LATIN-AMERICAN JOURNAL OF COMPUTING (LAJC), Vol VIII, Issue 1, January 2021

Ataques Zero-day: Despliegue y evolución

Zero-day attack: Deployment and evolution

ARTICLE HISTORY

Received 01 October 2020
Accepted 02 November 2020

Xavier Riofrío
Departamento de Eléctrica Electrónica y Telecomunicaciones
Universidad de Cuenca
Cuenca, Ecuador
xavier.riofrio@ucuenca.edu.ec

Fabián Astudillo-Salinas
Departamento de Eléctrica Electrónica y Telecomunicaciones
Universidad de Cuenca
Cuenca, Ecuador
fabian.astudillos@ucuenca.edu.ec

Luis Tello-Oquendo
College of Engineering
Universidad Nacional de Chimborazo
Riobamba, Ecuador
luis.tello@unach.edu.ec

Jorge Merchan-Lima
Departamento de Eléctrica Electrónica y Telecomunicaciones
Universidad de Cuenca
Cuenca, Ecuador
jorge.merchanl@ucuenca.edu.ec

Zero-day attack: Deployment and evolution

Ataques Zero-day: Despliegue y evolución

Xavier Riofrío
Departamento de
Eléctrica Electrónica y
Telecomunicaciones
Universidad de Cuenca
Cuenca, Ecuador
xavier.riofrio@ucuenca.edu.ec

Fabián Astudillo-Salinas
Departamento de
Eléctrica Electrónica
y Telecomunicaciones
Universidad de Cuenca
Cuenca, Ecuador fabian.
astudillos@ucuenca.edu.ec

Luis Tello-Oquendo
College of Engineering
Universidad Nacional de
Chimborazo
Riobamba, Ecuador
luis.tello@unach.edu.ec

Jorge Merchan-Lima
Departamento de Eléctrica
Electrónica y Telecomunicaciones
Universidad de Cuenca
Cuenca, Ecuador
jorge.merchanl@ucuenca.edu.ec

Resumen— En la ciberseguridad y la informática, el término "Zero-day" se relaciona comúnmente con problemas, amenazas y peligros, esto debido a la falta de conocimiento, experiencia o malentendidos relacionados. Un ataque de Zero-day se considera generalmente una nueva vulnerabilidad sin defensa; por lo tanto, el ataque consecuente tendrá una alta probabilidad de riesgo, y un impacto crítico. Lamentablemente, sólo unos pocos estudios están disponibles para comprender estas amenazas, y no bastan para construir nuevas soluciones para detectar, prevenir y mitigar estas dificultades. En este artículo, se presenta una revisión del ataque Zero-day, enfocándose en comprender su impacto real y algunas soluciones accesibles hoy en día. Este estudio presenta una referencia útil que proporciona a los investigadores conocimientos para comprender el problema actual relacionado con los ataques Zero-day. Este puede ser un punto de partida para desarrollar soluciones para combatir este problema.

Palabras clave— Zero-day, vulnerabilidad, ataque, impacto, implementación.

Abstract— In cybersecurity and computer science, the term “zero-day” is commonly related to troubles, threats, and hazards due to the lack of knowledge, experience, or misunderstanding. A zero-day attack is generally considered a new vulnerability with no defense; thus, the possible attack will have a high risk probability, and a critical impact. Unfortunately, only a few surveys on the topic are available that would help understand these threats, which are not enough to construct new solutions to detect, prevent, and mitigate them. In this paper, it is conducted a review of the zero-day attack, how to understand its real impact, and a few different accessible solutions nowadays. This study introduces a useful reference that provides researchers with knowledge to understand the current problem concerning zero-days attacks; hence they could develop solutions for facing them.

Keywords— Zero-day, vulnerability, attack, impact, deploy- ment.

I. INTRODUCTION

No operating system or software is entirely secure, humans develop them, and humans often make mistakes. In this sense, security is essential and constant updates are needed to cover emerging susceptibilities. These software holes are known as vulnerabilities; they can also result from misconfigurations or errors in the code, which create problems that could be exploited by several actors, such as cybercriminals, competitors, ethical hackers, or malicious people.

Zero-days are undiscovered vulnerabilities; this term was used initially for developers with “zero days” to fix a re- spective vulnerability. It demands their attention as urgent as possible, trying to avoid exposure as much as probable; although, usually at this stage, threat actors (hackers) have already taken advantage of it. They are dangerous because they are unknown, there are no preliminary data available, and these vulnerabilities are only known by threat actors. There are no updates available and no anti-virus scanners can detect them. Therefore, criminals are free to gain access through computer assets, getting benefits without

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2021

obstruction. Software with these bugs could be trendy, such as Microsoft systems, adobe software, or even security products as firewalls. It becomes even more critical and complex to control within web systems because they are built using several components or libraries from different vendors. It is a challenge to manage the different versions of these components or libraries and patch them; for example, a web application can be developed in Angular JS, which is a set of libraries in JavaScript with their modules or external add-ons (unknown sources) for complementary functions. In this case, the attack surface becomes enormous and it will be unmanageable for the development team to reduce the damage [1].

The zero-day term could be referred to as diverse ideas in the same context. Firstly, zero-day vulnerability refers to the software being exposed and further indicates that neither software ownership nor security products such as antivirus scanners knew its existence. Second, zero-day exploit refers to threat actors who have developed code or performs an action for this zero-day vulnerability to directly affects assets; generally is developed by the person who finds the fault. However, it could be exploited with negative or positive intentions; section II-C1 gives more details about the stakeholders. Finally, a zero-day attack consists of the direct abuse of a particular computer, application, system, or data, taking advantage of the zero-day vulnerability through a zero-day exploit. The latter represents the final objective of the two previous definitions [2], [3].

In general, the zero-day vulnerabilities are a problem with an underestimated impact. This problem is not considered extremely important for ordinary users because companies receive bug reports (or find their bugs) and just patch them. They minimize their errors, do not disclose related data, and avoid disclosing details as feasible. The reason to do this is to hinder cyber-criminals attention so that they do not take advantage of the exposure. Nevertheless, this will not prevent it from being exploited; usually, the exploit appears on the same day as announced, demonstrating that obscurantism is not a significant obstacle to threat actors. It is worth noting that an exploit is a malicious code that abuses flaws in software to infect, interrupt, or control a computer without the user’s consent and usually without their knowledge [4]. Furthermore, little analysis has been made of the real-life phases of the difficulties related to zero-day, which contributes to the fact that those in charge of computers are not seriously focused on addressing these issues [5].

In the following, two examples to illustrate zero-day attacks are described. The first one is Stuxnet; it was a type of zero-day attack and used as a digital weapon (a pioneer in this domain). This malware is classified as a virus/worm and was addressed at the uranium enrichment plant’s computers in Iran. It exploited five zero-day vulnerabilities to spread and gain privileged access to the systems. Microsoft patched one of the vulnerabilities on time; nevertheless, the Microsoft patch was not enough; criminals attacked, took control of the computers altering the plants’ settings, and achieved to shut down the nuclear plant. Although this happened in 2010, these vulnerabilities are still a threat today, especially CVE- 2010-25681, which is Windows Server 2003 vulnerability [7], [8].

The second is F5 BIG-IP, a modern one, which tries to demonstrate the problem in present days. This zero-day attack was disclosed in July 2020 and is a Remote Code Execution (RCE) vulnerability, which affects each product related to the BIG-IP for the company F5 Networks. This allows executing code in the vulnerable server by sending a specifically single HTTP request to the server hosting the traffic management user interface. The relevant role of the attack is the extensive vulnerable surface; the software is widely used around the world, and according to SHODAN2 there are more than 31000 recognizable devices of this type, as illustrated in Figure 1. This indicates that all of those are potentially vulnerable and need to be patched. Several authors developed exploits immediately; in less than one day, they were spread beyond the Internet (Twitter, Reddit, blogs, among others), demonstrating that this zero-day vulnerability could be exploited without too much knowledge. This example shows that a security hole could be exploited in hours, representing a significant threat to the valuables with critical impact [9], [10].

The concern about zero-day dangers is authentic and real. Researchers have focused on countering the problems and creating solutions considering the victim’s rapid reaction to minimize or disappear the risks presented by those vulnerabilities. Despite this, the most significant challenge when developing solutions is a lack of practical and concrete information; this is needed to test and find errors. Another limitation is the extremely low probability of finding a new bug; it takes millions of files to find a unique vulnerability; besides, false positives must be controlled. These reasons demonstrate the concern of a laboratory for investigating this issue because it allows us to have a better understanding of

how the attack is carried out and how it should be prevented and detected [2].
The main contributions of this study are the following:

(i) Creating a concrete and straightforward source of information to begin the understanding of what are the zero-day attacks.
(ii) Revealing the impact and defining the life cycle that could have an attack of this type.
(iii) Analyzing and comparing solutions existing nowadays to face these attacks.

The rest of this paper is organized as follows. Section II describes the real impact of these attacks in real environments; this is based on collected data from computers in use over the Internet. On the other hand, in Section III, some approaches that exist as countermeasures to prevent, detect, and mitigate this predicament are described. Section IV provides a general discussion of the zero-day, their current countermeasures solutions, and a brief analysis of the open issues and challenges that could be addressed in future works. Finally, conclusions and future work are explained in Section V.

II.ANALYSIS OF A ZERO-DAY

The effect and impact of a zero-day depend on the mode of detection, the affected product, who finds it, and other factors. These will mutate the difficulty depending on each unique scenario. This section explains some of the critical circumstances to analyze and consider a zero-day vulnerability's real impact. First, the deployment cycle of a zero-day attack is introduced; then, its lifetime cycle is explained; finally, the real-life impact of this attacks on several factors is discussed.

A.Zero-day deployment cycle

The deployment cycle for a zero-day attack could vary from each case. However, it is considered a common scenario with two significant phases that threat actors follow to proceed with the attack, as shown in Figure 2. The next steps are performed for white and black hat hackers to abuse weaknesses. These steps could be in a different order and may go through multiple repetitions [7], [12]. The term white and black hat hackers is a categorization where the principles of a hacker are focused. Both groups usually have extensive knowledge of how to break into applications, computer networks, and bypass security protocols. Black hat hackers can be involved in cyber espionage, terrorism, or just for challenging cybercrimes. Their primary motivation is financial, and they are responsible for writing malware and exploits. On the opposite, white hat hackers use their skills for the right team, called "ethical hackers" sometimes earn money for reporting bugs to the official sources [13].

1)Discovery phase: The goal of this phase is to find a zero-day vulnerability. The threat actor attempts to recognize, observe, detect, and even guess possible vulnerabilities of a respective surface target. Thus, with a clear idea of how the target is built and structured, the threat actor could audit and inspect it to determine a specific flaw and then move on to a triage stage to test their ideas and findings to generate an inherent exploit. In the following, it is presented each of the stages of this phase.

(i) Recognition: The initial action of exploitation is to discover what can be vulnerable, finding components to start searching defects issues. While more elements found, more chances of finding a security flaw. Therefore, security researchers typically use tools that help them search for these elements in an automated and agile method such a fuzzers or subdomain listers. However, they do not discard manual analysis that provides a more advanced strategy and adds the ability to go deeper into hidden vulnerabilities [14].

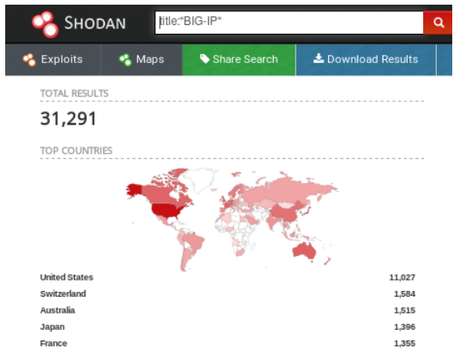


Figure 1: Number of BIG-IP devices connected and found in the internet [11].

(ii) Audit: The next action is to start looking for vulnerabilities in the components beforehand found. In this position, diverse operations may be performed; a threat actor can start analyzing the code directly, performing a binary analysis or reviewing the business logic. Although companies try to hide them, they will be accessible through disassembling tools. As previously affirmed, humans make mistakes, and the code is written for humans; consequently, it is a method to find their errors. Besides, other techniques are applied, such as binary analysis, fuzzing methods that consist of sending a set of payloads until finding one that harms the objective and performing a logical review of the application or system operation. The payload is part of an exploit, which is the portion of the code not related to propagation nor concealment, i.e., it is in charge of performing the malicious action. From the threat actor's point of view, this is where he takes advantage of the system [15].

(iii) Triage: This process involves identifying, tracking, and determining the root cause of the fault in the code; specifically, the part of the application code is being vulnerable and will be exploited. The reason is to exploit the error in the most optimal approach and have a significant and more harmful impact. In this step, there is usually a difference between white hat hackers, who will stop at one point not to damage critical assets, while black hat hackers will continue as much damage as possible because their goal is precisely that.

2)Exploitation phase: In this second phase, "Exploitation", researchers take their vulnerability found apriori and create a sufficiently functional exploit. For that, with the verdicts previously found, it starts to develop a potential exploit. It would help if they debugged with different techniques and types of attacks to take advantage of this flaw. Once the exploit is identified, the developed focus on their effectiveness and efficiency. Finally, concluding with the deployment of the PoC (Proof of Concept)3 in a real environment and with the risk of being compromised.

(i) Debug: At this point, the individual who found the vulner- ability should evaluate the techniques and approaches to exploit it and make the exploit effective. Once evaluated, the threat actor determines precisely what can do exploiting it, the potential impact, and other requirements needed to reduce the uncertainty and create a fully functional exploit. It is possible to return to previous phases, principally if it does not impact or find more complementary vulnerabilities. Generally, exploitation will consist of multiple vulnerabilities, such as Stuxnet's example, and each will contribute to the optimal exploitation.

(ii) Exploit: Once the accurate method is identified jointly with how and what will be exploited, it is necessary to begin applying and testing them, determining their effectiveness, and reviewing several scenarios to confirm the PoC. Initially, it will be a simple exploit, but as it is developed, it could increase the impact. For instance, to escalate privileges or automate actions. There is also the option to cover tracks and cleanup footprint to avoid an effortless discovery of this zero-day.
(iii) Deploy: Finally, the cycle continues to push this zero-day exploit in a real environment; notwithstanding secure laboratory variations, the real world may imply extra obstacles. For black hackers, they can do the damage to the compromised systems or sell it on the black market. However, for white hat hackers, it is up to them to create a tangible and real PoC or report the flaw directly to the corresponding entity. This topic is discussed in Section II-C2.

It is worth emphasizing that every zero-day vulnerability does not convert to a zero-day attack. Sometimes security issues do not lead to exploitable vulnerabilities, do not present a real impact, or they are identified for the company before someone can exploit them.

B.Lifetime cycle

Generally, a widespread belief is that a zero-day is working in the background for a short amount of time because vendors mitigate and release patches as soon as they appear, but this is not always the case. Furthermore, it is believed by the IT community that after being disclosed, this vulnerability will become obsolete or at least with a lower frequency of use [5]. This segment will answer these and other related myths, aiming to detail the authentic lifetime cycle of a zero-day, from its initial discovery until it is supposed dead.

The first point to answer is when a vulnerability appears in a production environment; due to improperly tested or ignored issues. Then, once in production, this error will be exposed for an indeterminate period, as shown in Figure 3. Besides, it will end when the vendor officially releases the patch, consequently going from being exposed indeterminately to being exposed for out-of-date software [16].

Nevertheless, the exposure time is not equal such as a zero-day attack sequence. When the vulnerability is found, criminals develop it, and the vulnerability will be exploited until official sources publish the respective CVE. Nevertheless, while criminals are abusing in

1Common Vulnerabilities and Exposures (CVE) is a document in a database with extensive information detailing vulnerabilities, technical issues, and the disclosure dates; this is a standard used and accepted for academia, govern- mental organizations, private developers, and the cyber-security industry [6]

2A search engine specialized in Internet-connected devices, used for secu- rity people to look for assets connected to the web

3A PoC in the cybersecurity field is used for demonstrating that an exploit is possible on an established system, which is an initial proof because sometimes it is not necessary to attack the objective; testers solely require to prove their idea [14].

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2021

this time interval, the vulnerability should be discovered by official sources or security researchers (company consulting team, bug hunters, or other community members). Finally, when the zero-day is disclosed publicly, security vendors such as antivirus scanners should develop their solution to find these new threats [17].

It should be highlighted that these actions frequently do not always occur in the corresponding order as Figure 3, although there is always an exposure time before the vulnerability disclosed publicly, and the patch released is always later or equal to this date.

Nonetheless, what happens after its disclosure? To have an answer, it is necessary to know when a vulnerability indeed dies. It is thought that it is dead when the patch is released, either because there is not enough information about its longevity or due to the fact that providers do not release this data for security reasons. However, the study of [7] shows that exploits have an average lifetime of 6.9 years, some of which remain active for more than ten years, new versions continually appear; therefore, they are considered 'immortal'. It also demonstrates that these zero-days will consider as a short lifetime cycle whether they have 1.51 years or less, but this occurs in hardly a quarter of the data analyzed. On the contrast, the longevity group will live more than 9.53 years, representing a 25 percent more longlasting exploits.

C.Real life impact

It has been already talked about attacks and how they are carried out, but is the impact visible and serious in real life?. Several factors influence in this topic, the next arguments are the most significant and are referenced to Figure 4.

1)Stakeholders in the zero-day surface: [13]. Security researchers concentrate on finding zero-day vulnerabilities and report them to the provider, sometimes for money and sometimes for fame. Usually, these vulnerabilities are released into the public domain through published vulnerability advisories, blogs, and news. Software vendors have their security team, but in most cases, this is not enough, then they launch an external consultancy for researchers.

These businesses are speedily expanding; programs such a HackerOne, BugCrown, or Vulnscope (for Latin America) pay independent researchers (called bug hunters) to find vulnerabilities in private programs. They do not exploit a full zero-day attack, instead of they develop a very basic PoC exploit for it and get a payment. In this way, vendors will have an external extensive security team that brings excellent results and they could offer more reliable products [14].

Lastly, enterprises specialized in zero-day such as Exodus Intelligence, ZDI and iDefense find these attacks and provide data for their subscribers to use for defensive testing and product protection measures. These groups belong to the white group.
In contrast, for the dark side, the stakeholders are nations, cybercriminals, competitors, or hobbyists with another motivation. They will sell their findings privately in different markets [13].

2)Zero day markets: In recent years, zero-day vulnerability markets have been growing exponentially and are divided by the buyer, the public vs. private, the vulnerability's nature, and the threat's objective. The subsequent categorization will focus on these points.
The first is a white market, used for bug hunters to report the found vulnerabilities over to the affected vendor. They use them for defensive purposes such as new patches or improvement of new versions. It depends on the vendors, whether should be disclosed or remains private.

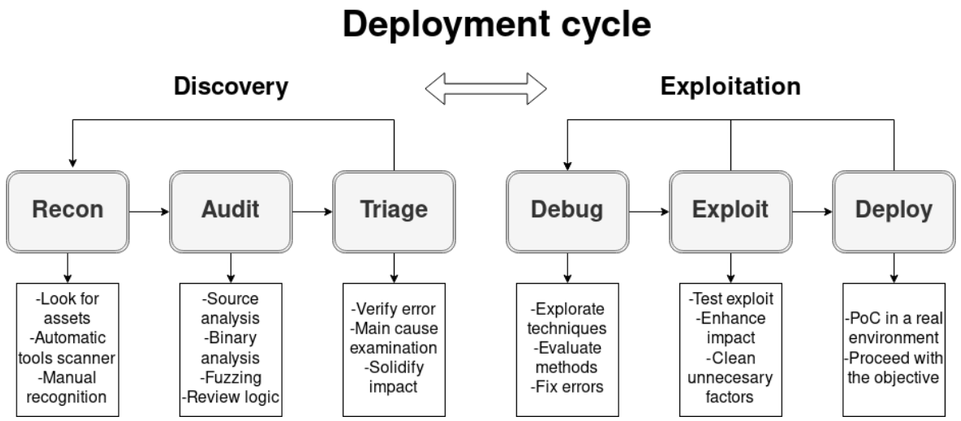


Figure 2: The zero-day attack deployment cycle.

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2021

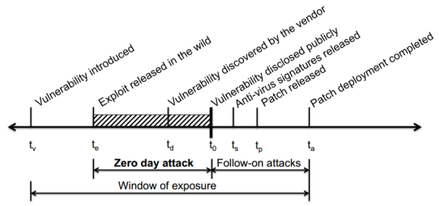


Figure 3: The time line for a zero-day attack [5].

Next is a grey market, vulnerabilities continue to be confidential among initiators and collaborators, which are used primarily for offensive attacks rather than defensive. They will remains in the background, used against the affected provider, although they are usually sold towards governments or national institutions which could use them in diverse purposes, for this reasons they shall not be divulged.

Finally, black markets are sold for criminals where the vulnerabilities are not disclosed because they will use them for illicit purposes. The buyers could be competitors vendors, cyber-terrorists or even nations. This market is the most profitable market because it is illegal and may pay large amounts of money for exploits capable of damaging an organization [13].

3)Evolution of the zero-day: The paper has been discussed the development of a zero-day attack and its lifetime cycle, without specifying what indeed occurs and how changing in these stages in a wild scenery. Here arises a point of inflection where the exploitation rises exaggeratedly, this point is after its disclosure: Zero-day vulnerabilities before disclosure regularly rises and remains running in a context such as the black market members, security researchers, or small groups of hackers.

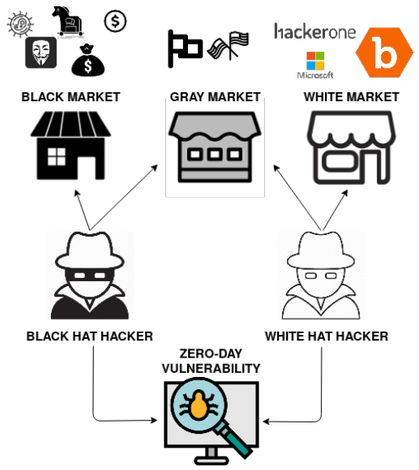


Figure 4: Markets and stakeholders related with zero-days.

Each group wants to avoid discovered their goal is to take advantage of the vulnerability as much as possible (for white and black hat hackers). Therefore, the number of attacks at this stage is low for two reasons: the number of threat actors is low, and the number of targets is limited. Most zero-day attacks do not exceed 1000 attempts, as shown in Figure 5. This point is directly proportional to how is the exploit evolution and variations (Figure 6); the malware remains hidden and continues without threat; therefore, it has no problem attacking, do not need to change to be effective.

On the other hand, once the flaw is publicly exposed, ***Zero-day vulnerabilities after disclosure*** increased logarithmically. This fact is produced because whether a system vulnerability is revealed or widespread, each actor in this environment will have the possibility of exploiting that (they indeed would attack). Although no extensive information is revealed in the CVEs, exploits and attacking methods are immediately developed. Figure 6 demonstrates that malware variations also present a logarithmic growth. Consequently, victims of this vulnerability are more exposed at this point and have a significant probability of being attacked, representing an extreme boosting number of attacks after t0. This behavior is exhibited in Figure 5.

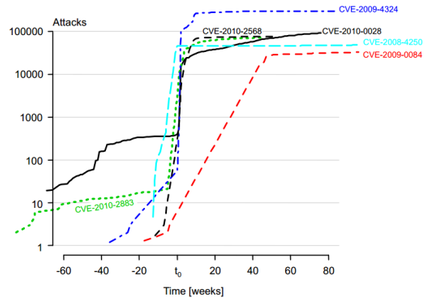


Figure 5: Number of attacks before and after the CVE disclosure. t0 is the disclosure date [16].

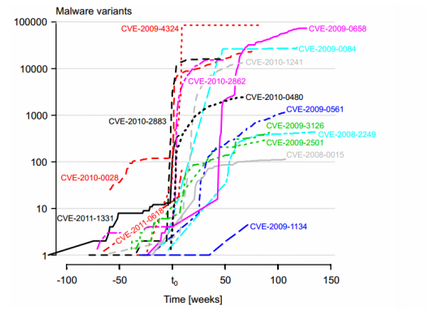


Figure 6: Number of exploit variations before and after the CVE disclosure. t0 is the disclosure date [16].

In this segment, the dangerousness of a zero-day was exposed, in both cases, before and after its disclosure. To emphasize, after the CVE (or another method) is published, it is just a matter of time for someone develops an exploit and spread over the web, facilitating the exploitation of a resource and increasing the attack rate. This amount of attacks will increment and last over time until patches and security solutions will be implemented. Furthermore, this would remain for a considerable amount of time due to the exploit evolution and variations with the same CVE.

III. COUNTERMEASURES AND TECHNIQUES

Security mitigation and countermeasure perform an essential role in the exploitability of a vulnerability. Exploitable vulnerabilities and affected services can be retained or at least deferred over time. There are different ways to counteract the direct influence on the threat implications of zero-day attacks. Standards such as ISO 27000 or NIST4 have various approaches to dealing with computer issues. In the case of ISO 27001, section A-12 refers to "Detection, prevention and mitigation controls to protect against malware shall be implemented, combined with appropriate user awareness" which is also applicable for the case of zero-day [18]. The explanation of them is below, followed by contemporary examples in Section IV-A.

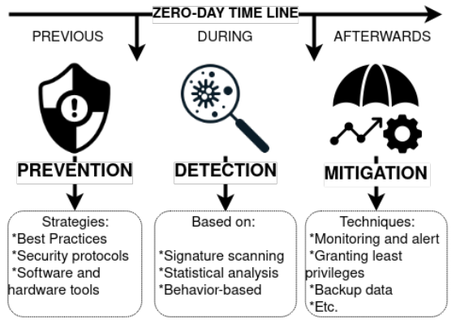


Figure 7: Countermeasures details for zero-days expressed in a timeline.

A.Detection

Any feature with Internet access is exposed to zero-day attack, but antivirus scanners, Internet Detection System (IDS), and other mechanisms are commonly employed to detect possible threats in a system or organization. Nevertheless, these do not operate for new vulnerabilities; some strategies have been developed to dispense or to limit the damage caused by this problem. Figure 7 illustrates that they act during the attack and use the methods explained as follows:

(i) Signaturebased: It is a method regularly employed by antivirus scanners and attempts to discover nonpolymorphic malware (worms and viruses). A signature is a unique part of an exploit; it is usually a string with an impossible date or a hash value. Anti-viruses scanners and security software have signatures databases that are used to find malware [4], [15].

A database contains the signatures previously found in old files, and these are compared with new data to find evolved malware and harmful files. For zero-day, a payload comparison of old malware is carried out. As previously mentioned, a zero-day is exploited with several vulnerabilities, but these are not necessarily zero-day. These libraries are continually updated. The comparison procedure depends on the algorithm used. For instance, in [19], an approach based on the decoding and encoding of the requests received (used as a signature) is presented; these requests are analyzed by a neural network to detect anomalies in the requested parameters.

(ii) Statistical-based: Statistic techniques are one of the most popularly used methods for detecting difficulties in a network, software or unknown intruder. This technique saves computer records of occurred events (conditions, performances, network or memory statistics, among others) that left a trace in the past and compares them with a new running record. If a record does not match, an alert is triggered when different stats are discovered, and they are blocked whether it is necessary. Consequently, if more data would gather, the detection algorithm will be more reliable and more precise.
In the case of zero-day, it can be contradictory because there is no preliminary data. Nevertheless, the method can be applied with procedures based on statistics profiles from system's data to detect new actions in an identical scenario. For example, in [12] a probabilistic approach is presented to detect abnormal patterns in a network, uncovering these dilemmas.

(iii) Behavior-based: They are based directly on the conduct of a system or application, trying to forecast an expected behavior to detect anomalies, whether it would differ or not. Behavior-based strategies use unusual marks that can leave an attack, but the problem with this method is that a large number of false positives (or false negatives) may occur. For this field, it could be beneficial to rely only on the regular conduct of the defended system because these vulnerabilities would not exist in a database. The methodology proposed in [20] aims to analyze flows in real-time, finding anomalies that may be zero-day attacks, but as mentioned above, they have a large number of false positives/negatives anomalies that can be counterproductive.

4The National Institute of Standards and Technology from the United States

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2021

B.Prevention

In business, zero day vulnerabilities can be chaotic, causing severe consequences for companies. Although this has already been mentioned and sounding repetitive,decisions and security efforts must be applied to prevent zero-day attacks (among others).

If an entity has a well established and conscious security structure and policies, a zero day attack is likely to be considerably less damaging. On the other hand, not having defenses will open up the possibility of unfortunate accidents with economic, structural and reputation effects [21]. Consequently, it is indispensable to take into account the probability of occurrence and common safety recommendations. This countermeasure is performed previous the assault, and below is a brief explanation of strategies to prevent them:

(i) Security best practices manuals and resources: The standards are studied, created, and imposed for the prevention of failures that cause problems; for this explanation, it is necessary to apply these standards for preventing.

(ii) Security patching and updates: The new versions of the software are bringing for a reason; vendors should know their vulnerabilities and updates can appropriately protect the system from zero-day exploitation.

(iii) Advance security hardware and software: Secure programming, Quality Assurances (QA), and other measures to regulate the responsibility of programmers are not enough. Therefore, it is necessary to use tools that complement this and reduce zero-day attacks, for example, secure code scanners could be used.

(iv) Security protocols: Inventing and creating something that already exists is entirely insecure; the protocols are tested and approved for several years, entities should not implement algorithms or methods that are easy to implement. For example, the cryptographic algorithms. Generally, they are ordinary security operations of an organization; hence there are not many approaches that perform prevention in this field. However, in [21], they perform a risk assessment for these threats focusing on the method of attack graph-based security metric, which analyzes the risks quantitatively, examining access vector, access complexity, among others factors. This solution contributes to having a risk control for unknown vulnerabilities, preventing and reducing the impact on an organization.

On the other hand, as future work, we will propose creating a tool for the prevention of zero-days attacks that will focus on day zero (disclosure day) to control and reduce the risk on the vulnerable assets of an organization.

C.Mitigation

Having a zero-day issue will not last just for a day, an organization can suffer from these attacks and not find a solution (an official patch) for a considerable amount of time, as mentioned in II-B. In this time gap, damage mitigation is needed; thus, security flaws may be less damaging or nonexistent until a solution would be available. Therefore, the zero-day mitigation approach will focus on the point where it begins until the point where an official patch is provided, in Figure 3 these points are represented from t_a to t_e.

To apply mitigation into the real world, different methodologies and standards can pursue that apply measures for different environments. In this aspect, the leading best practices should be implemented: monitoring the behavior of a resource, granting least privileges, only relying on verified sources, using white lists, and finally having backup measures in case of data loss.

An example that applies mitigation to zero-day is found in [22]. They propose an approach using a critical data sharing protocol in the scenarios with a potential zero-day threat, evaluating the risk that can categorize them to establish a level of confidence. In this process, the case of a zero-day attack is to guarantee that the least important data will be exposed, expecting an early detection that would not compromise more critical assets.

IV. DISCUSSION

The full zero-day cycle explains the whole process involved, the attacker's mind, and the interests behind these types of attacks. The complete cycle is not short, as is generally believed; it takes time to find a vulnerable point and, consequently, develop an exploit for these attacks. Stakeholders are involved

in this branch not only because of damaging or accessing susceptible systems. Behind it, the main interest is the money, as analyzed in section II-C2. Vulnerability markets move a massive amount of capital, regardless of the legal or illegal team.

It is not simple to join this world; researchers require a great set of skills and experiences to develop a zero-day attack. It is also essential an adequate computing power and resources for performing actions at this level and analyzing the different factors that a system may have for a potential flaw. Although this becomes more feasible today through virtualization and other solutions, it is a fact that the resulting attack change in a real environment, then it is necessary to perform tests and attacks on existing real assets. Thus, the threat actors must have security measures to hide his identity and conceal their attack (even more critical) because if the zero-day vulnerability is detected, it will become public and patched.

The life extension of a zero-day exploit is much longer than commonly thought. For this reason, in IT management, it could be necessary to take certain precautions respecting to vulnerabilities. Having out of date systems might remain dangerous for up to 10 years after the zero-day exploit has been launched (and patched). That is why it would be convenient to have a tool which could alert as soon as possible when a zero-day appears, and whether possible assets vulnerable are present to a new potential attack. In the next section are presented the existing solutions.

A.Existing solutions

Currently, exceptional studies exist that try to counteract the predicament of zero-day attacks; each individual is attempting to resolve this problem in a particular direction by concentrating on its objective with diverse methodologies. In Table I, some of the contemporary studies are presented, identifying their main aspects and approaches.

Firstly, Table I shows that most studies are focused on the network environment, and the main objective protected will be through it. Here, distinct techniques and mechanisms are used for distinct solutions; however, only [20] tries to give a real-time solution, which should be the most effective.

Furthermore, it reflects that the most of approaches propose to perform a zero-day detection([12], [19], [23], [24], [25], 26]); this is logical because if it could detect 100% of them, zero-days would cease to exist. However, it is not possible due to this vulnerability's nature; consequently, it is essential to think about other containment methods. Finally, this table shows that different detection strategies are employed as a solution; most of them were mentioned in III-A.

A single prevention method is used, which focuses on analyzing the risks that certain assets may offer to find preventive methods that follow specific standards. On the other hand, there are a few more solutions for mitigation ([22], [27], [28]), and it is essential to highlight the solution presented in [28], which includes two countermeasures in one, detection and mitigation. They decide to detect zero-day but are aware that their tools could fail, then they propose to have a mitigation mechanism through reliable protocols and different treatment to avoid having more vulnerable data exposed.

To conclude, there are limited approaches related to web attacks; this is "Tang2020" focuses on WAFs, which tries to detect web-only vulnerabilities such as SQLI (SQL injection), XSS (Cross-Site-Scripting), RCE (Remote Code Execution), among others. In the future, zero-day web-based solutions will be necessary to develop, as all assets are currently being moved to the cloud systems and related operations.

B.Open issues and research challenges

The main open issues are web applications, cloud computing, virtualization, and others omitted in common zero-day studies. The reason is that defending against cyberattack's surface represents a fundamental challenge, and the main issue is recognizing the point of attacks and the system vulnerabilities that cybercriminals could exploit [29]. These are the current trends and they are growing exponentially; that is why it is imperative to start studying these areas. However, massive companies related to this field have their research programs and their bug bounty programs, but this may not be enough in the real world because threat actors are constantly innovating and finding new ways of exploiting them, resulting in critical future problems that have not been analyzed yet.

Since 2017, an exponential increase in ransomware5 has risen, where different types of zero-day vulnerabilities have been exploited to create this malware. The losses are millionaires affecting companies as large as small, and almost 50 percent of these attacks end up with organizations losing

5A ransomware is considered a type of malware that implements cryptog- raphy to harm data from a device. It encrypts the victim's data with a secret key to block access from a genuine user [15].

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2021

Table I: Various measures to counteract a zero-day attack.

Comparison of multiple countermeasures projects related to zero-day

Model	Countermeasure	Oriented to	Mechanism or Technique	Year
Towards probabilistic identification of zero-day attack paths [12]	Detection	Networks	* Networks based data * Statistical-based * Compute probabilities with a Bayesian network	2017
ZeroWall: Detecting Zero-Day Web Attacks through Encoder-Decoder Recurrent Neural Networks [22]	Detection	Web Application Firewalls (WAFs)	* Signature-based * Semantic patters in requested parameters	2020
The Performance of Machine and Deep Learning Classifiers in Detecting Zero-Day Vulnerabilities [23]	Detection	Networks Operating system	* 34 Machine/deep learning classifiers	2019
An Adaptive Real-Time Architecture for Zero-Day Threat Detection [20]	Detection	Honeypots in the networks	* Behavior-based * Real-time processing and classification	2018
Repids: A multi tier real-time payload-based intrusion detection system [24]	Detection	Networks	* Behavior-based * Reliable data * Low false alarms	2018
An Attack Graph Based Procedure for Risk Estimation of Zero-Day Attacks [21]	Prevention	Networks	* Attack Graph Based Security Metric * Using standards and good practices * Risk analysis	2016
LISABETH: automated content-based signature generator for zero-day polymorphic worms [25]	Detection	Malware Worms	* Devising algorithms * Signature and content based * Low false alarms	2008
A case study of unknown attack detection against zero-day worm in the honeynet environment [26]	Detection	Honeypot in networks Worms	* Traffic monitoring * Polymorphic recognition * Signature generation and based	2009
A framework for mitigating zero-day attacks in IoT [22]	Mitigation	IoT Networks	* Identified critical data * Reliable protocols	2018
Cyber resilience recovery model to combat zero-day malware attacks [27]	Mitigation	Networks	* Intrusion detection methods * Incident rate control * Using NIST SP-800-61 standard	2016
A Consensus Framework for Reliability and Mitigation of Zero-Day Attacks in IoT [28]	Detection Mitigation	IoT Networks	* Signature-based * Reliable data * Context behavior	2017

their data or paying criminals to recover them [30]. This problem results in a research challenge for controlling the future. Therefore, it is fundamental to consider this situation to mitigate and detect these growing and critical threats. future. Therefore, it is fundamental to consider this situation to mitigate and detect these growing and critical threats.
Deep learning and related methods usually require massive data, which means a powerful capacity of processing and other high computing characteristics. Consequently, trying to detect an anomaly when an attack is running could be useless or unlikely. New strategic approaches should be considered to defend and counterattack because traditional tactical strategies would not work in the future, such as an IDS with an approach in both detection and prevention capabilities [31].

V. CONCLUSIONS AND FUTURE WORK

Zero-day vulnerabilities and attacks can be highly critical for indefinite fields of computing. They operate a malicious behavior before the disclosure day, but can continue after their patch or until other solutions are released. Although the human factor may be the fault factor, prevention and mitigation could minimize the risk and avoid these issues.

This paper presented a detailed study of how a zero-day behaves and operates, from discovering the vulnerability to the attack performed in a real environment. This learning process includes a background of essential knowledge, and the zero-day cycle is developed in a general idea. Besides, understanding the threat actor's role is addressed to explain how it works behind the scene and assimilates all the back- ground motivational factors.

A state-of-the-art comparison was also exhibited regarding countermeasures, which explains the current solutions' approach and mechanisms. It shows that most of these solutions focus on detection, while prevention is underestimated, and limited solutions are available. It is advisable to continue digging on this approach. Finally, the open problems were exposed jointly with research challenges for future work to counteract the impact.

Conclusively, we plan to develop a zero-day laboratory that prevents and detects attacks launched on a day zero based on the reasons outlined in this study. Building on this can test several scenarios to avoid attacks by searching for assets, finding a vulnerability, and reducing the damage exposed there.

Furthermore, this archetype will be divided into two main modules. The first module has the task of preventing attacks published and dispersed throughout the web, analyzing their relevance and the assets' attack surface to be protected in an entity. However, this will be complemented by a detection module. The use of big data will be integrated for detection to analyze massive data to find behavioral anomalies that may present specific patterns of Zero-day attacks.

This proposal will differentiate this approach by having a hybrid model of prevention and detention simultaneously, besides applying big data to find spontaneous and random information not analyzed in standard strategies mentioned before. This method will aim to provide a solution that can detect anomalies with a low rate of false positives or false negatives.

Finally, the comparison in Table I showed that there is no systematic review of literature exclusive related to Zero-day studies. Therefore, conducting a SLR in this field is relevant and it is proposed as future work for the project members.

ACKNOWLEDGMENT

The authors would like to thank the financial support of the Ecuadorian Corporation for the Development of Research and Academy (RED CEDIA) for the development of this work, under Research Team GT-II-2018 (Cybersecurity). The research team was co-financed by the Research Department of the University of Cuenca (DIUC), Cuenca-Ecuador.

REFERENCES

[1] E. Chien, and L. O'Murchu, "Zero-day vulnerability: What it is, and how it works" [Online]. Available: https://us.norton.com/internetsecurity-emerging-threats-how-do-zero-day-vulnerabilities-work-30sectech.html [Accessed: Nov.25, 2020].

[2] S. Akshaya and G. Padmavathi. "A Study on Zero-Day Attacks," In Proceedings of International Conference on Sustainable Computing in Science (SUSCOM), pp. 2170–2177, 2019.

[3] A. Ye, Z. Guo, and Y. Ju, "Zero-Day Vulnerability Risk Assessment and Attack Path Analysis Using Security Metric," International Conference on Artificial Intelligence and Security, 11635(2016), pp. 266–278, 2019.

[4] P. Szor. "The art of computer virus research and defense". Pearson Education, 2005.

[5] L. Bilge, and T. Dumitras, "Investigating zero-day attacks," the magazine of USENIX & SAGE, 2013.

[6] MITRE. "Common Vulnerabilities and Exposures - CVE: The Standard for Information Security Vulnerability Names", 2019.

[7] L. Ablon, and A. Bogart, "Zero Days, Thousands of Nights: The Life and Times of Zero-Day Vulnerabilities and Their Exploits," Rand corporation, 2017.

[8] National Institute of Standards and Technology. "NVD – CVE-2010-2568" [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2010-2568 , [Accessed: Nov.25, 2020].

[9] National Institute of Standards and Technology. "NVD - CVE-2020-5902"[Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2010-5902 [Accessed: Nov.25, 2020].

[10] F5 Networks. "Article: K52145254: TMUI RCE vulnerability CVE-2020-5902"[Online]. Available: https://support.f5.com/csp/article/K52145254 [Accessed: Nov.25, 2020].

[11] SHODAN Search engine. "BIG-IP affected Software", 2020.

[12] X. Sun, J. Dai, P. Liu, A. Singhal and J. Yen, "Towards probabilistic identification of zero-day attack paths," IEEE Conference on Communications and Network Security, CNS 2016, pp. 64–72, 2017.

[13] L. Ablon, M. Libicki, and A. Abler "Markets for Cyber-crime Tools and Stolen Data: Hackers' Bazaar," Rand Corporation, 2014.

[14] T. Walshe and A. Simpson, "An Empirical Study of Bug Bounty Programs," In IBF 2020 - Proceedings of the 2020 IEEE 2nd International Workshop on Intelligent Bug Fixing, 2020.

[15] X. Riofrío, F. Salinas Herrera and D. Galindo, "A Design for a Secure Malware Laboratory," In Advances in Intelligent Systems and Computing, volume 1099, pp. 273–286, 2019.

[16] L. Bilge and T. Dumitras, "Before we knew it: An empirical study of zero-day attacks in the real world," In Proceedings of the ACM Conference on Computer and Communications Security, 2012.

[17] L. Glanz, S. Schmidt, S. Wollny and B. Hermann, "A vulnerability's lifetime: Enhancing version information in CVE databases," In ACM International Conference Proceeding Series, volume 21-22-Octo, 2015.

[18] International Organization for Standardization. "ISO/IEC 27001:2013". Information technology — Security techniques — Information security management systems — Requirements, 2013.

[19] R. Tang, Z. Yang, Z. Li, W. Meng, H. Wang, Q. Li, Y. Sun, D. Pei, T. Wei, Y. Xu and Y. Liu, "ZeroWall: Detecting Zero-Day Web Attacks through Encoder-Decoder Recurrent Neural Networks," IEEE INFOCOM 2020 - IEEE Conference on Computer Communications, pp. 2479–2488, 2020.

[20] A. Lobato, M. Lopez, I. Sanz, A. Cardenas, O. Duarte, and G. Pujolle, "An Adaptive Real-Time Architecture for Zero-Day Threat Detection," IEEE International Conference on Communications, 2018-May:1–6, 2018.

[21] M. Keramati, "An attack graph based procedure for risk estimation of zero-day attacks," In 2016 8th International Symposium on Telecom- munications (IST), pp. 723–728. IEEE, sep 2016.

[22] V. Sharma, J. Kim, S. Kwon, I. You, K. Lee and K. Yim, "A framework for mitigating zero-day attacks in IoT," eprint arXiv:1804.05549, pp. 1–4, 2018.

[23] F. Abri, S. Siami-Namini, M. Adl Khanghah, F. Mirza-Soltani and A. Siami-Namin, "The Performance of Machine and Deep LearningClassifiers in Detecting Zero-Day Vulnerabilitie," In Proceedings - 2019 IEEE International Conference on Big Data, Big Data 2019, 2019.

[24] A. Jamdagni, Z. Tan, X. He, P. Nanda and R.Ping Liu, "RePIDS: A multi tier Real-time Payload-based Intrusion Detection System," Computer Networks, 2013.

[25] L. Cavallaro, A. Lanzi, L. Mayer and M. Monga, "LISABETH: Automated content-based signature generator for zero-day polymorphic worms," In Proceedings - International Conference on Software Engineering, 2008.

[26] I. Kim, D. Kim, B. Kim, Y. Choi, S. Yoon, J. Oh and J. Jongsoo "A case study of unknown attack detection against zero-day worm in the honeynet environment," In International Conference on Advanced Communication Technology, ICACT, 2009.

[27] H. Tran, E. Campos-Nanez, P. Fomin and J, Wasek, "Cyber resilience recovery model to combat zero-day malware attacks," Computers and Security, 2016.

[28] V. Sharma, K. Lee, S. Kwon, J. Kim, H. Park, K. Yim and S. Young Lee, "A Consensus Framework for Reliability and Mitigation of Zero-Day Attacks in IoT," Security and Communication Networks, 2017.

[29] M. Conti, T. Dargahi, and A. Dehghantanha. "Cyber threat intelligence: Challenges and opportunities". In Advances in Information Security. Springer, 2018.

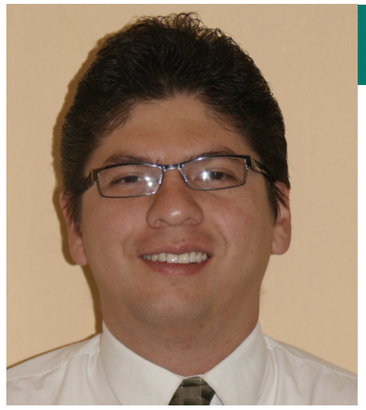
[30] A. Fagioli, "Zero-day recovery: the key to mitigating the ransomware threat," Computer Fraud and Security, 2019.

[31] K. Kim, M. Erza-Aminanto and H. Chandra, "Summary and further challenges," In Network Intrusion Detection using Deep Learning, Springer, pp. 69–70, 2018.

AUTHORS

Xavier Riofrío

Xavier Riofrío, a graduate computer science engineer from Universidad de Cuenca and a master of Cybersecurity with honours at the University of Birmingham certified by the UK government. The specialisation is in computer security and highly qualified for Penetration Testing, Ethical Hacking, amongst others. This leads me to be interested and enthusiastic about everything that entails the research field and Cybersecurity. That is why I have been part of diverse projects at the University of Cuenca.



Fabián Astudillo-Salinas

Darwin F. Astudillo-Salinas received the B.S.E (C.S) degree from Universidad de Cuenca, Cuenca, Ecuador, in 2007, and the M.S. and Ph.D. degrees from the "Institut National Polytechnique de Toulouse", Toulouse, France, in 2009 and 2013, respectively. Since 2013, he has been a Full-Time Researcher with the Department of Electrical, Electronic, and Telecommunications Engineering, Universidad de Cuenca, Cuenca, Ecuador. His research interests include network coding, wireless sensor networks, vehicular networks, networked control systems, simulation of networks, performance of networks and cybersecurity.

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2021

Luis Tello-Oquendo

Luis Tello-Oquendo received the electronic and computer engineering degree (Hons.) from Escuela Superior Politécnica de Chimborazo, Ecuador (2010), the M.Sc. degree in telecommunication technologies, systems, and networks (2013), and the Ph.D. degree (Cum Laude) in telecommunications from Universitat Polit\'ecnica de Val\'encia (UPV), Spain (2018). He was Graduate Research Assistant with the Broadband Internetworking Research Group, UPV (2013 - 2018) and Research Scholar with the Broadband Wireless Networking Laboratory, Georgia Institute of Technology, Atlanta, GA, USA (2016-2017). He is currently an Associate Professor with the Universidad Nacional de Chimborazo. His research interest includes 5G and beyond cellular systems, IoT, machine learning.

Jorge Merchan-Lima

Jorge Merchan-Lima,Since 2017 it has actively participated in research projects focused on digital signal processing, energy efficiency, data analysis, cybersecurity, and information security. I have written and published technical, scientific articles in national and international conferences.
Collaborator in the analysis of computer and electronic components in "FUNCIONA," a member of the cybersecurity working group in CEDIA, collaborator in a private company in compliance with ISO 27001, 27002, 27701, PCI, threat/vulnerability analysis, offensive security, and data protection.