Mesopotamian journal of Cybersecurity
Vol.2022, **pp**. 57–64
DOI: https://doi.org/10.58496/MJCS/2022/007; ISSN: 2958-6542
https://mesopotamian.press/journals/index.php/CyberSecurity

Mesopotamian
Journal of cybersecurity

Research Article

Zero-Day Vulnerabilities: Unveiling the Threat Landscape in Network Security

Raheela zaib[1],*, , Kai-Qing Zhou [2] ,

[1] *Punjab university gujranwala campus, Pakistan*

[2] *Deputy Dean of school of communication and electronic engineering, Jishou University, Jishou, Hunan, China*

ARTICLE INFO

Article History

Received 13 Aug 2022
Accepted 29 Nov. 2022

Keywords

Cybersecurity

Network Security

Threat

Zero-Day Vulnerabilities

ABSTRACT

Today's digital ecosystem is particularly vulnerable to attacks due to zero-day vulnerabilities. Attackers take advantage of these flaws, which software developers and security experts are unaware of, leading to catastrophic outcomes. Through an analysis of their lifetime, discovery approaches, exploitation methods, disclosure protocols, and patching strategies, this study tries to reveal the danger landscape of zero-day vulnerabilities. We examine the effects of zero-day vulnerabilities on network security and the difficulties that businesses confront by reviewing the relevant literature and discussing actual cases. We also investigate preventative measures and methods of defense, such as intrusion and anomaly detection systems and cutting-edge AI. We stress the need for responsible disclosure, prompt patching, and continued research to counter these difficult to detect dangers. In order to better understand zero-day vulnerabilities, implement countermeasures, and respond to them, this article is an invaluable resource for researchers, security experts, and organizations.

1. INTRODUCTION

Protecting our increasingly linked digital world is the responsibility of network security[1, 2]. Cybercriminals who want to take advantage of security holes in new technologies are a growing concern[3]. Zero-day vulnerabilities, which are defects in software or systems that have never been discovered before, are a particularly dangerous type of vulnerability. Since these flaws have not been made public, there is no way to repair them or protect against them. The goal of this article is to reveal the security landscape as it pertains to zero-day vulnerabilities in networks[4, 5]. We can better comprehend the impact of zero-day vulnerabilities on networked systems and devise effective defence methods if we have a thorough awareness of their nuances. In this study, we investigate the full spectrum of zero-day vulnerabilities, from discovery methodologies to attacker exploitation strategies, disclosure to patching, and potential preventative measures.

To lay a firm groundwork,[6] we give a thorough introduction to the topic and survey the existing literature on security flaws and dangers. In this setting, the unique characteristics of zero-day vulnerabilities and their impact on network security can be better understood. We stress the vital nature of resolving this issue by analysing historical examples and their outcomes. To fully appreciate the danger posed by a zero-day vulnerability, it is necessary to understand its development over time. There are a number of factors to consider at each stage of a vulnerability's lifecycle, from its initial discovery by security researchers or malicious actors to its final exploitation, disclosure, and patching by software makers. Unravelling this lifecycle will allow us to identify weak spots early[7, 8] on, assess the efficiency of existing defences, and develop better preventative measures.

Additionally, this study delves into the methods used by security researchers, such as vulnerability research forums, bug bounty programmes, and automated vulnerability scanners, to unearth zero-day vulnerabilities. At the same time, we explore the techniques that attackers use to take advantage of these openings, including exploit kits, targeted attacks, and advanced persistent threats (APTs)[9, 10]. Through this analysis, we are able to comprehend the complexity and possible damage of zero-day attacks on network safety. We also explore the murky world of security flaw disclosure and fixes. Cooperation between security researchers, software providers, and the broader security community is essential for responsibly disclosing zero-day vulnerabilities. We discuss the difficulties and ethical issues that arise throughout this procedure, and we stress the importance of applying fixes quickly to protect against zero-day exploits[11].

We give case studies of significant instances as illustrative examples of the real-world effects of zero-day vulnerabilities. These cases illustrate how organisations have successfully responded to assaults in the past and the lessons they've learnt from those experiences. By analysing these events, we can learn more about the nature of the current threat scenario. This paper's overarching goal is to teach readers everything they need to know about zero-day vulnerabilities and how they affect network safety. We add to the existing body of knowledge in this crucial domain by investigating their lifetime, discovery approaches, exploitation methods, disclosure processes, and patching strategies. Furthermore, we investigate defensive mechanisms and mitigation solutions, including both conventional and futuristic methods, such as AI. In order to properly protect networked systems from these elusive and powerful threats, businesses, researchers, and security professionals need a thorough grasp of the threat environment of zero-day vulnerabilities.

2. BACKGROUND AND LITERATURE REVIEW

2.1 Vulnerabilities and Security Threats

In today's interconnected world[12, 13], network security is of utmost importance. Protecting private information and maintaining the security of interconnected systems have emerged as critical concerns in light of our ever-increasing reliance on digital infrastructure. However, as new vulnerabilities and security threats arise, the landscape of network security is always changing.To gain unauthorised access, interrupt services, or compromise data, attackers can take advantage of vulnerabilities in software, systems, or protocols. Errors in the code, poor design, improper settings, or using an out of date version of the programme can all lead to vulnerabilities[14]. Unauthorised access, data breaches, malware infections, denial of service assaults, and social engineering are all examples of security concerns. Zero-day vulnerabilities are particularly problematic, but all vulnerabilities are a problem. When it comes to software security, zero-day vulnerabilities (often abbreviated as 0days) are the worst kind to encounter. There are no known patches or safeguards to prevent their exploitation due to a lack of prior information.

When discussing vulnerabilities[15, 16], the term "zero-day" refers to the amount of time from when the flaw was found and when the vendor was informed. Both security researchers and cybercriminals have the potential to uncover zero-day vulnerabilities. Researchers try to notify manufacturers and give them time to patch vulnerabilities via responsible disclosure practises, however attackers are free to exploit zero-days without restriction. Zero-day vulnerabilities can have a devastating effect on network safety. Targeted attacks, undetected system penetration, sensitive data exfiltration, and network compromise are just some of the ways that attackers might use zero-days to their advantage. These flaws can avoid detection by conventional security systems for long periods of time, putting businesses and individuals at risk of suffering devastating financial losses, damaging their reputations, and even experiencing physical harm to their facilities or vital infrastructure.

2.4 Existing Literature on Zero-Day Vulnerabilities

Researchers and security experts have invested a great deal of time and energy into studying zero-day vulnerabilities in recent years.[17, 18] The definition, characteristics, discovery methodologies, exploitation methods, disclosure processes, and patching strategies of zero-day exploits are all explored in depth over a wide range of published works. Existing research has shown that zero-day vulnerabilities are difficult to discover and remedy because of their uniqueness. From manual analysis to automated scanning and vulnerability research forums, all have been explored by researchers interested in zero-day discovery. Attackers' strategies for taking advantage of these holes, such as writing complex exploit code or incorporating it into targeted assaults, have also been investigated.

In addition, the literature has covered the best practises and ethical aspects related to disclosing zero-day vulnerabilities. To guarantee that patches are developed and released in a timely manner, researchers and vendors must work together as part of a responsible disclosure procedure. Researchers, vendors, and the larger security community's dynamic has been analysed to pinpoint areas for better communication and cooperation. In conclusion, there is a never-ending fight against security flaws and threats in network infrastructure. Since zero-day vulnerabilities are never made public, it is impossible to know whether or not they are being exploited. The definition, methods of detection, exploitation, disclosure, and patching of zero-day vulnerabilities are all discussed in detail in the vast body of literature available. This study intends to contribute to the understanding of zero-day vulnerabilities and their effect on network security by evaluating the existing body of knowledge on the topic.

3. ZERO-DAY VULNERABILITY LIFECYCLE

There is a well-defined cycle that zero-day vulnerabilities go through, beginning with discovery and ending with disclosure and patching. To fully appreciate the dangers posed by zero-day vulnerabilities and craft adequate countermeasures, it is essential to have a firm grasp on this lifecycle's stages. Here, we'll examine the various phases of the lifecycle and the factors that must be taken into account at each one.

- Discovery

At this point, a zero-day exploit has been discovered for the first time. Security researchers, independent vulnerability hunters, and even chance discoveries are all possible avenues via which this can occur. Researchers use a wide variety of methods, including manual analysis, fuzzing, reverse engineering, and code auditing, to find security flaws. Extensive testing, analysis of software behaviour, and the identification of anomalies that point to the presence of a vulnerability are common parts of the discovery process.

- Exploitation

An attacker can take advantage of a zero-day flaw in a system or network as soon as it is identified. Crafting exploit code or creating exploit kits that automate the process are both examples of exploit tactics. To obtain unauthorised access, execute arbitrary code, elevate privileges, or undertake other malicious operations, attackers exploit these vulnerabilities. There are two main ways in which zero-day vulnerabilities are exploited: individually or as part of larger malware operations with far-reaching effects.

- Disclosure

In the disclosure phase, a zero-day vulnerability is responsibly disclosed to the software vendor or other appropriate parties. Responsible disclosure is a common practise among security researchers, who wait until vendors have had a chance to fix a vulnerability before making it public. This procedure enables manufacturers to fix the flaw and safeguard their customers. The disclosure of zero-day vulnerabilities, however, presents moral and practical difficulties. Researchers need to tread carefully along the disclosure timeframe, avoiding early publication of the vulnerability and its potential for abuse.

- Patching

The term "patching" is used to describe the process through which software providers release updates or fixes to resolve security flaws. The moment a vendor learns of a zero-day flaw, they throw themselves into researching its effects. After that, they work on creating and testing a fix to close the security hole. Protecting users and avoiding threats requires prompt patching. When it comes to widely used software, however, patching can be difficult because it involves collaboration between the manufacturer and users to achieve widespread acceptance.

- Challenges and Considerations

There are many things to think about and difficulties to face throughout the zero-day vulnerability lifetime. Finding zero-day vulnerabilities can be a lengthy and difficult process for security researchers, necessitating in-depth technical knowledge and access to susceptible software or systems. Proactive measures, constant monitoring, and analysis of newly released software are common in the discovery process[19, 20].

There is a serious danger that attackers would exploit zero-day vulnerabilities. The longer attackers can make use of zero-day vulnerabilities before they are discovered and patched, the higher their return on investment will be. This means that defenders will have less time to identify threats and take countermeasures. Furthermore, because zero-days are unknown vulnerabilities, standard security practises may not be able to stop their use. When dealing with a zero-day vulnerability, responsible disclosure is essential. Ethical concerns for security researchers include striking a balance between protecting consumers and allowing vendors to create patches while still ensuring transparency and accountability. The window of opportunity for attackers can be reduced through timely and coordinated disclosure. The process of patching has its own unique difficulties. Delays in patch deployment are sometimes inevitable due to the modern software's complexity, the requirement for exhaustive testing, and the necessity for coordinated action between vendors and end users. Even after fixes are made available, some users may still be at risk if they fail to upgrade their computers immediately. To sum up, there are four main phases to the zero-day vulnerability lifecycle: discovery, exploitation, disclosure, and patching. Researchers, vendors, and end users all face new obstacles and considerations at each juncture. For effective defence methods, responsible disclosure, and rapid patching, knowledge of this lifecycle is essential.

4. TECHNIQUES FOR ZERO-DAY VULNERABILITY DISCOVERY

The identification of previously unknown security flaws is a vital first step in assessing the threat they pose to a network and developing countermeasures. In order to find these gaps in security, researchers use a wide variety of tools and approaches. Here we will discuss some of the most important methods for finding vulnerabilities with zero-day exploits.

Manual code analysis is a primary method for finding zero-day vulnerabilities. Researchers that specialise in software security spend a lot of time poring over source code. Examining the codebase to find examples of unsafe or careless coding, reasoning, or implementation is part of this approach. Researchers can find flaws in the code that were missed during development or testing by performing a thorough analysis.

Fuzzing is an automated method of finding software flaws; it is also known as fuzz testing and fuzzing. Fuzzing is a technique used to test software for unusual behaviour by deliberately introducing unpredictable, random, or faulty data into it. Researchers can find security problems including buffer overflows, format string vulnerabilities, and input validation flaws by exploiting the software with malicious data. Fuzzing has been shown to be useful in finding zero-day vulnerabilities in both commercial and open-source software, and it may be done with a wide variety of tools and frameworks. By analysing the source code or firmware of a device or piece of software, reverse engineering might reveal hidden features and functionality. Researchers in the field of cyber security often resort to "reverse engineering" to discover flaws in computer programmes that aren't immediately obvious. The produced code is a treasure trove of information for developers and security analysts alike, as it reveals previously unknown features, flaws, and vulnerabilities. Software's behaviour can be better understood by reverse engineering, and potential attack routes that zero-day vulnerabilities could use can be uncovered.

As a group effort, finding vulnerabilities like zero days through bug bounty programmes has become increasingly common. Companies and developers of software often compensate people who find and report security flaws in their products. Researchers can be incentivized to look for zero-day vulnerabilities and responsibly disclose them through bug bounty programmes, rather than abusing them for malevolent purposes. These initiatives encourage a community-based method of vulnerability discovery, which is useful for spotting security flaws and fixing them before they are exploited. Automated vulnerability scanners are programmes that routinely probe systems or programmes for security flaws. In order to identify potential security flaws, these scanners rely on specified rule sets or signatures. Indirectly, they can aid in the discovery of zero-day vulnerabilities despite being primarily intended to detect known vulnerabilities. Automated scanners can identify possible areas of concern by spotting aberrant behaviour or unexpected responses, which may suggest the presence of previously undisclosed vulnerabilities.

The identification of zero-day vulnerabilities relies heavily on cooperation and sharing of knowledge. There is a lot of communication and collaboration between the security research community, industry personnel, and the vulnerability research communities. Increasing the likelihood of finding zero-day vulnerabilities requires researchers to combine their efforts and resources. Collaboration is facilitated and an atmosphere favourable to zero-day discovery is fostered through the use of online platforms and forums specifically designed for vulnerability research. In conclusion, finding zero-day exploits requires a variety of methods. Unknown flaws can be found by a combination of methods, including manual code analysis, fuzzing, reverse engineering, bug reward programmes, automated vulnerability scanners, and cooperation through information sharing. Researchers generally use a variety of techniques to boost their chances of discovering zero-days, as each has its own advantages and disadvantages. To remain ahead of attackers and reduce the risks associated with zero-day vulnerabilities, ongoing research, innovation, and collaboration are necessary.

5. ZERO-DAY VULNERABILITY EXPLOITATION

Once a zero-day flaw is identified, it can be used by attackers to compromise systems, acquire access, or commit other forms of mischief. To successfully exploit a zero-day flaw, an attacker must have in-depth knowledge of not only the flaw but also the software or system being attacked and their motivations for doing so. Here, we explore the tactics and procedures hackers use to take advantage of zero-day flaws.

Attackers frequently utilise custom-made exploit programmes to take advantage of zero-day flaws. To exploit a vulnerability, attackers first study its characteristics, such as its triggering conditions and the susceptible component's behaviour, and then write targeted code to exploit it. This exploit code is designed to exploit the vulnerability and carry out nefarious tasks including executing arbitrary code, gaining elevated privileges, or leaking sensitive data. It takes a high level of technical expertise and familiarity with the inner workings of the targeted software or system in order to build exploit code. Attackers can create exploit payloads that take advantage of the vulnerability using a variety of computer languages, assembly code, and scripting languages. Customised exploit code makes it more challenging to identify and counteract vulnerabilities. Exploit kits are collections of exploits, often including previously unknown vulnerabilities, that come ready to use. Kits like these make it easy for attackers with no technical knowledge to exploit vulnerabilities, both known and zero-day. To maximise their impact, exploit kits frequently go after widely used programmes or plugins that are installed on a big number of computers.

Exploit kits are used to take advantage of zero-day vulnerabilities by scanning a user's system when they visit a compromised website or interact with malicious content. If an exploitable application is found, the exploit kit will automatically deliver the appropriate exploit. Exploit kits simplify zero-day exploitation, allowing attackers to simultaneously compromise a large number of systems. There is a strong correlation between zero-day vulnerabilities and targeted assaults and APTs. Attacks that are targeted at certain people, businesses, or industries are meticulously prepared and carried out. However, advanced persistent threats (APTs) are campaigns that are carried out covertly and over a long period of time by well-resourced threat actors for the purpose of espionage or data theft.

Targeted and APT attackers who use zero-day flaws frequently employ elaborate, multistep exploit chains. They do a lot of research on the target to learn about its infrastructure, software, and potential weak spots. Once a zero-day flaw is discovered, it becomes a powerful weapon in their hands. By using the zero-day vulnerability, the attacker can defeat the target's defences and get a foothold in the network, opening the door to further penetration and data exfiltration. Recently, zero-day exploits and vulnerabilities have been traded on underground markets. Intruders, including nation-state actors, intelligence organisations, and cybercriminal groups, can use these marketplaces to sell their zero-day exploits to prospective customers. Customers may use this weakness to spy on competitors, get sensitive information, or launch targeted attacks. By providing a robust economic ecosystem for exploiting zero-day vulnerabilities, zero-day marketplaces may encourage the discovery and development of additional zero-day exploits. The existence of such markets emphasises the importance of ethically responsible disclosure of zero-day vulnerabilities. There are major obstacles to preventing the exploitation of zero-day vulnerabilities. Traditional security methods, such as antivirus software or intrusion detection systems, may not detect or prevent the exploitation of these vulnerabilities since the suppliers are unaware of them. However, organisations can take preventative measures to lower the danger posed by zero-day flaws.arentheses). 3.5-inch disc drive" is an example of a commerce term that uses English units.

6. ZERO-DAY VULNERABILITY DISCLOSURE AND PATCHING

Disclosure of zero-day vulnerabilities[22, 23] and the subsequent deployment of patches are essential for protecting against such dangers. A vulnerability can only be fixed once software suppliers are made aware of it and given time to create and release patches. We discuss the process of disclosing zero-day vulnerabilities and the difficulties of releasing patches in a timely manner. Reporting security flaws to software makers and other interested parties in a reasonable and coordinated fashion is known as "responsible disclosure." Researchers that find zero-day vulnerabilities typically follow responsible disclosure procedures to give vendors time to fix the flaw before making the information public.

The responsible disclosure process typically involves the following steps:

Notification: The researcher notifies the affected vendor or relevant organization about the zero-day vulnerability. This notification includes detailed information about the vulnerability, its impact, and any relevant proof-of-concept code or examples.

Coordination: The researcher and the vendor work together to validate and understand the vulnerability. This may involve further communication, sharing of additional information, or collaboration to address any questions or concerns.

Patch Development: The vendor develops patches or security updates to address the vulnerability. This process can involve rigorous testing and quality assurance to ensure that the patches effectively mitigate the vulnerability without introducing any new issues.

Patch Release: Once the patches are ready, the vendor releases them to the public. This is typically accompanied by release notes or security advisories that detail the vulnerability, its impact, and instructions for users on how to apply the patches.

Public Disclosure: After a reasonable amount of time has passed to allow users to apply the patches, the researcher may publicly disclose the vulnerability. This ensures that users are aware of the issue and the importance of applying the available patches..

7. CASE STUDIES AND EXAMPLES

Stuxnet Worm:

The Stuxnet worm is a well-known example of a sophisticated cyber weapon designed to attack industrial control systems; in this case, those of Iran's nuclear facilities. Stuxnet, first discovered in 2010, is a sophisticated cyberweapon that exploits a number of previously unknown vulnerabilities simultaneously. Siemens SCADA systems were employed in Iran's nuclear programme, and the worm deliberately went after them.To get access to the centrifuge controls used in uranium enrichment, Stuxnet used zero-day vulnerabilities in Windows and Siemens' industrial software. The worm spread from its initial source via portable media and network sharing. The story of Stuxnet showed how devastating zero-day vulnerabilities can be to infrastructure, and how urgently we need strong defences and patches to ward off the most sophisticated of attacks.

WannaCry Ransomware:

In 2017[24], tens of thousands of computers were compromised all around the world by the WannaCry malware. WannaCry quickly propagated across networks because it took advantage of a zero-day vulnerability in the Windows SMB (Server Message Block) protocol. Users' files would be encrypted, making them inaccessible until a ransom was paid in Bitcoin. The attack disrupted operations and cost a lot of money at businesses in the healthcare, government, and financial sectors. The WannaCry attack was a prime example of the speed and devastation that may result from exploiting a zero-day vulnerability. It drove home the need for prompt patching and proactive defence measures to forestall or lessen the effects of such assaults.

These examples highlight the seriousness of zero-day vulnerabilities and the importance of implementing strong cybersecurity measures such as finding and reporting vulnerabilities and fixing them as soon as possible. Protecting against zero-day exploits requires constant monitoring, sharing of threat knowledge, and proactive defence tactics.

TABLE.I VULNERABILITIES BASED ON DIFFERENT CRITERIA

Category	Description
Software/Application	Vulnerabilities specific to software or applications
Operating System (OS)	Vulnerabilities targeting operating systems
Web Browser	Vulnerabilities affecting web browsers
Network/Protocol	Vulnerabilities in network protocols or services
Firmware	Vulnerabilities in embedded firmware
Hardware	Vulnerabilities related to hardware components
Mobile/Smart Devices	Vulnerabilities impacting mobile or smart devices
Industrial Control System	Vulnerabilities in industrial control systems
Cryptographic	Vulnerabilities related to cryptographic algorithms
Privilege Escalation	Vulnerabilities that allow unauthorized privilege escalation
Remote Code Execution	Vulnerabilities enabling remote execution of arbitrary code
Denial of Service (DoS)	Vulnerabilities leading to service disruption or unavailability
Information Disclosure	Vulnerabilities that expose sensitive information
Authentication Bypass	Vulnerabilities bypassing authentication mechanisms
SQL Injection	Vulnerabilities allowing SQL injection attacks
Cross-Site Scripting (XSS)	Vulnerabilities facilitating cross-site scripting attacks

8. CONCLUSION

Since attackers might gain an advantage by exploiting previously undisclosed flaws in software or systems, zero-day vulnerabilities constitute a serious danger to network security. This paper has covered a wide range of ground in its examination of zero-day vulnerabilities, from the nature of the threats they pose to the methods used to exploit them to the procedures for disclosing them and fixing them. Due to the dynamic nature of technology and the ever-evolving intelligence of attackers, it is essential to take a preventative and multi-pronged strategy to fixing zero-day flaws. Effective vulnerability identification and responsible disclosure rely heavily on cooperation between security researchers, software manufacturers, and the wider cybersecurity community. As a result, exploitable vulnerabilities can be patched faster and less frequently. In addition, businesses should set up intrusion detection and prevention systems, divide their networks into separate zones, and use secure coding practises to keep their data safe. To ensure the rapid adoption of patches and limit the danger of exploitation, timely patch management, user education, and awareness campaigns are essential. Several of the problems with zero-day vulnerabilities have been brought to light in this paper, including the need for coordinated responsible disclosure and rapid patching. To remain ahead of attackers and reduce the risks posed by zero-day vulnerabilities, the cybersecurity community must continuously do research, innovate, and collaborate. Keeping an eye out, adjusting to new risks, and keeping a proactive security stance are all crucial as technology develops. The resilience of organisations against zero-day vulnerabilities and the protection of network infrastructure and sensitive data can be improved through an understanding of the threat landscape, the use of effective vulnerability discovery techniques, the promotion of responsible disclosure practises, and the implementation of robust defence strategies. Finally, researchers, vendors, and end-users must all work together to fix zero-day security flaws. Our ability to thwart zero-day attacks and make the Internet safer for everyone will improve if we pool our resources, exchange information, and put security first.

Funding

None.

Conflicts of Interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors would like to express their gratitude to the Department of Data Science and Analytics, Fatoni University for their moral support. Please accept my sincere gratitude for the useful recommendations and constructive remarks provided by the anonymous reviewers.

References

[1] U. K. Singh, C. Joshi, and D. Kanellopoulos, "A framework for zero-day vulnerabilities detection and prioritization," *Journal of Information Security and Applications,* vol. 46, pp. 164-172, 2019.

[2] L. Ablon and A. Bogart, *Zero days, thousands of nights: The life and times of zero-day vulnerabilities and their exploits*. Rand Corporation, 2017.

[3] S. H. Abdullah and A. H. Ali, "Radio Frequency Radiation Power Density measurements at Mobile Base Stations in Alam City," *Eurasian Journal of Engineering and Technology,* vol. 11, pp. 157-166, 2022.

[4] Y. Roumani, "Patching zero-day vulnerabilities: an empirical analysis," *Journal of Cybersecurity,* vol. 7, no. 1, p. tyab023, 2021.

[5] A. H. Ali and M. Z. Abdullah, "A novel approach for big data classification based on hybrid parallel dimensionality reduction using spark cluster," *Computer Science,* vol. 20, no. 4, 2019.

[6] M. Albanese, S. Jajodia, A. Singhal, and L. Wang, "An efficient approach to assessing the risk of zero-day vulnerabilities," in *2013 International Conference on Security and Cryptography (SECRYPT)*, 2013, pp. 1-12: IEEE.

[7] Z. A. Mohammed, M. N. Abdullah, and I. H. Al Hussaini, "Predicting incident duration based on machine learning methods," *Iraqi Journal of Computers, Communications, Control and Systems Engineering,* vol. 21, no. 1, pp. 1-15, 2021.

[8] O. N. Al-Khayat, S. Y. Ameen, and M. N. Abdallah, "WSNs power consumption reduction using clustering and multiple access techniques," *International Journal of Computer Applications,* vol. 87, no. 9, pp. 33-39, 2014.

[9] S. A. Abed, A. H. Ali, O. A. Mohamad, and M. Aljanabi, "Reliability allocation and optimisation by using Kuhn-Tucker and geometric programming for series-parallel system," *International Journal of Computer Aided Engineering and Technology,* vol. 16, no. 4, pp. 488-496, 2022.

[10] S. A. Abed, M. S. Fiadh, and A. H. Ali, "Reliability Allocation and Optimization Problem for Waste Treatment Plant (WTP)," *Eurasian Research Bulletin,* vol. 5, pp. 6-13, 2022.

[11] M. N. Abdullah and K. E. Dagher, "Airborne Computer System Path-Tracking Based Multi-PID-PSO Controller Design," *International Journal of Intelligent Engineering and Systems,* vol. 14, no. 3, pp. 403-411, 2021.

[12] M. G. Yaseen, M. Aljanabi, A. H. Ali, and S. A. Abd, "Current cutting-edge research in computer science," *Mesopotamian Journal of Computer Science,* vol. 2022, pp. 1-4, 2022.

[13] Z. E. Kanoon, A. S. Al-Araji, and M. N. Abdullah, "Enhancement of Cell Decomposition Path-Planning Algorithm for Autonomous Mobile Robot Based on an Intelligent Hybrid Optimization Method," *International Journal of Intelligent Engineering & Systems,* vol. 15, no. 3, 2022.

[14] M. N. Abdulla, I. Al-Mejibli, and S. K. Ahmed, "An investigation study of hospital management information system," *IJARCCE*, vol. 6, pp. 406-411, 2017.

[15] A. S. Dawood and M. N. Abdullah, "Adaptive performance evaluation for SDN based on the statistical and evolutionary algorithms," *Iraqi Journal of Computers, Communications, Control & Systems Engineering (IJCCCE),* vol. 19, 2019.

[16] F. H. Faris, A. T. Humod, and M. N. Abdullah, "A comparative study of PI and IP controllers for field oriented control of three phase induction motor," *Iraqi J. Comput. Commun. Control Syst. Eng,* 2019.

[17] A. H. Ali, H. Kumar, and P. J. Soh, "Big Data Sentiment Analysis of Twitter Data," *Mesopotamian Journal of Big Data*, vol. 2021, pp. 1-5, 2021.

[18] Z. H. Salih, G. T. Hasan, M. A. Mohammed, M. A. S. Klib, A. H. Ali, and R. A. Ibrahim, "Study the effect of integrating the solar energy source on stability of electrical distribution system," in *2019 22nd International Conference on Control Systems and Computer Science (CSCS)*, 2019, pp. 443-447: IEEE.

[19] A.-H. A. Salih, A. H. Ali, and N. Y. Hashim, "Jaya: an evolutionary optimization technique for obtaining the optimal Dthr value of evolving clustering method (ECM)," *International Journal of Engineering Research and Technology,* vol. 11, no. 12, pp. 1901-1912, 2018.

[20] A. H. Ali and M. Z. Abdullah, "Recent trends in distributed online stream processing platform for big data: Survey," in *2018 1st Annual International Conference on Information and Sciences (AiCIS)*, 2018, pp. 140-145: IEEE.

[21] P. H. Barros, E. T. Chagas, L. B. Oliveira, F. Queiroz, and H. S. Ramos, "Malware-SMELL: A zero-shot learning strategy for detecting zero-day vulnerabilities," *Computers & Security,* vol. 120, p. 102785, 2022.

[22] M. Fidler, "Regulating the Zero-Day vulnerability trade: A preliminary analysis," *ISJLP,* vol. 11, p. 405, 2015.

[23] R. Kaur and M. Singh, "A survey on zero-day polymorphic worm detection techniques," *IEEE Communications Surveys & Tutorials,* vol. 16, no. 3, pp. 1520-1549, 2014.