Long Tail of Security Vulnerabilities and Nation State APT Actors

Daniel Wilusz[1] [0000-0001-8678-6312], Dominik Sadowczyk[2],
Adam Wójtowicz[1] [0000-0003-1276-0915], and Leszek Tasiemski[2]

[1] Department of Information Technology, Poznan University of Economics and Business
{wilusz, awojtow}@kti.ue.poznan.pl
[2] WithSecure Corporation
{dominik.sadowczyk, leszek.tasiemski}@withsecure.com

Abstract. Recently numerous Advanced Persistent Threat groups originating from various countries have been identified, carrying out a wide range of attacks from spear phishing to exploits focused on various entities both commercial and governmental and even military. Many of them exploit zero-day unknown vulnerabilities for which no patch is available, however there are also many cases in which the patch is publicly known and perfectly accessible to the software administrator, but still it is not applied to vulnerable software. This phenomenon is analyzed in the presented work. The list of most commonly exploited vulnerabilities has been cross-referenced with commonly available reports of APT actors' activity, and checked against the raw data from a massively used vulnerability management solution. The authors postulate that APT groups successfully exploit the "long tail" of security vulnerabilities that remain unpatched for months and even years, despite the availability of a fix.

Keywords: cybersecurity, software vulnerabilities, security patch, unpatched vulnerability, vulnerability long tail, advanced persistent threat

1 Introduction

Advanced Persistent Threat (APT) is an adversary with sophisticated levels of expertise, motivation and significant resources often provided by nation states, which allow it to create opportunities to achieve its objectives by using multiple attack vectors [9], [21], [22]. APT is characterized by a repeated and prolonged pursuit of its objectives, adaptation to the target's defense mechanisms, and determination to maintain the level of attack intensity to achieve its objectives [22]. Numerous APTs have been identified originating from various countries [11], [14], [18], [19], [29], [31], carrying out a wide range of attacks from spear phishing to exploits focused on various entities both commercial and governmental and even military [5], [17].

Nation state actors take advantage of vulnerable software to conduct espionage or sabotage operations in cyberspace. Many of them exploit unknown vulnerabilities (so called zero-days), for which no patch is available as even the vendor is not aware of their existence. The serious challenge of zero-days is extensively researched in literature [2], [3], [16], [27], [28]. However, in practice there are many cases in which the patch is publicly known and perfectly accessible to the software administrator, but despite those circumstances it is not applied to vulnerable software [15], [24]. This phenomenon is analyzed in the presented work. The authors of this article postulate that APT groups successfully exploit the "long tail" of security vulnerabilities in systems that remain unpatched for months and even years [13], despite the availability of a fix. In this paper the long tail of security vulnerabilities is defined as a set of well-known vulnerabilities that have not been patched by entities that should mitigate them within the average time taken to fix critical cybersecurity vulnerabilities. The average time taken to fix cybersecurity vulnerability that has been reported in recent sources varies from 200 days [25] to 205 days [33] (256 days for high severity vulnerabilities) and the value of 205 days has been applied to the presented analysis.

While analyzing the phenomenon of long-tail vulnerabilities the following research questions arise:

- What is a specificity of the vulnerabilities that are frequently unpatched in software despite the availability of the patch?
- What is the risk level related to the existence of long-tail vulnerabilities in the context of known APT attacks (CVE severity)?
- What are the main groups of factors that build barriers to applying the patch in time for software administrators?
- What are possible countermeasures or recommendations for various stakeholders that could mitigate the risks?

The rest of the paper is organized as follows. In Section 2 the background for the presented work is described. The research method used in the data analysis of the long tail vulnerabilities and corresponding APTs is presented in Section 3. Section 4 presents and explains the results of the analysis. A discussion of the findings and conclusions are presented in Section 5.

2 Background

In the vulnerability timeline the following key points can be distinguished:

1. Vulnerability is introduced by a vendor or open-source contributor.
2. The vulnerability is detected and confidentially reported to the security organization or to the software vendor.
3. The vulnerability description is published.
4. The vulnerability is patched and the patch is released (in some cases order of the activities 3 and 4 is changed).
5. The patch is deployed by the end user.

Vulnerabilities are introduced into software during the development process as a result of various intentional or unintentional events: bugs in source code, bugs in software libraries, and errors in configuration or testing procedures. Users introduce vulnerabilities to their systems as a result of third-party software installation, its upgrade (new features – new bugs) or installation of custom extensions. Sometimes a vulnerability may stem from improper configuration – like the use of unsafe protocols or weak cryptographic ciphers. Vulnerability identification can occur as a result of testing performed by the testing team, analysis of the source code by the product community, or exploitation attempts performed by external or internal adversaries. It may be a matter of discovering a weakness in a cryptographic procedure, rendering it vulnerable. Once a vulnerability is discovered, it is described and published by security organizations such as National Vulnerability Database (NVD) [23] or MITRE's CVE [20]. Once the specification of the vulnerability is known, software vendors release a patch, and security organizations make preliminary security recommendations. Once the patch is released, system administrators install it on their systems.

The majority of software vendors and security researchers respect the code of "responsible disclosure", where a memorandum on releasing the information is mutually agreed between the vendor and researcher. Thanks to this, the information about the vulnerability is published simultaneously with the fix, minimizing the time when systems are exposed to an attack. In current-day bug bounty programs, responsible disclosure behavior is also motivated by a financial reward to the researcher.

The area of research on security vulnerabilities is dominated by numerous zero-day vulnerability studies [26]. Researchers are working on detecting and preventing zero-day attacks, proposing a range of techniques and frameworks [2], [3], [16], [27], [28]. Kotzias et. al examined the issue of patching delays across tens of millions of client hosts for dozen client-side and over one hundred server-side applications and noted that up to nine months is required to patch 90% of server-side hosts [15]. Sarabi et. al. studied the patching behavior of more than 400 thousand users and found that many hosts stay unpatched even with known and exploited vulnerabilities [24]. Allodi et. al. have developed a theoretical model that proves the following theses: an attacker exploits only one vulnerability for a given software version, frequently chooses vulnerabilities that require low attack complexity, and prepares the exploitation of new vulnerabilities in a slow manner [1].

Despite numerous studies on new vulnerabilities and the problem of patching known vulnerabilities, to the best of our knowledge, there are no studies on the matching of known, often unpatched vulnerabilities (named in this paper as "long tail"), with known APT groups for which history of exploitation of these vulnerabilities has been officially proven.

3 Method

In order to analyze the phenomenon of the long tail of security vulnerabilities, the authors took a list of most commonly exploited vulnerabilities, maintained and published officially by the CISA organization [6], and cross-referenced it with commonly available reports of APT actors' activity [4], [12], [18], [19], [29]. That list of CVEs was then checked against the raw data from the vulnerability management solution [32]. Because of anonymization requirements, the results are shown as a percentage of vulnerable assets out of total active assets known to the product. The asset is defined here as a host or other type of device that has an IP address and is connected to a corporate network. It can be a server, desktop, workstation, network device, printer, mobile (rare case), or IoT device. It is also indicated when the vulnerability was first reported and when it was seen most recently to give insight into a time span and the long tail effect of vulnerabilities.

Because of confidentiality reasons, absolute numbers cannot be used in this analysis, and it operates on percentage values of affected assets in the whole asset pool of the database. It has to be stressed that even if percentage values are low, it means thousands of instances in absolute numbers. Also, the percentage values are calculated in relation to all assets, all architectures, and all operating systems. For instance, missing security updates for Windows affect only assets running this operating system, but the percentage is calculated relative to the size of the full asset base, which obviously includes many non-Windows machines. Lastly, presented data come from the vulnerability management product database [32] which tends to be used by mature organizations, that actively manage their vulnerabilities (SMB and SME organizations; the product is being sold through a channel where partners are usually MSP/MSSP IT companies providing services to their end customers). It is therefore safe to assume that the long tail percentages would look much worse in the general population of global assets. On the other hand, APT actors tend to target mature organizations, hence the target group of APT actors can be compared to a representative sample of vulnerability management product customers.

4 Results

The 200,000 raw data sample obtained on 29. August 2022 from WithSecure(TM) Elements Vulnerability Management solution [32] was filtered to include vulnerabilities that were first reported before 205 days of the survey (which is the average time taken to fix cybersecurity vulnerability [33]). The data was sorted by the percentage of assets affected by the vulnerability. A CVE code or set of codes was then identified for a particular vulnerability (many CVEs can be related to one vulnerability). Then, based on the CISA Known Exploited Vulnerabilities Catalog [6], vulnerabilities have been limited to only those for which an exploit has been reported and finally the list of vulnerabilities has been limited to the top twenty-five. Vulnerabilities with an assigned percentage of affected assets and the time of first reporting are shown in Table 1. Analysis of the data in Table 1 reveals that the top 25 list is dominated by vulnerabilities reported earlier than 3 years since the time of the research. Figure 1 shows a slow decline in the number of vulnerabilities in the top 25 list relative to their first report time.

Table 1. Long tail vulnerabilities based on raw data obtained from solution [32], first reported before the average time taken to fix cybersecurity vulnerability, sorted by the percentage of affected assets (top 25), limited to those for which an exploit has been reported [6], with corresponding CVE codes

VID	Vulnerability	Percentage affected [%]	First time reported	CVE
1	Teamviewer through 14.7.1965 Improper Authentication Vulnerability	1.71	18.02.2020	2019–18988
2	Intel Management Engine Components Privilege Escalation Vulnerability	0.91	22.11.2017	2017–5689
3	Apache Log4j2 Remote Code Execution Vulnerability (Authenticated Check for Windows)	0.82	15.12.2021	2021–44228
4	January 2022 Security Updates	0.65	17.01.2022	2022–21882
5	Remote Desktop Services Remote Code Execution Vulnerability BlueKeep	0.60	01.06.2019	2019–0708
6	Apache Tomcat before 7.0.100, 8.5.51 and 9.0.31 File Inclusion Vulnerability	0.54	27.02.2020	2020–1938
7	October 2021 Security Updates (including remote code execution in MS Office)	0.54	15.10.2021	2021–37976
				2021–37975
8	Samba before 4.10.18, 4.11.13, 4.12.7 Netlogon Elevation of Privilege Vulnerability	0.47	22.09.2020	2020–1472
9	November 2021 Security Updates	0.41	12.11.2021	2021–41379
				2021–42278
				2021–42287
				2021–42292
				2021–42321
10	September 2021 Security Updates	0.41	27.09.2021	2021–38647
				2021–38645
				2021–38648
				2021–38649
				2021–38646
				2021–40444
11	March 2021 Security Updates	0.39	05.03.2021	2021–21193
				2021–27059
				2021–26855
				2021–26857
				2021–26858
				2021–27065
				2021–26411
				2021–21166
12	May 2021 Security Updates	0.38	17.05.2021	2021–31207
				2021–31166
13	October 2017 Security Updates	0.38	24.10.2017	2017–11774
				2017–11826
14	July 2020 Security Updates	0.37	20.07.2020	2020–1350
				2020–1040
				2020–1147

Table 1 (continued). Long tail vulnerabilities based on raw data obtained from solution [32], first reported before the average time taken to fix cybersecurity vulnerability, sorted by the percentage of affected assets (top 25), limited to those for which an exploit has been reported [6], with corresponding CVE codes

VID	Vulnerability	Percentage affected [%]	First time reported	CVE
15	MS17-010: Microsoft Windows SMB remote code execution (WannaCry)	0.33	19.05.2017	2017–0143
16	Apache Tomcat before 7.0.82, 8.0.47, 8.5.23 and 9.0.1 Remote Code Execution Vulnerability	0.32	30.10.2017	2017–12617
17	September 2017 Security Updates	0.31	14.09.2017	2017–8759
18	Oracle Java is Missing June 2013 Critical Patch	0.31	03.06.2014	2013–2465
19	January 2021 Security Updates	0.27	19.01.2021	2021–1647
20	January 2018 Security Updates	0.27	16.01.2018	2018–0798
				2018–0802
21	MS15-081: Vulnerabilities in Microsoft Office Could Allow Remote Code Execution	0.27	14.12.2015	2015–1642
22	Oracle Java is Missing April 2013 Critical Patch	0.26	03.06.2014	2013–2423
23	Oracle Java is Missing February 2013 Critical Patch	0.25	03.06.2014	2013–0431
24	Adobe Reader is Missing APSB13-15 Security Update	0.25	14.12.2015	2013–2729
				2013–3346
25	Adobe Reader is Missing APSB13-07 Security Update	0.25	02.11.2016	2013–0640
				2013–0641

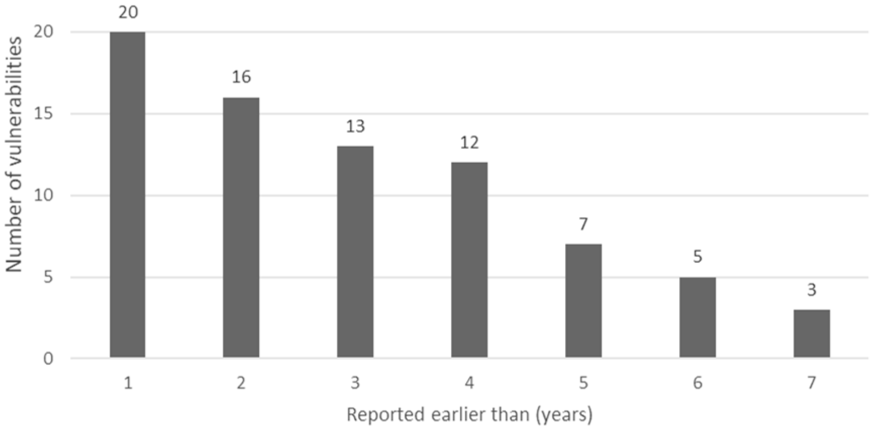

Fig. 1. Number of vulnerabilities older than given number of years

In the next step a base score and a CVSS 2.0 vector have been assigned to each CVE code based on the NIST National Vulnerability Database to facilitate comparative analysis of attributes of long tail vulnerabilities. These data are presented in Table 2. It is worth noting that 24 of the 49 CVE's have a base score greater than 7, which denotes high severity. Moreover, none of the CVEs in Table 2 has a base score indicating low severity. Further analysis of the data in Table 2 shows that for as many as 41 of the 49

CVEs a network is the access vector, 22 CVEs are characterized by low access complexity and as many as 43 CVEs require no authentication. In terms of impact on the attributes of confidentiality, integrity, and availability, the 20 CVEs are characterized by an impact on all three.

Table 2. Top 25 long tail vulnerabilities with CVSS 2.0 vectors assigned [23]

VID	CVE	Base Score	Access Vector	Access Complexity	Authentication	Confidentiality impact	Integrity impact	Availability Impact
1	2019-18988	4,4	Local	Medium	None	Partial	Partial	Partial
2	2017-5689	10	Network	Low	None	Complete	Complete	Complete
3	2021-44228	9,3	Network	Medium	None	Complete	Complete	Complete
4	2022-21882	7,2	Local	Low	None	Complete	Complete	Complete
5	2019-0708	10	Network	Low	None	Complete	Complete	Complete
6	2020-1938	7,5	Network	Low	None	Partial	Partial	Partial
7	2021-37976	4,3	Network	Medium	None	Partial	None	None
	2021-37975	6,8	Network	Medium	None	Partial	Partial	Partial
8	2020-1472	9,3	Network	Medium	None	Complete	Complete	Complete
9	2021-41379	4,6	Local	Low	None	Partial	Partial	Partial
	2021-42278	6,5	Network	Low	Single	Partial	Partial	Partial
	2021-42287	6,5	Network	Low	Single	Partial	Partial	Partial
	2021-42292	6,8	Network	Medium	None	Partial	Partial	Partial
	2021-42321	6,5	Network	Low	Single	Partial	Partial	Partial
10	2021-38647	7,5	Network	Low	None	Partial	Partial	Partial
	2021-38645	4,6	Local	Low	None	Partial	Partial	Partial
	2021-38648	4,6	Local	Low	None	Partial	Partial	Partial
	2021-38649	4,6	Local	Low	None	Partial	Partial	Partial
	2021-38646	6,8	Network	Medium	None	Partial	Partial	Partial
	2021-40444	6,8	Network	Medium	None	Partial	Partial	Partial

Table 2 (continued). Top 25 long tail vulnerabilities with CVSS 2.0 vectors assigned [23]

VID	CVE	Base Score	Access Vector	Access Complexity	Authentication	Confidentiality impact	Integrity impact	Availability Impact
11	2021-21193	6,8	Network	Medium	None	Partial	Partial	Partial
	2021-27059	8,5	Network	Medium	Single	Complete	Complete	Complete
	2021-26855	7,5	Network	Low	None	Partial	Partial	Partial
	2021-26857	6,8	Network	Medium	None	Partial	Partial	Partial
	2021-26858	6,8	Network	Medium	None	Partial	Partial	Partial
	2021-27065	6,8	Network	Medium	None	Partial	Partial	Partial
	2021-26411	5,1	Network	High	None	Partial	Partial	Partial
	2021-21166	6,8	Network	Medium	None	Partial	Partial	Partial
12	2021-31207	6,5	Network	Low	Single	Partial	Partial	Partial
	2021-31166	7,5	Network	Low	None	Partial	Partial	Partial
13	2017-11774	6,8	Network	Medium	None	Partial	Partial	Partial
	2017-11826	9,3	Network	Medium	None	Complete	Complete	Complete
14	2020-1350	10	Network	Low	None	Complete	Complete	Complete
	2020-1040	7,7	Adjacent	Low	Single	Complete	Complete	Complete
	2020-1147	6,8	Network	Medium	None	Partial	Partial	Partial
15	2017-0143	9,3	Network	Medium	None	Complete	Complete	Complete
16	2017-12617	6,8	Network	Medium	None	Partial	Partial	Partial
17	2017-8759	9,3	Network	Medium	None	Complete	Complete	Complete
18	2013-2465	10	Network	Low	None	Complete	Complete	Complete
19	2021-1647	7,2	Local	Low	None	Complete	Complete	Complete
20	2018-0798	9,3	Network	Medium	None	Complete	Complete	Complete
	2018-0802	9,3	Network	Medium	None	Complete	Complete	Complete
21	2015-1642	9,3	Network	Medium	None	Complete	Complete	Complete
22	2013-2423	4,3	Network	Medium	None	None	Partial	None
23	2013-0431	5	Network	Low	None	None	Partial	None
24	2013-2729	10	Network	Low	None	Complete	Complete	Complete
	2013-3346	10	Network	Low	None	Complete	Complete	Complete
25	2013-0640	9,3	Network	Medium	None	Complete	Complete	Complete
	2013-0641	9,3	Network	Medium	None	Complete	Complete	Complete

Table 3. Long tail vulnerabilities cross-referenced with data on APT actors' activity [18], [19], [29], [12], [4]

VID	CVE	CVE category name	CISA TOP	APT	Suspected origin
3	2021-44228	Apache Log4j2 JNDI configuration, log messaging, and parameterization features not protecting against attacker-controlled LDAP and other endpoints	2021	Magic Hound	Iran
				Aquatic Panda	China
				Lazarus	North Korea
				Mercury	Iran
8	2020-1472	Netlogon Elevation of Privilege Vulnerability	2020 2021	FIN7	Russia
				DragonFly	Russia
				Wizard Spider	Russia
				menuPass	China
11	2021-26855	Microsoft Exchange Server Remote Code Execution Vulnerability	2021	HAFNIUM	China
				Threat Group-3390	China
	2021-26857	Microsoft Exchange Server Remote Code Execution Vulnerability	2021	HAFNIUM	China
				Threat Group-3390	China
	2021-26858	Microsoft Exchange Server Remote Code Execution Vulnerability	2021	HAFNIUM	China
				Threat Group-3390	China
	2021-27065	Windows Container Execution Agent Elevation of Privilege Vulnerability	2021	HAFNIUM	China
				Threat Group-3390	China
	2021-26411	Internet Explorer Memory Corruption Vulnerability	–	APT37	North Korea
13	2017-11774	Microsoft Outlook Security Feature Bypass Vulnerability	–	APT 33	Iran
				APT 34	Iran
14	2020-1040	Hyper-V RemoteFX vGPU Remote Code Execution Vulnerability	–	Sandworm Team	Russia
				Kimsuky	North Korea
				APT 28	Russia
				ATP 33	Iran
16	2017-12617	Apache Tomcat JSP Code Injection and Remote Execution Vulnerability	–	Sea Turtle	Iran
17	2017-8759	.NET Framework Remote Code Execution Vulnerability	–	APT-C-01	China
				BlackOasis	Middle East
				Cobalt Group	ND
				Leviathan	China
20	2018-0798	Microsoft Office Memory Corruption Vulnerability	–	Tonto Team	China
				Higaisa	South Korea
				BRONZE BUTLER	China
				Threat Group-3390	China
	2018-0802	Microsoft Office Memory Corruption Vulnerability	–	APT 37	North Korea
				Tonto Team	China
				Confucius	India
				Tropic Trooper	China
				Inception	Russia
				BRONZE BUTLER	China
25	2013-0640	Adobe Reader and Acrobat via a Crafted PDF Document	–	DarkUniverse	ND

Finally, the long tail vulnerability list has been cross-referenced with commonly available reports of APT actors' activity [4], [12], [18], [19], [29] and CISA Top Routinely Exploited Vulnerabilities [7], [8]. The matching is presented in Table 3. It is worth noting that the first 6 CVEs listed in Table 3, are included in the Top Routinely Exploited Vulnerabilities list prepared by CISA. This shows that common, known, unpatched vulnerabilities have not only been actively exploited by APTs, but are directly listed by security organizations as frequently exploited. In other words, organizations are not patching known vulnerabilities, despite widely available warnings of their active exploitation.

Interesting relationships become apparent when the data in Tables 2 and 3 are combined. As many as 13 of the 14 vulnerabilities listed in Table 3 do not require authentication. In addition, 7 of the 14 CVEs have a base score exceeding 9, meaning that they allow all three information security attributes (confidentiality, integrity, availability) to be completely compromised with at least a medium attack complexity. CVEs with a mean base score were also used by APT groups, e.g., CVE-2021-26857, CVE-2021-26858, CVE-2021-27065, CVE-2021-26411, CVE-2017-11774, and CVE-2017-12617. The aforementioned CVEs are characterized by a network attack vector, lack of authentication, and partial violation of confidentiality, integrity, and availability. Thus, even CVEs having only a medium base score can pose a threat.

5 Discussion and Conclusions

The presented analysis shows several widespread vulnerabilities that are a few years old and still run unpatched, despite a fix from the vendor being available. After initial hype and interest, even the most prominent vulnerabilities, such as Log4j (CVE-2021-44228) or ProxyShell (CVE-2021-34473, CVE-2021-34523, and CVE-2021-31207), fade, get forgotten, while vulnerable systems invite attackers to take advantage of unpatched security holes.

The general findings from this work are as follows: (1) based on raw real-world data, a non-negligible percentage of active servers is still affected by long-tail vulnerabilities despite the availability of the patches; (2) these identified long-tail vulnerabilities have a historical record of exploitation by foreign APTs that are described in cybersecurity professional sources. Therefore, (3) there is a significant risk that APTs will take advantage of their expertise and materialize the vulnerabilities into attacks on systems, data or users whenever it becomes beneficial for them. Particularly, APT Groups commonly exploit CVEs that do not require authentication and are accessible remotely from the network. There is one exemption from that rule, namely CVE-2020-1040, which requires access from an adjacent network and authentication with a single factor.

There are several reasons for the long tail of unpatched vulnerabilities. Corporate inertia plays a role. For some organizations, the process of introducing a software patch, especially in a server infrastructure or OT installations is a time consuming and complex endeavor. The reasons are frequently procedural or even compliance related. In this case it is often a "stability over security" imbalance. Some organizations and individuals use software (or hardware-software) solutions that have reached the end of life (EoL), thus they are not supported by the vendor anymore (e.g., Microsoft Windows XP). EoL software continues to function, but does not receive any regular security updates, hence quickly becomes permanently vulnerable. Some vendors may not be aware that their product embeds third-party modules that are vulnerable, especially when it comes to open-source libraries, like Log4j [14]. Without a specialized software updating product, some organizations, especially small ones, may lack visibility of critical vulnerabilities in their environment and lack tools for monitoring the patching progress. Lastly, introducing a patch, unless done fully automatically, introduces additional work, usually for IT staff. It may be forgotten, or other tasks may take priority.

Patching should become a fully automated process, without introducing unnecessary delays for human input (with some exceptions, such as critical infrastructure). The immediate nature of the patching process should correspond with the rapid exploitation of published vulnerabilities. Currently, within hours from releasing the information, cyber criminals start scanning for exposed targets. Protection must catch up with that pace. Nation states, on the other hand, need to be assumed to have knowledge of vulnerabilities months if not years ahead of the general public (zero-days) and ability to exploit them even before the vendor is aware of the bug and able to release a patch.

Governments and policymakers could consider introducing regulations that would mandate vendors to implement automatic patching mechanisms for all internet connected devices. There could be a time span within which critical vulnerabilities in vendor software or any third-party modules employed would have to be patched. Another step could be that the automatic patching mechanism could be impossible to disable for the user. Recent plans [10] made by the EU (Cyber Resilience Act) are a significant step in that direction toward embedded device manufacturers, among other things, requiring them to keep publishing security patches for a period of either five years, or the product expected lifetime, as well as to disclose incident within twenty-four hours of becoming aware of it.

Future work includes a more detailed vulnerability analysis:

- number of vulnerabilities on particular assets;
- coexistence of vulnerabilities on particular assets;
- type and severity of vulnerabilities occurring on particular assets;
- estimation of APT probability and impact.

References

[1] Allodi, L., Massacci, F., & Williams, J. (2021). The work-averse cyberattacker model: Theory and evidence from two million attack signatures. Risk Analysis, 42(8), 1623–1642. https://doi.org/10.1111/risa.13732

[2] Bilge, L., & Dumitras, T. (2012). Before we knew it: An Empirical Study of Zero-Day Attacks In The Real World. Proceedings of the 2012 ACM Conference on Computer and Communications Security – CCS'12. https://doi.org/10.1145/2382196.2382284

[3] Blaise, A., Bouet, M., Conan, V., & Secci, S. (2020). Detection of zero-day attacks: An unsupervised port-based approach. Computer Networks, 180. https://doi.org/10.1016/j.comnet.2020.107391

[4] Bussoletti, F. (2019). Is Iran's apt34 behind the Sea Turtle Cyber Espionage Campaign? Difesa e Sicurezza, https://www.difesaesicurezza.com/en/cyber-en/is-irans-apt34-behind-the-sea-turtle-cyber-espionage-campaign/

[5] Chen, P., Desmet, L., & Huygens, C. (2014). A study on Advanced persistent threats. Advanced Information Systems Engineering, 63–72. https://doi.org/10.1007/978-3-662-44885-4_5

[6] CISA (2022), Known Exploited Vulnerabilities Catalog, https://www.cisa.gov/known-exploited-vulnerabilities-catalog

[7] CISA. (2021). Alert (AA21-209A) Top Routinely Exploited Vulnerabilities. CISA. Retrieved from https://www.cisa.gov/uscert/ncas/alerts/aa21-209a

[8] CISA. (2022). Alert (AA22-117A) 2021 Top Routinely Exploited Vulnerabilities. CISA. Retrieved from https://www.cisa.gov/uscert/ncas/alerts/aa22-117a

[9] Cole, E. (2013). The changing threat. Advanced Persistent Threat, 3–26. https://doi.org/10.1016/b978-1-59-749949-1.00001-2

[10] Dobberstein, L. (2022). EU puts manufacturers on Hook for Smart Device Security. The Register – Biting the hand that feeds IT, https://www.theregister.com/2022/09/16/eu_cyber_resilience_act/

[11] Flashpoint. (2022). Russian apt and ransomware groups: Vulnerabilities and threat actors who exploit them, https://flashpoint.io/blog/vulnerabilities-exploited-by-russian-apts-ransomware-groups/

[12] Fraunhofer-Institut für Kommunikation, Informationsverarbeitung und Ergonomie FKIE. (2022). malpedia. Malpedia, https://malpedia.caad.fkie.fraunhofer.de/

[13] Kaspersky Lab (2022). Targeted attack on industrial enterprises and public institutions, https://ics-cert.kaspersky.com/publications/reports/2022/08/08/targeted-attack-on-industrial-enterprises-and-public-institutions/

[14] Knapczyk, P. (2022). Overview of the cyber weapons used in the Ukraine – russia war. Trustwave, https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/overview-of-the-cyber-weapons-used-in-the-ukraine-russia-war/

[15] Kotzias, P., Bilge, L., Vervier, P.-A., & Caballero, J. (2019). Mind your own business: A longitudinal study of threats and vulnerabilities in enterprises. Proceedings 2019 Network and Distributed System Security Symposium. https://doi.org/10.14722/ndss.2019.23522

[16] Kumar, V., & Sinha, D. (2021). A robust intelligent zero-day cyber-attack detection technique. Complex & Intelligent Systems, 7(5), 2211–2234. https://doi.org/10.1007/s40747-021-00396-9

[17] Lemay, A., Calvet, J., Menet, F., & Fernandez, J. M. (2018). Survey of publicly available reports on Advanced Persistent Threat Actors. Computers & Security, 72, 26–59. https://doi.org/10.1016/j.cose.2017.08.005

[18] Madiant (2022), Advanced Persistent Threats (APTs), https://www.mandiant.com/resources/insights/apt-groups

[19] Mitre (2022), Groups, https://attack.mitre.org/groups/

[20] MITRE. (2022). CVE. Retrieved from https://cve.mitre.org/

[21] Nist (2012). NIST Special Publication 800-30 Revision 1, Guide for Conducting Risk Assessments. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-30r1.pdf

[22] Nist (2012). NIST Special Publication 800-39 Managing Information Security Risk. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-39.pdf

[23] NIST. (2022). National Vulnerability Database. NVD, https://nvd.nist.gov/

[24] Sarabi, A., Zhu, Z., Xiao, C., Liu, M., & Dumitraş, T. (2017). Patch Me if you can: A study on the effects of individual user behavior on the end-host Vulnerability State. International Conference on Passive and Active Network Measurement, 113–125. https://doi.org/10.1007/978-3-319-54328-4_9

[25] Security Magazine (2021), Average time to fix severe vulnerabilities is 256 days, https://www.securitymagazine.com/articles/95929-average-time-to-fix-severe-vulnerabilities-is-256-days

[26] Singh, A.P. (2017). A study on Zero Day malware attack. International Journal of Advanced Research in Computer and Communication Engineering, 6(1), 391–392. https://doi.org/10.17148/ijarcce.2017.6179

[27] Singh, U.K., Joshi, C., & Kanellopoulos, D. (2019). A framework for Zero-day vulnerabilities detection and prioritization. Journal of Information Security and Applications, 46, 164–172. https://doi.org/10.1016/j.jisa.2019.03.011

[28] Singh, U.K., Joshi, C., & Singh, S.K. (2017). Zero day attacks defense technique for protecting system against unknown vulnerabilities. International Journal of Scientific Research in Computer Science and Engineering, 5(1), 13–18. https://isroset.org/pub_paper/IJSRCSE/3-IJSRCSE-VC000412.pdf

[29] Stirparo, P., Bizeul, D., Bell, B., Chang, Z., Esler, J., Bleich, K., ... & Egloff, F. (2019). APT Groups and Operations, https://apt.threattracking.com

[30] Toulas, B. (2022). Microsoft: Iranian hackers still exploiting Log4j Bugs against Israel. BleepingComputer, https://www.bleepingcomputer.com/news/security/microsoft-iranian-hackers-still-exploiting-log4j-bugs-against-israel/

[31] Weedon J. (2015) Beyond ‘Cyber War’: Russia’s use of strategic cyber espionage and information operations in Ukraine. In: Cyber war in perspective: Russian aggression against Ukraine. Talinn: NATO CCD COE Publications. p. 67–77. https://ccdcoe.org/uploads/2018/10/Ch08_CyberWarinPerspective_Weedon.pdf

[32] WithSecure (2022), WithSecure Elements Vulnerability Management, https://www.withsecure.com/us-en/solutions/software-and-services/elements-vulnerability-management

[33] ZDNet (2021), Average time to fix critical cybersecurity vulnerabilities is 205 days: report, https://www.zdnet.com/article/average-time-to-fix-critical-cybersecurity-vulnerabilities-is-205-days-report/