A Strategy for Security Testing
Industrial Firewalls

Thuy D. Nguyen  Steve C. Austin  Cynthia E. Irvine

Department of Computer Science
Naval Postgraduate School

December 10, 2019
The views expressed in this material are those of the authors and do not reflect the official policy or position of the Naval Postgraduate School or the U.S. Government.
Topics

1. Introduction
2. Firewalls Under Test
3. Test Philosophy
4. Test Design
5. Implementation and Analysis
Motivation

Blind trust — Products meet all vendor security claims.

Industrial firewalls provide logical separation between corporate and ICS networks.

- Vulnerabilities can occur in proprietary hardware, firmware, and software
- March 2019: 10-hour DoS attack on US power grid due to unpatched firewall

Hypothesis: ICS firewalls do not always provide advertised functionality and are susceptible to exploits launched by open-source software.

Contribution: A demonstration of a repeatable methodology for testing ICS firewalls.

- Framed around functional, exception, and penetration testing
- Used to verify vendor claims on provided functionality & protection features
- Tested with two commercial ICS firewalls
Industrial protocols tested

- Modbus
- EtherNet/IP
  - CIP
  - EtherNet/IP
- Remote Method Invocation (RMI)

Source: NIST SP 800-82r2
Firewalls Under Test
Model 9211-ET consists of:

- Hardware base
- Tofino Central Management Platform
- Four loadable security modules (LSM)
  - Secure Asset Management
  - Firewall
  - Event Logger
  - Modbus TCP Enforcer
**SA Modes**

- **Predeployed**: Not configured
- **Passive**: Allow all traffic to pass through
- **Test**: Analyze traffic but does not enforce blocking policy
- **Operational**: Fully functional and blocking traffic per rulesets
- **Decommissioned**: All LSMs are deactivated; SA only listens for commands from CMP
Model TofinoXE-0200T1T1 consists of:

- Hardware base
- Tofino Configurator
- Five loadable security modules (LSM)
  - NetConnect
  - Firewall
  - Event Logger
  - Modbus TCP Enforcer
  - EtherNet/IP Enforcer
Xenon Modes

- **Passive**: Allow all traffic to pass through
- **Test**: Examine, but does not block traffic
- **Operational**: Fully functional, blocks traffic per rulesets
Product Claims

SA
- IP spoofing protection
- Rule creation
  ▶ Automatic: Based on protocols supported by CMP and PLCs
  ▶ Assisted: Based on user input derived from CMP log messages
- Secure communications between SA and CMP
  ▶ Wireshark detected SSH
- Software update must be performed via CMP update interface

Xenon
- Suggested rule creation based on observed traffic patterns
- SSH communications between Xenon and Configurator
- Software update
  ▶ Via Configurator update interface
  ▶ Directly from USB interface
Known Vulnerabilities

SA

- No CVE specific to SA
- SA uses OpenSSH v5, which has known vulnerabilities
  - CVE-2010-5107: Connection-slot exhaustion caused by fixed time limit in login logic
  - CVE-2017-15906: SFTP server allows creation of zero-length files while in read-only mode

Xenon

- SUT was automatically updated to v03.2.01 during initial installation
  - v03.2.00 fixed several CVEs
    - CVE-2017-11400: Attacker can modify USB firmware upgrade packages
    - CVE-2017-11401: Attacker can send malformed/crafted packets Modbus packets
    - CVE-2017-11402: Attacker can remotely activate rules to bypass firewall
Test Philosophy
Flaw Hypothesis Methodology (1)

- A way to conduct systematic penetration testing
- Use various forms of evidence to develop counter examples to assertions of truth about the system
  - Manuals, design documents, verification evidence, etc.
- Support different types of testing
  - Whitebox, graybox, blackbox
- Most effective if product vendors cooperate

We use the FHM as a guideline for blackbox testing of ICS firewalls
Flaw Hypothesis Methodology (2)

Technical stages
- Flaw Generation
- Flaw Confirmation
- Flaw Generation
- Flaw Elimination
How We Used FHM

Our testing was constrained to available public interfaces and documentation
- No binary analysis

Testing phases
1. Review (in detail) vendor documentation, protocols, related CVEs
2. Design tests with enumerated expected results
3. Execute tests and populate test database
4. Analyze test results (expected vs. observed)

FHM mapping
- Phase 1 → Flaw Generation
- Phases 2, 3, 4 → Flaw Confirmation
- Back end of Phase 4 → Flaw Generation
Test Design
Approach

Assumptions
- Attacker has access to corporate network
- Attacker has intimate knowledge of system and processes
- Firewall is between attacker and PLC

Scope
- Functional testing
- Exception testing
- Penetration testing

Phases of operation under test
- Discovery
- Configuration
- Operational
Test Plan (1)

Per-test description

- Test objective
- A set of preconditions that must be met before running each test
  - SUT’s mode of operation
  - Rules to be enforced by active LSMs
  - Kali Linux configuration
- Test operation to be performed
- Special conditions that affect test execution (as applicable)
  - Ex: If Modbus LSM is active, must have at least one Modbus rule to test USB load
- Expected results
Test Plan (2)

Functional testing

Objective: Verify vendor claims

- Tests using open-source tools (Nessus, Metasploit, Wireshark)
  - IP spoofing protection
  - SYN flood protection
  - Support for rule creation
  - Modbus LSM functionality
  - EtherNet/IP LSM functionality (Xenon only)
  - Secure communications between firewall and management platform

- Tests to verify mode transitions using USB device
Exception testing

Objective: Assess how SUT responds to unusual conditions

- Tests to check boundary conditions of Modbus commands and register values
  - Use Metasploit ModbusClient module
  - Send FC16 Write and FC03 Read commands with register values exceeding valid range (0-49999)

- Tests to check USB configuration load process for exceptions
Test Plan (4)

Penetration testing

Objective: Assess how SUT responds to exploits

- Tests common to both SA and Xenon
  - ARP poisoning
  - Web server stack buffer overflow
  - SSHv2 fuzzing
  - SSH enumerate users
  - SSH version scanner
  - SSH key exchange DoS
  - Remote syslog long tag DoS

- Xenon-specific tests
  - Java RMI registry interfaces enumeration
  - Java RMI server insecure endpoint code execution scanner
  - Java RMI server insecure default configuration Java code execution
### Summary of Tests

<table>
<thead>
<tr>
<th></th>
<th>D</th>
<th>C</th>
<th>O</th>
<th>UC</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SA tests</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Functional</td>
<td>4</td>
<td>4</td>
<td>9</td>
<td>5</td>
<td>22</td>
</tr>
<tr>
<td>Exception</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>Penetration</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>0</td>
<td>21</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>13</td>
<td>13</td>
<td>18</td>
<td>9</td>
<td>53</td>
</tr>
<tr>
<td><strong>Xenon tests</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Functional</td>
<td>4</td>
<td>4</td>
<td>10</td>
<td>4</td>
<td>22</td>
</tr>
<tr>
<td>Exception</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>Penetration</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>16</td>
<td>16</td>
<td>22</td>
<td>7</td>
<td>61</td>
</tr>
</tbody>
</table>

D=discovery; C=configuration; O=operational; UC=configuration via USB
Implementation and Analysis
<table>
<thead>
<tr>
<th>Exploit</th>
<th>Metasploit Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARP poisoning</td>
<td>auxiliary/spoof/arp/arp_poisoning</td>
</tr>
<tr>
<td>ABB web server stack buffer overflow</td>
<td>exploit/windows/scada/abb_wserver_exec</td>
</tr>
<tr>
<td>SSH Version 2 fuzzing</td>
<td>auxiliary/fuzzers/ssh_version_2</td>
</tr>
<tr>
<td>SSH user enumeration</td>
<td>auxiliary/scanner/ssh/ssh_enumusers</td>
</tr>
<tr>
<td>SSH version scanning</td>
<td>auxiliary/scanner/ssh/ssh_version</td>
</tr>
<tr>
<td>SSH key exchange DoS</td>
<td>auxiliary/dos/windows/ssh/sshsax_sshd_keyexchange</td>
</tr>
<tr>
<td>Rsyslog Logn Tag DoS</td>
<td>auxiliary/dos/syslog/rsyslog_long_tag</td>
</tr>
<tr>
<td>Java RMI registry interfaces enumeration</td>
<td>auxiliary/gather/java_rmi_registry</td>
</tr>
<tr>
<td>Java RMI server insecure endpoint code execution scanning</td>
<td>auxiliary/scanner/misc/java_rmi_server</td>
</tr>
<tr>
<td>Java RMI server insecure default configuration</td>
<td>exploit/multi/misc/java_rmi_server</td>
</tr>
<tr>
<td>Java code execution</td>
<td></td>
</tr>
</tbody>
</table>
# Test Results

<table>
<thead>
<tr>
<th>SA</th>
<th>Functional</th>
<th>Exception</th>
<th>Penetration</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discovery</td>
<td>P=3; F=1</td>
<td>P=2; F=0</td>
<td>P=6; F=1</td>
<td>P=11; F=2</td>
</tr>
<tr>
<td>Configuration</td>
<td>P=3; F=1</td>
<td>P=2; F=0</td>
<td>P=5; F=2</td>
<td>P=10; F=3</td>
</tr>
<tr>
<td>Operation</td>
<td>P=7; F=2</td>
<td>P=2; F=0</td>
<td>P=5; F=2</td>
<td>P=14; F=4</td>
</tr>
<tr>
<td>USB Config.</td>
<td>P=0; F=5</td>
<td>P=3; F=1</td>
<td>P=na; F=na</td>
<td>P=3; F=6</td>
</tr>
</tbody>
</table>

- P=Passed; F=Failed

<table>
<thead>
<tr>
<th>Xenon</th>
<th>Functional</th>
<th>Exception</th>
<th>Penetration</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discovery</td>
<td>P=3; F=1</td>
<td>P=2; F=0</td>
<td>P=9; F=1</td>
<td>P=14; F=2</td>
</tr>
<tr>
<td>Configuration</td>
<td>P=3; F=1</td>
<td>P=2; F=0</td>
<td>P=8; F=2</td>
<td>P=13; F=3</td>
</tr>
<tr>
<td>Operation</td>
<td>P=8; F=2</td>
<td>P=2; F=0</td>
<td>P=8; F=2</td>
<td>P=18; F=4</td>
</tr>
<tr>
<td>USB Config.</td>
<td>P=4; F=0</td>
<td>P=3; F=0</td>
<td>P=na; F=na</td>
<td>P=7; F=0</td>
</tr>
</tbody>
</table>

- P=Passed; F=Failed

Nguyen, Austin, Irvine (NPS) Security Testing ICS Firewalls 29 / 35
# SA Failed Functional Tests

<table>
<thead>
<tr>
<th>Test</th>
<th>Expected</th>
<th>Observed</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Functional testing</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[DP] SYN flood (in Passive mode)</td>
<td>SA allows all traffic</td>
<td>SA blocked exploit</td>
</tr>
<tr>
<td>[CP] SYN flood w/ PPS rate of 10</td>
<td>SA enforces PPS rate limit</td>
<td>SA blocked exploit</td>
</tr>
<tr>
<td>[OP] 1. Address spoofing – IP Only</td>
<td>1. SA blocks Nessus (FW rules)</td>
<td>1. SA blocked scan (Modbus rules)</td>
</tr>
<tr>
<td>[OP] 2. SYN flood w/ PPS rate of 10</td>
<td>2. SA enforces PPS rate limit</td>
<td>2. SA blocked exploit</td>
</tr>
<tr>
<td>[UC] 5. Mode Change via USB, O → T</td>
<td>5. Successful mode change</td>
<td>5. Unsuccessful mode change</td>
</tr>
</tbody>
</table>

**Modes:** P=Passive; T=Test; O=Operational  
**Phases:** DP=Discovery; CP=Configuration; OP=Operational  
**UC=USB Configuration**
## SA Failed Exception and Penetration Tests

<table>
<thead>
<tr>
<th>Test</th>
<th>Expected</th>
<th>Observed</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Exception testing</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[UC] Mode Change via USB, P → O</td>
<td>SA denies requested mode change</td>
<td>SA transitioned from P to O</td>
</tr>
<tr>
<td><strong>Penetration testing</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[DP] Rsyslog malformed tag DoS</td>
<td>SA allows msg to PLC; CMP accepts msg</td>
<td>SA allowed msg to PLC; CMP rejected msg</td>
</tr>
<tr>
<td>[CP] 1. Rsyslog malformed tag DoS</td>
<td>1. SA blocks msg to PLC; CMP accepts msg</td>
<td>1. SA blocked msg to PLC; CMP rejected msg</td>
</tr>
<tr>
<td>[CP] 2. ARP poisoning</td>
<td>2. Asset inventory is updated with spoofed assets; ARP table is poisoned</td>
<td>2. Asset inventory was not updated; ARP table was not poisoned</td>
</tr>
<tr>
<td>[OP] 1. Rsyslog malformed tag DoS</td>
<td>1. Same as Configuration, Test 1</td>
<td>1. Same as Configuration, Test 1</td>
</tr>
<tr>
<td>[OP] 2. ARP poisoning</td>
<td>2. Same as Configuration, Test 2</td>
<td>2. Same as Configuration, Test 2</td>
</tr>
</tbody>
</table>

Modes: P=Passive; T=Test; O=Operational / Phases: DP=Discovery; CP=Configuration; OP=Operational / UC=USB Configuration
## Xenon Failed Functional Tests

<table>
<thead>
<tr>
<th>Test</th>
<th>Expected</th>
<th>Observed</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Functional testing</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[DP] SYN flood (Passive mode)</td>
<td>Xenon allows all traffic</td>
<td>Xenon blocked exploit</td>
</tr>
<tr>
<td>[CP] SYN flood with PPS rate=10</td>
<td>Xenon enforces PPS rate limit</td>
<td>Xenon blocked exploit</td>
</tr>
<tr>
<td>[OP] 1. Address spoofing – IP Only</td>
<td>1. Xenon blocks Nessus scan per Modbus ruleset</td>
<td>1. Xenon did not block scan</td>
</tr>
<tr>
<td>[OP] 2. SYN flood with PPS rate=10</td>
<td>2. Xenon enforces PPS limit</td>
<td>2. Xenon blocked exploit</td>
</tr>
</tbody>
</table>

Modes: P=Passive; T=Test; O=Operational  /  Phases: DP=Discovery; CP=Configuration; OP=Operational  /  UC=USB Configuration
## Xenon Failed Penetration Tests

<table>
<thead>
<tr>
<th>Test</th>
<th>Expected</th>
<th>Observed</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Penetration testing</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[DP] Rsyslog malformed tag DoS</td>
<td>Xenon allows message to PLC; Configurator accepts message</td>
<td>Xenon allowed message to PLC and blocked msg to Configurator</td>
</tr>
<tr>
<td>[CP] 1. Rsyslog malformed tag DoS</td>
<td>1. Xenon blocks message to PLC; Configurator accepts msg</td>
<td>1. Xenon allowed message to PLC and blocked msg to Configurator</td>
</tr>
<tr>
<td>[CP] 2. ARP poisoning</td>
<td>2. Asset inventory is updated with spoofed assets; ARP table is poisoned</td>
<td>2. Asset inventory was unchanged; ARP table was not poisoned</td>
</tr>
<tr>
<td>[OP] 1. Rsyslog malformed tag DoS</td>
<td>1. Same as CP, Test 1</td>
<td>1. Xenon blocked messages to PLC and Configurator</td>
</tr>
<tr>
<td>[OP] 2. ARP poisoning</td>
<td>2. Same as CP, Test 2</td>
<td>2. Same as Configuration, Test 2</td>
</tr>
</tbody>
</table>

Modes: P=Passive; T=Test; O=Operational / Phases: DP=Discovery; CP=Configuration; OP=Operational / UC=USB Configuration
Summary

Conclusion
- Our tests did not reveal any major issues with the vendor claims
- Notable observations
  - IP spoofing protection only worked when both IP and MAC addresses were spoofed
  - Mode change did not behave as expected when SA was in Test mode

Future work
- Test Xenon with PLCs supporting EtherNet/IP natively
- Add fuzz testing
- Include other industrial firewalls
  - Stratix 5950 Security Appliance uses Cisco firewall technology — Known to be susceptible to common exploits, e.g., ICS-CERT Advisory ICSA-18-184-01
Questions

Thuy D. Nguyen, Naval Postgraduate School, tdnguyen@nps.edu
Steve C. Austin, austinsc1011@gmail.com
Cynthia E. Irvine, Naval Postgraduate School, irvine@nps.edu