Surviving the ICS Vulnerability Avalanche

New Technologies for New Security Threats

Eric Byres, P. Eng.
Chief Technology Office
Tofino Security, a Belden Brand
Welcome to the Patch Treadmill
The New World of Security

• For the past 30 years, Industrial Control System (ICS) products were designed for:
  ▪ safety
  ▪ reliability
  ▪ efficiency
  ▪ ease-of-use

• Security was not a design requirement:
  ▪ Protocols are not secure
  ▪ User processes not secure
  ▪ Underlying subsystems not secure

• Industry is facing a decade of security issues
IT Approach to Vulnerabilities

• In the IT world we can scan for vulnerabilities on the network
• Or our vendor announces that a vulnerability has been discovered
• Then we patch…
Let’s Scan for Vulnerabilities!

- A gas utility hired a security company to conduct penetration testing on their corporate IT network
- Consultant ventured into the SCADA network
- Penetration tool locked up the SCADA system
- Gas utility was not able to send gas through its pipelines for four hours.
Then Let’s Go Patch!

• Vulnerabilities are addressed in commercial IT world with a constant patch cycle

• Example: Adobe Reader 9.0 - 33 patches in under 4 years!
How Many Patches Does ICS Need?

- PCN in US refinery studied in Fall 2008
- 85 Computers on PCN (good data for only 78)
- 272 distinct processes
- 48 processes had entries in NVD
- 5,455 published vulnerabilities found
- 2,284 vulnerabilities after patching of OSs
How Many Patches Does ICS Need?

- What about ICS applications not in NVD?
- Est. 60,181 KLOC of non-NVD/non-OS code per PC
- Vulnerability/KLOC ratio (0.03%) used to estimate residual vulnerabilities for ICS applications not listed in the NVD

- **Estimated average of 1806 undiscovered vulnerabilities per control system computer**
- This is fertile target for Fuzzing tools like Crain’s Aegis or Codenomicon’s Defensics
Impact of Patches

• 14.8% - 24.4% of fixes for OS post-release bugs are incorrect and have impacts to end users\(^1\)

• 43% of the incorrect fixes resulted in crashes, hangs, data corruption or security problems

Impact of Patches

• Faulty patches may:
  ▪ Fail to properly resolve the vulnerabilities
  ▪ “Break” functionality that was present in previous versions

• Good patches may:
  ▪ Require shutdown and restart of process
  ▪ Remove functionality previously relied on
  ▪ Require staff with special skills to be present
Patching For Slammer Vulnerability

• Major oil company
• Numerous production platforms in Gulf of Mexico
• MS02-039 released July 2002 and company started roll-out
• Issues with server restart required Windows expert to be present
• Experts not certified for platform access
• Slammer hits 6 months later – few systems patched
ICS Vendor #1 Patching Case History

• Internal testing revealed security vulnerabilities in mission critical SIS product
• Embedded OS supplied by 3rd party
• OS vendor refused to address vulnerabilities
• No patches possible
ICS Vendor #2 Patching Case History

- PLC Vulnerability found by researcher
- Patches would remove backdoors (good)
- Backdoors used by troubleshooting teams (bad)
- Corporate QA cycle is 4 months for all changes
- Customers reluctant to patch as it may delay service and increase downtime
- 10% patch download rate for previous patches
Tofino Patch Release Case History

- Tofino version 1.6 was released on Sept 23, 2010
- Addressed security and performance issues
- Upgrade was offered to all users
  - No charge if downloaded in 30 days (installation optional)
  - All were contacted via multiple emails
  - Repeated offer for 30 days due to low initial acceptance
- Only 30% downloaded the free upgrade
Bad News About Patches for ICS

• Continuous patching will not work for ICS
• Not realistic to expect quick patching
• Vendor:
  ▪ Product QA requirements delay release
  ▪ Possible that no reasonable patch exists
  ▪ Patch impact on other critical product functions
• Customer:
  ▪ Downtime concerns
  ▪ Possible patch impact on operations or safety
  ▪ Legacy product support
  ▪ Manpower limitations
Possible Solutions to the SCADA Protocol Problem
VPN/Encryption Solutions

- Allows transmission over untrusted networks
- Provides authentication of devices
- Adds significant latency
- Garbage In / Garbage Out problem
- Complex to deploy (especially key management)
Adding Authentication Proxy Devices

😊 Provides authentication of devices
😊 Can be complex to deploy
😊 Requires per device hardware
😊 Garbage In / Garbage Out problem
Adding Authentication to SCADA Protocols

- Provides authentication of devices
- Can be complex to deploy
- Standards take a long time to change
- Requires replacement of all old equipment
- Garbage In / Garbage Out problem
Using Deep Packet Inspection Firewalls

• “White List” the Network - Use protocol-aware filters to allow only a restricted subset of possible traffic
  ☑ One device can protect multiple devices
  ☑ Can be simple to deploy
  ☑ Will block yet to be discovered vulnerabilities
  ☹ Needs DPI for each protocol

© 2013 Belden Inc. | belden.com | @BeldenInc
What is Deep Packet Inspection?
Stateful Filtering of TCP/IP Traffic

- Most firewalls are designed to filter at the TCP/UDP and IP layers:
  - Source IP Address
  - Destination IP Address
  - Destination TCP Port Number
- Also known as ACL or Layer-3/4 stateful firewalls
- Very important for good security
- BUT the upper layers or payload are NOT inspected
Deep Packet Inspection Firewalls

- Deep Packet Inspection firewalls are designed to **both** filter at the:
  - TCP/UDP and IP layers (just like a regular firewall)
  - Session, Presentation and Application layers
- First acts as Layer 3/4 firewall
- Then performs DPI
- Can inspect commands, services, objects and addresses in SCADA and process control protocols
What Deep Packet Inspection Is Not

- DPI firewalls are not an IDS like SNORT
- DPI firewalls don't use signatures.
- DPI is based on the deterministic and repeatable filtering of packets based on parsing at all layers
- Think of DPI as Wireshark in real-time
2008 DHS Warning to Energy Companies

“A vulnerability has been identified and verified within the firmware upgrade process used in control systems deployed in Critical Infrastructure and Key Resources (CIKR)… development of a mitigation plan is required to protect the installed customer base and the CIKR of the nation.

Firmware Vulnerability Mitigation Steps [include] blocking network firmware upgrades with appropriate firewall rules.”

- BUT traditional firewalls only provide complete protocol allow or deny
Bsecamp Firmware Upload Attack

- Basecamp 2012 Santamartta released 7 attacks against Controllogix PLCs:
  - Attack #1 Change the IP
  - Attack #2 Forcing a CPU Stop
  - Attack #3 Crash CPU
  - Attack #4 Dump 1756-ENBT’s module boot code
  - Attack #5 Reset 1756-ENBT module
  - Attack #6 Crash 1756-ENBT module
  - Attack #7 Firmware Update

// CIP - Unconnected send
// Service: 0x97
// Class: 0xc0
unsigned char packetDump[]="\x00\x00\x00\x00\x04\x02\x00\x00\x00\x00\x00\x00\xb2\x00\x08\x00\x97\x02\x20\xc0\x24";
MITIGATION:

“Alstom has produced a patch…”

“NCCIC/ICS-CERT encourages asset owners to take additional defensive measures…”

“Block DNP3 traffic from traversing onto business or corporate networks through the use of an IPS or firewall with DPN3-specific rule sets.”
What is Needed

• Validation of packet conformance to spec
  ▪ e.g. Block messages with invalid length fields
• Ability to filter by service, command or function
  ▪ e.g. Block messages that write to the RTU
• Ability to filter by register, object or memory location
  ▪ e.g. Block reads of register xxx
• Ability to handle messy protocols
  ▪ e.g. OPC - port hopping
  ▪ e.g. GOOSE – no IP header to filter on!
History of DPI for SCADA
modbusfw

• In 2004 Venkat Pothamsetty and Matthew Franz of Cisco created the opensource “modbusfw”
• Supported filtering for four header values: function code, unit ID, reference number and the length
• Command line based:
  ```bash
  iptables -A FORWARD -p tcp -m modbus --funccode 3 --allowtcp 1 -j ACCEPT
  ```
• Authors noted issues in correct firewall response when dropping packets
Digital Bond's SCADA IDS Signature

• Modbus TCP IDS Signatures written for Snort
• Grouped in the three categories:
  ▪ Unauthorized Modbus Use (authorized Modbus clients and servers are entered as variables)
  ▪ Modbus Protocol Errors
  ▪ Scan Detection
• Some signatures have been integrated into commercial IDS/IPS products
What was Missing

• Capability to do complex filtering such as
  ▪ Allow only HMI A to PLC B using FC3 (read) on registers 40010 – 40019 and 40030 – 40049

• Combining both strict “allow” rules and compliance checking in one solution

• Low latency

• Correct handling of sessions when packets are dropped

• Support for product edge cases

• Aggressive field testing!
Tofino Modbus TCP Enforcer

- Project started in Feb 2008
- Began field testing beta product in July 2008
- Received Modbus.org certification in Oct 2008
- Released for sale in Fall 2008
- 1000’s of units installed world wide
Tofino Modbus TCP Enforcer
What We Learned #1: DPI in the Real World

• Must make DPI work with real products, not the protocol specification
• Many SCADA products do not comply with the specifications:
  ▪ Must create “special” cases to allow them to function in field
  ▪ Must allow user control over sanity checking/state tracking
  ▪ Example: Simple dropping of disallowed packets can cause some HMIs to lockup!
  ▪ Any DPI product must be designed to handle corner cases to be accepted by field staff
What We Learned #2: Make it Simple

- Engineers rarely understand what is “on-the-wire”
- 99% of the options are never used
Developing DPI Firewalls for Ethernet/IP and CIP
EtherNet/IP and CIP

- EtherNet/IP and CIP are two protocols defined by the ODVA for industrial control systems
- EtherNet/IP is the “session” layer protocol
- CIP (Common Industrial Protocol) is the application layer protocol that runs on top of EtherNet/IP
- Other protocols, such as PCCC and Modbus, can be embedded in CIP frames.
Analysis of EtherNet/IP

- The architecture of the EtherNet/IP and CIP layers are not trivial and contain many moving parts
- Session establishment utilizes a set of commands with varying results and dynamic fields
- Session establishment spread over 3 layers, so any DPI engine must communicate between layers
- CIP in itself carries complexity based on the type of connection, whether it is using a message router or Unconnected Send affects the packet structure
EtherNet/IP Header Structure

- The EtherNet/IP header is a 24 byte fixed length header with a trailing optional data portion
- The “Command Specific Data” field encapsulates CIP

<table>
<thead>
<tr>
<th>Structure</th>
<th>Field Name</th>
<th>Data Type</th>
<th>Field Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encapsulation header</td>
<td>Command</td>
<td>UINT</td>
<td>Encapsulation command</td>
</tr>
<tr>
<td></td>
<td>Length</td>
<td>UINT</td>
<td>Length, in bytes, of the command specific data portion of the message, i.e., the number of bytes following the header</td>
</tr>
<tr>
<td></td>
<td>Session handle</td>
<td>UDINT</td>
<td>Session identification (application dependent)</td>
</tr>
<tr>
<td></td>
<td>Status</td>
<td>UDINT</td>
<td>Status code</td>
</tr>
<tr>
<td></td>
<td>Sender Context</td>
<td>ARRAY of octet</td>
<td>Information pertinent only to the sender of an encapsulation command. Length of 8.</td>
</tr>
<tr>
<td></td>
<td>Options</td>
<td>UDINT</td>
<td>Options flags</td>
</tr>
<tr>
<td>Command specific data</td>
<td>Encapsulated data</td>
<td>ARRAY of 0 to 65511 octet</td>
<td>The encapsulation data portion of the message is required only for certain commands</td>
</tr>
</tbody>
</table>
CIP Messaging Structure

- The CIP header has multiple static fields and a request path that contains a set of CIP segments
  - This includes port segments, logical segments, network segments, symbolic segments, data segments, and key segments
  - Not every packet contains every type of segment.
- There is always a CIP service executing some action upon the logical class segment
CIP Messaging Structure

- For DPI design, the logical object class is a main field of interest, with the logical segment encoding being the most important.
- The “logical segment” is an 8 bit field with:
  - The 3 highest bits denote the “logical segment type”
  - The following 3 bits denote the “logical type”
  - The last 2 bits denote the “logical format”
What’s Important?

- How can you interpret this complexity and design a usable filter mechanism to ensure credible network traffic in an efficient manner?

- Two main tasks for DPI:
  - Sanity Check – Does this message conform to spec?
  - Identify critical fields or actions - What makes sense to filter for security?

- Pick carefully - You don’t want to sanity check or filter absolutely every field or else:
  - Latency will be excessive
  - The end user will be confused with too many options
  - Configuration mistakes will occur
CIP Rule Definition Complexity

- Few PLC programmers have the background to accurately complete all the filtering fields in CIP
Make it Simple

1. Select Client
2. Select Server
3. Select Protocol
4. Select Allowed Services
5. Select Options
Need to Allow Special CIP Functions? Add an Advanced Tab
A Case History of DPI Firewalls for Ethernet/IP
Securing Pipeline Compressor Controls

- Compressor packages are complex, high risk systems
- BUT monitoring of platforms is critical in remote locations
- Risk of rogue insiders and unpatched vulnerabilities (even with VPN)
Read-only Controller Firewall

- DPI firewall inspects each SCADA message to ensure only data-read commands are allowed to the controllers.
- Requires no configuration, no network changes and no disruption to monitoring or turbine operations.
The Future of DPI for SCADA and ICS
Future Enforcers

- IEC-104
- DNP3
- ProfiNet
- S7-Comms
- 61850 MMS
- 61850 GOOSE
- HTTP for ICS
The GOOSE Enforcer

• Prototype GOOSE DPI Firewall created in 2013
• Blocks multiple attack types:
  ▪ **State Attacks**: prevents attacker from sending responses when no request has been sent
  ▪ **Invalid Devices**: filtering to ensure addresses within a GOOSE packet in valid range and ensuring that devices that are sending these messages are authorized to do so
  ▪ **Buffer Overflow Attacks**: sanity checking against malformed messages
• Under 1 mSec latency
The Enforcer Software Development Kit

- Allows easy creation of Tofino Enforcers
  - Requires no HW configuration
  - Pre-built and configured multi-homed Linux system

- Provides a common development platform including:
  - Preconfigured layer 3 & 4 firewall
  - Example deep-packet application
  - Debugging tools

- Easy queuing of pre-filtered traffic to the DPI appl
Conclusions

- There is a need for an effective security solution for IDS/SCADA systems
- Deep Packet Inspection of ICS protocols significantly increases reliability and security
- There is a way to make use of EtherNet/IP DPI without adding overwhelming complexity to the end-user
- SCADA/ICS DPI is applicable to real world situations

Don’t look for signatures – white list the network!
Questions?