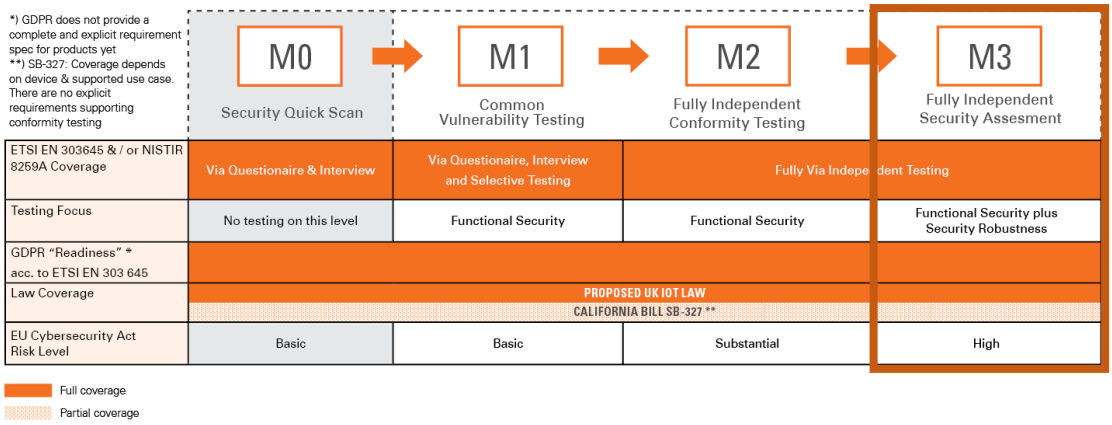
DESCRIPTION M3 TEST PROGRAM



M3 is a comprehensive customized vulnerability testing campaign backed by a penetration and security robustness testing for products with high risk exposure.

M3 is conducted in 3 steps:

1. The customer is sent a basic questionnaire to determine the functionality and capabilities of the device and the related mobile application and cloud services.
2. The customer provides appropriate test samples and all necessary information. Security experts from SGS perform an independent security assessment of the product.
3. Based on the assessment results SGS is preparing a technical report.

The purpose of this test program is to provide a customized and independent, thorough and state-of-the-art security assessment of the product, including all components and interfaces of the device. Where M0-M2 focus on the conformity assessment to specific security standards, M3 gives a detailed view on the state of the device's security. Several security experts from various fields are part of the security evaluation. The range of tests and applied testing methodologies, respectively, is most efficiently done in a white-box setting providing the highest level of security assurance. The more information the evaluators get access to, the more tests can be performed. Naturally, the duration of a security assessment, especially that of a penetration test, cannot be exactly determined a priori. In accordance with customer deadlines, the assessment is time limited where various activities are conducted in parallel. Any discovered findings which would require more time for investigations, will be mentioned accordingly.

The security assessment is done in several phase:

1. **Security requirements analysis**
   The security requirements are usually provided by the customer. The requirements document describes the expected security features and lists assets which need to be protected. Security standards like EN 303 645 can also function as a security requirements document.
2. **Setup, initialization and investigation**
   Security evaluators setup and initialize the device and the test environment. They will make themselves familiar with the product. They will get an understanding of the solution and its functionality.

3. **Documentation analysis**
   Provided documents are analysed for security relevant information.
4. **Functional tests**
   Security functions implementing the security requirements are tested. Specific requirements might need detailed information about the product leading to a white-box setting.
5. **Vulnerability analysis**
   a. Vulnerability identification: identify any vulnerabilities based on the collected information.
   b. Conduct a penetration test of the device. Validate the existence of the identified vulnerabilities.

For the most effective execution, we require 5 samples of the devices since several activities can be performed in parallel.

In the following section we provide an overview of the vulnerability analysis performed.

DEVICE TESTING

The security of an IoT device depends on all components of its ecosystem. The same also holds for the components of the device itself which consist of the following elements:

- **Network**: interfaces and protocols used for remote communication.
- **Firmware**: software and operating system of the device.
- **Hardware**: the device hardware, e.g., chip set, storage, JTAG, UART ports, sensors, USB interfaces etc.
- **Cryptography**: cryptographic algorithms as well as protocols and their implementation to support security functions.

Testing activities are split amongst these elements.

Network

These activities are focused on network-based attacks on remote interfaces, like Wi-Fi, Ethernet as well as BLE or ZigBee. They include (but are not limited to):

- *Automated vulnerability scanning*: commercial and open-source vulnerability scanners are used to identify any vulnerable services on the remote interfaces.
- *Communication*: the communication is analysed in order to verify that the information transmitted is protected in terms of integrity, authenticity, and confidentiality.
- *Wireless personal area networks*: interfaces like BLE and ZigBee require specific tests and equipment. The configuration and security of these interfaces is assessed for example by spoofing, passive and active transmission interception, or abuse of improperly configured device's services.
- *Port scanning*: the interfaces are scanned to determine running services.
- *Assessment of running services*: identified services are further analysed. For example,
  - a discovered web server will be tested against common vulnerabilities,
  - a discovered SSH service will undergo a brute force attack,
  - a discovered web application will be tested for OWASP Top 10 Web Application Security Risks,
  - etc.

- *Malformed input testing*: fuzzing frameworks are applied to test the robustness of protocols and services running via remote interfaces. During these tests, the device under test should continue normal operation according to its intended use. The device under test should not show unexpected behaviour such as crashes, non-responsive hangs, exceptions, connection loss, or disclosure of sensitive data.

Firmware Analysis

These activities focus on the software on the device and its configuration. A requirement for the analysis is to have access to the firmware. Test activities include (but are not limited to):

- *Static code analysis*: if source code is available, static code analysis tools are applied searching for software vulnerabilities, like buffer overflows, missing input validation, hard coded credentials and more.
- *Binary analysis*: the firmware is analysed and methods from reverse engineering are applied to investigate the application and its working principles in more detail. Thereby, it is checked, for example, whether the application contains hardcoded cryptographic keys, passwords, or other sensitive information. A software composition analysis is performed, identifying software components in the binary, like operating system, services, and third-party libraries. Public research and vulnerability databases are consulted to identify security vulnerabilities and weaknesses of the software components.
- *Default configuration*: the security of the device in its default configuration is analysed.
- *Update mechanism*: it is checked whether authenticity and integrity mechanisms for the implemented update mechanism are implemented properly. Furthermore, it is checked whether a rollback protection has been implemented.
- *Logging*: if the device supports logging of security-relevant events, it is verified that events such as login-attempts or change of credentials are properly logged and that log files cannot be tampered with, if access to this level can be obtained.
- *Authentication mechanisms*: for implemented authentication mechanisms, basic cybersecurity principles are investigated such as whether a password complexity policy exists, whether a time-out to prevent brute-force attacks exists, whether hardcoded credentials exist that cannot be changed. Furthermore, the user management and the revocation of user authorizations is analysed.
- *Secure storage*: it is checked how sensitive information like security credentials are stored on the device.
- *Decommissioning*: in case there is a dedicated process for the decommissioning of the device, the (secure) deletion of all data (config data and sensitive data), if access to this level can be obtained, is verified.

Hardware Analysis

These activities focus on the hardware of the device. Test activities include:

- *Identification of components and interfaces*: the device is opened, and checked for suspicious hardware components (ICs, sensors, etc.), interfaces, and potential hardware trojans, e.g., unlabelled chips, and chips that are unexpected for this particular device or device class. All found PCBs, ICs, interfaces, sensors, actuators and perform an open source intelligence (OSINT) will be documented. This test possibly reveals undocumented interfaces, e.g., communication chips.
- *Memory extraction*: the firmware and the configuration from the memory or debug interfaces is extracted, and it will be checked if the firmware is encrypted. If access

to a reference firmware is given, the extracted firmware is compared against this reference firmware.

- *Identified interface tests*: for all identified interfaces, e.g., debug interfaces, it will be checked whether they are active.
- *Documented interface tests*: local interfaces, as documented, are analysed (e.g. RS232, USB). The connection is analysed in regard of handling commands and if the interface is accepting standard protocols. In addition, input validation of the interface and underlying software is applied, where best practice application security measures are verified. The customer statement that the USB interface is only to be used for input devices will be verified by connecting other device classes and by checking the functionality.
- *Malformed input testing*: fuzzing frameworks will be applied to test the robustness of the protocols and services running via local interfaces. During these tests, the device under test should continue normal operation according to its intended use. The device under test should not show unexpected behaviour such as crashes, non-responsive hangs, exceptions, connection loss, or disclosure of sensitive data.

Cryptography

These activities focus on the evaluation of the used cryptographic algorithms, protocols, and parameters across all assessed components. For the security assessment the test activities include:

- *Cryptographic primitives*: assess the used algorithms compared to best practices and recommended standards.
- *Cryptographic protocols*: assess the used protocols compared to best practices and recommended standards.
- *Keys and parameters*: assess key lengths and types, algorithm and protocol parameters compared to best practices and recommended standards.

MOBILE APPLICATION TESTING

The basis for mobile application testing is provided public international standards, recommendations, and expertise. For example, the security standard Mobile Application Security Verification Standard (MASVS)[1] and the Mobile Security Testing Guide (MSTG)[2] published by OWASP provides specific requirements for mobile applications in general. Additional tests like source code analysis or fuzzing can be conducted depending on the customized scope and security requirements analysis.

CLOUD SERVICES TESTING

The basis for backend testing is provided by public international standards, recommendations, and expertise. For example, security guidelines like OWASP's Top 10 for Web Applications[3] and similar provide requirements around relevant cloud services. Note that the scope is limited

[1] https://mobile-security.gitbook.io/masvs/
[2] https://owasp.org/www-project-mobile-security-testing-guide/
[3] https://owasp.org/www-project-top-ten/

to the device's context, i.e., only functionality which is relevant to and/or used by the device is within scope of the interview.

DISCLAIMER

SGS does not warrant that, even in the case there have been no findings during SGS's security assessments and security tests, the test object as described above has no security flaws.

The test results were found at the time of initial testing and or market surveillance and are indicative to products with the listed Version Number and model identifier. The test results are subject to change should there be any change in the manufacturing processes and bill of material used (Hardware and Software).

SGS is not a manufacturer, supplier or distributor of products and makes no warranty, representation, or guarantee regarding the suitability of the products for any particular purpose, nor does SGS assume any liability whatsoever arising out of the use of the product. Buyers shall not rely solely on any data and performance specifications or parameters provided by SGS. Information provided in this document is proprietary to SGS, and SGS reserves the right to make any changes to the information in this document at any time without notice.

HISTORY

Version	Date	Author	Changes
1.0	Nov 10, 2020	SGS Cybersecurity Services, Graz	Release