Transparent and sustainable Internet of Things security

Dr. Rauli Kaksonen
Test of Things

Me, Rauli

Test of Things 2024-
- Co-founder, CTO

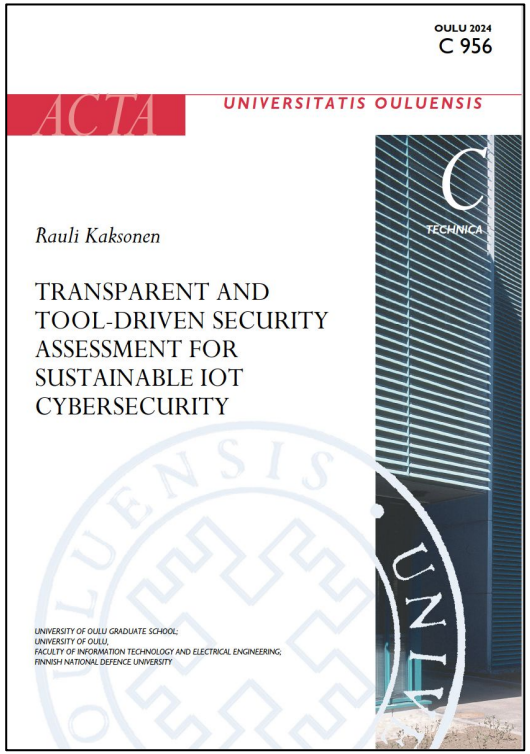
Oulu University 2019-2024
- DSc. 2024

Synopsys 2015-2017
- R&D Group Director

Codenomicon 2001-2015
- Co-founder, CTO, etc.
- The Defensics product family

VTT 1994 - 2001
- PROTOS project
- MSc. 1997, Lic. Tech. 2001

rauli.kaksonen@testofthings.com



https://urn.fi/URN:NBN:fi:oulu-202406264941

Transparent and sustainable Internet of Things security

Contents:

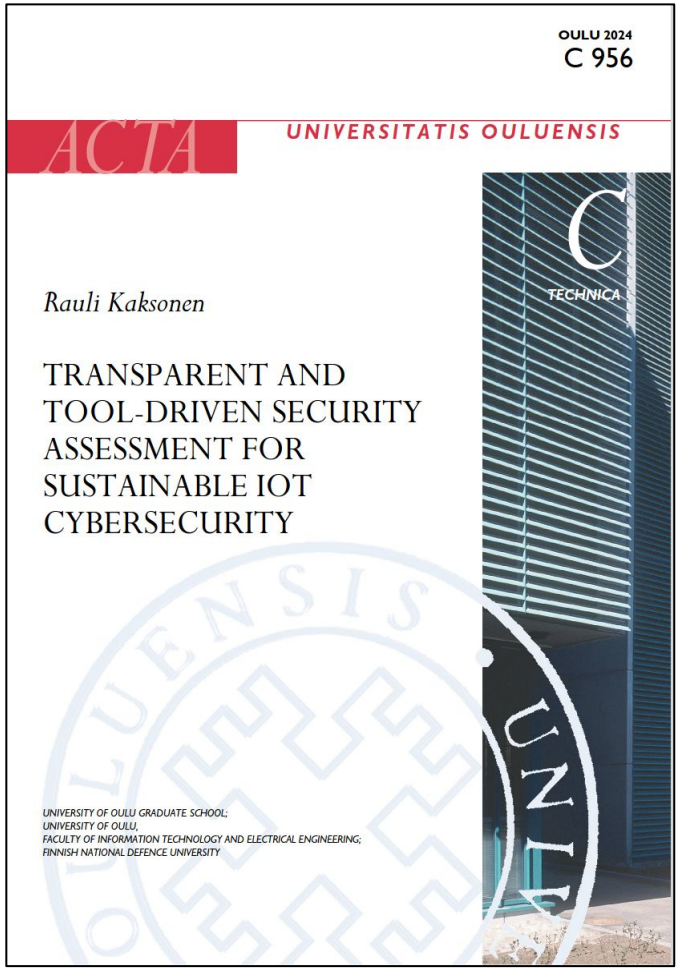
1. Background and motivation
2. Information asymmetry in IoT supply chains
3. IoT security assessment and standards
4. Automation and sustainable security

—

5. Workshop preview: *Transparent and Sustainable IoT Product Security by Toolsaf Framework*

Doctoral thesis

https://urn.fi/URN:NBN:fi:oulu-202406264941



ToT test of things

Background and motivation

Economy and society runs on IoT

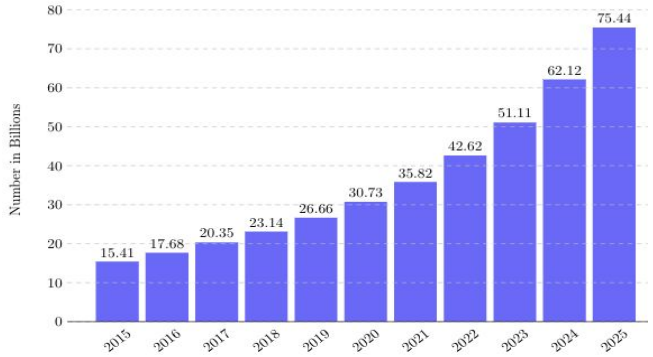
Size of IoT economy [copied from my thesis]:

- The number of devices will grow by 16% in 2023 to 16.7 billion
- IoT enterprise spending was $201 billion in 2022.
- EU briefing puts the number of IoT devices to 14.6 billion by 2022, which should double by 2030
- By 2023, the number of IP-connected devices should be two times the global population.

We are dependent on IoT for enterprises and society to work

Next: Some cases of IoT security incidents

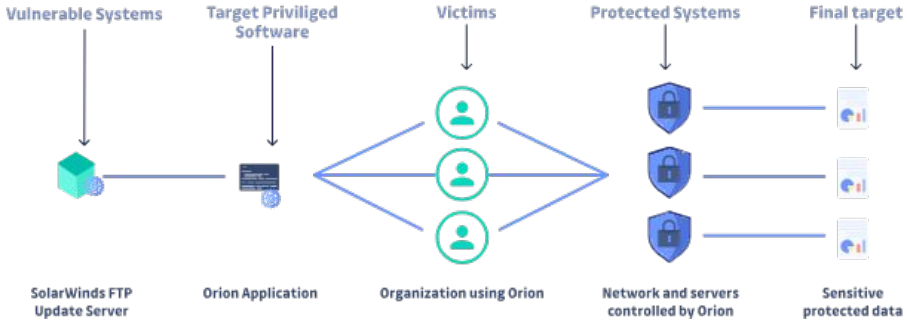
Kuva 1. Internet of Things. Lähde: Huffington Post



Case: SolarWinds supply chain attack 2020

First phase

- Attacker compromised SolarWinds network
  - SolarWinds is maker of Orion network management software
- Inserted malicious component to Orion product
  - Place the compromised SW to update system



Second phase

- Orion updates are automatically downloaded and installed
  - Malware infects the systems where new version of Orion is installed
- Malware opened a backdoor to inspect and compromise victim networks
- Estimated 18 000 organizations installed the compromised software
  - US government offices
  - DHS, Microsoft, Cisco, FireEye
  - Multiple Fortune 500 enterprises

Case: The Target data breach 2014

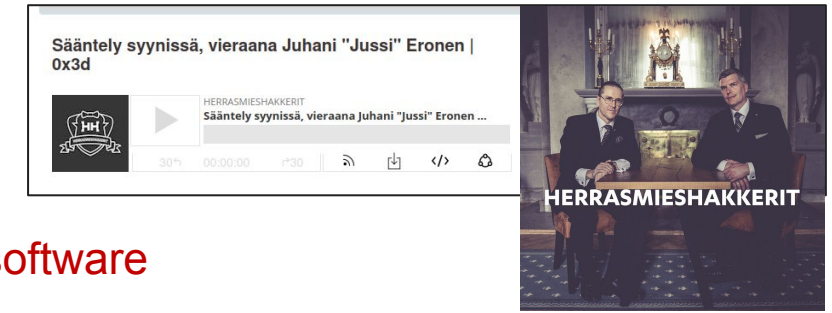
- Initial access:
  - Attacker compromised account in HVAC vendor Fazio Mechanical Services
  - The vendor had access to Target's systems
    - Attacker used this route to enter Target's network
- Lateral movement inside Target's systems
  - Deployed malware to point-of-sale devices
  - Captured sensitive data, such as personal and payment information
- Motivation:
  - Selling the acquired data in dark markets
- Damage:
  - Reputation loss
  - Target agreed to pay $18.5 million in law settlements

Cases: SNMP vulnerability 2001 - CVE-2002-0013

Simple Network Management Protocol (SNMP) vulnerability

- Multiple vendor SNMPv1 Trap handling vulnerabilities
  - May allow unauthorized privileged access, denial-of-service conditions, or unstable behavior
- Injection vector are SNMP ports (UDP port 162)
  - Anyone who can send UDP over network, could compromise the devices using vulnerable SNMP software
  - Discovery method
    - Model-based protocol fuzzing
- Root cause
  - Oversight on implementing ASN.1/BER decoding
  - ASN.1 complexity
- Many SNMP components, used in different software
  - Total of 284 vendors provided a statement!
  - An IoT supply chain vulnerability
    - Back then nobody used terms “IoT supply chain” or “IoT” :D

Full disclosure: We found these vulnerabilities in PROTOS project (OUSPG and VTT)

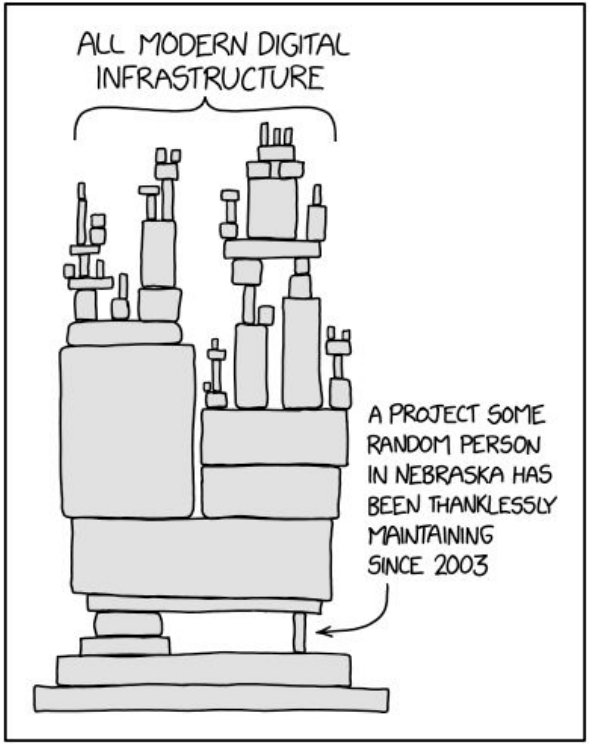


Lisää tästä casesta myös uusimassa HH jaksossa: https://hakkerit.libsyn.com/

“Vertical” supply chain security

- IoT systems are assembled from components
  - Software and hardware components
  - Created by commercial or open-source developers, themselves using sub-components
  - These are “upstream” components
- Vulnerabilities mostly come through supply chain dependencies

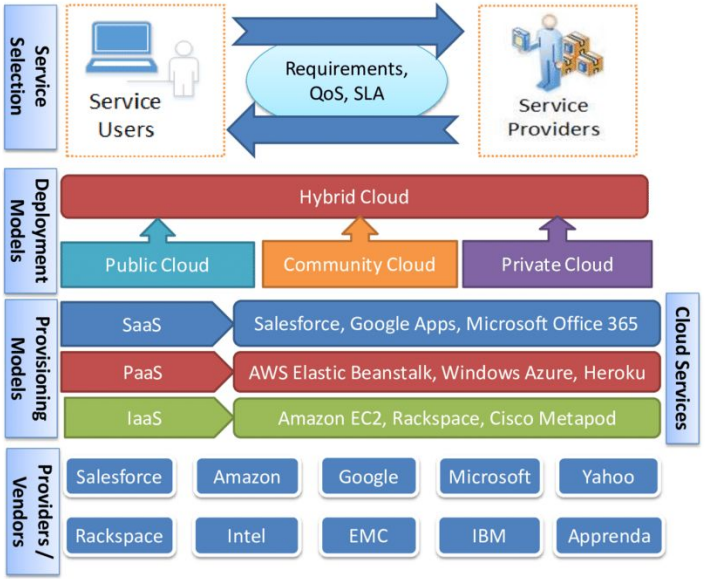
A modern enterprise network is filled with IoT devices



https://xkcd.com/2347

“Horizontal” supply chain security

- Companies are also dependent on other companies, also referred as “supply chain”
  - Vendors and subcontractors which maintain sub-systems within the company
  - APIs and services provided by other companies
  - Cloud provides which provide servers, e.g. AWS, Google Cloud, Microsoft Azure
  - Also: Electricity company, Telecom provider, … !
- A vulnerability in other company may:
  - Compromise services you are dependent on
  - Allow the attacker to move into your company network



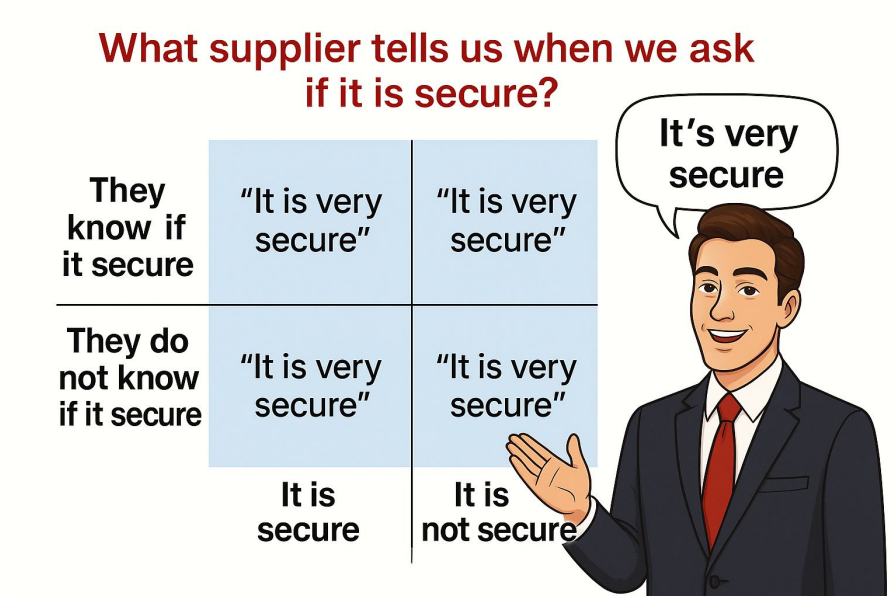
Problem with supply chain security management

- The security posture of a system or company includes the supply chain
  - Threat and risk assessment must include supply chains
- However, there is often lack of information about the security of the supply chain
  - Vertical supply chain
    - We do not know if the used components are secure, and/or
    - We do not know which components we are using 😕 !
  - Horizontal supply chain
    - We only know the interface towards them - inside is unknown

Information asymmetry in IoT supply chain

Information asymmetry

- There is an information asymmetry between us and our suppliers
  - We want to only use secure (enough) components and service
  - The supplier knows the security posture, or they do not know it
  - We do not known their security posture, or if they know it even themselves
- See possible supplier answers =>



Fixing the supply chain information asymmetry

- Move security evidence along the supply chain
  - Security-related data of the components or products
  - E.g. Software bill of materials (SBOM)
- Verify security posture independently
  - Use security testing and assessment to probe the supply chain components, services and systems
  - E.g. open-source software could be code-reviewed
- Transfer liability to supplier or insurer
  - Require the supplier or insurer to take responsibility of damages of possible security breaches
- Demand security standards
  - Use common security standards, which dictate the minimum security levels for the components
- Regulate
  - Force all entities to minimum cybersecurity level by laws and regulations
  - E.g. EU Cyber Resilience Act (CRA)

Alternatively the impact of vulnerability in a component can be limited: virtualization, containerization, and sandboxing.

Software Bill of Materials (SBOM)

- Modern software is less written, but more assembled from components
  - Majority of the software is not written inside the house
  - Thus, most risks are likely to be in the software components you use
- The *silver bullet* for this problem: SBOM
  - List the used software components
  - Map the list against CVE data to identify vulnerabilities (which may affect us - or not)
- However, SBOM does not solve the whole problem
  - How the component have been built?
  - How the component have been assessed?
  - How the component should be used for it to be secure?
  - How many non-public vulnerabilities there may be?

Security assessment is difficult

- Can’t we fix information asymmetry by testing and assessing the components ourselves?
  - Independently from the vendor (2nd or 3rd party)
  - All security testing techniques apply:
    - Security reviews and audits
    - Security scanning, fuzzing, … other tools
    - Pen-testing
    - Reverse engineering
- Problem: is it difficult and expensive
  - Some analysis can be done looking at interfaces of a component
    - Exposed network services, used protocols, etc.
    - Usability of security features, e.g. password change
  - Better analysis requires access to internals
    - Reverse engineering required expert skills and special tools
    - Even with source code and other internals available, analysis takes skills and time

Security standards etc.

Requiring a supplier to conform to cybersecurity standard

- Good in the way that supplier can do one standard and meet expectations of many customers
- Challenges
  - **Security theater**; how to ensure the component or service is actually secure
    - Strict compliance verification looks narrowly to the requirements
    - There are cracks between the standard requirements
    - Standard scope may be limited, e.g. devices only
  - It takes time and resources achieve standard compliance
    - Often requires expensive consulting/expert work
    - May be too expensive for small companies
  - What about using open-source or other components which are not standard compliant?
- Some standards
  - ISO 27001, Common Criteria, EN 18031, IEC 62433, ISO/SAE 21434

Legal contracts, insurance, and regulation

Legal contract solution:

- Demand suppliers to compensate for any damages caused by security incidents

Insurance solution:

- Insurance company pays the damages from incidents

Regulatory solution

- Make it illegal to have bad cybersecurity

Problems:

- Not all damages are financial: reputation, personal injuries, user mistrust
- Setting the price of legal protection or insurance requires knowledge of the risk
- Small and open-source supplies will not or cannot take responsibilities
- Proving that negligence was the cause of an incident is difficult
    - No perfect security -> vulnerabilities will happen -> bug itself is not a crime

IoT security assessment and standards

IoT security assessment

As we learned from previous slides, IoT security assessment is required to solve the supply chain insecurities:

Move security evidence along the supply chain

➔ Vendor must create the security information package

Verify security independently

➔ Security must be assessed independently of the vendor within the context of the supply chain

Transfer liability to supplier

➔ Liability premium or insurance fee requires assessing the risks related to the component

Demand security standards

➔ Standard compliance must be verified

Regulate

➔ Regulatory compliance must be verified

Aften an indent, the security of the breached system must be evaluated to see who is liable

Security standards and regulation

The state of IoT security combined with the importance of it has triggered action

- IoT security standards, guidelines and best practices
- IoT security regulation
  - In EU and other regions
- High-level requirements are quite similar in most of these
  - Reuse of requirements is norm (and good thing)
  - However, there is long tail of smaller differences between the standards

IoT security standards

- EN 18031 (Radio Equipment Directive)
  - Radio equipment connected to network
- ETSI EN 303 645
  - Consumer IoT security
- IEC 62443
  - Industrial IoT security
- And others…

Table 6. List of the security requirement sources from Paper II. Thr rows give the IoT domain, number of requirements, number of categories, and proportions of shared categories.

Source	Domain	Reqs	Cats.	Shared by			
				not	2-5	6-9	10-
C2 IoT Security Baseline [110]	Generic	30	28	0	21%	54%	25%
CTIA C.security Test Plan [111]	Generic	42	30	7%	40%	20%	33%
Cybersecurity Label [12]	Consumer	26	23	0	17%	48%	35%
ENISA Security Baseline [37]	Generic	129	91	4%	40%	40%	16%
ETSI EN 303 645 [18]	Consumer	90	58	5%	33%	40%	22%
FDA Draft Guidance [20]	Health	132	77	1%	40%	42%	17%
GSMA IoT Sec. Assessment [23]	Generic	437	117	7%	47%	32%	14%
ISASecure CSA [112]	ICS	351	109	5%	46%	36%	14%
IMDA IoT C.security Guide [113]	Generic	95	80	6%	34%	41%	19%
Internet Society IoT Trust [25]	Generic	85	60	2%	42%	40%	17%
IoT Security Initiative [114]	Generic	67	49	4%	43%	37%	16%
IoTSF Security Compliance [24]	Generic	366	125	6%	48%	34%	12%
IoT Acceleration Consortium [115]	Generic	63	30	10%	37%	33%	20%
NCSC Health Requirements [116]	Health	230	104	17%	40%	32%	11%
NHTSA Best Practises [117]	Vehicle	73	42	5%	40%	38%	17%
NISTIR 8259a [19]	Generic	27	18	0	44%	11%	44%
Source average		140	65	5%	38%	36%	21%
Total		2243	269	23%	51%	20%	6%

Transparent and tool-driven security assessment for sustainable IoT cybersecurity. Doctoral thesis. Kaksonen, Rauli (2024)

Which are the most common IoT security requirements?

Product requirements

Category	Sources	Prop.	Category	Sources	Prop.
Security design	16	9.3%	Security hardware	13	1.9%
– security architecture	16	8.9%	– tamper protection	11	0.4%
– subsystems	10	2.2%	Backend security	9	1.9%
– least privilege	15	1.5%	Cryptography	13	5.9%
Secure programming	7	1.9%	Data protection	16	14%
Delivery & deployment	14	4.8%	– protect secrets	11	0.7%
– deployment integrity	14	4.5%	– protection in transit	15	1.1%
– no global cred.	10	0.4%	– comm. standards	13	0.4%
Administration	13	5.6%	– decommission	11	0.7%
– secure config.	10	3.7%	Service availability	12	2.2%
Interface security	16	9.3%	Failure security	6	1.1%
– min. attack surface	13	4.8%	Audit logging	11	1.9%
– no unused interfaces	11	0.4%	Intrusion detection	15	5.2%
– validate input	10	1.1%	– software integrity	11	0.7%
Authentication	16	13%	Incident response	8	1.9%
– passwords	12	4.1%	System updates	16	5.9%
– auth. brute force	10	0.4%	– update security	14	0.4%
– component auth.	14	2.2%	Usability of security	10	2.2%
Access control	10	2.6%			

Transparent and tool-driven security assessment for sustainable IoT cybersecurity.
Doctoral thesis. Kaksonen, Rauli (2024)

Which are the most common IoT security requirements? Process requirements

Transparent and tool-driven security assessment for sustainable IoT cybersecurity. Doctoral thesis. Kaksonen, Rauli (2024)

Category	Sources	Prop.	Category	Sources	Prop.
Vendor security	8	3.7%	Security standards	14	2.2%
Policies & laws	10	6.3%	– comm. standards	13	0.4%
Development process	12	8.6%	Vuln. management	14	3.7%
– ext. components	10	1.1%	– known vuln.	10	0.7%
– security tests	10	4.1%	User communication	12	1.9%
Security requirements	14	4.5%			
– risk analysis	11	0.7%			

- **A notable missing category is "*privacy*" which was only present in 9 sources (required 10 to show up in the table)**

What needs testing?

Which attack vectors are most common in IoT?

- HTTP servers
- Network interfaces
- Local privilege elevation (after initial break-in)

Table 3. Attack vector distribution by subsystems and selected CVE data sets. Percentages given per subsystem and data set (i.e. row).

Subsystem / Data set	Data size	HTTP server	HTTP client	File	Local	Other	Unkn.	Other vectors
IoT device								
NVD++	22	27%	-	-	14%	41%	18%	vectors[1]
IoT malware	12	92%	-	-	-	8%	-	upnp
Network device								
NVD	10	40%	-	-	-	30%	30%	ip layer-2 tftp
IoT malware	21	95%	-	-	-	5%	-	smb
Mobile app								
Mobile app	61	3%	3%	5%	34%	20%	34%	tls wifi
Frontend								
NVD	28	82%	-	-	4%	-	14%	-
CISA	23	74%	-	-	-	-	26%	-
Mobile app	10	90%	-	-	-	-	10%	-
Backend								
NVD	48	52%	2%	8%	12%	10%	15%	ip tls udp
CISA	34	74%	-	3%	12%	3%	9%	ssh
OS								
NVD	21	-	-	5%	71%	14%	10%	vectors[2]
CISA	35	-	23%	9%	57%	3%	9%	smb
SW component								
NVD	24	-	12%	25%	8%	4%	50%	tls
CISA	19	-	~~95%~~	-	-	-	5%	-

[1] bluetooth dns ieee-1905 ip mqtt ntp wifi

[2] bluetooth sftp smb

Transparent and tool-driven security assessment for sustainable IoT cybersecurity. Doctoral thesis. Kaksonen, Rauli (2024)

What are most common vulnerability types in IoT?

- Input validation problems
- Authentication failures
- Other resource protection failures

*) Statistically significant increase

Table 5. Names of the presented root CWEs.

ID	CWE name
74	Improper Neutralization of Special Elements in Output Used by a Downstream Component ('Injection')
119	Improper Restriction of Operations within the Bounds of a Memory Buffer
287	Improper Authentication
20	Improper Input Validation
672	Operation on a Resource after Expiration or Release
706	Use of Incorrectly-Resolved Name or Reference
200	Exposure of Sensitive Information to an Unauthorized Actor
269	Improper Privilege Management
863	Incorrect Authorization

Table 4. CWE distribution per subsystem and selected CVE data sets. Percentages given per subsystem and data set (i.e. row). The asterisk* marks proportion significantly differing from the baseline.

Subsystem	Data	CWE-								
Data set	size	74	119	287	20	672	706	200	269	863
IoT device										
NVD++	26	8%	38%*	12%	12%	-	8%	4%		8%
IoT malware	11	82%*	18%	9%	-	-	-	-	-	-
Network device										
NVD	10	40%	20%	-	-	-	10%	-	-	-
IoT malware	20	80%*	5%	5%	-	-	10%	-	-	-
Mobile app										
Mobile app	51	8%	-*	39%*	-	2%	4%	16%*	-	8%*
Frontend										
NVD	27	48%*	-	-	4%	4%	-	7%	-	11%*
CISA	16	19%	12%	19%	-	-	19%*	-	19%*	-
Backend										
NVD	40	42%*	10%	5%		2%	5%	2%	5%	5%
CISA	31	35%	3%	13%	13%*	-	-	3%	10%	-
OS										
NVD	16	-	38%*	6%	-	6%	-	6%	19%*	12%*
CISA	33	-*	12%	9%	-	12%*	-	3%	52%*	-
SW component										
NVD	22	18%	27%	9%	5%	9%	-	-	-	-
CISA	16	6%	31%	-	12%	38%*	-	-	-	-
Baseline 2020-21		20%	14%	5%	4%	3%	3%	3%	3%	2%

Transparent and tool-driven security assessment for sustainable IoT cybersecurity. Doctoral thesis. Kaksonen, Rauli (2024)

OWASP Top 10 IoT vulnerabilities

A01 Broken Access Control

A02 Cryptographic Failures

A03 Injection

A04 Insecure Design

A05 Security Misconfiguration

A06 Vulnerable and Outdated Components

A07 Identification and Authentication Failures

A08 Software and Data Integrity Failures

A09 Security Logging and Monitoring Failures

A10 Server Side Request Forgery (SSRF)

Automation and sustainable security

Automating security testing

Automation has been pushed in software testing for a while

- Continuous integration / continuous delivery (CI/CD) require comprehensive testing
- AI can be a great help for test creation

Could automation solve supply chain vulnerability challenges?

- A tool can do much more tests than a human
- Running a security tool does not (necessarily) require security expert
- Tools scale to large number of products, versions, variants, and configurations
- Mostly to verify the product requirement
    - Process and human aspects hard to test by tools.. but some tools exist!

So, what we should test by the tools?

According the standards

- Security design
- Interface security
- Authentication
- Data protection
- System updates

Process-wise:

- Security requirements
- Security standards
- Vulnerability management
- Privacy

According vulnerability information

- Test all subsystems
  - Half of the vulnerabilities are in the backend
  - Only 10 - 20% in the IoT devices
- Most vulnerabilities are in used components
  - Tracking and timely mitigation is essential for IoT security
- Network interfaces
  - HTTP servers in particular
  - Injection vulnerabilities
  - Authentication
- Used security protocols
- Misconfigurations
- Mobile application permission problems
- 80 - 90% of published CVEs are potentially relevant for IoT systems

Do the popular tools match our needs?

Popular implies usability and effectiveness? Then tools to probe:

- Network / web interfaces and traffic
- Input validation
- Internal components and configuration?

Table 10. Security tool categories ranked by popularity of their top three tools.

Tool category	Top three tools	Avg. rank
Network scan	metasploit, nmap, sqlmap	3.3
Packet injection	metasploit, mitmproxy, sqlmap	4.3
Host interactive	metasploit, mimikatz, powersploit	6.7
Web security	nikto, mitmproxy, wpscan	9.3
Disassembly	ghidra, radare2, apktool	10.3
Packet capture	wireshark, tcpdump, aircrack-ng	11.7
Traffic analysis	wireshark, tcpdump, ntop	12.7
Password online	sqlmap, ophcrack, aircrack-ng	14.0
File analysis	radare2, clamav, pdfcrack	38.3
Password offline	hashcat, john, rainbowcrack	31.7
Malware execution	radare2, cuckoo, inetsim	50.0
Disk forensics	ddrescue, sleutkit, exundelete	50.7
Other	cyberchef, social-engineer-toolkit, steghide	52.0
Memory forensics	volatility	72.0

Transparent and tool-driven security assessment for sustainable IoT cybersecurity. Doctoral thesis. Kaksonen, Rauli (2024)

Not all standard tests can or should be automated

- E.g. ETSI TS 103 701 specifies total of 226 test
  - It turns out, full automation is not feasible
- Automate: 56 “security perimeter” tests
  - Attack surface or defence-in-depth
  - Testing and re-testing is critical:
    - The path in to the system
    - Changes in attack techniques and best practices
    - Regression in product updates
- What is left for manual testing
  - 120 "tests" for certification documentation
    - Stable through the product lifetime
  - 50 UI usability tests
    - Not directly part of attack surface
    - User-visible; review without security expert?

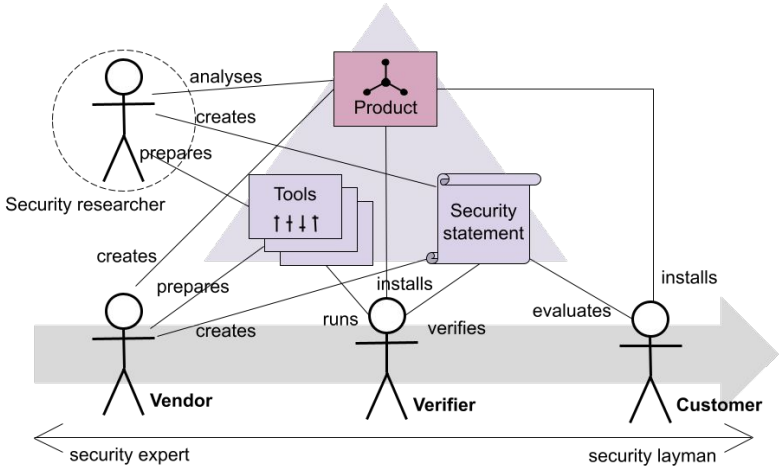
The proportions vary in other standards, but they likely have more and less testable requirements

Test units		**C**	**F**
All, T_A	226	124	102
IXIT and ICS review, T_X	106	106	-
Cross-references, T_C	14	8	6
Product security tests, T_P	106	10	96
- **Security perimeter tests, T_S**	**56**	10	46
- Product UI tests, T_U	50	-	50

Transparent and tool-driven security assessment for sustainable IoT cybersecurity. Doctoral thesis. Kaksonen, Rauli (2024)

TDSA: Tool-driven security assessment

Proposed in my thesis



IoT products are described by **security statements**

- Characteristics essential to security risk
  - Attack surface
  - Security controls
  - Update mechanism
  - Etc.
- Machine-readable and verifiable by tools!

Process for using TDSA:

- Creation of security statement
  - Vendor, contractor, security researcher
- Verification of security statement
  - Third party lab (3.), customer (2.)
- Use of security statement in IoT supply chain
  - Risk analysis in all phases
  - Verify product versions and configurations
  - Compare against security policy or standard
  - Compare multiple products

Security statement verification

Check that the security statement is accurate

Steps:

1. Set-up the product or component
2. Get the security statement
3. Run security tools; collect results
4. Use the product; capture traffic
5. Tool output is compared against the security statement
6. Complementary manual checks

=> Verdict: fail, pass, inconclusive

== do the product match the security statement?

Table 12. Security tools and services employed in the presented test automation.

Name	Description	Home page
APKPure [136]	APK repository	https://apkpure.net
Apktool [137]	APK analysis tool	https://apktool.org
Black Duck [132]	SCA service (commercial)	https://protecode-sc.com
Censys [138]	Internet search service	https://search.censys.io
Github [118]	Repository hosting service	https://github.com
HCIdump [139]	BLE recorder	https://www.bluez.org
Mitmproxy [140]	MITM attack tool	https://mitmproxy.org
Nmap [80]	Network scanner	https://nmap.org
Ssh-audit [141]	SSH test tool	https://github.com/jtesta/ssh-audit
Testssl.sh [126]	TLS test tool	https://testssl.sh
Tcpdump [142]	Traffic recorder	https://www.tcpdump.org
ZAP [143]	Web security scanner	https://www.zaproxy.org

Transparent and tool-driven security assessment for sustainable IoT cybersecurity.
Doctoral thesis. Kaksonen, Rauli (2024)

TDSA discussion

Automated testing coverage

- Is the tool test coverage significant?
- Huge improvement over “no testing”

How coverage could be improved?

- More testable requirements (in standards)
- Use of standard protocols and components
- New tools

Verification in supply chain

- Security statement allows to compare tool results to the claimed security posture
  - Attack surface
  - Security controls
  - Etc.
- The tools should be publicly available
  - Scripts or knowledge how to run the tools should be available, too

Some problems with TDSA

- How many would be able and willing to run the tools for independent verification?
- Can we trust open-source tools?
  - …but do we trust the human evaluators
- Handling of test failures

Summary

Supply chain vulnerabilities is a challenge for securing our IoT

- There is information asymmetry between client and the supplier

TDSA is an idea to make IoT supply chain more secure by transparency

- Security statement describes the security posture
- It can be verified by tools and automation
  - Allow independent security testing through the supply chain
  - Part of the security assessment remains manual work
- Sustainable: matches the scale of IoT

Currently push for supply chain security is SBOM

- Make total sense

=> Extensions to SBOM has been proposed

- Service/cloud SBOM
  - What cloud services are required?
- Configuration BOM, can mean either how…
  - the component is configured
  - the environment should be configured
- SLSA (Supply-chain Levels for Software Artifacts)
  - Data to confirm that software is built from the expected source
  - Can include passing of tests or other assessments in pipeline
  - Additional security: *in-toto attestation*

TDSA would complement these extensions

Workshop preview:
Transparent and Sustainable IoT Product Security by Toolsaf Framework