Borneo International Journal eISSN 2636-9826; Vol. 7 (3); 2024; 26-33
Published by *Majmuah Enterprise*
www.majmuah.com

Zero-Day Exploits and Vulnerability Management

IDAH PINDAI ZENGENI, MOHAMAD FADLI BIN ZOLKIPLI
School of Computing, Universiti Utara Malaysia
06010 Sintok Kedah, MALAYSIA
Email: idahz99@gmail.commail.com , m.fadli.zolkipli@uum.edu.my | Tel: +60142456304| Tel: +60177247779

Received: July 16, 2024
Accepted: July 23, 2024
Online Published: Sept 01, 2024

Abstract

Zero-day vulnerabilities pose significant threats to enterprise cybersecurity, exploiting unknown weaknesses in software before patches are available. This paper explores the lifecycle of zero-day exploits, from discovery through exploitation, disclosure, and patching, emphasizing the critical need for proactive vulnerability management. Case studies such as the Log4Shell and Microsoft Exchange vulnerabilities illustrate the devastating impact of these exploits on enterprise systems and data security. The discussion underscores the importance of early detection, swift response, and collaboration with software vendors to minimize exposure and mitigate risks effectively. Initiatives like bug bounty programs and responsible disclosure policies are highlighted as essential strategies for leveraging global expertise in identifying and addressing vulnerabilities. By embracing advanced detection technologies and continuous monitoring, organizations can enhance their resilience against evolving cyber threats and safeguard their digital assets. Ultimately, proactive cybersecurity measures and a collaborative approach are essential for mitigating the risks associated with zero-day vulnerabilities in today's dynamic threat landscape.

Keywords: Zero-day; vulnerability management; exploit; vulnerabilities; cybersecurity

1. Introduction

With the evolving complexity of technology, hackers and malicious parties are constantly seeking new ways to exploit systems. Zero-day exploits represent one of the most dangerous threats in the realm of cybersecurity due their capabilities of attacking systems with no existing defenses. Zero-day vulnerabilities are previously unknown vulnerabilities and bugs in operating systems, networks, and general software that create openings for external users or hackers to conduct illegal activities before patches have been released (Hamid et al., 2023 as cited in Parrend et al., 2018). It is referred to as "zero-day" because the developers or vendors have zero days to address the vulnerability once it's known (Patil & Shekokar, 2023). Traditionally cybersecurity measures relied on known signatures and established attack patterns but this alone is not adequate when dealing with zero-day exploits due to their unknown nature. Shifting to more proactive and preventative strategy has become increasingly important. Vulnerability management has become a critical component when dealing with zero-day exploits. Vulnerability management is as a component of information security management is concerned with minimizing risks associated with vulnerabilities (Nyanchama, 2005). This proactive approach helps in minimizing the window of exposure and reducing the potential impact of zero-day exploits.

2. Zero-day vulnerabilities in enterprise

2.1 Zero day vulnerabilitiy lifecycle

Enterprises, with their extensive and interconnected systems, are particularly vulnerable as attackers can exploit these flaws to access sensitive information, disrupt operations, and cause financial and reputational damage. Understanding the lifecycle of a zero-day vulnerability is crucial for enterprises to develop effective strategies for prevention, detection, and response. The lifecycle of a zero-day vulnerability consists of several key stages which are discovery, exploitation, disclosure and patching (Zaib & Zhou, 2022). The lifecycle of a zero-day vulnerability begins with the discovery phase. This stage involves the identification of a security flaw or zero-day exploit by researchers, hackers, or security professionals. The discoverer assesses the vulnerability's severity and potential impact, determining how it can be exploited. For enterprises, early detection is critical, as it allows for a quicker response to mitigate potential damage. During this phase, the vulnerability remains unknown to the vendor, leaving the software unprotected and susceptible to a zero-day attack. Following discovery, the exploitation phase occurs. In this stage, malicious actors develop and deploy exploits to take advantage of the vulnerability. They often create malware or other attack vectors to breach systems, steal data, or achieve other malicious objectives. For enterprises, this phase represents a significant threat, as attackers can infiltrate networks and cause extensive damage before any protective measures are in place. The rapid and stealthy nature

of exploitation underscores the importance of robust security practices and monitoring systems to detect unusual activities early on.

As the vulnerability is exploited, it eventually comes to light through detection by security researchers or through anomalies in system activities, leading to the disclosure phase. During this stage, the vulnerability is disclosed either to the vendor for responsible patching or publicly, depending on the discoverer's intentions. Responsible disclosure involves informing the vendor privately, allowing them to develop a patch before the vulnerability is widely known. Public disclosure can create urgency but also increases the risk as attackers may exploit the vulnerability before a patch is available. For enterprises, prompt disclosure and collaboration with vendors are essential to expedite the development of a fix. The final stage is patching, where the software vendor creates and distributes a patch to fix the vulnerability. This process can vary in duration based on the complexity and severity of the issue. Enterprises must prioritize applying the patch immediately to mitigate the associated risks. This phase involves updating software, systems, and potentially implementing additional security measures. Continuous monitoring remains necessary to ensure that the vulnerability is effectively neutralized. The patching phase highlights the importance of proactive and efficient responses to secure enterprise systems against zero-day vulnerabilities.

2.2 Detecting Zero-day vulnerabilities

To detect zero-day vulnerabilities there is myriad of methods ad tools available that can be utilized in an enterprise environment it may be required to combine these strategies due the complexity of their system. The strategies include:

Static analysis involves examining the source code of an application without executing it (Lenarduzzi et al., 2023) to identify potential security vulnerabilities. This technique relies on automated tools such as SonarQube, FindBugs, or Coverity Scan to scan the codebase for known patterns of vulnerabilities, like SQL injection, cross-site scripting (XSS), and buffer overflows (Hanif et al., 2021). These tools analyze the structure and syntax of the code to flag potential security issues before the software is run. In addition to automated tools, static analysis can also involve manual code reviews by experienced security professionals who can spot more nuanced issues that automated tools might miss. By addressing vulnerabilities at the code level, static analysis helps in early detection and remediation, reducing the risk of exploits in deployed applications.

Dynamic analysis examines an application during its execution to identify vulnerabilities that may not be evident through static analysis alone. This method also known as behavior analysis, involves executing infected files within a controlled environment to observe and analyze their behavior (Sihwail et al., 2019) which could detect anomalies that could indicate security weaknesses. Techniques such as Runtime Application Self-Protection (RASP) and behavioral analysis tools like AppDynamics or New Relic are used to monitor real-time application behavior. Dynamic analysis performs better than static analysis in detecting anomalies but its process requires longer runtimes (Cen et al., 2024). By simulating real-world attack scenarios, dynamic analysis helps uncover vulnerabilities that manifest only during execution, such as memory leaks, race conditions, and improper error handling.

Fuzz testing, or fuzzing, is a technique used to discover vulnerabilities by inputting large amounts of random or semi-random data (Cheng et al., 2024) into an application and observing its response. Automated fuzzing tools like AFL (American Fuzzy Lop), Peach Fuzzer, and Microsoft's Security Risk Detection are commonly used for this purpose. The goal is to trigger unexpected behaviors, such as crashes, hangs, or security breaches that indicate underlying vulnerabilities. Custom fuzzing scripts can also be developed to target specific protocols or application functionalities. Fuzz testing is particularly effective in identifying buffer overflows, input validation errors, and other security weaknesses that arise from unexpected input handling, providing valuable insights into the application's robustness.

Reverse engineering involves analyzing the binary code of an application to understand its structure, functionality, and potential vulnerabilities. This process typically uses tools like IDA Pro, Ghidra, and Binary Ninja (Zhang et al., 2024) to disassemble and decompile the software, allowing security experts to examine its internal workings. Reverse engineering is crucial for uncovering hidden vulnerabilities in proprietary software, firmware, and even hardware, as it reveals how the application processes data and interacts with the system. This method is often employed by both security researchers to find vulnerabilities and attackers to exploit them. By understanding the application's code at a low level, reverse engineering helps identify flaws that are not visible through source code analysis alone. Network traffic analysis involves observing network activities to identify specific patterns and extract valuable information from the network traffic. This process can be applied in various fields, including network asset probing and anomaly detection (Shen et al., 2023), that may indicate security vulnerabilities. Tools like Wireshark, tcpdump, and intrusion detection systems (IDS) such as Snort and Suricata are used to capture and scrutinize network packets. This method helps identify anomalies such as unusual data transfers, unexpected communication with external servers, or abnormal traffic volumes that could signify a zero-

day exploit or other malicious activity. By analyzing network traffic, enterprises can detect and respond to threats in real-time, protecting against data breaches, unauthorized access, and other network-based attacks. Network traffic analysis provides valuable insights into the security posture of an application and helps in early detection of potential security incidents. A comprehensive incident response and recovery plan is essential for effectively dealing with zero-day vulnerabilities. This plan should outline clear procedures for detecting, responding to, and recovering from security incidents. Key components include incident identification and classification, containment strategies, eradication processes, and system recovery methods. Regular drills and simulations help ensure that the incident response team is prepared to handle real-world scenarios. Effective incident response minimizes the damage caused by zero-day attacks, reduces downtime, and helps maintain business continuity. Additionally, post-incident analysis provides valuable insights that can be used to strengthen security measures and prevent future attacks.

2.3 Mitigating Zero-day vulnerabilities

With the threat of zero-day vulnerabilities, patching is of high priority. Regular updates and patching are essential for mitigating zero-day vulnerabilities as they help close security gaps that could be exploited by attackers. IT vendors must release patches promptly to address discovered vulnerabilities, and organizations should implement these patches as soon as they are available to minimize exposure. The release time of patches is crucial for IT vendors to deal with the difficulties of securing their products (Roumani, 2021). This requires an efficient patch management process, where systems and applications are routinely monitored for updates, and patches are tested and deployed systematically to avoid disrupting business operations. Staying current with patches ensures that known vulnerabilities are addressed, reducing the risk of exploitation. With the threat of zero-day vulnerabilities, patching is of high priority. When it comes to patching release time is important for IT vendors to deal with the difficulties of securing their products (Roumani, 2021). Advanced threat detection involves using sophisticated technologies and methodologies to identify and respond to potential security threats, including zero-day vulnerabilities. Solutions such as Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS), and Security Information and Event Management (SIEM) systems play a vital role in detecting abnormal activities and potential exploits. Machine learning and artificial intelligence can enhance these systems by identifying patterns and anomalies that traditional methods might miss (Olubudo, 2024). By continuously monitoring network traffic, system behavior, and application performance, advanced threat detection systems can provide real-time alerts and automated responses to mitigate the impact of zero-day attacks.

Endpoint security is critical in protecting individual devices such as laptops, desktops, mobile devices, and servers from zero-day vulnerabilities. Implementing robust endpoint security measures, including antivirus software, anti-malware tools, and Endpoint Detection and Response (EDR) solutions, helps to detect and block potential threats at the device level (Arfeen et al., 2021). EDR solutions, in particular, offer advanced capabilities such as real-time monitoring, threat hunting, and automated responses to suspicious activities. By securing endpoints, organizations can prevent attackers from exploiting vulnerabilities to gain access to sensitive data or propagate malware across the network. A comprehensive incident response and recovery plan is essential for effectively dealing with zero-day vulnerabilities. This plan should be systematic and outline clear procedures for detecting, responding to, and recovering from security incidents. Key components include incident identification and classification, containment strategies, eradication processes, and system recovery methods. Regular drills and simulations help ensure that the incident response team is prepared to handle real-world scenarios. Effective incident response minimizes the damage caused by zero-day attacks, reduces downtime, and helps maintain business continuity. Additionally, post-incident analysis provides valuable insights that can be used to strengthen security measures and prevent future attacks. Comprehensive training programs are vital for equipping employees with the knowledge and skills needed to recognize and respond to zero-day vulnerabilities. Security awareness training should be provided to all staff members to help them identify phishing attempts, social engineering tactics, and other common attack vectors. Technical training for IT and security personnel should focus on advanced threat detection, incident response, and the latest security best practices. Regular training sessions and updates ensure that employees stay informed about emerging threats and evolving attack techniques. A well-trained workforce is a critical line of defense against zero-day vulnerabilities, as human error is often a significant factor in security breaches.

3. Vulnerability management

Vulnerability management is a proactive security strategy aimed at identifying and addressing vulnerabilities throughout the product's lifecycle. The goal is to ensure that any security weaknesses are detected and remedied promptly, maintaining the integrity and safety of the product over time (Ding et al., 2019).

3.1 Vulnerability management lifecycle

The identification phase is the foundation of the vulnerability management lifecycle, focusing on the discovery of security weaknesses within an organization's systems, applications, and networks. This stage employs a combination of automated scanning tools, such as vulnerability scanners and intrusion detection systems, along with threat intelligence feeds to pinpoint potential vulnerabilities. Manual assessments, including penetration testing and code reviews, complement these automated efforts by providing a deeper, human-insight perspective. The goal is to create a comprehensive inventory of vulnerabilities that could be exploited by malicious actors, forming the basis for subsequent evaluation and remediation steps. Once vulnerabilities are identified, the assessment phase involves a detailed evaluation of each vulnerability's severity, exploitability, and potential impact on the organization. This stage prioritizes vulnerabilities based on factors such as the criticality of the affected systems, the sensitivity of the data involved, and the likelihood of exploitation. Tools like Common Vulnerability Scoring System (CVSS) scores and risk matrices are often used to quantify and rank the vulnerabilities. The outcome of this phase is a prioritized list of vulnerabilities that guides the organization's remediation efforts, ensuring that the most critical issues are addressed first to minimize risk.

The remediation phase is focused on addressing the identified and assessed vulnerabilities to mitigate potential risks. This involves applying patches, updates, or configuration changes to affected systems and applications. In some cases, remediation may also include implementing workarounds or temporary fixes if immediate patching is not feasible. Effective remediation requires coordination between various teams, including IT operations, security, and development, to ensure that fixes are applied without disrupting business operations. The objective is to eliminate or reduce the exposure of vulnerabilities, thereby strengthening the organization's security posture. After remediation efforts are implemented, the verification phase ensures that the applied measures have effectively resolved the vulnerabilities without introducing new issues. This involves re-scanning the environment and conducting additional tests to confirm that the vulnerabilities have been successfully mitigated. Verification also includes validating that the applied fixes do not negatively impact system functionality or performance. This step is crucial to maintain confidence in the security measures and to ensure that the organization remains protected against the identified threats. The monitoring phase is an ongoing process that involves continuously observing the environment for new vulnerabilities and assessing the effectiveness of applied remediation measures. This stage utilizes automated tools, security information and event management (SIEM) systems, and threat intelligence to detect emerging threats and vulnerabilities in real-time. Continuous monitoring enables organizations to stay ahead of potential risks by promptly identifying and addressing new vulnerabilities. It also ensures that previously applied remediation efforts remain effective and that the organization's security posture adapts to the evolving threat landscape.

4. Role of Bug Bounty Programs and Responsible Disclosure

As part of their cybersecurity practices, several organizations have begun implementing Bug Bounty Programs (BBP) and Responsible Disclosure (RD) policies to manage vulnerabilities (Ding et al., 2019).

4.1 Bug Bounty Programs

Typically organizations relied on internal security experts to handle their security operations and outsourced experts for vulnerabilities discovery (Akgul et al., 2020). However as of recent organizations have started to crowd source hackers to discover their system’s vulnerabilities. This approach allows for organizations to harness diverse expertise to be bug hunters (Sivagnanam et al., 2021).Popular platforms facilitating bug bounty programs include well-known names such as HackerOne, Bugcrowd, and Synack. These platforms connect organizations with a global community of ethical hackers who actively seek out and report vulnerabilities in exchange for financial rewards or recognition. To mitigate the risks associated with zero-day vulnerabilities, many organizations have adopted bug bounty programs and responsible disclosure policies. These initiatives encourage security researchers and ethical hackers to discover and report vulnerabilities in a controlled and ethical manner, allowing organizations to address flaws before they can be exploited maliciously. Bug bounty programs offer financial rewards to individuals who identify and report security vulnerabilities in an organization's software or systems. These programs leverage the skills of the global security research community, incentivizing the discovery of vulnerabilities that might otherwise go unnoticed. Companies such as Google, Microsoft, and Facebook have successfully implemented bug bounty programs, resulting in the identification and remediation of numerous critical vulnerabilities. Bug bounty programs not only enhance an organization's security posture but also foster a collaborative relationship with the cybersecurity community.

While bug bounty programs have proven to be an effective approach for identifying and mitigating vulnerabilities, there are notable challenges associated with this strategy. One of the most prominent challenges is the communication gap

between the bug hunters (security researchers) and the bug bounty program managers (Akgul et al., 2020). This communication barrier can lead to several issues, impacting the overall effectiveness of the program.

4.2 Responsible disclosure policies

Responsible disclosure policies outline the process by which security researchers report vulnerabilities to an organization (Kinis, 2018). This typically involves a period of confidentiality, during which the organization can develop and deploy a patch before the vulnerability is publicly disclosed. Responsible disclosure ensures that vulnerabilities are addressed promptly and reduces the risk of exploitation by malicious actors. Effective communication and collaboration between researchers and organizations are key components of successful responsible disclosure practices. Implementing bug bounty programs and responsible disclosure policies can significantly enhance an organization's ability to identify and mitigate zero-day vulnerabilities. By proactively engaging with the cybersecurity community and fostering an environment of cooperation, organizations can improve their defenses against emerging threats and reduce the likelihood of successful zero-day exploits.

5. Recent Zero-Day exploits

Enterprises face heightened vulnerability to zero-day exploits owing to the intricate and interconnected infrastructure of their systems. Recent data from 2023 reveals that zero-day vulnerabilities targeting enterprise technologies have outpaced the general rise in in-the-wild exploits, a trend that has gained momentum over the last five years (Stone et al., 2024). This surge underscores the escalating risk enterprises face from sophisticated cyber threats designed to exploit unknown vulnerabilities before they can be effectively defended against. In December 2021, the Log4Shell vulnerability (CVE-2021-44228) was discovered in the Apache Log4j 2 Java library, a widely used tool for logging error messages in Java applications (Pauley et al., 2023). This remote code execution (RCE) flaw allows attackers to send specially crafted strings to Log4j, which can then load and execute a malicious payload from an attacker-controlled server. The exploit leverages Log4j's JNDI (Java Naming and Directory Interface) feature, enabling attackers to remotely control any device running Java apps using Log4j for logging. Since Log4j is embedded in numerous popular programs, including Apple iCloud and Minecraft, the vulnerability put hundreds of millions of devices at risk (Ma et al., 2023). Despite being present since 2013, it was not exploited until 2021. The MITRE Common Vulnerabilities and Exposures (CVE) database assigned it the highest risk score of 10 out of 10 (NVD - CVE-2021-44228, n.d.). Following its discovery, a patch was quickly released, but during the peak of exploitation, security researchers observed more than 100 Log4Shell attack attempts per minute, highlighting the significant and widespread impact of this vulnerability.

In January 2021, four critical zero-day vulnerabilities (CVE-2021-26855, CVE-2021-26857, CVE-2021-26858, and CVE-2021-27065) were discovered in Microsoft Exchange servers, actively exploited in chained attacks (Narang, 2021). These vulnerabilities collectively enabled attackers to gain full control over targeted Exchange servers by bypassing authentication, executing arbitrary commands, and writing files to any location on the server. The SSRF vulnerability (CVE-2021-26855) allowed attackers to send arbitrary HTTP requests and authenticate as the Exchange server (Mitigate Microsoft Exchange Server Vulnerabilities | CISA, 2021), while the other vulnerabilities facilitated remote code execution and persistent backdoors. The widespread exploitation led to significant data exfiltration, ransomware installations, and other malicious activities, underlining the critical importance of immediate patching and robust security measures to protect vital infrastructure. In 2024, Google Chrome faced a series of at least eight zero-day vulnerabilities (The Hacker News, 2024), including the high-severity CVE-2024-4671, a use-after-free flaw in the Visuals component. This particular vulnerability allowed attackers to exploit dangling pointers left by freed memory, leading to arbitrary code execution. Other vulnerabilities affected critical components such as the V8 JavaScript engine (Smith, 2024), posing severe risks by enabling attackers to bypass security mechanisms and compromise user systems. Despite the quick identification and patching by the Google Chrome security team, the recurrence of such zero-day vulnerabilities in a widely used browser like Chrome emphasizes the ongoing challenges in securing web applications and the necessity for continuous vigilance and timely updates to mitigate emerging threats.

These cases vividly illustrate the severe risks enterprises face from these vulnerabilities. These incidents have shown how attackers can exploit unknown weaknesses to infiltrate and disrupt critical platforms, resulting in data breaches, operational disruptions, and substantial financial repercussions. As cyber threats continue to evolve in sophistication and frequency, it is imperative for enterprises to prioritize proactive cybersecurity measures and swift response strategies. By enhancing detection capabilities, implementing robust patching protocols, and fostering collaboration with security experts and software vendors, enterprises can better safeguard their systems and mitigate the impact of zero-day vulnerabilities effectively. This proactive approach not only strengthens their defense against emerging threats but also reinforces resilience in an increasingly complex digital landscape.

6. Conclusion

In conclusion, zero-day vulnerabilities represent a formidable challenge in today's interconnected digital landscape, exposing enterprises to unprecedented risks of data breaches, operational disruptions, and reputational damage. As technology evolves, so too do the tactics of malicious actors who exploit these unknown weaknesses in software systems. The lifecycle of a zero-day vulnerability—from discovery to exploitation, disclosure, and patching—highlights the critical need for enterprises to adopt proactive cybersecurity measures. By implementing robust vulnerability management frameworks, leveraging advanced detection technologies, and fostering collaboration through initiatives like bug bounty programs, organizations can enhance their resilience against emerging threats. Moving forward, the imperative lies in not only fortifying defences against zero-day exploits but also in cultivating a culture of vigilance and responsiveness within cybersecurity practices. Case studies such as the Log4Shell and Microsoft Exchange vulnerabilities underscore the urgency of timely detection and swift mitigation strategies. By prioritizing early detection, rapid response, and effective communication with software vendors, enterprises can mitigate the impact of zero-day vulnerabilities and safeguard their critical assets. Embracing these proactive measures is essential for navigating the evolving threat landscape with resilience, ensuring that organizations remain ahead of potential risks and disruptions in an increasingly digital world.

Acknowledgment

I would like to extend my gratitude to all members of the School of Computing who participated in this study. This study was conducted as part of the Hacking and Penetration Testing Project. I also wish to acknowledge the support provided by Universiti Utara Malaysia, which made this work possible.

References

Hamid, K., Iqbal, M. W., Aqeel, M., Liu, X., & Arif, M. (2023). Analysis of Techniques for Detection and Removal of Zero-Day Attacks (ZDA). In Communications in computer and information science (pp. 248–262). https://doi.org/10.1007/978-981-99-0272-9_17

Akshaya, S., & G, P. (2019). A study on Zero-Day attacks. Social Science Research Network. https://doi.org/10.2139/ssrn.3358233

Samuel, D. (2023). Zero-day Vulnerabilities: An In-depth analysis. ResearchGate. https://doi.org/10.13140/RG.2.2.12775.01445

Walshe, T., & Simpson, A. (2020). An Empirical Study of Bug Bounty Programs. 2020 IEEE 2nd International Workshop on Intelligent Bug Fixing (IBF). https://doi.org/10.1109/ibf50092.2020.9034828

Akgul, O., Eghtesad, T., Elazari, A., Gnawali, O., Grossklags, J., Votipka, D., & Laszka, A. (2020). The hackers' viewpoint: Exploring challenges and benefits of bug-bounty programs. In Proceedings of the 2020 Workshop on Security Information Workers (WSIW) (Vol. 20).

Sivagnanam, A., Atefi, S., Ayman, A., Grossklags, J., & Laszka, A. (2021). On the Benefits of Bug Bounty Programs: A Study of Chromium Vulnerabilities. https://www.semanticscholar.org/paper/On-the-Bene%EF%AC%81ts-of-Bug-Bounty-Programs%3A-A-Study-of-Sivagnanam-Atefi/cd5260de3e18acc98f4291ffe0128f38c70b027a

Maulani, I. E., & Anggraeni, R. (2023). Bug Bounty Hunting: A case study of successful vulnerability discovery and disclosure. Devotion, 4(8), 1735–1740. https://doi.org/10.59188/devotion.v4i6.486

Zaib, R., & Zhou, K.-Q. (2022). Zero-Day Vulnerabilities: Unveiling the Threat Landscape in Network Security. Mesopotamian Journal of Cybersecurity, 57–64. https://doi.org/10.58496/mjcs/2022/007

Mahajan, J. S. (2023). Identification of Zero-Day exploits. ScholarWorks. http://hdl.handle.net/20.500.12680/xw42ng79r

Deshpande, A., Patil, I., Bhave, J., Giri, A., Sable, N. P., & Chavan, G. T. (2023). Detection and Notification of Zero-Day attack to Prevent Cybercrime. 2023 4th International Conference for Emerging Technology (INCET). https://doi.org/10.1109/incet57972.2023.10170141

Williams, T. L. (2021). Cybersecurity: Zero-Day Vulnerabilities and Attack Vectors (Order No. 28315877). Available from ProQuest Dissertations & Theses Global. (2508000298). http://eserv.uum.edu.my/dissertations-theses/cybersecurity-zero-day-vulnerabilities-attack/docview/2508000298/se-2

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

Roumani, Y. (2021). Patching zero-day vulnerabilities: an empirical analysis. Journal of Cybersecurity, 7(1). https://doi.org/10.1093/cybsec/tyab023

Radhakrishnan, K., Menon, R. R., & Nath, H. V. (2019). A survey of zero-day malware attacks and its detection methodology. TENCON 2019 - 2019 IEEE Region 10 Conference. https://doi.org/10.1109/tencon.2019.8929620

Bompos, K. (2020). Development time of Zero-Day cyber exploits in support of offensive cyber operations. https://apps.dtic.mil/sti/pdfs/AD1126359.pdf

Kukutla, T. R. (2023). Exploring the depths of Zero-Day vulnerabilities. ResearchGate. https://www.researchgate.net/publication/376271277_Exploring_the_Depths_of_Zero-Day_Vulnerabilities

Teodorescu, C. A. (2022). Perspectives and reviews in the development and evolution of the Zero-Day attacks. Informatică Economică, 26(2/2022), 46–56. https://doi.org/10.24818/issn14531305/26.2.2022.05

Cuppah, D., & Hanumanthappa, M. (2020). Design and analysis of a hybrid security framework for Zero-Day Attack. ResearchGate. https://www.researchgate.net/publication/341423830

Regi, S., Arora, G., Gangadharan, R., Bathla, R., & Pandey, N. (2022). Case study on detection and Prevention methods in zero day attacks. 2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO). https://doi.org/10.1109/icrito56286.2022.9964873

Marbukh, V. (2023). Towards Security Metrics Combining Risks of Known and Zero-day Attacks: Work in Progress. NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium. https://doi.org/10.1109/noms56928.2023.10154439

Ķinis, U. (2018). From Responsible Disclosure Policy (RDP) towards State Regulated Responsible Vulnerability Disclosure Procedure (hereinafter – RVDP): The Latvian approach. Computer Law & Security Review, 34(2), 416-428. https://doi.org/10.1016/j.clsr.2017.11.003

Stone, M., Semrau, J., & Sadowsk, J. (2024). We're All in this Together: A Year in Review of Zero-Days Exploited In-the-Wild in 2023. Retrieved from https://storage.googleapis.com/gweb-uniblog-publish-prod/documents/Year_in_Review_of_ZeroDays.pdf.

Cen, M., Deng, X., Jiang, F., & Doss, R. (2024). Zero-Ran sniff: A zero-day ransomware early detection method based on zero-shot learning. Computers & Security, 142, 103849. https://doi.org/10.1016/j.cose.2024.103849

Roumani, Y. (2021). Patching zero-day vulnerabilities: an empirical analysis. Journal of Cybersecurity, 7(1). https://doi.org/10.1093/cybsec/tyab023

Ding, A. Y., Limon, D. J. G., & Janssen, M. (2019). Ethical hacking for boosting IoT vulnerability management. ICTRS '19: Proceedings of the Eighth International Conference on Telecommunications and Remote Sensing. https://doi.org/10.1145/3357767.3357774

Lenarduzzi, V., Pecorelli, F., Saarimaki, N., Lujan, S., & Palomba, F. (2023). A critical comparison on six static analysis tools: Detection, agreement, and precision. Journal of Systems and Software/ the Journal of Systems and Software, 198, 111575. https://doi.org/10.1016/j.jss.2022.111575

Sihwail, R., Omar, K., Ariffin, K. Z., & Afghani, S. A. (2019). Malware detection approach based on artifacts in memory image and dynamic analysis. Applied Sciences, 9(18), 3680. https://doi.org/10.3390/app9183680

Hanif, H., Nasir, M. H. N. M., Razak, M. F. A., Firdaus, A., & Anuar, N. B. (2021). The rise of software vulnerability: Taxonomy of software vulnerabilities detection and machine learning approaches. Journal of Network and Computer Applications, 179, 103009. https://doi.org/10.1016/j.jnca.2021.103009

Cheng, H., Li, D., Zhao, M., Li, H., & Wong, W. E. (2024). A Comprehensive Review of Learning-based Fuzz Testing Techniques. 2024 10th International Symposium on System Security, Safety, and Reliability (ISSSR). https://doi.org/10.1109/isssr61934.2024.00024

Shen, M., Ye, K., Liu, X., Zhu, L., Kang, J., Yu, S., Li, Q., & Xu, K. (2023). Machine Learning-Powered Encrypted Network Traffic Analysis: A Comprehensive survey. IEEE Communications Surveys and Tutorials/IEEE Communications Surveys and Tutorials, 25(1), 791–824. https://doi.org/10.1109/comst.2022.3208196

Olubudo, P. (2024). Advanced Threat Detection Techniques in IT Security: Exploring machine learning algorithms for identifying ResearchGate. https://www.researchgate.net/publication/380938538_Advanced_Threat_Detection_Techniques_in_IT_Security_Exploring_Machine_Learning_Algorithms_for_Identifying_Sophisticated_Cyber_Threats

Arfeen, A., Ahmed, S., Khan, M. A., & Jafri, S. F. A. (2021). Endpoint Detection & Response: A Malware Identification Solution. 2021 International Conference on Cyber Warfare and Security (ICCWS). https://doi.org/10.1109/iccws53234.2021.9703010

Pauley, E., Barford, P., & McDaniel, P. (2023). The CVE Wayback Machine: Measuring Coordinated Disclosure from Exploits against Two Years of Zero-Days. IMC '23: Proceedings of the 2023 ACM on Internet Measurement Conference. https://doi.org/10.1145/3618257.3624810

Ma, C., Bosack, M., Rothschell, W., Davis, N., & Garg, V. (2023, November). Wanted hacked or patched: bug bounties for third party OpenSource software components. Usenix. https://www.usenix.org/sites/default/files/opensourcebugbounty_login_final.pdf

NVD - CVE-2021-44228. (n.d.). https://nvd.nist.gov/vuln/detail/CVE-2021-44228

Narang, S. (2021, March 2). Day vulnerabilities in Microsoft Exchange server exploited in the wild. tenable. https://www.tenable.com/blog/cve-2021-26855-cve-2021-26857-cve-2021-26858-cve-2021-27065-four-microsoft-exchange-server-zero-day-vulnerabilities

Mitigate Microsoft Exchange Server vulnerabilities | CISA. (2021, July 19). Cybersecurity and Infrastructure Security Agency CISA. https://www.cisa.gov/news-events/cybersecurity-advisories/aa21-062a

The Hacker News. (2024). Update Chrome Browser now: 4th Zero-Day exploit discovered in May 2024. https://thehackernews.com/2024/05/google-detects-4th-chrome-zero-day-in.html

Smith, M. (2024, May 15). New Chrome Zero-Day vulnerability CVE-2024-4761: What you need to know and how to stay safe. Cyber and Fraud Centre - Scotland. https://cyberfraudcentre.com/new-chrome-zero-day-vulnerability-cve-2024-4761