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

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 [1]

[1] Western Electric Coordinating Council. Lesson Learned: Risks Posed by Firewall Firmware Vulnerabilities. North American Electric Reliability Corporation. Sept. 2019.

Contribution

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

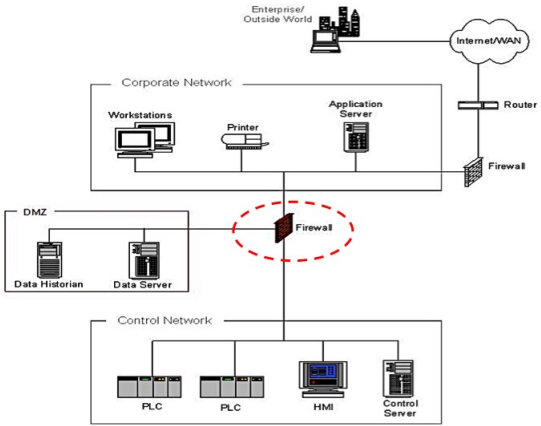
Contribution: A demonstration of a repeatable methodology for testing ICS firewalls.

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

Firewalls in ICS Network

Industrial protocols tested

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



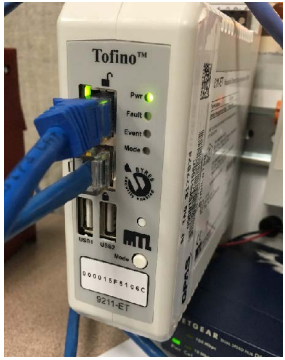
Source: NIST SP 800-82r2

Firewalls Under Test

Tofino Security Appliance (SA)

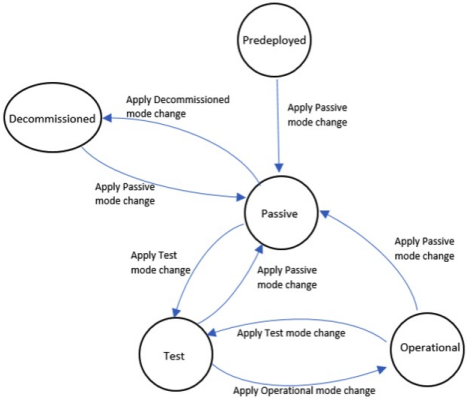
Model 9211-ET consists of:

- 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



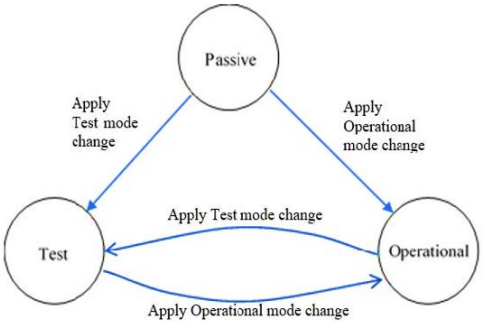
Tofino Xenon

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)

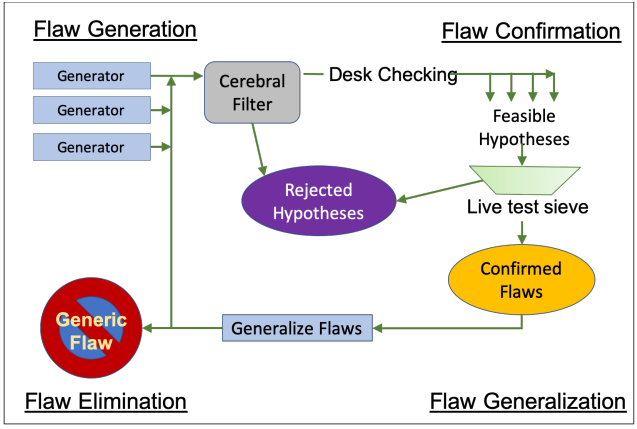
- A way to conduct systematic penetration testing
- Use various forms of evidence to develop counter examples to assertions of truth about the system
  - Manuals, design documents, verification evidence, etc.
- Support different types of testing
  - Whitebox, graybox, blackbox
- Most effective if product 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 $\longrightarrow$ Flaw Generation
- Phases 2, 3, 4 $\longrightarrow$ Flaw Confirmation
- Back end of Phase 4 $\longrightarrow$ 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

Test Plan (3)

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

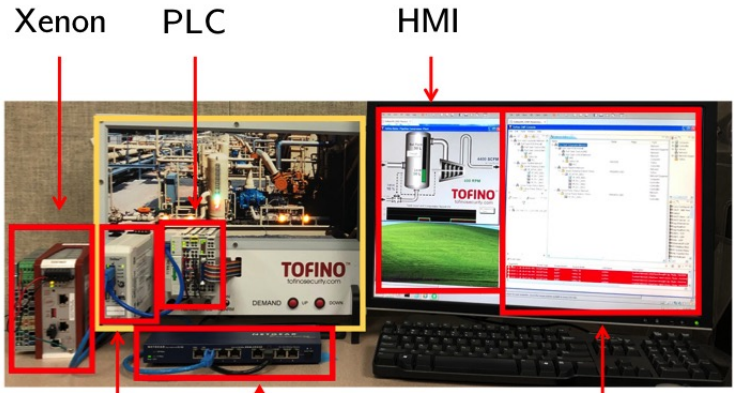
Summary of Tests

	D	C	O	UC	Total
SA tests					
Functional	4	4	9	5	22
Exception	2	2	2	4	10
Penetration	7	7	7	0	21
Total	13	13	18	9	53
Xenon tests					
Functional	4	4	10	4	22
Exception	2	2	2	3	9
Penetration	10	10	10	0	30
Total	16	16	22	7	61

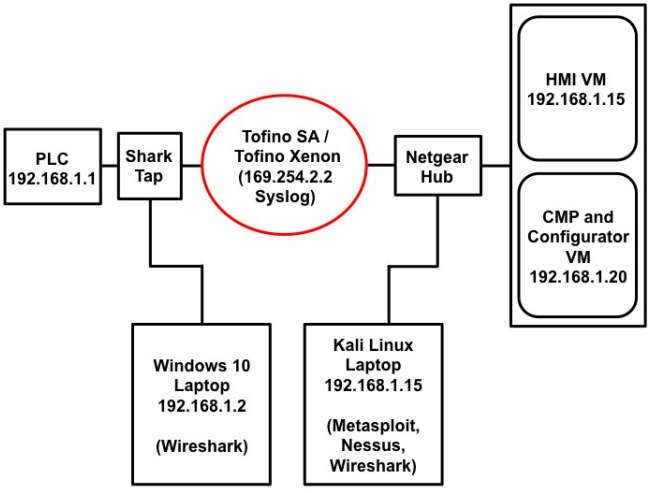
D=discovery; C=configuration; O=operational; UC=configuration via USB

Implementation and Analysis

ICS Test Network



Test Topology



Metasploit Modules Used for Penetration Testing

Exploit	Metasploit Module
ARP poisoning	auxiliary/spoof/arp/arp_poisoining
ABB web server stack buffer overflow	exploit/windows/scada/abb_wserver_exec
SSH Version 2 fuzzing	auxiliary/fuzzers/ssh_version_2
SSH user enumeration	auxiliary/scanner/ssh/ssh_enumusers
SSH version scanning	auxiliary/scanner/ssh/ssh_version
SSH key exchange DoS	auxiliary/dos/windows/ssh/ shsax_sshd_keyexchange
Rsyslog Logn Tag DoS	auxiliary/dos/syslog/rsyslog_long_tag
Java RMI registry interfaces enumeration	auxiliary/gather/java_rmi_registry
Java RMI server insecure endpoint code execution scanning	auxiliary/scanner/misc/java_rmi_server
Java RMI server insecure default configuration Java code execution	exploit/multi/misc/java_rmi_server

Test Results

SA	Functional	Exception	Penetration	Total
Discovery	P=3; F=1	P=2; F=0	P=6; F=1	P=11; F=2
Configuration	P=3; F=1	P=2; F=0	P=5; F=2	P=10; F=3
Operation	P=7; F=2	P=2; F=0	P=5; F=2	P=14; F=4
USB Config.	P=0; F=5	P=3; F=1	P=na; F=na	P=3; F=6
	P=59%; F=41%	P=90%; F=10%	P=76%; F=24%	P=72%; F=28%

P=Passed; F=Failed

Xenon	Functional	Exception	Penetration	Total
Discovery	P=3; F=1	P=2; F=0	P=9; F=1	P=14; F=2
Configuration	P=3; F=1	P=2; F=0	P=8; F=2	P=13; F=3
Operation	P=8; F=2	P=2; F=0	P=8; F=2	P=18; F=4
USB Config.	P=4; F=0	P=3; F=0	P=na; F=na	P=7; F=0
	P=82%; F=18%	P=100%; F=0%	P=85%; F=15%	P=85%; F=15%

SA Failed Functional Tests

Test	Expected	Observed
Functional testing		
[DP] SYN flood (in Pasive mode)	SA allows all traffic	SA blocked exploit
[CP] SYN flood w/ PPS rate of 10	SA enforces PPS rate limit	SA blocked exploit
[OP] 1. Address spoofing – IP Only [OP] 2. SYN flood w/ PPS rate of 10	1. SA blocks Nessus (FW rules) 2. SA enforces PPS rate limit	1. SA blocked scan (Modbus rules) 2. SA blocked exploit
[UC] 1. Mode Change via USB, P $\rightarrow$ T	1. Successful mode change	1. Unsuccessful mode change
[UC] 2. Mode Change via USB, T $\rightarrow$ O	2. Successful mode change	2. Unsuccessful mode change
[UC] 3. Mode Change via USB, T $\rightarrow$ P	3. Successful mode change	3. Unsuccessful mode change
[UC] 4. Mode Change via USB, O $\rightarrow$ P	4. Successful mode change	4. Unsuccessful mode change
[UC] 5. Mode Change via USB, O $\rightarrow$ T	5. Successful mode change	5. Unsuccessful mode change

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

SA Failed Exception and Penetration Tests

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

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

Xenon Failed Functional Tests

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

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

Xenon Failed Penetration Tests

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

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