



**Methods for Testing and Specification (MTS);
Security Testing;
Security Validation Methodology for IoT components**

get full document from standards.iteh.ai

Reference

DTR/MTS-TST12SecValMet

Keywords

IoT, methodology, ML, security, testing

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Important notice

The present document can be downloaded from the
[ETSI Search & Browse Standards](#) application.

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format on [ETSI deliver](#) repository.

Users should be aware that the present document may be revised or have its status changed,
this information is available in the [Milestones listing](#).

If you find errors in the present document, please send your comments to
the relevant service listed under [Committee Support Staff](#).

If you find a security vulnerability in the present document, please report it through our
[Coordinated Vulnerability Disclosure \(CVD\)](#) program.

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.

No recommendation as to products and services or vendors is made or should be implied.

No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.

In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Copyright Notification

No part of this document may be reproduced in any form, by any means and in any media, without the prior written authorization of ETSI and except as expressly permitted below.

By way of exception and when the document is a normative deliverable (European Standard (EN), Technical Specification (TS), Group Specification (GS) or ETSI Standard (ES)), ETSI authorizes to reproduce and incorporate into products, services and technical documentation only those extracts (e.g. templates) that are strictly necessary for the technical implementation of the normative deliverable, to ensure compliance with the latter. Nothing in this notice shall be construed as limiting any mandatory exceptions to copyright provided by applicable law.

© ETSI 2026.
All rights reserved.

Contents

Intellectual Property Rights	4
Foreword.....	4
Modal verbs terminology.....	4
Executive summary	4
Introduction	5
1 Scope	7
2 References	7
2.1 Normative references	7
2.2 Informative references.....	7
3 Definition of terms, symbols and abbreviations.....	9
3.1 Terms.....	9
3.2 Symbols.....	9
3.3 Abbreviations	9
4 Security testing overview	10
4.1 General	10
4.2 Current state of SAST	11
4.2.0 General.....	11
4.2.1 Secure coding standards.....	11
4.2.2 Code review	12
4.2.3 Symbolic execution.....	12
4.2.4 Taint analysis	13
4.2.5 AI-driven static analysis	13
4.3 Current state of DAST	13
5 Specification of a security validation methodology	16
5.1 General	16
5.2 Static Application Security Testing (SAST).....	17
5.3 Dynamic Application Security Testing (DAST)	18
5.4 Interactive Application Security Testing (IAST)	19
5.4.1 General.....	19
5.4.2 Verification of SAST findings.....	19
5.5 Firmware Fuzzing.....	22
5.5.0 Firmware Fuzzing Introduction	22
5.5.1 Static Memory containing global variables	22
5.5.2 Stack and Heap sanitization	23
6 Validation of security patches	23
6.1 Introduction	23
6.2 Genetic grey-box fuzzing for security patch validation.....	23
History	25

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables (European Standard (EN), Technical Specification (TS), Group Specification (GS) or ETSI Standard (ES)) may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the [ETSI IPR online database](#).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™**, **LTE™** and **5G™** logo are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM®** and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Technical Report (TR) has been produced by ETSI Technical Committee Methods for Testing and Specification (MTS).

The aim of the present document is to outline a unified methodology for security validation of IoT components, with a particular focus on automated security testing techniques and their integration into supply-chain-oriented frameworks such as Device Security Passports (DSPs). The present document has been influenced by research and implementation results from the DOSS project ("Secure-by-Design IoT Operation with Supply Chain Control").

Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the [ETSI Drafting Rules](#) (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Executive summary

The security of Internet of Things (IoT) systems depends critically on the trustworthiness of their constituent components, including firmware, embedded software, third-party libraries (incl. open-source libraries) and in-house applications. The present document describes a security validation methodology for IoT components that supports secure-by-design development and supply-chain-aware validation.

The methodology integrates:

- **Static Application Security Testing (SAST):** Early-lifecycle analysis of source and binary code to identify security weaknesses before deployment. The methodology combines rule-based analysis, secure-coding standards and AI-enabled vulnerability prediction and classification.
- **Dynamic Application Security Testing (DAST):** Runtime testing of binaries, firmware and services without requiring source code, with emphasis on coverage-guided fuzzing and specialized embedded-firmware fuzzing.
- **Interactive Application Security Testing (IAST):** Combined use of SAST and DAST results to focus dynamic testing on the most relevant code regions and to confirm or negate static analysis findings, thereby reducing false positives.
- **Security patch validation:** Test techniques to validate that a security patch fully addresses a vulnerability and does not leave alternative exploit paths open.

The methodology is designed to integrate with supply-chain-oriented artefacts such as a **Device Security Passport (DSP)** [i.7]. Testing artefacts (findings, coverage metrics, evidence of patch validation) are exposed to the DSP so that component manufacturers, integrators and operators can:

- assess component security posture;
- prioritize remediation activities;
- support regulatory and certification processes (for example in the context of the Cyber Resilience Act); and
- maintain an auditable security history across the full component life-cycle.

Introduction

The security of IoT systems is increasingly recognized as a critical concern across industrial, consumer and critical infrastructure domains. IoT products are typically assembled from heterogeneous hardware and software components originating from multiple vendors, open-source communities and in-house development efforts. These components are updated independently, reused across product families and operated in diverse environments for extended periods of time. This supply-chain-centric reality complicates traditional, monolithic approaches to security assurance and creates a need for methodologies that are capable of addressing components both individually and in the context of the wider system in which they operate.

At the same time, the regulatory landscape is evolving significantly. The European Cyber Resilience Act (CRA) [i.1] introduces binding obligations for the security of products with digital elements, including IoT devices, across their full lifecycle. These obligations require manufacturers, importers and distributors to demonstrate that their products have been designed, developed and maintained in accordance with appropriate security practices. Structured, repeatable and traceable security testing is therefore no longer a purely technical concern but is becoming a prerequisite for regulatory compliance and market access.

Existing security testing practices are, however, fragmented. The following limitations are observed in current industrial practice:

- Static analysis is often used in software development, but it often produces incorrect results and is not linked to dynamic testing. Dynamic testing and fuzzing campaigns are often scoped to individual products or selected interfaces, executed late in the development lifecycle.
- Lack of interactive techniques that combine static and dynamic analysis.
- Patch validation is commonly performed in an ad-hoc manner, with limited automation and insufficient traceability of the evidence generated.

These limitations are particularly consequential in IoT deployments, where the attack surface is distributed across many heterogeneous devices, communication interfaces, network services and backend components, and where the exploitation of a single residual vulnerability can compromise entire fleets of products simultaneously. The growing emphasis on regulatory frameworks such as the CRA further increases the need for reliable, auditable and repeatable security testing processes.

The DOSS project ("Secure-by-Design IoT Operation with Supply Chain Control") has addressed these challenges by introducing the Supply Trust Chain (STC) concept. The STC proposes a Device Security Passport (DSP) for components, which records security-relevant information across a component's full lifecycle, and a Component Tester (CT), which generates and provides automated security testing evidence to the DSP platform. The security testing methodology developed within DOSS integrates static, dynamic and interactive application security testing as well as structured patch validation into a coherent, supply-chain-aware process.

The present document generalizes and abstracts the methodology developed in the context of the DOSS project, with the objective of making it applicable across IoT domains and independent of any specific project tooling or platform infrastructure. It provides an overview of the current state of the relevant security testing techniques, specifies a unified methodology that defines how these techniques are combined through explicit information flows and complementary capabilities, and describes how the resulting testing evidence can be structured for integration with supply-chain artefacts such as the DSP. In doing so, the present document aims to support component manufacturers, system integrators and device operators in establishing security validation processes that are technically rigorous, lifecycle-aware and aligned with emerging regulatory and certification requirements.

Sample Document

get full document from standards.iteh.ai

1 Scope

The present document specifies a methodology for automated security testing of IoT devices. It covers third-party, open-source, and in-house software and serves as a basis for later tooling and framework implementation. The methodology integrates Static Application Security Testing (SAST), including specialized vulnerability predictions, as well as Dynamic Application Security Testing (DAST), and Interactive Application Security Testing (IAST) with a particular focus on fuzzing approaches. In addition, the present document provides a detailed description of the validation process for security patches.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents may be useful in implementing an ETSI deliverable or add to the reader's understanding, but are not required for conformance to the present document.

- [i.1] [Regulation \(EU\) 2024/2847](#) of the European Parliament and of the Council of 23 October 2024 on horizontal cybersecurity requirements for products with digital elements and amending Regulations (EU) No 168/2013 and (EU) 2019/1020 and Directive (EU) 2020/1828 (Cyber Resilience Act).
- [i.2] DOSS Deliverable D3.1: "Unified component testing methodology", v1.0, 2024.
- [i.3] DOSS Deliverable D2.2: "Integrated IoT supply chain concept and architecture", 2024.
- [i.4] A. Takanen et al.: "Fuzzing for Software Security Testing and Quality Assurance", 2018.
- [i.5] R. Baldoni et al.: "A survey of symbolic execution techniques", ACM Computing Surveys, 2018.
- [i.6] M. Siavvas et al.: "Static-analysis-based approaches for secure software development", 2018.
- [i.7] ETSI TR 104 285: "Methods for Testing & Specification (MTS); Device Security Passport".
- [i.8] R. Seacord: "Secure Coding in C and C++", Addison-Wesley Professional, 2013.
- [i.9] [MISRA C:2025](#): "Guidelines for the use of the C language in critical systems", MISRA Limited, 2025.
- [i.10] [MISRA C++:2023](#): "Guidelines for the use of the C++17 language in critical systems", MISRA Limited, 2023.
- [i.11] D. Davidson et al.: "Fie on firmware: Finding vulnerabilities in embedded systems using symbolic execution", Proceedings of the 22nd USENIX Conference on Security, Berkeley, 2013.
- [i.12] K.-K. Ma et al.: "Directed symbolic execution", Static Analysis, pp. 95-111, 2011.
- [i.13] N. Corteggiani et al.: "Inception: system-wide security testing of real-world embedded systems software", 27th USENIX Security Symposium, 2018.