Perspectives and Reviews in the Development and Evolution of the Zero-Day Attacks

Cosmin Alexandru TEODORESCU
Bucharest University of Economic Studies, Romania
cosmin_04@outlook.com

Zero-day attacks are among the most dangerous security incidents affecting both home users and corporate environments. Since 2021 broke the record for zero-day attacks, this study will present the state-of-the-art by clarifying the concept and giving a detailed analysis of the field. Unknown threats such as zero day or unknown malware software usually avoid traditional antivirus or antimalware protection solutions. This type of cyberattack disrupts the activity of companies, causing loss of time and money or compromising confidential data. By their nature, antivirus signatures cannot stop unknown threats. New and old security vendors claim that their "next generation" solutions use signature-based detection based on revolutionary technologies such as machine learning to identify zero-day attacks. Despite captivating stories and tempting words, the effectiveness of these solutions is unscientific, and it is rarely supported by reliable sources. The results of this paper will present the development of different well-known attacks by analyzing their evolution.

Keywords: *Zero-day, Attacks, Vulnerability, Exploit, Security*
DOI: 10.24818/issn14531305/26.2.2022.05

1 Introduction

The zero-day exploit is one of the most feared forms of cybersecurity attack. The phrase "zero-day" is often used in cybersecurity and computer science to describe problems, dangers, and hazards that arise from lack of knowledge, expertise, or misunderstanding. More explicitly, "zero-day" or “never before seen” refers to the fact that the related software developer or vendor had no prior knowledge of the vulnerability in issue and had zero days to patch it before it was exploited [1]. A zero-day attack is a novel vulnerability without known protection; as a result, the attack has a high-risk likelihood and a critical impact. These vulnerabilities are generally so dangerous that they trade for millions of dollars on the dark Web [2].

A zero-day attack requires immediate attention to minimize exposure as much as possible; however, threat actors have typically already exploited it at this point. They are dangerous, since they are unidentified, there is no preliminary data, and threat actors are the only ones who know about them. There are no accessible updates, and antivirus scanners are unable to identify them. As a result, criminals have unrestricted and unauthorized access, or they might damage and even compromise a system.

Zero-day attacks can be classified into two groups, according to the target actor [1]. On the one hand, there are targeted zero-day attacks, where the target is specific and can be in the form of governments and public institutions or senior workers with privileged access to company systems. On the other hand, nontargeted attacks against a significant number of home or business users who utilize a susceptible system, such as an operating system or browser, are common. The objective of most attackers is to infiltrate these systems and use them to create enormous botnets. The WannaCry assault, which leveraged the EternalBlue flaw in the Windows SMB file protocol to compromise over 200,000 devices in a single day, was a recent example [3]. Hardware, firmware, and the Internet of Things can all be targets of nontargeted assaults.

1.1 The distinction between vulnerability, exploit, and attack

The terms vulnerability, exploit, and attack are frequently used in conjunction with the phrase zero-day. Zero-day vulnerability is one that is discovered by security researchers or malicious actors in software, firmware or hardware before the developer or vendor is aware of it. Because the software vendor and the developer are unaware of the vulnerability, no fixes have been issued. Hackers will be able to quickly take advantage of the flaw. Typically, zero-day vulnerabilities are only found after an attack has occurred and a computer forensic and cybersecurity investigation has been completed [4].

A hostile actor uses a zero-day exploit to attack a system with a zero-day vulnerability. Hackers usually design code that allows them to exploit the system.

A zero-day attack is when a zero-day exploit is used to harm a system or steal valuable data from a vulnerable system. Zero-day attacks are often carried out using zero-day malware that includes an exploit. It might take minutes, days, weeks or months for a vulnerability to be identified, let alone fixed, after hackers have initiated an attack.

2 Literature Review

This section presents existing review related works for a zero-day attack. Academic studies indicate that there is an ever-increasing interest in this area.

According to Singh et al. [5], a zero-day vulnerability represents a software or application flaw that the vendor is unaware of or has not yet patched. The work indicates that the goal of the attackers is to steal sensitive information such as enterprise data or legal documents. However, they also present the fact that hackers might have two possibilities: to be able to assist the program manufacturer by sharing details about the detected flaw or to sell the taken data to a black-market broker, who could then resell the exploit at the greatest possible price.

Fagioli [6] claims that zero-day exploits are employed in a wide range of assaults with a broad range of objectives. However, when combined with ransomware, they may be very harmful to unprepared businesses. Many businesses face a binary decision: pay the ransom or lose their data. According to studies, only half of those who pay the ransom get their data back [7], [8], [9].

Several strategies have been developed to defend against zero-day attacks by a large number of researchers. Blaise et al. [10] classify them in two main categories: knowledge-based and anomaly-based techniques, while Aoudni et al. [11] and Singh et al. [5] classify existing security schemes into more categories such as statistical-based, signature-based, behavior-based, and hybrid detection-based techniques [12].

Attack profiles are formed from historical elements in statistical-based methodologies. The profile settings of previous exploits are modified based on those detected aspects, allowing for the identification of assaults [13]. On the other hand, these statistically based solutions cannot be used to defend and detect attacks in real time [14].

The signature-based detection approach develops a library of various malware signatures. Depending on the user's preferences, these signatures are cross-referenced with network files, local files, email, or on-line downloads. These libraries are updated on a regular basis to include new signatures, which are often the signatures of newly exploited vulnerabilities [15]. According to Lobato et al. [16], Snort [17] and Bro [18] are examples of knowledge-based (or signature-based) systems that use a signature database to discover attacks that fit specific patterns, such as harmful byte sequences or known malware signatures.

The behavior-based mechanism identifies important features of worms in order to forecast future behavior of a web server or victim device to prevent unexpected activities [19].

Hybrid-based methods overcome the shortcomings of the strategies mentioned above by combining them in various ways [19], [20]. Kaur and Singh [21] used a hybrid technique to construct a zero-day attack detection system based on this technique.

Kermati [22] presents challenges in assessing the security of zero-day attacks and proposes a unique attack graph-based technique to assess the danger of these attacks. The method used in this research can estimate the risk of unknown attacks considering the impact of known vulnerabilities on the network.

Bilge and Dumitras [23] analyzed field data collected from 11 million Windows hosts over a period of 4 years. The main results of their study showed that 11 of the 18 vulnerabilities discovered were not known zero-day vulnerabilities, meaning that zero-day attacks are serious risks that can cause substantial damage to companies. Also, an important conclusion is the fact that zero-days can persist anywhere from 19 days to 30 months, with an average of about 10 months and a median of 8 months.

3 Evolution of zero-day exploits

In recent years, there has been a series of zero-day attacks and vulnerabilities that have dominated the media and exposed the general public to the risks we face as we share more information online. Lesser known to the public zero-day attacks are:

- Shellshock - A vulnerability located in the Bash command shell and a "perfect 10" Common Vulnerability Scoring System (CVSS) score. It affected the Unix and Linux systems, which are used by thousands of websites, servers, and other systems, putting them at risk.
- Heartbleed - The first big security flaw to receive its own logo and website was an OpenSSL vulnerability that was published in 2014. It was the first exploit with its own "brand" and had the ability to disrupt the Internet faster than any influencer.
- Stagefright - Through an MMS, a series of vulnerabilities could enable remote code execution and privilege escalation on a susceptible Android user device. In 2015, at Black Hat, Stagefright was shown live on stage.
- Meltdown and Spectre - In January 2018, Meltdown and Spectre were revealed. Hardware flaws affected virtually all of today's computer chips, and the resulting chaos created by poor repair and fixes nearly surpassed the exploit's harm.
- F5 BIG-IP – Disclosed in July 2020. The delivery of a single HTTP request to the server responsible for hosting the traffic management user interface allowed executing code on the targeted server.

According to Project Zero [24], 51 exploits were discovered in use in 2021, more than double the figure for 2020 and more than any other year on record. The database is meant to keep track of zero-day exploit cases that have been discovered "in the wild." This signifies that the vulnerability was discovered as a zero-day vulnerability in real-world attacks against users (i.e., not known to the public or the vendor at the time of detection). Information was gathered from a variety of public sources.

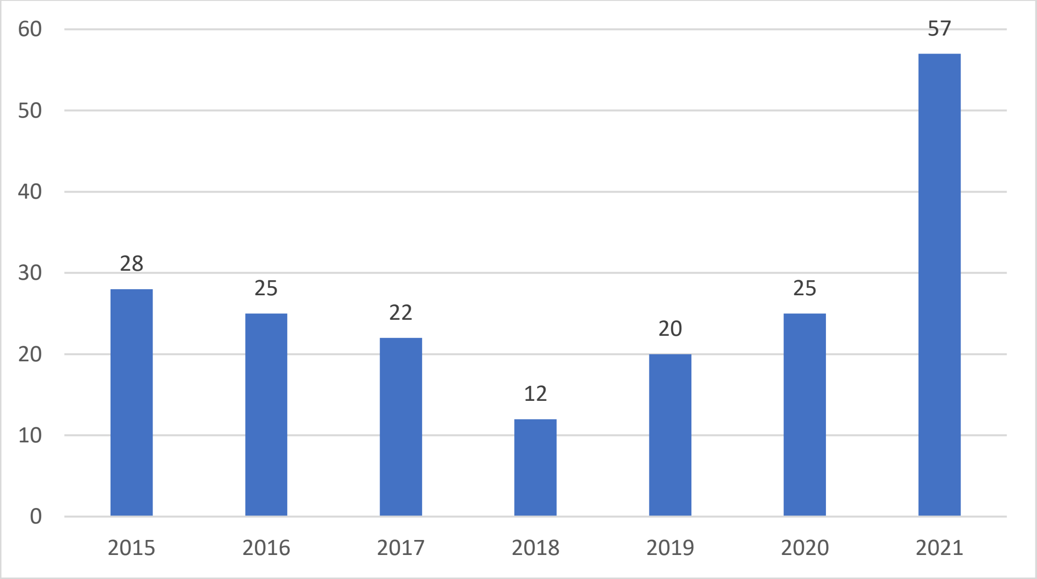


Fig. 1. Evolution of zero-day exploits. Source: [24]

The increasing global expansion of hacking tools is one factor leading to the greater number of reported zero-days. Government-sponsored hackers are at the top of the food chain. In 2021, China is suspected to be responsible for nine zero-day attacks [25]. The United States and its allies definitely have some of the most advanced cyber capabilities, and there is a discussion of deploying those tools more aggressively in the future. Most nations that want to build strong exploits don't have the ability or capacity to do so locally, so they buy them. Purchasing zero-day exploits from the exploding exploit market is now easier than ever [26]. What was formerly unaffordable and high-end is now more commonly available.

One shift in the pattern might be more money available for protection, thanks to the greater bug bounty and awards offered by computer companies for discovering new zero-day vulnerabilities. There are, however, better tools available. Defenders have progressed from being able to identify only relatively simple assaults to being able to identify more complicated hacks. Large-scale detection attempts are carried out by companies like Microsoft and CrowdStrike. Whereas older techniques, such as antivirus protection, meant fewer eyes on unusual behavior, today a big corporation may detect little anomalies across millions of workstations and track them back to the zero-day exploit.

4 Development of well-known zero-day attacks

The impact and the eventual outcome of a zero-day depend on multiple aspects, such as detection tools, the affected target, the entity that finds it, and many other factors. These will adjust the difficulty of each unique scenario. However, the architecture of a zero-day from an evolutive perspective is quite similar. From my personal perspective and analysis, there is a common scenario with six major phases that threat actors pursue to carry out the attack, as depicted in Figure 1. These stages can be performed in any sequence and may be repeated several times.

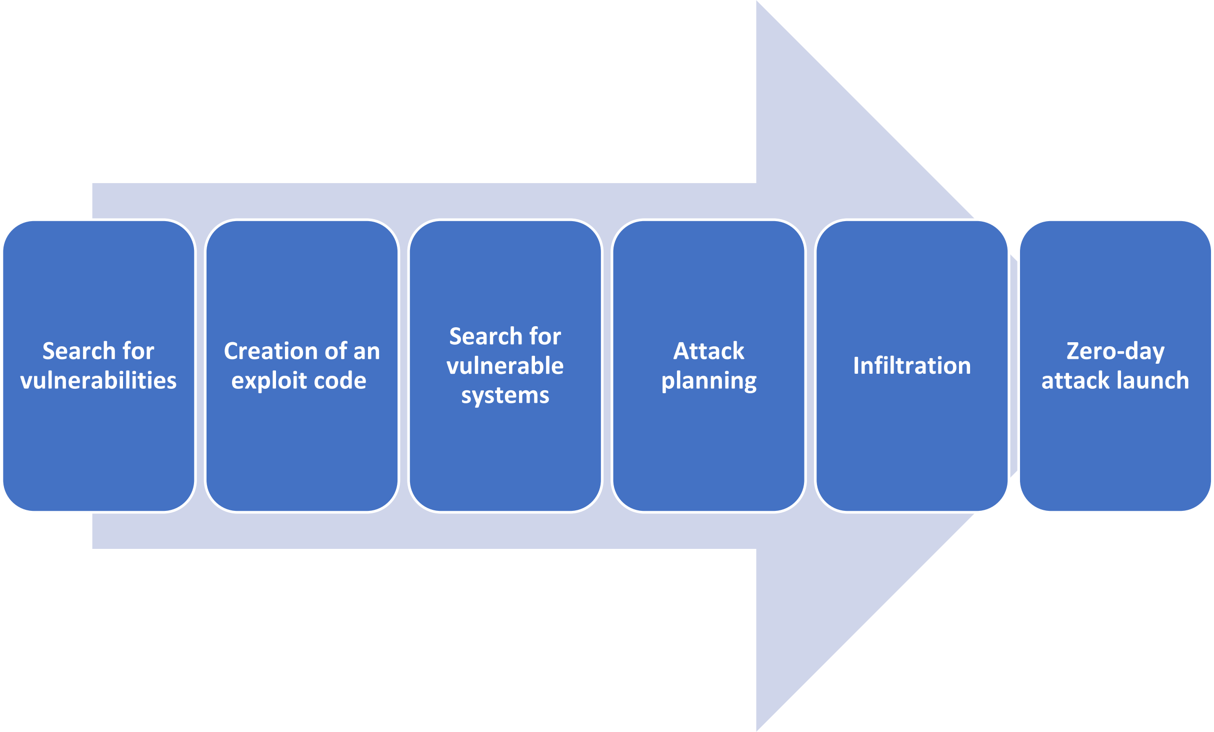

Search for vulnerabilities
Creation of an exploit code
Search for vulnerable systems
Attack planning
Infiltration
Zero-day attack launch

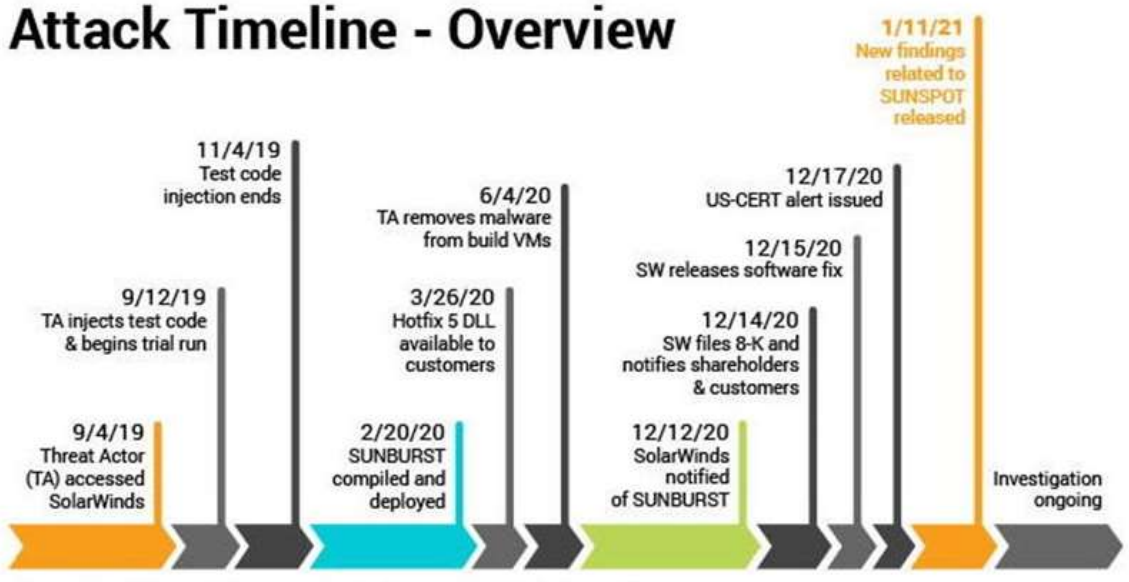
Fig. 2. Development of a zero-day attack

The following section describes and explains two of the most severe exploit-based attacks known. The first case refers to one of the most dangerous cyber-espionage incidents, due to the high profile of targets and the continuous pursuit for threat actors to explore vulnerabilities in order to gain access to the victim and its clients, on a classical supply chain attack scenario. The second incident represents a rare example of a zero-day attack that could have widely affected Microsoft Windows users in the 2000s. Each of these attacks is analyzed on the basis of the architecture presented above.

4.1 SolarWinds attacks

On 8 December 2020, FireEye announced a breach carried out by a possible state-sponsored threat actor. The target of the attack, FireEye, represents a major cybersecurity company with clients ranging from major governments to highly valuable enterprises. December 14, 2020 was the day when Washington Post published an article in order to report that FireEye was among the dozen victims of the hack [27]. According to the article, the breach was part of a broad espionage campaign carried out by Russian state actors known as APT29 or Cozy Bear, later grouped as Nobelium by Microsoft [28]. The method used was the insertion of malicious code into the Orion Platform, which represents an IT performance monitoring system with privileged access to clients in order to obtain logs and system performance data [29]. Hackers therefore performed a supply chain attack in order to target a third party with access to an organization's systems rather than attacking the specific network directly. Judging by the timeline of the events, I assess that the date when the attack was revealed represented the initial point for threat actors to start the search for vulnerabilities, as well as the beginning of the exploit code building process.

The access to targeted systems was gained via trojanized software updates. Adversaries used Orion software updates that were distributed between March and June 2020 in order to plant the Sunburst malware [30] on the target's servers [29]. However, as later observed by the investigators, initial access to SolarWinds systems was obtained in September 2019, when the Sunspot malware was deployed on the SolarWinds build server [31], representing a type of developer software used to assemble small components into large software applications. Investigations showed that once a build command was performed, the malware would silently replace source code inside the Orion app with files that loaded the Sunburst malware, resulting in versions of the Orion app that carried malicious software.


Attack Timeline - Overview
9/4/19
Threat Actor (TA) accessed SolarWinds
9/12/19
TA injects test code & begins trial run
11/4/19
Test code injection ends
2/20/20
SUNBURST compiled and deployed
3/26/20
Hotfix 5 DLL available to customers
6/4/20
TA removes malware from build VMs
12/12/20
SolarWinds notified of SUNBURST
12/14/20
SW files 8-K and notifies shareholders & customers
12/15/20
SW releases software fix
12/17/20
US-CERT alert issued
1/11/21
New findings related to SUNSPOT released
Investigation ongoing

All events, dates & times approximate and subject to change pending completed investigation

Fig. 3. Timeline of the SolarWinds supply chain attack. Source: [32]

Although it was suspected that the initial malware code and the associated attack came from a Russian-based threat actor, other nation-state cyber actors have also used SolarWinds to launch attacks [29]. In the same period of time the Sunburst attack launched, suspected nation-state hackers based in China exploited SolarWinds by using a different malware identified as Supernova. The malicious software is based on two components: a Web shell, an unsigned Windows.dll file build to appear as legitimate code on Orion, and an exploit that uses an API authentication bypass flaw (CVE-2020-10148) used in order to run the Web shell [33]. The vulnerability allows commands to be run without being authenticated to the API, aspect used by the Supernova attackers in order to install the Web shell into the targeted Orion software running on victims internal severs [34]. Similar intrusions on the same network and the lack of a digital signature led researchers from Palo Alto Networks [35] and Secureworks to admit that the Spiral threat group, suspected of Chinese origin, is to blame for the intrusions [36]. This represented the point where the creation of exploit codes and infiltration of the target have already been capitalized. The next steps, according to Fig. 2 were to restart the work for an exploit and to look for a way to penetrate systems with a zero-day exploit this time.

According to my representation, all phases of the attack architecture were already completed in July 2021, when the launch of a zero-day attack was discovered by threat hunters at Microsoft, a fact that showed that the interest in exploiting SolarWinds was still ongoing. The malicious code was deployed using a zero-day vulnerability in the Orion Platform. Tracked as CVE-2021-35211, the vulnerability lies in the SolarWinds Serv-U product, which was used by customers in order to transfer files across networks. Exposing the Serv-U SSH to the Internet gave attackers the ability to run malware code with high system privileges, performing therefore an attack called remote code execution. From this point on, attackers are able to install and run malicious payloads as well as view, change, modify, or delete specific data [37]. Attributing the intrusions with high confidence to the Chinese DEV-0322 (short for "Development Group 0322") based on observed victimology, tactics, techniques, and procedures (TTPs), Microsoft Threat Intelligence Center (MSTIC) mentioned that the adversary is known for targeting entities in the US defense sector and software companies [38]. Regarding the same origin for threat actors exploiting SolarWinds systems, I assess that several phases of the zero-day attack development were shared and repeated, in order to develop a better exploit, capitalized as the latter zero-day attack. Therefore, threat groups might have shared knowledge regarding vulnerable systems, infiltration techniques, and means to carry out the final attack, according to Fig. 2. Overall, the continuous chain of attacks performed on SolarWinds systems and against company clients ranged from supply chain attacks to the exploration of vulnerabilities that led to zero-day software. The evolution of the interest manifested by nation-state hackers highlighted the fragility of modern networks and the sophistication of threat actors in their run to identify hard-to-find vulnerabilities in widely used software, conduct espionage, and extract data from targeted systems.

4.2 WebDAV exploit

On 10 March 2003 [39], unknown hackers accessed an undisclosed number of US Army Web servers, exploiting a previously unknown buffer overflow vulnerability (CAN-2003-0109) located in a section of Microsoft's Windows 2000 OS responsible of managing the Web Distributed Authoring and Versioning (WebDAV) protocol. WebDAV comes installed by default with Internet Information Server (IIS) Version 5.0 and allows documents to be assigned attributes and properties, thus enabling collaborative creation, editing, and searching from remote locations. Another particular aspect is that it also enables the writing of documents via HTTP. Therefore, if a threat actor is able to run malicious code with local privileges on a specific vulnerable system, the attacker would be able to take complete control of the system, including the ability to alter data, install programs, or create fully privileged accounts.

Sources involved in the investigation [40] mentioned that investigators were told about the vulnerability two days after the attack took place and there was no information that could certify the fact that any of US Army systems had been compromised. Administrators observed that the exploit was constantly performing network mapping as well as exfiltrating data from the terminal services port - port 3389 - to an unspecified region, but to the same destination continuously. Investigators mentioned that the use of a nonstandard port was likely the attempt to stay below the security radar since it is normally used for encrypted traffic that sniffers would not attempt to decipher.

According to my representation of a zero-day attack development phases, the results of the investigation signaled that the attacker did not plan the attack according to a specific exfiltration or damage-related purpose and may have skipped the build-up of a specific plan. Microsoft Windows contains a dynamic link library (DLL) named ntdll.dll. This specific DLL represents a core OS component used to interact with the Windows kernel. However, the buffer overflow vulnerability in ntdll.dll is used by many different Windows components. The WebDAV component of Microsoft IIS 5.0 is an example of a specific operating system software that uses ntdll.dll in order to process incoming WebDAV requests. However, because the vulnerable Win32 API component is being used by many other applications, it is possible that other exploit vectors exist, given the fact that the vulnerability could have been exploited in a more advanced manner. Therefore, I assess that the attacker went swiftly through the search and build phases of the exploit, and the found of the US Army target was improper addressed, as otherwise there could have been more damage done to the servers. A similar conclusion is drawn by the fact that Army sources mentioned that a file found on the hard drive of one of the affected servers contained the phrase "welcome to the Unicorn beachhead" [41].

According to Symantec Corp., in 2003 Microsoft IIS was estimated to run on about 25% of the Internet's Web servers, which represented approximately 4 million vulnerable systems [42]. Judging that the WebDAV attack yields full administrative privileges, it was assessed that if this vulnerability were to be coded into a worm, similar to CodeRed [39], [43], massive damage to computing systems worldwide could occur. However, no such worm has surfaced, and this may be due to the fact that the attack relies on the function of brute forcing the stack of remote machines, which would significantly slow down the worm. In a threat analysis book scenario [44], it took 12 iterations before command line access was obtained. These twelve attempts occurred in 3 minutes, which is a long time in the world of worms. For a worm to be successful, it must have a good scanning engine, a random number generator, and, most importantly, a small and quick attack vector.

Therefore, the described attack stresses the importance of going through all of the phases needed for a proper development of a zero-day attack, while missing a phase could ease investigators work. A proof-of-concept [45] developed 2 weeks after the attack emphasizes the basic actions needed to build the exploit and sets standards for one of the most exploited vulnerabilities from the Windows Kernel [46].

5 Conclusions

No operating system or software application is completely safe; they are created by people, and humans make mistakes. In this regard, security is critical and continuous upgrades are required in order to address new vulnerabilities. Incidents which represent zero-day attacks have been analyzed for many years; however, no investigation has yet measured the prevalence and the duration of these attacks in a real-world scenario, unless the disclosure of the corresponding vulnerabilities. Over the last decade, there has been an increase in the usage of zero-day exploits. Threat actors need more zero-day exploits to maintain own capabilities, fact that reflects increased cost to attackers related to security measures that cover known vulnerabilities. Therefore, the increasing demand for such capabilities and the ecosystem that supplies them is a greater challenge. Zero-day software used to be only tools for nation-state actors who owned the proper technical expertise to discover zero-day vulnerabilities, turn them into exploits, and then strategically put them into use. Starting mid to late 2010s, the number of private companies who joined the marketplace in order to sell these specific zero-day capabilities increased. However, nowadays groups no longer need to have technical skills; now they only need resources. Most of the zero-days that Google Threat Analysis Group (TAG) has discovered in 2021 [47] fall into this category: first developed by exploit brokers and then sold to and used by government-backed actors.

However, improvements made in detection and an increasing culture of disclosure contribute to the significant increase in zero-days detected in 2021 compared to 2020, thus reflecting more positive trends. The industry that protects users from zero-day attacks has long suspected that, overall, cybersecurity mechanisms detect only a small percentage of the zero-days that are actually being used. Therefore, the increasing detection of zero-day exploits is a positive aspect which allows getting vulnerabilities fixed, protecting users, and giving specialists that work in the field a wider picture of the ongoing exploitation in order to build and make more informed decisions on how to prevent and mitigate it.

References

[1] Mukherjee, A. (2022) *A Closer Look at The Zero-Day*. Threat Intelligence. Available online: https://www.threatintelligence.com/zero-day

[2] Digital Shadows. (2021). *Vulnerability Intelligence: What's The Word In Dark Web Forums?* Photon Research Team. Available online: https://www.digitalshadows.com/blog-and-research/vulnerability-intelligence-whats-the-word-in-dark-web-forums/

[3] Cynet. n.d. *Zero-Day Attacks, Exploits, and Vulnerabilities: A Complete Guide.* Available online: https://www.cynet.com/zero-day-attacks/zero-day-vulnerabilities-exploits-and-attacks-a-complete-glossary/

[4] Kaspersky. n.d. *What is a Zero-day Attack? - Definition and Explanation.* Available online: https://www.kaspersky.com/resource-center/definitions/zero-day-exploit

[5] Singh, U.K., Joshi, C., Kanellopoulos, D., A framework for zero-day vulnerabilities detection and prioritization, Journal of Information Security and Applications, Volume 46, 2019, Pages 164-172, ISSN 2214-2126, https://doi.org/10.1016/j.jisa.2019.03.011 .

[6] Fagioli, A. (2019). Zero-day recovery: the key to mitigating the ransomware threat. Computer Fraud & Security, 2019(1), 6–9. doi:10.1016/s1361-3723(19)30006-5

[7] Cyberthreat Defense Report. 2018. Cyberedge Group. Available online. https://cyber-edge.com/wp-content/uploads/2018/03/CyberEdge-2018-CDR.pdf.

[8] Sophos. 2021. The State of Ransomware 2021 Report. Available online: https://secure2.sophos.com/en-us/content/state-of-ransomware

[9] Kaspersky. 2021. Available online: https://www.kaspersky.com/about/press-releases/2021_over-half-of-ransomware-victims-pay-the-ransom-but-only-a-quarter-see-their-full-data-returned

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

[11] Aoudni, J., Donald, C., Farouk, A., Sahay, K.B., Babu, D.V., Tripathi, V., Dhabliya, D., Cloud Security Based Attack Detection Using Transductive Learning Integrated With Hidden Markov Model, Pattern Recognition Letters

(2022), doi: https://doi.org/10.1016/j.patrec.2022.02.01

[12] Joshi, C., Singh, U.K., Kanellopoulos, D. (2018). An enhanced framework for identification and risks assessment of zero-day vulnerabilities. International Journal of Applied Engineering Research, Vol. 13, No. 12, pp. 10861–10870.

[13] Kim, J. Y., Bu, S. J., & Cho, S. B. (2018). Zero-day malware detection using transferred generative adversarial networks based on deep autoencoders. Information Sciences, 460, 83-102.

[14] Kaur, S., & Singh, M. (2015). A proactive framework for automatic detection of zero-day HTTP attacks on educational institutions. Computer Fraud & Security, 2015(2), 10-16.

[15] Holm, H. (2014, January). Signature based intrusion detection for zero-day attacks:(not) a closed chapter?. In 2014 47th Hawaii international conference on system sciences (pp. 4895-4904). IEEE.

[16] A.G.P. Lobato, M.A. Lopez, I.J. Sanz, A.A. Cardenas, O.C.M.B. Duarte, G. Pujolle, An adaptive real-time architecture for zero-day threat detection, in: IEEE International Conference on Communications (ICC), 2018, doi: 10.1109/icc.2018.8422622

[17] Cisco, Snort - network intrusion detection & prevention system, 2018. Online: https://www.snort.org/

[18] V. Paxson, Bro: a system for detecting network intruders in real-time, Comput. Netw. 31 (23–24) (Dec 1999) 2435–2463, doi: 10.1016/s1389-1286(99)00112-7.

[19] Hammarberg, D. (2014). The best defenses against zero-day exploits for various-sized organizations. SANS Institute InfoSec Reading Room, 21.

[20] Kaur R, Singh M. 2014 a. Efficient hybrid technique for detecting zero-day polymorphic worms. In: Advance computing conference (IACC). 2014 IEEE International Feb; 2014. p. 95–100. 21-22.

[21] Kaur R, Singh M. 2014 b. A survey on zeroday polymorphic worm detection techniques. IEEE Commun. Surv. Tutor. 2014;16(3):1520–49

[22] Keramati, M. (2016). An attack graph based procedure for risk estimation of zeroday attacks. In 2016 8th International Symposium on Telecommunications (IST), pp. 723–728. IEEE.

[23] Bilge, L., & Dumitras, T. (2012). *Before we knew it.* Proceedings of the 2012 ACM Conference on Computer and Communications Security - CCS ’12. DOI:10.1145/2382196.2382284

[24] Project Zero. (2022). Available online: https://googleprojectzero.blogspot.com/p/0day.html

[25] Sadowski, J. (2021). *Zero Tolerance: More Zero-Days Exploited in 2021 Than Ever Before.* Mandiant Threat Intelligence. Available online: https://www.mandiant.com/resources/zero-days-exploited-2021

[26] O'Neill, P.H. (2021). *2021 has broken the record for zero-day hacking attacks.* MIT Technology Review. Available online: https://www.technologyreview.com/2021/09/23/1036140/2021-record-zero-day-hacks-reasons/

[27] Washington Post. (2020). Russian government hackers are behind a broad espionage campaign that has compromised U.S. agencies, including Treasury and Commerce. Available online: https://www.washingtonpost.com/national-security/russian-government-spies-are-behind-a-broad-hacking-campaign-that-has-breached-us-agencies-and-a-top-cyber-firm/2020/12/13/d5a53b88-3d7d-11eb-9453-fc36ba051781_story.html

[28] Lambert, J. (2021). The hunt for NOBELIUM, the most sophisticated nation-state attack in history. Microsoft Security. Available at: https://www.microsoft.com/security/blog/2021/11/10/the-hunt-for-nobelium-the-most-sophisticated-nation-state-attack-in-history/

[29] Oladimeji, S., Kerner, S.M. (2021). SolarWinds hack explained: Everything

you need to know. Techtarget. Available at: https://www.techtarget.com/whatis/feature/SolarWinds-hack-explained-Everything-you-need-to-know

[30] Eckels, S., Smith, J., Ballenthin, W. (2020). SUNBURST Additional Technical Details. Mandiant. Available at: https://www.mandiant.com/resources/sunburst-additional-technical-details

[31] CrowdStrike Intelligence Team. (2021). SUNSPOT: An Implant in the Build Process. Crowdstrike Blog. Available at: https://www.crowdstrike.com/blog/sunspot-malware-technical-analysis/

[32] Ramakrishna, S. (2021). *New Findings From Our Investigation of SUNBURST*. SolarWinds, Orangematter. Available at: https://orangematter.solarwinds.com/2021/01/11/new-findings-from-our-investigation-of-sunburst/

[33] Higgins, K. J. (2021). What We Know (and Don't Know) So Far About the 'Supernova' SolarWinds Attack. DarkReading. Available at: https://www.darkreading.com/attacks-breaches/what-we-know-(and-dont-know)-so-far-about-the-supernova-solarwinds-attack-/d/d-id/1340513?_mc=rss_x_drr_edt_aud_dr_x_x-rss-simple)

[34] National Vulnerability Database. (2020). CVE-2020-10148 Detail. Information Technology Laboratory. US Government. Available at: https://nvd.nist.gov/vuln/detail/CVE-2020-10148

[35] Tennis, M. (2020). SUPERNOVA: A Novel .NET Webshell. Unit 42, Palo Alto Networks. Available at: https://unit42.paloaltonetworks.com/solarstorm-supernova/

[36] Secureworks CTU Research Team. (2021). SUPERNOVA Web Shell Deployment Linked to SPIRAL Threat Group. SecureWorks. Available at: https://www.secureworks.com/blog/supernova-web-shell-deployment-linked-to-spiral-threat-group

[37] SolarWinds. (2021). Serv-U Remote Memory Escape Vulnerability. Available at: https://www.solarwinds.com/trust-center/security-advisories/cve-2021-35211

[38] Microsoft Threat Intelligence Center. (2021). Microsoft discovers threat actor targeting SolarWinds Serv-U software with 0-day exploit. Microsoft Security. Available at: https://www.microsoft.com/security/blog/2021/07/13/microsoft-discovers-threat-actor-targeting-solarwinds-serv-u-software-with-0-day-exploit/

[39] Bennett, A. (2003). US Army hacked with IIS vulnerability, worm expected soon. Computerworld. Available at: https://www.computerworld.com/article/2805974/us-army-hacked-with-iis-vulnerability--worm-expected-soon.html

[40] Verton, D., Sliwa, C. (2003). Military Investigates System Intrusion Involving Windows 2000 Security Flaw. Computerworld. Vol. 37. March 24. Pp. 14.

[41] Krill, P. (2022). Microsoft .NET MAUI framework arrives. ARN. IDG Communications. Available at: https://www.arnnet.com.au/article/698494/microsoft-net-maui-framework-arrives/

[42] Verton, D. (2003). US Army Web servers hacked. Computerworld. Available at: https://www.computerworld.com/article/2580922/u-s--army-web-servers-hacked.html

[43] Trend Micro. (2002). Enterprise Prevention and Management of Mixed-Threat Attacks: Why Current Antivirus and Content Security Approaches are Limited. White Paper. Available at: https://www.biz.netvigator.com/chi/pdf/eps_whitepaper.pdf

[44] SANS Institute. (2003). WebDAV: The new nemesis of IIS Administrators. Global Information Assurance Certification Paper. Available at:

https://www.giac.org/paper/gcih/499/webdav-nemesis-iis-administrators/105490

[45] Medina, R., Hernandez, H. (2003). IIS 5.0 WebDAV -Proof of concept. Bug: CAN-2003-0109. Available at: http://www.rs-labs.com/exploitsntools/rs_iis.c

[46] Dekel, K., Ronen, R. (2020). Case Study: Why You Shouldn't Trust NTDLL from Kernel Image Load Callbacks. Sentinel LABS. Available at: https://www.sentinelone.com/labs/case-study-why-you-shouldnt-trust-ntdll-from-kernel-image-load-callbacks/

[47] Lecigne, C., Resell, C. (2022). Protecting Android users from 0-Day attacks. THREAT ANALYSIS GROUP. Google. Available at: https://blog.google/threat-analysis-group/protecting-android-users-from-0-day-attacks/

Cosmin Alexandru TEODORESCU (Ph.D. candidate) is a cybersecurity specialist with an experience of over 5 years in the field. His qualification includes completing and gaining credentials for multiple cybersecurity-related courses, such as CISCO's Certified Network Associate and Cybersecurity Operations Associate, CompTIA's Linux+ and Security+, as well as EC Council's CPENT. His work focuses in investigating cybersecurity incidents, testing systems for vulnerabilities, as well as assessing security measures and developing best practices in the field.