button right of the filter box) or finding a Modbus packet and right-clicking on the **Function Code** line and choosing **Apply as Filter**.

To do this, go back to the main Wireshark window and remove the previous filter we were using. Replace it with:

```
modbus.func_code == 5
```

Make sure you use two equal signs or this will not work. You will see that no packets match that Modbus write command. Now try changing it to a **6**. This should also provide no results. However, when you try commands **15** and **16**, you should see several resulting packets where the Modbus master is writing registers to the Modbus field devices. So yes, we do have evidence that the master is actively controlling the field devices by changing their coils and registers.

© 2019, Justin Searle



EXERCISE 2.3: NETWORK CAPTURE ANALYSIS

FINDING THE NETWORK SPIKE

Click to select packet 12030 (27s = 5.55e+4).

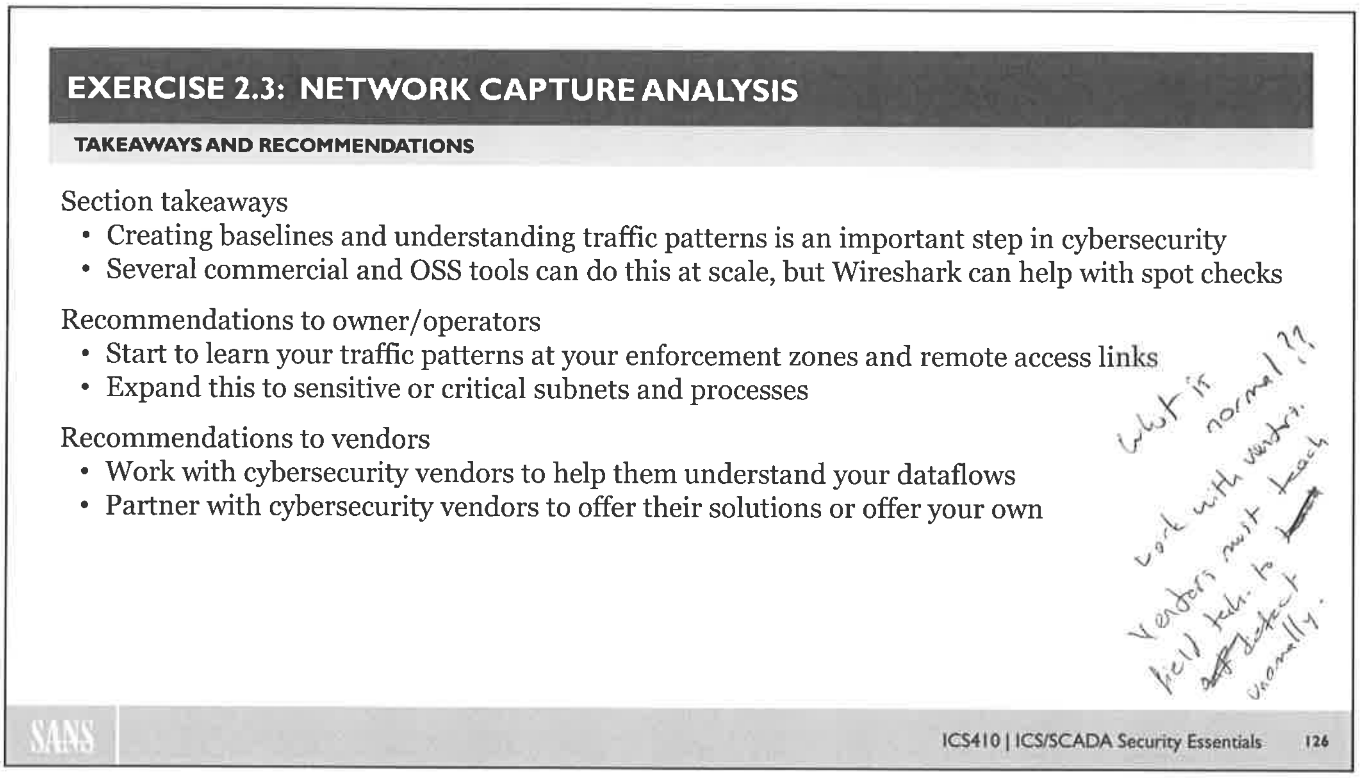
Name	Display filter	Color	Style	Y Axis	Y Field	Smoothing
✓ All packets		■	Line	Bytes		None
✓ TCP errors	tcp.analysis.flags	■	Bar	Packets		None
✓ Control	enip || mbtcp || pn_rt	□	Line	Bytes		None
✓ All packets	tcp.port == 1433	▨	Line	Bytes		None

You can graph multiple different filter strings to create baselines and identify the cause of traffic anomalies

SANS ICS410 | ICS/SCADA Security Essentials 125

If you use Wireshark's **IO Graph** feature under the Statistics menu, you will see there is a small spike of traffic right in the middle of the capture, around the 42-second timestamp. If you graph by bytes instead of packets, you will see the spike is even larger. If we want to determine if Modbus TCP was part of that traffic spike, we can use the multiple graph feature at the bottom of this window to show subsets of the traffic. The first graph line that shows in black has an empty filter and represents the default graph you see when you opened this window. Go to the second line and add a Modbus TCP filter of **modbus** and change its Y axis to **Bytes** to match the other entries. We can see in this graph that the Modbus TCP traffic accounts for about one-fourth of all the network traffic in this capture and that its communication rate stays constant. You can add a second line to the graph to indicate the sum of all four of the control protocols by using the filter **enip || mbtcp || pn_rt || s7comm**. Notice how flat and consistent all of these are. This is common for most ICSs. This shows that none of our control traffic is the cause of the traffic spike.

If you have extra time left in the lab, try to figure out which protocol is causing this spike. One idea that will help is to use the **Statistics > Endpoints** feature in Wireshark, on the TCP tab, and see which protocols are transmitting the most **Bytes**. After you see which protocols are speaking the most, go back to the **IO Graph** window and try graphing those protocols. Another method is to zoom into the IO graph and click on the spike to have Wireshark scroll to that general time period in the main window. This gets you close but doesn't identify the actual packets or protocol. But it will give you some ideas you can try to graph to see if you can find the matching graph. Of course, I show you the protocol in the graph above, but try to figure out how I came to that filter.



EXERCISE 2.3: NETWORK CAPTURE ANALYSIS

TAKEAWAYS AND RECOMMENDATIONS

Section takeaways
- Creating baselines and understanding traffic patterns is an important step in cybersecurity
- Several commercial and OSS tools can do this at scale, but Wireshark can help with spot checks

Recommendations to owner/operators
- Start to learn your traffic patterns at your enforcement zones and remote access links
- Expand this to sensitive or critical subnets and processes

Recommendations to vendors
- Work with cybersecurity vendors to help them understand your dataflows
- Partner with cybersecurity vendors to offer their solutions or offer your own

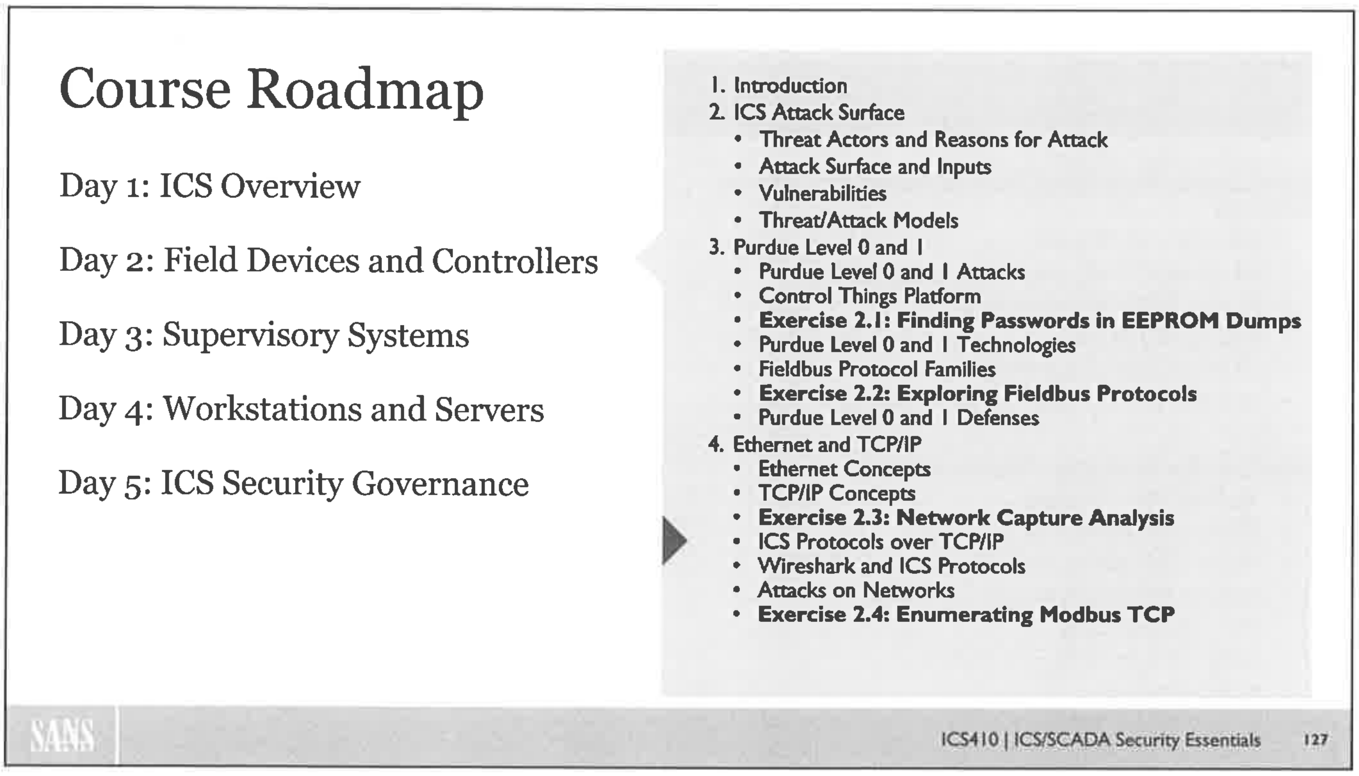
What is normal??
work with vendors.
vendors must teach ~~[illegible]~~
help tech. to ~~aff~~ detect
anomally.

SANS

ICS410 | ICS/SCADA Security Essentials 126

This page intentionally left blank.

© 2019, Justin Searle



Course Roadmap

Day 1: ICS Overview

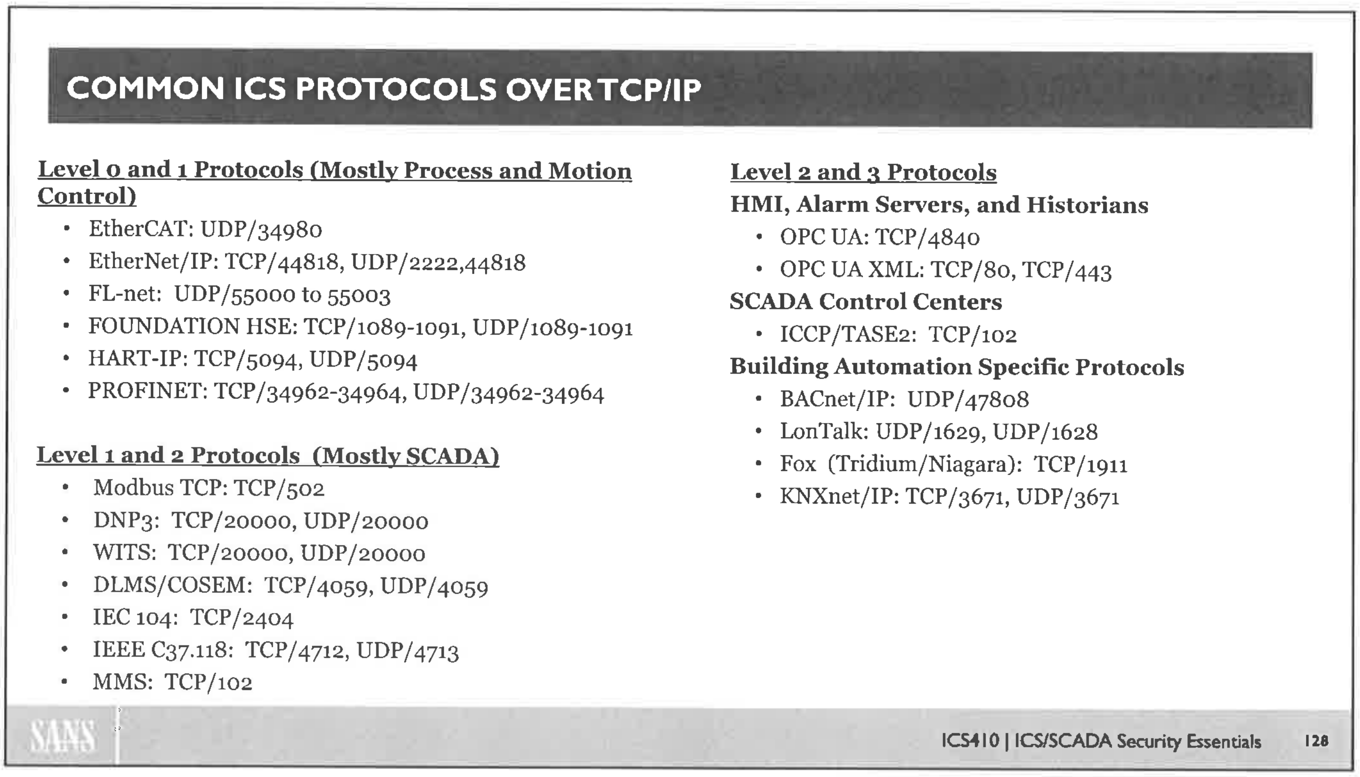
Day 2: Field Devices and Controllers

Day 3: Supervisory Systems

Day 4: Workstations and Servers

Day 5: ICS Security Governance

This page intentionally left blank.



COMMON ICS PROTOCOLS OVER TCP/IP

Level 0 and 1 Protocols (Mostly Process and Motion Control)

- EtherCAT: UDP/34980
- EtherNet/IP: TCP/44818, UDP/2222,44818
- FL-net: UDP/55000 to 55003
- FOUNDATION HSE: TCP/1089-1091, UDP/1089-1091
- HART-IP: TCP/5094, UDP/5094
- PROFINET: TCP/34962-34964, UDP/34962-34964

Level 1 and 2 Protocols (Mostly SCADA)

- Modbus TCP: TCP/502
- DNP3: TCP/20000, UDP/20000
- WITS: TCP/20000, UDP/20000
- DLMS/COSEM: TCP/4059, UDP/4059
- IEC 104: TCP/2404
- IEEE C37.118: TCP/4712, UDP/4713
- MMS: TCP/102

Level 2 and 3 Protocols

HMI, Alarm Servers, and Historians

- OPC UA: TCP/4840
- OPC UA XML: TCP/80, TCP/443

SCADA Control Centers

- ICCP/TASE2: TCP/102

Building Automation Specific Protocols

- BACnet/IP: UDP/47808
- LonTalk: UDP/1629, UDP/1628
- Fox (Tridium/Niagara): TCP/1911
- KNXnet/IP: TCP/3671, UDP/3671

SANS ICS410 | ICS/SCADA Security Essentials 128

A larger, more comprehensive list can be found at http://en.wikipedia.org/wiki/List_of_automation_protocols

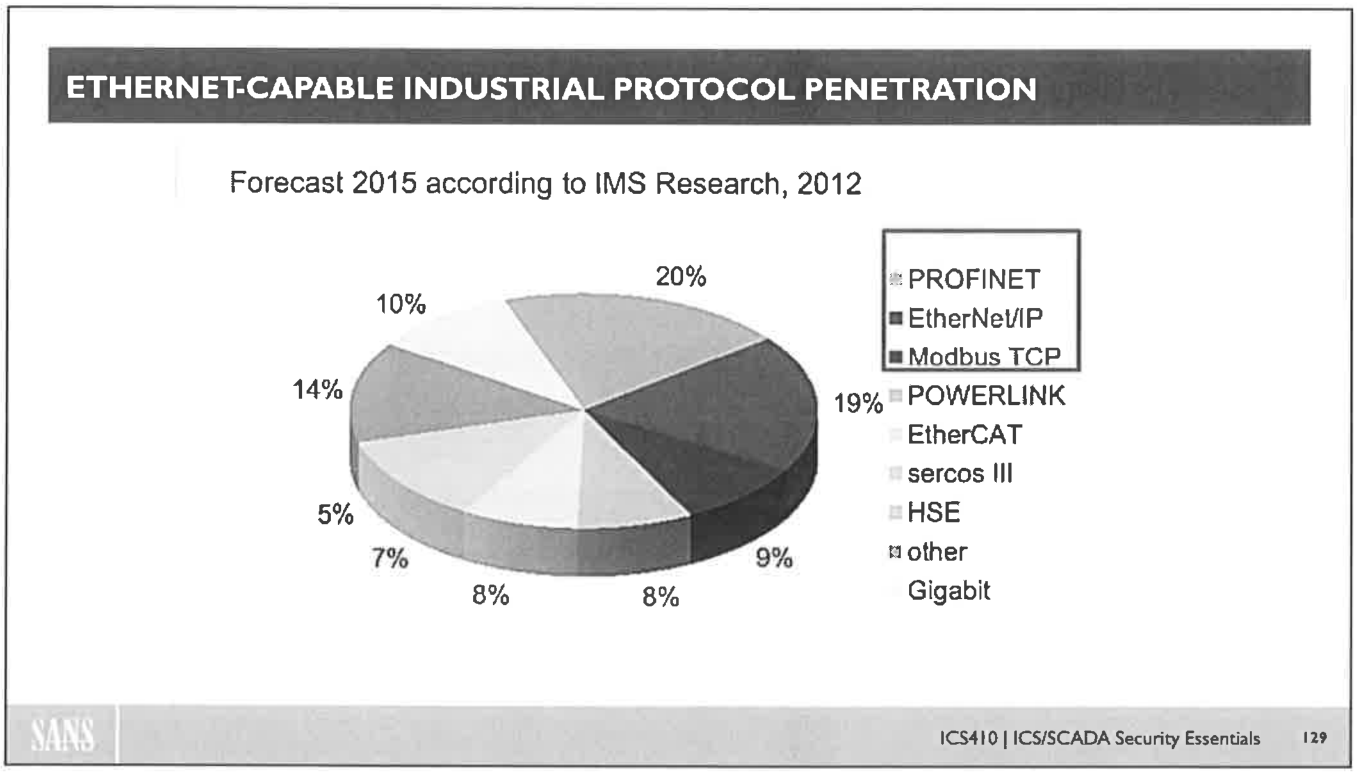
There are a number of open protocols used for industrial control systems. Open protocols are available for any vendor which choose to join the appropriate consortium and follow the standard body's requirements, which is a bit different from concept of open protocols by the Internet Engineering Task Force (IETF) and their Request for Comments (RFCs) in the IT world. The open standard process in the ICS world is typically open to all consortium members and is carried out through a collaboration to develop the protocol for common use.

Most ICS vendor equipment will support at least one or more open protocols and quite often one proprietary protocol for that vendor for device management and maintenance using vendor proprietary tools. ICS competitors have made agreements in the past to share and support each other's proprietary protocols, but that is an exception to the rule, not the norm. And ICS suppliers are not the only ones creating proprietary protocols. It is common for owner/operators to develop in-house protocols to solve small-scale control or engineering challenges, and in the modern world, much like IT, this is often through the use of web services.

There has been a significant increase in the demand for well-accepted or standard and open protocols to support system interoperability and sustainability. The auto industry, for example, is using a small number of protocols for use in vehicles, and the debate on smart grids in the US had a lot to do with supporting interoperability through the selection of standards to be used for grid modernization.

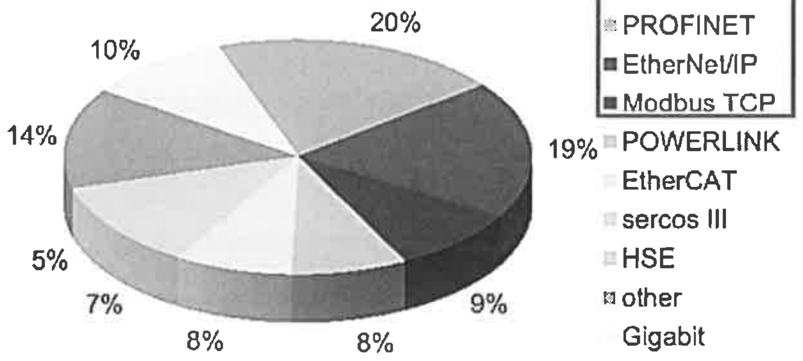
Remember that although many ICS protocols are not TCP/IP-based, it is common to encapsulate them in TCP/IP protocols. If an ICS protocol doesn't have a TCP/IP-based version, you may see the serial-based version of the protocol encapsulated in a TCP or UDP packet. In fact, many of the TCP/IP-based versions are just that, a formal standard on how to encapsulate the raw serial traffic in TCP or UDP, such as in the case of Modbus TCP. You may even see protocols that are encapsulated directly in an Ethernet frame or their own frame on an Ethernet medium, as in the case of IEC 61850's Goose messaging.

© 2019, Justin Searle



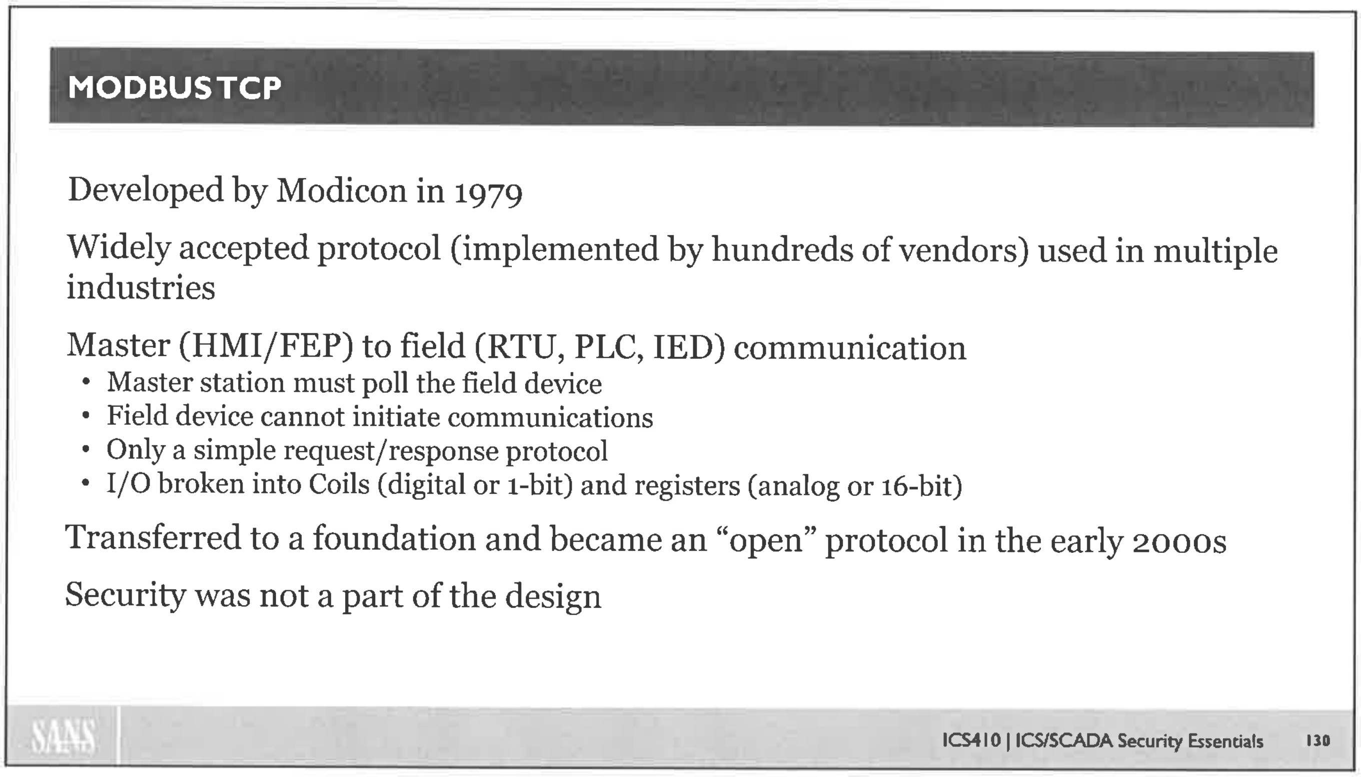
ETHERNET-CAPABLE INDUSTRIAL PROTOCOL PENETRATION

Forecast 2015 according to IMS Research, 2012



Ethernet-capable protocols have been growing in industrial automation for the reasons provided on the previous slide. The current market leaders have been PROFINET, EtherNet/IP, and Modbus TCP, as they are used for a number of automation and industrial missions, but you will find a large adoption of application-specific protocols such as DNP3 IP in the electric power industry. IMS Research published this graph in 2012.

The use of Ethernet networks down to the plant floor has resulted in traditional serial protocols used for I/O traffic to develop IP versions. The FieldComm Group (previously known as the HART Communication Foundation) is the body tasked with developing and promoting the use of the HART communication protocol for diagnostics and input/output of process equipment and variables. The FieldComm Group released HART-IP in 2012. HART-IP is beginning to see growing adoption. These modifications for use on Ethernet networks have received attention from security researchers. In January 2014, Russian security researcher Alexander Bolshev discussed security weaknesses in HART and HART-IP.



MODBUS TCP

Developed by Modicon in 1979

Widely accepted protocol (implemented by hundreds of vendors) used in multiple industries

Master (HMI/FEP) to field (RTU, PLC, IED) communication

- Master station must poll the field device
- Field device cannot initiate communications
- Only a simple request/response protocol
- I/O broken into Coils (digital or 1-bit) and registers (analog or 16-bit)

Transferred to a foundation and became an “open” protocol in the early 2000s

Security was not a part of the design

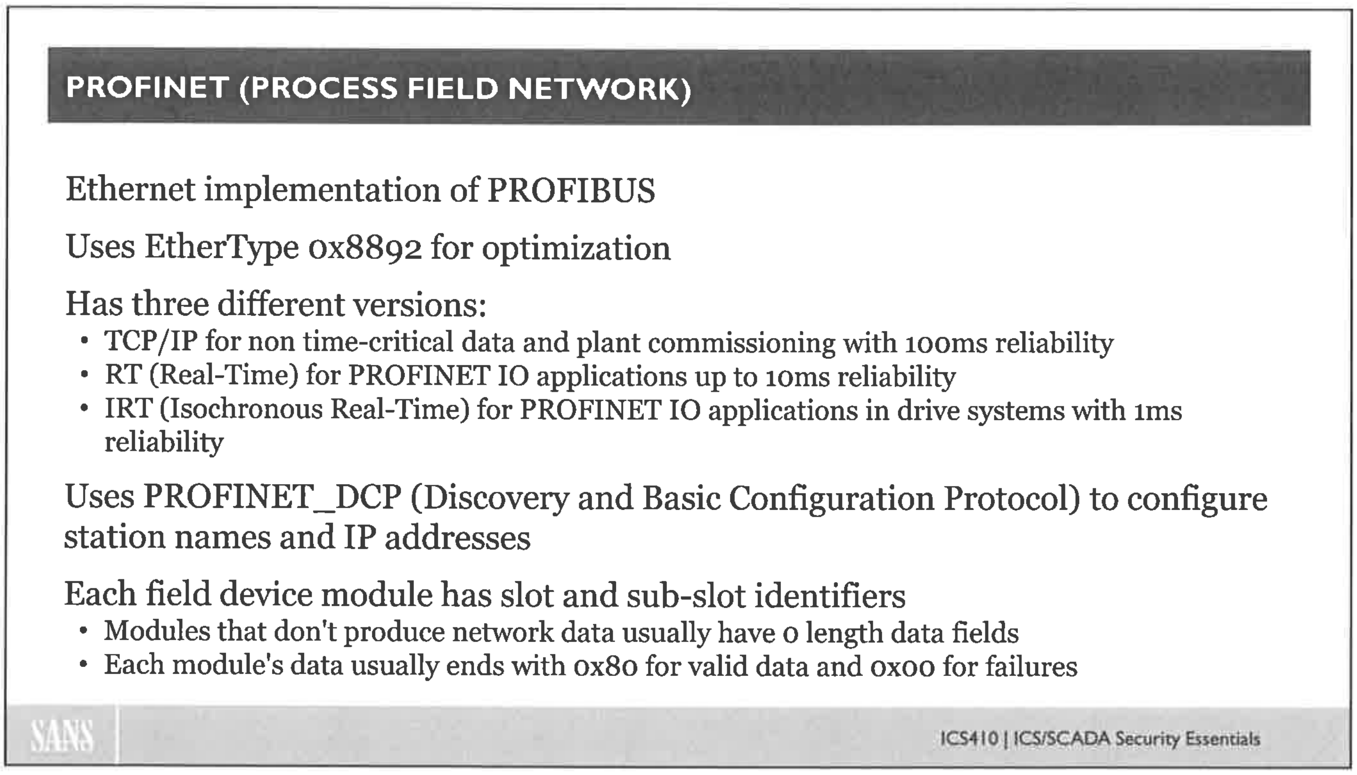
SANS ICS410 | ICS/SCADA Security Essentials 130

Modbus was designed in the late '70s to provide simple and robust communications from masters (HMIs) to field devices (PLCs). Modbus became a de facto standard with its wide adoption across multiple industries and applications, designed as a request-and-response protocol that is unable to report by exception (requires polling from the master). Although developed by a vendor, it has since been moved to a foundation and made into an open standard. Modbus is a serial protocol, but a TCP/IP version was developed. Security is not included in the serial or TCP design, and the protocol is vulnerable to interception and injection. It is critical to restrict access to servers and clients and protect communication paths.

References:
Modbus foundation: http://www.modbus.org

As a popular open protocol, one can find SNORT signatures available online for NIDS. A good resource for this is:
http://www.sans.org/reading-room/whitepapers/detection/snort-intrusion-detection-modbus-tcp-ip-communications-33844

EU → IEC 104 } only in IEC 104
US → DNP3 } energy sector -



PROFINET (PROCESS FIELD NETWORK)

Ethernet implementation of PROFIBUS

Uses EtherType 0x8892 for optimization

Has three different versions:

- TCP/IP for non time-critical data and plant commissioning with 100ms reliability
- RT (Real-Time) for PROFINET IO applications up to 10ms reliability
- IRT (Isochronous Real-Time) for PROFINET IO applications in drive systems with 1ms reliability

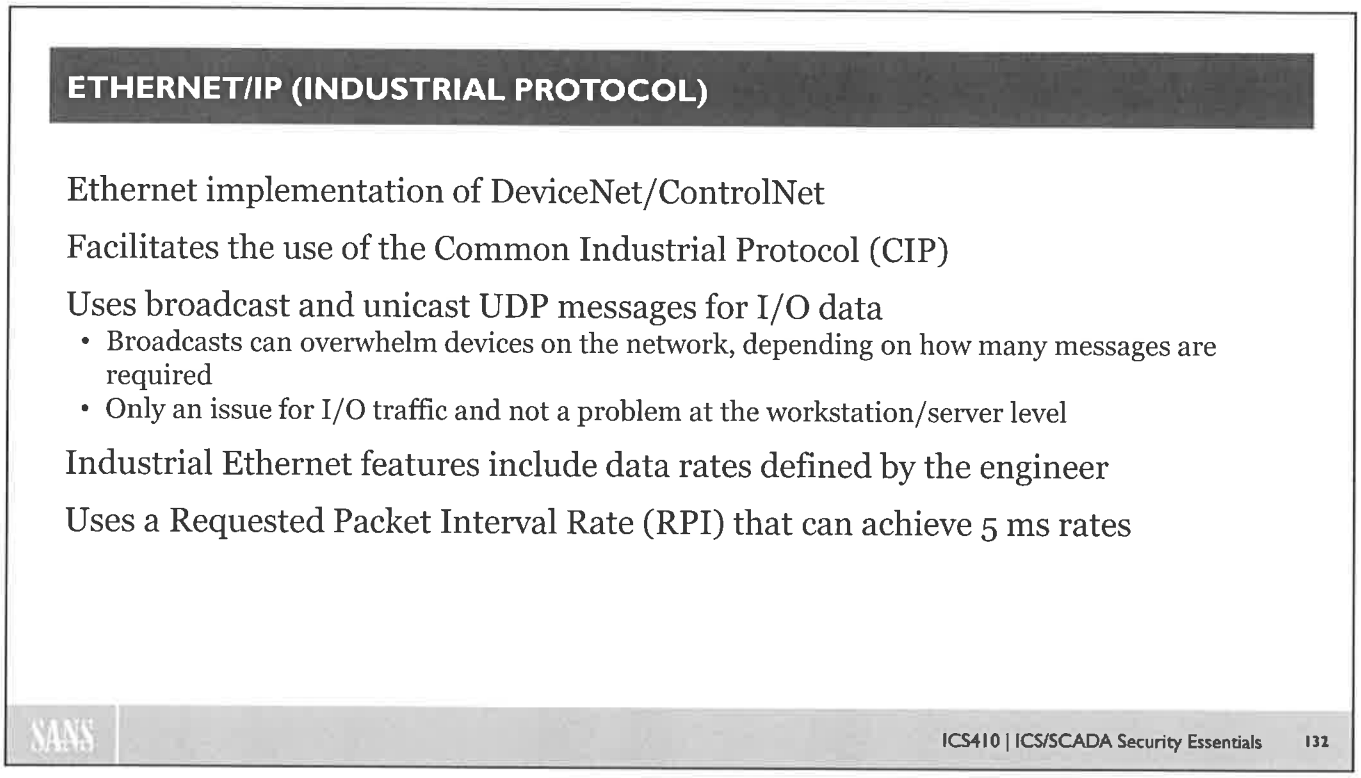
Uses PROFINET_DCP (Discovery and Basic Configuration Protocol) to configure station names and IP addresses

Each field device module has slot and sub-slot identifiers

- Modules that don't produce network data usually have 0 length data fields
- Each module's data usually ends with 0x80 for valid data and 0x00 for failures

SANS ICS410 | ICS/SCADA Security Essentials

This page intentionally left blank.



ETHERNET/IP (INDUSTRIAL PROTOCOL)

Ethernet implementation of DeviceNet/ControlNet

Facilitates the use of the Common Industrial Protocol (CIP)

Uses broadcast and unicast UDP messages for I/O data

- Broadcasts can overwhelm devices on the network, depending on how many messages are required
- Only an issue for I/O traffic and not a problem at the workstation/server level

Industrial Ethernet features include data rates defined by the engineer

Uses a Requested Packet Interval Rate (RPI) that can achieve 5 ms rates

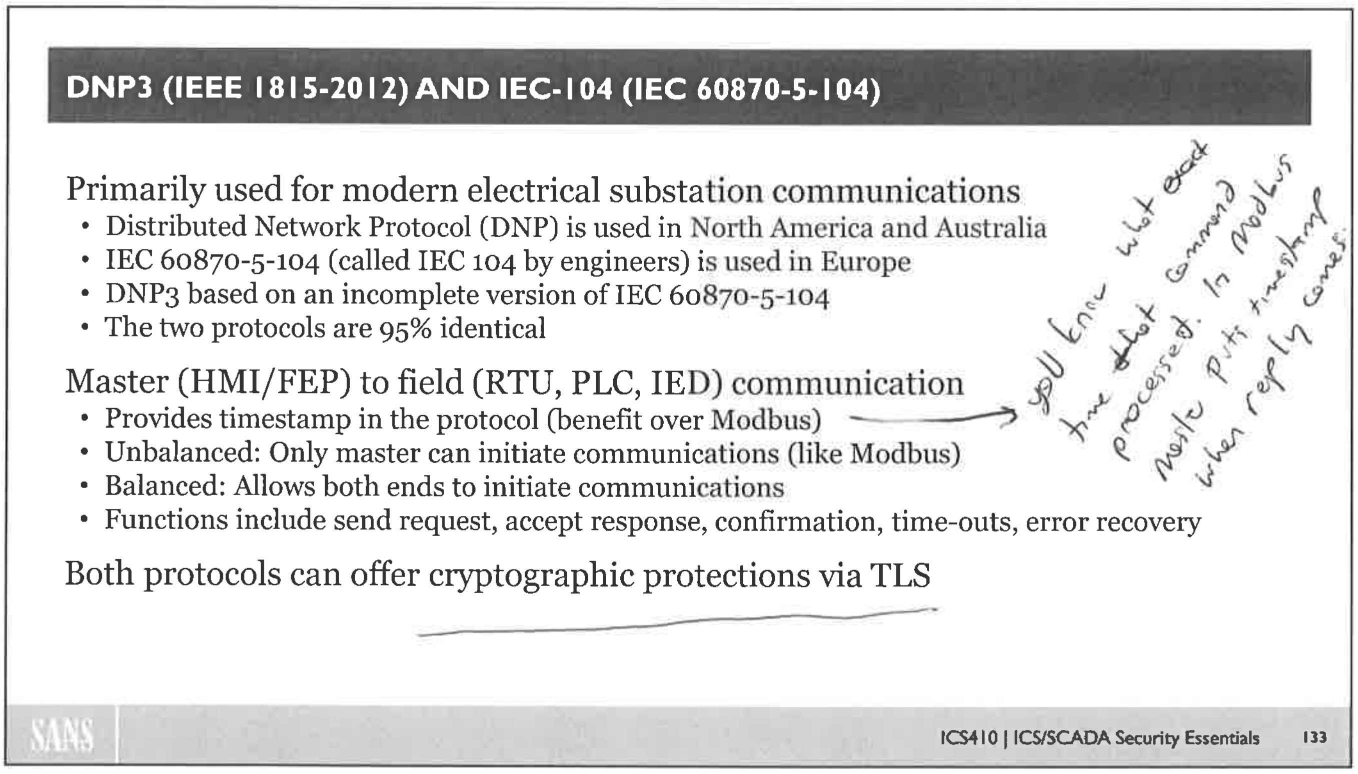
EtherNet/IP was built so that manufacturing automation could leverage Ethernet infrastructure to manage connections between devices (hosts, PLCs, instruments, actuators, CNC machines, and such). EtherNet/IP arrived on the market in 2001 and has been developed to enable a multitude of control system applications/roles. EtherNet/IP was used to easily integrate automation or plant networks with enterprise systems and enable internet connectivity. The standard is managed by Open DeviceNet Vendors Association (ODVA). It is designed to be robust and can be described as complex. EtherNet/IP is based on the Common Industrial Protocol (CIP), which is implemented at the session layer and higher in the OSI model. In addition to EtherNet/IP technologies, ODVA is responsible for CIP protocol, DeviceNet, and ControlNet.

Ethernet was first used to connect automation systems to company networks for the purposes of more easily sharing data. The non-deterministic nature of Ethernet limited its penetration into the control systems. EtherNet/IP (IP stands for Industrial Protocol versus Internet Protocol) was designed to overcome some of those challenges to better align with the needs of industrial automation and control. EtherNet/IP is used to connect isolated pockets of automation from batch, continuous process, discrete, safety, and drive control in one plant into a networked system. The final penetration is the use of EtherNet/IP down to the actual sensor or smart device. Ethernet-based networks can be found connecting controllers to variable-speed drives, actuators, flow meters, and so on.

EtherNet/IP used broadcast UDP messages for I/O data. Broadcasts can overwhelm devices on the network, depending on how many messages are required. IGMP snooping on a managed switch can help mitigate if the network connects several devices. This is only an issue for I/O traffic and not a problem at the workstation/server level. Industrial Ethernet features include data rates that can be defined by the control engineer to best meet the needs of the process. For example, EtherNet/IP uses a Requested Packet Interval Rate (RPI) and can achieve 5 ms to 10 seconds. EtherNet/IP can broadcast and newer versions are able to support unicast messages.

Reference:
https://en.wikipedia.org/wiki/EtherNet/IP

© 2019, Justin Searle



DNP3 (IEEE 1815-2012) AND IEC-104 (IEC 60870-5-104)

Primarily used for modern electrical substation communications

- Distributed Network Protocol (DNP) is used in North America and Australia
- IEC 60870-5-104 (called IEC 104 by engineers) is used in Europe
- DNP3 based on an incomplete version of IEC 60870-5-104
- The two protocols are 95% identical

Master (HMI/FEP) to field (RTU, PLC, IED) communication

- Provides timestamp in the protocol (benefit over Modbus) → you know what exact time ~~that~~ that command processed. In Modbus master puts timestamp when reply comes.
- Unbalanced: Only master can initiate communications (like Modbus)
- Balanced: Allows both ends to initiate communications
- Functions include send request, accept response, confirmation, time-outs, error recovery

Both protocols can offer cryptographic protections via TLS

DNP is an open standard and achieved needed interoperability between vendors supporting electricity transmission and distribution systems. DNP stands for Distributed Network Protocol and is designed for SCADA Master to field device communications. DNP3 is widely used in electricity applications and is used far less in other industries. The protocol was designed with reliability requirements at the top of the list. The protocol lacked any native authentication mechanism and did not safeguard data integrity outside of error checking.

While IEC 60870-5 was being developed, a group of vendors in North America implemented an incomplete version of it and named it DNP3. DNP3 development attempted to maintain synchronization with the standard; however, it resulted in a slightly different protocol which is predominantly used in North America and Australia, while IEC 60870-5 is primarily used in Europe.

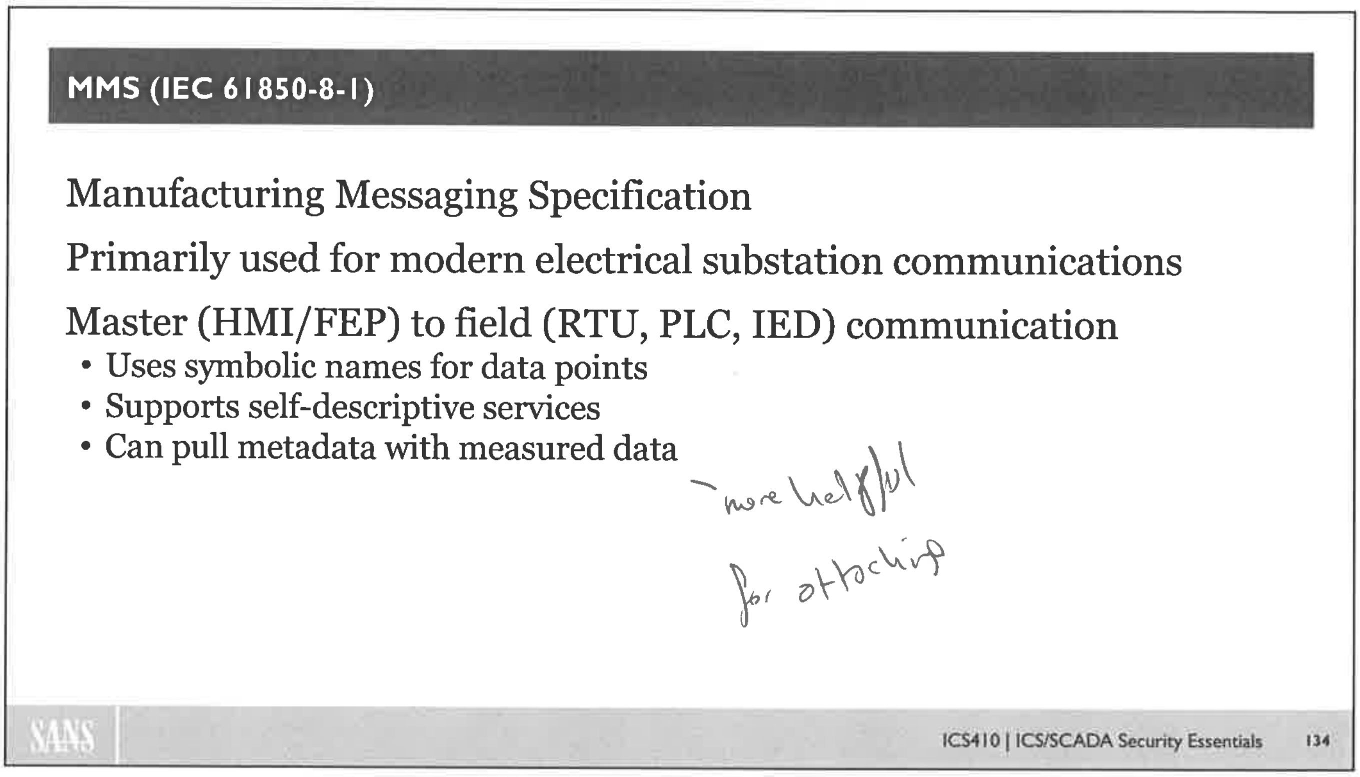
A secure authentication capability has been developed but not widely implemented. Additional controls and security solutions are typically recommended when security is a requirement based on the particular application and use of the protocol.

References:

YouTube playlist for a DNP3 Protocol Primer: http://bit.ly/2NaZttt

YouTube playlist for a IEC-104 Protocol Primer: http://bit.ly/2SHwpjc

Secure DNP3 reference: http://www.digitalbond.com/scadapedia/protocols/secure-dnp3/



MMS (IEC 61850-8-1)

Manufacturing Messaging Specification

Primarily used for modern electrical substation communications

Master (HMI/FEP) to field (RTU, PLC, IED) communication

- Uses symbolic names for data points
- Supports self-descriptive services
- Can pull metadata with measured data

– more helpful for attacking

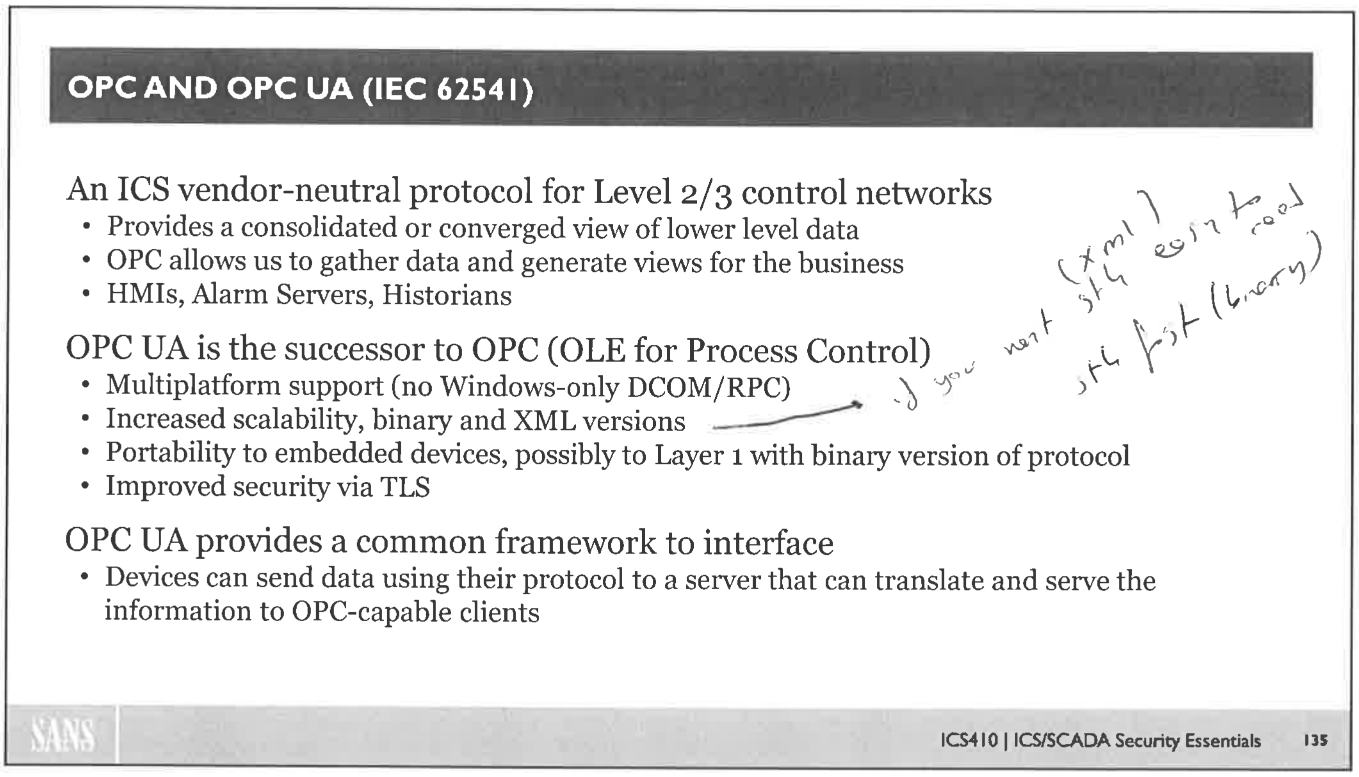
SANS ICS410 | ICS/SCADA Security Essentials 134

References:

YouTube playlist for a IEC-61850 Primer: http://bit.ly/2SWd8tn

© 2019, Justin Searle

OLE for Process Control.



OPC AND OPC UA (IEC 62541)

An ICS vendor-neutral protocol for Level 2/3 control networks

- Provides a consolidated or converged view of lower level data
- OPC allows us to gather data and generate views for the business
- HMIs, Alarm Servers, Historians

OPC UA is the successor to OPC (OLE for Process Control)

- Multiplatform support (no Windows-only DCOM/RPC)
- Increased scalability, binary and XML versions → if you want sth easy to read (xml) sth fast (binary)
- Portability to embedded devices, possibly to Layer 1 with binary version of protocol
- Improved security via TLS

OPC UA provides a common framework to interface

- Devices can send data using their protocol to a server that can translate and serve the information to OPC-capable clients

SANS ICS410 | ICS/SCADA Security Essentials 135

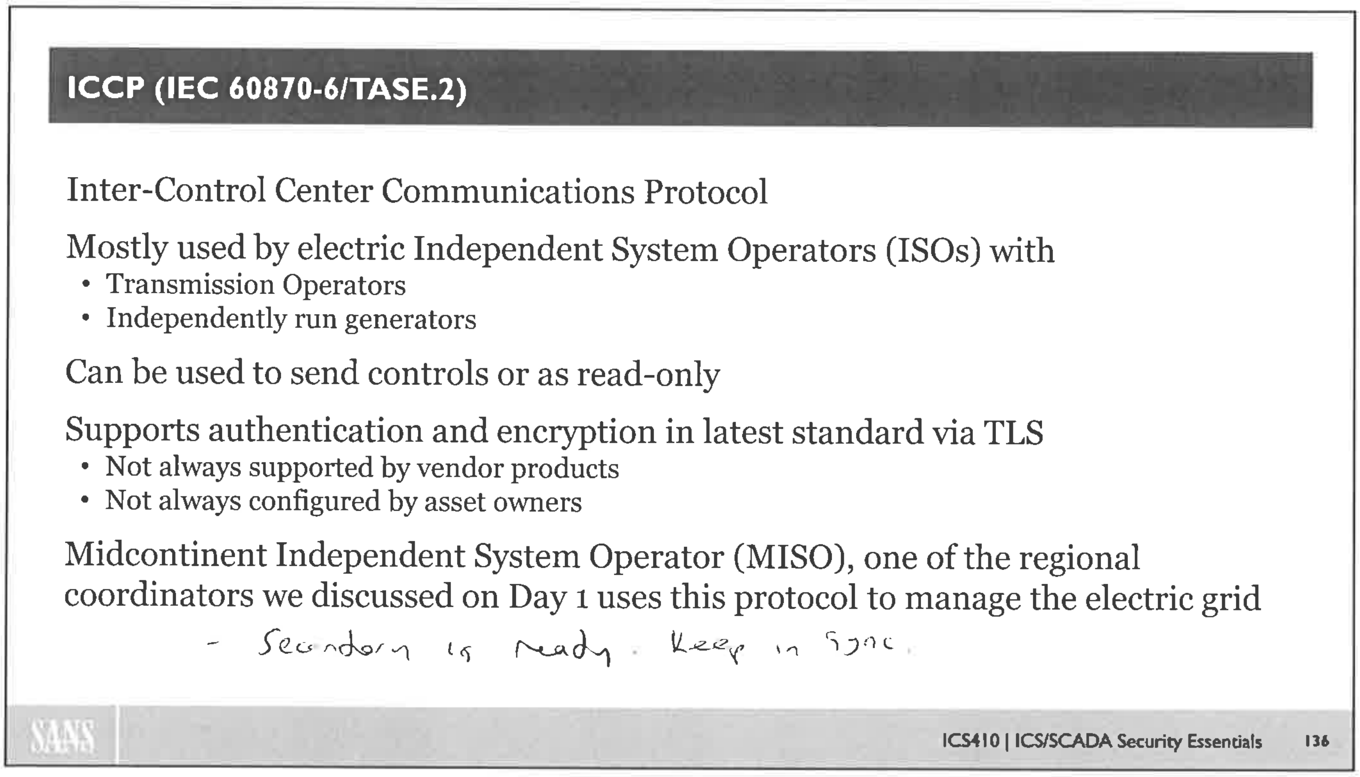
OLE is a technology developed by Microsoft and stands for Object Linking and Embedding; it is used to embed and link to documents and other objects within an application. OPC was designed to gather relevant data from important systems and provide a consolidated or converged view of that with data from another disparate system. Previously, control system vendor A would need to write custom interfaces and code to see control system B data, or humans would enter data manually, as business intelligence systems developed and users outside of the control system environment wanted to see operations performance data, production schedules, maintenance work, and more. There was a growing need to get data in a single system to allow viewing and access. By using OPC entities, we were able to gather data and generate the views that were needed for the business.

Often described as the "interoperability standard" for industrial automation, OPC originally stood for Object Linking and Embedding for Process Control. OPC was developed through industry collaboration in the late '90s. OPC has been specified by the community to stand for "open platform communications." The standard has continued to grow and evolve as it has integrated new technologies. It acts like a common bridge allowing communications that come to the OPC server to be provided to OPC clients. Based on OLE, COM, and DCOM from Microsoft, DCOM-related vulnerabilities have resulted in susceptible systems using OPC. DCOM/RPC is complex and this results in bugs that are difficult to anticipate and discover. DCOM-based vulnerabilities, such as MS-086 needed to be addressed by control system end users. Also, some consider OPC a firewall-unfriendly protocol that takes extra care to segment properly.

OPC Servers can be a bridge for an attacker from one system to another.

References:
http://www.opcfoundation.org
See ICS-CERT advisory: http://ics-cert.us-cert.gov/alerts/ICS-ALERT-11-285-01

Good OPC security reference: http://www.pacontrol.com/download/OPC-Security-WP2.pdf



ICCP (IEC 60870-6/TASE.2)

Inter-Control Center Communications Protocol

Mostly used by electric Independent System Operators (ISOs) with
- Transmission Operators
- Independently run generators

Can be used to send controls or as read-only

Supports authentication and encryption in latest standard via TLS
- Not always supported by vendor products
- Not always configured by asset owners

Midcontinent Independent System Operator (MISO), one of the regional coordinators we discussed on Day 1 uses this protocol to manage the electric grid

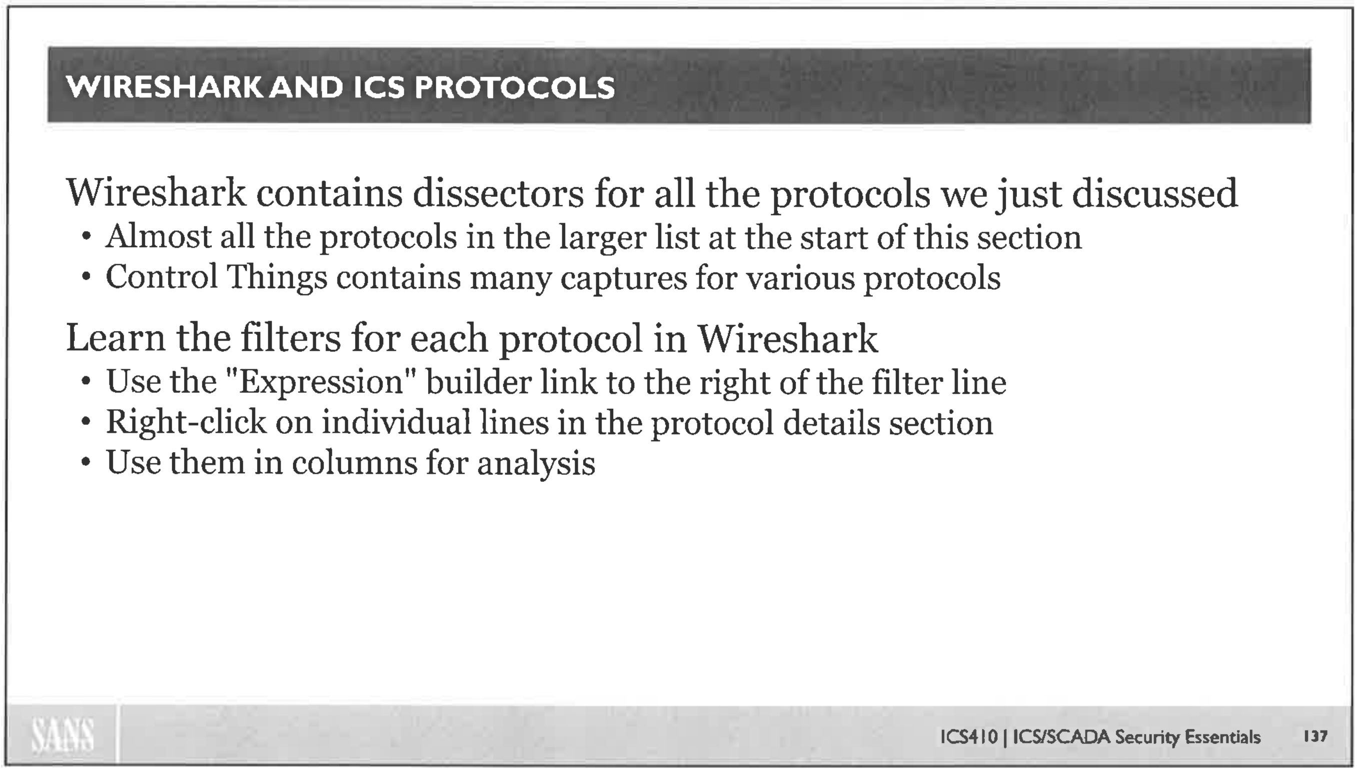
- Secondary is ready. Keep in sync.

SANS ICS410 | ICS/SCADA Security Essentials 136

The Inter-Control Center Communications Protocol (ICCP) is a standardized protocol used for formatting and the exchange of data between control centers and utilities. The most widespread use of ICCP is probably the network connecting electric Independent System Operators (ISOs) with Transmission Operators and other market participants, such as independently run generators. ICCP may be used to communicate real-time usage, command setpoints to generators, or even provide simple messaging between users. The ICCP protocol provides no means for authentication or encryption, and thus any security must be provided at the physical or transport layers by use of air gaps or firewalls.

An ICS example of a wide area communications system for Supervisory Control and Data Acquisition is ICCP. ICCP is utilized to provide SCADA data existing in one control center to another control center. Typically, organizations that have a common system from a primary to a backup control center utilize system replication techniques and vendor-supplied protocols; however, when one organization with one control center wants a subset of SCADA data from another organization at a different control center and often on a different system, they can utilize ICCP to achieve this goal. The requirement for ICCP to function is that both systems at each independent control center support ICCP protocol and message formats.

ICCP also provides the ability for a control center environment to connect with numerous ICCP feeds from a variety of other ICCP sources or peers. The example previously given of MISO is an example of one control center receiving ICCP feeds from dozens of other control centers. This aggregation of data and control capability provides MISO with a wide area view of the MISO footprint and provides the capability needed to ensure overall system reliability.



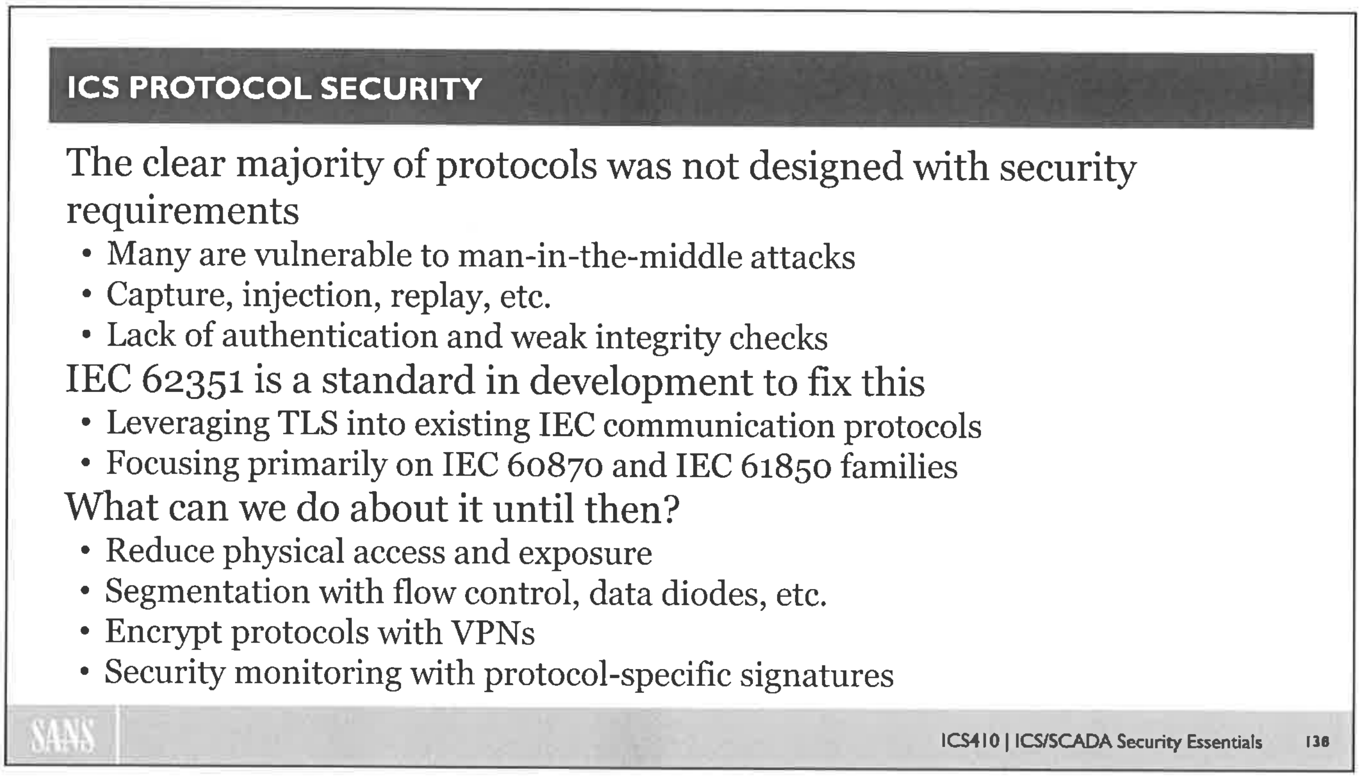
WIRESHARK AND ICS PROTOCOLS

Wireshark contains dissectors for all the protocols we just discussed
- Almost all the protocols in the larger list at the start of this section
- Control Things contains many captures for various protocols

Learn the filters for each protocol in Wireshark
- Use the "Expression" builder link to the right of the filter line
- Right-click on individual lines in the protocol details section
- Use them in columns for analysis

Wireshark Display Filters include a number of ICS protocols and characterize a number of the important fields. Filters help remove the noise from a packet capture to let you see only the packets you are interested in. It is important to understand what is traversing the wire, so use these filters to drill down inside of a protocol to see what is moving to and from hosts, but don't forget to examine what is occurring outside of the expected protocol traffic.

References:
Resource: https://www.wireshark.org/docs/dfref/#section_d
Modbus Example: https://www.wireshark.org/docs/dfref/m/modbus.html
DNP3 Example: https://www.wireshark.org/docs/dfref/d/dnp3.html



ICS PROTOCOL SECURITY

The clear majority of protocols was not designed with security requirements

- Many are vulnerable to man-in-the-middle attacks
- Capture, injection, replay, etc.
- Lack of authentication and weak integrity checks

IEC 62351 is a standard in development to fix this

- Leveraging TLS into existing IEC communication protocols
- Focusing primarily on IEC 60870 and IEC 61850 families

What can we do about it until then?

- Reduce physical access and exposure
- Segmentation with flow control, data diodes, etc.
- Encrypt protocols with VPNs
- Security monitoring with protocol-specific signatures

ICS410 | ICS/SCADA Security Essentials 138

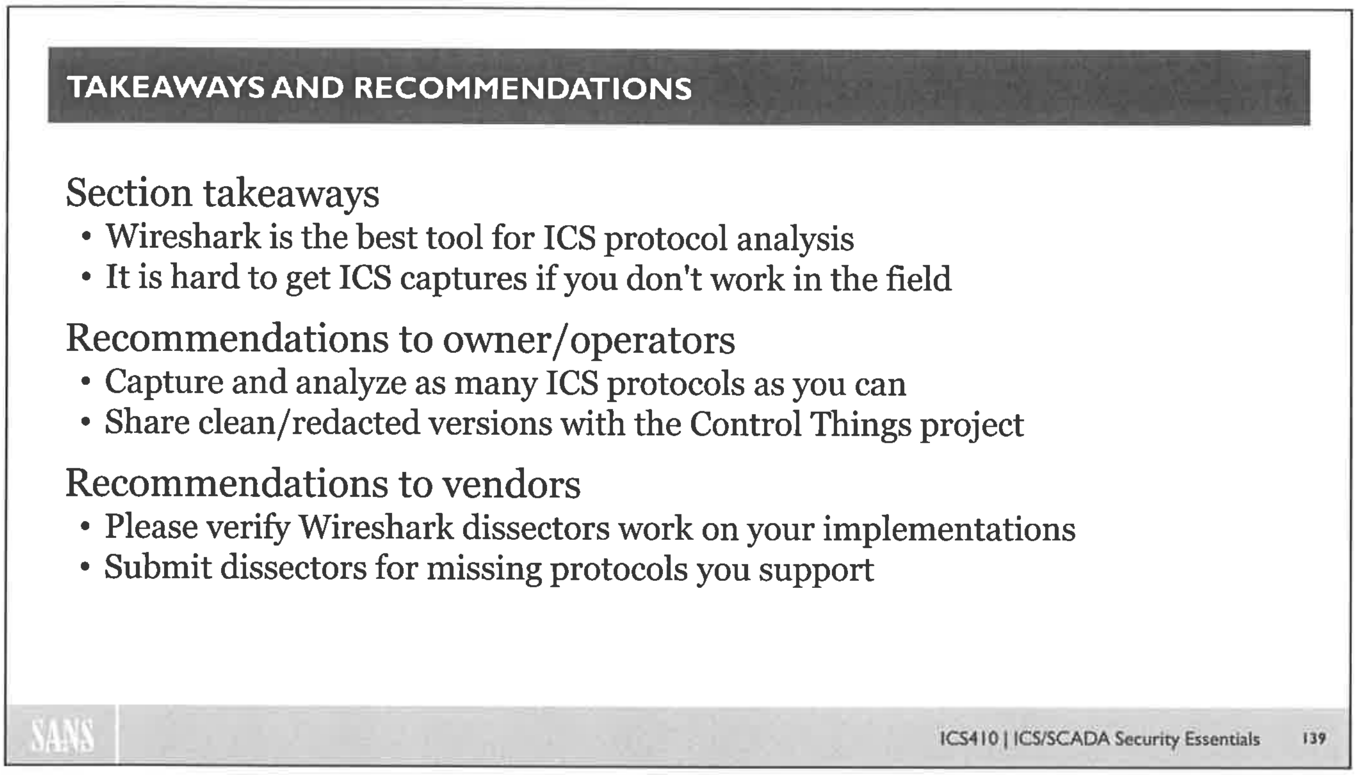
The majority of industrial protocols were designed when systems were less connected and accessible. There are few protocols that have basic security features developed into the design. Protocols typically lack authentication and basic data integrity protections. Many are vulnerable to MITM attacks and allow for capture, injection, replay/spoofing, and so on. Attacks can be crafted if the attacker can reverse engineer the protocol with the aid of captured traffic or open standard information.

IEC 62351 is a standard to help bring security to the TC 57 series of protocols like IEC 60870-5, IEC 60870-6, IEC 61850, IEC 61970, and IEC 61968. This includes authentication, digital signatures, and encryption.

The most common defenses relied upon include basic segmentation and enforcement. Firewalls, Restricted ACLs, and the use of data diodes keep the unprotected network and bus communications out of the hands of would-be attackers. It is also important to keep attackers physically away from devices. Some protocols have had specific signatures developed for use in IDS/IPS.

Reference:
https://en.wikipedia.org/wiki/IEC_62351

© 2019, Justin Searle



TAKEAWAYS AND RECOMMENDATIONS

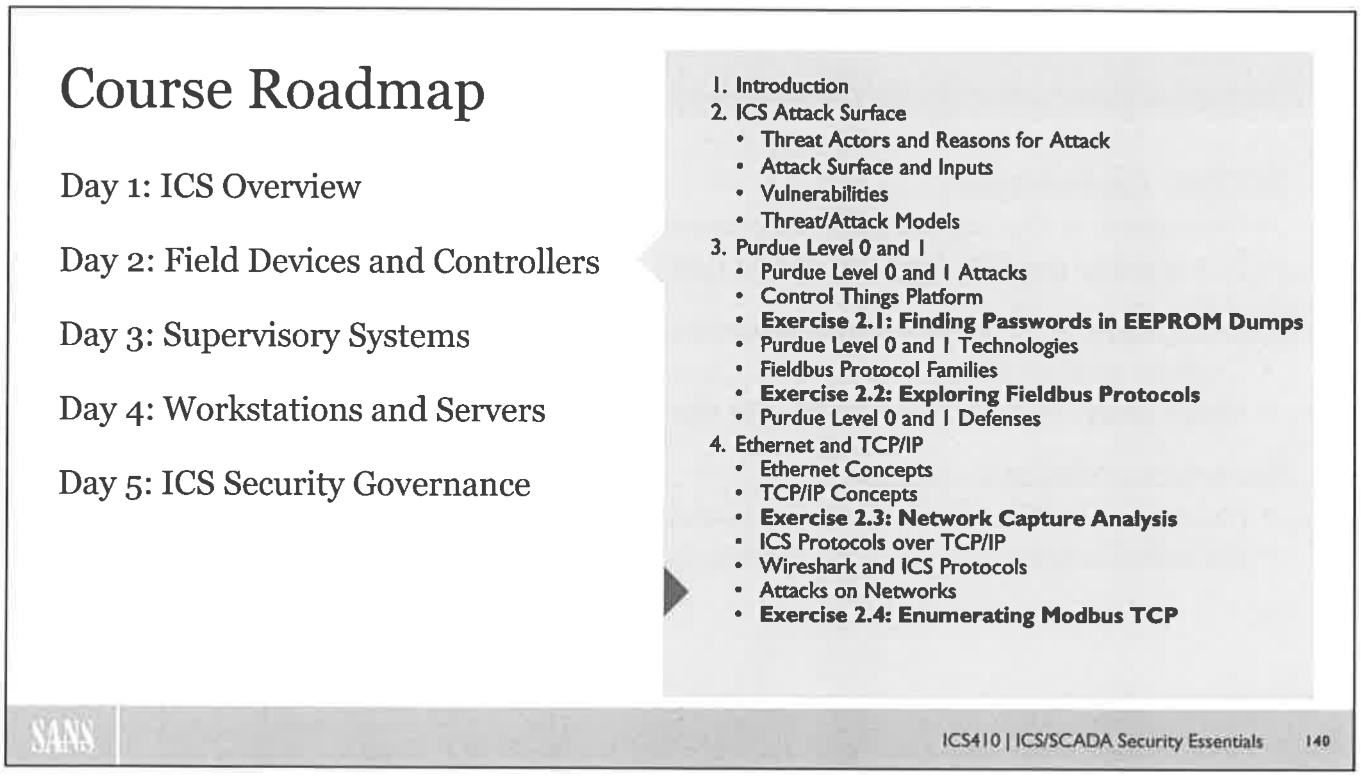
Section takeaways
- Wireshark is the best tool for ICS protocol analysis
- It is hard to get ICS captures if you don't work in the field

Recommendations to owner/operators
- Capture and analyze as many ICS protocols as you can
- Share clean/redacted versions with the Control Things project

Recommendations to vendors
- Please verify Wireshark dissectors work on your implementations
- Submit dissectors for missing protocols you support

SANS ICS410 | ICS/SCADA Security Essentials 139

This page intentionally left blank.



Course Roadmap

Day 1: ICS Overview

Day 2: Field Devices and Controllers

Day 3: Supervisory Systems

Day 4: Workstations and Servers

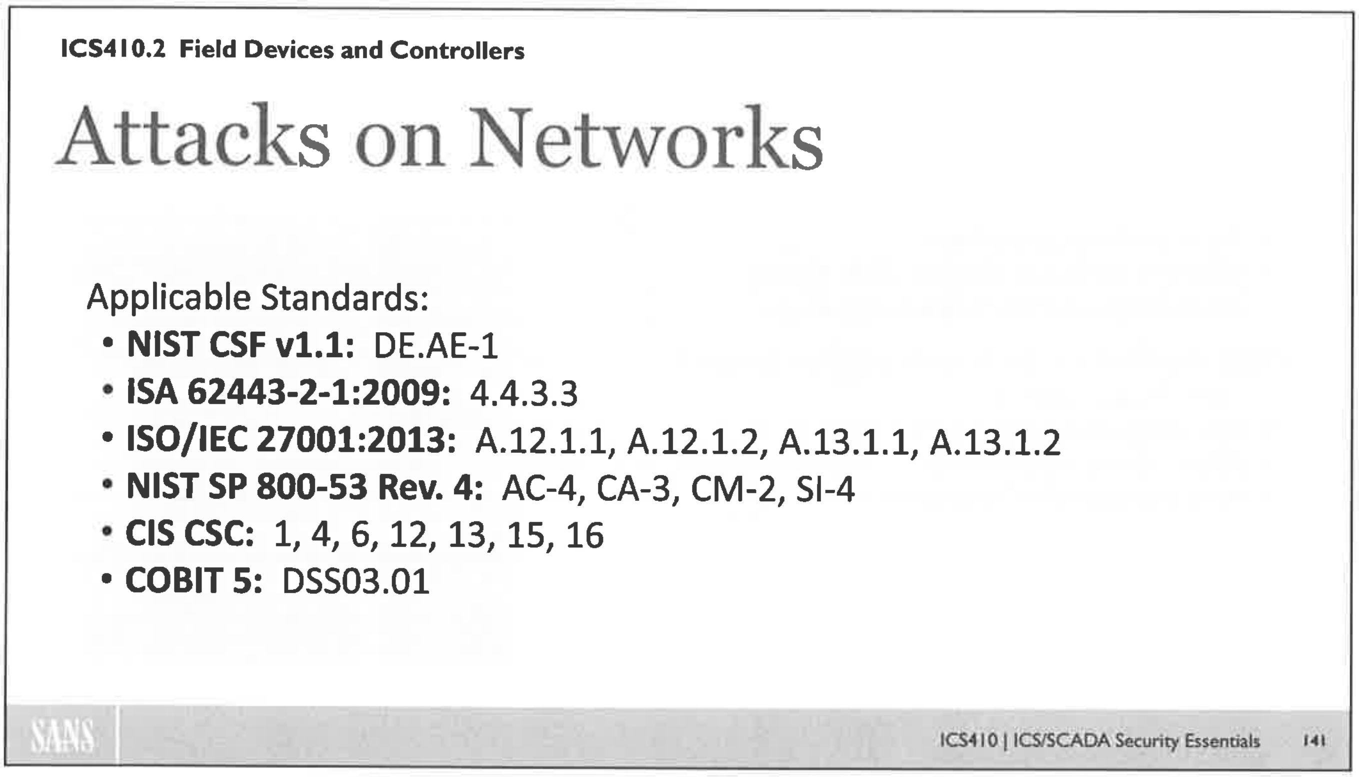
Day 5: ICS Security Governance

1. Introduction
2. ICS Attack Surface
   - Threat Actors and Reasons for Attack
   - Attack Surface and Inputs
   - Vulnerabilities
   - Threat/Attack Models
3. Purdue Level 0 and 1
   - Purdue Level 0 and 1 Attacks
   - Control Things Platform
   - **Exercise 2.1: Finding Passwords in EEPROM Dumps**
   - Purdue Level 0 and 1 Technologies
   - Fieldbus Protocol Families
   - **Exercise 2.2: Exploring Fieldbus Protocols**
   - Purdue Level 0 and 1 Defenses
4. Ethernet and TCP/IP
   - Ethernet Concepts
   - TCP/IP Concepts
   - **Exercise 2.3: Network Capture Analysis**
   - ICS Protocols over TCP/IP
   - Wireshark and ICS Protocols
   - Attacks on Networks
   - **Exercise 2.4: Enumerating Modbus TCP**

SANS ICS410 | ICS/SCADA Security Essentials 140

This page intentionally left blank.

© 2019, Justin Searle



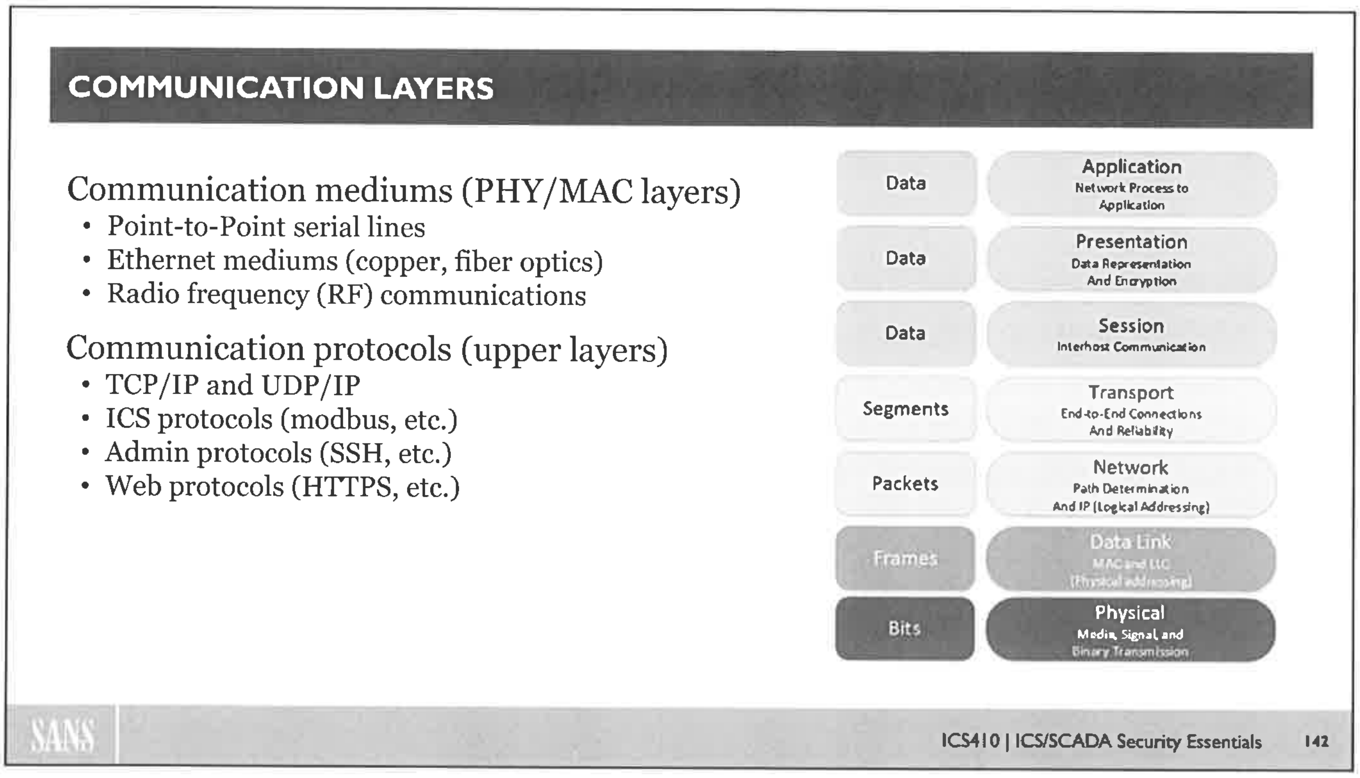
Attacks on Networks

Applicable Standards:

- **NIST CSF v1.1:** DE.AE-1
- **ISA 62443-2-1:2009:** 4.4.3.3
- **ISO/IEC 27001:2013:** A.12.1.1, A.12.1.2, A.13.1.1, A.13.1.2
- **NIST SP 800-53 Rev. 4:** AC-4, CA-3, CM-2, SI-4
- **CIS CSC:** 1, 4, 6, 12, 13, 15, 16
- **COBIT 5:** DSS03.01

This page intentionally left blank.

© 2019, Justin Searle



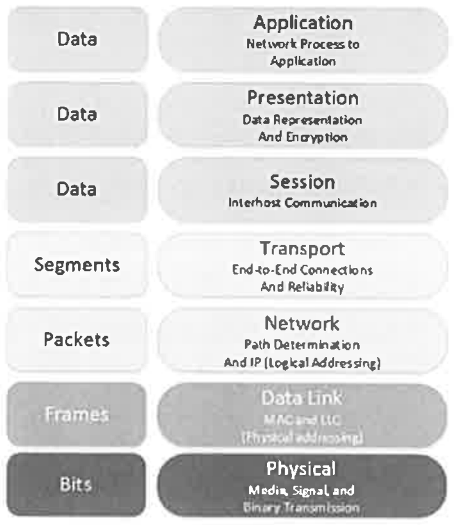
COMMUNICATION LAYERS

Communication mediums (PHY/MAC layers)

- Point-to-Point serial lines
- Ethernet mediums (copper, fiber optics)
- Radio frequency (RF) communications

Communication protocols (upper layers)

- TCP/IP and UDP/IP
- ICS protocols (modbus, etc.)
- Admin protocols (SSH, etc.)
- Web protocols (HTTPS, etc.)

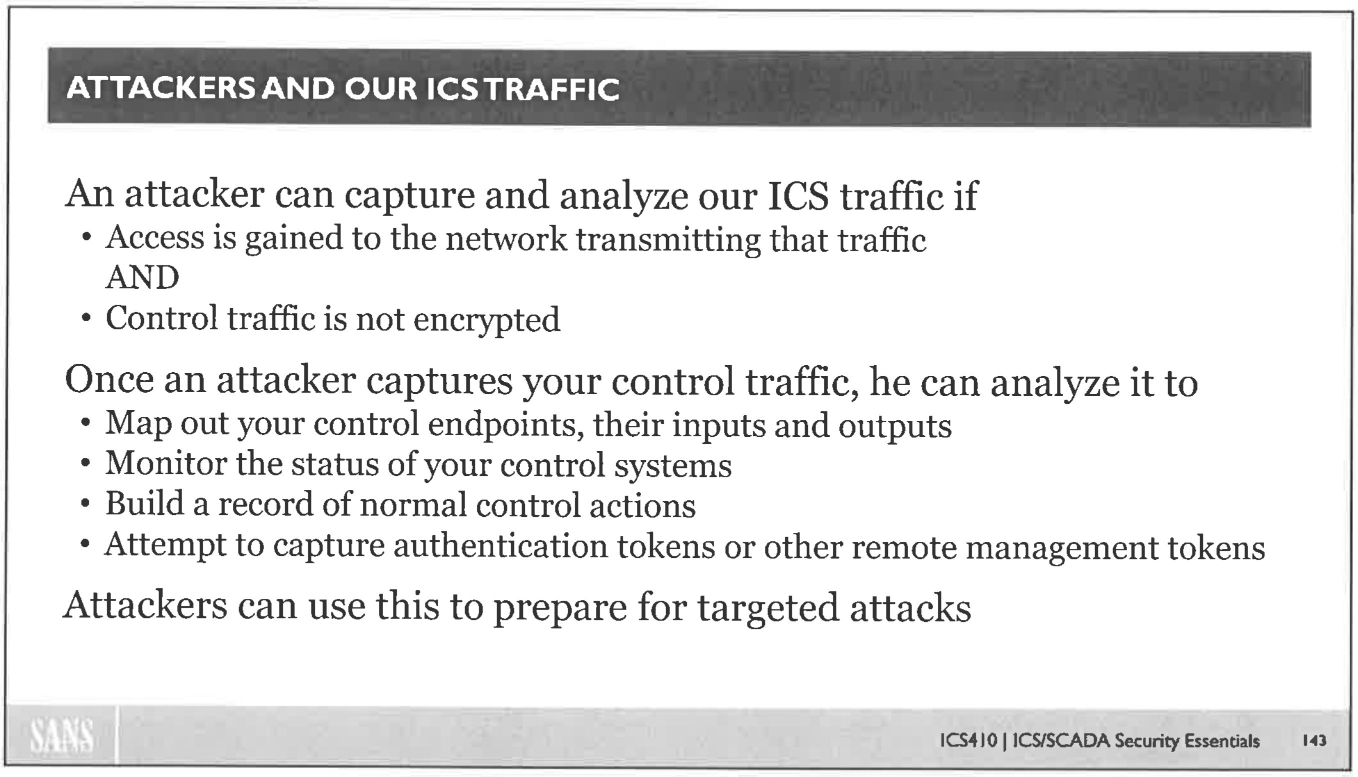


Data
Application
Network Process to Application
Data
Presentation
Data Representation And Encryption
Data
Session
Interhost Communication
Segments
Transport
End-to-End Connections And Reliability
Packets
Network
Path Determination And IP (Logical Addressing)
Frames
Data Link
MAC and LLC (Physical addressing)
Bits
Physical
Media, Signal, and Binary Transmission

SANS ICS410 | ICS/SCADA Security Essentials 142

Network communications can be very complex and often have multiple different layers. The Open System Interconnection (OSI) model defines seven different layers that network protocols can work on. The lower media layers, also called the PHY/MAC layers, are the lowest layer protocol that specifies the physical (PHY) media used, such as copper wire, fiber optics, or radio frequency (RF), and the media access control (MAC) or basic rules of how a device communicates on that medium. We generally refer to everything above the PHY/MAC layers as the upper layers, which consist of the network layer that handles addressing different devices, the transport layer that provides end-to-end connectivity with or without reliability, and what TCP/IP calls the application layer. The OSI model breaks this application layer into three parts: The session layer, the presentation layer, and the application layer. However, most TCP/IP protocols do not differentiate between those three layers because they usually handle all three functions.

© 2019, Justin Searle



ATTACKERS AND OUR ICS TRAFFIC

An attacker can capture and analyze our ICS traffic if

- Access is gained to the network transmitting that traffic
  AND
- Control traffic is not encrypted

Once an attacker captures your control traffic, he can analyze it to

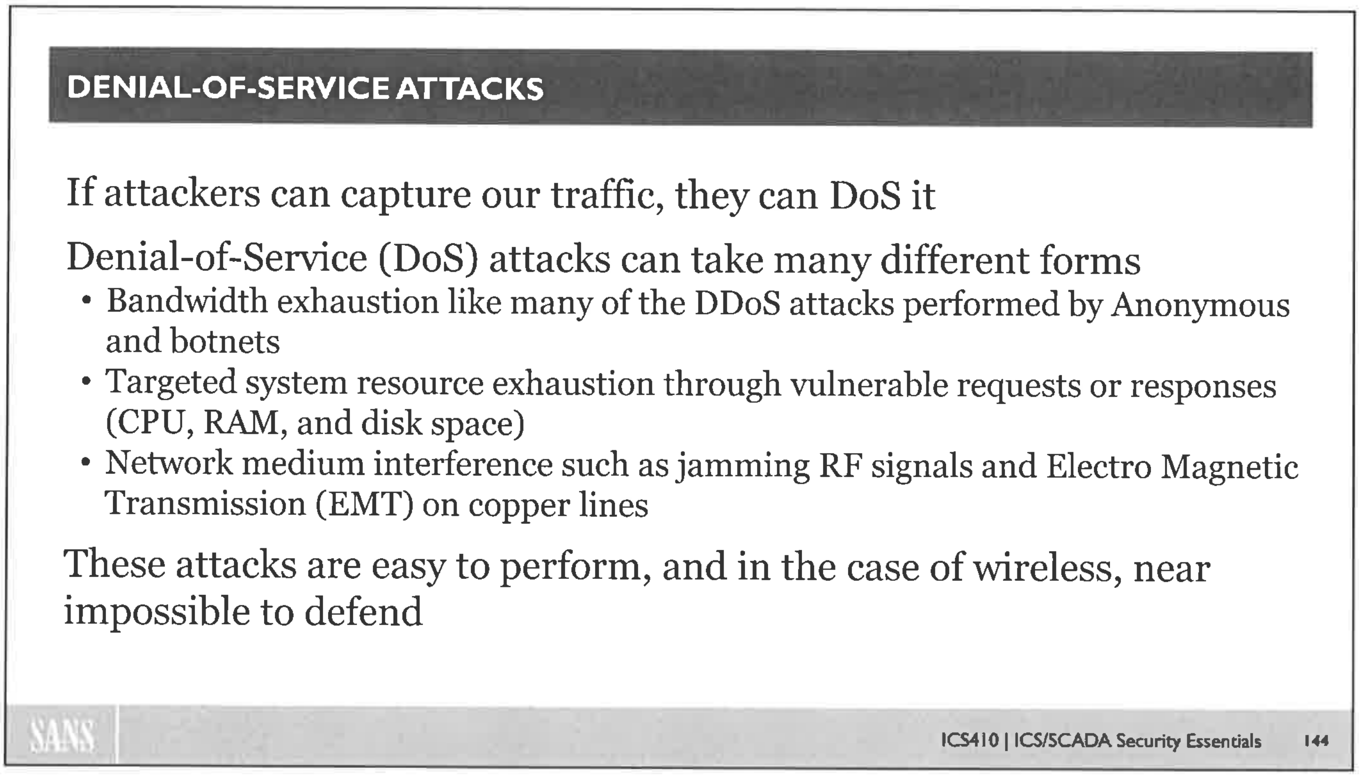
- Map out your control endpoints, their inputs and outputs
- Monitor the status of your control systems
- Build a record of normal control actions
- Attempt to capture authentication tokens or other remote management tokens

Attackers can use this to prepare for targeted attacks

If an attacker can gain access to the network medium, be it copper, fiber, or wireless, he can capture and analyze any network traffic passing across that wire. If that traffic is encrypted, the attacker must first decrypt the data to make it useful. However, that encryption helps only if there are no known weaknesses in its implementation, the attacker does not know the decryption key, or the attacker does not have control of one of the systems sending/receiving the encrypted traffic. Remember, encryption in a communications protocol is intended to protect the confidentiality of data between the two communication devices, not the data on those two devices.

Once an attacker captures your encrypted control traffic, he will usually analyze it to learn about the systems on the network and attempt to identify anything that may be of use to his attack. The attacker may map out the control endpoints of all your control systems, including all their assigned registers, coils, and endpoints. The attacker may decide to monitor the status of your control systems for a while to build a baseline of normal traffic and increase the odds of finding something useful like a control signal, a configuration update, or a control operator/engineer's traffic. The attacker may also build a record of all the normal control actions that your system receives so he can mimic the responses during an attack. The attacker may also attempt to capture authentication tokens or other remote management tokens. Attackers can use this to prepare for targeted attacks using techniques on the following pages.

© 2019, Justin Searle



DENIAL-OF-SERVICE ATTACKS

If attackers can capture our traffic, they can DoS it

Denial-of-Service (DoS) attacks can take many different forms

- Bandwidth exhaustion like many of the DDoS attacks performed by Anonymous and botnets
- Targeted system resource exhaustion through vulnerable requests or responses (CPU, RAM, and disk space)
- Network medium interference such as jamming RF signals and Electro Magnetic Transmission (EMT) on copper lines

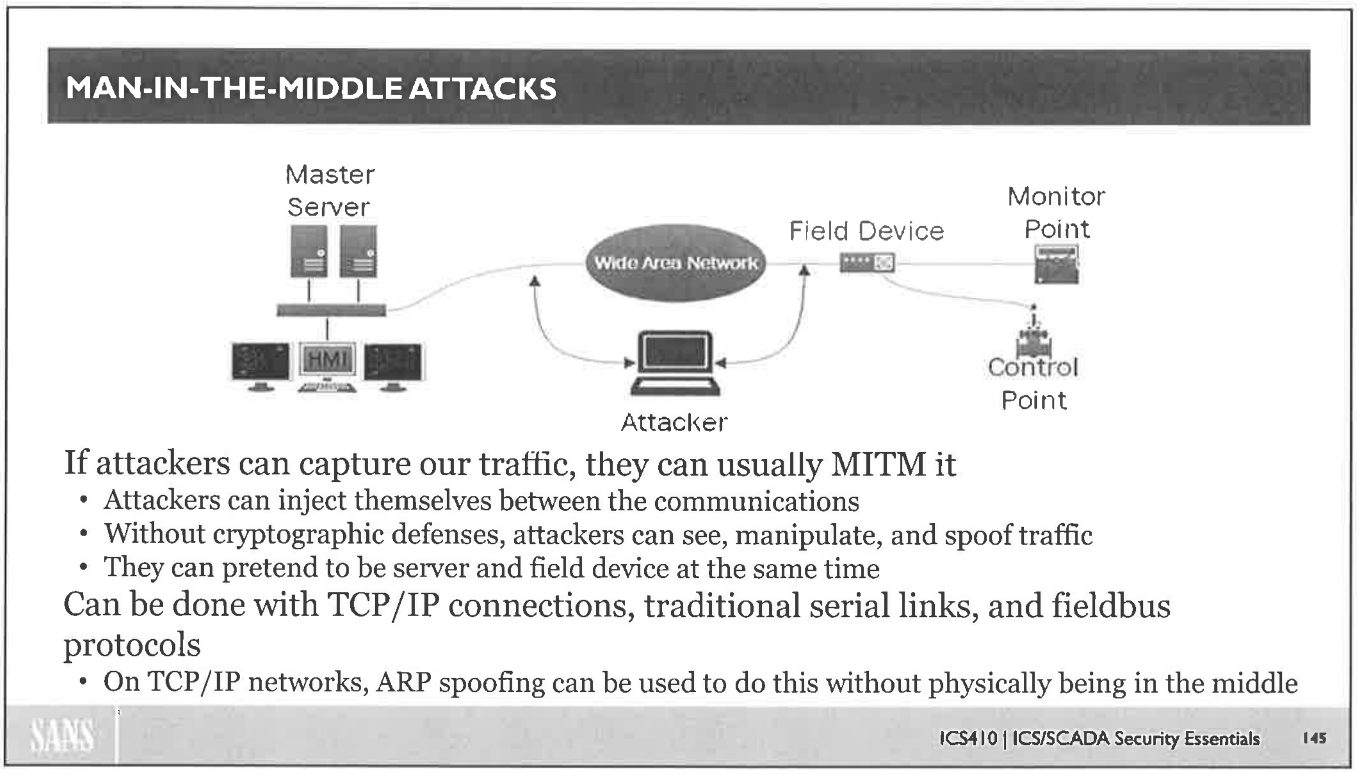
These attacks are easy to perform, and in the case of wireless, near impossible to defend

SANS ICS410 | ICS/SCADA Security Essentials 144

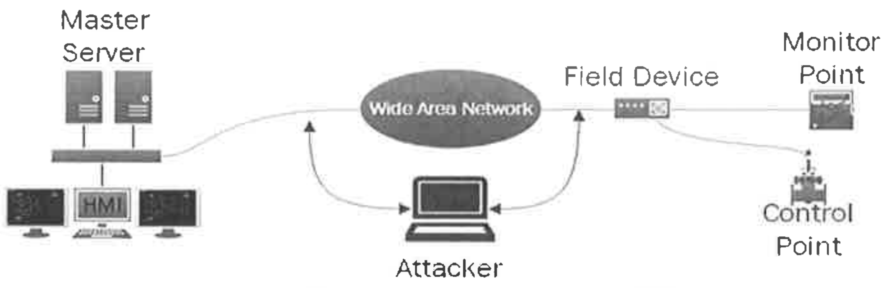
Denial-of-Service attacks can take many different forms but always focus on the exhaustion of some resource. Bandwidth exhaustion targets network bandwidth of the interconnected network devices and communication lines themselves, like many of the DDoS attacks performed by Anonymous and botnets. Exhaustion of CPU, RAM, and disk space of targeted systems can be another technique if the attacker finds a handling flaw or even a process-intensive request or response. Network medium interference, such as jamming RF signals and Electro Magnetic Transmission (EMT) on copper lines, is another technique attackers can use. Denial-of-Service attacks are often easy to perform for that attacker but can be very difficult to block as a defender. In the case of wireless, they are near impossible to defend against.

Denial-of-Service attack vectors need to be discussed when evaluating specific architecture designs, like the implementation of VLANS to segment networks of differing trust and criticality. An adversary with access to the lower security network could perform a Denial-of-Service attack that overwhelms the switch infrastructure and, in turn, impacts the more secure VLANs.

© 2019, Justin Searle



MAN-IN-THE-MIDDLE ATTACKS



If attackers can capture our traffic, they can usually MITM it

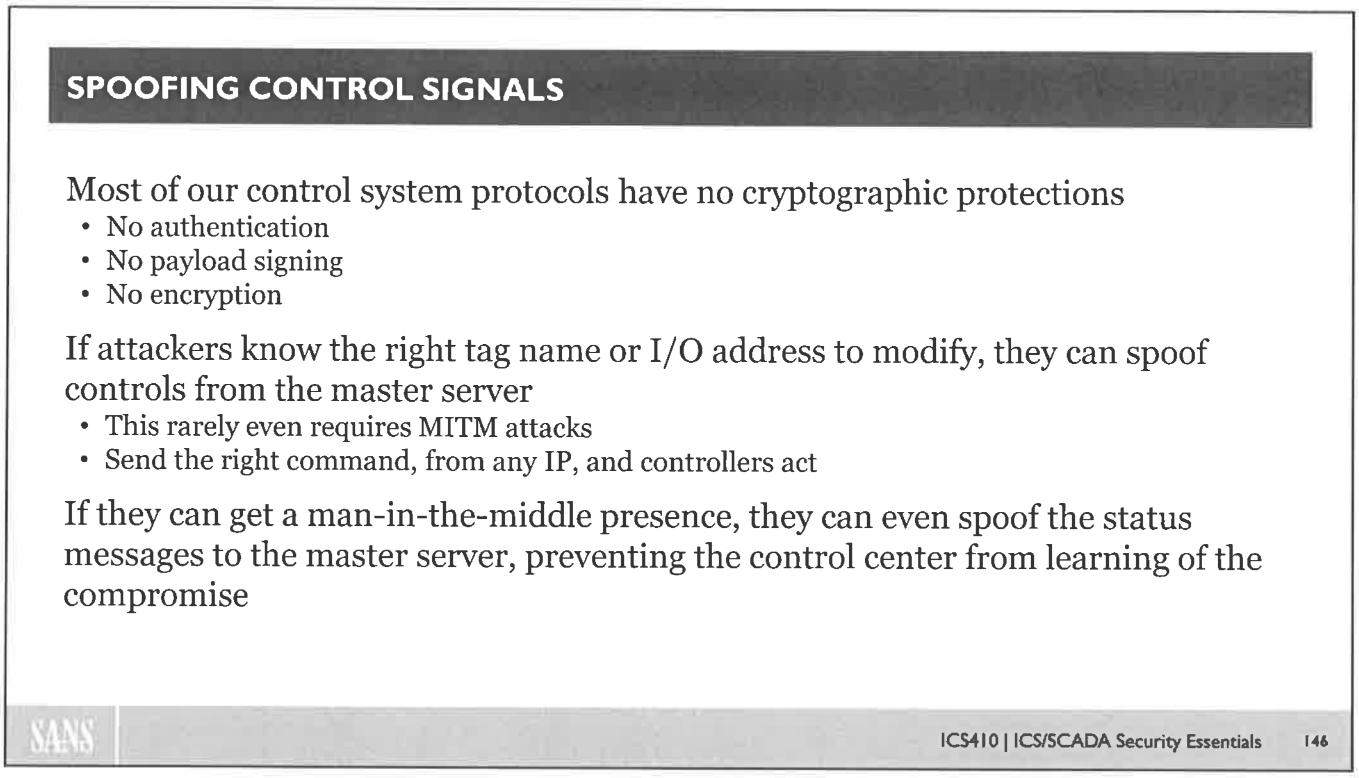
- Attackers can inject themselves between the communications
- Without cryptographic defenses, attackers can see, manipulate, and spoof traffic
- They can pretend to be server and field device at the same time

Can be done with TCP/IP connections, traditional serial links, and fieldbus protocols

- On TCP/IP networks, ARP spoofing can be used to do this without physically being in the middle

SANS ICS410 | ICS/SCADA Security Essentials 145

Attackers can inject themselves between the communications of any two devices if they can find access to the wired or wireless medium between those devices. Without cryptographic protections in the network protocol, attackers can usually see, manipulate, and spoof traffic to either device they intercept. This can work with TCP/IP connections as well as traditional serial links if the attacker has the right hardware and software. Using special network protocol attacks like ARP spoofing, the attacker doesn't need to be exactly between the two devices as he can leverage protocol weakness to reroute communications through the attacker's own machine, allowing him to see and manipulate that traffic.



SPOOFING CONTROL SIGNALS

Most of our control system protocols have no cryptographic protections

- No authentication
- No payload signing
- No encryption

If attackers know the right tag name or I/O address to modify, they can spoof controls from the master server

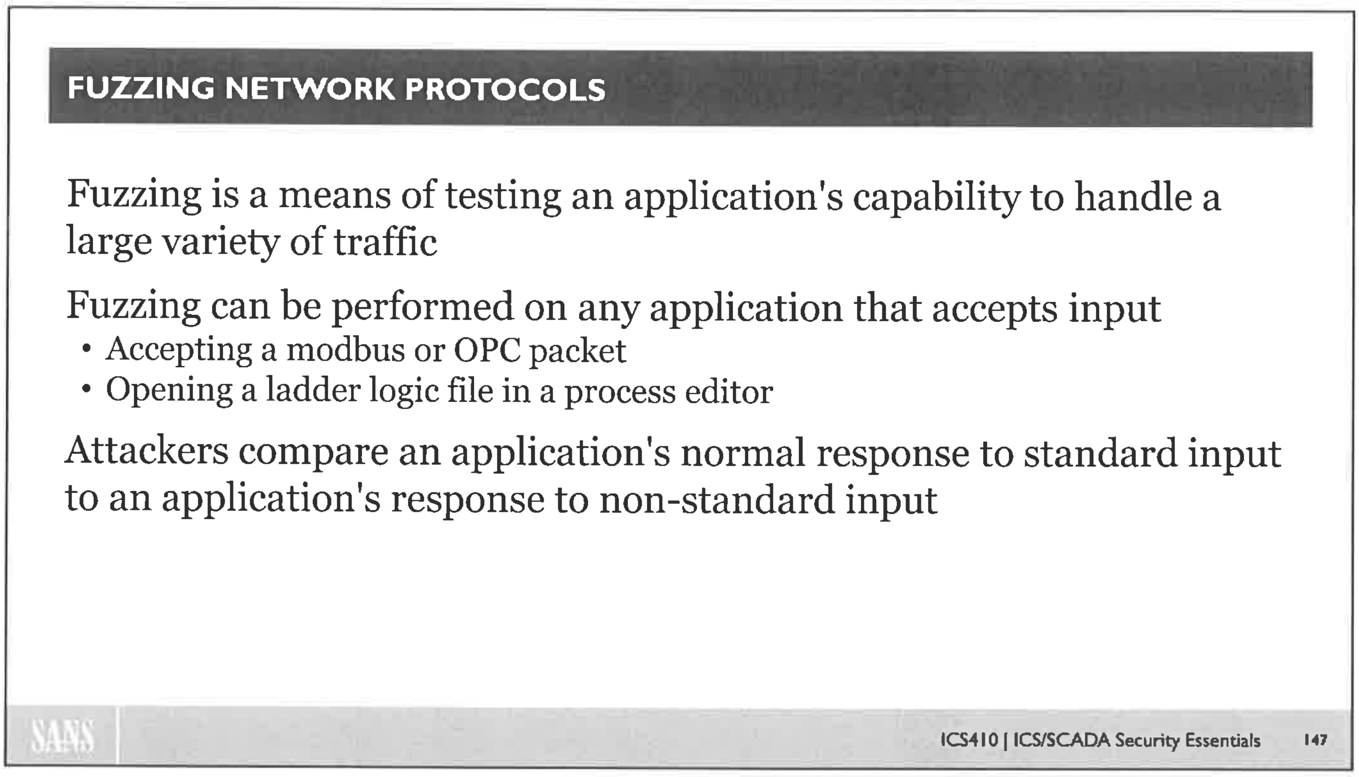
- This rarely even requires MITM attacks
- Send the right command, from any IP, and controllers act

If they can get a man-in-the-middle presence, they can even spoof the status messages to the master server, preventing the control center from learning of the compromise

ICS410 | ICS/SCADA Security Essentials 146

Many of our control system protocols come from old serial protocols that didn't need the security protocols we need today. When these protocols were migrated to work on Ethernet and TCP/IP networks, these protocols focused on minimal changes in an attempt to maintain backward compatibility. Take, for instance, modbus. It does not provide cryptographic protections or even authentication capabilities. It simply sends a command to read or write to a register or coil. If attackers know the right register value to modify, they can spoof controls from the master server and take manual control of the process. If they can get a man-in-the-middle presence, they can even spoof the status messages to the master server, preventing the control center from learning of the compromise and keeping them convinced that the system is operating normally.

© 2019, Justin Searle



FUZZING NETWORK PROTOCOLS

Fuzzing is a means of testing an application's capability to handle a large variety of traffic

Fuzzing can be performed on any application that accepts input
- Accepting a modbus or OPC packet
- Opening a ladder logic file in a process editor

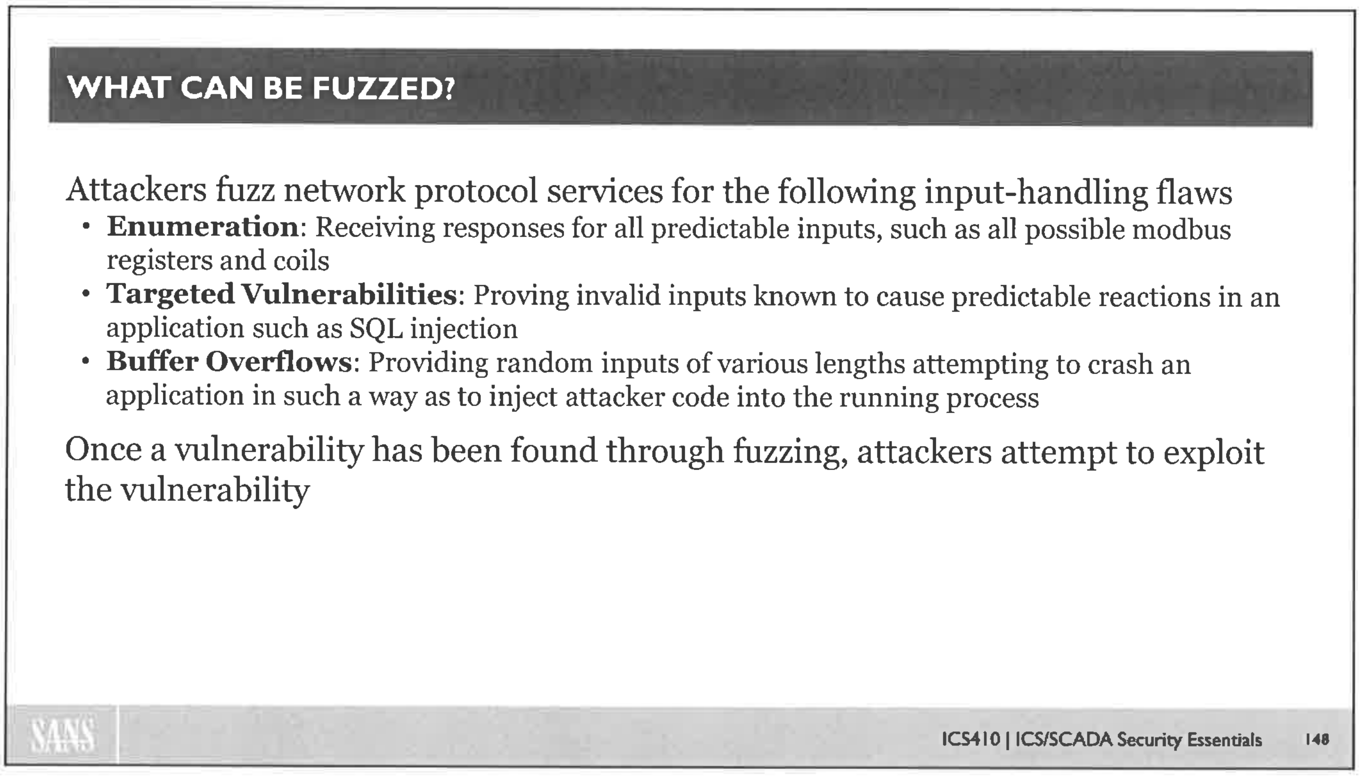
Attackers compare an application's normal response to standard input to an application's response to non-standard input

SANS | ICS410 | ICS/SCADA Security Essentials 147

Fuzzing is a technique attackers and penetration testers use to test an application's ability to handle a large variety of traffic, usually traffic not expected by the application. Fuzzing can be performed on any application that accepts input, simply by using a tool or a script to automatically send whatever data you want through the input. Some examples might be:

- Opening a ladder logic file in a process editor
- Accepting a modbus or OPC packet

Attackers compare an application's normal response to standard input to an application's response to non-standard fused input to determine any anomalies in the application's behavior. Once an anomaly has been found, the attacker attempts to determine if the anomaly is a sign of a vulnerability. For additional clarification and distinction on terms, consider penetration tests as focusing on exploiting vulnerabilities and vulnerability assessments as identifying vulnerabilities. In addition, when discussing penetration testing and fuzzing, the industry may reference two terms: White box testing and black box testing. White box testing takes the approach where the penetration tester is provided as much detail about the target as possible, while black box penetration testing provides little to no information to the tester regarding the target.



WHAT CAN BE FUZZED?

Attackers fuzz network protocol services for the following input-handling flaws

- **Enumeration**: Receiving responses for all predictable inputs, such as all possible modbus registers and coils
- **Targeted Vulnerabilities**: Proving invalid inputs known to cause predictable reactions in an application such as SQL injection
- **Buffer Overflows**: Providing random inputs of various lengths attempting to crash an application in such a way as to inject attacker code into the running process

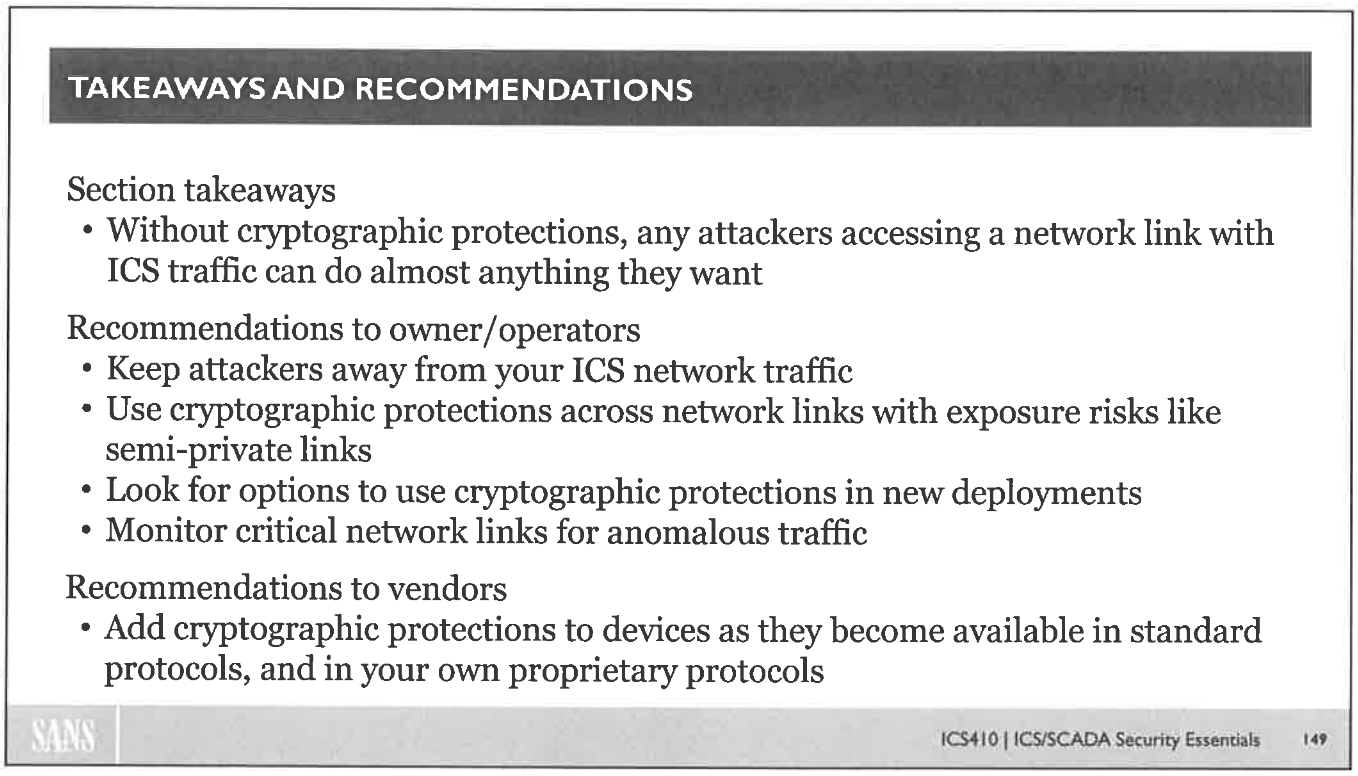
Once a vulnerability has been found through fuzzing, attackers attempt to exploit the vulnerability

Attackers will often fuzz network protocol services for the following input-handling flaws:

- **Enumeration**: Receiving responses for all predictable inputs, such as all possible modbus registers and coils. This allows the attacker to identify and track the values of each register and coil over a given period of time.
- **Targeted Vulnerabilities**: Proving invalid inputs known to cause predictable reactions in an application such as SQL injection. There are a number of different vulnerabilities that can be discovered using this method. In fact, most of the web-based vulnerabilities we discussed earlier today are found using this method. To do this, the attacker must use a special input file with known attack strings for the specific vulnerability they are searching for.
- **Buffer Overflows**: Providing random inputs of various lengths attempting to crash an application in such a way as to inject attacker code into the running process. This is similar to the targeted vulnerabilities above; however, this usually uses random input instead of a predefined list of attacks, and the result we try to identify is a crash in the network service, which could mean the application lost track of its instruction order in RAM. If this is the case, attackers may be able to control this crash behavior to run code of their choosing.

Once a vulnerability has been found through fuzzing, attackers attempt to exploit the vulnerability and gain a foothold in the system. After that foothold is gained, they will usually continue to look for additional vulnerabilities so they can pivot from one exploit to the next, digging deeper and deeper into our control networks.

© 2019, Justin Searle



TAKEAWAYS AND RECOMMENDATIONS

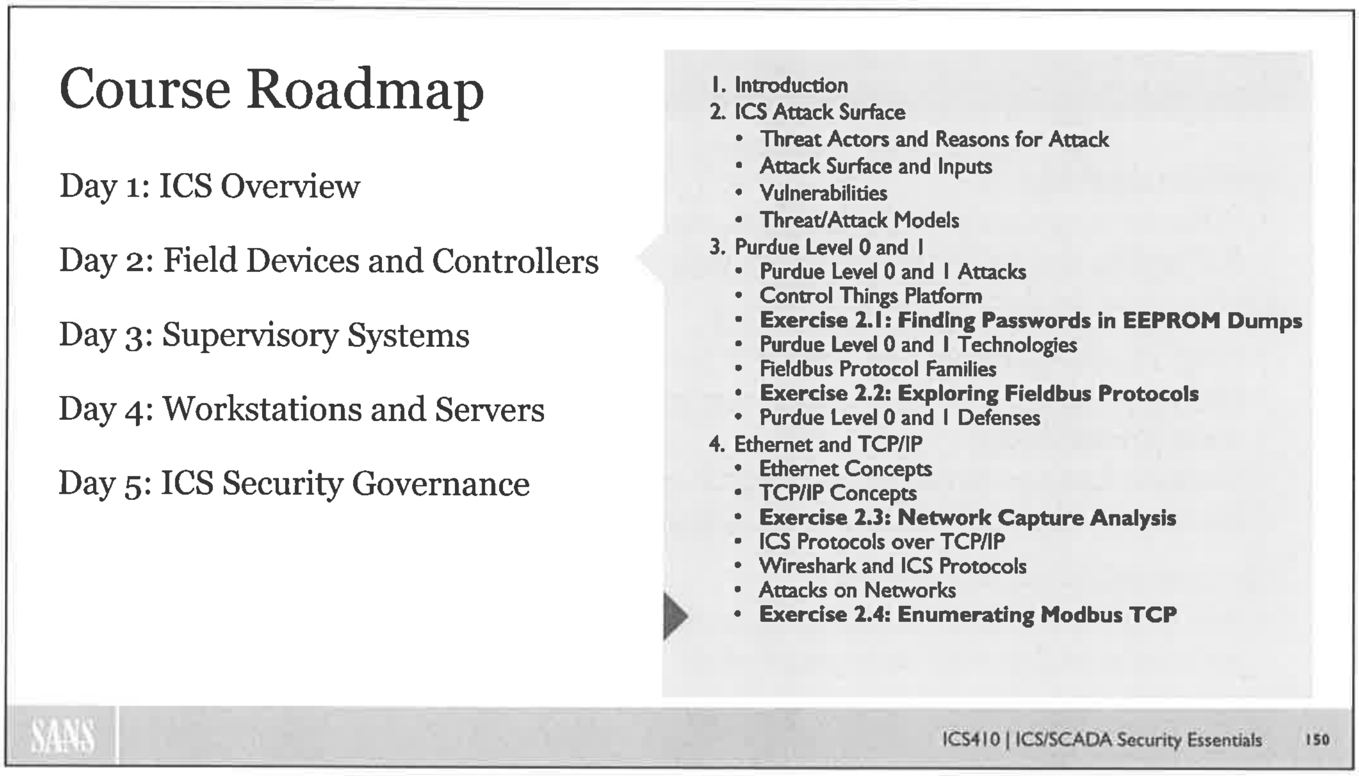
Section takeaways
- Without cryptographic protections, any attackers accessing a network link with ICS traffic can do almost anything they want

Recommendations to owner/operators
- Keep attackers away from your ICS network traffic
- Use cryptographic protections across network links with exposure risks like semi-private links
- Look for options to use cryptographic protections in new deployments
- Monitor critical network links for anomalous traffic

Recommendations to vendors
- Add cryptographic protections to devices as they become available in standard protocols, and in your own proprietary protocols

SANS ICS410 | ICS/SCADA Security Essentials 149

This page intentionally left blank.



Course Roadmap

Day 1: ICS Overview

Day 2: Field Devices and Controllers

Day 3: Supervisory Systems

Day 4: Workstations and Servers

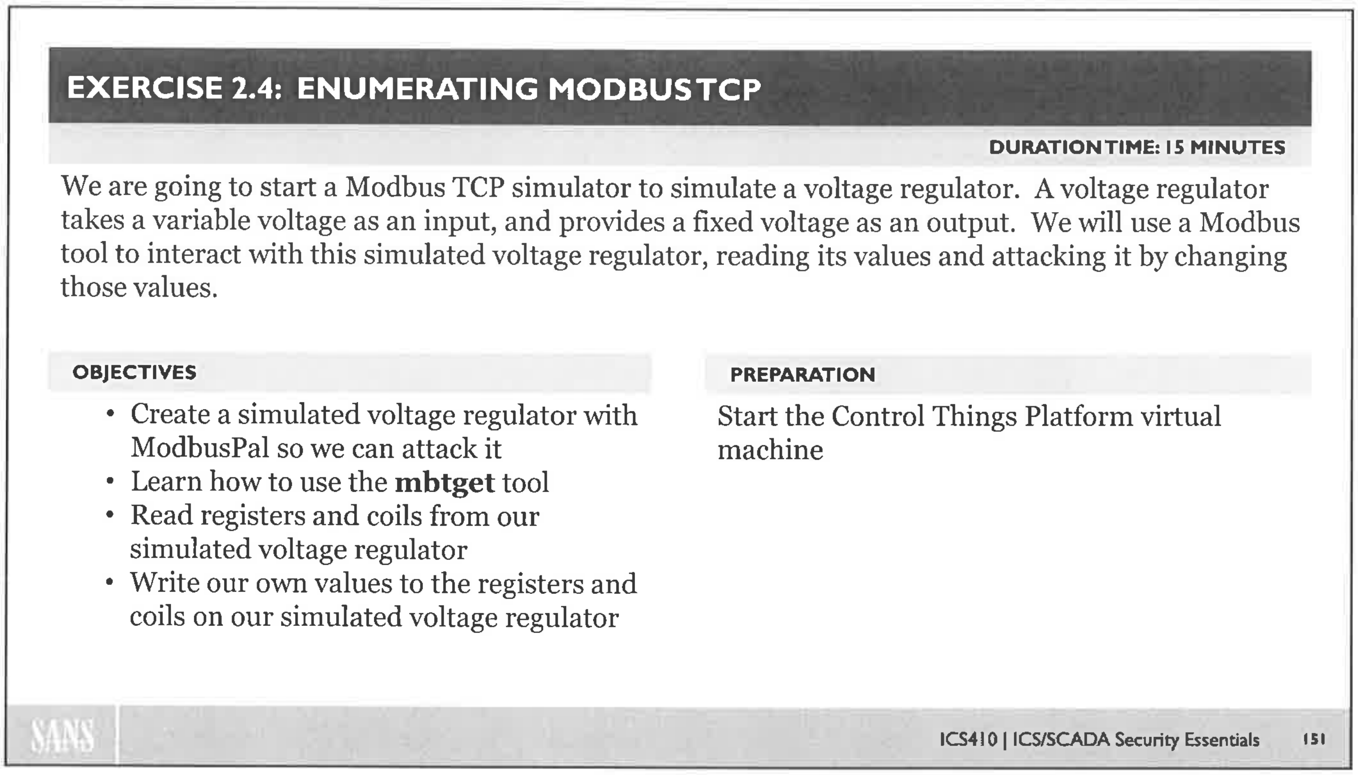
Day 5: ICS Security Governance

1. Introduction
2. ICS Attack Surface
   - Threat Actors and Reasons for Attack
   - Attack Surface and Inputs
   - Vulnerabilities
   - Threat/Attack Models
3. Purdue Level 0 and 1
   - Purdue Level 0 and 1 Attacks
   - Control Things Platform
   - **Exercise 2.1: Finding Passwords in EEPROM Dumps**
   - Purdue Level 0 and 1 Technologies
   - Fieldbus Protocol Families
   - **Exercise 2.2: Exploring Fieldbus Protocols**
   - Purdue Level 0 and 1 Defenses
4. Ethernet and TCP/IP
   - Ethernet Concepts
   - TCP/IP Concepts
   - **Exercise 2.3: Network Capture Analysis**
   - ICS Protocols over TCP/IP
   - Wireshark and ICS Protocols
   - Attacks on Networks
   - **Exercise 2.4: Enumerating Modbus TCP**

SANS ICS410 | ICS/SCADA Security Essentials 150

This page intentionally left blank.

© 2019, Justin Searle



EXERCISE 2.4: ENUMERATING MODBUS TCP

DURATION TIME: 15 MINUTES

We are going to start a Modbus TCP simulator to simulate a voltage regulator. A voltage regulator takes a variable voltage as an input, and provides a fixed voltage as an output. We will use a Modbus tool to interact with this simulated voltage regulator, reading its values and attacking it by changing those values.

OBJECTIVES

- Create a simulated voltage regulator with ModbusPal so we can attack it
- Learn how to use the **mbtget** tool
- Read registers and coils from our simulated voltage regulator
- Write our own values to the registers and coils on our simulated voltage regulator

PREPARATION

Start the Control Things Platform virtual machine

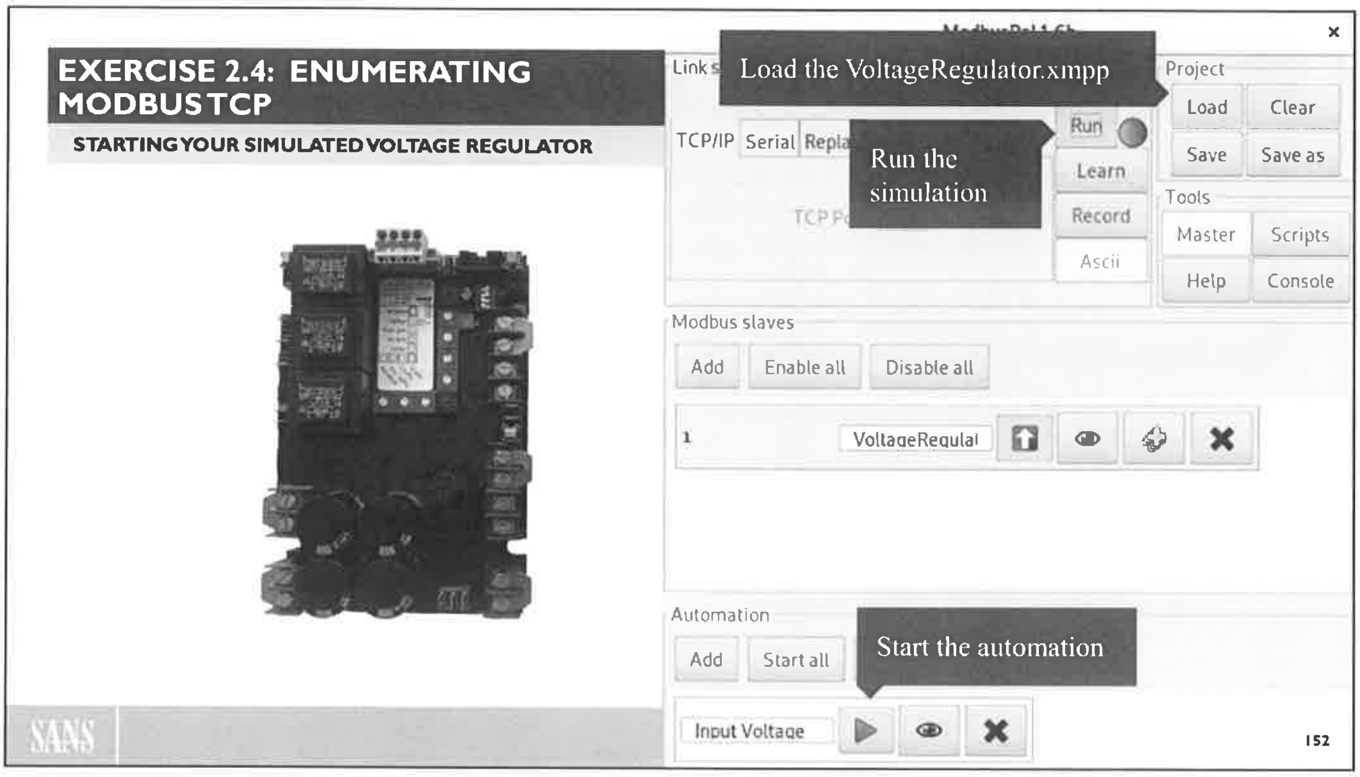
SANS

ICS410 | ICS/SCADA Security Essentials 151

For this lab, we are going to set up a Modbus field device. The software we will be using to simulate the field device is called ModbusPal. It is a Java application that you could run on your own computer, regardless of your operating system. For this lab, we will use the preinstalled version in Control Things Platform. Unfortunately, the ModbusPal feature to run saved configurations is broken, so we will need to set it up by hand. We will be simulating a very simple transformer that might be found in an electric utility's substation. Real transformers could have hundreds of Modbus data points, but for the sake of simplicity, we will set up only a couple of Modbus data points.

ModbusPal can be confusing the first time you use it. Because of this, we will show you screenshots of each action you need to take. Each screenshot has text bubbles telling you what to type or click. Think of this as the same as reading a comic book or manga. Start with the text box closest to the top of the screenshot and work your way down, performing the indicated action. The notes below each slide will mimic the instructions but will occasionally provide additional information or explanation.

If you have any questions, please ask your instructor, your class TA, or your course moderator. They should be able to help resolve any confusion with using this tool.



EXERCISE 2.4: ENUMERATING MODBUS TCP

STARTING YOUR SIMULATED VOLTAGE REGULATOR

We have created a voltage regulator for you that uses the Modbus TCP control protocol. To run it, we will need to start the ModbusPal program in Control Things. Click on the Activities main menu and then search for modbuspal to start. Once modbuspal is running, you will need to load the simulation, which should automatically open to the right folder when you click **Load**. Select VoltageRegulator.xmpp in that folder. Just in case something happens and you don't see that file, the full path to the file is:

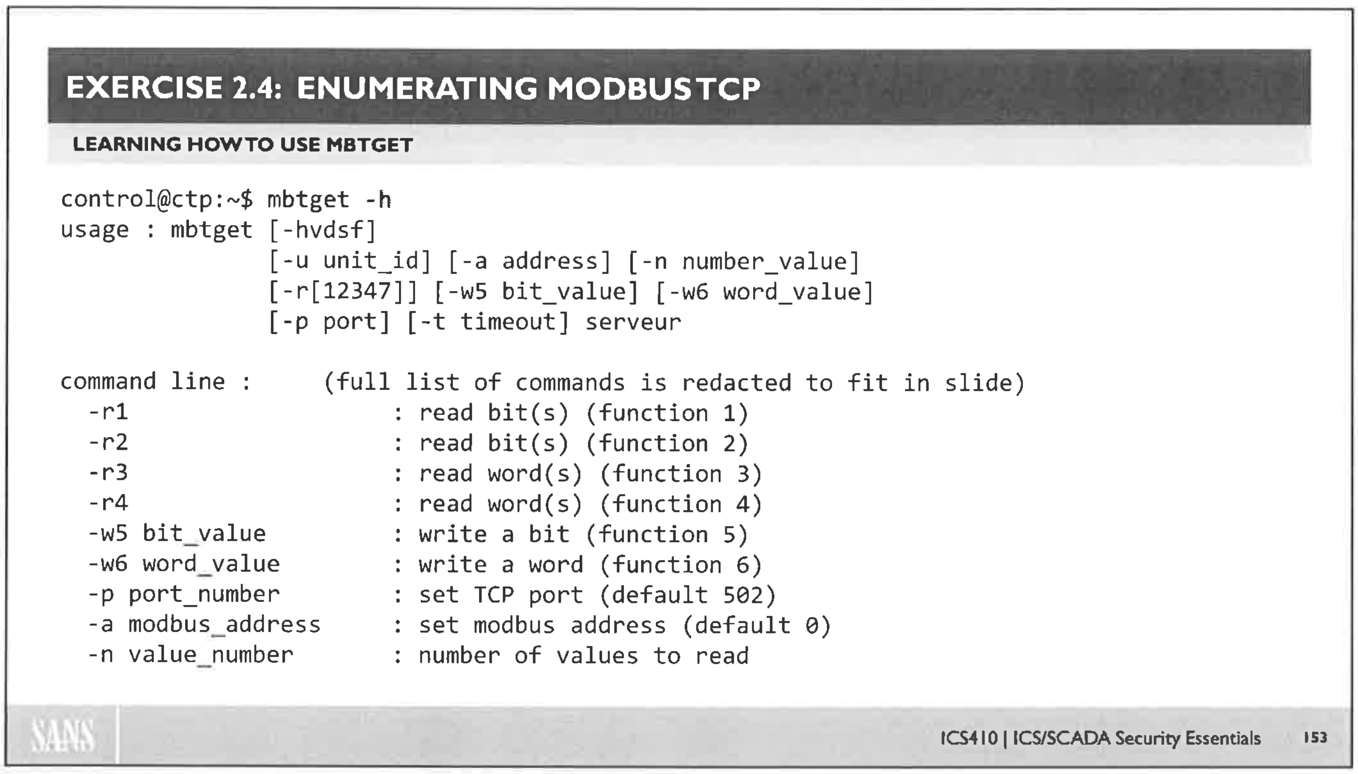
/home/control/Simulators/ModbusPal/VoltageRegulator.xmpp

Once that project is loaded, click **Run** and then start the automation by clicking the **green triangle play button** under the Automation section.

This simulator is configured to run on TCP port 10502 instead of the standard TCP 502 which Modbus normally runs on. It is also configured to listen on 127.0.0.1 or your localhost address.

Image Source:
https://en.wikipedia.org/wiki/Voltage_regulator#/media/File:EMRI_LXCOS_Voltage_Regulator.jpg

© 2019, Justin Searle


EXERCISE 2.4: ENUMERATING MODBUS TCP

LEARNING HOW TO USE MBTGET

```
control@ctp:~$ mbtget -h
usage : mbtget [-hvdsf]
               [-u unit_id] [-a address] [-n number_value]
               [-r[12347]] [-w5 bit_value] [-w6 word_value]
               [-p port] [-t timeout] serveur

command line :     (full list of commands is redacted to fit in slide)
  -r1                  : read bit(s) (function 1)
  -r2                  : read bit(s) (function 2)
  -r3                  : read word(s) (function 3)
  -r4                  : read word(s) (function 4)
  -w5 bit_value        : write a bit (function 5)
  -w6 word_value       : write a word (function 6)
  -p port_number       : set TCP port (default 502)
  -a modbus_address    : set modbus address (default 0)
  -n value_number      : number of values to read
```

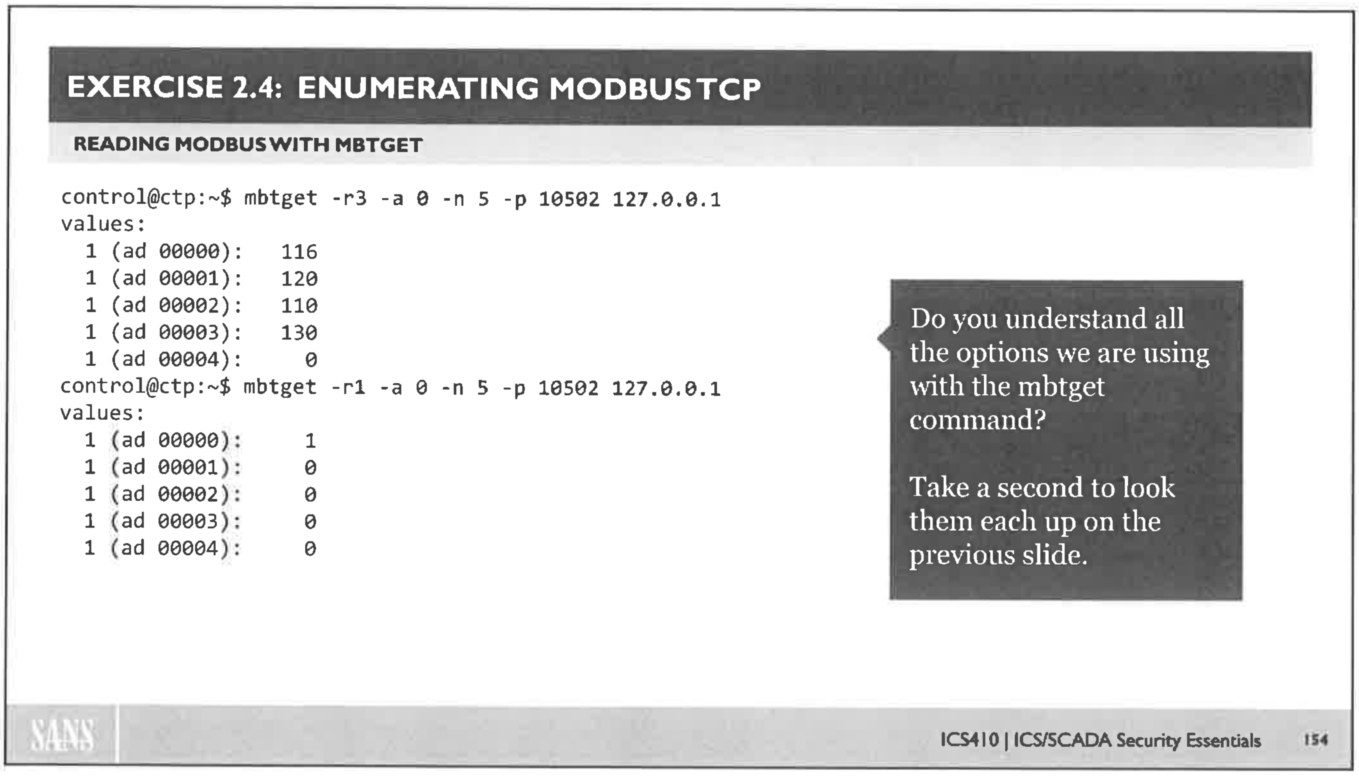
SANS ICS410 | ICS/SCADA Security Essentials 153

Click on the "Activities" link in the upper left corner of the screen to get to the main menu, and then click on the "Terminal" icon that looks like a black box on the bottom left. Once Terminal is open, type **mbtget –h** to see the list of options available.

We will be using a command-line tool called `mbtget` to interact with the Modbus TCP protocol, one of the most commonly used TCP/IP-based ICS protocols out there. Using a tool like mbtget, an attacker can easily create modbus commands to change the ICS process running: Causing it to stop, or worse, causing the ICS process to be modified in such a way as to cause maximum impact to the company or engineers running the process. For attackers to learn what modbus commands to send, an attacker can use tcpdump or Wireshark to help them understand more about the process. We will use Wireshark in this manner in a later lab so you can see how this is possible.

Because Modbus TCP is such a simple protocol, another way you could generate modbus commands is a generic tool like Netcat. It is a Windows and Linux tool that allows reading and writing to network connections via TCP or UDP. Netcat, often referenced as nc, can be used for a number of items, including port scanning, port forwarding, port listening, and as a backdoor to a system.

When communicating with modbus, we'll always want to specify an address (`-a`) and one of the various read or write commands. In cases where a modbus device has more than one slave or unit_id, you'll want to specify that as well (`-u`), but in our case, our VoltageRegulator has only a single unit_id of 0, which is implied by default.



EXERCISE 2.4: ENUMERATING MODBUS TCP

READING MODBUS WITH MBTGET

```
control@ctp:~$ mbtget -r3 -a 0 -n 5 -p 10502 127.0.0.1
values:
  1 (ad 00000):   116
  1 (ad 00001):   120
  1 (ad 00002):   110
  1 (ad 00003):   130
  1 (ad 00004):     0
control@ctp:~$ mbtget -r1 -a 0 -n 5 -p 10502 127.0.0.1
values:
  1 (ad 00000):     1
  1 (ad 00001):     0
  1 (ad 00002):     0
  1 (ad 00003):     0
  1 (ad 00004):     0
```

Do you understand all the options we are using with the mbtget command?

Take a second to look them each up on the previous slide.

SANS ICS410 | ICS/SCADA Security Essentials 154

Now let's assume that our VoltageRegulator is part of a larger ICS and is being monitored and controlled by a Modbus master. We are going to play the role of an attacker and attempt to interact with this device directly via Modbus TCP. The first thing we want to do is try reading the registers and coils. We can do this with the commands in the screenshot above. Notice that the register and coil addresses are one number lower than they are in the ModbusPal configuration window. This is a very common issue, as many vendor configuration utilities start the addressing at address 1; however, Modbus TCP starts its address numbers at 0. When in doubt, double check.

Try sending a read request to read the first 5 holding registers (analog I/O) by typing:

```
mbtget -r3 -a 0 -n 5 -p 10502 127.0.0.1
```

If you get a **`server connect "127.0.0.1:10502" failed`** response, check to make sure you click the **Run** button in ModbusPal. If you see a **0** as the output of address 0, then click the **green triangle** play button under the **Automation** section.

Now try sending a read request to read the first 5 coils (digital I/O) by typing:

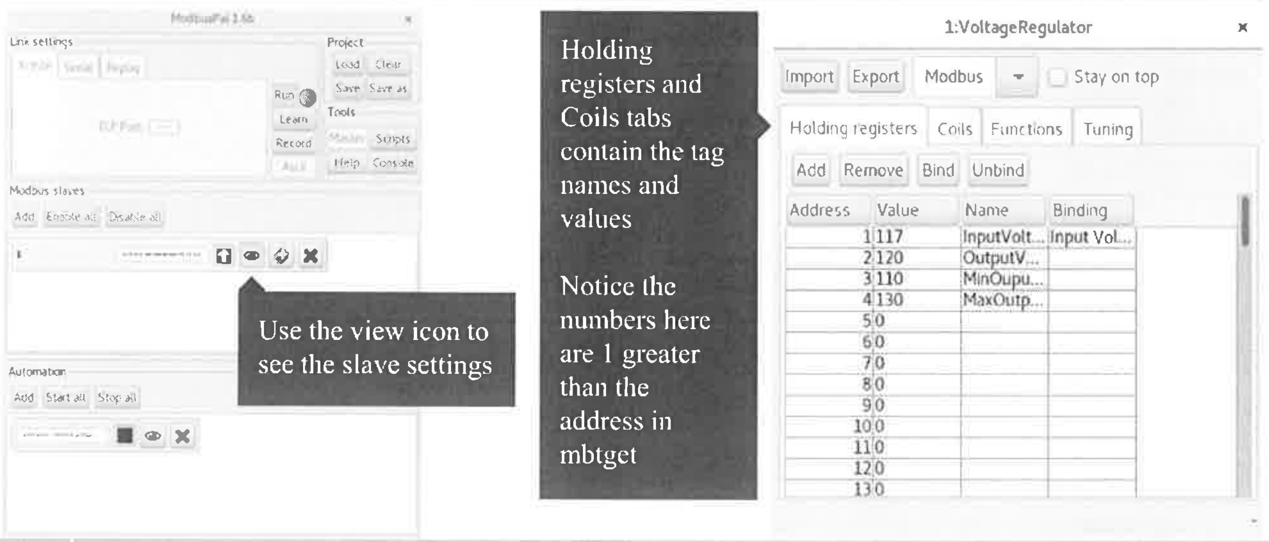
```
mbtget -r1 -a 0 -n 5 -p 10502 127.0.0.1
```

Try running these two commands a few more times to identify which values are changing and which values are not. Notice that there are no tag names for your registers and coils. If they are set in ModbusPal, they are not being retrieved by mbtget. This is because the Modbus TCP protocol does not support this feature. Actually, very few control protocols do if you remember our discussion on them. If you don't have documentation for a device that specifies what each register and coil represent, you are going to be guessing what each value means, which, in most cases, isn't very feasible. For this reason, attackers will look for device documentation or vendor configuration tools to identify what each value means. Of course, this assumes that the attacker is trying to perform some strategic action. If the attacker's whole goal is to cause damage, he could start randomly changing all the values, which could be catastrophic.

© 2019, Justin Searle

EXERCISE 2.4: ENUMERATING MODBUSTCP

VIEWING MODBUSPAL SETTINGS



Address	Value	Name	Binding
1	117	InputVolt...	Input Vol...
2	120	OutputV...	
3	110	MinOupu...	
4	130	MaxOutp...	
5	0		
6	0		
7	0		
8	0		
9	0		
10	0		
11	0		
12	0		
13	0		

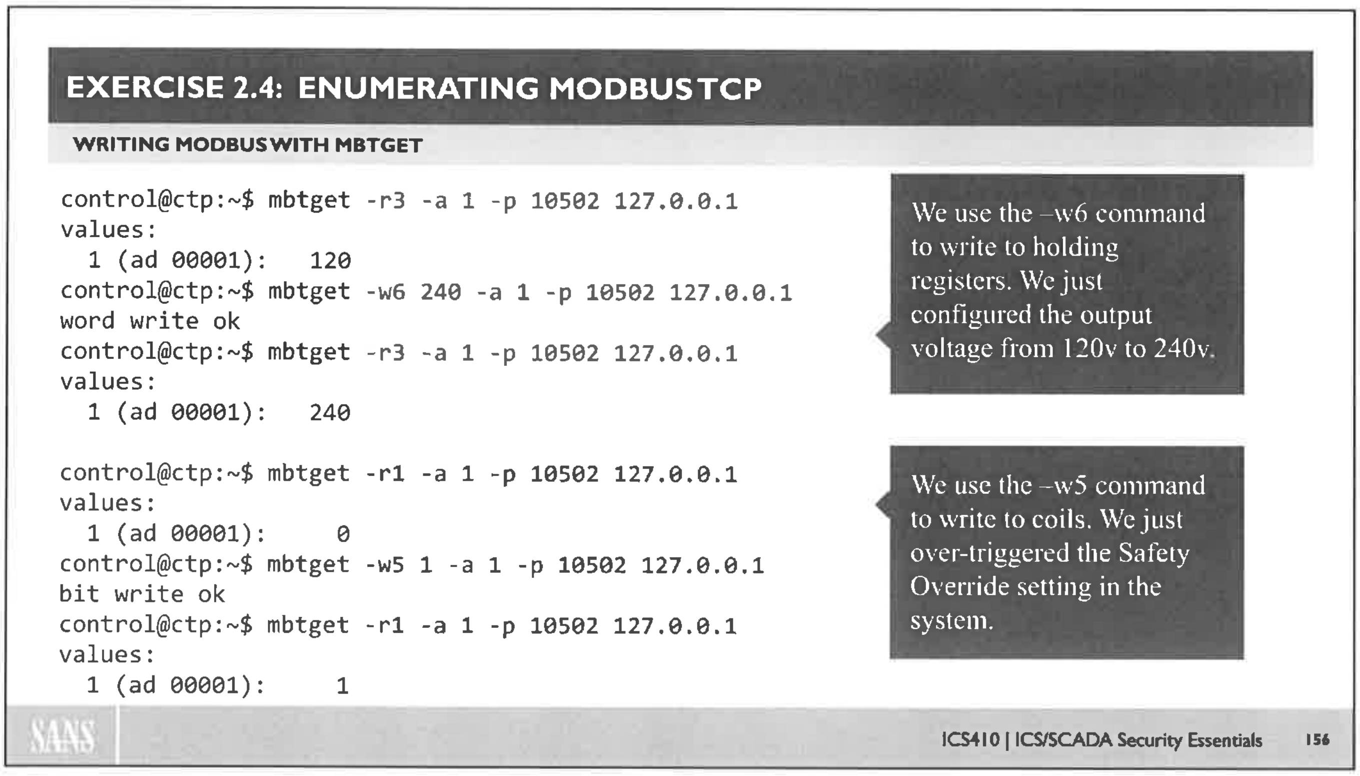
Holding registers and Coils tabs contain the tag names and values

Notice the numbers here are 1 greater than the address in mbtget

Use the view icon to see the slave settings

Since you know this is a voltage regulator, you can make some assumptions on what those registers and coils are, especially if I tell you it is configured to run in North America's 120v systems. Voltage regulators take a highly variable input voltage and fix it to a steady output voltage. What do you think these 5 registers and 5 coils do? Registers 0 and 1 are fairly easy to figure out, registers 2 and 3 are a bit harder, and register 4 is impossible to tell. Of course you'd have to add a 1 to each of those addresses when looking at ModbusPal since it starts the addresses at 1 where the actual protocol starts the addresses at 0. This is normal for Modbus and many of the older ICS protocols. As for the coils, they are all fairly impossible to tell since there are only two possible options.

Let's cheat and check out the ModbusPal configurations. In the Modbus Slaves section, click on the eye icon next to slave 1. Once again, I apologize for the orphaned tool and its non-readable text fields on this first screen. When you click on the eye, you will see a new window with a few tabs. Look under the "Holding registers" and "Coils" tabs to view the tag names and values for each register and coil.



EXERCISE 2.4: ENUMERATING MODBUS TCP

WRITING MODBUS WITH MBTGET

```
control@ctp:~$ mbtget -r3 -a 1 -p 10502 127.0.0.1
values:
  1 (ad 00001):   120
control@ctp:~$ mbtget -w6 240 -a 1 -p 10502 127.0.0.1
word write ok
control@ctp:~$ mbtget -r3 -a 1 -p 10502 127.0.0.1
values:
  1 (ad 00001):   240

control@ctp:~$ mbtget -r1 -a 1 -p 10502 127.0.0.1
values:
  1 (ad 00001):     0
control@ctp:~$ mbtget -w5 1 -a 1 -p 10502 127.0.0.1
bit write ok
control@ctp:~$ mbtget -r1 -a 1 -p 10502 127.0.0.1
values:
  1 (ad 00001):     1
```

We use the –w6 command to write to holding registers. We just configured the output voltage from 120v to 240v.

We use the –w5 command to write to coils. We just over-triggered the Safety Override setting in the system.

Because Modbus TCP doesn't support any authentication options, we can just as easily change any of the register and coil values as we can read them.

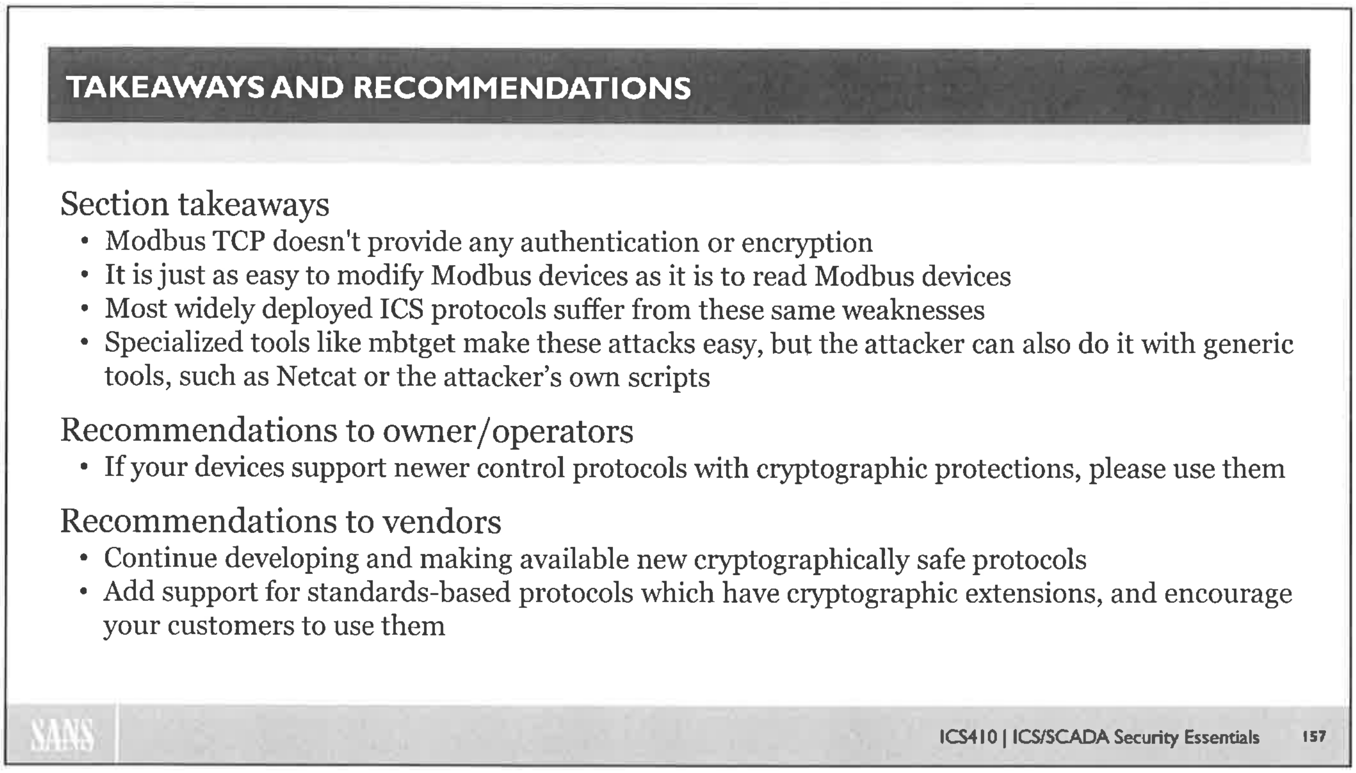
Verify the current value of register 2 with the command **`-r3 -a 1`** (this should still be set to 120), attempt to write a new value to register 2 with the command **`-w6 240 -a 1`** and then check the value of the register by reading it again with the command **`-r3 -a 1`** to verify you successfully changed it. This should now be set to 240; you can also click on the eye of the VoltageRegulator and see the values change. The full commands are shown above in the examples.

Now perform the same steps to the coil values. Verify the current value of coil 2 with the command **`-r1 -a 1`** (this should still be set to 0), attempt to write a new value to coil 2 with the command **`-w5 -a 1`** and check the value of the coil by reading it again with the command **`-r3 -a 1`** to verify you successfully changed it. This should now be set to 1. You can also click on the eye of the VoltageRegulator and then the coil tab to see the values change. The full commands are shown above in the examples.

If an attacker can figure out what each register and coil represent, he can create strategic scenarios to cause maximum damage to the system. For instance, the attacker could trigger the safety override system and change the output of the voltage regulator, possibly causing damage to devices receiving power from the voltage regulator. This is simplistic, yet similar to the type of payload we saw in Stuxnet on the Iranian nuclear facilities.

As a side note, if you haven't tried the -d option to mbtget, try any of the previous commands with it to see a raw dump of the Modbus TCP request and response. Can you remember the protocol breakdown of the Modbus TCP fields? The full Modbus standard is in the **Protocols/Modbus** folder on Control Things.

© 2019, Justin Searle



TAKEAWAYS AND RECOMMENDATIONS

Section takeaways

- Modbus TCP doesn't provide any authentication or encryption
- It is just as easy to modify Modbus devices as it is to read Modbus devices
- Most widely deployed ICS protocols suffer from these same weaknesses
- Specialized tools like mbtget make these attacks easy, but the attacker can also do it with generic tools, such as Netcat or the attacker's own scripts

Recommendations to owner/operators

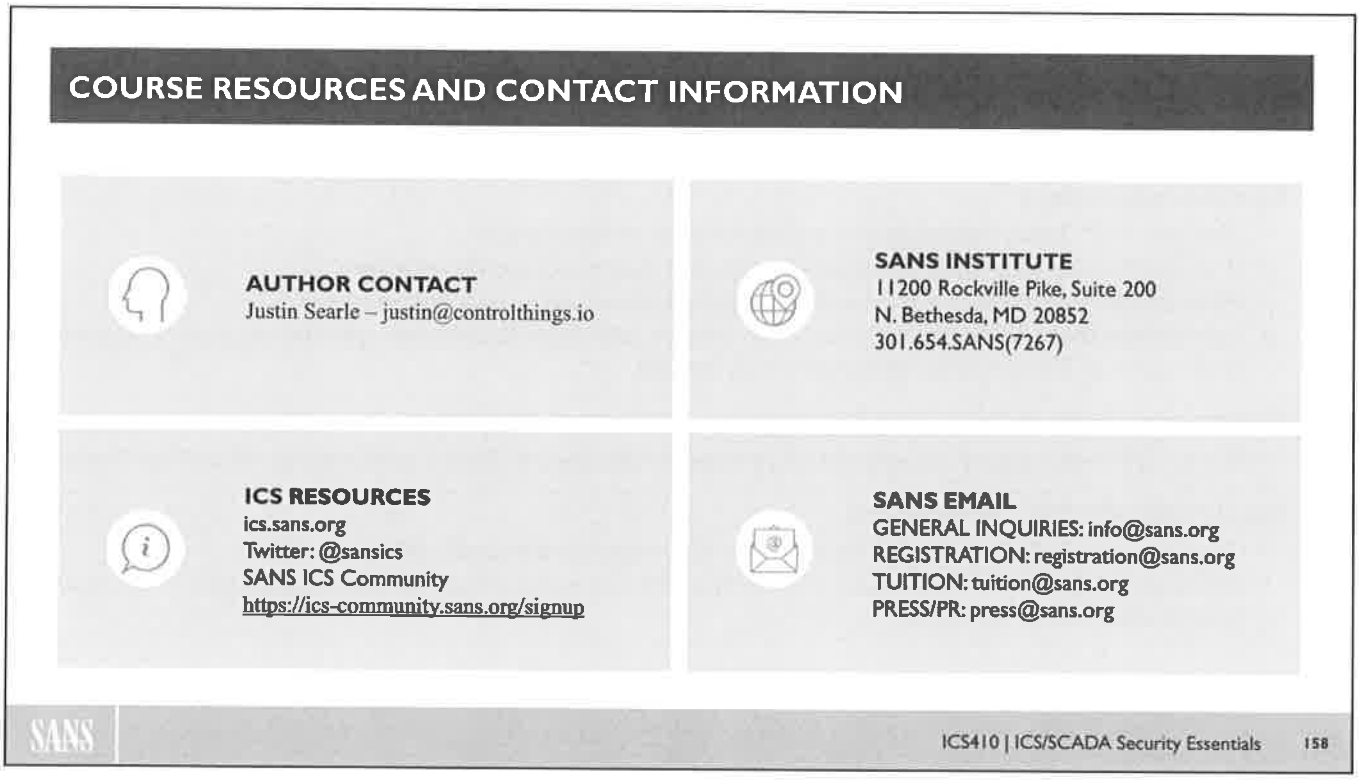
- If your devices support newer control protocols with cryptographic protections, please use them

Recommendations to vendors

- Continue developing and making available new cryptographically safe protocols
- Add support for standards-based protocols which have cryptographic extensions, and encourage your customers to use them

SANS ICS410 | ICS/SCADA Security Essentials 157

This page intentionally left blank.



COURSE RESOURCES AND CONTACT INFORMATION

AUTHOR CONTACT
Justin Searle – justin@controlthings.io

SANS INSTITUTE
11200 Rockville Pike, Suite 200
N. Bethesda, MD 20852
301.654.SANS(7267)

ICS RESOURCES
ics.sans.org
Twitter: @sansics
SANS ICS Community
https://ics-community.sans.org/signup

SANS EMAIL
GENERAL INQUIRIES: info@sans.org
REGISTRATION: registration@sans.org
TUITION: tuition@sans.org
PRESS/PR: press@sans.org

SANS ICS410 | ICS/SCADA Security Essentials 158

This page intentionally left blank.

© 2019, Justin Searle