

410.3 Supervisory Systems

SANS

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.

SANS

Supervisory Systems

© 2019 Justin Searle | All Rights Reserved | Version E01_01

The **SANS ICS 410**, 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, Eric 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.

TABLE OF CONTENTS**PAGE**

Enforcement Zone Devices	4
Understanding Basic Cryptography	17
Wireless Technologies	35
Wireless Attacks and Defenses	55
EXERCISE 3.1: Network Forensics of an Attack	68
Purdue Level 2 and 3 Attacks	78
Historians and Databases	101
EXERCISE 3.2: Bypassing Authentication with SQL Injection	118
HMI and UI Attacks	124
Password Defenses	137
EXERCISE 3.3: Password Fuzzing	148

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. Enforcement Zone Devices
 - Firewalls and NextGen Firewalls
 - **Data** Diodes and Unidirectional Gateways
2. Understanding Basic Cryptography
 - Crypto Keys
 - Encryption, Hashing and Signatures
3. Wireless Technologies
 - Satellite and Cellular
 - Mesh Networks and Microwave
 - Bluetooth and Wi-Fi
4. Wireless Attacks and Defenses
 - 3 Eternal Risks of Wireless
 - Sniffing, DoS, Masquerading, Rogue AP
5. **Exercise 3.1: Network Forensics of an Attack**
6. Purdue Level 2 and 3 Attacks
 - Historians and Databases
 - **Exercise 3.2: Bypassing Auth with SQL Injection**
 - HMI and UI Attacks
 - **Web-based Attacks**
 - Password Defenses
 - **Exercise 3.3: Password Fuzzing**

This page intentionally left blank.

Enforcement Zone Devices

Applicable Standards:

- **NIST CSF v1.1:** PR.AC-5
- **ISA/IEC 62443-2-1:2009:** 4.3.3.4
- **ISA/IEC 62443-3-3:2013:** SR 3.1, SR 3.8
- **ISO/IEC 27001:2013:** A.13.1.1, A.13.1.3, A.13.2.1, A.14.1.2, A.14.1.3
- **NIST SP 800-53 Rev. 4:** AC-4, AC-10, SC-7
- **CIS CSC:** 9, 14, 15, 18
- **COBIT 5:** DSS01.05, DSS05.02

This page intentionally left blank.

ENFORCEMENT ZONE DEVICES

ICS environments should use specialized network devices to protect control protocols and their data payloads

- **Network Firewalls** enforce rules limiting the type of network traffic between two or more networks
- **Application Layer Firewalls** are specialized network firewalls that can enforce rules for application layer protocols
- **Network Intrusion Detection Systems (NIDS)** monitor traffic for known attacks and other suspicious behaviors
- **Data Diodes** provide for one-way communication paths in or out of the control environment
- **Unidirectional Gateways** are a modern version of data diodes that extend data diode protections to non-unidirectional protocols

ISA-95/Purdue model recommends basic security zones for an ICS

As always, the right solution must be engineered to fit the environment

ICS environments should use specialized network devices to protect control protocols and their data payloads. This helps to keep attackers out of the control networks and provides defenders the visibility to understand what types of data passes enforcement zones. Network firewalls enforce rules created by security professionals to limit the type of network traffic between two or more networks. For instance, allow vendor X remote access to this control network but no one else, or allow this PLC to communicate with its HMI but no other network device. Application layer firewalls are specialized network firewalls that can enforce rules for application layer protocols (think OSI Layers 5–7). This can extend that PLC example to add rules to say the HMI can send only read requests but no write requests (aka command signals). Network intrusion detection systems (NIDS) monitor traffic for known attacks and other suspicious behaviors, such as ARP spoofing attacks or buffer overflow attacks. Data diodes provide for one-way communication paths in or out of the control environment to prevent any attacker from obtaining bidirectional access to a control network. Unidirectional gateways are a modern version of data diodes that extend data diode protections to non-unidirectional protocols like Modbus, DNP3, IEC 104, MMS, and OPC. As always with ICS, the right solution must be engineered to fit the environment.

The ISA-95/Purdue model recommends the following enforcement zones as a starting point for ICS networks:

- Safety Integrated System (SIS) zone
- Basic Control/PLC zone
- Supervisory/HMI zone
- Process Information/Data Historian zone
- IT Network zone

FIREWALLS

Firewalls can provide a number of benefits

- Limit traffic between network segments
- Filter conversations based on initial packet alone
- Perform NAT (Network Address Translation)
- Encrypt communications for VPN (IPSec)

Should be layered to provide Defense-in-Depth

Valuable to aid in intrusion detection and forensics

- Only if you log allowed and denied traffic
- Send that log to a SIEM or other analytics system

Firewalls are interesting in that they can play a variety of roles, each with significant benefits. Besides just enforcing an organization's security policies, firewalls can:

- Reduce risks by protecting systems from incoming and outgoing attempts to exploit vulnerabilities.
- Increase privacy by making it harder to gather intelligence about a site.
- Filter communications based upon content, such as offensive or malicious content coming in or proprietary content flowing out of an organization.
- Encrypt communications for confidentiality.
- Provide records concerning both successful and blocked network traffic, which may be critical for incident handling and forensics.
- Serve as a "noise filter" and conserve bandwidth.

Firewalls of different types can be cascaded effectively or otherwise applied in a myriad of network topologies. Some of the most secure networks intentionally use firewalls of different brands or types in series with similar rule sets as part of a Defense-in-Depth strategy. Even with firewalls, Defense-in-Depth needs to be practiced.

FIREWALL RULES

Traditional firewalls create rules using IP and TCP/UDP data

- permit tcp from 10.42.84.33 to 172.16.13.66 eq 502

Some application/nextgen firewalls can further restrict with application layer data

- permit modbus function read-multiple-coils

Nextgen firewalls usually include other security detection engines such as

- Intrusion detection systems (IDS)
- Detecting protocols on non-standard ports
- Web proxy capabilities

When a packet doesn't match an existing rule, it defaults to a set policy

- Default deny: More restrictive whitelist of allowed traffic
- Default allow: More permissive blacklist of traffic not allowed

Enforcement zone firewalls should use a default deny policy for **ALL** traffic in **BOTH** directions

Traditional firewalls are fast, but they can be fooled. Application layer and nextgen firewalls are at the opposite end of the spectrum. Among firewalls, they are the most inconvenient to manage, such as when a new protocol isn't yet supported; however, application layer and nextgen firewalls usually provide the best security. The primary difference between application layer firewalls and nextgen firewalls is usually the inclusion of other security detection engines such as intrusion detection systems (IDS).

Firewalls are designed with a default rule: If a packet doesn't match another rule, the default rule drops the packet, as should happen on enforcement zone firewalls. This is known as “deny all except that which is explicitly allowed.” Firewall administrators who override this rule create an “allow all except that which is explicitly denied” policy, which is dangerous on enforcement zone firewalls. This is one reason your security policy must be linked to your organizational policy. Either you make the detailed decisions necessary to establish firewall rules in accordance with the organizational policy, or you make them arbitrarily. They likely will not withstand organizational pressure over time if they are arbitrary.

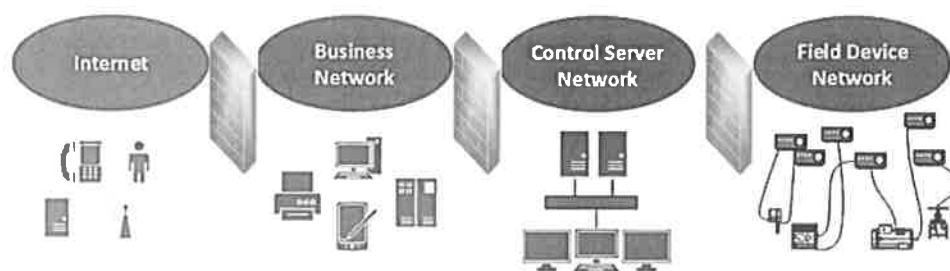
ICS TRAFFIC FILTERING

Devices labeled as firewalls by their vendors should be used in every enforcement zone

- Between Internet and Business Networks
- Between Business Network and Control Server Network
- Between Control Server Network and SCADA Field Device Networks

Control Server Network is usually considered the highest security zone

- Processes may be in the Field Networks; however, all of those processes are controlled by Control Servers



Ingress filtering refers to filtering applied to incoming traffic—from the perspective of your network. Generally, most of the firewall rules are applied to inbound traffic. Consider this simple example: All inbound packets should be dropped if they contain a source address from within the protected network address space. Whether these packets are the results of an attacker spoofing your address or a routing problem, they should not be allowed in. In the event that internal packets inadvertently have been routed to the public network, this rule will make both the routing error and the failure to block them with appropriate egress filtering conspicuous so that these errors can be corrected.

Egress filtering simply provides filtering for addresses, and more advanced Layer 7 egress filtering can be utilized to help identify unauthorized outbound communications. Because of personal firewalls, egress filtering applies to individual computers as well as to networks.

Flooding Denial-of-Service attacks often use a faked source address so that it is hard to pinpoint the location of the attacking computer. These attacks are not elegant; they simply spew packets at the maximum rate possible. They can be launched by malicious users who are "playing" with their computer systems, but also, they can be launched from compromised computers or systems infected with trojans or other malicious software.

If your site applies egress filtering at the access point between your site and the internet, you obviously are being a good neighbor (and being prudent with regard to downstream liability).

Egress filtering is also a wonderful intrusion detection technique, utilizing your firewall log files. Suppose one of your internal machines has been infected with a macro virus. Indirectly, you can detect this by noting its attempts to spread through outbound traffic. Failure to detect this and take action raises issues of downstream liability.

DATA DIODES

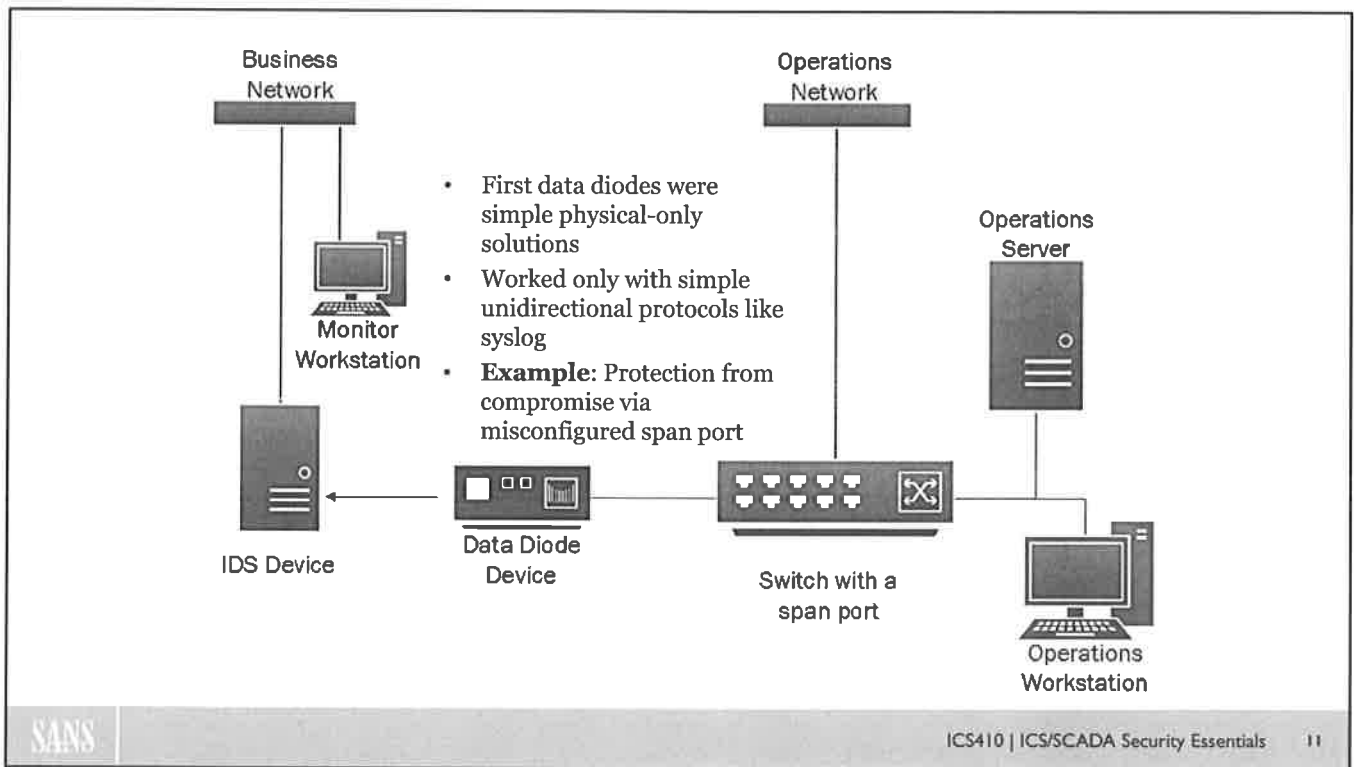
A **diode** is a semiconductor device with two terminals, typically allowing the flow of current in one direction only

A **data diode** typically references military technology that moves data into classified networks without risk of leaking classified information

Primary purpose is to prevent attackers from gaining bidirectional access

A diode has little to zero electrical resistance to current flow in one direction, and high or infinite resistance to current flow in the other direction, thus creating an electric component that allows current flow in a single direction. Typically, a diode has a bar on one end indicating the terminal where current will flow out of and not in; the input side is referred to as the anode and the output side is referred to as the cathode.

Data diodes are used routinely to protect secrecy in military and government networks. Data diodes are hardware-focused; software associated with diodes tends to be fairly primitive. In principle, you can turn diodes around to send data out of safety-critical networks instead of into confidentiality-critical networks, but diodes have limited support for industrial protocols.



Control system networks were typically isolated environments with little to no communication in or out. Over the years, the need for secondary site replication, remote support, asset performance management, business intelligence analysis, and management visibility needs have driven the reality of today's interconnected control system environments. The need to connect and provide data has always been examined in parallel with the level of risk added to the control system by connecting external environments.

The challenge is to provide a means to send data from the control network to a less secure network, but prevent traffic from the less secure network back in.

An example of this need can be seen in an approach often utilized by IDS tap points. An IDS system in a certain level of a network that wants to see network traffic in another more secure network would, if simply connected to both networks, effectively dual-homed and bypass the security controls in place between the networks, or alternatively, unique IDS collectors would need to exist in each network. This problem is solved through a physical one-way data flow solution. IDS vendors tend to solve this problem with either commercial network taps or span port configurations on network switches. If a switch's span port is misconfigured, an attacker could possibly use that port to compromise the protected network. To prevent this, we can use data diodes to ensure that no communication can flow into the protected network regardless of the span port's configuration.

MODERN DATA DIODES AND UNIDIRECTIONAL GATEWAYS

Modern version of a data diode

- Uses both hardware and software to extend data diode protections to non-unidirectional protocols
- Software components on the protected end gather data to send
- Hardware component (usually optical) enforces diode function
- Software components on the other end to either forward or simulate a server

These are not drop-in replacements for firewalls

- Must be able to pre-determine all data to collect on the secure side
- Must have support and a license for your desired protocol

Two most common vendors for ICS

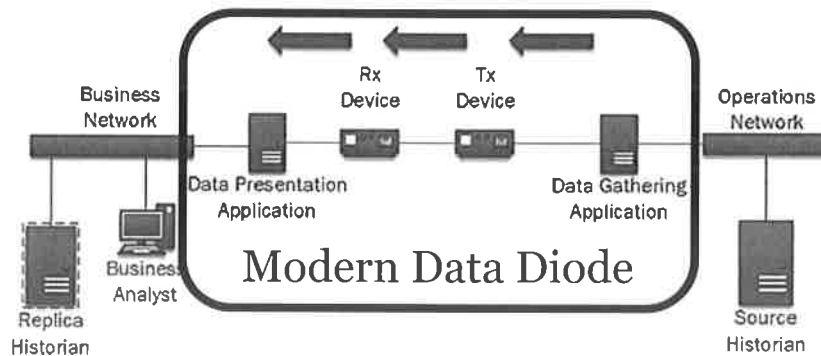
- Waterfall calls their devices unidirectional gateways
- Owl and most other vendors still call their products data diodes

Unidirectional gateway is the term the ISA SP-99 and IEC 62443 standards use to refer to the combination of hardware and software that allows information to flow out of control system networks unidirectionally. There are a number of solution providers that offer a method of solving this problem; some provide optical network interface cards that are capable of sending or receiving only data. Others provide a network appliance with this kind of optical isolation "under the hood." Still others provide a pair of network devices, each with a copper and an optical interface: One appliance able to transmit only on the optical interface and the other appliance able to receive only on that interface. The technology is deployed most commonly between control networks and corporate networks, and second-most-commonly to enable monitoring of networks of Safety Instrumented Systems without any risk of the monitoring connection being used to launch attacks on those systems.

Unidirectional gateways are not drop-in replacements for firewalls. If infrastructures are architected correctly, unidirectional gateways can be used in some scenarios to replace firewalls, but not in every situation. To use a unidirectional gateway, it has to support your protocol. You also have to predetermine EXACTLY what data you need from the secure network because the unsecured network cannot request anything.

MODERN DATA DIODES IN ICS

This approach can be used in data historians, file transfers, asset monitoring, ICCP, OPC, Modbus, and DNP3, to name a few



SANS

ICS410 | ICS/SCADA Security Essentials

13

Unidirectional gateway software replicates servers and emulates devices.

For example, the software queries historians on a protected ICS network for data, transmits the data unidirectionally to an external network (usually a corporate network), and then inserts the data into a replica historian. Corporate users and applications can interact with the replica in any way they want without putting the source historian or control system network at risk.

Unidirectional server replication works with what seems like protocols. The most commonly deployed unidirectional solution is historian replication. The second-most-commonly deployed solution is OPC replication. Gateway software on the inside network queries the real OPC servers for data, sends the data out to the corporate network, and emulates the original OPC servers, serving data to anyone who requests it using the OPC protocol on the corporate network. This way, no second historian server needs to be purchased. The full, bidirectional OPC protocol is used to gather data, the data is extracted from the protocol and sent across the unidirectional hardware, and the unidirectional gateway software stores the data on the corporate side in a real OPC server. That server uses the real, bidirectional OPC protocol to publish data on the corporate network.

The same approach can be used to monitor devices and systems using Modbus, DNP3, ICCP, and many other bidirectional protocols.

SHORTCOMINGS OF ENFORCEMENT ZONE DEVICES

Attacks at the application layer may sneak through

- Traditional firewalls look only at IP and TCP/UDP headers
- Application layer and nextgen firewalls may not support protocol or have adequate rules

Dual-homed devices may bypass these devices

- Dial-up, VPN, Wi-Fi, cellular

Doesn't stop sneakernets (USB drives)

Organizations may let down guard in other security areas

- Passwords, patches, hardening, monitoring, encryption

Management sees these devices as silver bullets

With the value that enforcement zone devices offer, it can be tempting to think that they are cure-alls. They are not. Enforcement zone devices are not bulletproof. They do not stop all attacks. In fact, they can be attacked themselves.

Although these solutions provide a technical means to resolve the problem of information sharing from a critical network to the business network, they do not prevent human threats. There is still a need to provide adequate physical security for the critical network components to prevent unauthorized access.

Even with data diodes in place, an individual intentionally or unintentionally may simply wire around the data diode and connect the two networks together, effectively bypassing some of the valuable controls put in place.

There is also the growing risk of mobile devices, which can be easily carried into critical network locations and connected.

Many people foolishly have blind faith in enforcement zone devices. You will hear statements like, "We are behind a firewall. Why do we need to put patches on our systems, or use access controls on our web servers?" One of the downsides of having enforcement zone devices is that an organization can become careless about other aspects of security. The best way to think of enforcement zone devices conceptually is like an umbrella. When you use an umbrella, it keeps a lot of the rain off you, especially your head. However, some of those raindrops get through the perimeter defense. In information warfare, we call these leaks.

ENFORCEMENT ZONE DEVICE SUMMARY

Provide a measure of protection for all protected hosts at a reasonable cost

Can be a primary intrusion detection sensor

Packet filters, stateful inspection, and application gateways provide a mix of capabilities to meet requirements

Enforcement zone devices like Data Guards and data diodes are specialized application layer firewalls for ICS networks

This page intentionally left blank.

TAKEAWAYS AND RECOMMENDATIONS

Section takeaways

- Basic firewalls are a must in enforcement zones
- There are many features in modern firewalls that can help
- Unidirectional gateways are good choices, but not drop-in replacements for all firewalls

Recommendations to owner/operators

- Whitelist all traffic, both directions, through enforcement zones
- Monitor and store conversations through enforcement zones

Recommendations to vendors

- Document all required traffic patterns for your clients

This page intentionally left blank.

Checkpoint and PaloAlto is leads vendor when the subject is ICS firewalls.

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. Enforcement Zone Devices
 - Firewalls and NextGen Firewalls
 - Data Diodes and Unidirectional Gateways
2. Understanding Basic Cryptography
 - Crypto Keys
 - Encryption, Hashing and Signatures
3. Wireless Technologies
 - Satellite and Cellular
 - Mesh Networks and Microwave
 - Bluetooth and Wi-Fi
4. Wireless Attacks and Defenses
 - 3 Eternal Risks of Wireless
 - Sniffing, DoS, Masquerading, Rogue AP
5. **Exercise 3.1: Network Forensics of an Attack**
6. Purdue Level 2 and 3 Attacks
 - Historians and Databases
 - **Exercise 3.2: Bypassing Auth with SQL Injection**
 - HMI and UI Attacks
 - Web-based Attacks
 - Password Defenses
 - **Exercise 3.3: Password Fuzzing**

This page intentionally left blank.

Understanding Basic Cryptography

Applicable Standards:

- **NIST CSF v1.1:** PR.DS
- **ISA/IEC 62443-3-3:2013:** SR 3.1, SR 3.4, SR 3.8, SR 4.1, SR 4.2
- **ISO/IEC 27001:2013:** A.8.2.3, A.13.1.1, A.13.2.1, A.13.2.3, A.14.1.2, A.14.1.3
- **NIST SP 800-53 Rev. 4:** MP-8, SC-8, SC-11, SC-12, SC-28
- **CIS CSC:** 13, 14
- **COBIT 5:** APO01.06, BAI02.01, BAI06.01, DSS04.07, DSS05.02, DSS05.03, DSS06.06

This page intentionally left blank.

CRYPTOGRAPHY INTRODUCTION

Cryptography has multiple goals

- **Confidentiality** via encryption
- **Integrity** via hashing
- **Authentication** via pre-shared key or asymmetric key
- **Nonrepudiation** via digital signatures

All four have their place in industrial control systems

- HMI and other engineering/administrative interfaces
- Communications between field devices and their control servers
- Communications between control servers and other servers such as historians and corporate servers

In control systems, the four main goals to be accomplished are confidentiality, integrity, authentication, and nonrepudiation. Confidentiality is focused on making sure the disclosure of the information is properly protected. For confidentiality, encryption algorithms such as AES, RC4, and IDEA are commonly used. Integrity is focused on making sure the information is accurate and not altered in an unauthorized manner. For integrity, hashing algorithms such as MD5 and SHA are used. Authentication is focused on proving that someone is who they say they are. For authentication, a pre-shared key or asymmetric keys such as RSA and ECC are used. Nonrepudiation is proving that what someone sent came from them and has not been modified or altered. For nonrepudiation, we create a digital signature that is asymmetric and combine it with hashing.

Cryptography is used to protect all forms of communications, such as:

- HMI and other engineering/administrative Interfaces
- Communications between field devices and their control servers
- Communications between control servers and other servers such as historians and corporate servers

KEYS

Key

Plaintext → Algorithm ← Ciphertext

Keys permit the existence of public algorithms

- The strength of a cryptosystem rests with the strength of its keys
- Recommended key length depends on algorithm
- Strongest keys are randomly chosen among entire keyspace

40 bits of protection
001110101001010101011010101010101500110

128 bits of protection
0011101010010101011010101010000110 001110101001010101011010101000110 00111010100101010101010100011001000110

128-bit keys offer approximately a trillion times more protection than 40-bit keys

SANS | ICS410 | ICS/SCADA Security Essentials 20

A *cryptosystem* is the collection of all possible inputs and all possible outputs, in addition to the algorithm and keys. *Cryptographic keys* are simply values used to initialize a particular algorithm. The important aspect of keys in regard to cryptosystems is that only the key, not the algorithm, needs to be protected. This means that algorithms may be widely distributed and their internal workings publicly documented. It is only the key that must be protected from thievery by communicating entities.

The uniqueness of cryptographic keys is just as important as the keys themselves. *Keyspace* is a critical concept concerning cryptographic keys. The larger the keyspace (or total number of possible keys), the less likely an attacker is to discover a given key through brute force. A brute force attack on a key involves trying every possible key until finding one that works. For instance, the Caesar Cipher had a keyspace of only 25 possible keys, which is trivial to exhaust or brute force. It should be impossible for an attacker to guess a cryptographic key that matches the one used to encrypt correspondence. Let's say the total number of possible unique car keys is approximately 200,000. Although that might not be accurate, even if the number is 10 million, the total number of possible unique cryptographic keys for a given cryptosystem needs to be exponentially larger simply to afford the keyspace protection against guessing an encryption key through brute force. Why? It's far easier to use a computer to iterate through a billion cryptographic keys than it is to physically re-create a million car keys. In short, a cryptographic keyspace must be absolutely enormous to afford sufficient protection.

KEYS – ICS CHALLENGES

Based on the deployment of devices, cryptographic keys are often the same for a large subset of devices

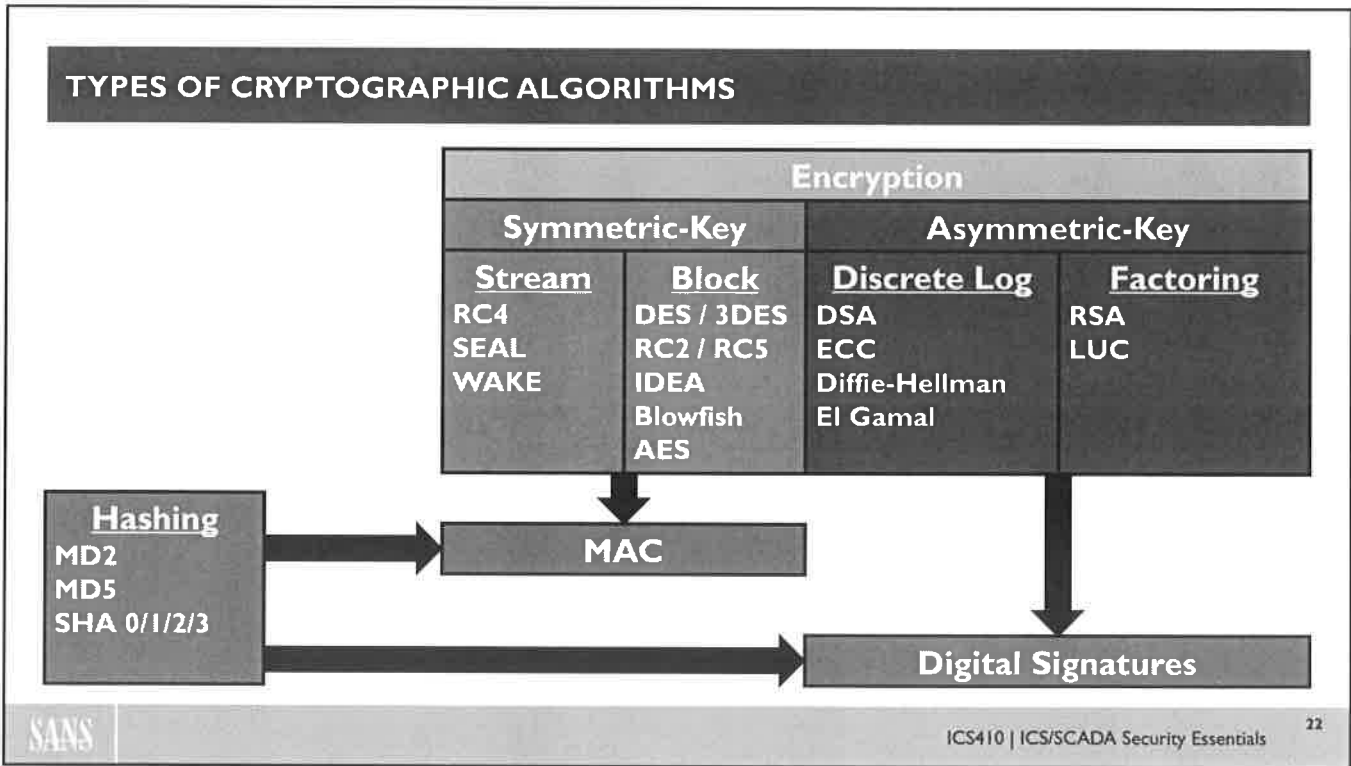
Based on the location of the devices, keys are often not changed as often as they should be

To more easily manage devices, keys are often built into software and applications and not always properly protected

Although managing and controlling keys is always a challenge with cryptography, they present a unique challenge with control systems. Hardware that is deployed is often configured by the vendor with keys preloaded into the systems. Therefore, not only is it possible to have multiple devices that all have the same keys but also all products from a given vendor could have the same keys not only for the devices at your organization but for all devices that vendor produces. If these pregenerated keys become public, it could represent a big risk and exposure.

The location of the devices also creates a challenge. Because some devices are not easy to access and/or require strict testing and approval before making changes, after keys are loaded into a device, they often do not change.

From an operational perspective, the people monitoring the system should not necessarily have to know about crypto and/or have to remember the keys. Therefore, keys are often hardcoded into applications and programmed to increase efficiency and reduce errors. However, the method in which the keys are loaded into the programs often leaves them exposed because they are not properly protected.



There are two different types of encryption algorithms. Symmetric stream ciphers are fast, and asymmetric factoring algorithms are slow. Diffie-Hellman is great for secure key exchanges, but not necessarily optimal for encryption. It is important to note that not all the algorithms listed are technically considered secure, but some ICS systems still utilize legacy algorithms.

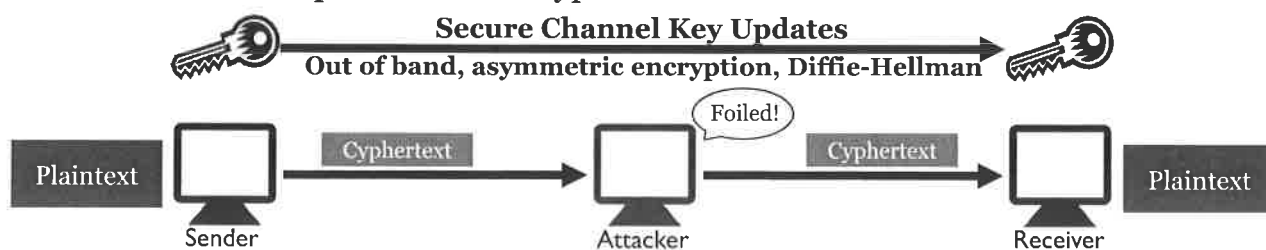
Let's look at an example of how these algorithms might be used. If we want to encrypt a message, that is, transform it in such a way that prying eyes cannot read it, we may choose either *symmetric* algorithms such as RC4 or AES, or *asymmetric* algorithms such as RSA or ECC, but not any of the hashing algorithms (such as MD5 or the SHA family). And when we combine these two types of encryption with a hashing algorithm, we can get a stronger and more secure form of data integrity, which we will discuss later in this section.

As we can see, cryptography in ICS systems promises to be a challenging topic, but when completed with this module, the reader will have a rudimentary understanding of all of the above.

SYMMETRIC-KEY CRYPTOSYSTEMS

"Secret Key" Encryption

- Single key for encryption/decryption
- Fast and efficient algorithms
- Nonrepudiation not included
- Common in ICS protocols if encryption is used



Same for both stream and block cyphers

Symmetric-key cryptography uses a single key for both encryption and decryption; this key is shared secretly between the sender and receiver. Because symmetric-key encryption uses only one key for both encryption and decryption, the key must be kept secret and is also referred to as *secret key encryption*. The primary application of symmetric encryption is privacy, where only the parties with the key can encrypt and decrypt messages for each other.

The big issue with secret keys is managing the key creation and exchange to avoid key compromise. Also, the greater the number of parties that share the secret key, the greater the exposure of the key.

The bottom line is this: Because symmetric-key cryptosystems are so much faster than asymmetric-key systems but lack the latter's key management and digital signatures, the two are often combined to achieve the best of both worlds.

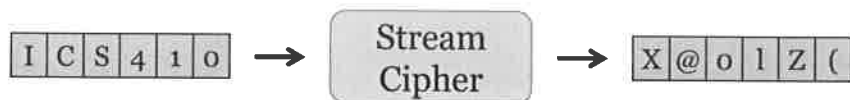
There are a number of symmetric encryption schemes in common use today, all believed to be mathematically strong. If a cryptanalyst cannot defeat the ciphers by finding a weakness in the mathematical algorithms, then the remaining approach is a brute force attack to guess all possible keys.

Examples of symmetric encryption schemes in common use today are the Advanced Encryption Standard (AES), Blowfish, the Data Encryption Standard (DES), Triple DES, and the International Data Encryption Algorithm (IDEA).

SYMMETRIC-KEY STREAM CIPHER

Encrypts 1 bit of data at a time

- Examples: RC4, A5/1, and GSM
- Plaintext length is equal to ciphertext length
- Initialization vectors (IV) are used to prevent cyphertext repetition



Fast but becoming less popular due to:

- Management overhead
- Security concerns
- Poor implementation often allows for decryption without the key

Included among symmetric encryption options are *stream ciphers*. In a stream cipher, the input data is encrypted one bit at a time without adding additional encrypted content for each bit of input plaintext. This gives a stream cipher the property of identical length of plaintext and ciphertext data, possibly excluding protocol header information or checksums.

Of stream ciphers, RC4 is the most commonly used, popularly for SSL/TLS encryption. The A5/1 stream cipher is also popular, used for GSM phone calls and SMS messages, whereas the E0 stream cipher is used to protect Bluetooth transmissions.

A stream cipher is considered to be fast as an encryption function, but is becoming less popular due largely to added protocol requirements for use as well as security concerns.

SYMMETRIC-KEY BLOCK CIPHER

Encrypts one block of data at a time:

- Plaintext is padded to the next block length

Output is always divisible by block length:

- DES and 3DES = 64 bits or 8 bytes
- AES-128 and AES-256 = 128 bits or 16 bytes



Used with a cipher mode:

- Such as ECB, CBC, or CTR
- Determines how to encrypt plaintext blocks after the first
- Some block ciphers use IVs to prevent repetitious ciphertext

In contrast to a stream cipher, a block cipher encrypts one block of data at a time, typically 8 or 16 bytes, often but not always matching the length of the key. An example of this is AES-256 in which the key is 256 bits; however, the algorithm is still only a 128-bit block cipher. A block cipher cannot encrypt content that is shorter than the block length, so it uses a padding function to increase the final block of plaintext to an even block length prior to encryption.

When a block cipher is used to encrypt data, it is done with a block cipher mode that influences how the data is encrypted, such as Electronic Code Book (ECB), Cipher Block Chaining (CBC), and Counter mode (CTR).

Getting into the details of cipher modes is beyond the scope of this class, but here is a link for you if you are interested in learning more:

https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation

Not good if same block encrypt

Always some output

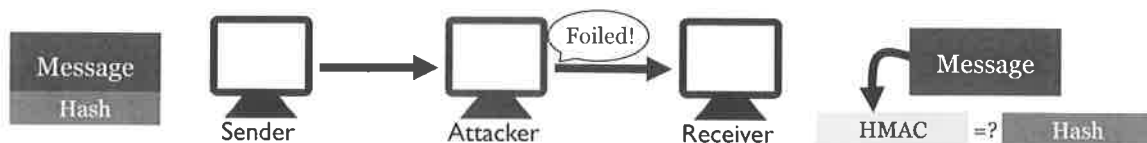
HASH FUNCTIONS

Hash functions summarize data into a fixed-length output value:

- Irreversible process (NOT ENCRYPTION!)
- Examples: MD2, MD4, MD5, RIPEMD-160, SHA-1, SHA-2, SHA-3
- FIPS 140 recommends SHA-3 algorithms

We can use openssl to produce various hashes and HMAC hashes

```
# openssl dgst -md5 message
MD5 (message) = 399b0b50ffdbd4550aa278a28fa1868f
# openssl dgst -md5 -hmac "secretkey" message
HMAC-MD5 (message) = a56e46f2cd6b08489fd92457d3c4c79e
```



SAAS

ICS410 | ICS/SCADA Security Essentials

Remember that there are three types of cryptography algorithms: Secret key, public key, and hash functions. Unlike secret key and public key algorithms, *hash functions* (also called *message digests*) have no key. Instead, a fixed-length hash value is computed based on the plaintext that makes it impossible for either the contents or *length* of the plaintext to be recovered.

The primary application of hash functions in cryptography is message integrity. The hash value provides a digital fingerprint of a message's contents, which ensures that the message has not been altered by an intruder, virus, or by other means. Hash algorithms are effective because of the extremely low probability that two different plaintext messages will yield the same hash value.

There are several well-known hash functions in use today:

- **Message Digest 2 (MD2):** Byte-oriented produces a 128-bit hash value from an arbitrary-length message designed for smart cards.
- **MD4:** Similar to MD2, this function is designed specifically for fast processing in software.
- **MD5:** Similar to MD4 but slower because the data is manipulated more, this function was developed after potential weaknesses were reported in MD4.
- **Secure Hash Algorithm (SHA):** Modeled after MD4 and proposed by NIST for the Secure Hash Standard (SHS), this function produces a 160-bit hash value. This was published by NIST as FIPS PUB 180-1. NIST superseded PUB 180-1 with FIPS PUB 180-2, released August 1, 2002. The -2 release provided three more SHA algorithms: SHA-256, SHA-384, and SHA-512, with further modifications in late documents. For more on this standard: <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf>

MESSAGE AUTHENTICITY CHECK (MAC)

Message Authentication Code (MAC) = function(hash, secret)

- Adversary doesn't know secret, so they cannot re-create a valid hash
- Examples: HMAC, CBC-MAC, PMAC, UMAC and VMAC

We can use openssl to produce various hashes and MACs

```
# openssl dgst -md5 message
MD5 (message) = 399b0b50ffdbd4550aa278a28fa1868f
# openssl dgst -md5 -hmac "secretkey" message
HMAC-MD5 (message) = a56e46f2cd6b08489fd92457d3c4c79e
```



SANS

ICS410 | ICS/SCADA Security Essentials

There is a significant problem in the use of hashes for valid authentication, as shown in the previous slide. When the transmitter includes the hash along with the original content, a man-in-the-middle (MITM) attacker can intercept the message and hash prior to delivery, change the message, and simply replace the original hash with one that is computed over the changed content. Upon receipt, the victim does not know that the message has been modified, only that the sent hash is correct.

To address this limitation, the IETF developed a method of hashing known as a Hashed Message Authenticity Check (HMAC) hash. In an HMAC hash, the transmitter creates a hash with the assistance of a secret value known to the transmitter and the recipient. To recompute the hash, the attacker must also know the secret value, preventing him from creating valid hashes for intercepted data.

The HMAC function is independent of the hashing method itself, using the hashing function as a naming suffix (for example, HMAC-MD5, HMAC-SHA-512, and more). Any hashing function can also be used as an HMAC hash.

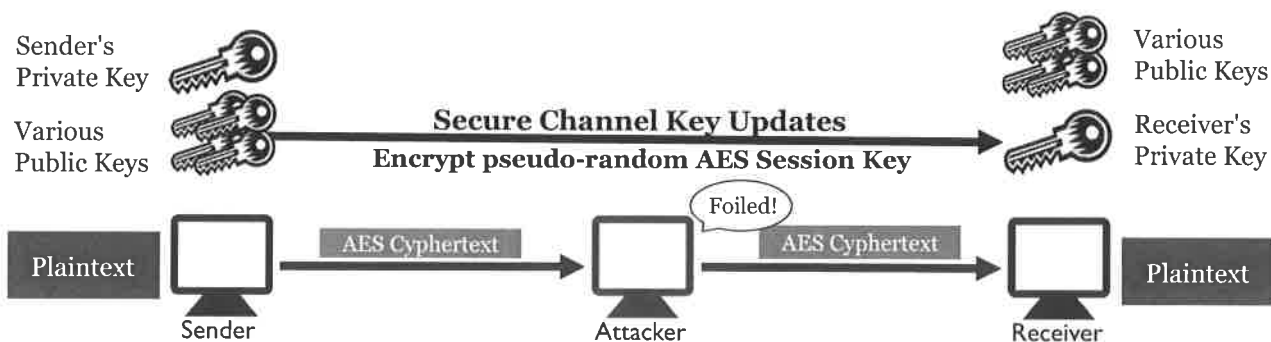
The HMAC hash is computed using the hashing function twice, as shown on this slide. Initially, the secret "K" is XOR'd with a fixed value known as the IAD (repeating 0x36-bytes) before being concatenated with the message M. This value is then hashed using the named hashing function (for example, MD5, SHA-512). After computing the inner hash, the secret "K" is XOR'd with the fixed value known as the opad (repeating 0x5c bytes) before being concatenated with the inner hash. The resulting value is then hashed using the named hashing function a second time to produce the HMAC hash.

Although most UNIX operating systems include tools to compute SHA-1 or MD5 hashes over arbitrary content (sha1sum, md5sum), these tools do not accommodate the computation of HMAC hashes. To compute an HMAC hash, we can use the OpenSSL command-line utility as shown on this slide, specifying our secret value on the command line and the filename of the content to hash ("message" in this example).

ASYMMETRIC-KEY CRYPTOSYSTEMS

"Public-Key" Encryption

- Two keys per entity: Public/private key pair
- Slower and more resource intensive, so generally used to wrap symmetric encryption like AES
- Public keys widely distributed within digital certificates
- Technical nonrepudiation via digital signatures
- Used as a secure channel for symmetric-key exchange



SANS

ICS410 | ICS/SCADA Security Essentials

28

The management problems associated with symmetric keys are so overwhelming that they virtually preclude their use by themselves in e-commerce. We can use public key computation to develop a shared message key. Also, algorithms like Diffie-Hellman can be used to exchange a secret key. Again, the general idea is to exchange keys securely, perhaps only once, to secure a given session, such as a visit to a webpage to execute a credit card transaction.

Public key cryptography or asymmetric encryption methods have two keys: One used for encryption and the other for decryption. From a mathematical standpoint, anything that is encrypted with one of the keys can be decrypted only with the other key. Asymmetric encryption has many applications, but the primary ones today are key exchange (for symmetric encryption), authentication, and nonrepudiation.

Stanford University professor Martin Hellman and graduate student Whitfield Diffie first described modern asymmetric encryption publicly in 1976. Their paper described a two-key cryptosystem in which two parties could engage in a secure communication over a non-secure communications channel without sharing a secret key. The mathematical trick of asymmetric encryption depends on the existence of so-called *trapdoor functions*. These are mathematical functions that are easy to calculate with an inverse that is difficult to calculate. Here are two simple examples:

- *Multiplication versus factorization*: Multiplication is easy; given the two numbers 9 and 16, it takes almost no time to calculate the product: 144. But factoring is harder; it takes longer to find all the pairs of integer factors of 144, and then to determine the *correct* pair that was actually used.
- *Exponentiation versus logarithms*: It is easy to calculate, for example, the number 3 to the 6th power to find the value 729. But given the number 729, it is much harder to find the set of integer pairs, x and y , so that $\log_x y = 729$ and then, again, to determine what pair was actually used.

The previous examples are trivial, but they are examples of the concept; namely, the ease of multiplication and exponentiation versus the relative difficulty of factoring and calculating logarithms, respectively. Actual asymmetric encryption algorithms use integers that are prime and can be several hundred digits in length. Multiplying two 300-digit primes, for example, yields a 600-digit product; finding the two prime factors of a 600-digit number is beyond the capabilities of today's known methods. In this case, then, factoring is said to be *intractable* because of the difficulty of solving the problem in a timely fashion.

Keys are derived in pairs and are mathematically related; however, knowledge of one key by a third party does not yield knowledge of the other key. One key is used to encrypt the plaintext, and the other key is used to decrypt the ciphertext; it does not matter which key is applied first, but both keys are required for the process to work. In the real world, how are these asymmetric-key systems used? They are typically used to perform key exchange for symmetric-key algorithms.

Bottom line: Despite being much slower than symmetric-key cryptosystems, asymmetric-key systems are widely used because of their powerful key management and digital signatures—often in concert with symmetric-key systems to attain the best of both worlds.

The true history of asymmetric encryption—and answering the question of its invention—is somewhat murky. There is no question that Diffie and Hellman were the first to publicly publish on the topic. Their classic paper, "New Directions in Cryptography," appeared in the November 1976 issue of *IEEE Transactions on Information Theory*. Diffie and Hellman were not trying to solve the key exchange problem per se, but were trying to make the problem obsolete by inventing a scheme that used a split key; that is, one key for encryption and the second key for decryption. They published their *concept* of split-key crypto but did not identify a function that would work. Rivest, Shamir, and Adleman described their implementation in the paper "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," which was published in the February 1978 issue of the *Communications of the ACM (CACM)*.

Some sources, however, credit Ralph Merkle as the first to describe a system that allows two parties to share a secret using what is now called a Merkle Puzzle. His early work was largely misunderstood, and although he submitted a paper to *CACM* some years earlier, his description did not appear until April 1978. He certainly was not the first to publish, but did he have a workable idea before Diffie and Hellman?

The true invention of public key cryptography probably does not belong to anyone in the United States, however. The article "The Open Secret" in the April 1999 issue of *WIRED Magazine* reports that asymmetric encryption was probably first invented by James Ellis of the UK's Government Communications Headquarters (GCHQ) in 1969. Ellis' work was classified until the late 1990s, so there was no public mention of it, and it is possible that Ellis influenced the work of Diffie and Hellman. The US National Security Agency (NSA) claimed to have knowledge of this type of split-key crypto as early as 1966, but there is no known documentation.

DIGITAL SIGNATURES AND PAYLOAD SIGNING

One problem with HMAC is more than one entity has the key

- You can never have true nonrepudiation
- Digital signatures fixes this since each entity has its own key pair

This is useful for secure communications, especially in ICS protocols

- Master servers and PLCs can digitally sign sensitive requests like control signals and updates
- Signed payloads prevent both man-in-the-middle and spoofed control signals

```
openssl can sign # openssl dgst -sha256 -sign private.key -out signature payload
and verify as # openssl enc -base64 -in signature -out signature.base64
shown here # openssl enc -base64 -d -in signature.base64 -out signature
# openssl dgst -sha256 -verify public.key.pem -signature signature payload
```



SANS

ICS410 | ICS/SCADA Security Essentials

Semantic Digital Signatures are equivalent to signatures affixed to documents with pen and ink: The signature is meant to uniquely identify the signer.

Because pen and ink are useless in an electronic environment, cryptography—specifically asymmetric algorithms—is used to provide the required uniqueness. Handwritten signatures have long held protected legal status as an official recognition of approval on a paper document (despite the fact that handwritten signatures are notoriously easy to forge). In this digital age, it seems only fair that we have a method of signing electronic documents that are unique as well as difficult to forge.

Public-key cryptography allows users and systems to employ their private key to encrypt data, in effect *signing* the data. Because a given private key is intended for one and only one owner, use of the private key in encryption unmistakably associates the user's identity with the encrypted data. This is semantically equivalent to the user hand-signing the data.

Recipients of the signed data employ the user's public key to decrypt and, therefore, verify the sender's signature. In short, by leveraging asymmetric algorithms, users can establish a person's digital signature as authentic. In addition, users can also establish authenticity even if the sender denies having signed the data. This is called *nonrepudiation*.

HEARTBLEED AND ICS

Heartbleed is a vulnerability in the OpenSSL's TLS library

Allows an attacker to gain access to the RAM of the TLS client or server, allowing possible retrieval of

- Private keys and passwords
- Network traffic inside the TLS tunnel
- Other sensitive information

Several ICS products from Siemens, Innominate, and other vendors are vulnerable

Patching and network segmentation are key to minimizing impact

ICS-CERT: "Impact to individual organizations depends on many factors that are unique to each organization"

Heartbleed, which exploited a vulnerability in OpenSSL, impacts many ICS components. Many ICS vendors utilize cryptographic modules like OpenSSL to provide protection of information. Heartbleed allows for the private key and other sensitive information to be exploited.

"Impact to individual organizations depends on many factors that are unique to each organization," ICS-CERT warns. See more: <http://threatpost.com/siemens-update-on-heartbleed-patches-in-ics-scada/105725#sthash.bkWGm5ji.dpuf>

Reference:

Additional information on testing for the Heartbleed vulnerability: <http://www.hacklabs.com/team-penetration-testing/2014/4/8/testing-for-the-tls-heartbleed-vulnerability.html>

with Heartbleed, Attacker can steal a bunch of data from memory which includes keys, hidden data and any sensitive data collected in memory.

QUESTIONS YOU SHOULD ASK WHEN ASSESSING CRYPTO

What cryptographic algorithms are being used?

- How are they being used?
- Are the algorithms FIPS 140 approved?
- Are library implementations FIPS 140 validated?
- If symmetric-key block ciphers are used, which cipher modes are used?
- Are initialization vectors (IVs) used, changed, and their reuse prevented?

How are keys generated?

- Where are the keys stored and protected?
- How are they maintained, revoked, expired?

Who will be responsible for the management of the crypto functions?

- Are your developers trained in implementing crypto?
- Do you have security policies and procedures specifying how to handle crypto?
- Is awareness training given to all engineers and technicians dealing with crypto?

Organizations often rely on cryptography to implement a critical part of their overall security. However, they assume that by using crypto, it will instantly make them secure. As with any solution, if it is not designed, configured, and deployed correctly, it might not provide the protection that you believe it will provide.

Especially in control system environments, organizations have never performed an evaluation to determine whether cryptography is being implemented correctly and is providing the level of protection they are expecting. Based on the knowledge learned in this class, you should go back and start to ask questions, identify weaknesses within your environment, and put together a plan to address them.

COMMON MISTAKES

Most common mistakes made with cryptography in ICS

- Failure to use cryptography ← not using at all
- Confusing encoding with encryption ← alphabet
- Using older or weaker algorithms
- Using proprietary algorithms
- Relying solely on the vendor
- Not performing proper maintenance ← key management

Building and developing cryptographic solutions are not easy tasks. In some cases, organizations and vendors cut corners, which reduces or defeats the effectiveness of utilizing crypto to protect information. First, writing cryptographic algorithms is difficult and instead of utilizing trusted algorithms, entities write their own. In many cases, these proprietary algorithms are not robust or secure. The vulnerabilities create backdoors that can be used to compromise the system.

Second, the power of encryption is through the use of a secure key. This allows only the intended parties to encrypt or decrypt the information. Encoding is where information is scrambled, so it is not visible to the human eye but can easily be decoded by anyone. Therefore, encoding is not a robust method of protecting information, and cryptographic algorithms with keys should always be used for "sensitive" information.

Third, typically vendors will embed crypto into the hardware. Organizations typically assume that everything is done correctly and that no maintenance is required on their part. Also, some of the crypto that is embedded into hardware has older, weaker algorithms. In some cases, these issues can be changed if an organization catches them before the hardware goes through final testing. However, after the hardware is deployed, it is much harder to make changes.

Finally, sensitive information about crypto keys and implementation details should never be embedded into the application or programming code. This code is either available in source code or can be reverse engineered, which would allow sensitive information to be revealed.

TAKEAWAYS AND RECOMMENDATIONS

Section takeaways

- Cryptography is a powerful defense against attackers
- Can strengthen all aspects of data communications and storage

Recommendations to owner/operators

- Enable cryptographic protections in communication protocols when available
- Indicate preference of cryptography-protected protocols to vendors

Recommendations to vendors

- Enable cryptographic protections by default in products
- Have a section at end of install manual on how to disable for legacy
- Use to secure firmware, master server commands, admin functions
- Use to protect data on devices and systems

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. Enforcement Zone Devices
 - Firewalls and NextGen Firewalls
 - Data Diodes and Unidirectional Gateways
2. Understanding Basic Cryptography
 - Crypto Keys
 - Encryption, Hashing and Signatures
3. Wireless Technologies
 - Satellite and Cellular
 - Mesh Networks and Microwave
 - Bluetooth and Wi-Fi
4. Wireless Attacks and Defenses
 - 3 Eternal Risks of Wireless
 - Sniffing, DoS, Masquerading, Rogue AP
5. **Exercise 3.1: Network Forensics of an Attack**
6. Purdue Level 2 and 3 Attacks
 - Historians and Databases
 - **Exercise 3.2: Bypassing Auth with SQL Injection**
 - HMI and UI Attacks
 - Web-based Attacks
 - Password Defenses
 - **Exercise 3.3: Password Fuzzing**

This page intentionally left blank.

Wireless Technologies

Applicable Standards:

- **NIST CSF v1.1:** DE.CM-1
- **ISA/IEC 62443-3-3:2013:** SR 6.2
- **NIST SP 800-53 Rev. 4:** AC-2, AU-12, CA-7, CM-3, SC-5, SC-7, SI-4
- **CIS CSC:** 1, 7, 8, 12, 13, 15, 16
- **COBIT 5:** DSS01.03, DSS03.05, DSS05.07

This page intentionally left blank.

WIRELESS IN CONTROL SYSTEMS

Wireless provides connectivity where wiring isn't possible

ICS leverages many wireless technologies for interconnectivity and backhauls

- Mesh RF Wireless (WirelessHART, ISA100.11a, Wi-Fi Mesh, ZigBee)
- Licensed Radio / Microwave
- Cellular (CDMA, GSM, LTE)
- Satellite Uplink (VSAT, BGAN)

Traditional wireless technologies are used for field laptop connectivity, controller maintenance, and even HMIs

- Bluetooth
- Wi-Fi (IEEE 802.11a,b,g,n,ac...)

With the increasing prevalence of ICS equipment at remote locations where no hardwired telephone or network access can be obtained, wireless communications are becoming unavoidable for many ICS environments.

In areas where wireless communications need to be established within communication distance of other assets owned by the company, many companies are choosing to deploy mesh wireless, licensed radio, or microwave solutions. If out of range or too cost-prohibitive to implement, companies have widely deployed cellular connectivity to their remote sites with a secure VPN connection to their control center. For sites far too remote, such as offshore oil platforms, satellite uplinks may be the only choice to connect equipment to the company's central control center.

In addition to network backhaul, in the case of electric utilities where climbing a pole with energized equipment has inherent safety risks to the operator, equipment manufacturers have begun integrating Bluetooth interfaces in their pole-mounted equipment. These Bluetooth-enabled devices allow the operator to connect to and change the configuration of certain equipment from a laptop in her company truck.

Another interesting electric utility anecdote is that in substations a utility must establish High Voltage Protection (HVP) to isolate external telephone and network connections from any high voltage sources inside the substation. Because of HVP's relatively high cost, as well as time to engineer and construct, some utilities have been known to deploy wireless connectivity to their substations to avoid implementing HVP.

CELLULAR BACKHAULS

Commonly used as a WAN because of its low cost and universal availability

There are known weaknesses to some of the cellular technologies, making them vulnerable to attack at the PHY/MAC layer

- GSM can be intercepted by setting up a fake base station
- Forces no encryption on connections
- <http://www.pittnerovi.com/jiri/hobby/electronics/gsm>

Most cellular carriers offer private network options, which provide logical separation of control data from public data

*relative to ISP options
MPLS based
etc.
if endpoint has a phone trying to talk unencrypted, it starts to talk clear text.*

Cellular backhauls are commonly used as a WAN backhaul because of their relatively low cost and universal availability. Technologies like GSM, CDMA, and LTE through local carriers are often chosen based on availability, coverage, and cost. It is important to note that there are known weaknesses to some of the cellular technologies like GSM, which makes them vulnerable to attack at the PHY/MAC layer. For instance, GSM can be intercepted by using approximately \$800 worth of hardware and software to set up a fake base transceiver station (BTS) for GSM devices to connect to. This fake BTS station can be configured to mimic any carriers' identifiers, tricking GSM devices to connect erroneously to this fake BTS. When initialized, the fake BTS can force the client GSM device to use A5/0 encryption (cleartext, or no encryption), thus exposing the communications while acting as a proxy to forward the traffic through an alternative connection. Most cellular technologies do not have such extreme weaknesses; however, even this can be mitigated for the most part with proper protections such as encryption and authentication at the upper network protocol layers.

Most cellular carriers offer private network options, which provide logical separation of control data from public data—this is highly recommended for control system use. This data separation technology is similar to VLAN and MPLS data separation, which should be considered a semi-private network link with data exposed to the ISP at a minimum.

References:

- <http://www.pittnerovi.com/jiri/hobby/electronics/gsm/>
- <http://www.youtube.com/watch?v=Cx8iWWg-Ch0>
- <http://www.youtube.com/watch?v=bO5McFJBg6k>
- <http://openbts.org/>

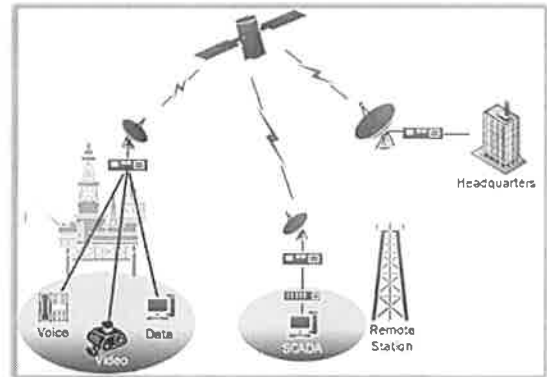
SATELLITE COMMUNICATIONS (SATCOM)

Often used for SCADA applications (oil fields, pipeline, electrical, and maritime applications – platforms, drilling ships, etc.)

- Inmarsat-C: Maritime and distress (GMDSS)
- FleetBroadband (FB): Maritime nav (ECDIS)
- SwiftBroadband: Aeronautical data and voice
- Classic Aero Service: Aeronautical data/voice
- VSAT: Fixed data and voice, often SCADA
- BGAN: Mobile data and voice, often SCADA

Security is not often part of the design...

- https://ioactive.com/pdfs/IOActive_SATCOM_Security_WhitePaper.pdf



Very Small Aperture Terminal (VSAT) provides reliable data, video, and voice communications from remote sites using (outdoor, indoor, and mobile) field units to relay communications from a satellite. VSAT applications are numerous, but many industrial uses include well-controlled data acquisition, pipeline SCADA, Electrical SCADA (transmission and distribution/AMI), wind farms, and maritime applications (oil platforms, drilling ships, and so on). VSAT provides a higher quality signal with adjustable bandwidth to meet needs, and can be cost-effective depending on the location and requirements. Terminals and equipment can be ruggedized and made specifically for industrial field applications.

VSAT provides the necessary flexibility to meet demands for SCADA/telemetry while also being able to provide voice phone services, used for video applications (security cameras). VSAT allows IP applications to move data reliably. Technology advancements (such as Forward Error Correction) have allowed VSAT networks much greater levels of availability and reliability. VSAT uses Time Division Multiple Access (TDMA) to capacity across multiple applications and allows for an assigned quality of service.

VSAT can operate in several frequency ranges to include C-Band (4–8 GHz), Ku-Band (12–18 GHz), and Ka-Band (26.5–40 GHz). There can be some overlap with other communication systems, such as WiMax.

Challenges can include some latency (measured in hundreds of milliseconds) over terrestrial broadband, which can be acceptable for many SCADA/telemetry applications. Planning for associated latencies (generally 600–700 ms) can allow for a number of applications. You need good Line of Sight (LOS) to the satellite and to maintain a position (can have issues with animals, landslides, etc.). There can be RF interference and some degradation of the signal due to rain/weather; in addition, lightning strikes can cause damage, and signals can be impacted by extreme solar activity.

End users can support their own virtual private network or pay for satellite services that provide Virtual Network Operator services to reduce cost and gain some of the benefits. Network Management Services allow for management of VSAT links and channels.

VSAT terminals can be configured to use AC power or DC power sources, such as solar panels. (Some remote applications require their own power source.)

VERY SMALL APERTURE TERMINAL (VSAT) TECHNICAL DETAILS

Security supports AES encryption and frequency hopping

Commonly used frequencies

- C-Band: Downlink 3.7–4.2 GHz, uplink 5.9–6.4 GHz
- Ku-Band (Americas): Downlink 11.7–12.2 GHz, uplink 14.0–14.5 GHz
- Ku-Band (Europe/Africa): Downlink 11.45–11.7 and 12.5–12.75 GHz, uplink 14.0–14.5 GHz
- Ka-Band: 18–40 GHz



VSAT networks are configured in one of these topologies

- Star topology with central uplink site such as a control center
- Mesh topology where VSAT terminals pass data directly to each other via satellite to minimize central uplink site
- Combination of both star and mesh topologies

The security of VSAT communication can be impacted by vulnerabilities at the endpoints (field units or hub), by interception or impacting the communication link itself, or by exploiting weaknesses in the protocol and services. As with any satellite-based wireless communications, physical hazards need to be considered. VSAT antennas need to maintain proper position and alignment with the relaying satellite. This requires unobstructed sky views and maintaining antenna position. The RF link can be impacted by unintentional or intentional RF interference and can degrade if having to transmit and receive through weather (such as rain or snow). VSAT, like any wireless link, can be vulnerable to interception and eavesdropping. Extreme solar activity can also disturb VSAT communications.

VSAT links can support Virtual Local Area/Private Networks and can support encrypting traffic. Several IP-based security controls can also be applied to the underlying traffic, but there are trade-offs and challenges to consider. IPsec is a logical solution, but the TCP headers may be encrypted, which may in older implementations result in latency-induced performance issues. VSAT links can support data encryption (SSL, SSH, PGP, and such), but this may cause additional challenges with the SCADA application. Many VSAT service providers rely upon bulk encryption of their channels (for example, DES 56 bit).

An attacker can purchase VSAT equipment and hack the terminal to intercept the communications and traffic of other users. The attacker needs to reverse engineer the embedded code on the terminal to tune in to different frequencies and time slots. A simple brute force attack can break the bulk encryption and allow the attacker to listen to the underlying communications.

Endpoint hub servers and computers and their applications must also be managed for security vulnerabilities. Several security incidents have included infections from network-propagating worms that were able to infect endpoint-attached computers across the VSAT link. VSAT link usage, measurable through network management applications and billing, was the first sign that an infection had occurred. Conflicker has cost a VSAT user or two a larger bill than they were expecting.

BROADBAND GLOBAL AREA NETWORK (BGAN)

A global satellite solution from Inmarsat

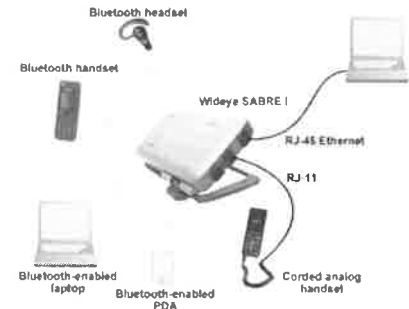
- Uses three geostationary satellites called I-4
- BGAN terminal is portable: About the size of a laptop
- Low power usage of 4 watts when idle to 22 watts transmitting

Three different services offered

- Standard service provides internet, phone, SMS, fax, and ISDN
- M2M service is low bandwidth commonly used for SCADA
- Link service is for fixed locations that require 5 to 30 gigabytes

Technical Details

- L-Band Transmit: 1626.5 MHz – 1660.5 MHz
 - L-Band Receive: 1525 MHz – 1559 MHz
 - Downlink and uplink speeds up to 492 kbps
 - Common latency is 800 ms to 1.5 seconds round trip
- } low frequency means low throughput
just use when there is no other option else.*



Broadband Global Area Network (BGAN) is provided by Inmarsat and uses three geostationary satellites called I-4 to provide almost global coverage. One advantage of BGAN terminals is their low power usage (4 watts idle to 22 watts burst transmitting) which is much less than satellite dish systems that use 90 to 150 watts when idle or transmitting.

There are three different services offered by Inmarsat:

- **Standard BGAN** service from Inmarsat provides the internet, phone, SMS texting, fax, ISDN, and streaming services.
- **BGAN M2M** Service, launched in February of 2012, is a low-bandwidth service for remote SCADA or M2M (machine-to-machine) equipment monitoring and control, offered in 2-, 5-, 10-, and 20-megabyte monthly service plans (Hughes 9502 BGAN terminal).
- **BGAN Link**, launched in March of 2012, is for **fixed** locations that require 5 to 30 gigabytes of data transferred per month (only available with Class 1 BGAN terminals [the Hughes 9201 or the Explorer 700/710]).

L-Band 1.5 to 1.6 GHz

Transmit: 1626.5 MHz – 1660.5 MHz

Receive: 1525 MHz – 1559 MHz

GPS frequency used on BGAN terminals is 1575.42 MHz (receive only)

Downlink speeds of high-end BGAN terminals are up to 492 kbit/s and upload speeds are also up to 492 kbit/s

Common latency is 1–1.5 seconds round-trip for the Background IP service. It is slightly better for the streaming services at 800 ms – 1 second

Reference:

http://en.wikipedia.org/wiki/Broadband_Global_Area_Network

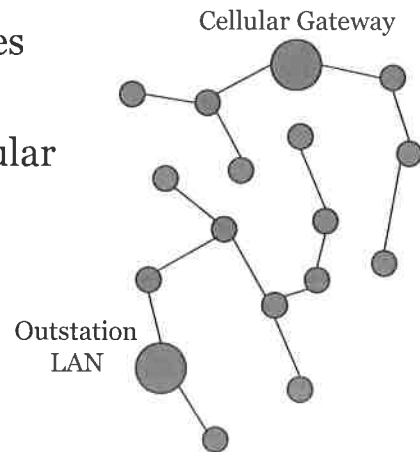
RF MESH NETWORKS

RF Mesh networks allow each participating device (or node) to route data to other devices

A small number of nodes act as a gateway to other, longer-distance networks such as cellular

Uses in ICS:

- WirelessHART 50-80% in market
- ISA100.11a
- ZigBee → in buildings for door automation



RF Mesh networks allow each participating device (or node) to route data to other devices, thus avoiding issues where all devices can't see a central access point. A small number of nodes act as a gateway to other nodes, allowing them to communicate longer distances until they reach a WAN network such as cellular.

Commonly used in wireless networks where:

There are large numbers of devices.

Devices have problems all seeing a central access point.

A more flexible "self-healing" network is preferred.

Latency is less of a concern.

Meshing is used in ICS protocols such as WirelessHART, ISA100.11a, and ZigBee.

WIRELESSHART (IEC 62591)

WirelessHART

Multivendor wireless standard for ICS released in 2007

Designed for process field device networks

Wireless Industrial Technology Consortium (WiTECK)

- Promoter members include: ABB, Emerson, Endress+Hauser, Pepperl+Fuchs, Siemens, and Softing

Based on existing HART communication protocol

Estimated 50–80% market share in wireless ICS

WirelessHART is also known as IEC 62591. It is a multivendor wireless standard created by ICS manufacturers for wireless sensor networks. It is managed by the Wireless Industrial Technology Consortium (WiTECK), which is made up of more than 37 different members. The WiTECK “promoter members” include well-known ICS manufacturers such as ABB, Emerson, Endress+Hauser, Pepperl+Fuchs, Siemens, and Softing.

WirelessHART is based on an existing network protocol called HART, or the Highway Addressable Remote Transducer Protocol. HART is a serial-based protocol like Modbus commonly used over 4–20 mA analog circuits. WirelessHART is basically a security-enabled wireless version of HART. All WirelessHART licensees use the same software code base to increase interoperability between hardware implementations, which is fairly uncommon in standards-based network protocols. However, this move appears to have greatly helped its adoption. Current estimates by market analysts place the WirelessHART market share between 50–80% for all wireless networks used in ICS networks, including Bluetooth, ZigBee, and Wi-Fi.

WIRELESSHART TECHNICAL DETAILS

Operates on the 2.4 GHz Industrial, Scientific, and Medical (ISM) band

Leverages IEEE 802.15.4 for PHY/MAC

1 Mesh network with channel hopping

0 Security

- Can be configured for unique join keys for each device
- Join key configured on field device maintenance port → only way. Not achievable.
- Join can be restricted by key, manufacturer, and product name/tag
- Successful join packet retrieves network key
- All AES-128 keys provided by automated key management
- All payloads encrypted with unique session key per device
- Rogue devices can't spoof other devices because of unique keys

Physical/MAC How you create the frame you send
like osm.cry. Every part has its own key.

Like Wi-Fi, Bluetooth, and Zigbee, the WirelessHART protocol operates on the 2.4-GHz ISM frequency band. WirelessHART leverages IEEE 802.15.4 (same as ZigBee) for its PHY/MAC (OSI Layers 1 and 2). On top of this, WirelessHART implements its own meshing protocol for "self-healing" capabilities, which facilitates deployment and limits single points of failures. WirelessHART also includes channel hopping, allowing it to work in the same physical area as other 2.4-GHz protocols.

As one of the more modern ICS protocols, WirelessHART built security into the protocol itself. WirelessHART can be configured so that every field device participating in the network can be configured with a unique join key. These join keys must be configured via a wired interface on each field device through the device's local maintenance port. All other keys needed for secure communications are automatically provided by a key management built into the network coordinator. To join a network, each device must use its unique join key to encrypt a join request packet. Successful authentication via this join packet provides the device with the shared network key, allowing it to join the mesh network. Administrators can further limit this device authentication by a unique device key, manufacturer, and product name/tag. After a device is on the mesh network, it will be provided a unique session key for it to encrypt its payload data. This prevents rogue devices that have somehow retrieved a join key and successfully accessed the mesh network from spoofing other devices. Because each device has its own session key, this provides a cryptographic-based nonrepudiation to verify data came from the device we believe it came from instead of some other rogue device on the same network.

ISA100.11A (IEC 62734)

Wireless standard developed by the International Society of Automation (ISA)

- 400+ automation professionals
- 250+ companies
- Standard development started in 2005



In 2008, WirelessHART granted ISA unlimited and royalty-free license to WirelessHART

- Was hoped to be adopted as ISA100 standard
- However, ISA100 was set on application-level neutrality

ISA100.11a was released in 2009

ISA100.11a is another wireless standard developed by the International Society of Automation (ISA) during the same time period that WirelessHART was being developed. ISA created the ISA100 committee in 2005, which consists of 400+ automation professionals from 250+ companies. After WirelessHART was ratified and released in 2008, the HART Communication Foundation (HCF), owners of WirelessHART, granted the ISA an unlimited and royalty-free license to WirelessHART. The hope was to build compatibility and conformance between the two standards; however, ISA100 had already made several design decisions which made it incompatible with the HART protocol. The largest of these incompatibilities was the use of 6LoWPAN to gain the benefits of IPv6, which is not used by the HART protocol. ISA100 released its first standard named ISA100.11a in 2009.

ISA100.11A TECHNICAL DETAILS

Similar to the WirelessHART standard with the following major differences:

- Upper layers are based on 6LoWPAN instead of HART
 - 6LoWPAN is an embedded version of IPv6, which compresses headers
 - Can tunnel legacy protocols
 - Provides subnets and higher scalability
 - Additional support for quality of service (QoS)
- Configurable 10–14 ms time slots for device communication windows (vs. 10 ms)
- Support for asymmetric join methods and over-the-air (OTA) device configuration

IPv6 for low band usage

ISA100.12 subcommittee was created to converge ISA100.11a and WirelessHART into a single standard, but the effort was abandoned in 2013

ISA100.11a is similar to the WirelessHART standard in their lower layers. They both use IEEE 802.15.4 and Direct Sequence Spread Spectrum (DSSS) for their PHY layer with Time Division Multiple Access (TDMA) and a meshing topology for their MAC layer; however, beyond that point, the two standards greatly deviate. ISA100.11a uses 6LoWPAN instead of HART for its network stack. This allows ISA100.11a to use IPv6 for its network layer, which provides subnetting, higher scalability, and the ability to tunnel legacy protocols. ISA100.11a also provides additional support for quality of service (QoS) and the ability to configure time slots greater than 10ms. Finally, ISA100.11a removed the requirement to configure security keys directly on the device via a maintenance port and instead offers support for asymmetric join methods and over-the-air (OTA) device configuration.

Having two different ICS standards has caused confusion and frustration in the ICS industry. As a result, subcommittee ISA100.12 was created to converge ISA100.11a and WirelessHART into a single standard; however, this effort was abandoned in 2013.

Reference:

For additional information concerning the differences between ISA100.11a and WirelessHART, please read this article from ISA:

<https://www.controlglobal.com/articles/2012/nixon-wireless-isa/>

ZIGBEE

Similar to WirelessHART and ISA100.11a

- Leverages IEEE 802.15.4 for PHY/MAC
- Usually uses 2.4GHz, but sometimes use sub-GHz
- Made for low-cost, low-power mesh networks →
- Certification requires 2+ years on a single battery

*certified
zigbee device
battery lasts in years*



ZigBee®

ZigBee uses application profiles to extend to different markets

- Home Automation, Smart Energy, Telecom, Health Care, RF4CE, Remote Control, Light Link, IP, Building Automation, Gateway, Green Power, Retail Services
- Can contain different security features, packet sizes, tunneled protocols, etc...

*install config template and
get different features*

Application profiles most used in ICS

- Smart Energy (1.x and 2.x): Smart meters to home automation links
- Building Automation: BACnet over ZigBee, used by Honeywell, Ingersoll-Rand/Trane, Johnson Controls, Schneider Electric, and Siemens → *embedded zigbee in devices*
- ZigBee IP: General-purpose implementation of 6LoWPAN (aka IPv6) over ZigBee

includes etc.

ZigBee is another emerging wireless technology, based on the IEEE 802.15.4 specification. Similar to Bluetooth technology, ZigBee is a competing wireless specification designed for replacing cables to some degree, but more for creating mesh networks. It also targets specific vertical markets instead of general consumers. While Bluetooth is attractive as a general-purpose cable replacement technology, ZigBee is designed for use in product tracking, medical device monitoring, industrial sensor monitoring, control networks, and home automation systems. Unlike Bluetooth, ZigBee is designed to be a simple protocol implementation, requiring less memory and processor resources to deploy ZigBee technology. (A complete ZigBee implementation is distributed by some vendors on a single 128-Kbyte memory card.) ZigBee is generally used in implementations that require low power consumption and rely on long multiyear battery life.

One example of a ZigBee deployment has been publicized by Honeywell International, embedding ZigBee radios in HVAC systems. With ZigBee wireless support, an administrator can connect to the HVAC device with a ZigBee client card to manage and monitor maintenance history, utilization, and control information.

References:

<http://www.zigbee.org/>

<https://en.wikipedia.org/wiki/Zigbee>

ZIGBEE SECURITY

Security optional

- AES may be too resource-intensive for lightweight devices
- Balance between battery life and security
- Can be mandatory for some application profiles

Encryption based on AES128 keys and CCM*

- Similar to Wi-Fi WPA2's AES-CCMP
- CCM* extends CCM by adding encryption-only and integrity-only options

Relies on master keys set by manufacturer, installer, or end user

- AES-128 network key must be set on each device
- Secondary link keys may be used in some profiles after network key established
- Some profiles add application-layer security like ECC in Smart Energy Profile

ECC for low power

KillerBee is a Python-based framework initially written by Joshua Wright

- Can capture, inject, and exploit ZigBee packets
- Limited ability to work with other IEEE 802.15.4-based protocols

The ZigBee specification has a section dedicated to the security of ZigBee networks, accommodating security at the MAC, network, and application layers. This allows ZigBee application developers to have extra flexibility in where they implement security functions, relying on security at the MAC layer, or at the upper-layer network and application functions. For simplicity, however, the ZigBee specification requires that the same key be used for all three layers of security, implying that if an attacker has compromised the MAC layer link key, he will also be able to decrypt traffic at the network and application layers using the same key.

The selection of a key is done at installation time, set by the manufacturer, installer, or end user. After specifying an initial "master" key, two ZigBee devices will establish a mutual link key that is used to authorize other ZigBee nodes and to encrypt and decrypt traffic. This is beneficial to the end user, as a compromised link key will reveal only the data between two nodes; it does not also reveal the secrecy of the master key and link keys used between other ZigBee devices.

KillerBee is a Python-based framework initially created by Joshua Wright to explore and exploit the security of ZigBee and other IEEE 802.15.4-based protocols.

Reference:

For more information on KillerBee, visit its project page: <https://github.com/riverloopsec/killerbee>

low amount of people control it
that's why it
is insecure.

PROPRIETARY WIRELESS PROTOCOLS

Often referred to as ISM Band and Microwave solutions

Vendors often leverage proprietary protocols for these links

- May not be TCP/IP-based
- Most do not use encryption at the PHY/MAC layers
- Vendors usually very reluctant to discuss the security details
- Frequency hopping and licensed bands often cited as their "security"
 - Good for communication benefits
 - Shouldn't be considered layers of security

encryption is
a must!
any protocol is insecure. Only
key + protect data
using encr.

Communications are only as strong as the upper layer protocols chosen by the vendor

- Enable security features offered by the vendor
- Give preference to solutions that leverage encryption and authentication in their upper layer protocols, even if link-layer encryption is provided

Vendors often offer field devices that use wireless technologies like communications over the ISM bands, RF meshing technologies, and Microwave communications. Most of these solutions leverage proprietary protocols for these links, most of which are not TCP/IP-based. This is important because the security capabilities of many TCP/IP protocols are well known; however, proprietary protocol security capabilities have to be taken at face value from the marketing team. Promises made in marketing literature are not always accurate, and proprietary protocols are not tested to the same level common TCP/IP-based protocols are. This does not mean proprietary protocols are inherently less secure, only that they pose a security unknown.

With these types of wireless communications, don't expect security at the PHY/MAC layers like we have with Wi-Fi and Bluetooth. PHY/MAC security is not often included with proprietary wireless communications. Because of this, communications across these links are only as strong as the upper-layer protocols chosen by the vendor. Give preference to solutions that leverage encryption and authentication in their upper-layer protocols, even if security is being offered at the PHY/MAC layer. Often, vendors claim that the use of frequency hopping and licensed RF bands provide security. While these technologies are good for communication benefits such as increased bandwidth, increased transmission power, and decreased noise, these technologies are superficial security defenses and easily bypassed. Frequency hopping and licensed RF bands should not be considered security defenses.

Finally, to have the most secure wireless communications possible, enable any security features offered by the vendor after carefully testing for compatibility with your ICS communication requirements.

BLUETOOTH

Wire replacement technology

- No line-of-sight requirement
- Up to seven simultaneous connections
- Supports data, voice, and content-centric applications with Bluetooth profiles

Multiple versions exist

- 1.0, 1.1, 1.2: Speeds up to 0.7 Mbit/sec
- 2.0: Speeds up to 2.1 Mbit/sec
- 2.1: Speeds up to 3 Mbit/sec, Secure Simple Pairing (SSP)
- 3.0: Speeds up to 24 Mbit/sec
- 4.0, 4.1, 4.2: Bluetooth Low Energy (BLE) with AES-CCM introduced (1 Mbit/sec) and options extended
- 5: Doubles BLE speed (2 Mbit/sec) and range

Bluetooth gained significant traction in ICS devices since 2010

- Often being used in field technician handheld devices
- Allows technicians to take measurements without entering safety zones needing PPE



↓
generally used

Bluetooth was first announced in 1998. It promised to be an affordable, low-power wireless solution that would be attractive as a cable-replacement technology. Using Bluetooth, vendors would no longer need to ship unique networking cables with their products, relying on an inexpensive, integrated wireless card for all connectivity needs.

Unlike infrared networks, which had limited success as a cable-replacement technology, Bluetooth has no line-of-sight requirements, making it much more flexible and user-friendly. Supporting data, voice, and content-centric applications such as streaming video or other data sources, Bluetooth networks meet a wide range of functionality requirements, making them attractive to different connectivity needs. Capable of supporting up to seven simultaneous connections, a single Bluetooth adapter can communicate with several nearby devices simultaneously without being forced to connect and disconnect from different networks. With several billion Bluetooth devices being deployed in 2010, organizations will continue to see growth in Bluetooth devices.

Secure Simple Pairing (SSP) was introduced in version 2.1 and provides a variety of different methods to pair devices, including:

- Just Works: No user interaction required (most headphones use this)
- Numeric Comparison: User confirms if the numbers on both devices match (most cars use this)
- Passkey Entry: User must enter some code on one of the devices
- Out of band (OOB): Uses different technology to exchange info needed to pair, such as NFC (latest headphones using this)

In the case of electric utilities, where climbing a pole with energized equipment has inherent safety risks to the operator, equipment manufacturers have begun integrating Bluetooth interfaces in their pole-mounted equipment. These Bluetooth-enabled devices allow the operator to connect to and change the configuration of certain equipment from a laptop in her company truck.

BLUETOOTH SECURITY

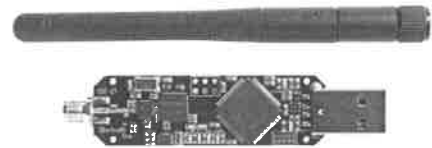
Encryption based on Eo stream cipher

- Based on PIN which can be 4–16 characters in length
- Key generated using PIN in key derivation based on SAFER+ block cipher
- Keys are used to authenticate Bluetooth "peers" and to encrypt transmission data
- Largest weakness exists during pairing process

vulnerable when they are pairing

Ubertooth is an open source hardware/software project for assessing Bluetooth

- Can capture all traffic: 1600 packets per second across 79 channels
- Some devices must use fixed PINs
- Sniffing risk when devices first pair
- Presentation link below



Bluetooth security provides a simple (from the end-user perspective) means of authentication and data encryption when communicating between devices. Authentication starts with a user selecting a personal identification number (PIN) to authenticate other devices in the Bluetooth piconet ("small network"). The PIN is then entered manually into each device that needs to communicate with other Bluetooth-capable devices.

Bluetooth security relies on the secrecy PIN selected by the user and the MAC address of the Bluetooth device, known as a BD_ADDR (pronounced "bee dee adder"). When two devices connect for the first time, they use the PIN and BD_ADDR information to generate permanent link keys that are stored on each device. Subsequent communication between the devices does not require the PIN input from the user again, instead relying on the stored link keys for authentication and encryption.

In many cases, however, Bluetooth devices have security requirements or a desire for a secure operating environment without the ability to uniquely select a PIN. This is often found with Bluetooth Headset devices that do not have a Human Machine Interface (HMI). The lack of an HMI results in a fixed PIN selection for the headset, commonly "0000" or "1234."

Since the PIN is needed only at the initial pairing of devices, the risk of a static PIN is a target for an attacker when the devices initially pair. The Bluetooth Special Interest Group (SIG), which certifies and specifies how Bluetooth networks should interoperate, initially dismissed this issue by recommending customers only pair devices in a trusted environment, free from the threat of unknown attackers sniffing the wireless medium. However, recent research published by the University of Tel Aviv, Israel, by Yaniv Shaked and Avishai Wool presented findings that could be exploited by an attacker to force Bluetooth devices to re-pair in a hostile environment to mount an attack against the Bluetooth PIN.

Reference:

Presentation on use of Ubertooth: <https://www.youtube.com/watch?v=HU5qi7wimAM>

IEEE 802.11 WIRELESS (WI-FI)

, laptop to another
Supports ad-hoc and infrastructure networks
Supports roaming, fragmentation, and reliable data delivery (positive acknowledgment)



Branched into:

- 802.11b supports up to 11 Mbps @ 2.4 GHz
 - 802.11a supports up to 54 Mbps @ 5 GHz
 - 802.11g supports 22–54 Mbps @ 2.4 GHz
 - 802.11n supports 54–600 Mbps @ 5 GHz, 2.4 GHz
 - 802.11ac supports 433–1300 Mbps @ 5 GHz
- } more than 1 antenna.*

Some vendors like ABB-Tropos and Firetide are providing Wi-Fi mesh solutions for ICS and IoT (802.11S)

The 802.11 standard was approved in 1997 by the IEEE 802 Committee. This standard has several key elements that make it the most widely adopted wireless LAN standard in use today:

- Supports ad-hoc and infrastructure networks.
- Accommodates roaming between multiple access points without losing connectivity.
- Supports large data packets through the use of fragmentation at Layer 2.
- Provides reliable data delivery when experiencing interference due to a requirement to positively acknowledge all data traffic received from an access point or wireless station.
- Builds on existing standards for data encapsulation (802.2 LLC).
- Power conservation techniques extend the battery life of wireless devices.

Reference:

<https://standards.ieee.org/search-results.html?q=802.11>

it can authenticate you before giving you IP addr.

WI-FI SECURITY

WEP (Wired Equivalent Privacy) released in 1997

- Based on RC4 encryption with CRC-32 checksum and a 24-bit IV, which causes IV reuse
- Multiple weaknesses and exploit optimizations allow WEP to be cracked in minutes

WPA (Wi-Fi Protected Access) released in 2003 as subset of 802.11i

- A temporary solution that was hardware compatible with WEP (firmware updatable)
- Uses Temporal Key Integrity Protocol (TKIP) as replacement for CRC-32, but there are issues

WPA2 released in 2004 as full implementation of 802.11i

- Required access point (AP) and network card (NIC) replacement (AES-CCMP)

WPA2 Enterprise, based on 802.1X authentication, removed the shared key

- PEAP (Protected Extensible Authentication Protocol) authenticates to RADIUS or Active Directory
- EAP-TTLS (EAP Tunneled Transport Layer Security) uses client/server certificate pairs per device

WPA3 was released in January 2018 and devices began certification in October 2018

- WPA3 Personal gains password-less encryption on open networks and password guessing defenses
- WPA3 Enterprise gains stronger cryptography with GCMP-256, HMAC-SHA384, and ECDSA

Just these are not be enabled. Other are useless. longer key length

The Wi-Fi Protected Access (WPA) specification was adopted by the Wi-Fi Alliance before the IEEE 802.11i specification was completed to give organizations an opportunity to improve the security of wireless networks. In 2003, many organizations were becoming increasingly concerned about the security of wireless networks, without a clear solution from the IEEE to replace WEP. Although the IEEE 802.11i committee had formalized a replacement for WEP, the 802.11i specification was incomplete.

The Wi-Fi Alliance adopted the 2003 draft of the 802.11i specification and started performing interoperability testing for vendors using the Temporal Key Integrity Protocol (TKIP). This testing process certified vendor products as WPA-compliant, focusing on the implementation of TKIP as a mechanism to replace WEP on existing hardware. After the 802.11i specification was ratified in June 2004, the Wi-Fi Alliance also adopted the AES-CCMP cipher mechanism designed for new hardware. The testing process for compliance with AES-CCMP became known as WPA2.

In 2017, KRACK (Key Reinstallation Attack) exploits flaw in third step of CCMP 4-step handshake. To fix this, you must patch either the AP or Host to protect link against KRACK.

WEP (Wired Equivalent Privacy) – https://en.wikipedia.org/wiki/Wired_Equivalent_Privacy

WPA (Wi-Fi Protected Access) – https://en.wikipedia.org/wiki/Wi-Fi_Protected_Access

KRACK (Key Reinstallation Attack) – <https://en.wikipedia.org/wiki/KRACK>

EAP Types – https://en.wikipedia.org/wiki/Extensible_Authentication_Protocol

TAKEAWAYS AND RECOMMENDATIONS

Section takeaways

- Wireless is far greater than just Wi-Fi and Bluetooth
- Used at all Purdue levels for control, supervisory, and remote access

Recommendations to owner/operators

- Identify everywhere in the control network you use wireless
- Create and enforce policies on use and configuration of each wireless technology

Recommendations to vendors

- Only implement wireless solutions in their most secure state
- Support for legacy wireless should require extra step to disable security

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. Enforcement Zone Devices
 - Firewalls and NextGen Firewalls
 - Data Diodes and Unidirectional Gateways
2. Understanding Basic Cryptography
 - Crypto Keys
 - Encryption, Hashing and Signatures
3. Wireless Technologies
 - Satellite and Cellular
 - Mesh Networks and Microwave
 - Bluetooth and Wi-Fi
4. Wireless Attacks and Defenses
 - 3 Eternal Risks of Wireless
 - Sniffing, DoS, Masquerading, Rogue AP
5. **Exercise 3.1: Network Forensics of an Attack**
6. Purdue Level 2 and 3 Attacks
 - Historians and Databases
 - **Exercise 3.2: Bypassing Auth with SQL Injection**
 - HMI and UI Attacks
 - Web-based Attacks
 - Password Defenses
 - **Exercise 3.3: Password Fuzzing**

This page intentionally left blank.

Wireless Attacks and Defenses

Applicable Standards:

- **NIST CSF v1.1:** DE.CM-1
- **ISA/IEC 62443-3-3:2013:** SR 6.2
- **NIST SP 800-53 Rev. 4:** AC-2, AU-12, CA-7, CM-3, SC-5, SC-7, SI-4
- **CIS CSC:** 1, 7, 8, 12, 13, 15, 16
- **COBIT 5:** DSS01.03, DSS03.05, DSS05.07

This page intentionally left blank.

ICS WIRELESS DISADVANTAGES

The three eternal truths of wireless security

- Denial-of-Service attacks are easier and near impossible to defend against
- Network capture is possible, regardless of frequency or hopping techniques
- Attacker has at least a limited ability to communicate on the wireless network

Focused security defenses on the higher-level network protocols

- Use lower-level defenses if available (Example: Wireless authentication/encryption in cellular)
- Encrypt data at application layer or use VPN, if possible

ICS networks sometimes rely upon wireless communications. When this occurs, we should always assume the following weaknesses are inherent in our system:

- Denial-of-Service attacks are much easier and near impossible to defend against. As we discussed earlier, it is easy to create wireless noise on the same frequencies our wireless communications use.
- Network capture is possible, regardless of RF frequency used or use of hopping technologies. While frequency hopping algorithms help increase the difficulty of traffic capture, its true purpose has more to do with increased bandwidth and decreased noise than security. With enough time, an experienced wireless attacker will be able to follow your hopping pattern. For a great example of this, check out the research Mike Ossmann did with Bluetooth and his resulting Ubertooth project. Bluetooth is one of the fastest hopping and widest channel commercial protocols available, and it can be tracked between hops.
- An attacker has at least a limited ability to communicate on the wireless network. While authentication and encryption mechanisms at various protocol layers may inhibit full communications on the network, the authentication process and decryption routines both offer inputs that attackers can fuzz and attempt to find vulnerabilities in.

Security defenses should be focused primarily on the higher-level network protocol because we should assume that most wireless protocols allow at least partial access to the MAC layer.

we see how Point to Point Microwave in ICS. One station to another one.

Microwave

ITU-DEFINED ISM BANDS

Frequency Range		Bandwidth	Center Frequency	Notes
6.765 MHz	6.795 MHz	30 kHz	6.780 MHz	Subject to local acceptance
13.553 MHz	13.567 MHz	14 kHz	13.560 MHz	
26.957 MHz	27.283 MHz	326 kHz	27.120 MHz	
40.660 MHz	40.700 MHz	40 kHz	40.680 MHz	
433.050 MHz	434.790 MHz	1.84 MHz	433.920 MHz	Region 1: Europe, Africa, Middle East, Russia, Mongolia, but most accepted worldwide
868.000 MHz	870.000 MHz	2 MHz	869.000 MHz	Not ISM, but SRD band for Europe and India
902.000 MHz	928.000 MHz	26 MHz	915.000 MHz	Region 2: Americas, Greenland, and Pacific Islands
2.400 GHz	2.500 GHz	100 MHz	2.450 GHz	(Used by Wi-Fi, Bluetooth, and ZigBee)
5.725 GHz	5.875 GHz	150 MHz	5.800 GHz	(Used by Wi-Fi)
24.000 GHz	24.250 GHz	250 MHz	24.125 GHz	
61.000 GHz	61.500 GHz	500 MHz	61.250 GHz	Subject to local acceptance
122.000 GHz	123.000 GHz	1 GHz	122.500 GHz	Subject to local acceptance
244.000 GHz	246.000 GHz	2 GHz	245.000 GHz	Subject to local acceptance

SANS

The table above is a list of the common RF frequencies set aside across the world for industrial, scientific, and medical (ISM) purposes. Another way to read this is any device that doesn't pay a lot of money to purchase its own bandwidth uses these frequencies. These frequencies are the first place attackers look for proprietary wireless communications because 95% of all our RF communications use these frequencies. Next to Wi-Fi's and Bluetooth's 2.4 GHz, the most common here in the United States is the 915-MHz band, which is really 902–928 Mhz, but usually called by its center band frequency, followed by the 433.920-MHz band. In Europe, the most common band to look for proprietary traffic is 869 MHz.

One RF technology we often use in the ICS world is microwave, which is loosely defined as anything between 3 GHz and 300 GHz, but in some cases, it may mean frequencies as low as 300 Mhz. When discussing microwave frequencies with vendors, it's usually best to get the actual frequency range clarified at the beginning of the conversation.

WIRELESS DOS

Attackers can identify frequencies used in communications and generate noise on those frequencies

- This is possible on ISM bands and microwave frequencies, not just on Wi-Fi technologies
- Barrier to entry for attackers on ISM and microwave frequencies is only marginally higher than on Wi-Fi

If the attacker understands the PHY/MAC wireless protocol being used, targeted disconnect and renegotiation techniques might be possible

- Barrier to entry for attackers here is greater because attack tools do not exist outside of Wi-Fi, Bluetooth, and ZigBee
- The level of effort for an attacker to create a tool for a new protocol is usually a few weeks and should be assumed possible

! when they re-connect
attacker can see
handshake

Attackers can identify frequencies simply by researching the device they are targeting or using a spectrum analyzer near the target device to measure the frequency it is using. Once the frequency is known, attackers can use special hardware such as software-defined radio (SDR) devices to generate noise on those frequencies, thus prohibiting viable communications. This is possible on both ISM bands and microwave frequencies, not just on Wi-Fi technologies. Barrier to entry for attackers on ISM and microwave frequencies is only marginally higher than on Wi-Fi, as the cost for SDR devices is quickly dropping, and is currently at \$20 for spectrum analysis and data capture and \$300 for broadcast ability. With appropriate electronics experience, jamming devices for specific frequencies can often be made for \$20–\$30.

If the attacker understands the PHY/MAC wireless protocol being used and has the right hardware and software, they may be able to perform targeted disconnect and renegotiation techniques in the target protocol. However, a barrier to entry for this type of attack is greater than simply DoS via frequency jamming because attack tools with this ability do not exist for most network protocols outside of Wi-Fi, Bluetooth, and ZigBee. But it would be wise to remember that the level of effort for an attacker to create a tool for a new protocol is usually a few weeks and should be assumed possible; and, once created, it can be disseminated on the internet very quickly. This reinforces the need to protect wireless protocols with strong security defenses.

WIRELESS PACKET CAPTURE

Wireless data is encoded on wireless signal carriers through a process called modulation

Once a wireless signal is captured, the attackers only need to demodulate the signal to recover the binary packets

Frequency hopping makes this demodulation much harder, but it is still very possible for attackers

Amplitude Modulation

1111000101100010001



Frequency Modulation

1111000101100010001



From a conceptual perspective, RF communications are relatively simple to understand. Wireless data is encoded on wireless signal carriers through a process called modulation. There are two primary types of modulation. The first is amplitude modulation (AM) which changes the power of the signal to encode data. For instance, a power level of 2.0 is equal to a binary 1 and a power level of 1.0 is equal to a binary 0. The second major type is frequency modulation, which slightly changes the frequency of the carrier wave to signify data. For instance, if the carrier wave is 915 Mhz, then a 916 Mhz could be a binary 1 and a 914 Hhz could signify a binary 0. Although both of these examples are overly simplistic, they give you enough information to understand the basic difference between AM and FM modulation and how they could each encode binary data on them.

Once an attacker captures a wireless signal, he only needs to demodulate the signal to recover the binary packets being sent. Often, attackers can determine the modulation technique by researching the device online and finding the information in vendor documentation, FCC certification documents, or patent filings. Attackers can also sometimes find this information through a physical attack on the device itself, but if worse comes to worst, the attacker can use his knowledge and try every modulation type to reverse the signal.

Frequency hopping makes this demodulation much harder, but it's still very possible for attackers to reverse, as discussed a few pages ago.

WIRELESS PACKET TRANSMISSION

Once an attacker knows the modulation being used, they can

- Extract data stream from signal
- Convert data stream to packets and related communications
- Reverse the processes to transmit

Software-defined radios (SDR) can be used for both capture and transmission

- The software is free and open source (GNU Radio, Universal Radio Hacker)
- Hardware for capture-only starts around \$20 (RTL-SDR)
- Hardware for capture and transmission starts around \$300 (HackRF)

Security at the PHY/MAC layer such as Wi-Fi WPA2

- Can limit the transmission attack surface
- But cannot eliminate it
- This reinforces the need for security defenses at the higher protocol layers

Once an attacker knows the modulation being used, he only needs to reverse the processes to transmit. Tools such as the GNU Radio software and special hardware called software-defined radios (SDR) can be used for both capture and transmission.

The software is free and open source.

The hardware can be purchased for as little as \$20 for capture and \$300 for transmission.

Security at the PHY/MAC layer, such as Wi-Fi uses, can limit the transmission attack surface, but not eliminate it. This once again reinforces the need for security defenses at the higher protocol layers.

WIRELESS SECURITY RISKS

Four greatest security risks when using any wireless technology

- **Eavesdropping:** Capturing the traffic
- **Masquerading:** Pretending to be your wireless network or devices
- **Denial of Service (DoS):** Blocking your traffic
- **Rogue APs:** Secret wireless links back to your network

As the popularity of wireless networks increases, their inherent security flaws are receiving more and more attention. In recent years, wireless security (or lack thereof) has become the press's media darling. Unfortunately, these reports are often incomplete or incorrect, and they often leave organizations with a false sense of security. This section focuses on the most critical security issues related to wireless networking:

- Eavesdropping
- Masquerading
- Denial of Service (DoS)
- Rogue APs

EAVESDROPPING

Attack software available to do this with common protocols

- WI-FI: Kismet and many others
- Bluetooth: Ubertooth
- ZigBee: Killerbee
- GSM: gr-gsm ← *sniffing GSM is
federal crime*

All these tools are on Control Things Platform; you just need hardware

Defenses

- Use whatever security is available in the wireless protocol
- Use SSL/TLS or VPN for passing sensitive application information

This page intentionally left blank.

MASQUERADING

An attacker spoofs the identity of a legitimate node or AP

- Tricks unsuspecting users into giving up sensitive information

Common attack tools to do this

- Wi-Fi: Karmesploit (part of Metasploit)
- GSM: OpenBTS

General Defenses

- Use SSL/TLS or VPN for passing sensitive application information

Wi-Fi defenses

- Educate users not to accept self-signed certificate warnings
- Use mutual-authentication wireless protocols like PEAP or TTLS

Masquerading describes the activities of an attacker who impersonates the identity of legitimate nodes or access points in a wireless network. The attacker accomplishes this by spoofing identity information to impersonate an otherwise authorized client or access point. By making clients think that they are communicating with a legitimate access point, attackers can trick unsuspecting users into giving up sensitive information, trick access points into believing that they are authorized clients, or launch Denial-of-Service attacks.

To bypass this method of access control, an attacker can simply masquerade as an authenticated client by changing his MAC address. Sometimes, when combined with a DoS attack against the impersonated system, the attacker changes his MAC address to a system that was actively communicating on the network. The captive web portal system checks the traffic's MAC address and, unable to differentiate the attacker from the legitimate user, grants the attacker unrestricted access to the victim network.

To protect against masquerading attacks, you must have some mechanism in place to authenticate users to an access point and authenticate the access point to users. By requiring the access point to present authentication credentials to the user, it is possible to mitigate attacks such as those implemented in the AirSnarf tool. This is possible using the IEEE 802.1X network authentication protocol with extensible authentication protocols that support mutual authentication, such as EAP/TLS, PEAP, and TTLS. However, in many cases, implementing mutual authentication protocols is impractical. For instance, hotspot locations cannot require that clients have 802.1X clients configured on their workstations and therefore must seek alternatives. SSL or TLS encryption protocols that utilize public-key infrastructure with digital certificates can be an alternative for hotspot access and other web-based applications. Using digital certificates to authenticate the web server or captive web portal system makes it much more difficult for an attacker to masquerade his identity as the legitimate network resource.

Unfortunately, many users have grown anesthetized by digital certificate warnings, and simply click through warnings generated by web browsers warning of mismatched digital certificates. This gives the attacker the opportunity to bypass this security mechanism and impersonate the characteristics of the digital certificate in an attempt to collect private information. It is critical for system administrators to successfully implement SSL and TLS systems with current digital certificates and educate users about the potential dangers of ignoring invalid certificate warnings.

DENIAL-OF-SERVICE ATTACKS

RF jamming techniques and tools are readily available

- Frequency hopping tries to help, but does little against today's technology
- Per-packet authentication rarely exists in wireless, allowing rogue disconnect signals
- Other weaknesses in protocol specifications can create additional DoS weaknesses

Defenses

Understand the impact of a DoS attack against your environment and prepare a response strategy

- Clearly understand the impact of such an attack against your network
- What is the effect of a DoS attack against your production networks?
- What alternatives will you pursue to reestablish connectivity to mission-critical systems?

Wireless intrusion detection systems can be an option in some cases

- Commonly available in Wi-Fi
- Not widely available in other wireless technologies... yet

Wireless networks are an easy target for Denial-of-Service attacks. Vulnerabilities in the wireless specifications, flaws in the firmware of popular wireless cards, and weaknesses in the nature of radio communications offer attackers opportunities to shut down wireless networks at their leisure. If an attacker has a powerful enough transceiver, he can generate so much radio interference that the targeted wireless network is unable to communicate effectively. Attackers can purchase or build RF-jamming equipment, which is effective at stopping all wireless activity, often covering all commonly used frequencies. These RF jamming attacks are specification-agnostic; they are equally effective against 802.11 and Bluetooth networks, as well as any other communication that uses the same frequency as the attacker.

Using commodity wireless cards, attackers can masquerade their identity as legitimate stations or access points to launch DoS attacks. Because most wireless specifications do not include any per-packet authentication mechanism, access points and stations do not have a way of verifying that each packet is indeed sent from its reported source address. Attackers utilize this weakness to send spoofed packets to victim clients on behalf of the access point, telling the victim to disconnect from the network. The victim station processes this packet as if the traffic was sent from the access point and disconnects from the wireless network. An attacker can send a sustained flood of these disconnect packets to the LAN broadcast address, thereby causing all stations to disconnect from the network, resulting in a sustained DoS attack.

The best mitigation strategy for DoS attacks against wireless LANs is to clearly understand the impact of such an attack against your network and prepare an appropriate response strategy. What is the effect of a DoS attack against your production networks? In the event that you are under attack, what alternatives will you pursue to reestablish connectivity to mission-critical systems?

To quickly identify and assess the impact of DoS attacks, organizations should consider deploying wireless intrusion detection systems using commercial or open source tools. Wireless IDSs allow administrators to react quickly to attacks against their network and might provide enough information to identify and locate attackers. While these WIDS devices are commonly available for Wi-Fi, they are not usually available for other wireless technologies commonly found in the ICS world. Vendors are working to create such systems, which may make sense for deployment at major remote stations and takeout points; however, these systems are unlikely to ever be feasible for all wireless endpoints on large ICS grids such as the energy sector.

ROGUE ACCESS POINTS

Unauthorized APs connected to a private network

- Users may have deployed Wi-Fi APs out of ignorance
- Often installed with default settings, and no security
- Permit full access to a network for attackers that find them
- Contribute to unauthorized information disclosure
- Attackers can deploy their own using cellular or other protocols, making them harder to find

Defenses

- Perform routine audits looking for them
- Wi-Fi-based rogue APs can be found with audit tools like Kismet
- Spectrum analyzers with directional antennae can find non-Wi-Fi devices
- Port scanning might reveal indicators like web-based admin pages
- Network captures can show traffic with hop counts set too low

A rogue access point (AP) is connected to a wired network without the authorization to provide wireless access to end users or attackers. And of all the wireless protocols we have discussed, Wi-Fi is the most commonly used for this. Users who want wireless access or are unhappy with existing wireless services expose an organization's network by connecting an access point to the wired networks. These access points are commonly meant for home use and rarely offer anything beyond the most basic security settings. Attackers who identify these rogue access points can exploit the basic security settings and gain access to internal network resources. Administrators are often unaware of rogue access points on their network until they are discovered as part of a vulnerability assessment or system compromise analysis.

Rogue access points can also be intentionally dropped in your network by attackers. If an attacker wants to be stealthy, they can use non-Wi-Fi APs like cellular or proprietary RF devices bridging the attacker into your network.

Rogue access points with little or no security pose an obvious threat to any organization and are usually only an issue with Wi-Fi. Often, organizations that have adopted "no wireless" policies are plagued with rogue access points due to the lack of any wireless access available to users. To mitigate rogue Wi-Fi access points, administrators should perform rogue AP detection using commercial or open source tools like Kismet and Wireshark. This method requires an administrator equipped with a laptop or handheld device to walk through the hallways and offices of all the buildings that might be risks for rogue access points to detect any unauthorized wireless activity. Unfortunately, this can be a difficult venture for large campuses, which require alternative detection methods.

Scanning wired networks for characteristics that resemble wireless access points is an alternative, yet less-reliable method of detecting rogue access points. Using vulnerability assessment tools such as Nessus, an administrator can scan all the nodes on the wired network to identify webpages, login banners, and other characteristics to identify potential rogue access points. This method is useful in detecting users who are not trying to hide their activity with otherwise stealthy tactics, such as shutting off ICMP echo responses and disabling administrative interfaces that might be used to identify them.

To detect rogue APs based on other wireless technologies, a spectrum analyzer with a directional antennae can be useful to find these devices; however, the knowledge needed for a security auditor to do this is much greater.

TAKEAWAYS AND RECOMMENDATIONS

Section takeaways

- Be aware of common misconceptions in wireless security
- Wireless can be used securely, but there is always increased risk
- At least with wireless we think of security more than with wired links

Recommendations to owner/operators

- Find and document where you are using wireless
- Enable security features in wireless technology
- Encrypt the application layer inside the wireless link

Recommendations to vendors

- Only provide solutions with cryptographic protections enabled

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. Enforcement Zone Devices
 - Firewalls and NextGen Firewalls
 - Data Diodes and Unidirectional Gateways
2. Understanding Basic Cryptography
 - Crypto Keys
 - Encryption, Hashing and Signatures
3. Wireless Technologies
 - Satellite and Cellular
 - Mesh Networks and Microwave
 - Bluetooth and Wi-Fi
4. Wireless Attacks and Defenses
 - 3 Eternal Risks of Wireless
 - Sniffing, DoS, Masquerading, Rogue AP
5. **Exercise 3.1: Network Forensics of an Attack**
6. Purdue Level 2 and 3 Attacks
 - Historians and Databases
 - **Exercise 3.2: Bypassing Auth with SQL Injection**
 - HMI and UI Attacks
 - Web-based Attacks
 - Password Defenses
 - **Exercise 3.3: Password Fuzzing**

This page intentionally left blank.

EXERCISE 3.1: NETWORK FORENSICS OF AN ATTACK

DURATION TIME: 20 MINUTES

We are going to validate the network forensics report for the SANS Holiday Hack 2013 Challenge.

In reality, this is the winning solution to the Holiday Hack Challenge. We will validate the first five pieces of forensics evidence. Test your skills to see if you can find the packet or packets reference in each of the findings in this exercise.

OBJECTIVES

Enhance our ability to investigate forensics evidence of ICS attacks

Enjoy the exploration of **It's a Hackerful Life**, Ed Skoudis's creative cybersecurity retelling of **It's a Wonderful Life**

PREPARATION

Start your Control Things Platform VM

Open the network capture

Protocols/Combined/sansholidayhack2013.pcap in Wireshark

For the full background on the challenge, you can visit the original challenge page here:

<https://pen-testing.sans.org/holiday-challenge/2013>

EXERCISE 3.1: NETWORK FORENSICS OF AN ATTACK

EXECUTIVE SUMMARY

The following text is the executive summary of a network forensics report regarding a successful attack of a fictional power grid.

We will use Wireshark to verify the first five findings in the report, which start on the next slide, but first here is an overview of the attack.

A spear phishing email was sent to Don Sawyer and received on 11 December 2013. The email contained a malicious link but the victim did not click the link, hopefully a user security awareness training success.

An attempt to modify the additives database table in the water treatment control system failed due to insufficient privileges for the account used by the malicious actor on 11 December 2013. The attack on the HMI was partially successful in that the HMI web application is vulnerable to SQL injection and the attackers were able to gain filesystem access and therefore would be expected to overcome the limited rights that prevented the modification of the additives table in the database.

...continues below...

Attempts to access the administrator console of the HMI failed after repeated attempts to guess passwords for 'administrator' and 'admin', eventually prompting a lockout of the user. Hopefully this is a lockout based on the malicious user's session and not a potential Denial-of-Service situation for authorized users.

The traffic system network was port scanned on 11 December 2013 from 10.21.22.253 and following the scan revealing ICS devices responding from port 502 (tcp/Modbus), the attackers downloaded the tool modscan from Google Code and attempted to use this tool against 10.21.22.23, the traffic lights controller for Main & Potter. The commands sent to 10.21.22.23 appear to have been rejected by the device.

On 12 December the attacker's attempts to use default credentials for a PLC controller at 10.25.22.22 appear to have failed as the available data indicates they searched for the default credentials but were unsuccessful despite multiple attempts to authenticate to the controller.

While the PCAP leaves some questions as to the network architecture, segmentation, and internet connectivity, in one interpretation there could have been some segmentation and an effort to air-gap the critical infrastructure network components. Further on this later, but a well segmented network would have been a defensive measure.

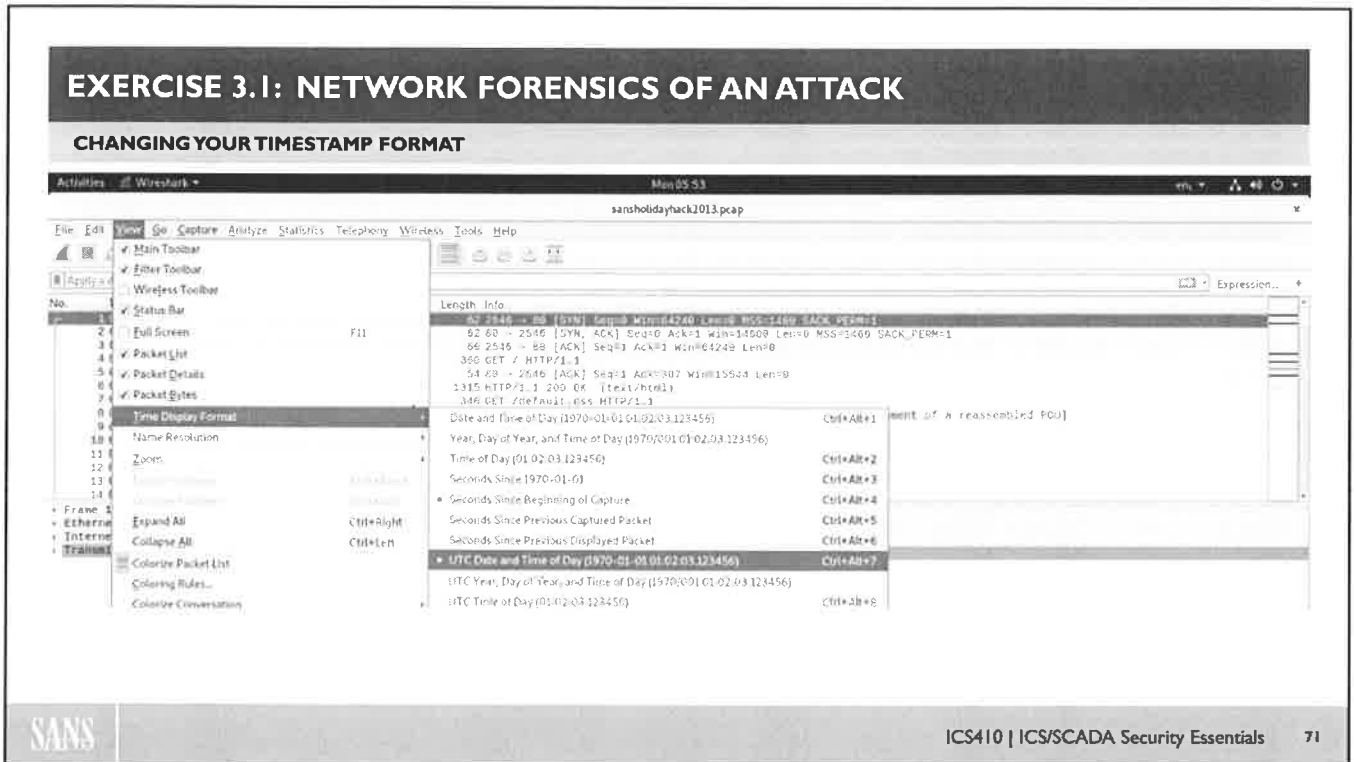
George changed the default credentials on 10.25.22.22 for the MicroLogix 1100, preventing an attempt by the attackers to use factory default credentials to access an HMI control page.

George also separated authorities on the water processing system such that the database account for viewing and logging state data was not authorized to make changes in the additives table.

While controls for other critical infrastructure were compromised through other means, access to the HMI for the power control was made possible by the execution of malware sent to Don Sawyer on which 10.25.22.253 was opened to the attacker, followed by successful password guessing against the account for 'ernie' over SMB to gain access to 10.25.22.58 (SCADA2). The attackers then loaded Meterpreter and are believed to have used Metasploit's VNC module to remotely control the system. Spawning of a command shell, manipulating power controls, and the creation of a message to George Bailey in a text editor can be observed on the streaming video of 10.25.22.58 and a view of the city.

EXERCISE 3.1: NETWORK FORENSICS OF AN ATTACK

CHANGING YOUR TIMESTAMP FORMAT



Before we start verifying the forensics report, we first need to change the default timestamp format in Wireshark. By default, Wireshark shows how many microseconds have passed since the first packet captured. For our forensics report (and most forensics reports for that matter) we need to change Wireshark's timestamp view to UTC mode. You can do this under the View menu as shown in the screenshot above.

EXERCISE 3.1: NETWORK FORENSICS OF AN ATTACK

DOWNLOADING THE NETWORK MAP

12/09/2013 20:53:01

10.25.22.253 (Train Management Workstation) browsed to web server hosted at 10.25.22.250 (documents.valleyelectric.co.nw), first downloading the document **/files/TrafficSystemNetworkMap.pdf** containing network details on Bedford Falls Traffic System

If you are having a hard time finding this packet, try the following Wireshark filter:

```
http && ip.addr == 10.25.22.253
```

This will help you narrow into the attacker's HTTP traffic. Now look at the Info column for the GET request for the network map PDF. You should see the request in packet 16 and the response in packet 42.

While beyond the intended scope of this lab, you can export that PDF selecting packet 42, going to the dissector sections in the middle, and finding the **Media Type: application/PDF** line. Right-click on that line and select **Export Packet Bytes** and save it as **TrafficSystemNetworkMap.pdf**.

EXERCISE 3.1: NETWORK FORENSICS OF AN ATTACK

DOWNLOADING THE TRAIN SWITCHING DIAGRAM

12/09/2013 20:53:15

10.25.22.253 then downloaded the file **/files/BedfordFallsTrainSystem.pdf** containing details on the Bedford Falls Train Switching System Network Components

This one should be easy for you now. You use the same techniques and filters as the last one.

The request is in packet 54 and the response in 112.

EXERCISE 3.1: NETWORK FORENSICS OF AN ATTACK

ATTEMPTED CONNECTION

12/11/2013 14:30:02

A user at 10.25.22.253 (Train Management Workstation) attempts to connect to port 80 of 75.99.175.194, an IP address registered to Counter Hack in New Jersey. The TCP connection requests received RSTs from 75.99.175.194.

If you try using the `http` filter on this one, you won't find the RST packets. Because the the web server rejected the traffic, there is no established HTTP, thus they are not visible with the `http` filter.

Try this filter:

```
ip.addr == 10.25.22.253 && tcp.flags.reset == 1
```

You could add `&& tcp.port == 80` to that, but then you'd miss the fact that port 80 wasn't the only port the attacker was trying to reach and getting RST responses from...

EXERCISE 3.1: NETWORK FORENSICS OF AN ATTACK

PHISHING EMAIL

12/11/2013 14:32:20 -> 10.25.22.253 (Train Management Workstation) retrieves email via POP3 using valid credentials for dsawyer@valleyelectric.co.nw (Don Sawyer) containing one spear phish email. The phish claimed to be from George Bailey but used a domain name one character off from the actual domain name for Valley Electric, george.bailey@valleyelectric.co.nw. The SMTP timestamps and header data indicate the email was sent on Friday 6 December 2013 at 10:53:05 (GMT-0500) from an Apple Mac OS X.

The phish email content sought to have the victim click on a link in order to monitor the Simatic S7-1200 PLC. Clicking on the link was designed to load a Java payload (hook.js) hosted at 10.2.2.2 port 3000. Substituting the 'i' in valleyelectric.com with a 'l' does a fairly decent job at disguising the bogus sender but also note that George is misspelled at the end of the email.

While the source IP address of this phish email is not available and appears to have been sent prior to the start of this packet capture, the SMTP Message-Id indicates it originated from the domain 'hasborg.com' which is registered to Joshua Wright through GoDaddy at IP address 66.135.33.108.

From the available data there is no indication that the user fell victim to this spear phish embedded link attack. Twenty-nine seconds after downloading the email the user's very next session was a direct access to 10.25.22.23 (Train PLC computer).

I think this one is fun. Use the Wireshark filter **pop** for the POP3 protocol and you can see Don Sawyer logging in with his username **dsawyer** in packet 3372 and password **Fashionista** in packet 3375. Packet 3380 says he has 1 email, and packet 3381 requests that email. The email is downloaded in packet 3382, and once it is selected, you can see it in the dissector breakdown in the middle.

Look for the elements of the phishing email contents mentioned above.

You can see the additional traffic mentioned in the last paragraph with this Wireshark filter:

```
ip.addr == 10.25.22.253 && ip.addr == 10.25.22.23
```

EXERCISE 3.1: NETWORK FORENSICS OF AN ATTACK

PORT SCANNING

12/11/2013 15:36:29

Network scanning via SYN packets from 10.21.22.253 (unidentified workstation in the traffic control network subnet) aimed specifically at a limited number of ports: 102, 502, 1089, 1090, 1091, 4000, 4848, 20000, 34963, 44818 of the following traffic control systems:

- 10.21.22.1
- 10.21.22.10 Traffic Grid Controller PLC
- 10.21.22.22 Corner of Vine & Elm
- 10.21.22.23 Corner of Main & Potter
- 10.21.22.24 Corner of Main & Elm

The last three devices highlighted above responded to 10.21.22.253 with SYN/ACK for port tcp/502.

Identifying port scans with Wireshark is a bit tedious, especially if don't have an idea of which IP is doing the port scanning. Luckily, we do have a suspect IP. Let's first narrow in on all the TCP conversations that 10.21.22.253 tries to initiate by identifying any packet coming from 10.21.22.253 whose TCP Flags only have the SYN bit set.

```
ip.src == 10.21.22.253 && tcp.flags == 0x002
```

Notice we use a `tcp.flags == 0x002` instead of `tcp.flags.syn` because only the first is making sure SYN is set to 1 and all other flags are set to 0. This will show you the attacker's valid traffic and the attacker's port scan traffic, you just have to look for a pattern where you see a variance of IPs and ports over a short period of time. Now, if you want to see the results of his portscan, try this filter:

```
(ip.src == 10.21.22.253 && tcp.flags == 0x002) || (ip.dst == 10.21.22.253 && tcp.flags == 0x012)
```

Those two symbols in the middle of that filter are the pipe symbol, which stands for a logical OR operation. So we basically are telling Wireshark to show us 10.21.22.253's outbound SYN requests or any SYN ACK response to 10.21.22.253. Remember, if a port is open, it responds with a SYN ACK, but if it is closed, it responds with a SYN RST or doesn't respond at all. For our case, we are mostly interested in which ports attacker's detected as open, so we only used the SYN ACK filter. If you scroll through the responses, you will see three SYN ACK responses in the middle on packets 4434, 4437, and 4439 which represent the three highlighted bullets above.

EXERCISE 3.1: NETWORK FORENSICS OF AN ATTACK

TAKEAWAYS AND RECOMMENDATIONS

Section takeaways

- Wireshark is an amazing tool for troubleshooting and analyzing network communications
- Packet captures can provide a wealth of information during incident analysis

Recommendations to all attendees

- We examined one small piece of the large simulated attack that is available in the packet capture; if you have time tonight, examine further and tell us what you find
- The full network forensics report (actually the winning answer to the Holiday Hack Challenge) can be found on your course USB in the "Lab & Exercise Supplements" folder.

The submissions for this particular challenge were amazing, detailed, and very creative. Be sure to participate next year.

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. Enforcement Zone Devices
 - Firewalls and NextGen Firewalls
 - Data Diodes and Unidirectional Gateways
2. Understanding Basic Cryptography
 - Crypto Keys
 - Encryption, Hashing and Signatures
3. Wireless Technologies
 - Satellite and Cellular
 - Mesh Networks and Microwave
 - Bluetooth and Wi-Fi
4. Wireless Attacks and Defenses
 - 3 Eternal Risks of Wireless
 - Sniffing, DoS, Masquerading, Rogue AP
5. **Exercise 3.1: Network Forensics of an Attack**
6. Purdue Level 2 and 3 Attacks
 - Historians and Databases
 - **Exercise 3.2: Bypassing Auth with SQL Injection**
 - HMI and UI Attacks
 - Web-based Attacks
 - Password Defenses
 - **Exercise 3.3: Password Fuzzing**

This page intentionally left blank.

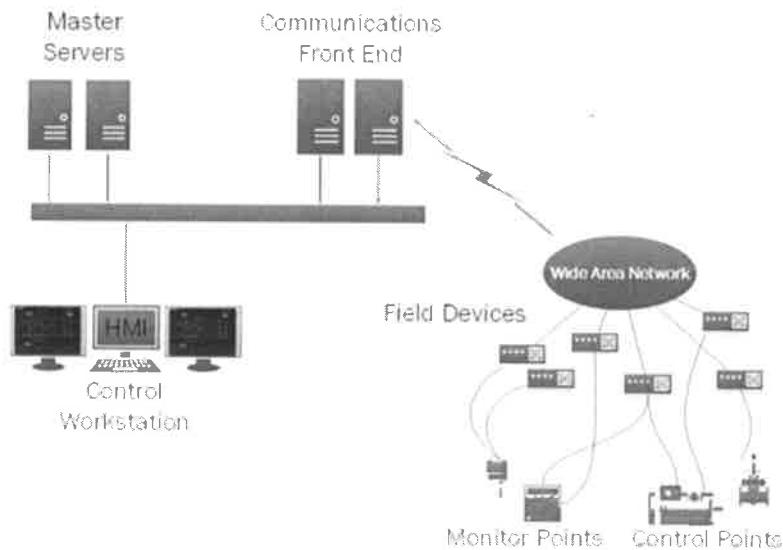
Purdue Level 2 and 3 Attacks

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

This page intentionally left blank.

MOST OF OUR MOST CRITICAL ASSETS ARE SERVERS



When it comes to ICS servers and workstations, you can find them just about anywhere in the control network. Although the majority of master servers will be located in a handful of centralized locations, you often find intermediary servers deployed at remote stations in the field network. All these systems may be attacked and must be considered when we look at our security defenses. The centralized master servers usually present the largest risk to the organization because they usually control the largest number of devices and processes. However, the intermediate servers also present a risk to the organization, which is usually larger than any single field device.

As for workstations, these can often be found evenly distributed across the control network, including in the control center, in distributed offices, with field technicians, and even in the remote stations themselves. Although workstation is a generic term, in the context of this class, this includes any computer running a commodity operating system like Windows or Linux, whose primary purpose is to facilitate the use of software applications to be used by one person at a time. This could include desktops, laptops, thin clients, and in some cases, even mobile devices. This includes many modern-day HMIs as well.

WHY AND HOW ATTACKERS TARGET LEVEL 2 AND 3 NETWORKS

Why target ICS servers and workstations?

- TCP/IP networks are more familiar to attackers
- Directly monitor and control ICS processes (**HMIs**)
- Greater context about the ICS processes (**project files**)
- High-level controls to those processes (**master servers**)

How do they reach ICS servers and workstations?

- Most often through the company's internet perimeter
 - Propagate through control network perimeters
 - Often leverage trust relationships between servers
- Remote access connections for engineers or vendors
- Attacks can come from field networks, but less common

Remote attacks offer little personal risk to attacker

Attackers will often prefer to target servers and workstations for a number of different reasons. First, because they directly monitor and control ICS processes, they provide a greater value to the attacker than a remote station or a single field device. Workstations and servers, especially HMIs, usually provide the attacker with greater context about the ICS processes and high-level control to those processes. Servers and workstations are also usually connected to traditional TCP/IP networks and run commodity operating systems. This makes them more familiar targets for most attackers, allowing them to use most of the security tools in their arsenal.

Attackers can often reach ICS servers and workstations through the company's internet perimeter. This allows an attacker to perform remote attacks without the need for physical access, which, when successful, can enable them to manipulate the control processes without exposing themselves to physical risk. Remember, this is true even if the attackers must traverse multiple internal networks and eventually even the control network's perimeter.

Attackers can also leverage trust relationships that exist from databases utilized throughout an ICS environment.

TARGETS ON WORKSTATIONS AND SERVERS

ICS software is the primary target on the workstations and servers

- This software needs to be configured and hardened
- Each vendor's software has different security capabilities
- Must be secured according to the vendor's guidance
- This guidance is often limited or non-existent

Third-party solutions may need to be used

- Mitigate weaknesses that can't be mitigated in ICS software
- Bump-in-the-wire security solutions like VPNs
- Secure proxy interfaces with centralized authentication and logging
- Application whitelisting and sandboxing solutions

But we must also worry about the underlying OS

In an ICS environment, attackers' primary targets are often the control software running on each of the workstations and servers. This is what gives them access to the control system and control over the processes. Each ICS vendor's software needs to be configured and hardened in a secure manner to defend against attack. The first level of defense for this software should be the security configurations the software provides. Each piece of ICS software has different security capabilities, and each must be secured according to the vendor's guidance.

Often, we will find that ICS software does not have adequate security controls built in. In these cases, third-party solutions may need to be used to mitigate any identified weaknesses in the ICS software that can't be mitigated through the software's own controls. Examples of this are VPN gateways, bump-in-the-wire security solutions, secure proxy interfaces with centralized authentication with logging, application whitelisting software, and application sandbox solutions. These third-party solutions often help us provide security defenses missing in our ICS software, and also provide secondary layers of defense if a lesser yet similar feature is already available in the ICS software.

THE UNDERLYING OPERATING SYSTEM

Most ICS servers and workstations run general-purpose OS

generally windows

Most underlying OSs expose inputs to the network

- TCP/IP stack
- Network services
- Client network applications

Vulnerabilities in general-purpose OSs are the most documented and exploited

- OS compromised == hosted ICS applications compromised
- Must also harden the operating system from attack

Most ICS servers and workstations, regardless of their location or primary use, run commodity operating systems (OS) like Microsoft Windows and Linux/UNIX. This becomes a common attack surface for attackers across all your ICS systems. A vulnerability in one of your ICS system's OS may likely be a vulnerability in another of your ICS system's OS, even if it's running different ICS software, serving different purposes, and located in a completely different part of the control network.

To provide network benefits to the business, most underlying OSs expose their TCP/IP stack, network services, and client network applications to the network. This makes up the initial attack surface for an operating system. If vulnerabilities are found in these OS components, they are often some of the most documented and exploited of all vulnerabilities. This makes the OS become an initial attack point for the ICS system and becomes an enabler to jump from one process or application to another process or application, including the ICS software installed on the system. Control of the operating system means at least partial, if not full, control of the ICS software installed and running on that system, including any ICS communications passing through that system.

ICS SERVER AND WORKSTATION OPERATING SYSTEMS

ICS workstations mostly use Microsoft Windows

ICS servers can be running on

- Workstation OSs like Windows XP and 2000 Workstation?!?
 - Common on black box system (provided and maintained by the vendor)
 - Might be lucky enough to have VAX VMS and DOS, too?!?
- Microsoft Windows Server versions
- Linux from Red Hat, Novell SUSE, Oracle, Ubuntu, or Debian
- Traditional UNIX systems like HP-UX and AIX
- Mainframe technologies like IBM System Z

We'll be focusing on the most common of these

- Microsoft Windows
- Linux (which is mostly applicable to UNIX)

This page intentionally left blank.

SOCIAL ENGINEERING (SE)

The art of manipulating humans to do what you want with a combination of logic and emotion

- Convincing the help desk you need your password reset (at least the person you are pretending to be)
- Getting a field technician or engineer to insert a USB drive into his or her field laptop (to infect the laptop with a trojan)
- Gaining physical access to guard-protected locations
- Convincing a group of managers or engineers to open an email attachment or click on a web link

SE is often used to enhance other technical attacks

One method attackers often leverage is social engineering attacks. It is often said that if you can't find a vulnerability in a system, just have the user do the attack for you. Social engineering uses a combination of logic and emotion to influence a human to perform an action for you. This could be as simple as convincing someone on the help-desk team to reset a user account's password for you or getting a field technician to insert a malicious USB drive into his field laptop. Social engineering can also be used to bypass guard stations to gain access to physically restricted areas or creating a phishing email that convinces a manager or senior engineer to install a virus attached to the email.

Social engineering can be used by itself to perform some attack or it can be used in combination with other digital attacks to enhance the overall effect of the attack. Implementing email filtering, website content filtering, and a security awareness program that teaches users to identify social engineering activity and URL redirection efforts are keys in defending social engineering attacks.

PHISHING AND SPEAR PHISHING

Phishing is the process of sending an attack to a large number of individuals

- Attacks are more general and sent to a larger variety of targets
- The more people it is sent to, the more will likely fall victim

Spear phishing is similar, but it targets a small number of people

- Attacks are very specific and convincing, often modified for each target
- More effort is spent to increase the odds that the target will fall victim

Phishing is an attack that is sent to a large number of individuals in hopes that a percentage of them will fall for the attack. This could be convincing them to open a virus attached to an email or getting them to click on some link to go to a malicious webpage that uses their browser in some sort of attack. Phishing attacks are a general type of attack and sent to a larger variety of targets. The more individuals the phishing attack is sent to, the more will likely fall victim to the attack.

Spear phishing is similar to normal phishing attacks, but instead, it targets a small number of people. These types of attacks are very specific to the target audience and much more convincing. Often, the attacker will modify the spear phishing attempt for each person he is targeting or for each department he is targeting. At a minimum, a spear phishing attack will target a single company or single related group of people such as a group working on common standards. In spear phishing attempts, the attacker spends more effort in creating a convincing attack with hopes that the extra time spent will increase the odds that the targets will fall victim.

MALWARE

Virus: Parasitic malware that relies on executable code insertion and user interaction to spread

Often targets client systems



Types of malware

- **Trojans:** Malware that pretends to be good software
- **Backdoors:** Malware that provides remote access for attackers
- **Bots:** Malware that checks into a command-and-control server
- **Worms:** Malware that can self-replicate

A *virus* is a malware specimen that has the capability to replicate and possesses parasitic properties. A virus is a parasite because it cannot exist by itself; instead, it must attach itself to another executable program, data file, or area of coding (like a boot sector).

The *payload* of the virus executes when the user or computer launches the code to which the virus is attached. Instructions in the payload allow the virus to replicate and give it the opportunity to do its author's bidding. Some viruses do little more than spread and serve as a nuisance. Others can do serious damage, such as destroy data or degrade system performance. Even if a virus isn't intentionally designed to cause damage, merely infecting other program code is damaging by itself. Infections can cause errors, lockups, and operational problems, and at the very least, the virus is taking up CPU cycles.

Another way to have malicious code execute when the machine starts up is to place it in the boot sector of the computer's disk. Every disk has a boot sector, regardless of whether or not it is actually bootable. When a PC is powered up, it looks for boot information in the order dictated by the machine's BIOS. If any of the media in the drives specified by the BIOS has an infected boot sector, the infection will get transferred to the boot drive. Once the infection is complete, malicious code will get loaded into memory at startup. A malware specimen that places malicious code into the boot sector is called a *boot record infector*.

Historically, the vast majority of boot record infectors have been viruses. That is why you are much more likely to hear about some "boot record virus," and will rarely (if ever) hear the term "boot record worm."

MALWARE CAPABILITIES

Destruction of data
Leaking confidential information
Providing backdoor access
Countless other opportunities

Basically, anything any application on your computer can do

- Limited only to attacker's effort and imagination

Anything that can impact the confidentiality, integrity, and availability of critical information represents a risk to security. Worms and viruses represent a threat because they can access confidential records, and, possibly, retain your records for future use. From a business perspective, there is usually monetary loss associated with losing critical files, revealing sensitive information, or providing unauthorized access to internal systems.

Destroying Data

Destroying data is one of the most insidious actions a malware specimen can take after infecting a system. For example, the CIH virus had a particularly destructive payload. CIH was programmed to activate every year on April 26, at which point it overwrote data on the computer's hard drive. Additionally, the virus attempted to overwrite the flash-BIOS of the infected system, often rendering the computer unusable. CIH is also known as the Chernobyl virus, because April 26 marks the anniversary of the nuclear plant disaster that occurred in Chernobyl, Ukraine, in 1986.

If you lose data as a result of a malware infection, your most practical means of recovery are to retrieve files from backup. If backups are not available and lost data was very valuable, you may be able to restore it via low-level forensic recovery techniques. Such methods, however, tend to be time-consuming and expensive. Unfortunately, destruction of data is only one danger associated with a malware infection.

Leaking Information

The possibility that a malware incident led to information leaking to unauthorized parties can be as devastating as the destruction of data. You may recall that the Melissa virus often resulted in sensitive Word documents being sent to recipients listed in the victim's email address book. Of course, a document is only one type of information whose confidentiality can be compromised by malware. The Caligula virus was programmed to locate the victim's Pretty Good Privacy (PGP) private key file and transmit it to the creator of the virus via FTP. The Marker virus, discovered about half a year later, used a similar technique to obtain information about the infected user from the system's registry and transferred the data to the author's FTP site. This capability allowed Marker to maintain a trail of infected users, empowering its creator to study relationships between members of the targeted organization.

IT MALWARE AFFECTS ICS

Conficker worm was found on multiple energy ICSs

- Internet-spreading worm found its way into generation plants
- 2008: Many North American ICS organizations affected
- Was very difficult to remove in ICS



Source: Electricity sector incident reporting



Conficker Infections

There were a number of Conficker infections that found their way onto ICS, according to the electricity sector cyber incident and reliability event reporting. This worm was designed to exploit the MS-086 DCOM/RPC vulnerability. The worm found its way onto plant DCS systems and was difficult to remove. Infections indicate weaker architectures and infections between hosts. Malware in ICS environments remains the leading cause for ICS cyber incidents. In most cases, the infections are not targeted or customized and simply find their way onto hosts that are vulnerable through USB media or mobile laptops. Infections are often detected as the laptop or USB leaves the ICS and is plugged into an enterprise network with up-to-date virus signatures.

STUXNET

Discovered in June 2010

Targeted Iran's nuclear facilities

Attacked Siemens Step 7 software to reprogram PLCs

Excellent write-up on Stuxnet:

- http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/stuxnet_0_5_the_missing_link.pdf

Stuxnet was a computer virus discovered in June 2010. It infected a number of computers worldwide. Further research showed that Stuxnet was designed to target specific Siemens systems running nuclear facilities in Iran. It did this by infecting the target system and searching it for the existence of Siemens' Step 7 software, which is used to configure Siemens control equipment. If it found such software, it then analyzed the Step 7 project files to determine if the infected system was the targeted Iranian nuclear facilities and, if so, reconfigured the control systems to perform an action to disable the facilities. If the infected system didn't contain the Siemens Step 7 software, or if the Step 7 software wasn't the targeted Iranian system, then it would try to spread itself to other systems to continue its search for its Iranian targets.

Reference:

There is an excellent write-up on the Stuxnet worm by Symantec on their website:

http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/stuxnet_0_5_the_missing_link.pdf

STUXNET VERSIONS

Table 1

Evolution of Stuxnet versions

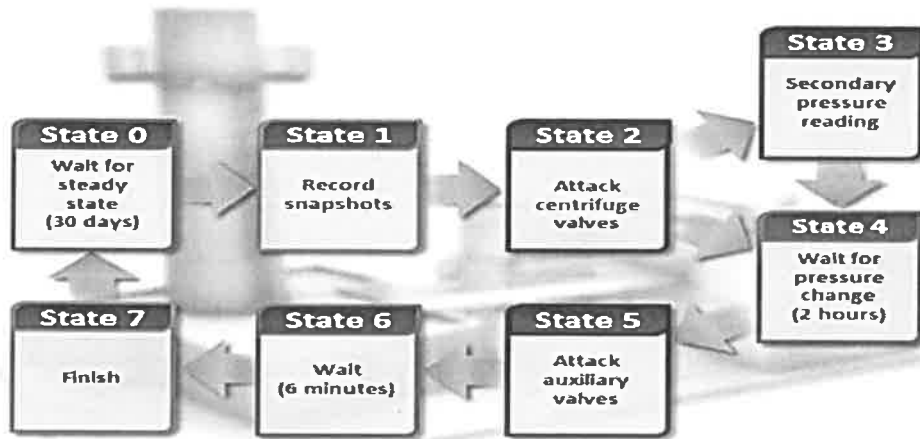
Version	Date	Description
0.500	November 3, 2005	C&C server registration
0.500	November 15, 2007	Submit date to a public scanning service
0.500	July 4, 2009	Infection stop date
1.001	June 22, 2009	Main binary compile timestamp
1.100	March 1, 2010	Main binary compile timestamp
1.101	April 14, 2010	Main binary compile timestamp
1.x	June 24, 2012	Infection stop date

Security researchers were able to obtain several different versions of the Stuxnet virus. Using a combination of those versions, details from the command-and-control (C&C) servers, and reverse engineered code in those versions, they have been able to create a tentative timeline for the multiple Stuxnet versions released. Planning for this virus appears to have occurred long before the virus was released because the command-and-control servers were registered clear back in 2005. Through reverse engineering, researchers were able to find compile timestamps for version 1.0 dating back to 2009, a full year before the virus was initially found. Amazingly, the 0.5 version was initially discovered in 2007.

However, security researchers didn't understand the true nature of this virus at the time, and it wasn't fully understood until it was rediscovered in 2010 when version 1.1 was released.

All versions of Stuxnet had timeout features on its spreading capability, and the latest version stopped spreading itself as of June 24, 2012.

STUXNET EXPLOIT PAYLOAD PROCESS



http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/stuxnet_0_5_the_missing_link.pdf

Here is a state machine of the exploit process Stuxnet performed when it found the targeted Iranian nuclear facilities. The diagram above and the state descriptions below are from the Symantec report referenced above.

State 0 (Wait): Perform system identification and wait for the enrichment process to reach steady state before attack. This can take approximately 30 days.

State 1 (Record): Take peripheral snapshots and build fake input blocks for replaying later.

State 2 (Attack centrifuge valves): Begin replaying fake input signals. Close valves on most centrifuges.

State 3 (Secondary pressure reading): Open both centrifuge and feed stage valves in the final stage of a single cascade to obtain a low-pressure reading.

State 4 (Wait for pressure change): Wait for desired pressure change or time limit. This can take up to approximately two hours.

State 5 (Attack auxiliary valves): Open all auxiliary valves except valves believed to be near the first feed stage (stage 10). Wait for three minutes in this state.

State 6 (Wait for attack completion): Wait for six minutes while preventing any state changes.

State 7 (Finish): Reset and return to State 0.

STUXNET EXPLOITS

Table 2

Evolution of Stuxnet exploits

Vulnerability	0.500	1.001	1.100	1.101	Description
CVE-2010-3888			X	X	Task scheduler EOP
CVE-2010-2743			X	X	LoadKeyboardLayout EOP
CVE-2010-2729		X	X	X	Print spooler RCE
CVE-2008-4250		X	X	X	Windows Server Service RPC RCE
CVE-2012-3015	X	X	X	X	Step 7 Insecure Library Loading
CVE-2010-2772		X	X	X	WinCC default password
CVE-2010-2568			X	X	Shortcut .lnk RCE
MS09-025		X			NtUserRegisterClassExWow/NtUserMessageCall EOP

In version 0.5, Stuxnet was capable of only replication through the infection of Siemens Step 7 project files through the Insecure Library Loading exploit. As shown in the table, later versions increased in the ability to spread through Microsoft vulnerabilities.

STUXNET REPLICATION

Table 3

Evolution of Stuxnet replication

Replication Technique	0.500	1.001	1.100	1.101
Step 7 project files	X	X	X	X
USB through Step 7 project files	X			
USB through Autorun		X		
USB through CVE-2010-2568			X	X
Network shares		X	X	X
Windows Server RPC		X	X	X
Printer spooler		X	X	X
WinCC servers		X	X	X
Peer-to-peer updating through mailslots	X			
Peer-to-peer updating through RPC		X	X	X

Stuxnet leveraged a number of different methods to replicate itself from one system to another system. Initial infection is believed to have occurred by USB drive. After initial infection, Stuxnet tried to spread via other USB drives that were inserted into the system and through a number of different network protocols including file shares, Windows RPC services, print spoolers, Symantec WinCC web servers, and through peer-to-peer services. Once again, a replication deadline was hardcoded into Stuxnet so it would stop attempting to spread itself after the specified date in 2009 or 2012, depending on the version.

ICS MALWARE 2011-2012

Duqu (2011)

- Nearly identical to Stuxnet, but different payload
- Used for cyber espionage, not attack
- A second version (Duqu 2.0) was used in 2015 to attack several targets, including Kaspersky Labs

Flame (2012)

- Contained nearly identical modules, but more complex
- Used for cyber espionage, not attack

Shamoon (2012)

- Also used for cyber espionage
- Deleted files after sending to C&C, then overwrote boot sector
- Used to compromise approximately 30,000 systems at Saudi Aramco in 2012
- Shamoon2 was found in 2016, also targeting Saudi Arabian organizations

Since Stuxnet, there have been several viruses targeting ICS systems, some of which have been proven to share some heritage and code with Stuxnet. Duqu and Flame were viruses released in 2011 and 2012, respectively, but neither was used for attack. Instead, both viruses seem to have been used for cyber espionage, collecting information from various systems and sending that information back to its command-and-control (C&C) server. Also in 2012, another virus named Shamoon was found targeting oil and gas sectors. A group named the "Cutting Sword of Justice" claimed responsibility for this virus, which infected approximately 30,000 systems at Saudi Aramco, the largest oil producer in the world. Shamoon differed from Duqu and Flame in that it deleted any files it stole from the system and then proceeded to overwrite the system's boot sector, requiring the system to be completely reinstalled or restored from backup.

ICS MALWARE 2013–2014

Havex/Dragonfly (2014)

- Had been used in the wild since early 2013
- Enumerates OPC endpoints and tags, causing lockups and DoS effects
- Three ICS vendor websites were compromised and made to offer up Havex as trojan
- A white paper about this is included on your course USB

BlackEnergy (2014)

- Exploits CVE-2014-4114 vulnerability in Windows OLE
- Windows Vista, Server 2008, Windows 7, Server 2008 R2, Windows 8 and 8.1, Windows Server 2012 and R2, Windows RT and RT 8.1
- Delivery via PowerPoint (observed)
- Industrial control systems targeted
- Two additional versions (2 and 3) exist; version 3 was used in the Ukraine Power Grid attacks

Havex (also known as Dragonfly) differs from the others in that it specifically targets the OPC protocol, attempting to find and enumerate OPC endpoints. The attackers in charge of Havex were so aggressive that they compromised at least three ICS vendor websites and replaced official ICS software downloads with a trojan version of Havex pretending to be the vendor's software.

BlackEnergy Timeline

GlobSec attendees targeted May 2014.

Targeting of European entities using CVE-2013-3906 June 2014.

Darkmoon variant appears to have been ready since September 10, 2014.

PowerPoint since September 12, 2014.

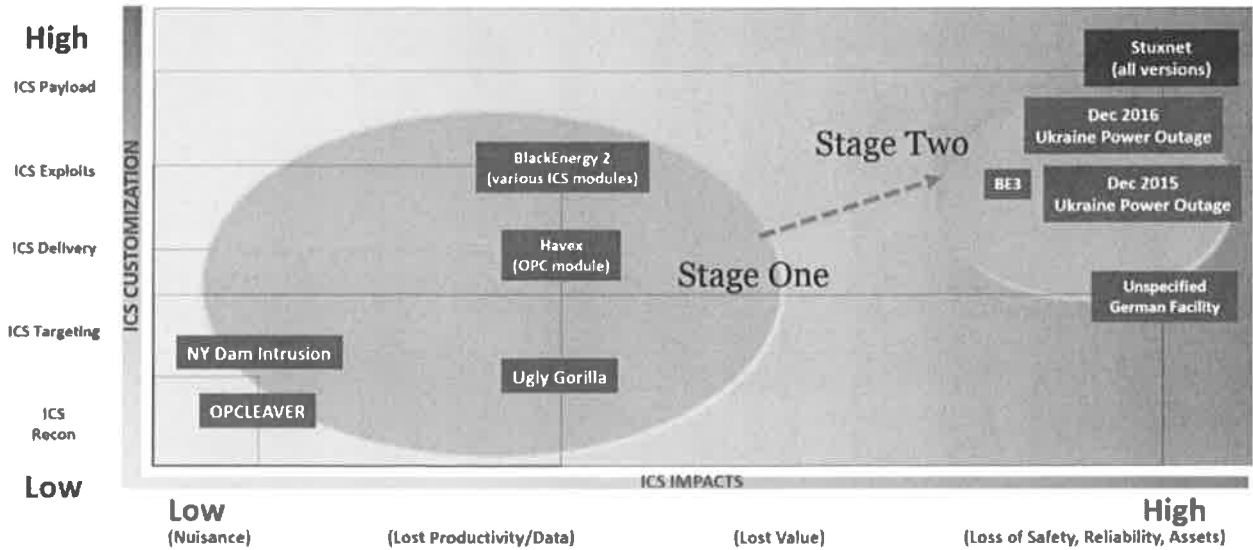
Payload C2 activity detected September 24, 2014.

MS14-060 patch available October 14, 2014.

Circumventing patch in October after release.

New CVE-2014-6352 and temporary fix released October 21, 2014.

RECENT ICS THREAT AND ACTOR LANDSCAPE



SANS

ICS410 | ICS/SCADA Security Essentials 97

In 2014, the world became aware of several campaigns to target ICS. The most notable have been Havex and BlackEnergy2 (BE2). The use of an OPC-related module associated with the Havex trojan brought the ICS connection to light. Both Havex-associated incidents and BE2-related campaigns have demonstrated success by taking advantage of ICS supply chain trust and work habits. BE2 also directly exploited internet-facing ICS components.

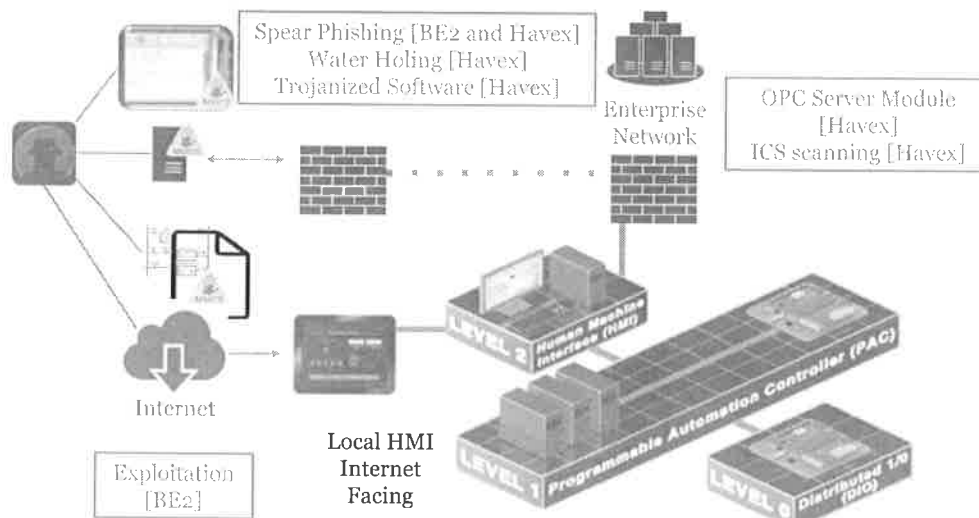
With the proper understanding of which attack paths expose an ICS to attacks, an organized defender can prioritize his efforts. Reducing attack paths by managing exploitable vulnerabilities in your organization will help reduce your overall exposure to the threat landscape. As attackers lose lower-cost options, the defender begins to win.

It is important to note the following:

- Greatest number of ICS cyber incidents continue to be non-targeted malware, but...
- Emergence of targeted ICS attacks
- Custom ICS exploits and features
- Gaps and segmentation are expected
- Targeting trusted ICS relationships

Some experts believe that Stuxnet, Havex, and BE2 have moved us from the era of accidental infections and insiders to targeted and ICS-customized attacks.

BE2 AND HAVEX ICS DELIVERY TECHNIQUES



Two major techniques are employed: Exploiting trust relationships and work habits (spear phishing, water holing, trojanized software), and direct exploitation of vulnerable ICS components (across multiple vendors, including Siemens SIMATIC WinCC, Advantech/BroadWin WebAccess, and GE CIMPLICITY) that are internet-facing.

Adversaries targeted the personnel with spear phishing emails. Recent ICS-targeted technical threats have included this style of targeting with phishing emails as was observed in the Havex and BlackEnergy2 campaigns. Targeting and delivery techniques have focused on specific individuals, trusted relationships with ICS and industrial suppliers, and the need to download files.

The phishing emails would have contained a document that hosted malicious code. Once opened, the malicious code would have targeted a vulnerability in an application on the target's system. Once the application was exploited, the target system would have opened a remote connection point allowing adversaries access to the network. The second stage of the attack would have involved the adversary accessing the network and, as observed in previous cases, establishing a foothold on the network through the compromise of small sets of workstations. Internal reconnaissance by the adversary from these systems would have provided access of credentials or unsecured systems and connections. This type of reconnaissance is typically performed through the use of keyloggers, network scanning, and the compromising of systems such as Active Directory.

You must have safe procedures for moving files from vendors (especially available on a webpage), such as the following:

- Strong points to check all files prior to moving into production
- Request out-of-band file
- Request hash of file and compare before moving into production
- Water-holing defenses and conducting ICS-related web sessions in an isolated non-trusted host

There are not many public examples of ICS-specific attacks. Two of the best examples are Havex and Stuxnet. Even related attacks, such as Duqu and Flame, did not actually cause any damage to or infect ICS. However, through threat intelligence research (often takes place with honeypot research) and firsthand knowledge of other non-public incidents in the industry, we all know that there are definite trends and that the threats are real.

Air-gap-jumping malware is a common capability employed now either intentionally or unintentionally through USBs and laptops. Attacks that have manipulated specific protocols have been shown to be effective. Adam Crain and Chris Sistrunk have done great research on the capabilities of crashing master stations from remote sites through DNP3. Exploit tools found specifically targeting ICS facilities have been observed heavily in honeypot research. Process-focused effects have been observed in a few incidents where attackers have wanted specific outcomes. Firmware manipulations were observed in honeypot research (Kaspersky Labs showcased this as well), and the manipulation of firmware updates was observed in Havex. A few non-public incidents have found that adversaries deleted important operations data on the network after gaining access.

Attack trends are useful in understanding the threat landscape. We all know there are threats, but what do we do to actually reduce these attack paths? Some notable trends to consider include:

- ICS-specific targeting (**Stuxnet, Havex, BE2**)
- Focus on "river crossing" gap jumping (**Stuxnet**)
- Protocol custom/capable attacks (**Havex**)
- ICS-specific exploit tool development (**Researchers, BE2**)
- ICS-specific exploit tool(s) used (**Honeypot research, BE2**)
- Process-focused effects (**Stuxnet, other incidents**)
- Firmware (**Honeypot research**)
- Data destruction/resource depletion (**Related incidents, BE2 module**)

TAKEAWAYS AND RECOMMENDATIONS

Section takeaways

- General IT malware can affect ICS if it enters the control networks
- Targeted ICS malware can be much more impactful
- All leverage unpatched systems

Recommendations to owner/operators

- Build strong perimeters
- Monitor all traffic through those perimeters
- Patch your ICS systems where you can, especially in Level 3

Recommendations to vendors

- Enhance your CERT efforts with customers

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. Enforcement Zone Devices
 - Firewalls and NextGen Firewalls
 - Data Diodes and Unidirectional Gateways
2. Understanding Basic Cryptography
 - Crypto Keys
 - Encryption, Hashing and Signatures
3. Wireless Technologies
 - Satellite and Cellular
 - Mesh Networks and Microwave
 - Bluetooth and Wi-Fi
4. Wireless Attacks and Defenses
 - 3 Eternal Risks of Wireless
 - Sniffing, DoS, Masquerading, Rogue AP
5. **Exercise 3.1: Network Forensics of an Attack**
6. Purdue Level 2 and 3 Attacks
 - Historians and Databases
 - **Exercise 3.2: Bypassing Auth with SQL Injection**
 - HMI and UI Attacks
 - Web-based Attacks
 - Password Defenses
 - **Exercise 3.3: Password Fuzzing**

This page intentionally left blank.

Historians and Databases

Applicable Standards:

- **NIST CSF v1.1:** PR.DS-1
- **ISA/IEC 62443-3-3:2013:** SR 3.4, SR 4.1
- **ISO/IEC 27001:2013:** A.8.2.3
- **NIST SP 800-53 Rev. 4:** MP-8, SC-12, SC-28
- **CIS CSC:** 13, 14
- **COBIT 5:** APO01.06, BAI02.01, BAI06.01, DSS04.07, DSS05.03, DSS06.06

This page intentionally left blank.

DATABASES IN ICS ARCHITECTURES

Operational databases are often essential cyber assets

- Most have high availability requirements
- Many have redundant servers for critical control applications

Historians are used to serve information to parts of the ICS

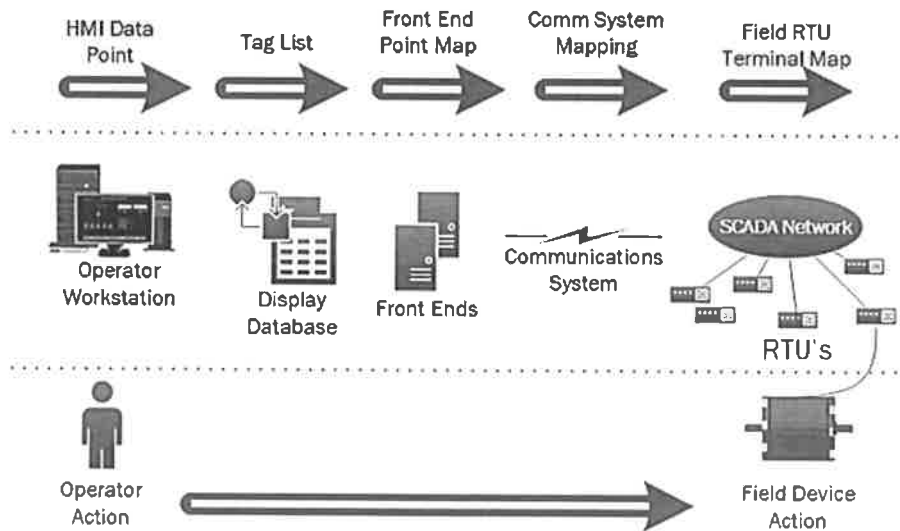
- Some to the support engineer planning in operations network
- Some to the support accounting system in the business network
- Often accomplished through data replication to another historian
- Many can require remote access for vendor support

Many of the databases that are relied upon by the ICS for "real-time" operations are considered to be critical or essential applications/servers. You will typically find high availability and highly redundant deployments with automatic failover. Historians are used to aggregate process data and provide views of that data over time. They often serve data out into Purdue levels 3 and 4 and rely upon third-party data translation, like OPC servers, to receive data from different components in Purdue levels 1 and 2.

It is common to place a historian in the DMZ level and replicate and other databases to a slave historian or database in business and enterprise networks. Vendor supplied, third-party databases and historians typically require remote access for vendor support and troubleshooting. The supporting enforcement zone can use traditional firewalls, restricted ACLS, or data diodes to secure the infrastructure from less-trusted networks.

Note: You must look much further than words on paper when describing a master database to a slave database. One-way data flows often begin with 2-way handshakes/sessions to establish the connection. Good database security is important on both sides of a data replication scheme.

MAPPING THE DATABASES



Starting from the bottom swim lane, you can see the perspective of the operator:

Experienced operators have typically developed through their work experience a necessary line-of-sight conceptual understanding, in which they can translate how a relatively routine task of clicking an icon on a particular computer screen will result in a physical event occurring locally in the plant or potentially many miles away. Conversely, when experienced operators see a system alarm or questionable value in one of their computer systems, they will have a reference point or working context to visualize a condition that may exist in the field.

The middle swim lane provides the perspective of the underlying technology that is required to connect the operator workstation to the remote field device.

The top swim lane provides the perspective of the backend databases and point mappings that are required to deliver the correct message in the correct format to the final destination.

ICS DATABASE GROUPINGS

Feed data into the processes

- Runtime databases
- Real-time configuration databases
- Hierarchical databases for point mapping
- Tag databases
- Formula/recipe databases
- Scientific dataset databases
- State estimating databases

Collect information from the process

- Historians
- Alarm databases
- Analytic databases
- Trending databases

Build projects and configure systems/devices

- Project databases
- Configuration databases
- Scheme databases

Supporting applications and user interfaces

- Facility databases
- Asset databases
- GIS databases
- User interface databases
- Forecast databases
- Scheduling databases
- Assignment databases
- Security databases (SIEM, AV, etc.)

There are four major functional categories for ICS databases as you think about an ICS. The ICS implementation typically relies on ICS supplier/vendor-supplied runtime databases.

The runtime database contains information that must be accessed by the ICS, such as setpoints, tags, and so on. Changes to this database would impact the process being monitored and managed. The security of these databases is most influenced by how the ICS supplier has programmed the database and integrated it into the ICS solution.

The most interconnected type of database provides historian functions and aggregation of things like event logs and alarming. The databases can take data from the ICS and supporting systems and often share that data beyond the ICS. These solutions can be provided by the ICS vendor or by third parties.

Project databases are usually provided by the ICS vendor as a way to build "projects" and configure ICS devices, such as PLCs. They are most commonly found as a part of the Engineering Workstation. They can be used to choose settings and manage the software and firmware on devices as well as map points and more. They are considered offline tools but are used to generate files that have online implications.

The final types of databases are associated with supporting applications. A control room for a power system may have special-purpose databases to display power system assets and facilities on a chart or a map. The GIS-based solutions help present system information in a geographic manner or logical (HMI-like) manner. This application is good for outage management and more.

HISTORIANS

Historical data loggers/databases

- Store data for trending, displayed on the HMI or other analytic machines
- Data from some historians may be used to make automated control decisions
- Provide data/event traceability in manufacturing processes
- Can contain large sets of data, down to torque used on specific screws

Data is often shared past the DCS

- Various ICS vendors have operational historians in Level 2
- These feed into enterprise/aggregate historians like PI

OSIsoft's PI historian

- Largest player by far in enterprise/aggregate historians
- If PI historian only feeds business, put it in Control DMZ
- If used for control, place in Level 3 with read-only replica in Control DMZ

Historians are usually referred to as "real-time databases," which is a little misleading by definition, but it underscores the fact that the data supports operation time requirements. Many implementations use a common SQL server and can have web interfaces to the dataset.

Operational historian refers to a database software application that logs or archives time-based process data. Historian software is used to record trends and historical information about industrial processes for future reference. It captures plant management information about production status, performance monitoring, quality assurance, tracking and genealogy, and product delivery with enhanced data capture, data compression, and data presentation capabilities.

Operational historians are like enterprise historians, but differ in that they are used by engineers on the plant floor rather than by business processes. They are typically cheaper, lighter weight, and easier to use and reconfigure than enterprise historians. Having an operational historian enable "at-the-source" analysis of the historical data is not typically possible with enterprise historians.

Reference:

http://en.wikipedia.org/wiki/Operational_historian

Some historian applications have even been used for control. That architecture is not commonplace, but it is occurring.

OSIsoft's PI (Process Information) historian is prevalent in ICS applications. SCADApedia states, "The PI System is a historian that also has many data analysis and management applications." OSIsoft developed PI in the 1980s, and it has grown to be the most widely deployed historian in the control system space, by far. OSIsoft states that "65% of Global 500 process and manufacturing companies use the PI System." Its deployment in critical infrastructure sectors makes it important to secure. One of the PI System's strengths developed over decades is a large number of interfaces that allow almost any data to be sent to and stored in PI.

There are hundreds of interfaces that cover specific SCADA/DCS/control system vendor applications, control system protocols, IT protocols, and much more. It is literally difficult to find data that cannot be sent and stored in a PI server.

The most widely used PI interface is the OPC interface. OPC is the universal translator protocol in the control systems world and available in most systems for interoperability with other systems. For this reason, and for the robustness of this interface, it is often the default choice for sending data to PI.

OSI PI has even been used for security event correlation and management, as a supplement to its traditional event logging and correlation. A project completed by Digitalbond developed the Portaledge tool for this specific purpose.

Digital Bond's Portaledge project uses the PI System to aggregate and correlate security events and detect cyber attacks. The security event data is sent via PI interfaces. The data is then normalized using the PI Module Database, and then attack correlation calculations are performed using ACE.

This allows PI System users with the appropriate OSIsoft licenses to add a SCADA SIEM capability to their PI System.

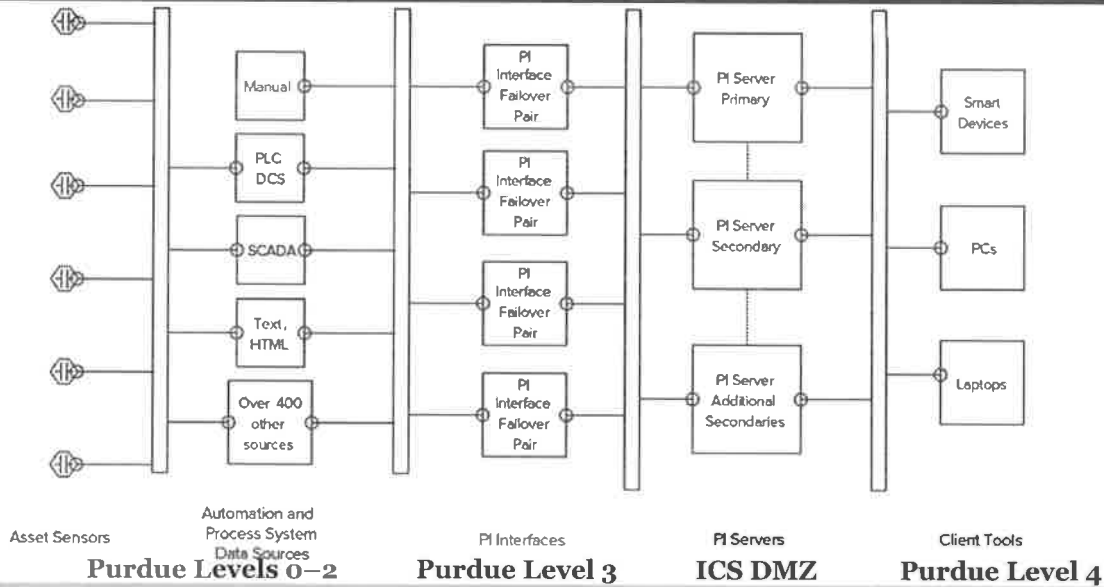
Reference:

SCADApedia: <http://www.digitalbond.com/blog/2007/03/28/scadapedia/>

Many industrial control systems have focused on operational control and were never designed to scale appropriately to accommodate and store large amounts of data collection for management business intelligence systems.

Many ICS environments have used simple worksheet programs to store limited point data and configurations. Larger systems always required available databases to serve the application and needs of the system. Today, especially in manufacturing, the need for traceability for products means all sorts of data specific to production runs is being collected and stored.

ENTERPRISE/AGGREGATE HISTORIAN ARCHITECTURE



SANS

ICS410 | ICS/SCADA Security Essentials

108

This great picture from OSIsoft shows how databases are used within the ICS and beyond. The picture develops information to collect from left (plant floor or field) to right (control room and higher-level control applications) on out to applications and systems in the business network or external organizations.

Data sources include pumps, valves, sensors, and actuators as discussed previously. These are points in the system tied in by inputs and outputs. This data is collected and processed by devices and applications in the control network. There might even be third-party historians running to manage data specific to certain equipment or subprocesses. This data typically flows out to the DMZ level, so it can be archived and shared to support applications in and outside of the ICS network. Data is usually served out to production support networks or even the corporate business network, where uses range from production scheduling, supply chain management, billing, and accounting.

If the data in your aggregate historian is used to feed data back into your control processes, or if the majority of the data it stores is not used by the business, you should consider moving the historian to Purdue Level 3 and replicate the data needed by the business to a replica historian in the ICS DMZ.

Reference:

<https://www.osisoft.com/pi-system/#tab1>

ATTACKING HISTORIANS AND DATABASES

Historians and ICS databases store sensitive information

- Attackers find value in historian data for financial gain such as
 - Recipes/formulas in pharma drug processes
 - Records of which drill sites pulled the most resources in oil and gas
 - Product designs, schematics, and measurements in manufacturing

Filesystem and database encryption help protect this data from physical theft

- Do not help if attackers access it through authorized application
- Authorized applications access data unencrypted...
- Vulnerabilities in application authentication processes
- SQL injection attacks through vulnerable applications

good for Laptops

Historians and database servers provide similar but different purposes in control systems. A historian's job is to store historical records of control processes' status messages, measurements, and any control actions issued to the control system. In turn, databases in ICS environments usually store miscellaneous data needed for application functionality, user information such as authentication credentials, and most other non-process records. Attackers are often after data stored by these systems for financial gain or enhanced exploitation through control of the control system.

Although there are several different security controls that can help protect us from such attacks, some of them aren't as helpful as they seem on the surface. Security controls like filesystem and database encryption both help protect this data from compromise if an attacker gains physical access to the server hard drives. However, these encryption defenses do not help if attackers access the historian or database through an authorized application designed to use these backend storage mechanisms. This is because the application is hardcoded to connect to these datastores, so any attacks that the attacker sends through that application are pre-authenticated to those backend datastores and access the data unencrypted. Attacks that bypass encrypted datastore controls could be vulnerabilities in application authentication processes like authentication bypass or SQL injection attacks through unprotected application inputs. Attacks through the application must be stopped by the application, or they will have access to everything the application has access to.

DEFAULT CONFIGURATIONS

All systems come with default settings to help administrators configure the systems

- Default admin username and password
- Default running services
- Default security settings

These defaults can easily be found in system documentation from the vendor

Attackers use this documentation to learn potential weaknesses in specific systems

All systems come with default settings to help administrators configure the systems. Some of these default settings can be admin accounts with preset passwords, various services running by default, and various security settings disabled or set to low. Attackers can find these defaults in system documentation from the vendor by doing simple Google searches. Attackers use this documentation to create lists of potential weaknesses to try when they find that system.

DEFAULT USERNAMES AND PASSWORDS

Many systems use "admin" or "administrator" as the default user with a preset password

- These are easy to find in system documentation
- Often easy to find with Google
- Sometimes already included in security tools

Username is rarely changed

Password is frequently still set to default

This main administrator account often becomes a shared account among admins

Many systems use "admin" or "administrator" as the default user with a preset password. Often, this password is the name of the vendor or just "password". Regardless, these passwords are easy to find in system documentation on the vendor's website. Attackers can also easily find the more common system default passwords with simple Google searches because there are many websites that keep long lists of default usernames and passwords for systems. Another place that attackers can find these passwords is in their security tools. Some security tools include long lists of username and password combinations to try on systems by default.

Most systems never have their default username changed, so attackers often will guess that the primary user is "admin" or "administrator." After their tools try the default password for the account, the attacker tools will often try common words from an English dictionary (or local language dictionary) and even words from the company's main websites. This latter technique is becoming more common and effective because the main administrator account often becomes a shared account among admins and has the password set to a word that relates to the company.

DEFAULT SERVICES

Many control systems, especially master servers, support multiple different protocols

- ICS protocols for monitoring and control
- User interfaces for administrative configuration
- Protocols for remote system updates
- Default services from the underlying operating system

Defaults usually have all available services running

Installation documentation often provides initial examples using unsecure protocols to simplify install instructions

- Telnet and FTP instead of SSH and SFTP
- HTTP instead of HTTPS

Defaults are assumed to be recommended settings

Many control systems, especially master servers, support multiple different protocols. These protocols can range from ICS protocols for monitoring and control, such as Modbus, and user interfaces for administrative configuration, like HTTP, to protocols for remote system updates like FTP. Many of these default protocols are very insecure, allowing an attacker to capture passwords during authentication or hijack sessions if they miss the authentication process.

Most master servers, when you first set them up, have all available services running, both secure and insecure.

Installation documentation for many vendors often provides initial examples using insecure protocols because they are easier to use and don't usually require pre-configuration. Protocols such as Telnet, FTP, and HTTP are often chosen instead of SSH, SFTP, and HTTPS because these services often require special server-side keys to be created and installed first. If vendors provide all their examples using these insecure protocols, it can lead ICS operators to lean toward the use of these insecure protocols, often without the full understanding of how insecure they are.

DEFAULT SECURITY SETTINGS

When ICS systems have advanced security configurations, these usually must be configured before use

- Cryptographic key expiration and renewal
- Enhanced encryption capabilities
- Administrator roles with access to all assets
- Device-to-device authentication

Sometimes, these features can't be used due to vendor incompatibilities

Sometimes, these features aren't enabled because of concerns with control process interference

When ICS systems have advanced security configurations, these usually must be configured before engineers and operators can use them. Advanced security configuration can come in a number of different categories, such as cryptographic key expiration and renewal, enhanced encryption capabilities, administrator roles with access to all assets, and device-to-device authentication. Most of these configurations require the administrator to configure all systems that will communicate with each other with the same configuration. This can become very challenging on large installations and can cause communication problems if configured incorrectly.

Another issue ICS system administrators deal with is that a lot of the advanced security features can't be used on one machine because the machine it needs to talk to is from a different vendor and doesn't support that feature. A lot of security configuration problems are introduced due to vendor incompatibilities.

And finally, sometimes these features aren't enabled because businesses or engineers might have concerns with advanced security features getting in the way or interrupting the primary control processes. At the end of the day, the control system has a job to do, and nothing can prohibit that job from being completed. For this reason, security professionals must be selective with their security controls and work to ensure they do not interrupt or hinder operations.

MISSING SECURITY PATCHES

Most master servers run on commodity operating systems

- Windows and Linux are most common
- These come out with frequent security patches
- There are usually one or two 0-day exploits out before the patches are available, which provide remote code execution
- Exploits are often created within days by security researchers using the patch to identify the flaw

Failure to install patches results in trivial exploitation

Installing these security patches is a challenge itself

These risks must be balanced and accepted by management

In a control systems environment, most master servers created in the last 15 years run on commodity operating systems such as Windows and Linux. Older systems may be running on DOS or an old version of UNIX, or possibly even older mainframe hardware. With modern systems such as Windows and Linux, these operating systems come out with security patches on at least a monthly basis. These updates usually contain at least one critical security patch that fixes remote code execution vulnerabilities, which is a vulnerability that allows an attacker to run any code he likes on the system. In addition to this, there are often one or two 0-day exploits out before the patches are even available, and there are often exploits created within days of the patch being released due to security researchers using the patch to identify the exact location and nature of the flaw.

When it comes to security patches, they can be a two-edged sword in the ICS world. Failure to install these patches can result in attackers trivially exploiting master servers if the attackers can get past perimeter defenses. However, the installation of these security patches is a security risk itself because the patch may interrupt control system processes or cause instability in the master server software. While this sounds similar to the challenges we faced years ago in traditional IT environments, it is still a real problem in the ICS world. We as an ICS industry must work through this challenge and find the right balance for each of the systems we are in charge of. Never patching is not a solution, but neither is blindly patching everything. Improving your staging and testing environments to include some of your major systems will help provide a method to test patches before deployment, but not all of our ICS systems can replicate in staging due to cost and other constraints.

REMOTE CODE EXECUTION

Remote code execution (RCE) vulnerabilities allow attackers to insert their own code into a computer's memory through another program's running process

If successful, the attacker's code runs in the context of the compromised process with the same privileges

Application whitelist defenses usually do not catch these exploits because the compromised process was already permitted to run

A favorite technique of an attacker is to execute special attack payloads like Metasploit's Meterpreter, which provides a shell, data collection tools, and the ability to jump to other running processes

Buffer overflow vulnerabilities allow attackers to insert their own code into a computer's memory through another program's running process. If successful, the attacker's code runs in the context of the compromised process with the same system privileges that the process is running. For example, if an OPC service that is running as administrator is exploited with a buffer overflow attack, the attacker's code injected into the OPC process also runs as administrator.

Application whitelist defenses usually do not catch buffer overflow attacks because the whitelist software usually checks only the original executable to verify it is permitted to run and does not monitor the process for injection, which is much more difficult to do. So, when an attacker exploits a whitelisted service, his exploits run just fine because the compromised process was already permitted to run.

A favorite technique of attackers is to execute special attack payloads such as Metasploit's Meterpreter, which provides a shell, data collection tools, and the ability to jump to other running processes. This is a much more economical method for an attacker because he can use the same initial payload for any vulnerability he finds. Then, using that Meterpreter payload, he can then decide on what he wants to do with the process.

DATABASE SECURITY

Secure historian/DB communication paths between Level 3 and 4

- Use a dedicated system in the control system DMZ
- Assume attackers will attempt to exploit this path, pen test the implementation

Harden database configurations

- Use unique username/passwords per application/database pair
- Use unique authentication certificate pairs per application/database if available
- Don't use domain admin user accounts
- Limit unsafe function calls, especially in SQL interfaces

Engage historian engineer and database administrator (DBA) in hardening

- Patch historian/DB software and associated applications
- Apply separation of duties to admins
- Oversight of remote troubleshooting and support
- Monitor and audit your processes
- Audit successful and failed logins

Historians, in particular, can be specially called out from an attack surface analysis as they often transcend network trust boundaries and exist in a less-trusted corporate network as a slave being fed from a master database in the ICS DMZ. This may provide a path for someone to compromise the ICS network.

Specific concern stems from SQL injection vulnerabilities. One way to mitigate this type of attack is to develop good database security practices. Many security issues with ICS use of databases result from configuration issues or architecture oversights. A little context can be helpful here, but it is important to note that databases may simply have been an understudied component in field work and by ICS researchers.

Traditional housekeeping applies to databases as patches may be issued by vendors that fix known bugs or are security fixes.

These are the basic database security "to do's":

- The person performing the role as the historian DBA should be fully engaged in your DB security efforts.
- DBAs should, however, fall under an appropriate scheme to ensure a separation of duties and the need to seek approval to make changes to the production environment.
- Housekeeping should include a strategy to patch the supporting server and database application.
- You should audit the DB processes and look for hardcoded application-based passwords.
- Security testing of the database should include tests against the network connection, protocols used by the DB, host server, web interfaces, check susceptibility to SQL injection, DB accounts, and certificates.
- Don't use the same certificate to encrypt/authenticate both client and server connections.

TAKEAWAYS AND RECOMMENDATIONS

Section takeaways

- Historians store time-series data from processes
- Databases can store just about anything
- Attackers target both for their sensitive data

Recommendations to owner/operators

- Enable enhanced security controls where it makes sense
- Don't forget to protect the underlying OS

Recommendations to vendors

- Support client/server certificate pairs for all DB connections
- Make DB and historian connections secure by default

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. Enforcement Zone Devices
 - Firewalls and NextGen Firewalls
 - Data Diodes and Unidirectional Gateways
2. Understanding Basic Cryptography
 - Crypto Keys
 - Encryption, Hashing and Signatures
3. Wireless Technologies
 - Satellite and Cellular
 - Mesh Networks and Microwave
 - Bluetooth and Wi-Fi
4. Wireless Attacks and Defenses
 - 3 Eternal Risks of Wireless
 - Sniffing, DoS, Masquerading, Rogue AP
5. **Exercise 3.1: Network Forensics of an Attack**
6. Purdue Level 2 and 3 Attacks
 - Historians and Databases
 - **Exercise 3.2: Bypassing Auth with SQL Injection**
 - HMI and UI Attacks
 - Web-based Attacks
 - Password Defenses
 - **Exercise 3.3: Password Fuzzing**

This page intentionally left blank.

EXERCISE 3.2: BYPASSING AUTHENTICATION WITH SQL INJECTION

DURATION TIME: 15 MINUTES

We are going to explore vulnerabilities in a web application called Dojo Control. It is a shift-logging application for engineers and operators to log what happens during their shifts. We are going to attempt to bypass authentication by leveraging an SQL injection vulnerability.

OBJECTIVES

- Test the login fields for SQL injection
- Bypass authentication and log in as the first user in the database
- Bypass authentication and log in as a specific user you specify

PREPARATION

- Start your Control Things Platform VM
- If Firefox prompts you to reset your settings, tell it no
- If FoxyProxy is enabled (fox head icon next to address bar is blue), disable it by clicking on it and choosing **Completely disable FoxyProxy**.

We are going to look at an authentication process in a web application. This web application isn't to an ICS system; the authentication process and weakness are similar to that which we find in web-based HMI or administration interface to an ICS server. If ICS personnel fail to choose strong passwords for their credentials, attackers can attempt to brute force their way into the application by fuzzing (also known as guessing) a known user's password. This is what we will be doing in this lab.

The web application we'll be looking at can be found on the Control Things virtual machine by opening a browser and typing "localhost" into the address bar. Our goal in this exercise will be to fuzz the password for user "john." We'll do this by using a security testing tool call Zed Attack Proxy (or ZAP for short) and a list of common passwords from a tool called John the Ripper.

EXERCISE 3.2: BYPASSING AUTHENTICATION WITH SQL INJECTION

TEST FOR SQL VULNERABILITIES

The screenshot shows a web application interface with a navigation menu on the left containing links for Register, Login, Pentester Help, About, Toggle Hints, Vuln List, Credits, and Reset DB. The main content area has a header that says "If you do not have an account, [Register](#)". Below this is a form with the text "Enter your username and password". The form has two input fields: "Name:" and "Password:". The "Name:" field contains a single quote character ('). The "Password:" field contains ten dots. A dark grey callout box points to the "Name:" field with the text: "1 a) Test the username field (labeled in the form as 'Name') with a **single quote** and put **aaaaaaaa** in the password". Another dark grey callout box points to the "Name:" field with the text: "1 b) This causes an SQL error message because our single quote doesn't have a second single quote to close the quotation. Notice the three single quotes in a row for username?". Below the form, a light grey box displays an error message: "You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'aaaaaaaa' at line 1". Below the error message, the SQL statement is shown: "SQL Statement:SELECT * FROM accounts WHERE username=''' AND password='aaaaaaaa'".

SANS

ICS410 | ICS/SCADA Security Essentials 120

SQL injection is a vulnerability that allows an attacker to put database commands (in the SQL query language) into an input and have the backend database run those commands. One of the easiest ways to detect basic SQL injection flaws is to simply put a single quote (and sometimes a double quote instead) into an input field and see if you get an SQL error message back. If you get an SQL error message from the backend database, you have confirmed the vulnerability exists.

STEP 1 – On the login page, try testing the name field by placing the single quote in it, and using **aaaaaaaa** or any other string in the password field. This should also give you an error message in response as seen above.

When you test for SQL injection, you only want to test one field at a time, so the **aaaaaaaa** in the password field is just to put something there to prevent the application from complaining about an empty password. The real test is the single quote in the username field as shown above. As also shown above, you should get an SQL error message in response, validating the vulnerability exists. This caused an SQL error message because our single quote doesn't have a second single quote to close the quotation. Notice the three single quotes in a row for the username field?

```
SELECT * FROM accounts WHERE username=''' AND password='aaaaaaaa'
```

The application is taking the username and password that we give it and inserting them into a pre-created SQL query created by the developer. To exploit this vulnerability, we are going to first try bypassing login without knowing either a username or a password. Normally when an application does a database lookup for a username and password, it is often looking in some credential table to see if a username and password pair exists in one of the table rows. When we see the SQL injection error message on the last step, we could confirm that this is just what the developers are doing in this application. If it returns a TRUE, then the application logs us in as the user we specified.

EXERCISE 3.2: BYPASSING AUTHENTICATION WITH SQL INJECTION

LOGGING IN AS THE FIRST USER

If you do not have an account, please register.

Enter your username and password.

Name:

Password:

2a) Type SQL code that forces every username comparison in the database to TRUE and comment out any other SQL code from the developer. Make sure you put a space after the two hyphens.

Welcome to **You are logged in as admin**

MonkeyIII

Core Controls Internal Employee Blog

For clarification, here is the developer's intended SQL query:

```
SELECT * FROM accounts WHERE username='yourusername' AND password='yourpassword'
```

We are going to exploit this by adding our own SQL code that forces every username comparison in the database to TRUE. To get rid of the password check, we will simply add an SQL comment character (the double hyphen) which puts the remaining portion of the developer's SQL query into a comment.

STEP 2 – As shown above, in the name field, type `' or 1=1; --` (there is a space after the `--`). Also, make sure you type the one single quote at the beginning. In the password field, just put in a garbage password like `aaaaaaaa` again so the application doesn't complain about not having a password. Look in the screen capture above if you have questions about what to type in the two fields. Now go ahead and click the submit button.

If you entered this query correctly, this is the SQL command the web server passes to the database:

```
SELECT * FROM accounts WHERE username='' or 1=1; -- ' AND password='aaaaaaaa'
```

However, because the database sees the double hyphens as a comment, everything after the double hyphens will be ignored, so the actual SQL command the database runs is:

```
SELECT * FROM accounts WHERE username='' or 1=1;
```

No matter which row the database checks, this will return that user's information to the application that the application interprets as a successful login. Because the database starts its comparison on the first row of the table, the application will log us in as that user, which, in most cases, is the admin user of the app.

EXERCISE 3.2: BYPASSING AUTHENTICATION WITH SQL INJECTION

LOGGING IN AS A SPECIFIC USER

Register
Login

Pentester Help

About
Toggle Hints
Vuln List
Credits
Reset DB

If you do not have an account, click here to register.

Enter your username and password.

Name:
john' --

Password:
aaaaaaaa

3a) Type SQL code that comments out the password check and looks only to see if a user named john exists. Once again, make sure you put a space after the two hyphens.

3b) Successful exploit! We logged in as john without a password.

Welcome to
You are logged in as john

I like the smell of confunk

Core Controls Internal Employee Blog

Now log out so we can exploit this vulnerability in a slightly different way. Say that the first user doesn't have the access you need; however, you know a username that does have the appropriate level of access to the application. We can use the same vulnerability to log in as that user while still bypassing the need to know his password. We do this by adding our own SQL code that simply looks for the desired username in the table and comments out the password checks again.

STEP 3 – Type `john' ; --` (there is a space after the `--`) into the username field and type `aaaaaaaa` into the password field. Once again, make sure you put a space after the two hyphens. Look in the screen capture above if you have questions about what to type in the two fields. Now go ahead and click the submit button.

```
SELECT * FROM accounts WHERE username='john' ; -- ' AND password='aaaaaaaa'
```

Everything after the double hyphens will be ignored as a comment, so we are effectively running the SQL command:

```
SELECT * FROM accounts WHERE username='john';
```

The database will start checking the accounts table, row by row, looking for the username john. Once it finds it, it will return John's information to the application, which the application interprets as a successful login.

STEP 4 – Use sqlmap for advance exploitation

For advanced users, you can try using sqlmap from a command prompt to list the available databases with the following command, which all goes on a single line. Just take the defaults of any questions you are asked.

```
sqlmap -u "http://localhost/index.php?page=login.php" --data  
"user_name=j&password=p&Submit_button=Submit" --dbs
```

EXERCISE 3.2: BYPASSING AUTHENTICATION WITH SQL INJECTION

TAKEAWAYS AND RECOMMENDATIONS

Section takeaways

- SQL injection flaws in the authentication process make it trivial to bypass that authentication
- SQL injection flaws can be used to read, change, or delete data in the database
- SQL injection flaws allow an attacker to run any SQL queries he wants
- In some cases can also read/write to files on the server and execute system commands

Recommendations to owner/operators

- Test your web applications for SQL injection and other web vulnerabilities
- Use unique username/password for each application/database pair

Recommendations to vendors

- Test your web applications for SQL injection and other web vulnerabilities
- Recommend using unique username/password for each application/database pair

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. Enforcement Zone Devices
 - Firewalls and NextGen Firewalls
 - Data Diodes and Unidirectional Gateways
2. Understanding Basic Cryptography
 - Crypto Keys
 - Encryption, Hashing and Signatures
3. Wireless Technologies
 - Satellite and Cellular
 - Mesh Networks and Microwave
 - Bluetooth and Wi-Fi
4. Wireless Attacks and Defenses
 - 3 Eternal Risks of Wireless
 - Sniffing, DoS, Masquerading, Rogue AP
5. **Exercise 3.1: Network Forensics of an Attack**
6. Purdue Level 2 and 3 Attacks
 - Historians and Databases
 - **Exercise 3.2: Bypassing Auth with SQL Injection**
 - HMI and UI Attacks
 - Web-based Attacks
 - Password Defenses
 - **Exercise 3.3: Password Fuzzing**

This page intentionally left blank.

HMI and UI Attacks

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.

HUMAN MACHINE INTERFACES (HMIS)

Access to the HMI usually means access to the process

Can be traditional indicator and switch-based

Can be serial or fieldbus-based interfaces (common at Purdue Level 2)

Can be network-based (common at Purdue Level 3)

- Most modern HMIs are now web interfaces
- Some leverage web services to a user frontend
- Some older ones may use ICS protocols like OPC

Other user interfaces in ICS

- Engineering interfaces
- Administration interfaces
- Maintenance interfaces
- Analytic applications

Human Machine Interfaces can come in a number of different forms. They can be traditional indicator lights and switch panels, serial interfaces connected to display panels, or even network-based, using protocols like RPC-based OPC and HTTP. These HMIs show us the status of our control systems and often give us a method to modify the system behaviors or manually override the system. Because of this, HMIs are a primary target for attackers.

UKRAINIAN POWER GRID ATTACKS

In the summer of 2015, attackers sent phishing emails to at least three Ukraine electric utilities

- Emails had malware, which infected via a MS Office vulnerability
- Malware installed BlackEnergy 3, providing attackers with initial foothold

Attackers stole credentials from employees

- Used these credentials to gain remote access and company VPN connectivity
- At which point attackers stopped using BlackEnergy 3

Over a six-month period, attackers mapped out the networks and planned their attacks

In December 2015, attackers initiated their attack

- Shut down power at substations using the operator HMIs
- Bricked serial gateways in Level 1/2 by uploading corrupted firmware to prevent restoration
- Ukrainian utilities had to resort to manual operations for several months

One year later, the attackers returned, and returned again a few months after that

For more detailed walkthrough and analysis of the Ukrainian power grid attacks, we have included a Defense Use Case on your student USB. It is one of many DUCs and white papers we have included on the USB for you.

DEFAULT AND WEAK PASSWORDS

Many HMIs are protected by a password

Default passwords are usually found in the vendor documentation

If HMI passwords are shared among many users, the password is often based on company terms

Weakly chosen passwords can be easy to guess with fuzzing techniques, also known as brute force or dictionary attacks

Another weakness of HMI systems is their passwords. Many HMI systems do not have passwords as they are seen as inhibitors to control engineering trying to do their work, especially in critical situations. When HMI systems do have passwords, the password is sometimes left as the default password that came preset by the vendor or is set to a password that is shared among many different users. Attackers can find default passwords in vendor documentation, and shared passwords are often set to simple words related to the company to make them easier for employees to remember. Simple words are easy to guess with various different fuzzing techniques. Fuzzing techniques are another name for brute force attacks and dictionary attacks, where an attacker uses a specialized program to make thousands of password guesses in a short period of time.

WEB-BASED ATTACKS

Many modern HMIs are now web-based

Common web vulnerabilities affect them and may even affect non-web-based applications:

- Authentication Bypass
- Weak Session Management
- SQL Injection (SQLi)
- Cross-Site Scripting (XSS)
- Cross-Site Request Forgery (CSRF/XSRF)
- Local and Remote File Inclusions (LFI and RFI)

These are a few of the OWASP Top 10 web vulnerabilities

- Open Web Application Security Project

There are many modern HMIs that now come with web-based interfaces. This makes it easier to deploy HMI stations because any workstation or laptop can become an HMI if its browser is pointed to the right location. However, there are many vulnerabilities that affect modern-day web applications that can be present in our new HMI interfaces.

In the following slides, we will discuss a number of common web vulnerabilities that affect modern web-based HMIs and may even affect non-web HMI applications and user interfaces in our control systems, including:

- Authentication Bypass
- Weak Session Management
- SQL Injection (SQLi)
- Cross-Site Scripting (XSS)
- Cross-Site Request Forgery (CSRF/XSRF)
- Local and Remote File Inclusions (LFI & RFI)

*Online tool/web Dev testing -
- Damn vuln. Web app*

Reference:

https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

AUTHENTICATION BYPASS

Authentication bypass can happen in a number of different ways

Often, it is a developer that forgets to require every webpage to verify the user is logged in

If the attacker knows the right request to send and the application doesn't verify the requester is logged in for that request, that request works without authentication

Most applications have the capability to protect their user with a username and password. This is what we call an authentication control. However, these authentication controls can be bypassed if a developer implements them incorrectly. This can happen in a number of different ways.

The most frequently found examples of this have been when a developer forgets to require every webpage to verify the user is logged in. Because every page in a web application is usually generated by a different piece of code written by a developer, every piece of code must double check the user is still logged in before it provides the requested output or action for the user. If the attacker knows the right request to send and the application doesn't verify the requester is logged in for that request, the request works without authentication. Sometimes, this might be on some benign request that may not affect the security of the system, but sometimes this might be on a highly sensitive request such as control signals. Unfortunately, it isn't too uncommon for a developer new to security to require only the login page to check for authentication and permit every other page to be submitted without authentication. In these cases, an attacker needs to know only what each of those requests looks like to make them, or often even just the right address to see the list of menu options presented after successful login.

WEAK SESSION MANAGEMENT

Once you give an application a username and a password, the application usually gives you a secure cookie with a session token

Your browser must send this cookie back to the server for every request so the server knows who you are

If the attacker can obtain your cookie or guess its contents, he can hijack your session

Once you give an application a username and a password, the application usually gives you a secure cookie with a session token. This is because HTTP is a sessionless protocol and has not built in a way to track that you are still the same person coming from the same browser on later requests. These cookies are often something like **jsessionId=a1d0c6e83f027327d8461063f4ac58a6**. Your browser must send this cookie back to the server for every request so the server knows who you are. However, if this session ID is something simple such as `userid=42` and an attacker can guess your `userid` number, the attacker would be able to hijack your session, or basically trick the server that all traffic coming from the browser is also your traffic, allowing the attacker to perform whatever action you can perform in the application. One of the favorite techniques of attackers is to immediately change your password so they can get back into the application but you cannot.

SQL INJECTION (SQLI)

Many inputs in applications are used in backend database queries

- Username and password to match correct credentials
- Search fields are used to find matching data in the database

If developers use these inputs from the user improperly, an attacker could add SQL commands in the input and have them run on the database

With SQL injection attacks, attackers can read and write to your database, and they also can often interact with your operating system and its files

Many inputs in applications are used in backend database queries. For instance, you usually type in your username and password, which the application uses in a search query to the database to see if the username and password match an existing user's username and password. Another example is search fields in applications. You may search for a field device name to have the application retrieve data on that device, but the application must first use that field device name you entered in a Structured Query Language (SQL) query to the database to verify that field device exists.

If developers use these inputs from the user improperly without properly sanitizing them through parameterized queries, an attacker can add his own SQL commands in the input and have the application run his commands on the database. With SQL injection attacks like this, attackers can not only read and write to your database, but they can also interact with your operating system and its files. This means that SQL injection vulnerabilities can provide system shells to an attacker, allow them to modify application settings by changing raw data in the database, or look for other vulnerabilities that will allow the attacker administrative access on that system or jump from the database to some other system in your network.

CROSS-SITE SCRIPTING (XSS)

Other inputs that applications get from a user may be displayed back on the page

- If you search for the term "plc53" in a search field, it may say something in response like "Here are your search results for plc53"

If developers do not properly handle that input, attackers could add JavaScript to the input and have it execute in other users' browsers

XSS attacks can do anything to the user's browser that the application can do, such as issue control signals or make configuration changes

Other types of input we often find in web applications are inputs that get information from a user and then display that information back on the page. For instance, if you search for the term **plc53** in a search field, it may say something in response like: **Here are your search results for plc53**. If developers do not properly handle that input by limiting the characters used and encoding the output before placing it back on the page, the attackers could add JavaScript to the input and have it execute in other users' browsers. So, to continue the example above, instead of searching for **plc53**, the attacker could search for `<script>alert(42)</script>` and have an alert box pop up in your browser. Now, this isn't an attack but rather a common test script attackers use to find XSS vulnerabilities. If the pop-up alert occurs, that input is vulnerable to XSS.

XSS attacks can do anything to the user's browser that the application can do. So, if you are dealing with an XSS in your HMI, stop and ask yourself, *What is the worst thing that one of my control engineers could do in this application?* Often, this is issuing some kind of critical control signal (or hundreds of them) or making configuration changes to system processes. XSS attacks may also force your users to make connections to other servers and exploit vulnerabilities like CSRF, which we will discuss next.

- Stored XSS: put command and wait someone execute it.
- reflective XSS: I tricked you click on the link
with force you execute the script.

CROSS-SITE REQUEST FORGERY (CSRF) ALSO KNOWN AS XSRF

If someone gave you a link to click on that looked like this and you clicked on it, what would you expect to happen?

- <http://www.google.com/search?q=ControlThings>

Now if someone gave you a link like this to click on and it was a valid link for the application, what would you expect to happen if you were logged in?

- <http://hmi.example.com/disconnect?meter=35499>

Now what if an attacker hid this link and tricked you to click on it, or had your browser automatically click on it using JavaScript?

This is a Cross-Site Request Forgery (CSRF) attack

If someone gives you a link to click on that looks like this and you clicked on it, what would you expect to happen?

<http://www.google.com/search?q=ControlThings>

Personally, I would expect my browser to do a Google search for "ControlThings" and show me the results. Now, what if someone gives you a link like this to click on, and it is a valid link for the smart meter HMI application? What would you expect to happen if you were logged in to that HMI? (Note, the following link is fictional and goes nowhere.)

<http://hmi.example.com/disconnect?meter=35499>

Well, if you are logged in, even if you are logged in on another tab, your browser should send that request with the proper session cookies, (because you *are* logged in) and issue the disconnect command to meter 35499. Now, what if an attacker hid this link and tricked you to click on it or had your browser automatically click on it using JavaScript so that it all happened silently in the background without you noticing? In fact, an attacker can write a little script that tells your browser to disconnect meters 1 – 10,000,000 silently in the background if he can trick you to stay on the attack page long enough; perhaps by tricking you to watch a one-hour presentation on Stuxnet. ☺

This is a Cross-Site Request Forgery (CSRF) attack. This is a normally expected feature of the HTTP protocol; however, it is a feature we don't always want, such as in the case of the smart meters. Developers must use special hidden fields called CSRF tokens to add some unpredictable input that the attack cannot guess to protect applications from this flaw.

LOCAL AND REMOTE FILE INCLUSIONS (LFI AND RFI)

Some inputs in applications are used to distinguish files on the filesystem

- For instance, you need to view the log file from 1976-06-30, so you select it in a pull-down calendar widget in the application and the application gives you that file from the historian

Attackers can request other files on the filesystem that aren't log files

If the developer doesn't block this, they gain access to sensitive files on the filesystem such as

- System users and passwords (/etc/passwd and /etc/shadow)
- Historian and database authentication credentials
- Control system settings and backup files
- Malicious executable the attacker finds a way to upload

The last category of inputs in applications that we are going to discuss are inputs used to distinguish files on the filesystem. For instance, say you need to view the log file from 1976-06-30 (my birthday), so you select that date in the pull-down calendar widget provided in the application and the application gives you the log file from the historian for that date.

If the developer uses your date input as the name of the file he pulls from the hard drive, attackers can request other files on the filesystem that aren't log files and have the application deliver those. If the developer doesn't block this by using a different name for the file on the filesystem, which he can track in the database, the attacker can gain access to sensitive files on the filesystem. Files attackers often look for such vulnerabilities are:

- System users and passwords (such as /etc/passwd and /etc/shadow on Linux and UNIX systems)
- Historian and database authentication credentials that are often found in application configuration files
- Control system settings and backup files
- Malicious executables the attacker finds a way to upload and trigger to execute through the RFI or LFI vulnerability

TAKEAWAYS AND RECOMMENDATIONS

Section takeaways

- HMIs are the primary targets for attackers
- Historians and master servers are the secondary targets

Recommendations to owner/operators

- Identify what defenses and monitor points you have between remote access and your HMIs
- Verify history length of records for remote access and HMI login

Recommendations to vendors

- Provide defenses on HMI that meet both cybersecurity and safety needs on HMIs and other user interfaces

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. Enforcement Zone Devices
 - Firewalls and NextGen Firewalls
 - Data Diodes and Unidirectional Gateways
2. Understanding Basic Cryptography
 - Crypto Keys
 - Encryption, Hashing and Signatures
3. Wireless Technologies
 - Satellite and Cellular
 - Mesh Networks and Microwave
 - Bluetooth and Wi-Fi
4. Wireless Attacks and Defenses
 - 3 Eternal Risks of Wireless
 - Sniffing, DoS, Masquerading, Rogue AP
5. **Exercise 3.1: Network Forensics of an Attack**
6. Purdue Level 2 and 3 Attacks
 - Historians and Databases
 - **Exercise 3.2: Bypassing Auth with SQL Injection**
 - HMI and UI Attacks
 - Web-based Attacks
 - Password Defenses
 - **Exercise 3.3: Password Fuzzing**

This page intentionally left blank.

Password Defenses

Applicable Standards:

- **NIST CSF v1.1:** PR.AC-1
- **ISA/IEC 62443-2-1:2009:** 4.3.3.5.1
- **ISA/IEC 62443-3-3:2013:** SR 1.1, SR 1.2, SR 1.3, SR 1.4, SR 1.5, SR 1.7, SR 1.8, SR 1.9
- **ISO/IEC 27001:2013:** A.9.2.1, A.9.2.2, A.9.2.3, A.9.2.4, A.9.2.6, A.9.3.1, A.9.4.2, A.9.4.3
- **NIST SP 800-53 Rev. 4:** AC-1, AC-2, IA-1, IA-2, IA-3, IA-4, IA-5, IA-6, IA-7, IA-8, IA-9, IA-10, IA-11
- **CIS CSC:** 1, 5, 15, 16
- **COBIT 5:** DSS05.04, DSS06.03

This page intentionally left blank.

PASSWORDS IN ICS

Typical ICS components where users may need to provide password entry include:

- HMI applications
- Equipment keypads
- Embedded web servers
- Diagnostic interfaces (Telnet, RS232, etc.)



Industrial control systems utilize passwords in some ways that are unique to these environments. While you'll use application-level passwords in ICS the same way you'd use them for enterprise applications, there are nuances to both the places passwords are used as well as the capabilities of using different types of passwords in ICS.

Many industrial control systems will have a panel with a small LCD display and buttons for configuring, monitoring, or operating the ICS. These panels may have full QWERTY keyboards or a simple numeric keypad. An image of one such device is in this slide.

Industrial control systems will also commonly have web servers or other diagnostic interfaces that sometimes are loosely documented and may not even allow modification of hardcoded passwords.

We recommend that organizations with ICS perform routine audits of new equipment and software being deployed to identify the attack surface and any points where passwords or other credentials are utilized and may need to be documented or changed to ensure compliance and security of their environment.

PASSWORD CAPABILITY IN ICS VENDOR PRODUCTS

IT Password Practices

- Require password complexity
 - Password length
 - Special characters
 - Alphanumeric
 - Uppercase lowercase
- Password reuse restrictions
- Password expiration
- Strong support with AD
- Salted hashes commonly stored
- Vendors avoid default passwords

ICS Password Practices

- Limited complexity
 - Length often limited to 6 or 8
 - Some only allow upper/lowercase
 - Some only allow numeric
 - Special chars often not permitted
- Limited password reuse
- Expiration uncommon
- Some support for RADIUS or AD
- Poor adoption for hashing
- Hardcoded defaults common

Most new IT systems have mature password capabilities allowing for each organization to set granular policies meeting their industry and organization's own standards for security. Unfortunately, most ICSs do not yet have any such functionality.

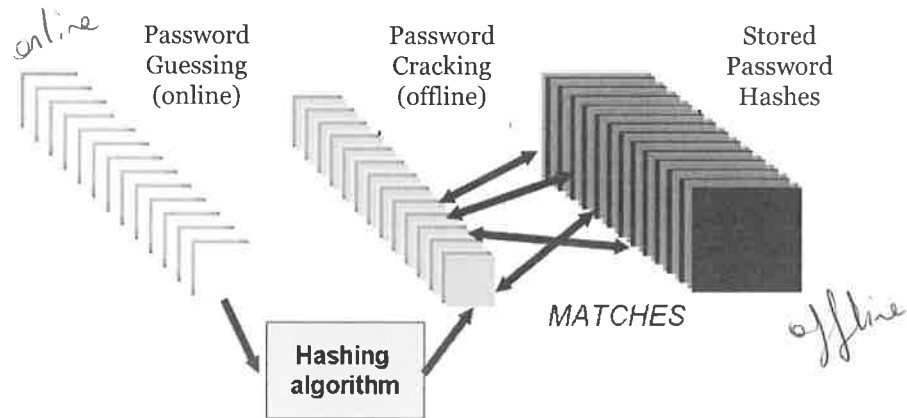
In many cases, ICS products won't allow for passwords containing special characters or exceeding a small default length limit such as eight characters.

Scariest of all, many ICS products have been analyzed by independent security researchers and found to contain hardcoded passwords, which are not documented and in some cases, cannot be changed or disabled.

On a positive note, some ICS devices have rudimentary password integration such as RADIUS.

PASSWORD GUESSING AND CRACKING

Two types of attacks on passwords



Password cracking is an offline process of attempting to guess passwords, given password file information. This section begins with a discussion of what password cracking is, why it is important, and what methods are available.

Let's back up for a moment and think about why passwords are so important. Often, passwords are the first line of defense against interactive attacks on a system. Because it is easy for someone to figure out a user ID, the only thing protecting that user's account is her password. If an attacker can gather no helpful information to aid in the attack (such as password file contents or sniffed network traffic), he must resort to either creative or brute force password guessing.

If an attacker can at least read the password file or obtain a copy, his chances of successfully obtaining an actual password increase significantly. Even if the attacker obtains only a lowly user-level password, it's fair to assume he will log in to the target system as the user and then attempt to break into the root account via local vulnerabilities.

In many companies, passwords are more than just the first line of defense; they're the only security measure protecting servers and internal information. Because most user IDs consist of an employee's first initial and last name (or something similar), it's fairly easy to discover valid user IDs for individuals at a company. Then the only other piece of information needed to gain access is a user password. So, passwords must be protected, and they must be hard to guess.

Unauthorized disclosure, unauthorized modification, and unauthorized removal are all threats to password integrity. If users disclose their passwords (intentionally or not) by writing them down or sharing them with other people, malicious parties might obtain them. It's even worse if attackers can modify the password data directly because they could change passwords without needing to know the originals. Of course, changing a password is risky for an attacker; users tend to get suspicious when their passwords suddenly stop working.

Operating systems protect passwords by using strong cryptography to hide the original content. Even if the encrypted password is revealed, it is difficult to determine the original.

AUTHENTICATION TOKENS

One-time passwords are effective against password-guessing

- Each time the user logs on, they use a different password
- Password is only good for one session

Examples include

- Smart cards/tokens
- Challenge/response
- S/KEY

These can also be used as primary or secondary authentication

One-time passwords are effective against password-guessing. Because the passwords change each time the user logs in, there is really no password to guess. The drawbacks are implementation costs and complexities and ongoing operating costs.

There are strengths and weaknesses to each of the approaches to one-time passwords. You will want to research each possibility to see which one is right for you. This section covers the approaches briefly to explain some of the options available to protect against password guessing.

The most common way to implement one-time passwords is by using token-based devices such as SecurID tokens. A user must have such a device handy when logging on to the system. The device is triggered by the time of day, so every minute the password changes. When the user wants to log on, he reads the current password from the token's display and types it in at the password prompt.

Instead of a time-based algorithm, some devices also use what is known as challenge/response. The user presents her user ID to the system, which responds with a challenge. The user then types the challenge into the device, which generates a response. The user then types the response into the system at the password prompt.

Some software-based, one-time password systems exist that are usually less expensive. Often, the software itself is free and there are no cards to buy. The users themselves usually can handle the initialization of software-based, one-time password systems, whereas a trained staff often is required to program token-based devices. One common implementation is called S/KEY, which computes its one-time passwords when the system is first configured. Each user gets a precomputed list of passwords. Each time a user logs in, he uses a different S/KEY-generated password.

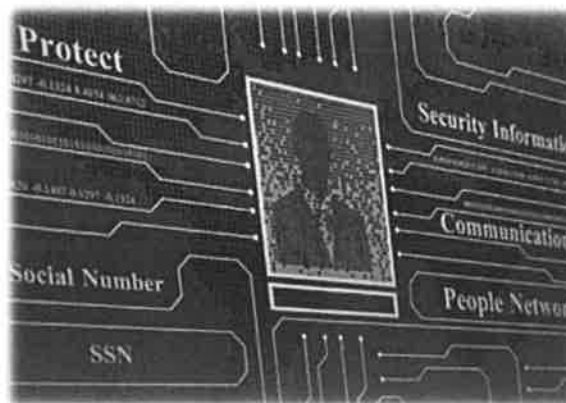
BIOMETRICS

Biometrics use physical characteristics to identify users

Common examples

- Photograph
- Fingerprint
- Iris

Also effective as primary or secondary forms of authentication



Biometric mechanisms use people's physical characteristics to identify them uniquely. The fingerprint is a commonly used physical characteristic that varies from person to person. We're probably all familiar with the finger and thumbprint scanners that take advantage of this human trait. Fingerprint scanners have advanced to the point of being incorporated into PC cards, which can be inserted into a desktop or notebook computer to authenticate users or be incorporated into the laptops themselves.

Even more common than the fingerprint biometric is the photo ID. This is useful because it can be used with "manual" biometric mechanisms. Computers need not be involved when comparing a photo to the face of a person standing there in the flesh.

Effectively, biometric authentication boils down to a long password that can never be reset. Biometric authentication systems that rely solely on biometrics for authentication are at risk when the identifying credentials for a person are stolen (such as reproducing a fingerprint left on a glass bottle) because it is not generally possible for people to change their credentials after they have been compromised. Some biometric authentication systems require a two-factor authentication (such as biometrics, and a PIN) to mitigate this risk.

Due to the unique nature of biometrics, we will discuss some other critical points in a biometrics environment.

Portal throughput is the amount of time it takes for the system to authenticate an entrant and begin processing the next entrant. Ten seconds generally are the accepted tolerable portal throughput duration. Although ten seconds seems a reasonable amount of time, at the beginning of the work day, allowing only six employees through the door each minute may not be sufficient. Contrast this with the few seconds it takes to swipe a badge.

Error rates indicate the accuracy of the biometrics system. Type A or Type I error is the percentage of readings in which the system fails to accept a genuine user. This is known as the False Reject Rate (FRR). False rejects may be due to an inability for the system to read the biometric (dirty scanner or user too far from reader).

ICS AND BIOMETRICS

Biometrics are offered by a growing number of ICS suppliers

Many operator view and access requirements have challenged adoption

Some biometric implementations suggest these uses actually improve accessibility to protected systems

- Reduced input errors
- Reduced operator fatigue

Many implementations will

- Prevent keyboard input prior to auth
- Provide read-only views through transparent locked screens



Biometrics can be a good solution for local HMIs near safety zones

Biometric mechanisms use people's physical characteristics to identify them uniquely. The fingerprint is a commonly used physical characteristic that varies from person to person. ICS adoption has been slow, as the technology was new and introduced some latency in early implementations, but improvements in the technology have addressed many concerns. In fact, some ICS supplier biometric implementations have claimed more reliable access to protected systems by reducing operator input errors associated with passwords and reducing fatigue. Complex passwords increase stress on individuals responsible for recalling and inputting them. In fact, some operators work on shifts with long dwells between on-section periods. This represents a challenge to recall passwords when you have had 7 to 10 days off.

Many ICS suppliers are beginning to offer biometrics. It is important that the supplier of the biometric solution realizes the operational requirements for engineer and operators and can effectively integrate the solution with the ICS software and hardware components. Testing and training for operational personnel is critical before implementing new biometric solutions. Some field or plant floor issues to contend with include how you accommodate contractors, technicians, and sector peer help in times of emergency or recovery from major events like storms. Plant floor and field ICS devices sometimes need to be accessible to new people to recover critical functions or put vital infrastructures like electricity back into service. Power utilities have MOUs for assistance and, after storms, line crews and technicians provide mutual aid as utilities follow industry standards. This new workforce can require access to systems.

SINGLE SIGN-ON AND RADIUS

Only have to log on once

Credentials are carried with the user

Simplifies user management

Allows for centralized management

Only have to remember one set of credentials

Should be used with multifactor authentication

Single Sign-On (SSO) is a technology that has been promising for years; however, it seems that we still have to use multiple sets of credentials on a regular basis. The SSO technology allows a user to log on once in the morning and then access any resources for which he has authorization. There is no need for him to repeat the logon procedure repeatedly.

There are different ways that Single Sign-On can be implemented. One of the older, and still common, ways is the use of scripts that will mimic the login process between different servers. It is easy to implement but has some serious security implications, as you often have credentials stored in plaintext files.

Another way that SSO is implemented is through the use of a central directory service, such as LDAP or Microsoft Active Directory. This allows the creation of a user account once on a single platform. From that single account, the user will be granted access to different platforms or services.

Kerberos, which is part of Windows 2000 and subsequent versions of the Windows operating system, can also offer the SSO through the use of tickets where credentials are stored. Not all operating systems are Kerberized or can make use of Kerberos. It is often necessary to install a third-party software package; there are some compatibility issues between the different versions of Kerberos.

SSO can save you a great amount of administrative time but will demand some initial investment in money and human resources. Ensure you select the type of technology that will properly support all of your platforms and legacy applications.

RADIUS (Remote Authentication Dial-In User Service) is largely used to authenticate accessing parties by username and password prior to granting access.

REGULATORY EXAMPLE OF PASSWORD REQUIREMENTS

NERC CIP-007 – Systems Security Management

Requires specific account management actions

- For individual accounts
 - Enforce authentication
 - Password complexity
 - Password change frequency
- For default and shared accounts
 - Inventory default and generic accounts
 - Identify individuals with access to shared accounts
 - Change default passwords
- For individual and default accounts
 - Account lockout or alerting

NERC CIP-007 Standard requires Responsible Entities to define methods to enforce authentication of interactive user access, where technically feasible. For password-only authentication for interactive user access, either technically or procedurally enforce:

- Password length the lesser of eight characters or the maximum length supported by the Cyber Asset
- Password complexity the lesser of three or more types of characters (for example, uppercase alphabetic, lowercase alphabetic, numeric, non-alphanumeric) or the maximum complexity supported by the Cyber Asset

Either technically or procedurally enforce password changes or obligation to change every 15 months

Where technically feasible, either:

- Limit the number of unsuccessful authentication attempts.
- Generate alerts after a threshold of unsuccessful authentication attempts.
- Identify and inventory all known enabled default or other generic account types.
- Identify individuals who have authorized access to shared accounts.
- Change known default passwords, per Cyber Asset capability.

TAKEAWAYS AND RECOMMENDATIONS

Section takeaways

- The process of fuzzing passwords is easy with the right tools
- Common passwords and any password based on a known word is easy to fuzz
- Passphrases are the best defense to protect against password fuzzing/guessing

Recommendations to owner/operators

- Create password policies that require strong passwords
- Enforce strong password creation where possible
- Educate users in strong passphrase creation
- Audit for weak passwords on systems that don't support strong password enforcement

Recommendations to vendors

- Support strong passphrases at least 64 characters long
- Do not restrict special characters
- Allow customers to create their own complexity requirements
- Support central authentication (AD or RADIUS) where it makes sense

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. Enforcement Zone Devices
 - Firewalls and NextGen Firewalls
 - Data Diodes and Unidirectional Gateways
2. Understanding Basic Cryptography
 - Crypto Keys
 - Encryption, Hashing and Signatures
3. Wireless Technologies
 - Satellite and Cellular
 - Mesh Networks and Microwave
 - Bluetooth and Wi-Fi
4. Wireless Attacks and Defenses
 - 3 Eternal Risks of Wireless
 - Sniffing, DoS, Masquerading, Rogue AP
5. **Exercise 3.1: Network Forensics of an Attack**
6. Purdue Level 2 and 3 Attacks
 - Historians and Databases
 - **Exercise 3.2: Bypassing Auth with SQL Injection**
 - HMI and UI Attacks
 - Web-based Attacks
 - Password Defenses
 - **Exercise 3.3: Password Fuzzing**

This page intentionally left blank.

EXERCISE 3.3: PASSWORD FUZZING

DURATION TIME: 15 MINUTES

We are going to explore vulnerabilities in a web application called Dojo Control. It is a shift-logging application for engineers and operators to log what happens during their shifts. We are going to attempt to perform a brute force attack by fuzzing a user's password.

OBJECTIVES

- Examine the raw HTTP login requests with ZAP
- Perform password fuzzing in a web application
- Analyze fuzzing output to identify which if any attempt was successful

PREPARATION

- Start your Control Things Platform VM
- If Firefox prompts you to reset your settings, tell it no

We are going to look at an authentication process in a web application. This web application isn't to an ICS system; the authentication process and weakness are similar to that which we find in web-based HMI or administration interface to an ICS server. If ICS personnel fail to choose strong passwords for their credentials, attackers can attempt to brute force their way into the application by fuzzing (also known as guessing) a known user's password. This is what we will be doing in this lab.

The web application we'll be looking at can be found on the Control Things virtual machine by opening a browser and typing "localhost" into the address bar. Our goal in this exercise will be to fuzz the password for user "john." We'll do this by using a security testing tool call Zed Attack Proxy (or ZAP for short) and a list of common passwords from a tool called John the Ripper.

EXERCISE 3.3: PASSWORD FUZZING

TRY TO LOG IN TO HTTP://LOCALHOST AS USER JOHN

Welcome to Dojo Control's Shift Logger!

Not logged in

Main Menu

[Home](#)
[Register](#)
[Login](#)

Pentester Help

[About](#)
[Toggle Hints](#)
[Vuln List](#)
[Credits](#)
[Reset DB](#)

Login

If you do not have an account, [Register](#)

Enter your username and password:

Name:

Password:

Try two or three password guesses for user "john" so you can see what the failed login page looks like

SANS

ICS410 | ICS/SCADA Security Essentials 150

Dojo Control is a web application that we have created to teach you how to perform security testing of web applications. Today, we will be using it to demonstrate password fuzzing attacks.

STEP 1 – Open the Firefox web browser on your Control Things virtual machine and type **localhost** into the Firefox address bar. Do not add a **www.** in front of it or a **.com** after it. This will get you to the web application we'll use for this exercise.

STEP 2 – Click on the **Login** link on the left-hand side of the website. Once you are on the login page, we want to try logging in as user "john" by guessing a few possible passwords he might have used.

Of course, the likelihood of you guessing the right password is fairly low for only a couple of tries; however, we are more interested in what a failed response page looks like right now. We are going to use this failed response page a little bit later in the lab, but for now just note that it says, **Bad username or password!** if we don't choose the right password. This is good because the web application isn't revealing if it's the username, the password, or both that are incorrect. We like seeing that; however, in this exercise, we are under the assumption that you have found out that **john** is a valid user account, perhaps by seeing his name on some other machine we've previously attacked, or by using LinkedIn that verified he was a control operator for this organization.

EXERCISE 3.3: PASSWORD FUZZING

PREPARING TO USE ZED ATTACK PROXY (ZAP)

Configure Firefox to use proxy Default for all URLs. Once this is done, you must start ZAP from the Activity menu's search feature before you try using Firefox again

Copyright 2018 Justin Searle
Dojo-Control is a [Control Things](#) Project.

Now we are going to use a second tool to see what the web browser is really sending to the web server when we try to authenticate. To do this, we are going to use a tool called the Zed Attack Proxy (ZAP) to see the raw HTTP requests from the web browser. Before we open up ZAP, we need to configure our Firefox browser in our Control Things virtual machine to use ZAP as its proxy. This means that Firefox will send all of its data to ZAP and trust that ZAP will send it to the correct web server.

STEP 3 – In the top part of Firefox to the right of the address bar, you will see a small red fox-like icon, which you might need to click on a double arrow icon to see. Click this and choose Options and add a new proxy setting with the following settings.

Proxy Type ★
HTTP

Color
#0055e5

Title or Description (optional)
ZAP

IP address, DNS name, server name ★
localhost

Port ★
8081

FoxyProxy

- Use proxy ZAP for all URLs (ignore patterns)
- Use proxy Default for all URLs (ignore patterns)
- Turn Off FoxyProxy (Use Firefox Settings)

Now close that tab (with the FoxyProxy settings) so you can see your Dojo Control web application again. Then go back to the FoxyProxy icon and choose use proxy ZAP for all URLs from the list of available proxies

STEP 4 – Warning, Firefox will not be able to talk to the web server now until ZAP is running, so start ZAP from the Activities main menu by clicking on the **Activities** link in the top left corner, then clicking into the **Search** box, and typing **ZAP**.

STEP 5 – When you first start ZAP, you may see a pop-up or two warning of an expired certificate and asking if you want to persist this session. If you see any of these, go ahead and just click OK or **Start**.

EXERCISE 3.3: PASSWORD FUZZING

EXAMINE THE LOGIN PAGE

The screenshot shows the ZAP tool interface. The top bar includes 'Quick Start', 'Request', and 'Response' tabs. The left sidebar shows 'Contexts' and 'Sites' with a site named 'http://localhost'. The main window displays the details of a selected request. The 'Request' tab is active, showing the raw HTTP request. The body of the request is a POST request to 'http://localhost/index.php?page=login.php' with the following headers and body:

```
POST http://localhost/index.php?page=login.php HTTP/1.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer: http://localhost/index.php?page=login.php

user_name=john&password=letmein
```

A callout box with a white background and a black border points to the request body, containing the text: "When you select a request in the History section below, you can see the request and response details in the top right part of the tool".

At the bottom of the screenshot, the 'History' tab is visible, showing a table of requests:

Id	Req. Timestamp	Met...	URL	Code	Reason	RTT	Size R...	Highes...	Note	Tags
1	1/18/18 5:52:3...	POST	http://localhost/index.php?page=login.php	200	OK	3...	3.005 ...	Medi...		Form, P...

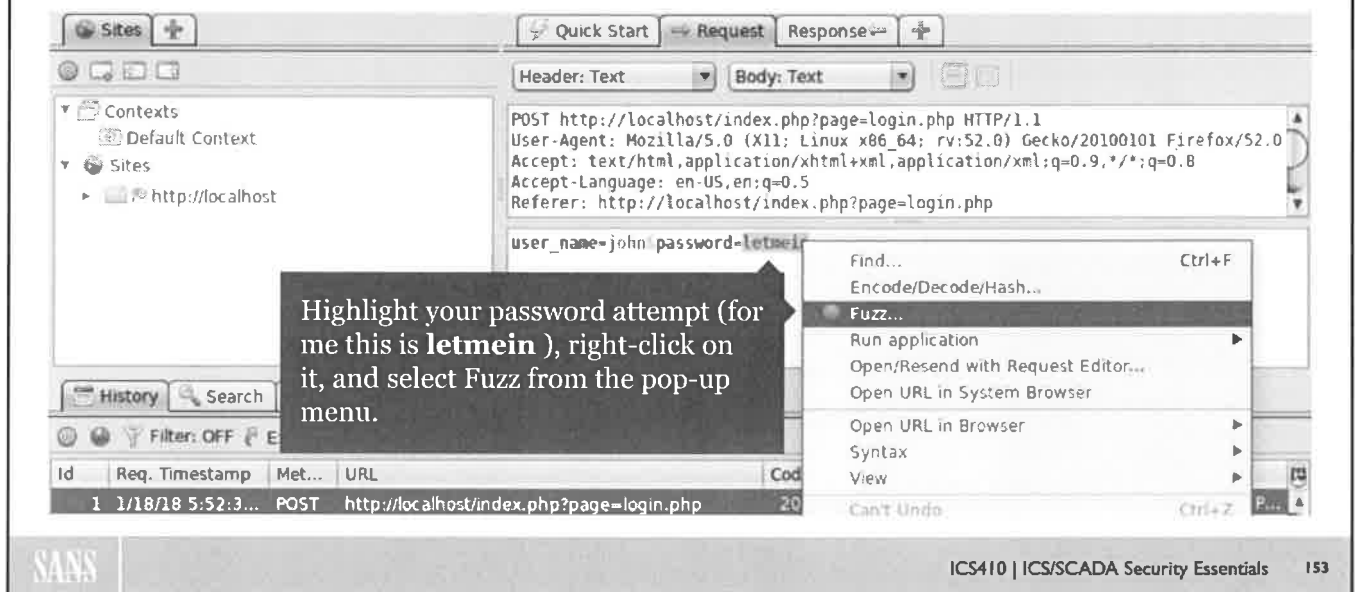
STEP 6 – Once you have ZAP up and running, go back to Firefox and try to guess the password **letmein** for user **john** in your browser, and then return to ZAP to see what the raw HTTP traffic looks like. You should see your login attempt in the bottom part of the window under the **History** tab. The screenshot above shows this. If you do not see the bottom of the ZAP tool where the History tab is supposed to be, or the right portion where the Request and Response tabs are, you may need to use the little graphical handles in ZAP to adjust the size of the sections by dragging them up and down or left and right. This sometimes happens if your screen resolution is set too low.

In the screenshot above, you can see my login attempt in the History section is selected. If you look in the upper-right portion of the screen, you'll see under the **Request** tab the raw HTTP request with the login attempt showing I tried the password **letmein**. This section where the username and password is located is called the POST parameters, and each parameter has a value. Looking closely, you'll see there are two parameters: One for the username and one for the password. Each of these parameter/value pairs is separated by an ampersand (&) character. You can see in my password guess that I guessed the password "letmein" (or "let me in" without spaces).

STEP 7 – Find one of your password guess attempts in your history and verify you can see where your password guess is in the last line of the request. You will need to find one of these before moving to the next step in this exercise because we are going to use that request as a template for our fuzzing attempts. Yours doesn't have to say **letmein** like the screenshot above, but you need to be able to know what your guess was in Firefox to find it in ZAP.

EXERCISE 3.3: PASSWORD FUZZING

USING ZAP TO FUZZ THE PASSWORD



The screenshot shows the ZAP interface with a request body containing the following text:

```
POST http://localhost/index.php?page=login.php HTTP/1.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer: http://localhost/index.php?page=login.php

user_name=john&password=letmein
```

The password 'letmein' is highlighted in blue. A context menu is open over it, with the 'Fuzz...' option selected. A callout box points to the highlighted password with the text: "Highlight your password attempt (for me this is **letmein**), right-click on it, and select Fuzz from the pop-up menu."

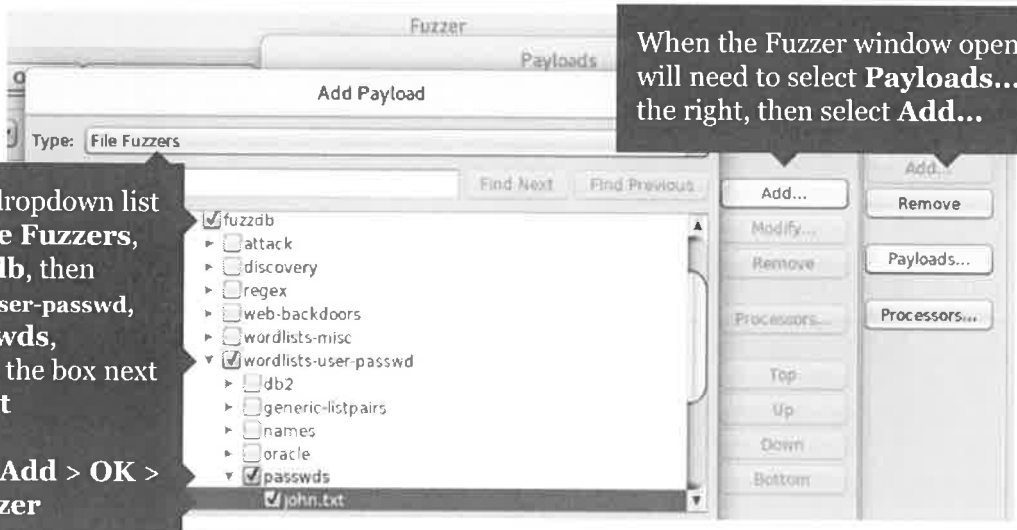
Id	Req. Timestamp	Met...	URL	Cod
1	1/18/18 5:52:3...	POST	http://localhost/index.php?page=login.php	20

STEP 8 – Once you have a sample request with one of your password guesses, highlight your password guess, right-click, and choose **Fuzz . . .** from the list of options.

This tells ZAP that you want to use this request as a template for your fuzzing attempts, and on each attempt, you want ZAP to replace the highlighted text with a new password guess.

EXERCISE 3.3: PASSWORD FUZZING

CONFIGURING YOUR FUZZER



When the Fuzzer window opens, you will need to select **Payloads...** on the right, then select **Add...**

From the dropdown list choose **File Fuzzers**, then **fuzzdb**, then **wordlists-user-passwd**, then **passwd**, then check the box next to **john.txt**

Then click **Add > OK > Start Fuzzer**

The screenshot shows the 'Fuzzer' window with 'Payloads' selected. The 'Add Payload' dialog is open, showing a tree view of fuzzer categories. Under 'File Fuzzers', 'fuzzdb' is selected and expanded. Under 'wordlists-user-passwd', 'passwd' is selected and expanded, with 'john.txt' checked. On the right, the 'Add...' button is highlighted.

Now we have to tell ZAP what it should replace the highlighted text with. ZAP comes with a large list of fuzzers, also called wordlists. The one we are most interested in for password guessing is **john.txt** in the **File Fuzzers > fuzzdb > wordlist-user-passwd > passwd** section. The other fuzzers in this category are also good lists to use for password guessing, but they are longer lists, which take much longer to run and aren't needed for this exercise.

STEP 9 – Choose the FuzzDB password category shown by checking the box next to **john.txt** and then click the **Add > OK >** and **Start Fuzzer** buttons to close all the pop-up windows and start fuzzing.

ZAP will now make one fuzzing attempt per item in the john.txt list. By the way, john.txt comes from a tool named John the Ripper, which is a tool used for cracking password hashes. It was created back in the '90s, but its list of common passwords is still surprisingly accurate today, which tells us we as humans keep failing to learn from our mistakes.

EXERCISE 3.3: PASSWORD FUZZING

EXAMINE FUZZING OUTPUT

After your fuzzer is done running, sort by one of the two **Size Resp.** columns and scroll to the top to find the unique entry and the password used

Task ID	Message Type	Code	Reason	RTT	Size Resp. Header	Size Resp. Body	Highest Alert	State	Payloads
95	Fuzzed	200	OK	10 ...	336 bytes	2,994 bytes			monkey
0	Original	200	OK	87 ...	229 bytes	3,005 bytes	Medium		
1	Fuzzed	200	OK	10...	229 bytes	3,005 bytes			12345
2	Fuzzed	200	OK	13...	229 bytes	3,005 bytes			abc123
3	Fuzzed	200	OK	28 ...	229 bytes	3,005 bytes		Reflected	password
4	Fuzzed	200	OK	24 ...	229 bytes	3,005 bytes			computer
7	Fuzzed	200	OK	78 ...	229 bytes	3,005 bytes			1234
490	Fuzzed	200	OK	21 ...	229 bytes	3,005 bytes			teresa
504	Fuzzed	200	OK	15 ...	229 bytes	3,005 bytes			wesley
509	Fuzzed	200	OK	14 ...	229 bytes	3,005 bytes			xyz123
513	Fuzzed	200	OK	8 ms	229 bytes	3,005 bytes			123123
518	Fuzzed	200	OK	18 ...	229 bytes	3,005 bytes			888888
985	Fuzzed	200	OK	8 ms	229 bytes	3,005 bytes			champion

Now there are two different ways we can examine our Fuzz output to determine whether we found the right password or not. We can either look for anomalies across all the responses, or we can look for known indicators of a successful or failed login attempt. To look for anomalies, we can look for differences in the response status, size, or sometimes even the response time (RTT).

STEP 10 – Look through all of your fuzz responses on the Fuzz tab. You should notice all the **Size Resp. Header** column sizes are **229** bytes, except one. One has a response header size of **336**. You can see the password guess of **monkey** was used in the payload column for this fuzz attempt. If we select that fuzz attempt and look in the upper right portion under the response tab, you can see we are given a **Set-Cookie** header. This is a common sign that we successfully logged in. Try going back to Firefox and using the password monkey for user john. This should log you in.

EXERCISE 3.3: PASSWORD FUZZING

TAKEAWAYS AND RECOMMENDATIONS

Section takeaways

- The process of fuzzing passwords is easy with the right tools
- Common passwords and any password based on a known word is easy to fuzz
- Passphrases are the best defense to protect against password fuzzing/guessing

Recommendations to owner/operators

- Create password policies that require strong passwords
- Enforce strong password creation where possible
- Educate users in strong passphrase creation
- Audit for weak passwords on systems that don't support strong password enforcement

Recommendations to vendors

- Support strong passphrases at least 64 characters long
- Do not restrict special characters
- Allow customers to create their own complexity requirements
- Support central authentication (AD or RADIUS) where it makes sense

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

This page intentionally left blank.

