# ICS Security Product Evaluation Test Plan

## Introduction

This test approach document describes the appropriate strategies, process, workflows and methodologies used to plan, organize, execute and manage testing of ICS security product vendor solutions.

## Scope

Outline various desired test scenarios, expected outputs or results vs actual results and comparison of multiple ICS security vendor products against the same data set, test data (e.g. same pcaps of ICS protocol traffic and or vulnerabilities).

### In Scope

The ICS security product evaluation test plan includes:

- Testing of all functional, application performance, security and use cases requirements listed in the use case section and the *functional test cases sections and any additional use cases as necessary or encountered in operational ICS environments.*
- Determining if product will cause unwanted disruptions to ICS operations equipment, processes, personnel, protocols etcetera (including adding risk, causing safety issues and or increased attack surface etc.)

### Out of Scope

The following are considered out of scope for this particular ICS security product evaluation test plan (*however these items should be covered in another test plan focused on vulnerability assessment, reverse engineering & exploitation, bug bounty etcetera of products)*:

- Reverse engineering of products
- Exploitation of products
- Vulnerability discovery and reporting of products

## Quality Objective

### Primary Objective

A primary objective of testing application systems is to: ***assure that the product supports the desired ICS use cases and specifically the in scope functional test cases expected and desired outputs, behaviors, results and visibility for ICS operational security use cases and functional requirements needs.***

Any changes, additions, or deletions to the requirements document, Functional Specification, or Design Specification will be documented and tested at the highest level of quality allowed within the remaining time of the project and within the ability of the test team.

### Secondary Objective

The secondary objective of testing application systems will be to: ***identify and expose all issues and associated risks, communicate all known issues to the applicable stakeholders, and ensure that all issues are addressed in an appropriate matter before integrating or recommending usage of products within specific ICS operational environments.*** As an objective, this requires careful and methodical

testing of the application to first ensure all desired functional features and capabilities of the products are scrutinized and, consequently, all issues with lack of features or changes needed to features and functions found are documented and dealt with appropriately.

- Does the product meet the expected and desired functional test cases and ICS operational use cases?
- If not will the product vendor add those features or modify the existing tested features to meet the desired outputs, capabilities and requirements?

# Roles and Responsibilities

Roles and responsibilities in the context of ICS operational product functional evaluation and testing. Some of the applicable roles in scope of this test plan are below

## Product Vendor Engineers & Design Team

Creator and provider of the product under functional test and evaluation. Provides support for licensing, training on basic features, provide equipment, software etcetera needed in test environment for ICS SME and Testers to use.

## ICS operations and security SME

ICS operations and ICS security subject matter experts design the use cases and functional requirements in the ICS Security Use Cases and Functional Test Cases sections. Determines these are the features for ICS that are needed from operational field experience and those are the priority functional capabilities that each product must prove it can meet. Evaluates expected and desired features, outputs, results against the actuals that occur when functional test cases for various use cases are actually executed. Sends the requested and desired recommended changes to the Product Vendor Team for consideration before endorsement or recommendation for product usage in ICS operational environments.

## Use Case and Functional Test Case Scenario Tester/Evaluator

Can also be the ICS SMEs but if not can also be independent analyst who execute the actual functional test cases and document what they actually encountered via data outputs, screenshots, logs, reports etcetera. The actual results, features, outputs etcetera documented for each functional test case steps will be shared with the ICS SMEs to determine if recommended and requested changes are needed or if findings of existing features are acceptable and meets expected results.

# Assumptions for Test Execution

Below are assumptions for the scope of ICS security product evaluation functional testing:

- Product Vendor will provide all applicable products and resources in scope of testing for duration agreed upon with ICS team and other applicable stakeholders
- Testing environment and location will be agreed upon by all stakeholders
- Testing environment will have supporting equipment and testing data (e.g. pcaps, logs, configs, other products, power supply, switches, PLCs etc.)
- Testing environment will not negatively impact ICS production environment (including product, environmental, personnel safety)
- Functional Test Cases, Steps, expected and desired outputs, results, requirements are defined in this document and understood before testing begins

- Spare parts will be available or easy and timely to acquire if needed
- Actual results will be documented in this test plan with artifacts added as needed addendums and inserted attachments

## Constraints for Test Execution

Below are some minimum assumptions of constraints that are expected to be encountered:

- Testing in ICS operational production will be prohibited and or unsafe and unwise to do initially
- Live production data especially anonymized data for specific customer production environments may not be available so sample data and anonymized data from similar sites and operational processes, equipment and protocols etcetera will be used
- Live submission of tickets to product vendor via JIRA/Confluence etcetera may not be available so a detailed checklist of findings, needed changes needs to be documented as one of the addendums and or inserted attachments to this test plan.
- Some stakeholders with the proper ICS and ICS security and safety expertise may or may not be available so functional test cases and steps need to be written in a way that aids those less familiar with the ICS operational use cases

## Definitions

**Requirement** – desired feature or function expected from the product under test

**Use Case** – ICS operational field scenarios encountered by ICS security SMEs in real world production environments

**Functional Test Case** – specific inputs, outputs, test scenarios with step by step actions. May include elements and sub elements of an overall ICS Use Case

**Enhancement**:

1) Any alteration or modification to the existing product to enable it to meet Requirement, Use Case, Functional Test Case expected results, outputs and capabilities.

2) Any new feature that does not exist but is needed to meet Requirement, Use Case, Functional Test Case expected output, results, features and capabilities

# Test Methodology

## Purpose

### Overview

The below list is not intended to limit the extent of the test plan and can be modified to become suitable for the particular project.

The purpose of the ICS Security Product Test Plan is to achieve the following:

- Define a consistent testing strategy for each ICS security product to be tested and evaluated against Use Cases and Functional Test Cases defined in this plan. Includes all the functional and quality (non-functional) requirements.
- Divide Overall ICS operational Use Cases into testable functions and steps with expected vs actual outputs, results etcetera (do not confuse with more detailed test spec). Be sure to also identify and include areas that are to be omitted (not tested) also.
- Define capture necessary enhancements as defined in definitions section of this plan.
- Identify testing risks (to safety, equipment, personnel etcetera).
- Identify required resources (roles as defined in Roles & Responsibilities section, products from vendors, testing environment resources) related to Use Cases and Functional Test Cases in scope for the particular engagement documented in this test plan.
- Provide testing Schedule and Scope.

### ICS Usability Testing

The purpose of usability testing is to ensure that the new components and features will function in a manner that is acceptable to the customer. This is to ensure that the vendor product does not have any features that will potentially cause an outage or safety issue with the ICS OT protocols, components, and operational processes. Functional Test Cases and Use Cases include real world ICS operations cases that should be easy for the products to support without causing ICS operational issues.

## Suspension Criteria and Resumption Requirements

This section should be used to define conditions upon which suspension of testing will happen if the listed conditions occur. Each condition should have a list of requirements needed in order for testing to resume. Examples below

- Testing will be suspended on the affected products if the product fails, does not have the correct version, license and or there are no spare parts

    - Resumption: spare parts, correct version and licenses etcetera are provided in the testing environment, properly set up and configured so that the test cases can be successfully executed.

## Test Completeness

Testing will be considered complete when the following conditions have been met:

### Standard Conditions:

- When ICS SME and or Functional Tester agree that actual results meet the expected results for each test case.
- When enhancements have been provided in an update from the vendor during testing or during resumption of test cases and the actual results now match the expected results.

# REVISIONS HISTORY

| Revision No. | Date of Issue | Author | Description |
|---|---|---|---|
| 1.0 | 5 August 2020 | Isiah Jones | Creator of Test plan template |
| | | | |
| | | | |

# ICS Security Use Cases

The below are example use cases encountered in various multisector multi-asset vertical and horizontal ICS production environments globally. Functional test cases with test data should be used from many of these use cases and other use cases encountered to evaluate functional abilities or lack thereof of for proposed ICS security product vendor solutions in scope of test and evaluation.

- Alerts on Function Code Scans
- Slaves initiating communication with each other
- Slaves initiating contact with Masters without the Master having sent a request
- Logic and firmware uploads and downloads to devices with time stamps and user/service auditing
- Track creation of OPC services, modification of services, cancellation of services (from the perspective of which devices, when (timestamp etc).
- Track the programmed values of set points, tags, timers, counters from the perspective of them going out of programmed ranges (so initial alerts would be to baseline the values and the commands then only alert again if they change outside of the threshold in the baseline)
- Zone crossing rules (such as devices communicating across defined zones) and unexpected connections and communications within zones as well
- Alert if polling intervals are outside of expected norms or coming from unexpected sources etc
- Alert on new protocols within zones and between zones that weren't used before and capture the details (e.g. values, objects, function codes etc).
- Can we pull in any type of GEO location data and have a GEO map that shows a graph map with different colors for different types of alerts and asset types and protocols communicating etc?
- Alert if cyclic slaves are reporting to each other or masters outside of their designated time windows or reporting data that they weren't configured to report or contain etc (e.g. DeviceNet slaves)
- Regulation, audit requirement and policy deviations in settings or activity on devices and applications

- Map process deviations within expected time intervals
- Map normal behavior of devices and operations engineers and alert on deviations from normal expected safe behaviors
- Actively stop known bad attacks in real-time
- Field device mechanical and electrical performance fault detection and alerting
- Geo location capable live network, device and protocol communications map
- Ability to build real-time and test case attack trees
- Hardware din rail mountable rugged device in addition to a software monitoring stack
- Track ports, services, protocols, firmware, OS, real-time OS (e.g. VxWorks, Nucleus), users (not just human users), device names, station names, I/O module/card names and assignments, location information, MAC address, IP (if it has one) etc then alert if they are changed and changed by whom and when (timestamps)
- Regulation, audit requirement and policy deviations in settings or activity on devices and applications
- Work flow approval process for operations engineers to track and document detected device changes as authorized or not to continuously improve baselines
- Customizable reports especially for asset inventory management and compliance reporting, vulnerability, patch and configuration management reporting etc
-

# FUNCTIONAL TEST CASES

Below are example Functional Test Cases for ICS

| № | Steps | Expected results | Actual Test results | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | Cisco Cyber Vision | Tenable. OT (Indegy) | CyberX (Microsoft) | Radiflow | Nazomi | Cyberbit | Dragos |
| | **Pre-steps:**<br>**1.** Ensure test environment, resources and vendor products are set up and functioning, communicating etc. | | | | | | | | |
| **1.** | **Identify Existing ICS OT protocols** | | | | | | | | |
| 1.1. | 1. Determine the ICS OT protocols expected.<br>2. Locate and replay or upload a pcap where Wireshark previously successfully identified the specific ICS OT protocols desired from step 1.<br>3. Document the specific ICS OT protocol expected in this test case (e.g. Modbus, BACNet, DNP3, GOOSE, MMS, PROFINET, HART, OPC UA, Fox, LON, CIP ENIP, etc.) | Vendor product detects and identifies the ICS OT protocol including its various types of commands discovered in the pcap that Wireshark previously enumerated or parsed | | | | | | | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **2.** | **Identify ports from identified ICS OT protocols in previous test case** | | | | | | | | |
| 2.1. | 1. Select Traffic in vendor product identified from pcap used in previous test case.<br>2. Drill down into traffic details for specific ICS OT protocols identified in previous test case<br>3. Locate associated port numbers for each protocol | Visibility of each identified ICS OT protocol from previous test case with the associated port number in use for specific data set **(e.g. standard Modbus RTU runs over TCP port 502 but could be on a different port based on the ICS system, process and environment the pcap was captured from).** *Vendor product should identify Modbus in the previous test case and the port in use within this test case.* | | | | | | | |
| **3.** | **Identify specific ICS OT Devices including Type of Device** | | | | | | | | |
| 3.1. | 1. Use findings from previous test cases to select specific ICS OT protocol and identify types of devices in data set. | Vendor product should identify different types of PLCs, VFDs, HMIs, Historians, Sensor etcetera speaking the selected ICS OT protocols from previous test cases in this test plan. | | | | | | | |
| **4.** | **Determine Functional Behavior of ICS components** | | | | | | | | |
| 4.1. | 1. Use data and results from previous test cases in this test plan.<br>2. Determine potential behaviors (e.g. reads, writes, PLC code uploads/downloads, set point changes etc.). | Vendor Products Enable User to evaluate and discover ICS component behavior based on information and analysis from previous test cases in this test plan **(e.g. product should be able to parse ICS OT protocol enough to show like Wireshark that PLC upload/download occurred, on what port, between what types of devices, by what user, at what time, what MAC, IP etc.)**<br><br>*May require functional tester to either be an experienced ICS security and operations SME and or for the non SME to ensure collaboration with an ICS SME to ensure the product meets expected results in this test case.* | | | | | | | |
| **5.** | **Vendor Generated and Custom User reports.** | | | | | | | | |
| 5.1. | 1. Use data from previous test cases in this test plan.<br>2. Create or Generate Vendor product reports that includes information from previous test cases (e.g. device type, port, protocol, behavior captured etc.)<br>3. Name the report.<br>4. Create or select report export and or email format and frequency | • Vendor product report should enable user to create a report in multiple formats (e.g. PDF, excel, HMTL) and export and or email report both ad hoc and at desired repeatable scheduled frequencies as determined by the user.<br>• Report should show all information from previous test cases. | | | | | | | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | • Report should allow user to select ranges, types of devices, ports, protocols etc. | | | | | | | |
| **6.** | **Identify CVEs, Active Attacks, Rule violations etc.** | | | | | | | | |
| 6.1. | 1. Use data from previous test cases in this test plan.<br>2. Select, locate and review alerts and details for specific active malicious activity (e.g. known indicators of compromise, or known violations of ICS OT protocol like Modbus slaves identified in previous steps initiating commands with each other or without a master previously sending a request).<br>3. Locate and review any identified and mapped known CVEs<br>4. Identify in user or vendor preset rules and condition or scenario based thresholds that should trigger alerts for user analysis | • Vendor product gives user details and identifies which CVE applies to which ICS OT protocol, device, MAC, IP, port etc.<br>• Vendor product alerts on violations of rules (both preset out of the box and customer user defined rules (e.g. Modbus slaves should not initiate commands to each other)<br>• Vendor product shows active known malicious attacks and which IOCs they map to (e.g. known malware and ransomware campaign signatures)<br><br>*Not always common but the best vendor products will also show a real time evolving attack map diagram/graphic of the behavior, including a simulation feature based on discovered data points from previous test cases mapped to known existing discovered CVEs even if no active attack is taking place* | | | | | | | |
| **7.** | **Additional Test Cases as Desired (can pull from ICS Security Use Cases section or other use cases)** | | | | | | | | |
| 7.1. | 1. Use outputs from previous test cases in this test plan.<br>2. Next step?<br>3. | Expected results and requirements | | | | | | | |

# ICS Security Enhancements

In this section list out which functional test number and scenario from previous section capture actual tested results that did not meet an expected requirements and list any requested and or recommended enhancements to help the vendor products add or modify features to meet the expected and required results for those test cases.

Example provided below

| Test Case Scenario & Number | Expected Result | Actual Result | Vendor | Requested/Required Enhancements/Features/Functionality |
|---|---|---|---|---|
| **5. Vendor Generated and Custom User reports** | • Vendor product report should enable user to create a report in multiple formats (e.g. PDF, excel, HMTL) and export and or email report both ad hoc and at desired repeatable scheduled frequencies as determined by the user.<br>• Report should show all information from previous test cases.<br>• Report should allow user to select ranges, types of devices, ports, protocols etc. | User unable to create custom report with ability to select specific protocol, type of device, date/time range etc. | Tenable.OT(Indegy) | Add option for user to create custom reports where they can select specific protocol, specific types of devices, specific commands/objects/function codes/behaviors, specific CVEs etc. within a specified date/time range. |
| | | | | |
| | | | | |
| | | | | |