Detection and Notification of Zero-Day attack to Prevent Cybercrime

[1]Atharva Deshpande
Bansilal Ramnath Agarwal Charitable Trust's
Vishwakarma Institute of Information Technology, Pune
atharva.22010368@viit.ac.in

[2]Isha Patil
Bansilal Ramnath Agarwal Charitable Trust's
Vishwakarma Institute of Information Technology, Pune
isha.22010352@viit.ac.in

[3]Jayesh Bhave
Bansilal Ramnath Agarwal Charitable Trust's
Vishwakarma Institute of Information Technology, Pune
jayesh.22010860@viit.ac.in

[4]Aum Giri
Bansilal Ramnath Agarwal Charitable Trust's
Vishwakarma Institute of Information Technology, Pune
aum.22010477@viit.ac.in

[5]Nilesh P. Sable
Bansilal Ramnath Agarwal Charitable Trust's
Vishwakarma Institute of Information Technology, Pune
drsablenilesh@gmail.com

[6]Gurunath T. Chavan
Bansilal Ramnath Agarwal Charitable Trust's
Vishwakarma Institute of Information Technology, Pune
gt.chavan@gmail.com

Abstract: **This paper seeks to understand how zero-day vulnerabilities relate to traded markets. People in trade and development arereluctant to talk about zero-day vulnerabilities. Thanks to years of research, in addition to interviews, The majority of the public documentation about Mr. Cesar Cerrudo's 0-day vulnerabilities are examined by him, and he talks to experts in many computer security domains about them. In this research, we gave a summary of the current malware detection technologies and suggest a fresh zero-day malware detection and prevention model that is capable of efficiently separating malicious from benign zero-day samples. We also discussed various methods used to detect malicious files and present the results obtained from these methods.**

Keywords: CVE, Honey Pot, Zeus, HPCs,Zero Wall, DECAF

I. INTRODUCTION

In recent years, open-source software has grown in popularity. That year, 31 million developers worked on his 96 million repositories.[2] OSS vulnerabilities are becoming more prevalent. The quantity of open- source vulnerabilities that have been made public has climbed by 53.8%. The day after the Apache Struts 2 remote code execution vulnerability was made public, exploit scripts began appearing in the wild.

An unpatched system used in the attack on Equifax exposed millions of personal records, including social security numbers. Administrators or users should make sure a stable version is available because installing software patches or upgrading to a new version causes service system downtime and adds totheir workload. You should look for frequent weaknesses and exposures that you typically try to avoid. As a CVE states that it is not. There is a critical security patch for this vulnerability.

However, Vendors of software can secretly fix flaws without submitting CVEs or publishing security issues in changelogs. One ofthe causes is the worry that including too many CVE entries and vulnerability fixes in the changelog would harm the software's reputation. These are some reports of CVEs and CVE vulnerabilities. So, let's examine what we refer to as cybercrime. Crimes done over the Internet and other computer networks are included in the category of "cybercrime." outperforms the traditional large-scale felonious assiduity." Cybercrime in military contexts, such as cyber warfare, requires a unique perspective. These types of attacks tend to have higher investment costs, be more targeted and aimed at achieving specific strategic objectives, and have a lower probability of success. As a result, the return on investment in these cases may be lower. In addition to financial considerations, geopolitical factors may also be at play and profit may not be the main motivation

II. LITERATURE SURVEY

Nowadays, systems are typically protected with advanced software and hardware, and it is necessary to carefully examine and understand a system in order to successfully attack it. This may involve identifying the types of protection in place, the operating system being used, and any vulnerabilities that may exist. The goal of such an attack may be to steal information that could be profitable or to misuse IT resources.

Run-time hardware feature-based zero-day malware pattern detection was addressed by authors in [1].Using a few currently accessible microarchitectural features and run-time hardware performance counters, the research demonstrated an ensemble learning-based technique to improve the performance of common malware detectors (HPCs). The method for identifying low-intensity denial of service attacks created using a hybrid neural network is described in the publication [2]. By lowering the frequency of undiscovered attacks (errors of the second kind), the method shows a high percentage of attack detection, and the speed of operation solely depends on how quickly incoming packets are processed. It is effective to identify traffic that is part of a time- distributed attack using the suggested method for detecting low-intensity attacks

The CVE (Common Vulnerabilities and Exposures) source in [3] defines a vulnerability as a software fault that can be used by hackers to enter a system or network without authorization. These vulnerabilities may arise from flawed patterns in complex systems, user neglect (such as using weak passwords or sharing passwords), or other factors. 0-day exploits, also known as zero-day vulnerabilities, are exploits that are not generally known. These exploits are often discovered by elite hackers and shared only with a small group of trusted individuals. The term "zero-day" refers to the fact that the value of an exploit drops significantly once it is publicly disclosed. Many zero-day exploits are discovered through the use of honeypots or by hackers themselves.

[4] evaluated a comprehensive architecture for incorporating a zero-day Web attack detection method into already-existing Web Application Firewalls with signature-based protection (WAFs). This approach is immediately applicable to a wide range of real-world circumstances. The zero-day Web assault detection problem is formulated as a neural machine translation quality evaluation problem in the prototype of ZeroWall using state-of-the- art neural machine translation methods, such as encoder-decoder recurrent neural networks.

From [5], The value of a zero-day vulnerability decreases rapidly after it is publicly announced; for example, after one day it may be worth half of its original value, and after 10 days it may be worth only a fraction of its original value. A computer vulnerability that is actively used before it is made available to the public is known as a zero-day exploit. Cesar Cerrudo has noted that attacks and zero-day vulnerabilities have always existed, although they may not have always been exploited for profit. It is important to distinguish between threats and vulnerabilities. For instance, in recent years, Heartbleed and Zeus were both well-known attacks.

- Zeus is a type of malware that targets vulnerabilities in using Windows systems to steal bank login information. It does this by capturing keystrokes and stealing information entered into online forms, a technique known as man-in-the-browser keystroke logging and formgrabbing.

- Heartbleed is a security flaw that takes use of a flaw in the free and open-sourceOpenSSL cryptography library. To implement the Transport Layer Security(TLS) protocol, OpenSSL is frequently used, which is a secure communication protocol designed to prevent tampering and eavesdropping.

- Zeus and Heartbleed are both examples of exploits that take advantage of vulnerabilities. However, there are some. Key difference between these two exploits

- Zeus is a series of programs developed to exploit flaws in Windows systems, while Heartbleed is a set of instructions written

- In this respect, Heartbleed is an attack that takes advantage of a vulnerability,whereas Zeus is a threat.

Zero-day vulnerabilities can leave computer systems vulnerable to targeted attacks that can cause unpredictable damage. Given in [6] In thisrespect, Heartbleed is an attack that takes advantage of a vulnerability, whereas Zeus is a threat. as they can be used to launch attacks on government agencies and critical infrastructure. As a result, zero-day exploits can be considered both a threat (if they cannot be detected) and a vulnerability (if they do not require additional support to remain hidden).

As from [7, 8, 11] There are various markets for zero-day vulnerabilities, and their value can vary depending on how unique and undetectable they are. These markets can be broadly divided into three categories: white markets, black markets, and grey markets.

White markets are legitimate, open markets where information technology companies pay researchers for zero-day vulnerabilities that they discover. Researchers may sell their discoveries to these companies rather than disclosing them publicly, as the value of a zero-day vulnerability can drop rapidly once it is made known.

Black markets are illegal markets where a variety of illegal goods and services, including zero-day vulnerabilities, are traded. These markets can operate both online and in person and in addition to zero-day vulnerabilities, may entail the sale of unlawful products and services.

Grey markets, also known as government markets, are less well known but are believed to exist in some countries. Some countries are suspected of acquiring zero-day vulnerabilities not for self-defense but rather for offensive objectives.. In some cases, these governments may even train prospective hackers and recruit the most talented ones to work for theirintelligence agencies.

The study in [9] concentrated on supervised and unsupervised learning techniques and introduced the Encrypted Zero-day Applications Classification (EZAC) method to increase the precision of classified encrypted data. To first categorise the encrypted flows, CNN is employed. A flow will be sent into the K-Means classifier if the classification result indicates that it is a zeroday application. The zero-day flows are divided into many categories by the K- Means classifier and then manually labelled.

Finally, these newly tagged flows are added to the CNN classifier.

There has been a significant amount of research in [10, 12] systems have been created with the focus of identifying malicious web traffic. Many NIDS and WAFs are rule-based, which means they look for patterns in existing attacks to spot new ones.

However, by [13], This method's drawback is that it can only identify attack types that have been observed before and for which signatures have been made. On the other side, anomaly- based systems do not rely on previously created signatures and can thus recognise new kinds of attacks, such as zero-day exploits. So, there is chance that these models will have less amount of precision during the detection phase.

A thorough method for analysing any new executables that enter the system and determining whether or not they are malware was proposed in [14]. The executable must go through three phases of this system before it canbe analysed and detected. The traditional signature and pattern matching method is used in the first stage. The machine learning method,which is used in the second phase,

To determine if the executable is malicious or benign, different characteristics and standards are used. In the third phase, a few sandbox toolsare employed to create a virtual secure environment to run the executables. The executables are then categorised based on the records of their different runtime readings.

Malware poses a serious risk to the privacy and data integrity of any computer system. The goal of research article [15] is to examine the currentdetection and analysis methods and create systems that can recognise the most recent dangers. This study primarily focuses on malware that exploits zero-day vulnerabilities and crypto jackers. Techniques used to detect malware include Completely automated analysis, static analysis, Interactive behavior analysis and Code reversal

III. DETECTION TECHNIQUES FOR Zero-DAY

To combat cyber attacks, many security solutions now in use rely on data gathered by the attackers themselves. Network-based security systems like intrusion detection systems and firewalls are examples of these types of solutions. However, these solutions have limitations because they are based on historical data provided by the attackers, and may not be effective at protecting against new or previously unknown attacks.

In order to be more effective at preventing attacks, organizations should adopt a proactive, objective, and passive strategy that offers defence against both known and unidentified threats without need for frequent updates or patches. Figure 1 shows the different malware detection techniques.

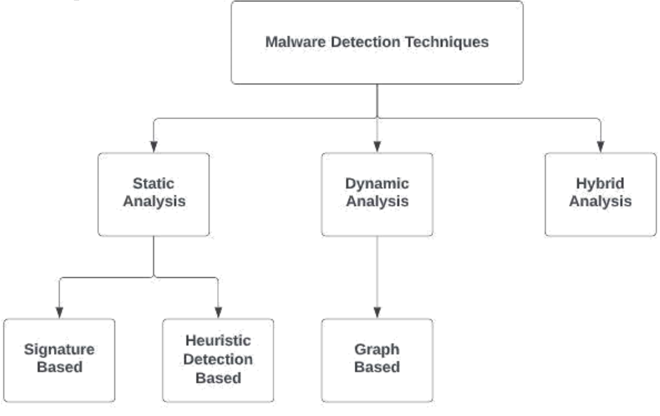

Figure 1 Malware Detection Techniques

3.1 Static Analysis:

Software testing without really running it. Basic static analysis looks at malware without looking at the source code or manual. File name, file size, file version, MD5 checksums or hashes, and file type are all examples of basic static analysis.

Types of Static Analysis:

a. **Signature-based detection technique:** This method is comparable to fingerprinting or pattern-matching masks.

b. **Heuristic based detection technique:** Heuristic detection is very similar to signature-based detection, except instead of looking for a particular signature, it looks for a predetermined set of instructions or commands within a programme that are typically absent from application programmes

3.2 Dynamic Analysis

Running the malware in a controlled setting, such as a sandbox, allows for dynamic analysis in order to observe and comprehend its behaviour and operation as well as to find technical indicators that can be incorporated into detection signatures

Types of Dynamic Analysis:

a. **Graph-Based analysis techniques:** This approach builds a similarity matrix between the instruction trace graphs using a combination of graph kernels

3.3 Hybrid Analysis

Both static analysis and dynamic analysis are combined in this method. Inorder to do static analysis, Hybrid Analysis stores fine-grained memory dump snapshots of the monitored runtime processes

IV. PROPOSED METHODOLOGY

We will attempt to provide a hybrid analysis system that simultaneously conducts static analysis and dynamic analysis.. First, we need to perform a static analysis to identify existing signatures. You should carry out dynamic analysis in a controlled setting, such as a sandbox, to acquire a thorough report on the behaviour of your sample. Using classification algorithms from the WEKA library, classification models are created using these traits. The system appears as shown in figure 2.

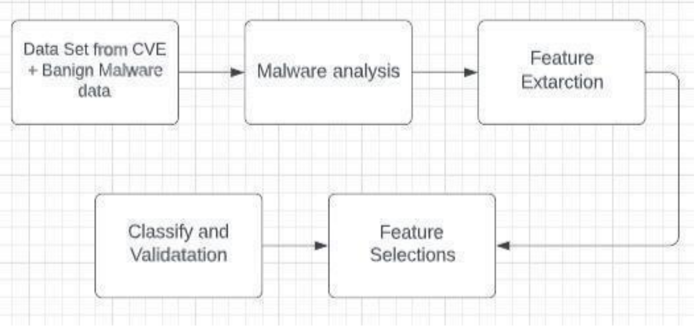
Data Set from CVE + Banign Malware data
Malware analysis
Feature Extarction
Feature Selections
Classify and Validatation

Figure 2. Malware Detection Flowchart

There are several methods that have been proposed for detecting similarities in vulnerable code and identifying patterns of vulnerabilities. Traditional token-based solutions eliminate whitespace and comments, swap out specified characters for variable and function names, and replace change identifiers, comments, and spaces with special characters.

These techniques can be used to detect rare types of code clones, such as type 1 and type 2 clones. Other approaches, For removing patterns from weak code and searching the code for these patterns, techniques like machine learning and deep learning have been suggested. For example, VulDeepecker employs a trained neural network to identify buffer overflows and resource management flaws brought on by library/API calls, while VulPecker uses a specific set of features to identify various software vulnerabilities.

The four stages of our strategy are information gathering, monitoring, decision-making, and feedback. We identify crucial assets throughout the information gathering stage and produce a reachability graph. The monitoring step makes sure that these essential assets' security standards are not disregarded. If an attempt is made to violate these requirements, the decision phase determines how to handle this attempt. Finally, the feedback phase sends overall information about the detection to the network layer.

A related reference suggests some very good methods for detecting and preventing zero-day exploits, but research has shown that none are very effective against typical attack data. Most of them tried single computer-based data they tried attacking their localhost. So, this is a type of attack where the attackers themselves collect data. As per our proposed method, we are trying to establish a system that will detect the exploit. There are four phases in the system, information gathering, monitoring, decision- making, and feedback

a. Information Gathering Phase:

This phase's goal is to list all essential assets along with any security requirements they may have. The owner of the asset or assets must provide this information. The IT database, for instance, may be considered the most important asset in any context including law firms. The firm's IT department management unit does this identification.

The owner of the asset(s) must once more specify the security need for each asset after it has been identified. In our scenario, the IT department of the company might merely need database integrity. The Firm is still able to issue transcripts and certifications of the cases as long as integrity is upheld. For the IT database of the law firm, availability and confidentiality could be just as crucial as integrity

b. Information Collection Phase:

The low-level connections between all other system objects and key assets are depicted in a reachability graph. For instance, the team manager might submit a modification code after receiving a verification code in order to edit an IT database. All activities and documents involved in this cycle have been recorded and included as critical assets. The reachability graph's vertices indicate significant items like activities and documents that are linked together depending on their interrelationships. This low-level capture, or creation of reachability diagrams, is accomplished by listening for system calls that indicate connections between significant items

c. Monitoring Phase:

Assuring that there are no attempts to circumvent the critical assets' security standards is the aim of this phase. A flag is raised and the "decision phase" is informed if such an attempt is made. In contrast to previous strategies, including [ZeroWall], our monitoring phase is distinct. Monitoring is carried out in [ZeroWall] to compile data on active programmes and processes. This c-is using Digital signatures. Instead of monitoring processes, we monitor assets at this phase. which we refer to as using machine detection as opposed to attacker detection. When the security criteria for vital assets are violated, we monitor them and alert the appropriate parties.

d. Decision Phase:

The Decision phase is the flagging phase of the system where any malpractice of the data in the monitoring phase is treated as the Red flag and will treat that asset as the exploit attack or if there is no malpractice detected then the green flag is treated as the safe trip for that regrading asset

To illustrate the system, here is an example similar to the proposed system. F1 and F3 files are recognised as significant assets with both file integrity and security requirements during the information identification process. To list every process that has the capacity to alter these two files, a reachability chart is created. Figure 3 and the reachability graph both illustrate this.

P1 can change F1,

P2 can alter F2,

P3 can enhance F3.

Any attempt by P5 to access F1 during the observation phase is regarded as a violation, and the decision phase receives a flag for it. Take note that the graph’s edges can be given weights. These weights can stand in for criteria such as access frequency, access time, access duration, and more.

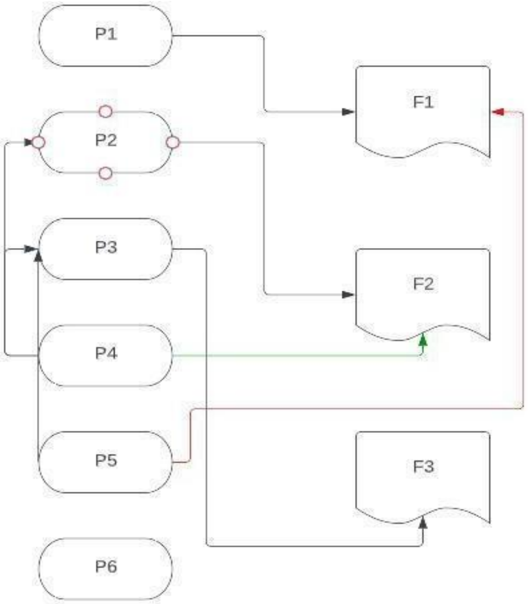

Figure 3. Proposed File System in Detection

The method we utilised benefits from DECAF's JIT VMI, which enables run-time modifications for the guest operating system. The plugin can always be loaded to produce the desired outcomes. In addition, our method gives malware analysts useful details about the targeted system call by requesting the system call parameters for additional investigation. The method we employed also permits system-wide API hooking by tracking all recently developed procedures. More crucially, it is invisible to the guest operating systems, making it challenging for processes that are already running to figure out if they are being watched

.V. **CONCLUSION**

Our research shows that market forces and cybercriminal activity related to zero-day vulnerabilities are driving research into zero-day vulnerabilities. It turns out that the goal is to maximize economic gain or opportunistically.

exploit weak points in government or non- military cybercrime situations to outwit rivals strategically. rice paddies Instead, military cybercrime makes significant investments in thecreation of sophisticated and covert assaults thatare intended to impact certain targets. We have been able to create a rather accurate and comprehensive image of the three markets known as the white, black, and government (grey) markets thanks to the data from the poll.

Given information gathered and the first-hand testimony of the interviewees, one question remains unanswered and may be subject to further speculation. Are there zero-day vulnerabilities in the intricate geopolitical environment that defines global military,political, and economic conflicts where cyberweapons are employed? However, if this trend continues, the distinction between cybercrime, cyberwarfare and legal 0-illegal cyberweapons could become even more confusing than it is today.

Internet users' computers and the internet both face serious threats from sophisticated viruses. Traditional antivirus software can only identify malware that has already caused harm and has been registered as malware. This research article describes the various types of malwares. Thus, it is necessary to modify machine learning algorithms in order to utilise their full potential in handling cybersecurity-related problems and threats.

REFERENCES

[1] Z. He, T. Miari, H. M. Makrani, M. Aliasgari, H. Homayoun and H. Sayadi, "When Machine Learning Meets Hardware Cybersecurity: Delving into Accurate Zero-Day Malware Detection," 2021 22nd International Symposium on Quality Electronic Design (ISQED), Santa Clara, CA, USA, 2021, pp. 85-90, doi: 10.1109/ISQED51717.2021.9424330.

[2] O. Belej and L. Halkiv, "Using Hybrid Neural Networks to Detect DDOS Attacks," 2020 IEEE Third International Conference on Data Stream Mining & Processing (DSMP), Lviv, Ukraine, 2020, pp. 61-66, doi: 10.1109/DSMP47368.2020.9204166.

[3] J. H. Jeong and S. G. Choi, "Hybrid System to Minimize Damage by Zero-Day Attack based on NIDPS and HoneyPot," 2020 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea (South), 2020, pp. 1650-1652, doi: 10.1109/ICTC49870.2020.9289589.

[4] R. Tang *et al.*, "ZeroWall: Detecting Zero-Day Web Attacks through Encoder-Decoder Recurrent Neural Networks," *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, Toronto, ON, Canada, 2020, pp. 2479-2488, doi: 10.1109/INFOCOM41043.2020.9155278.

[5] N. Innab, E. Alomairy and L. Alsheddi, "Hybrid System Between Anomaly Based Detection System and Honeypot to Detect Zero Day Attack," *2018 21st Saudi Computer Society National Computer Conference (NCC)*, Riyadh, Saudi Arabia, 2018, pp. 1-5, doi: 10.1109/NCG.2018.8593030.

[6] B. I. Hairab, M. Said Elsayed, A. D. Jurcut and M. A. Azer, "Anomaly Detection Based on CNN and Regularization Techniques Against Zero-Day Attacks in IoT Networks," in *IEEE Access*, vol. 10, pp. 98427-98440, 2022, doi: 10.1109/ACCESS.2022.3206367.

[7] K. Sornalakshmi, "Detection of DoS attack and zero day threat with SIEM," *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, Madurai, India, 2017, pp. 1-7, doi: 10.1109/ICCONS.2017.8250515.

[8] INNOCENT MBONA AND JAN H. P. ELOFF, "Detecting Zero-Day Intrusion Attacks Using Semi-Supervised Machine Learning Approaches" IEEE Access VOLUME 10, 2022, pp: 69822-69838 doi: 10.1109/ACCESS.2022.3187116

[9] Y. Li, Y. Lu and S. Li, "EZAC: Encrypted Zero-day Applications Classification using CNN and K-Means," *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, Dalian, China, 2021, pp. 378-383, doi: 10.1109/CSCWD49262.2021.9437716.

[10] S. Kumar and C. Bhim Bhan Singh, "A Zero-Day Resistant Malware Detection Method for Securing Cloud Using SVM and Sandboxing Techniques," *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, Coimbatore, India, 2018, pp. 1397-1402, doi: 10.1109/ICICCT.2018.8473321.

[11] S. -J. Bu and S. -B. Cho, "Integrating Deep Learning with First-Order Logic Programmed Constraints for Zero-Day Phishing Attack Detection," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, ON, Canada, 2021, pp. 2685-2689, doi: 10.1109/ICASSP39728.2021.9414850.

[12] J. H. Sejr, A. Zimek and P. Schneider-Kamp, "Explainable Detection of Zero Day Web Attacks," *2020 3rd International Conference on Data Intelligence and Security (ICDIS)*, South Padre Island, TX, USA, 2020, pp. 71-78, doi: 10.1109/ICDIS50059.2020.00016

[13] H. Al-Rushdan, M. Shurman, S. H. Alnabelsi and Q. Althebyan, "Zero-Day Attack Detection and Prevention in Software-Defined Networks," *2019 International Arab Conference on Information Technology (ACIT)*, Al Ain, United Arab Emirates, 2019, pp. 278-282, doi: 10.1109/ACIT47987.2019.8991124.

[14] F. Abri, S. Siami-Namini, M. A. Khanghah, F. M. Soltani and A. S. Namin, "Can Machine/Deep Learning Classifiers Detect Zero-Day Malware with High Accuracy?," *2019 IEEE International Conference on Big Data (Big Data)*, Los Angeles, CA, USA, 2019, pp. 3252-3259, doi: 10.1109/BigData47090.2019.9006514.

[15] K. Radhakrishnan, R. R. Menon and H. V. Nath, "A survey of zero-day malware attacks and its detection methodology," *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*, Kochi, India, 2019, pp. 533-539, doi: 10.1109/TENCON.2019.8929620.