A Framework Addressing Challenges in Cybersecurity Testing of IoT Ecosystems and Components

Steve Taylor[1][a], Martin Jaatun[2][b], Alan McGibney[3][c], Robert Seidl[4][d], Pavlo Hrynchenko[4], Dmytro Prosvirin[5], Rosella Mancilla[6][e]

[1]University of Southampton, Highfield Campus, SO17 1BJ, UK, [2]SINTEF Digital, PO Box 4760 Torgarden, 7465 TRONDHEIM, Norway, [3]Munster Technological University, Rossa Avenue, Bishopstown, Cork, Ireland, [4]NOKIA Bell Labs, Werinherstr. 91, 81541 Munich, Germany, [5]World Research Center of Vortex Energy, Rustavi Street Building 3 Apartment 47, Zaporizhzhya 69093, Ukraine, [6]Antonov Aeronautical Scientific & Technical Company, Academika Tupoleva Str. 1, KYIV 03062, Ukraine, [7]Engineering Ingegneria Informatica spa
Piazzale dell'Agricoltura, 24 - 00144 Rome, Italy
[a]https://orcid.org/0000-0002-9937-1762, [b]https://orcid.org/0000-0001-7127-6694, [c]https://orcid.org/0000-0002-0665-2005, [d]https://orcid.org/0000-0001-8518-8433, [e]https://orcid.org/0000-0002-6566-9560
s.j.taylor@soton.ac.uk, Martin.G.Jaatun@sintef.no, Alan.McGibney@mtu.ie, , robert.seidl@nokia-bell-labs.com, olegzaritskyi@gmail.com, dmytro.prosvirin@antonov-airlines.aero, RosellaOmana.Mancilla@eng.it

Keywords: IoT, Testing (Software Engineering; Penetration; Product Development), Full IoT Lifecycle Testing, Security by Design, Component Level Testing, System Level Testing, Cyber Threat Intelligence (CTI) Sharing.

Abstract: This paper describes challenges within IoT ecosystems from the perspective of cybersecurity testing along with a proposed approach to address them that will be investigated in a recently started Horizon Europe project named TELEMETRY. The key observations regarding the design of the framework are summarised as follows. There is a need to consider the full lifecycle of IoT components – at their design time, their integration into systems, and operation of those systems. Threats and risks can propagate when components are connected together in systems - vulnerabilities in one component can affect other components in a system. IoT devices present limitations to current testing and management due to geographical distribution, opacity and limited processing power. Risk assessment fulfils an important requirement because it enables assessment of what elements are important to the system's stakeholders, how these elements may be compromised, and how the compromises may be controlled. Feedback from operational monitoring of IoT devices can inform firmware updates / patches to the devices but there is a significant challenge in rolling out these patches to multiple low-power devices geographically distributed.

1 INTRODUCTION

This paper describes challenges within IoT ecosystems from the perspective of cybersecurity testing along with a proposed approach to address them, that will be taken by a recently funded Horizon Europe project named TELEMETRY, focusing on the key principles identiefied. The challenges and approach are presented as a position for disucssion and evaluation by the community.

1.1 Background

The societal and economic benefits from the rapid advance of the digital economies are at the core of the European Digital Agenda. This is bolstered by the Next Generation Internet (NGI) initiative with an emphasis on creating a more resilient, trustworthy and sustainable Internet for our digital future. As the Internet of Things (IoT) brings connectivity and networked intelligence to the physical things around us, highly distributed and complex infrastructures are emerging ultimately forming IoT ecosystems, which can be defined as: *systems of interconnected IoT devices with hardware, software, services and* *backbone network communication infrastructure to support the required system functionality*.

While these ecosystems bring many benefits and efficiencies, their inherent complexity, heterogeneity and dynamicity and distributed nature create challenges for the management of security, testing, validation, reliability and assurance at scale. The characteristics of IoT ecosystems encapsulate some key challenges for cybersecurity testing & assurance of hardware, software and service components such as: i) IoT devices are often “black boxes” to deployers and users, meaning that their structure and inner components are not accessible for testing; ii) IoT devices are hard to update due to the specific nature of their firmware and that the devices may be manifold and geographically distributed and iii) vulnerabilities in one component may allow threats and risks to propagate to other components in a given system.

To address these challenges, there is a need for tools, techniques and holistic methodologies for cybersecurity testing and vulnerability detection at both component level and also in the systems the components are integrated into (Figure 1). This will enable continuous assessment of IoT components and ecosystems over their whole lifecycle, supporting the propagation of assurance for component developers, system integrators and operators, who act on behalf of ecosystem’s eventual users.

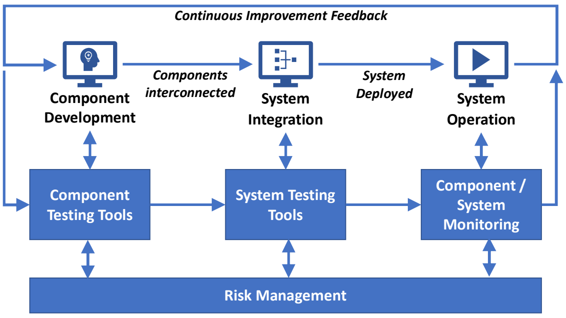

Figure 1: IoT Ecosystem Lifecycle & Testing Tools.

2 CONCEPT & APPROACH

Our approach is to develop a testing and risk management framework that consists of a suite of enabling tools and advanced methodologies that can be used to engineer and accelerate the implementation of testing, verification and assurance of IoT ecosystems (as presented in Figure 2). This framework targets the creation of a holistic decision support system for the three major stakeholder types: Component Developers (CDs), System Integrators (SIs) and System Operators (SOs), enabling a shared representation of assurance in existing and new IoT ecosystem deployments. Continuous assessment of the cybersecurity components & systems over their whole lifecycle is essential to ensure the provision of resilient digital infrastructures, systems and processes. The framework ensures a tight coupling exists between the *design phase* of components (support security by design through targeted testing tools), *components’ integration into systems* (through systemic risk analysis), *dynamic detection of vulnerabilities* and adaptation of the systems in their operational phase (to ensure continued secure operation).

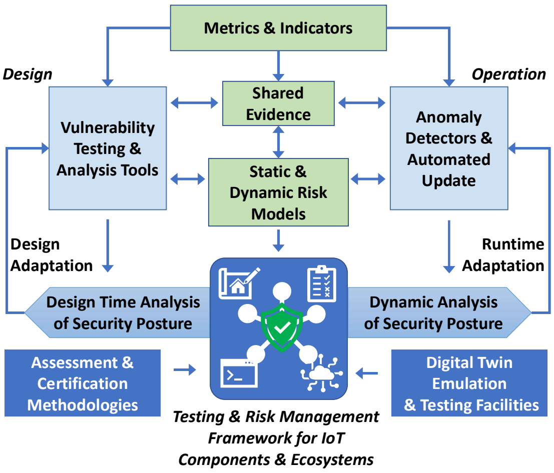
Metrics & Indicators
Design
Operation
Shared Evidence
Vulnerability Testing & Analysis Tools
Static & Dynamic Risk Models
Anomaly Detectors & Automated Update
Design Adaptation
Runtime Adaptation
Design Time Analysis of Security Posture
Dynamic Analysis of Security Posture
Assessment & Certification Methodologies
Digital Twin Emulation & Testing Facilities
Testing & Risk Management Framework for IoT Components & Ecosystems

Figure 2: Testing & Risk Management Framework

The framework explicitly supports feedback from operation to design. (McGraw, 2006) identifies "Security Operations" as one of seven "touchpoints" that constitute good practice for software security, offering "feedback from the field" where monitoring is used to identify security bugs and flaws in running systems, feedback allows for these to be fixed in the next development cycle. In a DevOps / Continuous Integration environment, development is continuous, and identified security bugs and design flaws can be rectified "without undue delay". The framework explicitly supports this feedback loop and brings DevSecOps and DevOps closer together via its monitoring tools and techniques that can feed into design-time tools.

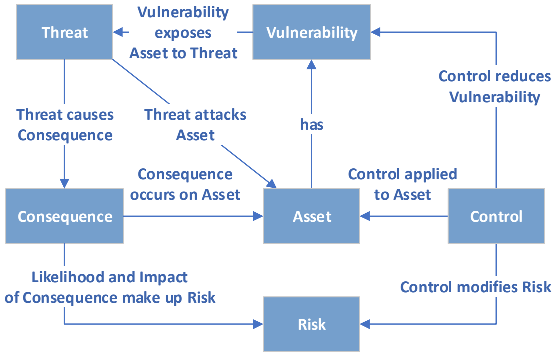
Threat
Vulnerability exposes Asset to Threat
Vulnerability
Control reduces Vulnerability
Threat causes Consequence
Threat attacks Asset
has
Consequence occurs on Asset
Control applied to Asset
Consequence
Asset
Control
Likelihood and Impact of Consequence make up Risk
Control modifies Risk
Risk

Figure 3: Risk Assessment Schema.

Risk assessment is a key element of our approach. The schema shown in Figure 3 (derived from ISO 27000[1]) defines the relationships between key elements of risk management. Assets (here software / hardware components or systems) have vulnerabilities, which expose them to threats. Threats cause Consequences, which are usually adverse, and the likelihood of a successful Threat attack on an Asset determines the likelihood of its Consequence. The impact (severity) of a Consequence combined with its likelihood leads to a specific Risk. Controls modify Risks, typically by acting to reduce the impact of the Consequence (mitigation) or to reduce vulnerabilities in Assets, which in turn reduces their exposure to Threats. Risk analysis at component and system level is a key integrating concept, which enables transparent assessment of the effects of vulnerabilities in components or systems.

Many of the framework's tools and techniques are aimed at detecting or testing for vulnerabilities in components and systems, and their output can be either used for benefit independently or can be used as input to the risk assessment tooling, where the detected vulnerabilities bootstrap the risk assessment cycle illustrated above and lead to risks. The risk assessment component uses an existing tool developed by one of the project partners that provides insight into risks to a vulnerable component, but also propagated threats and resulting risks to other components that are connected to it.

Advanced machine learning (ML) approaches will be used to proactively analyse the behaviour of components and systems with the detection of abnormal/anomalous activities that can provide insight into whether the components/systems are working as expected. The tools aim to support two types of monitoring, detection and testing: "black box", addressed via component and system level anomaly detection tools and techniques; and "white box", via IoT Software Bill of Materials[2], IIoT SBOM generation[3] and open-source software SBOMs that describe the sub-components that make up the software. Both testing types can inform component-level risk analysis.

The tools and framework approach is deliberately designed to be extensible so that its tools and infrastructure can integrate easily with third party tools (e.g. vulnerability scanners such as Wazuh[4]) to provide greater coverage of the space of devices / software / hardware components and systems and to provide a holistic risk management approach that offers evidence-based decision support to assess and enhance assurance across complex IoT ecosystems in order to reduce the likelihood and impacts of cybersecurity threats and to simplify compliance with internal, industrial and governmental regulations and certification standards such as the EU Cybersecurity Certification framework (EUCC).

The approach aims to determine methodologies for tools' use, individually, together, and alongside external third-party tools to support the complete lifecycle of components & systems and for compliance with relevant certification standards. For this, two aspects of the operational context for IoT ecosystems need consideration. Firstly there are application cases, where different organisations interact to add value (e.g. supply chains). The project has three use cases in aerospace, telecommunications and manufacturing, each of which represents a complex, collaborative ecosystem (including IoT supply chains) - they are all multi-stakeholder systems of interacting components with different domains of control. Secondly, there is a need to consider the representation of an IoT device in the context of a software supply chain, where an IoT device / software component has upstream dependencies and libraries. SBOMs will be used to determine upstream dependencies (e.g. third party libraries) used in the component for vulnerability and risk analysis at sub-component level.

[1] ISO/IEC 27000:2018. Information technology — Security techniques — Information security management systems — Overview and vocabulary. https://www.iso.org/standard/73906.html

[2] National Telecommunications and Information Administration, Software Bill of Materials https://ntia.gov/SBOM

[3] https://www.iiotsbom.com/

[4] Wazuh - The Open Source Security Platform https://wazuh.com/

3 SPECIFIC CHALLENGES & APPROACHES

This section describes key challenges identified for the specific case of IoT ecosystems, and for each challenge, a proposed approach is described for addressing the challenge.

3.1 Tools & techniques for semi-automated risk management of devices, components & systems

There are several tools and techniques for cybersecurity risk assessment: software tools such as securiCAD, ThreatModeler, OWASP Threat Dragon, Microsoft Threat Modeling Tool and CrypTier; and manual risk analysis is supported via methodologies such as OCTAVE, STRIDE, PASTA and Trike. These tools and techniques are focused on the enterprise and do not account for propagated threats between components and stakeholders, often found in supply chains (Boyes, 2015). Further, components (e.g. hardware, devices, software, data) are systems in their own right that aggregate sub-components - for example a software component can be built of bespoke code plus third party libraries. Components often appear as black boxes to system integrators and their threat and risk potential is unknown to the rest of the higher-level system. There is thus a need to consider both systemic and individual component-level perspectives when assessing the risks in scenarios where software and hardware is integrated into a higher-level system.

Our approach is to utilise and extend a cybersecurity-focused risk assessment toolkit named Spyderisk that has been in development for 9+ years and is available open source[5]. This follows the ISO 27005[6] methodology for cyber security risk management, which in turn supports ISO 27001[7] certification. It has been applied to trust in communication network situations targeting healthcare (Surridge et al. 2019), data privacy protection (Boniface et al 2022) and GDPR compliance (Taylor 2021). Via a knowledge base and reasoner, this toolkit supports modelling and automated risk assessment of socio-technical systems, where it automatically determines risk levels of a system-level model and suggests controls to lower risks. The toolkit's knowledge base will be extended to enable assessment at component-level, whereby a component can be modelled as a (sub)system in its own right, via supporting types of assets representing, e.g., third party libraries, operating systems, firmware; their relationships within the component; how vulnerabilities of these constituent parts can affect the component as a whole; threats that may affect a component; the risks that arise from threats; and controls that can address the risks. Threat propagation between a component and the higher level system in which it is integrated will also be considered, following the system-level focus of e.g. the IEC 62443 series of standards[8]. An initial runtime risk analysis at system level will be developed to support dynamic component-level risk assessment. The risk assessment will be supported by integrating vulnerabilities from advisory databases such as Mitre CVE[9] driven by identification of third party libraries in the SBOM manifest for open source software components, and alerts from vulnerability scanning tools such as Wazuh.

3.2 Component Level Runtime Monitoring and Vulnerability Detection

Numerous cybersecurity vulnerability scanners already exist, operating at hardware, infrastructure and application level, as either commercial products (e.g. Nessus or Tripwire IP360) or open source software (e.g OpenVas, Nmap or Wazuh). These are comprehensive and powerful, but there is a need to overcome two significant challenges. 1) many devices and components (especially IoT) are a "black box" to the deployers and operators of systems they are integrated into, hence there is a need to be able to monitor these components and detect vulnerabilities from their inputs and outputs alone. 2) events

[5] https://github.com/SPYDERISK

[6] ISO/IEC 27005:2022. Information security, cybersecurity and privacy protection. Guidance on managing information security risks https://www.iso.org/standard/80585.html.

[7] ISO/IEC 27001:2022. Information security, cybersecurity and privacy protection. Information security management systems Requirements. https://www.iso.org/standard/27001.

[8] https://www.isa.org/standards-and-publications/isa-standards/isa-iec-62443-series-of-standards

[9] Common Vulnerabilities and Exposures Database https://cve.mitre.org/cve/, currently migrating to https://www.cve.org/

generated from devices often result in a deluge of data - i.e. the devices repeatedly output similar, or only slight deviations from, previous measurements which can hinder the scalability of solutions and mask detections of genuine anomalies.

Our approach to address 1) is by monitoring the output of the devices over time and creating ML models to detect vulnerabilities or insights about the behaviour of the device compared to normal operation. Our approach to address 2) is via aggregation of events via time-series analysis, investigating different approaches for aggregation such as: thresholds for differentiation between normal and suspicious working conditions; silence periods for repetitions of the same events; and how to map multiple source events to a specific aggregated event (e.g. one reporting that a threshold is breached) which can be used to prioritize the order of these events.

3.3 System Level Runtime Monitoring and Vulnerability Detection

Runtime monitoring of components to detect anomalies and misbehaviours is needed with respect to the expected interaction baseline in systems of devices, software and hardware components. The analysis of these anomalies can provide insight valuable for assessment of components security (e.g., discovery of vulnerabilities) and thus to inform updates to components. The state of the art concerns Intrusion Detection / Prevention Systems (IDS / IPS), that monitors network traffic for suspicious activity and alerts when such activity is discovered, or additionally take preventative actions. IDS/IPSs are classified based on two detection techniques. Signature-based (Salunkhe, 2017) approaches monitor packets in the network and compare them against a database of attack signatures or attributes of known malicious threats. They have a low false-positive rate, but the need for signatures allows all unknown attacks to go undetected. Behaviour-based (Vengatesan, 2019) approaches identify anomalous behaviour using machine learning techniques to recognize a normalized baseline, to which all network activity is compared. Behaviour-based approaches are more susceptible to false positives and require an effective operating algorithm to correctly analyse the arrived packets. Some important limitations therefore persist with IDS: detection accuracy is (relatively) poor, the rate of false positives is still high, they have limited scalability, current techniques often fail to detect emerging attacks and they have very limited diagnostic facilities.

Our approach to address these challenges is to investigate an IDS that performs anomaly detection on IoT ecosystems. We will study the application of Federated Learning approach on behaviour-based detection aiming at reducing the number of false positives, increasing the detection accuracy and detection coverage even with limited resources (IoT devices).

.

3.3 Misuse Detection

Devices / software / systems are designed to perform a purpose with a specific usage in mind, and they are deployed in socio-technical systems with human users. These human users may either be unaware of acceptable operating conditions or may deliberately aim to misuse components with malicious intentions, so there is a clear need for detection of misuse of components in systems. Further, dynamic testing should not only cover illicit access to components but highlight component vulnerabilities due to the misuse, thus supporting continuous improvement of the components.

Our approach is to investigate the use of machine learning to detect the misuse of software components & systems based on baseline behavioural patterns identified in historic usage scenarios. We will investigate several approaches for the learning of user-interaction models and the detection of divergences in user behaviour from the norm, using similar principles to social engineering for capturing user aspects such as user functional footprint, temporal behaviour and statistical data distribution. These anomalies raise warnings that can identify aspects such as impersonation of an authorised user by an attacker, insider attacks or inadvertent misuse. A set of algorithms such as Gaussian, Bayesian, time series, autoencoders, deep neural networks, and other neural networks will be used for anomaly detection. The Misuse Detection Toolkit will also provide a synthetic data generator capable to simulate the misuse of software in web-based systems. For this purpose, an analysis of the misuse case-based interactions will be done.

3.4 Trusted Mechanisms to Facilitate Distributed Sharing of Testing, Verification & Cyber Threat Intelligence (CTI)

Data exchange and sharing among independent stakeholders, tools and services remains a real concern. Much valuable information available for component testing and evaluation (metrics, inputs and results) and risk assessment remain closed and siloed due to the complexity, interoperability, cost, privacy & security risks, fears over data sovereignty and lack of incentive for sharing. Thus there is a need to create a collaborative framework that allows Component Developers, System Integrators, System Operators (CDs, SIs, SOs), researchers, practitioners, industry and infrastructure providers, etc. to mutually benefit from sharing vulnerability related information, which will reduce redundancy and foster collaboration.

Our approach to address this is to create a trustworthy framework for sharing security-related information between tools and entities involved in dynamic cybersecurity testing based on Distributed Ledger Technology (DLT) providing functionality to automate secure data management services, policy enforcement and governance as well as the provision of a secure backbone for data flows across independent entities (Zhou, 2018). DLT underpinned by blockchain technology increases transparency across systems (contributing to assurances of provenance) (Montecchi, 2019); and supports auditability via features that allow for interrogation each stage of the lifecycle for compliance with policy, standards, or regulations (Lopez, 2020). Our approach will include secure bi-directional communication with external information sources, e.g., Cyber Threat Intelligence (CTI) repositories. The use of smart contracts will provide mechanisms to automate governance, auditing and assurance processes, to incentivise data sharing by lowering barriers for secure data exchange and enabling data owners to retain control & sovereignty over data. We will investigate the types of information that are needed for robust auditing, how they may be efficiently stored in Distributed Ledgers and how the information may be queried and interpreted by non-domain-experts. This will form a reference ontology for data sharing based current approaches e.g. Cyber Intelligence Ontology[10]. We will further investigate architectural approaches to integrate testing methodologies and enhance existing tools with DLT-based information sharing.

3.5 Service-Level Cybersecurity Testing

IoT applications leverage application programming interfaces (APIs) as a mechanism to interconnect and exchange often outside organizational boundaries. To ensure reliability and robustness of interactions and activities that occur within the system requires API analysis and security testing. Tools exist to support testing however this is typically done at design time and not on a continuous basis nor consider dynamic composition of applications over time.

Our approach to address this challenge is to develop and evaluate a security analyser for APIs with particular emphasis on identifying potential vulnerabilities such as cross-site scripting (XSS), SQL injection, cross-site request forgery (CSRF), compromise attacks, injections, and man-in-the-middle attacks (Badhwar, 2021). The security analyser aims to execute a series of tests to equate and quantify the security posture of API services running at the edge and/or cloud layers that will inform behaviour and reputation models of participants in IoT ecosystems (devices & services). The models can in turn be used to monitor performance of dynamic IoT applications and build a digital representation of the security posture of the application over time. This can then be utilised to adapt the operation of a particular component (e.g. via policy enforcement) to ensure protection across a larger ecosystem. REST API Penetration testing solutions such as Astra[11] can be investigated and extended to include additional security metrics (e.g. data anomalies, encryption mechanisms).

3.6 Testing Access Control to Components & Systems

A common problem with complex access control systems is that they provide a mechanism for the user to obtain permission from multiple policies, resulting in an accumulation of effective permissions and cumulatively constituting a certain level of risk. Determining whether an access level poses a potential security risk is a non-trivial task. This depends on the sensitivity of the restricted object and the reliability of the user, which together can give a certain level of risk (Parkinson, 2022) and assessment of these factors will improve decision-making when managing systems and access to them.

To create a flexible assessment of access control to system components indicators, which include both quantitative and qualitative indicators, the use of a mathematical apparatus of fuzzy logic is proposed. This approach allows us to model complex access control vulnerability criteria, including inaccurate, incomplete or vague data, and the result will be an assessment of the risk of granting more permissions

[10] Cyber Intelligence Ontology https://unifiedcyberontology.org/index.html

[11] [Astra] https://github.com/flipkart-incubator/Astra

to a user or vice versa. Previous work using fuzzy logic in discovering irregularities in access control systems implemented as a binary classification system will be extended to model issues of access control via determination of ontologies of access control concepts, their fuzzy logic representation and risk computation.

3.7 IoT Device SBOM Generation

A Software Bill of Materials (SBOM) is a standardized list of dependencies (source components, modules and libraries) used in a software component, to ensure transparency and to check the dependencies against vulnerability databases, for example by using a service like Snyk or a tool like OWASP's dependency-check. There are currently (at least) three competing SBOM formats; SPDX[12] from the Linux Foundation, CycloneDX[13] from OWASP, and SWID as defined in ISO/IEC 19770-2. NTIA has created a taxonomy for SBOM tools[14], but there is a lack of such tools for IoT applications, hampering IoT device developers in their ability to create SBOMs for their devices.

We will examine the different SBOM options and select the one that is best fit-for-purpose for IoT solutions, and develop and/or extend tools for automatically creating an SBOM according to the Produce-Build and Produce-Analyze paradigms to enable IoT device developers to create and update SBOMs for their devices quickly and easily.

3.8 IoT Device Fuzzing

Fuzzing is a technique that aims to find vulnerabilities and bugs in a program via generating numerous test-case inputs to the targeted program. Applying fuzzing technique on IoT device software applications or OS has received significant attention. However, this is not the case when it comes to the proprietary wireless communication stack such as 2G/3G/4G/5G (cellular) running on IoT devices. The cellular communication software running inside our mobile phone is different than in IoT devices, in terms of protocol and services perspective. In addition, security aspects such as confidentiality, integrity and availability are different in IoT devices (as compared to smartphones) due to low-latency and power-consumption requirements. Further, vulnerabilities in these cellular proprietary software stack enables compromise of mobile phones (Weinmann, 2012) and even modern cars (Bazhaniuk, 2017).

We aim to fill this gap by systematically conducting a dynamic security analysis of cellular communication stack of IoT devices in a standardized 4G/5G network in a semi-automated manner. In particular, we characterize IoT device specific security properties from 3GPP standards and build a database for generating test-cases and classify problematic behaviours for automated fuzzing frameworks.

Note that fuzz testing can be employed both at the network level and at the software component level, and in the Telemetry project we will also employ the former.

3.9 Tools & Techniques for Secure Update of Components & Systems

Secure update management is extremely challenging in resource constrained devices (such as IoT devices) that are geographically distributed and in different domains of control. The secure remote distribution of firmware updates requires secure communication channels, primarily Transport Layer Security (TLS), which has key agreement protocols built in. The protocol remains too heavy for some resource-constrained devices, and only works for TCP/IP, which is not always the case for IoT networks (e.g. LPWAN networks). Although lightweight key agreement schemes have been proposed in the literature, e.g. Noise Framework (Perrin, 2018), Julia Key Agreement (JKA) (Lundberg, 2021), there are still some open problems. Most schemes attempt to reduce the number of scalar multiplications and do not consider other constraints, such as limited payload size. The design of a supporting Public Key Infrastructure (PKI), particularly for highly distributed and decentralized systems, also needs to be addressed. Managing and evaluating access-control and authentication requests to grant the installation of firmware updates is an integral part of a secure firmware update solution. Previously, access-controls were levied on a one-to-one basis as the number of devices were relatively low, but this does not scale. Even though a lot of device access control solutions have been proposed, e.g. FlowFence (Earlence, 2016), (Bastys, 2018), ContexIoT (Yunhan, 2017), SmartAuth (Tian, 2017), IoTGuard (Celik, 2019), techniques do not match well with the setting of a highly distributed and decentralized

[12] https://spdx.dev/

[13] https://cyclonedx.org/

[14] SBOM Tool Classification Taxonomy https://ntia.gov/files/ntia/publications/ntia_sbom_tooling_taxonomy-2021mar30.pdf

system where multiple stakeholders have a shared responsibility of managing an IoT device (and hence each requiring a specific access to the device, which can change over time).

Our approach to address this is based on a lightweight key management solution, based on PKI, for distributed and decentralized systems. To authenticate the distribution of software updates, a novel lightweight cryptographic MAC algorithm for data authentication will be designed. Further optimizations will be achieved on protocol level, by exploring optimization tactics to further reduce the energy cost of data authentication. The device access control solution will be either based on cryptographic tokens (in case of symmetric-key cryptography being deployed, for example for energy-constrained devices), or on digital signatures.

3.10 Emulation Environment for Security Testing

Low-cost consumer edge devices have historically failed to implement common security mechanisms such as virtual memory, cryptographically secure pseudo-random number generators or basic exploit mitigations (such as ASLR or stack canaries) (Wetzels, 2017), (Fasano, 2021). The world of embedded systems needs the ability to conduct dynamic analysis in addition to static, and thus requires a way to move the software from a physical system to a virtual one, which models the hardware behaviour well enough - known as re-hosting. There are several approaches: pure emulation, hardware-in-the-loop, symbolic abstractions, or hybrid approaches; each with each their pros and cons (Fasano, 2021). Gustafson et al. point out the need for an automatic process to create models of peripherals that allow for execution of the firmware in a fully emulated environment, as done by their proof-of-concept tool, PRETENDER (Gustavson, 2019). Several tools have been published, tackling the same problem, including HALucinator (Clements, 2020) and Fuzzware (Scharnowski, 2022), but most of the latest tools for re-hosting are still proofs-of-concept supporting only a subset of the architecture (typically Cortex-M, for Fuzzware for instance).

We will build on top of those proofs-of-concept and collaborate with the industry partners in the project to 1) study the applicability of the tools to the use cases in the project; 2) further develop them if required, based on the needs from the partners; 3) study the concept of digital twins for embedded security, and how it can be used for security testing as part of the development process. In particular, we will make use of the automatically generated peripherals models from Fuzzware to develop the digital twins. Guidance on how to create and integrate such virtualized environments and digital twins for security testing will be provided.

4 CONCLUSIONS

This paper has described challenges for cybersecurity identified in the area of IoT ecosystems. Multiple challenges have been described, along with a proposed solution. These solutions will be investigated in a recently funded Horizon Europe research project, and results of these investigations will be reported in subsequent papers. As such the paper represents problem statement and hypotheses for solutions.

At a high level, key observations made during this work are summarised as follows.

- There is a need to consider the full lifecycle of IoT components – in their design, their integration into systems, and operation of those systems. Therefore, component level testing and system level testing is needed, especially considering that one component may be used in different systems with different effects.
- Threats and risks can propagate when components are connected together in systems - vulnerabilities in one component can affect other components in a system.
- IoT devices provide limitations to current testing and management due to geographical distribution, opacity and limited processing power.
- Monitoring and detection tools for IoT components' output can be used in a cybersecurity risk assessment for the component or the system. Risk assessment fulfils an important requirement because it enables assessment of what elements are important to the system's stakeholders, how these elements may be compromised, and how the compromises may be controlled.
- Feedback from operational monitoring of IoT devices can inform firmware updates / patches to the devices but there is a significant challenge in rolling out these patches to multiple low-power devices geographically distributed.

ACKNOWLEDGEMENTS

This work is within the Horizon Europe TELEMETRY project, supported by EC funding under grant number 101119747.

REFERENCES

Badhwar, R.: Intro to API security - issues and some solutions! The CISO’s Next Frontier pp. 239–244 (2021). https://doi.org/10.1007/978-3-030-75354-2_29

Iulia Bastys, Musard Balliu, and Andrei Sabelfeld. If this then what? controlling flows in IoT apps. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18), page 1102-1119, 2018.

Boniface, Michael, Carmichael, Laura, Hall, Wendy, McMahon, James, Pickering, J. Brian, Surridge, Mike, Taylor, Steve, Atmaca, Ugur Ilker, Epiphaniou, Gregory, Maple, Carsten, Murakonda, Sasi, & Weller, Suzanne. (2022). DARE UK PRiAM Project D3 Report: Privacy Risk Framework Application Guide (1.1). Zenodo. https://doi.org/10.5281/zenodo.7107466

Boyes, H., 2015. Cybersecurity and cyber-resilient supply chains. Technology Innovation Management Review, 5(4), p.28.

Z. Berkay Celik, Gang Tan, and Patrick McDaniel. IoTGuard: Dynamic Enforcement of Security and Safety Policy in Commodity IoT. In Proceedings of the 23rd Network and Distributed System Security Symposium (NDSS '19), San Diego, CA.

A. Clements et al., “HALucinator: Firmware Re-hosting Through Abstraction Layer Emulation,” presented at the 29th USENIX Security Symposium (USENIX Security 20), 2020, pp. 1201–1218. https://www.usenix.org/conference/usenixsecurity20/presentation/clements

Bazhaniuk, Oleksandr , Jesse Michael & Mickey Shkatov. Driving down the rabbit hole. Def-Con 2017. https://infocondb.org/con/def-con/def-con-25/driving-down-the-rabbit-hole.

Earlence Fernandes, Justin Paupore, Amir Rahmati, Daniel Simionato, Mauro Conti, and Atul Prakash. Flowfence: Practical data protection for emerging iot application frameworks. In 25th USENIX Security Symposium (USENIX Security '16), pages 531-548, Austin, TX.

A. Fasano et al., “SoK: Enabling Security Analyses of Embedded Systems via Rehosting,” in Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security, New York, NY, USA, Jun. 2021, pp. 687–701. doi: 10.1145/3433210.3453093.

E. Gustafson et al., “Toward the Analysis of Embedded Firmware through Automated Re-hosting,” presented at the 22nd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2019), 2019, pp. 135–150. https://www.usenix.org/conference/raid2019/presentation/gustafson

C. Todd Lopez, (2020). DoD Adopts 5 Principles of Artificial Intelligence Ethics, Department of Defense (Feb. 5, 2020), Available Online: https://www.defense.gov/Explore/News/Article/Article/2094085/dod-adopts-5-principles-of-artificial-intelligence-ethics/

Frans Lundberg, Juraj Feljan. Julia: fast and secure key agreement for IoT devices. In Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WISEC '21), pages 90-99, 2021.

G. McGraw, "Software Security", Addison-Wesley 2006

Montecchi, M., Plangger, K., and Etter, M. (2019). It’s real, trust me! Establishing supply chain provenance using blockchain. Business Horizons. 10.1016/j.bushor.2019.01.008.

Parkinson, S., Khana, S. Identifying high-risk over-entitlement in access control policies using fuzzy logic. *Cybersecurity* **5**, 6 (2022). https://doi.org/10.1186/s42400-022-00112-1

Trevor Perrin. The Noise Protocol Framework, Revision 34, 2018. https://noiseprotocol.org/noise.html

Salunkhe, U.R., Mali, S.N.: Security enrichment in intrusion detection system using classifier ensemble. J. Electr. Comput. Eng. (2017). https ://doi.org/10.1155/2017/17948 49

T. Scharnowski et al., “Fuzzware: Using Precise {MMIO} Modeling for Effective Firmware Fuzzing,” 31st USENIX Security Symposium (USENIX Security 22), 2022, pp. 1239–1256. https://www.usenix.org/conference/usenixsecurity22/presentation/scharnowski

Surridge M. et al. (2019) Modelling Compliance Threats and Security Analysis of Cross Border Health Data Exchange. In: Attiogbé C., Ferrarotti F., Maabout S. (eds) New Trends in Model and Data Engineering. MEDI 2019. Communications in Computer and Information Science, vol 1085. Springer, Cham.

Taylor, S., Surridge, M., and Pickering, B., "Regulatory Compliance Modelling Using Risk Management Techniques," 2021 IEEE World AI IoT Congress (AIIoT), 2021, pp. 0474-0481, doi: 10.1109/AIIoT52608.2021.9454188.

Yuan Tian, Nan Zhang, Yueh-Hsun Lin, XiaoFeng Wang, Blase Ur, Xianzheng Guo, and Patrick Tague. Smartauth: User-centered authorization for the internet of things. In 26th USENIX Security Symposium (USENIX Security '17), pages 361-378.

Vengatesan, K., Kumar, A., Naik, R., Verma, D.K.: Anomaly based novel intrusion detection system for network traffic reduction. In: 2nd International Conference on I-SMAC. IoT in Social, Mobile, Analytics and Cloud (2019)

Ralf-Philipp Weinmann. 2012. Baseband attacks: remote exploitation of memory corruptions in cellular protocol stacks. In Proceedings of the 6th USENIX conference on Offensive Technologies (WOOT'12). USENIX Association, USA.

Jos Wetzels: The RTOS Exploit Mitigation Blues, HardWear.io 2017 https://hardwear.io/document/rtos-exploit-mitigation-blues-hardwear-io.pdf

Yunhan Jack Jia, Qi Alfred Chen, Shiqi Wang, Amir Rahmati, Earlence Fernandes, Z. Morley Mao, and Atul Prakash. ContexIoT: Towards Providing Contextual Integrity to Appified IoT Platforms. In Proceedings of the 21st Network and Distributed System Security Symposium (NDSS'17), San Diego, CA.

L. Zhou, L. Wang, Y. Sun, P. Lv, BeeKeeper: A blockchain-based IoT system with secure storage and homomorphic computation, IEEE Access (2018).