Vulnerabilities in IoT Devices, Backends, Applications, and Components

Rauli Kaksonen[a], Kimmo Halunen[b] and Juha Röning[c]
University of Oulu, Oulu, Finland

Keywords: IoT, Vulnerabilities, Cyber Security, Information Security, NVD, CVE, CWE, CVSS.

Abstract: The Internet of Things (IoT) is the ecosystem of networked devices encountered in both work and home. IoT security is a great concern and vulnerabilities are reported daily. IoT is mixed into other digital infrastructure both in terms of sharing the same networks and using the same software components. In this paper, we analyze Common Vulnerabilities and Exposures (CVE) entries, including known exploited vulnerabilities, to describe the vulnerabilities in the IoT context. The results indicate that 88% of reported vulnerabilities are relevant to IoT systems. Half of the vulnerabilities are in the backend or frontend systems while 10-20% concern the IoT devices. HTTP servers are the vulnerability hotspots wherever they are located. Software components are used in all IoT subsystems and tracking and updating them is essential for system security. The results can be used to understand where and what kind of vulnerabilities are in IoT systems.

1 INTRODUCTION

The *Internet of Things* (IoT) is the heterogeneous ecosystem of networked devices. IoT is deeply mixed with networked infrastructure, such as home networks, intranets, control systems, or hospital networks. The IoT backend systems are connected with the other business functions. The IoT mobile applications share the same smartphones as other applications. The security of IoT is a great concern, partly because of this connection with critical infrastructure. It is not only the IoT system and its data at stake but the whole system of systems. Further, an IoT system is largely assembled from open source and other software components. Some of them are specific to IoT, but most are truly general-purpose.

It is not practical to separate IoT security concerns from the big picture of cyber security. In this paper, we analyze reported vulnerabilities and their connection to IoT. The goal is to understand how vulnerabilities are distributed and which are the most important flaw categories. This should be useful for security researchers, administrators, authorities, and others who want to understand IoT security better.

The common vulnerabilities and exposures (CVE) program maintains public vulnerability information (MITRE, 2022a). The National Vulnerability Database (NVD) is US government repository that further tracks and augments CVE data (NIST, 2022a). NVD includes also IoT vulnerabilities, but they are not explicitly identified. Some CVEs represent vulnerabilities which are used in cyber-attacks while many are never exploited. US Cybersecurity & Infrastructure Security Agency (CISA) maintains a *known exploited vulnerabilities catalog* which lists actively exploited CVEs (CISA, 2022).

A CVE entry usually has *Common Weakness Enumeration* (CWE) value(s), which describes the type of underlying weaknesses (MITRE, 2022b). The view CWE-1003 "Weaknesses for Simplified Mapping of Published Vulnerabilities" is a hierarchy of CWEs intended to be used in NVD (MITRE, 2022b; NIST, 2022a). The view has 35 root CWEs, which may have child CWEs. We use this as compression and only show the roots with child CWE counts summed in. Common Vulnerability Scoring System (CVSS) describes the characteristics of a vulnerability and calculate its technical severity (FIRST, 2022). The *Common Platform Enumeration* (CPE) is intended to unambiguously identify the vulnerable product or software (NIST, 2022b).

Information about vulnerabilities in IoT is abundant. Open Web Application Security Project (OWASP) is perhaps best known for its Top 10 Web Application Security Risks, but it has also published Top 10 Internet of Things Vulnerabilities (OWASP, 2022). Khoury et al. inspected 27 IoT malware sam-

[a] https://orcid.org/0000-0001-8692-763X
[b] https://orcid.org/0000-0003-1169-5920
[c] https://orcid.org/0000-0001-9993-8602

Kaksonen, R., Halunen, K. and Röning, J.
Vulnerabilities in IoT Devices, Backends, Applications, and Components.
DOI: 10.5220/0011784400003405
In *Proceedings of the 9th International Conference on Information Systems Security and Privacy (ICISSP 2023)*, pages 659-668
ISBN: 978-989-758-624-8; ISSN: 2184-4356

ples and identified the CVEs used by them to predict which are going to be used by future malware (Khoury et al., 2021). Several technical reports list CVEs used in IoT malware or botnets (Caspi, 2022; Caspi, 2021; Luptak and Palotay, 2021). Chatzoglou et. al. performed a thorough study of Android IoT application security (Chatzoglou et al., 2022). Mathas et al. analyzed vulnerabilities in open-source software applicable to 5G IoT devices (Mathas et al., 2021). Mirani et al. tested 13 IoT devices and discovered vulnerabilities in all of them (Mirani et al., 2019). We look closer at these studies in Section 3.4 and compare them to our results.

To the best of our knowledge, there are no studies which holistically analyze reported vulnerabilities in the IoT context. Our research questions in this study are: 1) **In the IoT context, where would one expect the vulnerabilities to appear?** 2) **What are the most frequent types of vulnerabilities in the IoT context?**

2 METHODS

We use the real vulnerability information from the NVD (NIST, 2022a). A careful analysis of the CVEs are required to understand what is vulnerable and how. The amount of detail and presentation varies greatly, so we chose to manually analyse the CVEs for accuracy. With around 20 000 CVEs added in a year, we can only analyze subsets. Table 1 summarizes the analyzed CVE data sets.

Firstly, we analyzed a random set of CVEs from 2021. Later, we created additional "NVD++" data set to get more insight to the IoT device vulnerabilities. The set is made up from randomly selected CVEs, which were analyzed in-depth only if they appeared to be relevant for IoT devices.

At the time of our research, the CISA set contains 143 known exploited vulnerabilities for the year 2021 (CISA, 2022). Vulnerabilities used in IoT malware, mainly device vulnerabilities, are listed by many sources (Caspi, 2022; Caspi, 2021; Khoury et al., 2021; Luptak and Palotay, 2021). We analyzed all these CVEs.

The frequency of mobile applications in the aforementioned CVE sets is low. To understand mobile application vulnerabilities better, we searched for CVE descriptions with the words "*mobile*", "*android*", "*ios*" or "*phone*" followed by "*app*" from years 2020-2021. The returned CVE set allows us to analyze applications in general, but unfortunately not specific to IoT. This search also discovered CVEs for other subsystems when the search words were mentioned in the CVE.

Table 1: Sets of analyzed CVEs.

Set	Years	CVE selection criteria
NVD	2021	Random
NVD++	2021	NVD with additional *IoT device* vulns.
CISA	2021	CISA known exploited vulnerabilities
IoT malware	2018-21	Vulnerabilities used in IoT malware
Mobile apps	2020-21	CVEs for mobile applications
Search hits	2021	CVEs with $>$ 5000 Google search hits

Finally, we performed a Google search using Google API Client python library using all CVE identifiers for the year 2021 (form "CVE-2021-dddd[d]") (Google, 2022). This gives us a view into CVEs by their web presence. The set "Search hits" is the list of CVEs with more than 5000 search hits.

As the data sets are selected with different criteria, we cannot derive conclusions about vulnerability proportions between the data sets. Thus, we focus on variations within each data set.

2.1 Inferring Subsystems

To place the vulnerabilities in the IoT context, we use a simple model with the following subsystems.

- *Devices* located on user premises, e.g. homes, factories, or hospitals. A *network device* provides general-purpose network connectivity and an *IoT device* delivers a special function or service.
- *Backend* system providing the business logic, data store, and other services required by the IoT system. A backend is typically located inside cloud service provider data centers.
- *Frontend* is the interface of the backend towards Internet. A frontend is connected by IoT devices, applications, or web browsers.
- *Mobile* applications acting as the user interface for the system.
- *Operating system* (OS) controls hardware and provides the basic functionality to build devices, services, applications, etc.
- Finally, all subsystems are built from a large number of general-purpose software *components*.

Devices are located outside of dedicated server rooms and connected by a network. They can be often reached by outsiders, either from the Internet, over a wireless network, or physically. We separate the Internet protocol (IP) *network devices* such as routers, switches, and firewalls from the "true" *IoT devices*, such as IP cameras, remote-controlled valves, and smart sockets. An IoT hub or gateway acting as an intermediary between IoT devices and/or IP-network is considered an IoT device.

The distinction between *frontend* and *backend* allows differentiating the vulnerabilities exposed over the Internet and the ones exploitable internally. IoT devices typically connect to frontend by a web Application Programming Interface (API) over HyperText Transfer Protocol (HTTP). As the frontend is open to the Internet, it is experiencing constant scanning and intrusion attempts. To be categorized to the frontend, a component must be intended to be Internet-facing, otherwise it is backend. In practice, the division is hard to make and the differences between the two subsystem results are indicative rather than conclusive.

Many CVEs are for the major operating systems: Android, iOS, MacOS, Windows, and Linux variants. Android is well-known as a mobile phone OS, but it is also used in IoT. Linux is used everywhere, in devices, desktops, and servers. Windows is used in the desktop, servers, but also in IoT devices. Apple's iOS is used in mobile phones, pads, and other personal devices. MacOS is to the best of our knowledge not used outside desktops. Apple has also tvOS.

When a vulnerability is reported in the interaction between a frontend and a device or mobile application, it is categorized as the CVE describes the situation. A CVE for a software component is categorized by the primary function of the component with the explicit software *component* category as a fallback. For example, a web API component goes to the *frontend* and a mobile application framework goes to the *mobile*. An image rendering component would be categorized as a component. Many vulnerabilities in a subsystem are actually in software components.

2.2 Inferring Attack Vectors

For our study, we identify the attack vector(s) which enable the exploitation of a vulnerability. CVSS already contain an *Attack Vector* metric, but it does not give details like the network protocol (FIRST, 2022). We infer a more detailed attack vector.

Browsing the CVE data quickly reveals that the most common attack vector is Hypertext Transfer Protocol (HTTP). HTTP is so prevalent that it is often not explicitly mentioned, but it must be deduced from the description. Vulnerabilities are also reported on Domain Name System (DNS), Server Message Block (SMB), Transport Layer Security (TLS), X.509 certificate handling, Internet Protocol (IP), Wireless Local Area Network (WLAN), Bluetooth, etc.

Some vulnerabilities are in the parsing of different *file* formats, with the vector of delivering the file into the system often unknown. This includes failing to properly sanitate stored files. Many vulnerabilities are *local*, they cannot be exploited unless the attacker has access to the system, e.g. as a regular user. Vulnerabilities requiring physical access to the vulnerable systems are included to the local attack vector. For some CVEs, it is not possible to infer the attack vector from the available information.

Table 2: CVSS version 3 metrics and values, with an asterisk marking the worst value for the defender.

	Metric	**Values**
AV	Attack Vector	Physical, Local, Adjacent, Network*
AC	Attack Complexity	Low*, High
PR	Privileges Required	None*, Low, High
UI	User Interaction	None*, Required
S	Scope	Unchanged, Changed*
C, I, A	Confidentiality, Integrity, Availability	High*, Low, None

2.3 CVSS Metrics

CVSS is used to estimate the technical severity of vulnerabilities (FIRST, 2022). The CVSS version 3 *Base metric group* contains the context and time-independent information, see Table 2.

The metric *Attack Vector* defines the context where exploitation is possible. (As said, we infer a more detailed attack vector for analyzed CVEs.) *Attack Complexity* is high if exploitation requires conditions beyond attacker control and low if it does not. *Privileges Required* describes the level of privileges the attack requires. A vulnerability may require *User Interaction* to succeed. *Scope* indicates if the security impact crosses security domains. Finally, the impact metrics *Confidentiality*, *Integrity*, and *Availability* estimate the effect of successful exploitation on the system security goals. The value we consider the *worst value* from the defender's point of view is marked by an asterisk in Table 2.

In the results, we highlight how the CVSS metrics vary between different CVE data sets and subsystems. To avoid the problem of converting CVSS metrics into numeric values, we use the proportion of the worst value. In NVD data the worst value is the most common value for all but the Scope metric, as the value *Unchanged* is more common than *Changed*. The proportions are compared to the averages from CVEs in 2020 and 2021.

2.4 Margin of Error

We calculate the *margin of error* (MOE) for some resulting proportions $M_e = z\sqrt{p(1-p)/n}$, where p is the proportion, n is the number of samples, and $z = 2.57$ for 99% confidence level or $z = 1.96$ for 95% confidence (Sheldon M., 2004, p. 260-262).

Table 3: Division of the vulnerabilities in subsystems by analyzed CVE data sets.

	NVD	CISA	Search hits
Total	**156**	**143**	**135**
Device	10%	14%	
Mobile app	1%		
Frontend	18%	16%	25%
Backend	31*%	24%	13%
OS	13%	24*%	31*%
Other	12%	17%	13%
Component	15%	13%	23%
95% margin	±7%	±7%	±8%

Table 4: Vulnerabilities reported in identified OSes in 2021 based on CPE information. MacOS and non-kernel Linux vulnerabilities are not added to the total.

	NVD all	CISA	Search hits
OS total	**11%**	**25%**	**30%**
Android	4.1%		
Linux kernel	0.8%		1.5%
(Linux all)	(4.5%)	(9.1%)	(23%)
Windows	2.7%	15%	22%
Apple iOS	1.6%	9.8%	6.7%
Apple tvOS	1.0%	2.8%	3.0%
(Apple MacOS)	(2.0%)	(8.4%)	(8.1%)
Other OS	1.9%		

3 RESULTS

The CVEs were selected for different data sets and a total of 577 CVEs were analyzed. 484 were from the year 2021, but some were older up to 2018. Table 3 gives the proportions of CVEs in different subsystems in NVD data, CISA known exploited data, and top search hit data. Some CVEs are in multiple data sets and/or subsystems, thus there is overlap. The OS row includes Android, Linux kernel, Windows, Apple iOS and tvOS, and other OSes considered relevant in the IoT context. Table 4 shows the breakdown of all CVEs 2021 by OS.

In the generic NVD data, the vulnerabilities are distributed 31% to backend, 18% to frontend, 15% in components, 13% in OSes, 10% in devices, and 1% in mobile. Thus, half of the CVEs are for the frontend or backend. Most of the 12% of CVEs not considered relevant are for desktop software or MacOS.

In CISA exploited vulnerabilities, the OS category is increased to 24% share and in search hits to 31%. The frontend vulnerabilities overtake the backend 25% to 13%. Component vulnerabilities are increased to 23% share.

The high search data share of OSes and components is likely due to the high visibility of Windows and Linux-related vulnerabilities. This can be observed in the large share of "Linux all" row in Table 4 for the search result. Many exploited vulnerabilities are in Windows, which remains the most exploited OS in this study, as well. The results indicate that while there are more vulnerabilities in backend components, the vulnerabilities in the frontend may be considered more serious. Many frontend components are closely followed and their vulnerabilities are widely reported, e.g. HTTP servers and security gateway products. The last row of Table 3 gives the largest margin of error with a 95% confidence interval for each set.

In table 4 the OS breakdown is based on the CPE information in all CVEs from 2021, not on analyzed CVEs only. The row "Linux all" contain CVEs for major Linux variants including various utilities which are not really part of the OS. For this reason, we only include kernel vulnerabilities in the OS count. MacOS is not included in the OS count. The "Other OS" row in the table does not capture those smaller OSes, which we did not encounter in our analysed CVEs, as we do not know their CPEs. Thus, the real proportion is likely higher. This also explains why the total proportion is 11% while in Table 3 it is 13%.

3.1 Common Attack Vectors

Table 5 shows the distribution of vulnerabilities by subsystems and attack vectors. The table presents CVE data set per row with the row percentages summing up to 100%.

The number of CVEs per row is in the column "Data size". The attack vectors summed in the "Other" column are listed in the column "Other vectors". Devices are split into *IoT devices* and *network devices* with the former containing the networking devices and the latter the "true" IoT devices. As the share of IoT devices in NVD data is low, an additional "NVD++" data set was created to analyze them better.

In the NVD data, the attack vectors for IoT devices were quite scattered as 41% are in "Other" attack vector. The HTTP server is the most frequent vector with 27% share in NVD data.

In the IoT malware data set all but one exploited vulnerability is in HTTP servers. The attack vectors for network devices are similar.

The CVEs for *mobile* applications were searched using keywords, as explained earlier. The largest group is the locally exploitable vulnerabilities. These are weaknesses which allow e.g. a rogue application to access sensitive data or functionality within the phone. Many of the issues with unknown attack vector are likely to be either local or HTTP connection issues, but the CVEs fail to pronounce this.

Table 5: Vulnerability distribution in subsystems and attack vectors.

Subsystem	Data set	Data size	HTTP server	HTTP client	File	Local	Other	Unknown	Other vectors
IoT device	NVD++	22	27%			14%	41%	18%	bluetooth dns ieee-1905 ip mqtt ntp wifi
	IoT malw.	12	92%				8%		upnp
Network device	NVD	10	40%				30%	30%	ip layer-2 tftp
	IoT malw.	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%	bluetooth sftp smb
	CISA	35		23%	9%	57%	3%	9%	smb
Component	NVD	24		12%	25%	8%	4%	50%	tls
	CISA	19		95%				5%	

Most of the *frontend* vulnerabilities are in HTTP servers, which is expected for the wide use of HTTP in connections. For the exploited frontend vulnerabilities, the share of HTTP servers is lower, but many of the unknown attack vectors are likely also HTTP. Nine HTTP server frontend vulnerabilities were also found by the mobile application CVE search. HTTP is the most important attack vector also for the *backend*, but there are several locally exploitable vulnerabilities, too. In operating systems, most vulnerabilities are locally exploitable, both in generic NVD data and CISA exploited vulnerability data.

Many of the generic software *components* are intended to handle input data in various unspecified situations and the attack vector is set to file or unknown. The large number of HTTP client vulnerabilities in CISA data are for *Chromium* web browser component (The Chromium Projects, 2022).

3.2 Common Weaknesses

Table 6 gives CWE distribution per subsystem and selected data sets. Only root CWEs from the view CWE-1003 are shown, with possible child CWEs included in the count. Proportions are given for subsystem and data set, i.e. per row. When there are no CWEs for specific subsystem and data set, the value is omitted. The table includes the CWEs with at least 11% share in at least one row. This includes all root CWEs with at least 2% share among all CVEs in 2020 and 2021. Asterisk marks cells, where the proportion differs from the baseline with 99% confidence.

Table 7 lists the names of the shown CWEs.

For *IoT devices* the most common weakness type in NVD data is CWE-119 “Improper Restriction of Operations within the Bounds of a Memory Buffer”. For the IoT malware data set, the dominating weakness is CWE-74 “Improper Neutralization of Special Elements in Output Used by a Downstream Component (’Injection’)”. For *network devices* the size of NVD data does not allow much interpretation, but the most common CWE in malware is also CWE-74.

For mobile applications there are many authentication and access control issues by CWE-287 “Improper Authentication” and CWE-200 “Exposure of Sensitive Information to an Unauthorized Actor” and to some extent in CWE-863 “Incorrect Authorization”. There are no CWE-119 issues, likely due to the use of managed languages, like Java.

Frontend NVD data set top weakness is CWE-74 for command injection. The exploited vulnerabilities are distributed to several CWEs. In the *backend* the top weakness in both NVD and in CISA data is CWE-74. Operating system top weakness is CWE-119, there are no CWE-74 vulnerabilities. In the exploited OS vulnerabilities, the top weakness is CWE-269 “Improper Privilege Management”. *Component* weaknesses roughly follow the baseline distribution. The peak weakness CWE-672 is due to Chromium browser issues (The Chromium Projects, 2022).

3.3 CVSS Scores

Table 8 shows the CVSS version 3 scores and base metrics for IoT vulnerabilities in selected CVE sets and subsystems. Only the CVEs with CVSS scores are included. The metrics are compared to the *worst value* proportions in baseline CVEs 2020-2021. The table shows the significant (99% confidence level) base metric changes with the minimal difference of 0.2. Base metrics are identified by abbreviation (see Table 5) followed by the delta to the baseline value. For example, in the CISA data set the proportion of CVEs with the worst value *High* in Confidentiality,

Table 6: CWE distribution per subsystem and selected data sets. CWE proportions are given for subsystem and data set. Asterisk marks cells, where the proportion significantly differs from the baseline.

Subsystem	Data set	Data	CWE								
		size	74	119	287	20	672	706	200	269	863
IoT device	NVD++	26	8%	38%*	12%	12%		8%	4%		8%
	IoT malw.	11	82%*	18%	9%						
Network device	NVD	10	40%	20%				10%			
	IoT malw.	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%*	
Component	NVD	22	18%	27%	9%	5%	9%				
	CISA	16	6%	31%		12%	38%*				
All 2020-2021		38902	20%	14%	5%	4%	3%	3%	3%	3%	2%

Table 7: Names of the shown root CWEs.

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

Integrity, and Availability is risen by 0.3, 0.4 and 0.3, respectively. In the NVD data set, there are no significant changes, which is expected as NVD is a random sample over the year 2021. This applies also in full data over the years 2018 and 2021, the yearly variation in all metrics is less than 0.2, thus the shown differences are not caused by trends.

The CVSS scores are elevated for the IoT malware, CISA, and top search hit CVEs. This is by the bigger CIA impact but for IoT malware also due to an increase in the Attack Vector (AV), User Privileges (PR), and User Interaction (UI) proportions, i.e. vulnerabilities exploitable through the network without user priorities or interaction. The exploitation of vulnerabilities in *mobile applications* also requires only a few user privileges.

For subsystems, *device* vulnerability CVSS scores did not significantly differ from the baseline, thus they are omitted. Subsystem CVEs are from the NVD data set, except the mobile apps which have their dedicated data set. AV is larger for *frontends*, as expected since those are publicly exposed. The *backend* and OS vulnerabilities often require initial user privileges to be exploited. OS vulnerabilities also have reduced AV. However, component vulnerabilities on average require few privileges.

Table 8: CVSS base metric changes in different CVE sets and subsystems, proportion of worst value.

	CVEs	Score	Metrics, worst value proportion
Baseline 2020-2021	37131	7.1	AV=0.7 AC=1.0 PR=0.6 UI=0.7 S=0.2 C=0.6 I=0.5 A=0.6
By CVE sets			
NVD	156	6.9	
CISA	143	8.5	C+0.3 I+0.4 A+0.3
IoT malw.	48	9.4	AV+0.3 PR+0.2 UI+0.3 C+0.4 I+0.5 A+0.4
Mobile app	82	6.7	PR+0.3 A-0.4
Search hits	134	8.2	C+0.3 I+0.3 A+0.3
By subsystems (Mobile app or NVD sets)			
Mobile app	61	6.5	PR+0.2 A-0.4
Frontend	28	6.5	AV+0.3 A-0.3
Backend	48	6.8	PR-0.2
OS	21	7.0	AV-0.5 PR-0.3 UI+0.2 C+0.4
Component	24	7.4	PR+0.3 UI-0.3 A+0.3

3.4 Comparison to Related Work

Table 9 presents vulnerability or risk categories from selected other sources. The rank or frequency of each category is shown, as given in the source. The results are mapped to our study by giving the number of CVEs for each category per relevant CVE data sets, if any. Also, a total count over the data sets is given. The mapping is performed using subsystems and/or CWE numbering. The entry "Not in CVE data" is used when the category cannot be expected to be found from CVEs.

OWASP Top 10 Internet of Things vulnerabilities is intended to describe the whole IoT system, thus we map it into CVEs in data sets NVD, CISA and Search hits. The highest vulnerability category is *Weak, Guessable, or Hardcoded Passwords* which covers all credentials, not just passwords. Just 9% of

Table 9: IoT vulnerability or risk categories from selected sources mapped into analyzed CVE sets, if possible.

OWASP Top 10 IoT	*Rank*	NVD (156)	CISA (143)	Search hits (135)	**Total** (381)
Weak, Guessable, or Hardcoded Passwords	*1*	11%	9%	7%	**9%**
Insecure Network Services	*2*	19%	23%	5%	**17%**
Insecure Ecosystem Interfaces	*3*	24%	31%	51%	**33%**
Lack of Secure Update Mechanism	*4*	Not in CVE data			
Use of Insecure or Outdated Components	*5*				**0%**
Insufficient Privacy Protection	*6*	Not in CVE data			
Insecure Data Transfer and Storage	*7*	26%	34%	27%	**27%**
Lack of Device Management	*8*	Not in CVE data			
Insecure Default Settings	*9*	Not in CVE data			
Lack of Physical Hardening	*10*	Not in CVE data			
Not mapped		*46%*	*42%*	*40%*	**43%**
OWASP Top 10 Web / Frontend	*Rank*	NVD (27)	CISA (16)	Search hits (20)	**Total** (56)
Broken Access Control	*1*	44%	56%	45%	**43%**
Cryptographic Failures	*2*				**0%**
Injection	*3*	52%	25%	15%	**36%**
Insecure Design	*4*		25%	10%	**7%**
Security Misconfiguration	*5*				**0%**
Vulnerable and Outdated Components	*6*				**0%**
Identification and Authentication Failure	*7*		19%	10%	**7%**
Software and Data Integrity Failures	*8*	7%		5%	**5%**
Security Logging and Monitoring Failures	*9*				**0%**
Server-Side Request Forgery	*10*		6%	5%	**2%**
Not mapped		*4%*	*12%*	*40%*	**20%**
Mirani et. al / Device	*Frequency*	NVD (16)	IoT malw. (31)		**Total** (47)
Buffer Overflow	*8%*	31%	10%		**17%**
Cross-Site Scripting	*92%*	12%			**4%**
Command/SQL Injection	*92%*	25%	81%		**62%**
Authentication Bypass	*38%*	6%	6%		**6%**
Authorization Bypass	*38%*	12%	6%		**9%**
Cross-Site Request Forgery	*38%*				**0%**
File Upload Path Traversal	*54%*	12%	6%		**9%**
Not mapped		*19%*	*0%*		**6%**
Mathas et. al / IoT Devices	*# of vuln.*	NVD++ (36)	IoT malw. (31)		**Total** (67)
Improper Certificate Validation	*2*	8%	6%		**7%**
Buffer Overflow	*104*	33%	10%		**22%**
Weak cryptography	*1*				**0%**
Sensitive data exposure	*22*	3%			**1%**
Race condition	*12*				**0%**
Broken access control	*0*	3%			**1%**
Not mapped		*53%*	*84%*		**67%**
Chatzoglou et. al / Mobile app	*Frequency*	Mobile app (51)			**Total** (51)
Permissions	*very common*	20%			**20%**
APK signing and Janus	*95%*				**0%**
Network security and certificates	*49%*	16%			**16%**
CWE-311	*100%*	6%			**6%**
CWE-327	*98%*	2%			**2%**
CWE-330	*100%*	2%			**2%**
CWE-345	*76%*	2%			**2%**
CWE-732	*98%*	2%			**2%**
CWE-74	*95%*	8%			**8%**
CWE-287	*46%*	39%			**39%**
CWE-200	*100%*	16%			**16%**
CWE-749	*55%*				**0%**
CWE-919	*39%*				**0%**
Tracker analysis	*95%*	Not in CVE data			
Shared library analysis	*95%*	Not in CVE data			
Outdated software components	*57%*	Not in CVE data			
Not mapped		*18%*			**18%**

the CVEs are mapped to it.

Category *Insecure Network Services* maps to all *device* vulnerabilities and *Insecure Ecosystem Interfaces* to *frontend*, and *mobile*. Lacking a category for OSes, Windows and Linux is placed into the latter and other OSes into the former category. With this, the categories cover 17% and 33% of the analyzed CVEs, respectively. The category *Insecure Data Transfer and Storage* covers 27% of the CVEs. The other categories cannot be easily mapped into CVEs. In the end, we are left with 43% of unmapped CVEs. These are *backend* and general-purpose *component* vulnerabilities.

It may appear that the component and OS subsystem vulnerabilities could be mapped into *Use of Insecure or Outdated Components* category. However, the category is for components which are not timely updated or are insecure in another way. This information is not available in the data.

OWASP Top 10 Web Application Security Risks is a popular information source relevant for the *frontend* subsystem (OWASP, 2022). The source lists the CWEs for each category and we map the CVEs to categories solely based on this. The resulting coverage is fairly good, with just 20% of the CVEs left unmapped. For general NVD data, this is only 4%.

Categories *Broken Access Control* and *Injection* are the most prevalent in CVEs with frequencies of 43% and 36%. The category *Cryptographic Failures* was not observed. Three other categories do not have any CVEs, again the reason is likely that even when these weaknesses are present no CVE is created.

Mirani et al. performed security testing on 13 home or small-office network devices (Mirani et al., 2019). We map the results to the *device* subsystem. The vulnerabilities are all in HTTP servers, but we did not limit the CVEs by HTTP attack vector, as we wanted to get the frequencies over all vectors. The mapping was very good with only 6% of CVEs left unmapped. The result is so high as all CVEs used by IoT malware were covered by vulnerabilities by the Mirani et al. Still, even with generic NVD data only 19% of CVEs were unmapped.

Mathas et al. searched for vulnerabilities from 11 software modules in the context of 5G IoT devices by source code analysis (Mathas et al., 2021). For each category, we show the number of found vulnerabilities. The relevant CWEs for categories are given and used by us to find the CVEs for the categories. Before this, view CWE-1003 child CWEs are replaced by their respective root CWEs for more loose matching. As these are true IoT devices we use the larger IoT device CVE data set *NVD++*. It appears that there is consensus only for the buffer overflows, which are found en masse by Mathas et al. and responsible for 22% of IoT device CVEs. Overall 67% of CVEs do not fall into the categories presented by Mathas et al.

Chatzoglou et al. performed a comprehensive study of various Android IoT applications (Chatzoglou et al., 2022). The analysis includes application permissions, packaging, trackers, shared libraries, outdated components, taint analysis, and dynamic analysis. Static analysis was performed for 41 devices, but dynamic analysis only for 13. We only include the former due to its larger coverage. This material maps to our *mobile app* subsystem. The first category *Permissions* is an aggregate category indicating that the studied applications tended to request many permissions and/or failed to set proper permissions. Suspicious permissions were discovered from all applications, but some of them are indeed necessary while some are not. For this, we did not put "100%" as the frequency.

A total of 20% of CVEs was identified to be caused by excessive permissions. *APK signing and Janus* are about the flaws in application packaging, which were not present in the CVEs. *Network security and certificates* are about the lack of encryption or bad certificate handling. This mapped into 16% of the app CVEs.

Chatzoglou et al. also listed the CWEs discovered from the applications by static analysis. We use them to pick relevant CVEs, but we again replace CWEs with the respective roots from the view CWE-1003. The table lists the discovered CWEs except for CWE-250 and CWE-913 as they were only found from one app and are not present in CVEs. Only two of the CWEs were frequently found from the analyzed CVEs, CWE-287 "Improper Authentication" and CWE-200 "Exposure of Sensitive Information to an Unauthorized Actor". Many CWEs were very common in the static analysis but much rarer in CVEs. It may be that static analysis finds weaknesses which are not real vulnerabilities or at least not exposed in the wild. *Tracker analysis* revealed that IoT applications are littered with different tracking modules which raise significant concerns over user privacy. This is not at all visible looking at CVEs. *Shared library analysis* examined mostly if included libraries were hardened against exploitation. Again, these issues are not present in the CVE data. The same goes for *Outdated software components*. Overall only 18% of CVEs were not mapped into any category.

3.5 Threats to Validity

As we use manual analysis for accuracy, the number of analyzed CVEs is limited and we may have missed some considered influential in some respect. However, as our focus was on the distribution of vulnerabilities, the role of individual CVE is small.

Some CVEs are somewhat vague and it is not possible to infer the subsystem or the attack vector. We believe the only systematic error by this is that HTTP appears less dominant as many CVEs fail to indicate HTTP even when it is the vector.

The quality of CPE data looks low and different values for the same product can be easily found. As we only used CPEs for OS detection the impact in our study is limited. A bigger worry is possible inconsistencies in CWE assignments. We did not have a chance to inspect this, but it is something which should be kept in mind interpreting the results.

Mostly we used data from the year 2021, but some data sets contain older material to get enough CVEs to analyze. As old vulnerabilities can be used to compromise unpatched systems today, we do not think this distorts the results. Further, when we looked at CVSS data, we did not observe significant changes in CVE metrics since 2018.

4 DISCUSSION

In this study, we looked at the distribution and type of vulnerabilities in IoT systems. IoT is fundamentally mixed into other digital infrastructure both in terms of sharing the same networks and using the same software components. Thus, IoT vulnerabilities cannot be easily isolated. We selected several sets of CVEs, mainly from the year 2021. As the quality and quantity of CVEs vary, we used careful manual analysis to understand what is vulnerable and the type of the vulnerability in IoT context. In the end, only 12% of the vulnerabilities we studied were not relevant to IoT.

In the first research question, we ask how reported vulnerabilities map into the IoT subsystems. The exact distribution varies in data sets as their selection criteria are different. Within generic NVD data, 31% of the issues can be assigned to the *backend*, 18% to the *frontend*, 15% to generic *components*, 13% to *OS*es, 10% to *devices*, and 1% to mobile *applications*.

Looking at the exploited vulnerabilities, then the proportion of backend vulnerabilities is reduced to 24%, frontend is risen to 16%, and OS to 24%.

In top search CVEs, these percentages are 13%, 25%, and 31%. There are no device issues in the top search CVEs. It appears that frontend and OS security problems are getting attention while IoT device security is not. If we look at the results for the OWASP IoT Top 10 then 17% of the CVEs are in devices and 33% in ecosystem interfaces. The numbers are different as OS vulnerabilities are not separately counted.

Many, if not the majority, of the reported vulnerabilities are in software components used in the subsystems. The 15% share of vulnerabilities in components cover only the truly general-purpose components. All subsystem categories include vulnerabilities in domain-specific components or whose underlying cause is a flaw in a component even when the report is about a device or application. Further, three out of five referred studies emphasize the use of bad or outdated components as a major source of vulnerabilities. Thus, monitoring and mitigating the vulnerabilities in the component supply chain is essential. This is not possible unless there is an up-to-date *bill of materials* (BOM) identifying the used components.

The second research question was about the type of vulnerabilities in IoT. The results indicate that HTTP servers are the hotspot of vulnerabilities in devices, frontends, and backends. This is not surprising as HTTP is such as common protocol. The most significant weakness category is CWE-74 “Improper Neutralization of Special Elements in Output Used by a Downstream Component (’Injection’)” and its child CWEs. IoT malware mostly uses injection problems in HTTP servers to propagate.

CWE-119 “Improper Restriction of Operations within the Bounds of a Memory Buffer” is still frequent in IoT devices and OSes. This is a bit frustrating as buffer overflow has been a problem for a long time, one could have expected them to be eliminated already.

Backend systems and OSes have many issues with the *local* attack vectors. These can be exploited by insiders, e.g. by a legitimate user or attacker who already has a beachhead in the system. Patching local issues is important to maintain the *defence in depth*.

Mobile applications vulnerabilities were infrequent compared to other subsystems. In them, many CVEs are related to different authentication and authorization weaknesses. These are exposed when connecting to frontend or IoT devices, which seems to be challenging to get right. Applications have also locally exploitable issues allowing a malicious application to compromise other applications in the smartphone. Chatzoglou et al. back this finding as most IoT-related applications seem to request far more permissions than required and/or fail to limit access to their resources by other apps.

Comparing our results to the other related studies give mixed signals. There is some consensus on

the big role of HTTP servers, authentication, authorization, and injection issues in IoT vulnerabilities. The danger of using outdated or bad software components is mentioned by several referred studies. Beyond these, there are significant differences in the results. This may be due to different assessment strategies. Especially many weaknesses found by static analysis do not seem to correspond to CVEs. It is difficult to say whether this is because static analysis finds weaknesses which are not vulnerabilities or because these vulnerabilities are hard to find. Comparison is also complicated by the fact that results are presented differently.

Finally, a reader should understand that there are categories of vulnerabilities which are not visible in CVE data. Many security fixes are done by the vendor silently without creating a CVE. Excessive data collection and other privacy issues and all too frequent private data disclosures are not tracked by CVEs. Missing security functionality, such as lack of software update mechanism or lack of audit log, is not covered by CVEs, either.

5 CONCLUSIONS

We analyzed 577 CVEs to learn the distribution of vulnerabilities in the IoT. IoT is fundamentally mixed with networks and infrastructure and a total of 88% of analyzed CVEs were potentially relevant for IoT systems.

An estimated half of the vulnerabilities are in the backend and frontend systems while only 10-20% concern the IoT devices.

Protecting HTTP servers, wherever they are, is a priority as they are vulnerability hotspots. Especially beneficial would be the elimination of the command injection vulnerabilities. Many vulnerabilities are in software components used in all IoT subsystems. Tracking these vulnerabilities and timely mitigating new vulnerabilities in them is essential for IoT security. Comparison to other IoT vulnerability studies found common trends, but also notable differences. Categories vary between studies, making comparison difficult. The use of consistent vulnerability categories would be highly desirable.

ACKNOWLEDGMENTS

This work is supported by Finnish Scientific Advisory Board for Defence (MATINE/2500M-0152).

REFERENCES

Caspi, O. (2021). Rapidly evolving IoT malware EnemyBot now targeting Content Management System servers and Android devices. Technical report, AT&T Alien Labs.

Caspi, O. (2022). AT&T Alien Labs finds new Golang malware (BotenaGo) targeting millions of routers and IoT devices with more than 30 exploits. Technical report, AT&T Alien Labs.

Chatzoglou, E., Kambourakis, G., and Smiliotopoulos, C. (2022). Let the cat out of the bag: Popular android iot apps under security scrutiny. *Sensors*, 22:513.

CISA (2022). Known Exploited Vulnerabilities Catalog, Cybersecurity & Infrastructure Security Agency. https://www.cisa.gov/known-exploited-vulnerabilities-catalog.

FIRST (2022). Common Vulnerability Scoring System SIG. https://www.first.org/cvss/.

Google (2022). Google API Client Python library. https://pypi.org/project/google-api-python-client/.

Khoury, R., Vignau, B., Hallé, S., Hamou-Lhadj, A., and Razgallah, A. (2021). An Analysis of the Use of CVEs by IoT Malware. In *Foundations and Practice of Security*, pages 47–62, Cham. Springer International Publishing.

Luptak, G. and Palotay, D. (2021). IoT Botnet Report 2021: Malware and Vulnerabilities Targeted. Technical report, CUJO LLC.

Mathas, C.-M., Vassilakis, C., Kolokotronis, N., Zarakovitis, C. C., and Kourtis, M.-A. (2021). On the design of iot security: Analysis of software vulnerabilities for smart grids. *Energies*, 14(10).

Mirani, S., Meyer, J., Ramgattie, R., and Sindermann, I. (2019). SOHOpelessly Broken 2.0: Security Vulnerabilities in Network Accessible Services. Technical report, Independent Security Evaluators LLC.

MITRE (2022a). Common Vulnerabilities and Exposures Home Page. https://cve.mitre.org/.

MITRE (2022b). Common Weakness Enumeration Home Page. https://cwe.mitre.org/.

NIST (2022a). National Vulnerability Database. https://nvd.nist.gov/.

NIST (2022b). Official Common Platform Enumeration Dictionary. https://nvd.nist.gov/products/cpe.

OWASP (2022). Open Web Application Security Project (OWASP) Foundation. https://owasp.org/.

Sheldon M., R. (2004). *Introduction to Probability and Statistics for Engineers and Scientists.*, volume 3rd ed. Academic Press.

The Chromium Projects (2022). The Chromium Projects Home Page. https://www.chromium.org/.