Sri Lanka Institute of Information Technology

A Cybersecurity Nightmare: Zero-day attack

Individual Assignment

IE2022 - Introduction to Cyber Security

Submitted by:

Student Registration Number	Student Name
IT21026416	Rukshana M.A.F

Date of submission

Table of Contents

Abstract

Cybersecurity is rising in popularity by the day due to persistent engagement of the people in the digital world. Threats against computer systems emerge from all around the world in the form of cyber-attacks, in which some of these can be protected against while others are the nightmare of cyber security professionals. The zero-day attack belongs into the latter category.

Zero-day exploit is an exploit code developed by attackers who launch persistent execution to generate vulnerabilities in the system. A vulnerability that a developer is not aware of, is simply termed as Zero-day vulnerability. Zero-day attack is the attacker exploiting the discovered unknown vulnerability.

Although organizations heavily invest to strengthen their security system, they cannot fully eliminate risk. Bypass security through an unpatched software vulnerability, disclosed only after several continuous attempts shall be performed by a determined hacker.

Therefore, in this report, the work analysis is focused on securing systems with necessary current defense techniques and enhancements in defense techniques by introducing new proposed systems are discussed. Furthermore, the report states recent events and the impact the attacks can cause.

1. Understanding Zero-day attack

Softwares are released to the market after undergoing several testing phases, however it is never perfect, there are always unnoticeable flaws. To resolve these problems, the updates are frequently released by the developer. Identified vulnerabilities could be reported back by users or self-tests done by the developers or could be identified by a notorious criminal with a malicious intent.

Zero day refers to a vulnerability or a flaw that has zero days between the time the flaw is first discovered, and the first attack launched by the threat actor. In simple terms, vendors have known about the exploit for zero days. Every digital asset we use in our day to day lives have vulnerabilities [1].

Zero-day attack, a nightmare for the cybersecurity professionals, exploits the digital holes of software which was left unpatched in where organizations with vulnerable machines can easily be exploited. Until a fix is released, it is often a battle amongst the threat actor and the developer, in which attacker attempting to exploit the flaw and developer working on solutions that patches the vulnerability [1]. Zero-day attack can also be launched by creating a vulnerability in the system by the attacker using zero-day exploit codes.

Zero-day attacks can emerge can be in the shape of trojan, viruses, polymorphic worms, and other malware. An example for a zero-day attack that placed a significant landmark in the history of attacks is the Stuxnet worm, a zero-day attack, created to damage Iran's nuclear program. Four different zero-day vulnerabilities are exploited in windows operating system using the Stuxnet worm [2].

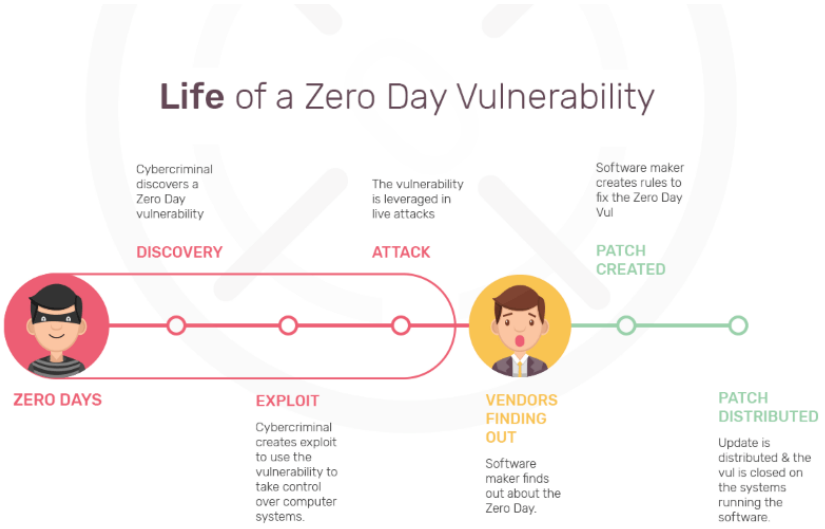

Figure 1: Life of a Zero Day Vulnerability [3].

The success rate of zero-day attack is most likely depends on the "window of exposure" of the organization, or the timeline between the detection of a vulnerability and release (and installations) of a patch that fixes it. A lengthy window of exposure can be in a known vulnerability due to organization's policies on patch management or the level of difficulty in developing the patch. The attack rests undetected when there is a longer window of exposure [1].

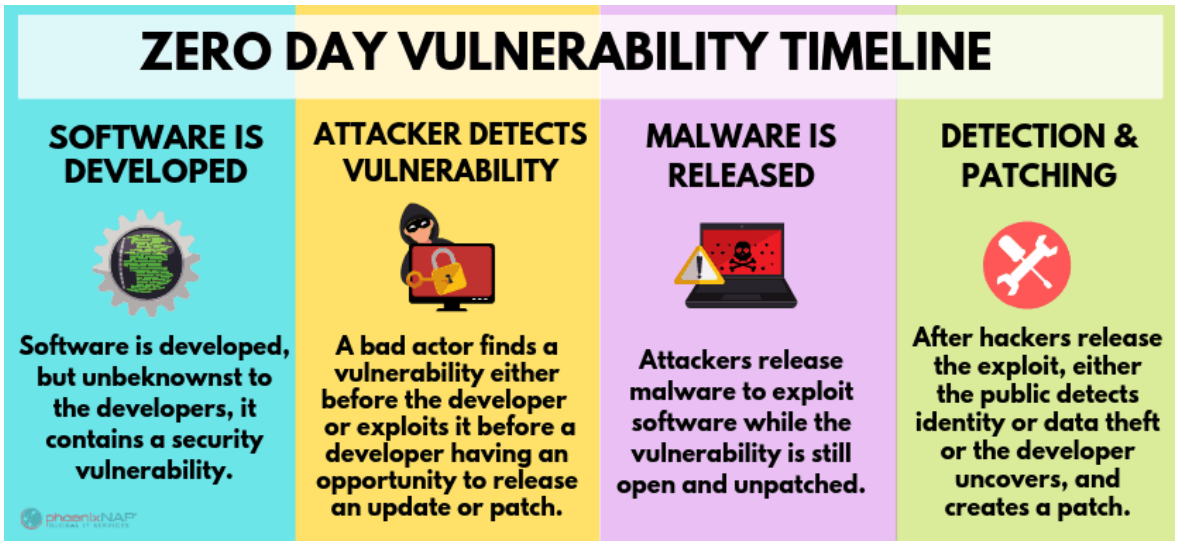

Figure 2: Vulnerability window [4]

In the dataset collected and gathered by the Google's Project Zero team from time range between from the mid-2014 to 2020. It is important to enlighten our knowledge of actual zero-day exploits that can help to enhance an organizational security posture [3].

The dataset contained a set of zero-day exploits that have been detected either in situations or found in the wild, where in wild means vulnerabilities that had been exploited after painstaking hours of attempts in finding vulnerabilities by the attackers. Dataset contains exploits created by NSA (National Security Agency) and leaked exploits by a hacking group called ShadowBrokers. Likewise, it also contains leaked tools of Hacking Team, which is an Italian private intelligence firm [3].

In total of 108 zero-day exploits were discovered between June 2014 and June 2019. In the wild, each year, on average of 20 zero-day exploits are found. The highest numbered zero-day attacks that exploited zero-day vulnerabilities is in the year 2015, with 28 detected exploits. Only by 2018, the lowest numbered zero-day attacks were detected, which is 12. That number is almost equivalent to the first 6 months of 2019 with 10 discoveries [3].

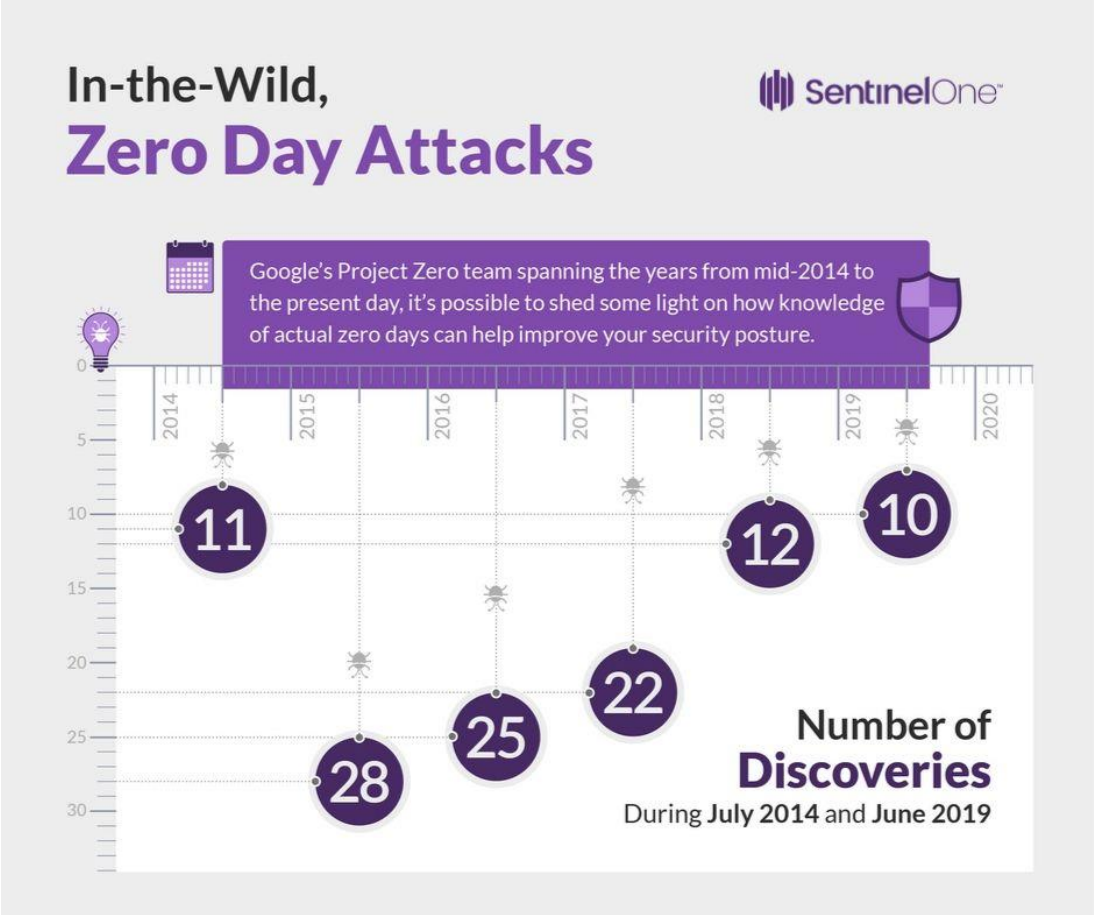

Figure 3: Number of discoveries between 2014 and 2019[5]

2. Creation of zero-day attack

Analysis and attack testing: In the beginning stage, the attackers search for flaws in software. A detailed analysis of software is needed, which can be done through observing the software source code and possessing a broad amount of awareness on the subject is required. The perpetrator shall implement numerous methods to accomplish this and is frequently where abnormal behaviors are detected [4].

Fuzz analysis: Identical to attack testing, fuzz testing is done by passing random values to the system, with the motive to find vulnerabilities. The process is carried out with the assistance of tools frequently. The attack has the tendency to be successful if this is not detected for long period of time [4].

Creation: At this stage, it is difficult to prevent zero-day intrusion, as once the attacker identifies an appropriate entry point and/or flaws, the zero-day exploitation is created. It is greatly challenging and difficult to detect such malware created to the launch attack [4].

Implementation and Deployment: During this stage, the developed malware is delivered into the victim's system. Some victim companies can remain inattentive that they have been affected until their crucial systems are endangered [4].

3. Recent Zero-day attacks

Certain successful Zero Day attacks are being discussed below,

Google's chrome a cross platform web browser is one of the top trustable web browsers prevailing currently (2022). In November 2015, chrome experienced a Zero-day exploit in which many android devices were prone to remote hacking. Remote total control of all android mobile devices even if devices were updated to its up-to date version in its android operating system. The vulnerability in the pre-installed JavaScript v8 engine, which exists in all android devices, was the digital crack for the attackers to take advantage of. For the exploit to be successful, the attacker simply needs to trick the victim into visiting a web page that holds malicious exploit code from chrome [5].

In January 2019, researchers identified a zero-day attack employed in an increasingly targeted attack in Europe. The exploit takes advantage and exposes a local privilege escalation flaw in Microsoft windows, especially in win32k.sys a NULL pointer dereferences. The Microsoft Security Response Center upon receiving the report and made swift fixation on the vulnerability and released a patch [6].

In January 2020, Microsoft warned the users of windows that Internet explorer is facing a zero-day attack launched by the attacker exploiting the vulnerabilities in the wild. And there is no patch available for it yet. The Zero-day vulnerability was traced as CVE-2020-0674, is a remote code execution issue problem is in a way where the objects in the memory of the Internet Explorer handled by scripting engine, triggers the library jScript.dll [7].

Stuxnet: Stuxnet worm zero-day exploit, primarily identified before the target's awareness of such zero-day vulnerability existence, was used to target computers engaged in manufacturing purposes which was operated in countries [8].

Prime target of the attack was Iran's uranium enrichment plants, to disrupt country's nuclear enrichment capabilities. These types of are usually conducted by state sponsored hackers focused on targeting foreign governments, terrorist groups and corporations [8].

The vulnerability was found on programmable logic controller (PLC) or programmable controller, which oversees the management of manufacturing processes that require high reliability, fault diagnosis of process, conveniency in programming such processes are assembly lines, robotic devices, machines or any other process that requisite the

mentioned requirements. The worm falsified the logic controllers, through vulnerabilities which resulted in holding out unforeseen commands on assembly line machinery by the controllers, impairing centrifuges used to separate nuclear materials [8, 9].

Sony zero-day attack: In 2014, the Sony Pictures faced a devastating zero-day exploit. The attack destroyed Sony's network and led to unauthorized disclosure of sensitive corporate data on file sharing sites. Breached data such as email communication details of Sony executives, movies released in the future and business plans. The exact details of vulnerabilities that led the Sony attack remains unknown [8].

RSA: In 2011, the security company RSA faced an access breach into the network by the hackers, where they used an unfixed vulnerability in Adobe flash. A small group of RSA employees were aimed to receive malicious emails with excel spreadsheet attachment, which were sent by attackers. These spreadsheets were loaded with embedded flash file that exploited the zero-day flash vulnerability. When one of the employees being received with such emails, opened the spreadsheet, the control of the computer was conquered by the attacker after succeeding in installing poison ivy administration tool. Attackers then pursued and stole sensitive information and transmitted to the external servers controlled by them. The security company confessed that amongst the stolen data was sensitive information connected to the company's SecurID two-factor authentication products, aimed to access sensitive data and devices globally [8].

DNC Hack: The DNC hack launched by Russian hackers, was a major landmark event that, released data about the Democratic National Committee (DNC) due to a zero-day attack. To gain access to the data at least six zero-day vulnerabilities were exploited. The vulnerabilities were in Adobe flash, Microsoft windows, and java. To exploit the vulnerabilities, Russian hackers engaged in spear-phishing. The purpose of the attack i.e., stealing password of the people related to DNC is achieved, through sending several emails with malicious trap link attachments within it to the phishing pages. Victims who clicked onto bit.ly and tiny.cc concealed URLs gave away the control of their personal computer and the DNC network [10].

Operation Aurora:

A Zero-day exploit engaged in the theft of intellectual property rights on several major enterprises including Google, Adobe systems and yahoo. Internet Explorer and Perforce were said to be the location of the existed vulnerabilities [8].

Aurora attack is opposed to software, which occurred silently without the knowledge of the user through an exploitation of a browser vulnerability. Aurora occupied advanced persistent threat (APT) technique which highly proved the success in stealing victim's intellectual property rights. The purpose of APT is to gain access to the chosen data and maintain a position in the targeted system for future use [11][12].

In APT, Advanced means the threat actor is familiar with techniques and intrusion tools with sophisticated skills on creating custom exploits. In the case of aurora attack, attackers launched spear phishing against the victim companies, resulted in gaining access to the victim's network. In short, sophisticated attackers fulfilled the advanced criterion [11].

Persistent means the attacker plans to fulfil the mission. In aurora's case, after breaching and gaining access to the victim's companies, they worked towards exploiting internal resources [11].

Threat means the attacker is driven, organized, and monetarily funded [11].

Consequences due to aurora attack were the modification of source code in repository, stolen intellectual property rights and more (Availability Confidentiality and Integrity disruption) [11].

4. Impacts of zero-day attack.

4.1. Data theft.

After discovering zero-day vulnerabilities, they can be exploited by the attackers to steal sensitive data from organizations, employees, and customers. Attackers have several

options to do with the data stolen, such as use the data to steal money, trade sensitive information in the dark web, engage in identity theft or coerce the victim [13].

4.2. Unauthorized control

Control of network, website, server, programs, or any other systems, will be held by hacker due to a victorious exploitation of discovered vulnerability. Moreover, attacker can extend his hack by passing malicious messages to effected victim's contact list [13].

4.3. Defacement of reputation.

Whether the company has found the patch or not, if the attack is publicized, the reputation of such company is highly damaged in a long run. The public's trust on the company will be reduced, that the system lacks protection against attacks [13].

4.4. Loss of production and productivity.

Zero-day exploits hinders in employees and company's productivity, once they take control of the production machines, digital communications, and other systems [13].

4.5. Financial loss

The system unavailability caused due to the attack, can result a tremendous amount of loss in the revenue, especially in the financial services domain. The loss of money can also result due to heavy spend on response actions, recovery methodologies, and investigations after the attack. The consequences force such enterprises and start-ups to shut down [13].

4.6. Legal implications

The highly reputed companies managing critical systems need to prove to the regulatory authorities, customers, or stakeholders that the breach in the system is not due to weak security posture. If unable to prove, the organization might face lawsuits and result in paying massive fines and penalties [13].

5. Security measures for detecting and preventing zero-day attacks.

5.1. Vulnerability scans

Few of zero-day exploits can be detected using vulnerability scans. Security vendors that provide vulnerability scanning solutions can stimulate digital assaults on software code, conduct code reviews, and attempt to discover new vulnerabilities that may have been introduced following a software update [8].

All zero-day exploits cannot be detected in this approach. Despite detecting some zero-day exploits through scanning, which alone is not enough. Organizations should immediately act upon the results of the scan, in which they should perform code reviews, sanitize their code with the intent of preventing zero-day exploit into affecting them. Most organizations, tend to be slow in responding to newly identified flaws, while quick exploitations of the zero-day exploit can be performed by the perpetrators [8].

5.2. Patch management.

This strategy involves in deploying software updates promptly for newly identified vulnerabilities. Although this strategy doesn't prevent zero-day attacks but can significantly reduce the risk of an attack by swiftly executing patches and software upgrades [8].

Nevertheless, there are three factors involved in delaying deployment of security patches [8].

1. Time taken for the software vendor to identify vulnerabilities.
2. Development of a patch.
3. Distribution to the users.

It could also take time to apply such patches on the organizational systems. The longer the time taken for this process, the greater the risk of a zero-day attack [8].

5.3. Input validation and sanitization

Numerous issues in the vulnerability scanning and patch management are solved by input validation. It is controlled by the security experts, and they provide protection to the organization while the organization is involved in patching the system or code sanitization, which takes plenty of time. This provides more flexibility, ability to adjust in situations and counter new threats in real time [8].

Amongst the most successful ways to prevent zero-day attack is implementing web application firewall (WAF) on the network edge, in which it reviews all the incoming traffics and filters out inputs that are malicious and that might target security vulnerabilities [8].

Furthermore, an advanced recent technique to fight against zero-day attacks is the runtime application self-protection (RASP). RASP agents reside inside the application, monitoring request payload with the application code context at runtime, to dictate a request is normal or suspicious, if suspicious RASP enables the applications to defend themselves [8].

5.4. Zero-day initiative

A program that is designed to encourage security researchers by rewarding for disclosing vulnerabilities, rather than trading information on the black market. The purpose is to create a greater community of vulnerability researchers who are capable to find new security vulnerabilities before hackers do and warn the software developers regarding them [8].

6. Traditional defenses against zero-day attacks

The general threat to every organization's network is the zero-day attack. The intentions of zero-day are predominantly spying on victim's operations, theft of confidential data, interference in the smooth operations of the system. Detection of exploit as accurate as possible during the exploitation time, destroy or reduce the damage generated because of the attack, are declared as the primary goal of defense techniques. In this part of discussion, in-depth analysis is given for few of the currently used zero-day defense techniques of the organizations [14, 15].

6.1. Statistical based defense technique.

The statistical based technique involves in maintaining a log of all past zero-day exploits that are identified and using the data from the log to produce profiles, which results in creating new parameters to detect attacks. In simple term, this technique allows or blocks the network traffic based upon the past profiles. To do the mentioned operations of the technique, it should first determine which network traffic is acceptable and doubtful. Network traffic is compared with the log for verification. Furthermore, as the log is updated on a regular basis, the longer the usage of this technique by any system, the more capable and precise the technique is at gaining knowledge and determining normal activities. The disadvantage of this technique is since it creates profiles from the past data in the log, which are in static nature; thus, this technique cannot detect real time zero-day attacks as they are unable to embrace dynamic behavior of the network environment [14], 15].

6.2. Signature based defense technique.

Signature based technique provides defense against network or system from attacks with malicious intent in the shape of worms or trojan horses, mostly in software packages for antivirus. The signature library should be persistently updated with newly detected virus signatures, so that the signature of such virus can be matched against network traffic. The downfall of the technique is the signature of the attack should be in the signature library before the system identify it with zero-day attacks which are not having familiar signatures [14].

6.3. Behavior based defense technique.

Behavior based technology does the prediction on how the traffic will be on the network flow. The objective is to predict the behavior of the network to detect and prevent unusual behavior of the network traffic in the network. The objective can be achieved through machine learning approach, which analyzes the network activities of the present and past of the victim’s machine, web server or server [14].

7. Proposed systems for better defense against zero-day attack.

Zero-day attacks undoubtedly are difficult to be detected with traditional cybersecurity practices. A determined attacker spends hours, weeks, days even years to evolve skills of discovering zero-day vulnerabilities. Thus, more advanced mechanisms must be implemented in detecting such attacks before the attackers [16]. Below are some of the proposed systems given to fight against zero-day attacks:

7.1. Zero-shot machine learning

Machine learning (ML) is a subgroup of Artificial Intelligence (AI). The performance and the efficiency of numerous technological applications improved due to the utilization of machine learning capabilities. ML models that are not practically realizable by the domain experts, are acknowledged for their exceptional capacity to learn and extract complex data patterns. The learnt pattern has been a disruptive innovation due its usage of prediction, classification, and revert time ahead events and scenarios, in which

operational and automation is required in multiple industries. For that reason, ML models are used in various domain due its exceptional success over traditional computing algorithms, where there is a challenge in performing essential operations. This motivation has led to the ML model implementation on domain of Cybersecurity, for the sake of strengthening the organization's security posture additionally. ML models have been used to safeguard computer networks against advanced persistent threat (APT). Number of internal and external threats in the organization, can be limited with the additional defense layers if they are designed efficiently. Detection of present-day attacks that need latest innovation detection capabilities are provided by ML [17].

One of the vital security tools that engages in threat detection as it penetrates through the organizational network is the Network Intrusion Detection Systems (NIDSs). NIDS scans arriving network traffic for any attack signatures or Indicator of Compromise (IOC) such as internet protocols, domain names, hash values that specify malicious traffic. The detection of Zero-day attacks is one the main and occurring challenges of protecting network by the NIDSs. It takes average of 312 days to detect zero-day attacks with signature based NIDS. Also, the assets protected with NIDSs tend to be vulnerable to such attacks. Hence to overcome the hindrance in detection of zero-day attacks, ML based NIDS is given attention [17].

ML based NIDS is aimed at scanning and analyzing approaching network traffic for any malicious intent. ML based NIDSs bestow greater potential in detection of zero-day attacks with the usage of its learnt behavioral pattern to detect network attacks [17].

The technique used to enhance and evaluate the generalizability of ML models to the unseen data classes is the Zero-day shot Learning (ZSL). It addresses the escalating class sets that might make it impossible to collect training samples for each of them. ZSL involves in identification of new data samples created from the previous unseen classes. One the obstacle the ZSL faces is the building of reliable NIDS, in which the analyzes the detection of new attack classes such as zero-day attacks which are not in the design phase [17].

A proposal for future defense against zero-day attack is raised, which is a new zero-day evaluation framework, inspired by ZSL. The approach assess how successfully an ML-based NIDS detects unseen assaults by employing a set of semantic properties learned

from visible attacks. The suggested future defense technique has two major steps. During the attribute learning stage, the models extract and network data features are mapped to the unique features of known attacks. During the inference phase, the relationship between the known attacks and the zero-day attacks are associated by the model to assist in their determination and categorization as malicious. A ML-based NIDS called Zero-day detection Rate (Z-DR) is used to measure the efficiency of learning model can be to rebuild distinguishing semantics learned from known attack classes to detect unknown attack classes [17].

7.2. Hybrid Security Framework for Zero-day attack

In the proposed hybrid model, the framework merges the signature-based technique and the behavior-based technique. It is accustomed to observing the traffic flow arriving at the network and check if it's a malicious threat or not [14].

In the proposed system architecture, there are 6 major components, they are as follows:

1. Packet acquisition
2. Packet extraction
3. Disassembly module
4. Analysis and evaluation module
5. Signature generation
6. Signature matching
7. Behavior analysis

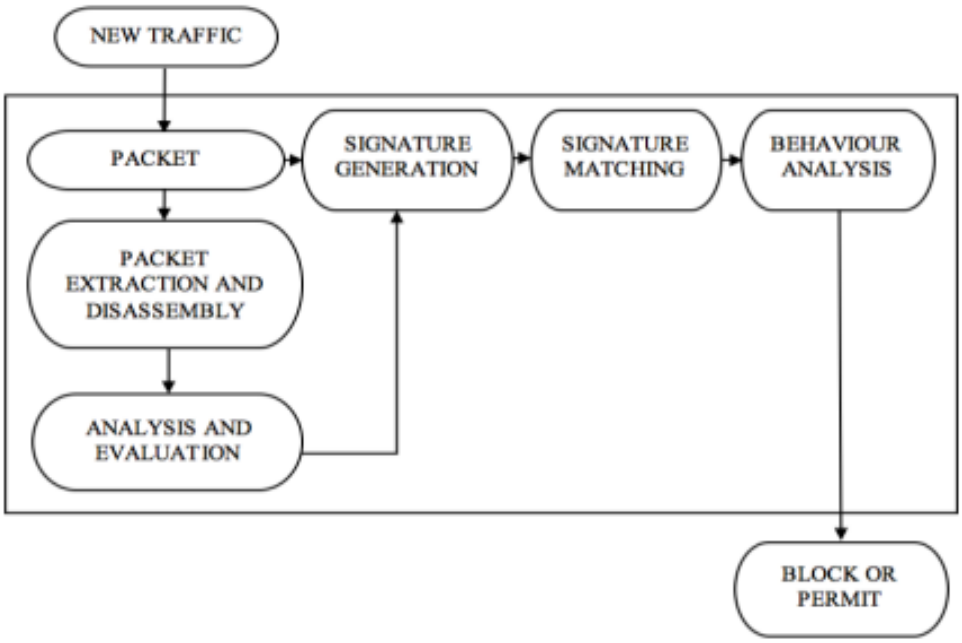

Figure 4: Signature-Behavior based hybrid technique [15]

When a new traffic flows into the network, all the packets that accords to the same flow are assembled by the packet acquisition module, and they forward the assembled packets to the extraction and disassembly module. Upon arriving to this module, extraction and disassembling process are executed in individual packets and forwarded them to the analysis and evaluation module. Then that module uses an Intrusion detection or prevention systems (i.e. IDS or IPS) to detect malwares bounded in the packet. Signature generation received the packet after they are reassembled. After the generation, the produced signature is matched with signatures in the database, if no matches found, the traffic is moved to the behavior analysis module to conduct traffic flow analysis by adopting Hidden Markov Model machine learning method. If an abnormality is detected, the traffic is blocked from entering the network. If no abnormality is detected, it is authorized to enter the network [14].

The aim of this model is to identify peculiarities and set apart the malicious content to avoid entering the network. With the help of machine learning, the limitations of zero-day attacks can be achieved [14].

7.3. Zero-day attack remedy (ZDAR) system

Occurrence of zero-day attack happens in the timeline between when the vulnerability is initially exploited and the patch development to counter that attack by the vendor [15]. There is a tough challenge in measuring the time span since it is difficult to decide the first discovery of the vulnerability. Sometimes vendors fail to determine if the vulnerability has already been exploited when they fix it. Nevertheless, a vulnerability can remain in the system for a very long time. In proportion to the FireEye, an average of 312 days may last for a typical zero-day attack [15].

In the proposed system, there are six main elements:

1. Data acquisition through traffic analyzer.
2. Extraction.
3. Transformation.
4. Supervised Classification.
5. Intrusion Detection System (IDS).
6. UI (server machine/ host/ client machine) portal [15].

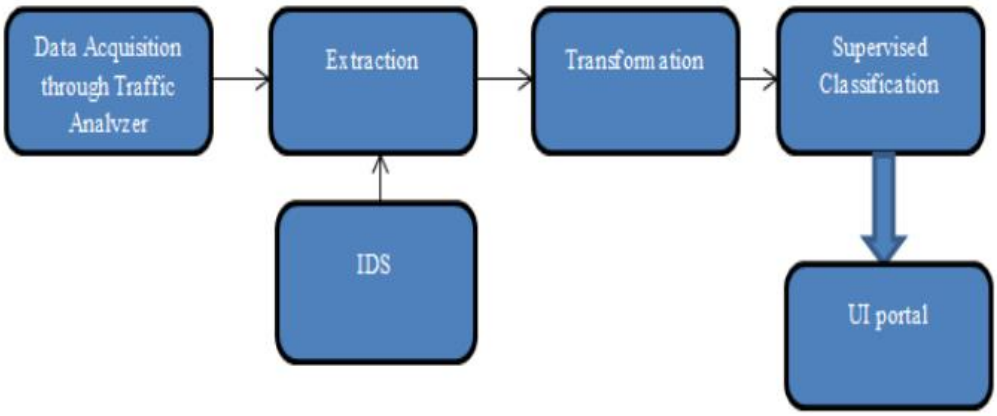

Figure 5: ZDAR Framework [15]

Module that captures data is the Traffic Analyzer (TA), which breaks the packets into components, collects and combines them belonging to the same flow. Duty of this module is to generate the features of the flow-level related with this flow. Activities carried out by the IDS or Intrusion Prevention System (IPS) are deep inspection of packet and marks the flow whether it associates to some threat. All the flow features and connected class tags are store in the information storage component. The extraction property extracts the statistical features on each flow while the transformation element converts them into more sturdy elements in which it will be used to build classifiers for malware flow detection. Classifiers are built in an offline style and are distributed to the incoming flows in the network. Emergencies of any new suspicious flows in the network are reported through the UI portal [15].

The primary goal of the proposed framework is to identify and prevent flows that is malicious from the network traffic and additionally classify them into a type of a prevailing malware, variation of a prevailing malware or an unknown (new) malware. To accomplish that, machine-learning based malware detection and classification framework is developed by the observation of company's features of network traffic [15].

This proposed system, thus provides correctness of supervised classification on known classes with the adjustability of unsupervised acquiring of knowledge for the new detection of malware [15].

7.4. ZeroWall framework

Zero-day attacks are challenging and consequential threats to web security as they are difficult to be detected by extensively used signature-based Web Application Firewalls (WAFs) [18].

Proposed system suggests Zero wall, which is an unsupervised approach used to successfully detect zero-day web attacks as it works with an existing WAF in pipeline [18].

Labeled data for training is often required by most of the supervised approaches, thus they are inappropriate approaches for zero-day attacks as the attacks are not seen previously. In other words, suitable approaches for zero-day attack detection are unsupervised approaches. The fundamental observation of zero wall is that the main structure of a harmless web request is a string following HTTP protocol. Although consistent syntax and semantic patterns are not there in a malicious web request compared to the harmless one [18].

The inspiration behind this proposed approach is the unsupervised self-translation machine-based encoder-decoder recurrent neutral network, i.e. when practiced with adequate sentences per language ‘A’, neutral network acknowledges that language good enough in such a way that it can convert the input in the ‘A’ into a latent representation

which converts back as an output sentence in ‘A’. Trained neutral network can tell if the sentence ‘s’ in unfamiliar language accords to language ‘A’ or not. Hence if the ‘s’ accords to language ‘A’ if the conversion quality is high, else ‘s’ does not accord to ‘A’ language [18].

In the Zerowall, the detection of zero-day can be done by mapping the zero-day detection problem to the machine translation quality assessment problem. The encoder-decoder recurrent neutral network is trained using past web requests allowed by an WAF, in which it is used to capture the semantics and patterns of the harmless web requests. In the real-time, the zero-day web requests (in which WAF cannot detect), self-translation machine cannot understand and translate back to its original request, hench this is considered as an attack [18].

Thus, the proposed technique puts forwards the following,

- A framework of combining the prevailing signature-based WAFs with the unsupervised machine learning based zero-day web detection approach.
- Prototype ZeroWall by adopting encoder-decoder recurrent neutral network [18].

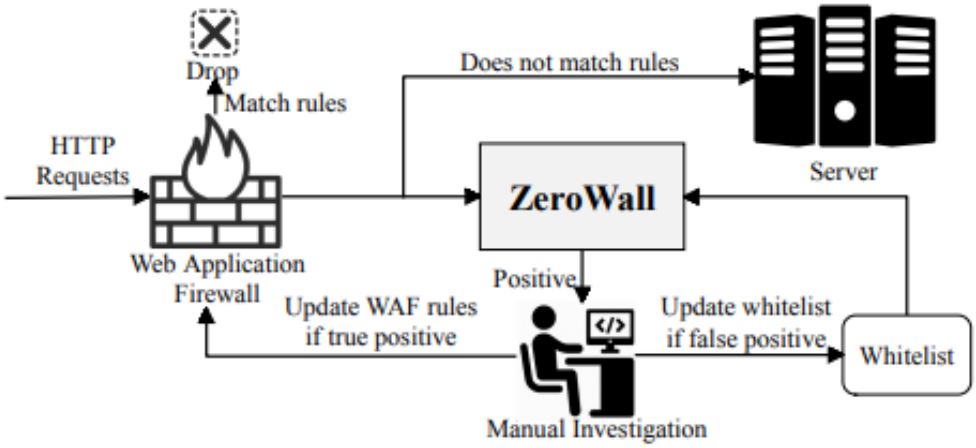

Figure 6: The workflow of Zerowall [18]

As it shows in the figure above, Zerowall is accustomed as a bypass system in the back of the WAF, in which web services are not burdened with extra overhead. The incoming HTTP requests undergoes through WAF, engages in the process of matching itself with the WAF rules in contemplation of detecting known attacks. If the requests does not match the rules, those unmatched rules are passed onto to the server and the traffic is reflected and sent to the Zerowall, which concentrates on detecting zero-day attacks. Another benefit from working together with WAF is during when known malicious samples are not friendly to the unsupervised algorithms. The performance of the unsupervised algorithm is upgraded when the prevailing WAF filters out known attacks. Also, it is a good solution to defend against malicious invasions such as poisoning traffic, in which large number of attacks are injected in the traffic [18].

Conclusion

Mankind is the weakest link in the computer security. They often consider the network is secured after implementation of many security defense techniques, even when everything seems quiet, they are not safe and still at risk. Anything handle by all networks or software are vulnerable to attack. Zero-day attacks are considered a burden to the cyber security professionals, as they appear from an unpatched vulnerability in the network. Importance of defense techniques comes into necessity. Networks are dynamic in nature and consist of greater number of uncertainties, thus maintaining a safe network is difficult, for that reason organization must persistently look for new methods to protect their network against hackers from exploiting vulnerabilities. This report mentions, proposed systems for that purpose. Furthermore, by studying the recent attacks, an expert in security can get an idea of how the zero-day attacks are discovered. Prevention is derived from awareness, to be aware we need to detect, thus this report also mentions some of the best security practices to detect and prevent zero-day attacks. In addition, the report illustrates some steps on how attacker creates a zero-day attack.

Zero-day threats are a nightmare to all the security experts in the organizations. If the zero-day vulnerability is exploited, they must fix it immediately. However, with current and future methodologies, securing systems become much easier.

References

[1] “Security 101: Zero-Day Vulnerabilities and Exploits – Security News,” trendmicro, Oct. 02, 2019. [Online]. Available:

https://www.trendmicro.com/vinfo/us/security/news/vulnerabilities-and-exploits/security-101-zero-day-vulnerabilities-and-exploits

[2] D. Hammarberg, “The Best Defenses Against Zero-day Exploits for Various-sized Organizations”, White Paper, 2014. [Online]. Available: https://www.sans.org/white-papers/35562/

[3] “Zero Day Survival Guide | Everything You Need to Know Before Day ”, SentinelOne, June. 13, 2019. [Online]. Available: https://www.sentinelone.com/cybersecurity-101/zero-day-vulnerabilities-attacks/

[4] “Understanding Zero-Day Attacks”, cyberdefense Inc, Jan. 15, 2019. [Online]. Available: https://cyberdefenses.com/understanding-zero-day-attacks/

[5] M. Kumar, “Chrome Zero-day Exploit leaves MILLIONS of Android devices vulnerable to Remote Hacking,” The Hacker News, Nov. 13, 2015. [Online]. Available: https://thehackernews.com/2015/11/android-hacking-chrome.html

[6] “Windows zero-day CVE-2019-1132 exploited in targeted attacks,” WeLiveSecurity, Jul. 10, 2019. [Online]. Available: https://thehackernews.com/2015/11/android-hacking-chrome.html

[7] M. Kumar, “Microsoft Warns of Unpatched IE Browser Zero-Day That’s Under Active Attacks,” The Hackers News, Jan. 18, 2020. [Online]. Available: https://thehackernews.com/2020/01/internet-explorer-zero-day-attack.html

[8] “What is zero day exploit,” Learning Center. [Online]. Available: https://www.imperva.com/learn/application-security/zero-day-exploit/

[9] “Programmable logic controllers,” Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Programmable_logic_controller

[10] “Recent Zero-Day Attacks: Top Examples and How To Prevent It,” PhishProtection.com, Dec. 16, 2019. [Online]. Available: https://www.phishprotection.com/content/zero-day-protection/recent-zero-day-attacks/

[11] R. Aslanoglu, S. Tekir, “Recent Cyberwar Spectrum and its Analysis,” In Proc Izmir. inst. Tech., Izmir, Turkey. July 2012.

[12] B.E. Binde, R. McRee and T.J. O’Connor. “Assessing outbound traffic to uncover advanced persistent threat,” SANS institute, May 2011. [Online]. Available: https://www.researchgate.net/publication/333634056_Assessing_Outbound_Traffic_to_Uncover_Advanced_Persistent_Threat

[13] “What are the Potential Impacts that Zero-Day Vulnerabilities Pose to Your Organization,” Indusface, Jan. 11, 2021. [Online]. Available: https://www.indusface.com/blog/what-are-the-potential-impacts-that-zero-day-vulnerabilities-pose-to-your-organizations/#:~:text=4.-,Loss%20of%20Production%20and%20Productivity,hampering%20employee%20and%20organizational%20productivity.

[14] D.C. Cuppah, G. Ambrish and M. Hanumanthappa. (2020). “Design and Analysis of a Hybrid Security Framework for Zero-day Attack,” Presented at Int. Conf. on Envision. Fut. knwl. Engr., Bangalore, India, May 2020.

[15] U.K. Singh, C. Joshi, and S.K. Singh, “ZDAR system: Defending against the unknown,” International Journal of Computer Science and Mobile Computing, vol. 19, pp. 143 – 149, Dec. 2016.

[16] “How to Detect and Prevent Zero-day Attacks,” Indusface, Jul. 20, 2017. [Online]. Available: https://www.indusface.com/blog/prevent-zero-day-attacks/

[17] M. Sarhan, S. Layeghy, M. Gallagher, and M. Portmann, “Zero-Shot Machine Learning to Zero-Day Detection,” arXiv 2021. [Online]. Available: https://www.researchgate.net/publication/358501705_From_Zero-Shot_Machine_Learning_to_Zero-Day_Attack_Detection

[18] R. Tang, Z. Yang, Z. Li, W. Meng, H. Wang, Q. Li, Y. Sun, D. Pei, T. Wei, Y. Xu, and Y. Lu, “ZeroWall: Detecting zero-day Web attacks through encoder- decoder recurrent neutral network,” in Proc. IEEE Conf. Comput. Commun. (INFOCOM), Jul, 2020, pp. 2479-2488.