WHITE PAPER

Cyber Security in HMI

TABLE OF CONTENTS

1 Introduction

The purpose of this document is to describe the Test Method of EXOR Device Security Assurance Center AKA Cyber Security Assurance.

The information reflects a high-level description of EXOR test method and EXOR does not represent or warrant it to be all-inclusive, accurate or exhaustive.
EXOR is making no representations or warranties, express or implied, as to the accuracy or completeness of the information, and that EXOR will have no liability with respect to any use or reliance upon any of the information.
This is a living document and EXOR may change it at any time at its sole discretion

Goal is to improve product quality by testing robustness of the protocol stack implementation and of communication interface and performing vulnerability scanning and security testing of the software applications and HMI applications and OS developed by EXOR.

This Document also contains the Security tests done in collaboration with Exor Brand Labels and key customers.

2 Purpose and Objective

This document gives an overview of the test process followed at DSAC.

2.1 Testing Process

EXOR employs a multitude of testing techniques to find security weaknesses of product.

- Uses both open source and commercial tools for testing security of product.
- Update the testing tools weekly with central database that contain state of the art of known issues
- Automatically execute daily regression tests that includes also cyber security tests.
- If any high critical issue detected, an automatic email is notified to R&D Team and QA Team for a deep analysis.
- With every new BSP (OS) released and any major software release, a test report is archived reporting cyber security potential issues detected organized for severity.
- Classification of severity of issues is done based on policies predefined by EXOR. Any High critical issue is fixed / managed and usually fixed as part of next released.
- The test report about cyber security available for customers on demand.

2.2 Tools for testing

The below tools are using for testing and assuring security.

Tools	Purpose	License
Nmap	Reconnaissance	Free / Open Source
NIKTO	Reconnaissance	Free / Open Source
NESSUS Manager Professional	Known Vulnerability checks	Commercial
IP Stack Integrity Checker (ISIC)	Flooding & Robustness checks	Free / Open Source
Python Script	Fuzz FTP	Free / Open Source
OWASP-ZAP	Vulnerability Checks	Free / Open Source
SQL-MAP	SQL Vulnerability checks	Free / Open Source
Burp-Suite	Traffic interceptor, Manipulates and responds requests, tests the randomness of session tokens and brute forcing	Professional
Wire shark	Testing encryption over communication	Free / Open Source
HULK	Testing DoS and DDoS attacks	Free / Open Source
Metasploit	Exploitation	Free/ Open Source
Msfvenom	Exploitation (Payload creation)	Free/Open Source
Powershell Empire	Exploitation	Free/Open Source
The Harvester	OSINT	Free/Open Source
Powersploit	Exploitation (windows PowerShell)	Free/Open Source
John the ripper	Password cracking (Brute force)	Free/Open Source
Dirbuster	Directory brute force	Free/Open Source
Openvas	Vulnerability scanning	Free/Open Source

3 Test cases

The testing policy employs a multitude of testing techniques as different test cases discover a different set of bugs and vulnerabilities. All tests are conducted in a coordinated manner according to defined test procedures with analysis based on defined pass criterion.

Following is a list of different types of test cases that are executed during the robustness tests.

1. Reconnaissance Test Cases
2. Known Vulnerability Checks
3. Flooding/Storm Test Cases
4. Fuzz Test Cases

3.1 Reconnaissance Test cases

The first phase of the test involves deriving information about the target. The information gathered is used in later tests to attack services and protocols that are known to be available on that target.

This test case includes the following tests:

- Open TCP discovery
- Open UDP discovery
- Operating system and application fingerprinting

The target is subjected to common port scans to identify open ports. In its simplest form port scanning sends out a request to connect to each target port sequentially and makes a note of which ports respond. There are 216 = 65536 different TCP ports and 65536 different UDP ports. Common services listen at well-known port numbers.

Additionally, the test attempts to fingerprint the operating system and running services.

A summary of the planned tests follows.

3.1.1 Open TCP discovery

The employed TCP discovery techniques include SYN and Connect scans. These scans can identify open TCP ports in many situations. The Connect scan establishes a full TCP/IP connection for each TCP port, whereas a SYN scan does not establish the TCP three-way handshake, but rather a half-open TCP connection. The SYN scan is a much stealthier scan.

SYN scan is another form of TCP scanning. Rather than use the operating system's network functions, the port scanner generates raw IP packets itself, and monitors for responses. This scan type is also known as "half-open scanning", because it never actually opens a full TCP connection. The port scanner generates a SYN packet. If the target port is open, it will respond with a SYN-ACK packet. The scanner host responds with an RST packet, closing the connection before the handshake is completed. If the port is closed but unfiltered, the target will instantly respond with an RST packet.

3.1.2 Open UDP discovery

The User Datagram Protocol (UDP) is a stateless protocol (no three-way handshake) and hence SYN or Connect scans are not possible. It is more difficult to derive information about a target at the UDP level. This UDP discovery test sends a UDP packet to the target port to elicit a response. If the service sends a response, then the port is labeled “open”.

3.1.3 OS and Application Fingerprinting

This test case involves Nmap and Nikto attempting to identify the operating system and other services that were present on the device/ services started by software application. Nmap OS fingerprinting works by sending different probes that are specially designed to exploit various ambiguities in the standard protocol RFCs. Nikto will test a web site for thousands of possible
security issues. Including dangerous files, mis-configured services, vulnerable scripts and other issues.

3.1.4 Test Justification

The target is subjected to common port scans to identify open ports, unauthorized services and vulnerabilities. The above tests can be described as non-intrusive, but they have triggered critical errors. In some instances, devices will continuously power cycle during tests. This typically implies a weak stack implementation. For example, the Connect scan creates many logs on the target. The sheer volume of logs may overwhelm the target and the device automatically initiates a reboot to bring the device back to a healthy operational state.

3.2 Known Vulnerability Checks

The focus of this test case is to discover if products cannot withstand vulnerability testing.

A pre-defined policy forms the baseline for vulnerability scanning and includes the following tests:

- Default scan by Nessus
- Dangerous scan by Nessus
- Web Scan by Nessus
- Scada Scan by Nessus
- Vulnerability Checks by Nmap
- Owasp-top 10 Scans by Owasp-Zap
- SQL Scan by SQL-MAP
- Intercepts, Manipulates Requests and Responses by Burp-Suite

Vulnerability scanning is an automated process conducted using scanning tools and techniques to identify existing weaknesses and security holes in a product. Security holes are identified based on

published vulnerability knowledge base. A vulnerability scanner comprises of a finite number of published vulnerabilities checks/tests.

3.2.1 Default Scan

This scan is a relatively non-intrusive/passive scan and hence it is often referred as a 'safe check'. The benefit of this scan is that it provides an indication of general product robustness because Nessus attempts to identify the vulnerability in the target without actually sending a malicious payload as part of the script's execution. Importantly, all found vulnerabilities are verified and in doing so eliminates any false positives. The 'safe check' scan should not crash a device or application.

3.2.2 Dangerous scan

In addition, the dangerous scan policy subjects the target to Denial of Service (DoS) and Destructive checks. These checks may potentially crash the target by exhausting resources or sending some malformed signal that the service does not handle gracefully. It is necessary that all the reported vulnerabilities to be verified for false positives/false negatives.

3.2.3 Web Scan

Another policy called HTTP Scan consists of scripts that will be executed against Web servers that run on the products. If a product does not support web server running on it, then these checks will be skipped.

3.2.4 Scada Scan

In addition to the regular network vulnerability checks, several dozen plugins that specifically discover and test SCADA devices / applications are created into a policy called SCADA scan policy. These are available with professional version of Nessus only. The SCADA family of plugins will produce vulnerability audit data that can be leveraged for a variety of NERC and other types of security audits involving process control networks. In the power industry, this can help create various lists of products by active SCADA protocol (ICCP, DNP3, etc.).

3.2.5 Zap-Scans

OWASP ZAP is a penetration testing tool that helps developers and security professionals detect and find vulnerabilities in web applications. OWASP ZAP performs multiple security functions including

- Passively scanning web requests
- Using dictionary lists to search for files and folders on web servers

- Using crawlers to identify a site's structure and retrieve all links and URLs
- Intercepting, displaying, modifying, and forwarding web requests between browsers and web applications

ZAP can scan through the web application and detect issues related to:

- SQL injection
- Broken Authentication
- Sensitive data exposure
- Broken Access control
- Security misconfiguration
- Cross Site Scripting (XSS)
- Insecure Deserialization
- Components with known vulnerabilities
- Missing security headers

3.2.6 SQL-Map Vulnerability Scans

SQL Injection is a code injection technique where an attacker executes malicious SQL queries that control a web application's database. With the right set of queries, a user can gain access to information stored in databases. SQLMAP tests whether a 'GET' parameter is vulnerable to SQL Injection.

3.2.7 BURP suite Professional

Burp Suite is a prominent web application security solution. It gives us the ability to **Automatic scan, manually test for vulnerabilities, intercepts HTTP messages, and change a message's body and header.**

3.2.8 Wire Shark

It allows security professionals to inspect packets for signs of malicious activities, such as suspicious traffic patterns, unauthorized access attempts, or data exfiltration. By capturing and analyzing packets, security analysts can identify potential threats, assess vulnerabilities, and strengthen network defenses and allows to analyse the encryption check over the communication.

3.2.9 DoS and DDoS Checks

HULK is very different from regular pentesting tools, attack scripts, and exploit methods. HULK generates a number of unique requests at irregular intervals from the same host. So, not only does it run a SolarWinds DDoS Attack Too, the script also tries to prevent the network's defense mechanism from guessing the attack pattern. This makes it really tough to filter the traffic/packets.Helps to check DDoS protection and vulnerabilites caused by dos attacks and crashes and also with the various custom scripts.

3.2.10 Test Justification

Vulnerability scanners are useful for identifying known vulnerabilities that have not been mitigated prior to deployment. In addition, Nessus is a useful tool for identifying vulnerable configurations, default passwords, and missing patches. The above tests can lead to critical errors, such as zero-day crashes, memory leakage, resource degradation, etc. and have latest update of plugins.

Note: Nessus is used by asset owners to scan deployed systems, and hence this makes a good case for carrying out such tests.

3.3 Flooding/Storm Tests

The flooding test case simulates denial of service (DoS) attacks, in an attempt to exhaust resources, such as network bandwidth, CPU time, or memory. These tests are primarily about testing reliability and availability of the target.

Flooding involves saturating the target machine with external communications requests, such that it cannot respond to legitimate traffic, or responds so slowly as to be rendered effectively unavailable. These kinds of attacks usually lead to a server overload. Security holes are identified based on product response to ICMP ping requests and service monitor requests. Following tests are performed against the product.

Flood with Randomly generated packets at each layer

This test sends a flood of randomly generated packets at a pre-defined traffic rate. Usually this rate will vary between 1% and 10% of link utilization depending on device capability. The packets that were sent are stateless and they verify the IP Stack robustness.

Conduct Dos attacks with Regular and Malicious Packets at all layers

Main aim of this test is to verify the traffic rate at which the device will be saturated. This test conducts flooding attacks with varying traffic rates in a single test. The traffic will be generally started at 10% of link utilization and then increases up to 100% in a linear fashion. The device saturation point will be analyzed and notified to the customers. Different tests are conducted at different layers.

3.3.1 Test tools

IP Stack Integrity Checker is the tool that is used to verify the robustness of IP stack on the devices with randomly generated packets. This tool can perform the flood attack at IP, TCP and UDP layers. Atlest 1000000 packets were sent to the target in each test.

Note: On Software applications only TCP and UDP flooding are performed on services started by application to check stability of TCP/IP stack on remote system running application under stress and ability of application to handle invalid TCP/ UDP packets or TCP packets with random payload sent at higher rates.

3.3.2 Test Justification

The flooding test cases attempt to exhaust resources, such as network bandwidth, CPU time, or memory and verify the reliability and availability of the target. Since the embedded devices are usually resource constrained, it is necessary to find the saturation point at which the device cannot process any more packets and notify the customers.

3.4 Fuzzing Tests

Fuzz testing involves systematically subjecting the product to a set of semi-random/ invalid or malformed protocol packets that violate the protocol's specification. This testing technique is used to discover software security weaknesses in protocol implementations and is a well-known technique to identify zero-day vulnerabilities in the software implementation.

Fuzz testing iterates through a protocol to identify various implementation weaknesses/vulnerabilities, e.g. coding errors (buffer overflows, and format string bugs). The sending data doesn't conform to the rules or standards and particularly tests boundary conditions. The vulnerabilities found, using fuzz testing, are frequently severe.

3.4.1 Test tool

Python script for FTP Fuzz are used to send random packets to the target to load the target and check the stability.

3.4.2 Test justification

This test case allows the detection of publicly disclosed security vulnerabilities in short time periods. The benefits of fuzzing are simplicity, efficiency and automation which lead to a considerable speed-up of the whole process of detecting and isolating security vulnerabilities. The above tests can lead to critical errors, such as 0-day crashes, memory leakage, resource degradation, etc.

3.5 FTP checks

File Transfer Protocol (FTP) and File Transfer Protocol Secure (FTPS) are used for transferring files between devices.

An FTP client can open an FTP session and store and retrieve files to and from the FTP server Focus applications are large monitoring and diagnosis networks, where e.g. thousands of plants have to independently send their data to servers and may fetch files containing updates, commands, etc.).

3.5.1 Test tools

The Ftp and Ftps are tested using tools like nmap and Metasploit with the commands like [nmap -Pn -p21 --script=ftp-bounce <target_ip>], metasploit module like [auxiliary/scanner/ftp/ftp_bounce], openvas and custom scripts.

3.5.2 Test Justification

FTP uses unencrypted data transfer and, hence, user credentials and file contents can be eavesdropped on. Using FTP for official file transfer can leave your data transmission exposed to many security attacks like:

– FTP Bounce Attack
_ FTP Brute Force Attack

3.6 Exploitation

Once the vulnerability has been identified we can use some tools to exploit the vulnerable part of the application and check whether the application can be vulnerable up to the root access if possible, to exploit the application and fix the path the system in which it has been exploited.

3.6.1 Test tools

Tools like **Metasploit**, **Msfvenom**, **Powesploit** and **Powershellempire** helps in creating payloads and exploits for certain kind of vulnerability used to exploit the security bugs in the application or system.

3.6.2 Test Justification

Exploitation is a piece of programmed software or script which can allow hackers to take control over a system, exploiting its vulnerabilities. Based on the vulnerabilities, we find exploits which help us to Fix the vulnerability earlier before it is exploited by malicious actors.

3.7 Brute-force Testing

A brute force attack is a hacking method that uses trial and error to crack passwords, login credentials, and encryption keys. It is a simple yet reliable tactic for gaining unauthorized access to individual accounts and organizations' systems and networks.

3.7.1 Test tool

Tools like Dirbuster, Burpsuite (intruder) and john the ripper helps to secure the login page from brute force attacks by testing brute force using these tools with random username and password. This helps to implement security measures for the login pages.

3.7.2 Test Justification

The biggest advantages of brute force attacks is that they are relatively simple to perform and, given enough time and the lack of a mitigation strategy for the target, they always work. Every password-based system and encryption key out there can be cracked using a brute force attack. In fact, the amount of time it takes to brute force into a system is a useful metric for gauging that system's level of security.

4 Test Scope for Products

Every new product / major Release of product must go through complete test cycle. Test set is defined based on following Criteria.

- Type of product to be tested i.e. embedded device / software application / Mobile Application
- Type of interface to be tested. i.e. wired or wireless (IEEE802.11)
- Protocols supported by product.

Complete test cycle for devices includes port scanning to identify open ports and running services. Based on the scan information the device is checked for known vulnerabilities related to the services running on the device. The device is then subjected to storm/flood/DoS tests as well. The performed tests also verify the robustness of the IP stack against resource starvation, crafted malformed or semi-malformed packets, invalid/random protocol data and published vulnerability databases. In addition, the ability to maintain proper operation under the used tools is required.

Full test cycle on Software applications includes port scanning to identify ports and services started by application. Based on the scan information application is checked for known vulnerabilities related to the services started by it. The application is then subjected to TCP/UDP storm/flood/DoS tests as well to check stability of TCP/IP stack on remote system running application under stress and ability of

application to handle invalid TCP/ UDP packets. TCP/UDP based Services started by the application are then subjected to crafted malformed or semi-malformed packets, invalid/random protocol data. In addition, the ability to maintain proper operation under the used tools is required. The web applications are subjected to web vulnerability scanning and traffic interception tests.

4.1 Test Scope for Product retesting

A complete test cycle is not always necessary for updates or patches of products that have been previously tested. The reduced test set (incremental test) for product updates/patches retest is defined by considering:

- Changes of the product that have been introduced since the last Complete Test of that product. (e.g.: new hardware, a modified communication stack)
- Changes in the testing portfolio (i.e., new testing tools or test suites changes in the testing tools, techniques etc.) with respect to the last use of the tools /techniques /cases to test the product.

4.2 Test Scope for regression testing.

The purpose of regression testing is to verify that previously reported issues have been successfully fixed. The scope is confined to retesting security faults reported in the most recent test.

Integrate subset of the tests in daily automatic test for any regressions.

5 How the product meet the challenges

5.1 Cryptographic tools and security functionalities

The HMI all security features to integrate optimally into an automation network. In particular, the HMIsupports the following security functionalities:

– Support of TLS v1.3
– Signed firmware updates
– Signed boot project

5.2 Secure protocols

The HMI supports the following secure server protocols:

– FTPS (Default Port: 21)
– HTTPS (Default Port: 443)

– Custom TCP protocols secured by TLS (Default Port: User defined)

FTP AND FTPS

File Transfer Protocol (FTP) and File Transfer Protocol Secure (FTPS) are used for transferring files between devices.

An FTP client can open an FTP session and store and retrieve files to and from the FTP server.

Focus applications are large monitoring and diagnosis networks, where e.g. thousands of plants have to independently send their data to servers and may fetch files containing updates, commands, etc.).

FTP Vulnerabilities:

FTP uses unencrypted data transfer and, hence, user credentials and file contents can be eavesdropped on. FTPS requires a certificate inside the PLC and should be preferred.
Using FTP for official file transfer can leave your data transmission exposed to many security attacks like:

– FTP Bounce Attack
– FTP Brute Force Attack

FTP Reduce Risk:

– FTP is disabled by default. Do not enable it if it is not required.
– Allow only connections to known devices
– Keep the server and client software / firmware up to date

Recommendation:

It is recommended to use secure protocols like FTPS instead of FTP if possible

HTTP AND HTTPS

Hypertext Transfer Protocol (HTTP) and Hypertext Transfer Protocol Secure (HTTPS) are used to request information from a server or send information to the client.

By default, HTTP uses TCP port 80 and HTTPS uses TCP port 443.

HTTPS transmits HTTP telegrams with encryption, commonly using TLS or SSL.

The HMI uses a webserver for the web visualization. Both protocols HTTP and HTTPS are supported. In case of HTTPS, We can use the System settings>services>web server and switch on the https connection.

HTTP Vulnerabilities:

– Broken Authentication
– Cross Site Scripting (XSS)

HTTP Reduce Risk:

– HTTP is disabled by default. Do not enable it if it is not required.
– Allow only connections to known devices
– Keep the server and client software / firmware up to date

Recommendation:

– It is recommended to use secure protocols like HTTPS instead of HTTP if possible

End of Document