

# Secure Validation Methodologies for ICS Patching

Mark Trump, FoxGuard Solutions

- ▶ Introduction to Patching
- ▶ Basics of a Secure Patching Plan
- ▶ Validation Methodologies



- ▶ Patching refers to the process required to address known weaknesses in a platform via updates, or “patches.”
- ▶ Once a patch (to an operating system, piece of firmware, or software application) is released, it essentially publicizes a vulnerability.
- ▶ Patches should be applied in a timely manner to minimize the risk of exploitation.
- ▶ Patching computers in an ICS is different from patching those on a corporate network.



## Risks of Patching:

- ▶ Any change to a system could cause an interruption or unpredictable behavior.



## Risks of NOT Patching:

- ▶ Stuxnet
- ▶ Conficker
- ▶ Aurora
- ▶ Night Dragon
- ▶ Code Red
- ▶ Sapphire
- ▶ SQL Slammer
- ▶ ...



- ▶ I'm not required to patch my systems.
- ▶ I don't know how to patch my systems.
- ▶ I don't think it's feasible to patch my systems.
- ▶ I don't have resources to maintain a patching program and/or the required documentation.
- ▶ I don't have a way to perform comprehensive validation of patches before they are applied.



**“Security patches** are found to be an effective means to escape the arms race as they **remediate the root cause of compromise...** an intelligent patching strategy is an **effective approach for reducing vulnerability risks.**”

Secunia ([www.secunia.com](http://www.secunia.com))

*Half Year Report | July 14, 2011*

- ▶ Assess
  - ▶ Compile
  - ▶ Validate
  - ▶ Implement
  - ▶ Document
- 
- ▶ Repeat as often as possible to ensure greatest protection from current threats...



- ▶ Identify what critical cyber/digital asset configurations must be patched.
- ▶ Topology drawings and network mapping diagrams are essential to identify critical devices.
- ▶ Consider types of attack – remote, local, USB, with/without authorization, corporate vs. control network based attacks, internal vs. external
- ▶ Determine applicability

## Systems

DCS, CEMS, Industrial Automation, etc.



## Platforms

Purpose driven devices



## Hardware

Windows®-based, Networking devices, Control systems



## Software

Operating system, Security applications (antivirus, intrusion detection), Third party software (Office, productivity, Java, Adobe), Proprietary software (SCADA software, control system code), Firmware for cyber/digital devices

- ▶ Compile patches that are applicable to your sites/systems
  - Operating system patches
  - Security software definitions and signatures
  - Third party software updates
  - Proprietary software updates
  - Firmware updates
- ▶ Patch lists are platform-specific
- ▶ Develop compilation process to ensure consistent patch deployment chronology for test and production environment

- ▶ To ensure safe deployment of patches *in the production environment*
- ▶ Quite possibly the most resource-intensive element of a secure patching plan
  - Test environment
  - Recurring labor time (quarterly... monthly...)
- ▶ Methodically test patches on a representative system in a non-production environment

- ▶ Create contingency plans
  - ▶ Backup systems
  - ▶ Tiered deployment (least to most critical)
    - Roll out updates on a lower priority system or “early adopter” in production environment
    - After observation, implement on remaining devices on the control system
- \* *Patches must be deployed as tested (in chronological order)***

- ▶ Document solutions for compliance/audit requirements and to ensure consistency
- ▶ Documentation may include:
  - Patch assessment
  - Step-by-step validation process
  - Validation results
  - Step-by-step implementation plan
  - Final checklist annotating completion of change

- ▶ Validation is more than best practice... for the energy sector, it is enforced by regulatory guidelines. (*CIP-007 R1*)
  
- ▶ Mitigate risks including:
  - Downtime
  - Profit loss
  - Safety/security issues
  - or worse.

- ▶ Set up a representative test environment
- ▶ Define or select a test standard (e.g. IEEE 829)
- ▶ Define and document test criteria
- ▶ Test to a standard

*As changes occur in the field, make sure these are reflected in the test environment and test standard.*

- ▶ Use assessment results to determine the variety of hardware, software, unique configurations, and simulation equipment necessary to replicate field conditions.
- ▶ Assemble into the most practical set of equipment needed to meet test criteria.
- ▶ Mimic network topology and connectivity.
- ▶ Adjust as needed...

- ▶ Define or select test standard that...
  - Is as comprehensive as possible.
  - Allows for non-standard configurations.
  - Can apply to all platforms.
  
- ▶ Use assessment results to determine test criteria needed to validate functionality following the change (patch/update).
  - Communications
  - Ports and services
  - Interoperability

- ▶ Process to verify functionality after a change
  - Test description
  - Test steps
  - Expected result
  - Actual result
  - Pass/Fail
  
- ▶ Upon completion of test plan
  - Document results
  - Educate team of any failed test cases
  - Create mitigation plan for failed test cases
  - Sign off to deploy

- ▶ It is unsafe to patch or make changes to critical systems without first validating the changes are safe to implement.
- ▶ In-depth knowledge of the system is required to implement a sound patch and validation process.
- ▶ Through rigid validation and deployment methodologies and documentation you can keep your system updated with the latest patches while mitigating the risk of change.

## *Questions?*

**Mark Trump** | [mtrump@foxguardsolutions.com](mailto:mtrump@foxguardsolutions.com)  
[www.foxguardsolutions.com](http://www.foxguardsolutions.com)