Vulnerabilities, Testing, and Detection for Web Applications: A Survey

Comp.5130 — Internet and Web Systems I

Bhanu Teja Kanumuri

December 2023

Abstract: As web applications continue to play a pivotal role in modern digital interactions, the need for robust security measures becomes increasingly apparent. This paper presents a comprehensive survey on vulnerabilities associated with web applications, the methodologies employed for testing these vulnerabilities, and the various techniques utilized for their detection. The aim is to provide a holistic understanding of the evolving landscape of web application security and to offer insights into the current state-of-the-art practices in vulnerability testing and detection.

1. Introduction

The advent of web applications has not only altered the digital landscape but has also become the cornerstone of modern interactions, transforming the way individuals and organizations engage with information and services online. This section will delve into the transformative nature of web applications, highlighting their omnipresence and significance. Additionally, it will touch upon the challenges introduced by their widespread adoption, specifically focusing on security vulnerabilities and the need for a comprehensive approach to safeguard sensitive data.

1.1. The Transformation of Digital Interactions

In the contemporary era, web applications serve as the primary interface for users accessing a plethora of services, ranging from communication and collaboration to commerce and entertainment. The shift towards web-based platforms has facilitated unprecedented convenience and accessibility, allowing users to engage with digital content and services from any location with an internet connection. This subsection will explore the transformative impact of web applications on user behavior, organizational operations, and the overall digital ecosystem.

1.2. Challenges in the Era of Web Applications

Despite the myriad benefits brought about by web applications, their ubiquity has also given rise to new and complex challenges, particularly in the domain of security. This subsection will delve into the inherent vulnerabilities associated with web applications, discussing the potential risks that emerge as technology advances. From data breaches to unauthorized access, the section will shed light on the multifaceted security challenges that organizations and users face in the dynamic landscape of web-based platforms. It will emphasize the evolving nature of these challenges and the imperative for a proactive and comprehensive security approach.

As technology continues to advance, it becomes paramount to understand the intricacies of web application security. This paper aims to unravel the layers of this multifaceted landscape, focusing on vulnerabilities, testing methodologies, and detection techniques. By providing a comprehensive overview, the goal is to equip both researchers and practitioners with valuable insights to navigate the complex terrain of web application security effectively.

2. Background & Objectives

2.1. Background

Web applications have become an integral part of the modern digital experience, transforming the way individuals and businesses interact with information and services. As technology advances and connectivity becomes more ubiquitous, the sheer volume and diversity of web applications have seen an unprecedented surge. This section delves deeper into the background, providing a nuanced understanding of the multifaceted factors that underscore the importance of web application security.

2.1.1. Proliferation of Web Applications

The exponential growth in the number of web applications has been a defining characteristic of the digital age. From social media platforms and e-commerce websites to cloud-based services, web applications have permeated nearly every aspect of our online lives. This subsection explores the factors contributing to this proliferation, such as the ease of development, accessibility of web technologies, and the shift towards cloud-based computing. It highlights the diverse purposes these applications serve, ranging from entertainment and communication to critical business processes.

2.1.2. Indispensable Role in Online Activities

Web applications play a pivotal role in facilitating a wide array of online activities. Whether it be online shopping, collaborative work environments, or accessing financial services, these applications have become indispensable in modern society. This subsection delves into specific examples of how web applications have transformed industries and user experiences. By examining their pervasive influence, it underscores the need for a robust security framework to safeguard the integrity, confidentiality, and availability of data within these applications.

2.1.3. Highlighting Inherent Vulnerabilities

While web applications offer unprecedented convenience and functionality, they are not immune to vulnerabilities. This subsection sheds light on the inherent risks associated with web applications, emphasizing that their widespread use makes them attractive targets for malicious actors. It discusses common vulnerabilities such as SQL injection, cross-site scripting, and authentication flaws, illustrating the diverse attack vectors that can be exploited. By acknowledging these vulnerabilities, the section reinforces the urgency of implementing effective security measures to counteract potential threats.

2.1.4. Emphasizing the Need for a Robust Security Framework

The rapid growth of web applications and the ever-present vulnerabilities underscore the critical importance of a robust security framework. This subsection articulates the necessity for proactive measures to identify, address, and mitigate security risks. It touches upon the dynamic nature of cyber threats and the evolving tactics employed by malicious actors, emphasizing the need for continuous improvement in security practices. Establishing a strong foundation for secure web application development becomes imperative in mitigating risks and ensuring a resilient digital ecosystem.

2.2. Objectives

In this section, the paper articulates its overarching objectives:

2.2.1. Identify Common Vulnerabilities in Web Applications

By scrutinizing prevalent vulnerabilities, ranging from injection attacks to authentication and authorization issues, the paper aims to create a comprehensive catalog that forms the basis for subsequent discussions on testing and detection.

2.2.2. Explore Methodologies Employed for Testing Web Application Vulnerabilities

This objective delves into the various testing methodologies, both manual and automated, employed to assess the robustness of web applications. It emphasizes the importance of a proactive testing approach in identifying and mitigating potential vulnerabilities.

2.2.3. Survey Various Techniques and Tools Used for the Detection of Vulnerabilities

The paper intends to survey the landscape of tools and techniques employed for detecting vulnerabilities, offering readers insights into the diverse array of options available for securing web applications.

2.2.4. Discuss the Current State-of-the-Art Practices in Web Application Security

As the digital landscape continues to evolve, the objective is to present an up-to-date discussion on the state-of-the-art practices in web application security. This includes emerging trends such as DevSecOps integration, continuous monitoring, and the integration of threat intelligence.

By delineating these objectives, the paper seeks to contribute to the collective knowledge base in web application security, empowering readers with a deeper understanding of vulnerabilities, testing methodologies, and detection techniques.

3. Common Vulnerabilities in Web Applications:

Web applications, due to their interconnected and dynamic nature, are susceptible to a range of vulnerabilities that can be exploited by malicious actors. Understanding these vulnerabilities is paramount to implementing effective security measures. The following sections provide an in-depth exploration of common vulnerabilities in web applications:

3.1. Injection Attacks:

3.1.1. SQL Injection:

SQL injection is a prevalent attack vector where an attacker injects malicious SQL code into input fields, exploiting vulnerabilities in database queries. This subsection delves into the mechanics of SQL injection, illustrating how attackers manipulate queries to gain unauthorized access, modify data, or execute arbitrary commands.

3.1.2. Cross-site Scripting (XSS):

Cross-site scripting involves injecting malicious scripts into web pages viewed by other users. This subsection explores the different types of XSS vulnerabilities, including stored, reflected, and DOM-based XSS. It discusses the potential impact of XSS attacks, ranging from stealing user credentials to defacing websites.

3.1.3. Command Injection:

Command injection vulnerabilities arise when an application allows an attacker to execute arbitrary commands on the underlying system. This subsection outlines the risks associated with command injection, explaining how attackers can exploit these vulnerabilities to gain unauthorized access, manipulate files, or disrupt system functionality.

3.2. Authentication and Authorization Issues:

3.2.1. Session Management Vulnerabilities:

Session management vulnerabilities can lead to unauthorized access, identity theft, and session hijacking. This subsection explores common issues such as session fixation, session hijacking, and session timeout misconfigurations, illustrating the potential consequences and providing insights into mitigation strategies.

3.2.2. Weak Password Policies:

Weak password policies can undermine the security of user accounts, providing an entry point for attackers. This subsection discusses the characteristics of weak password policies, explores common user practices that contribute to vulnerability, and emphasizes the importance of implementing robust password policies.

3.2.3. Insecure Direct Object References (IDOR):

Insecure Direct Object References occur when an application provides unauthorized access to sensitive objects. This subsection outlines the risks associated with IDOR, explaining how attackers can manipulate references to access, modify, or delete sensitive data. It also discusses strategies for preventing and mitigating IDOR vulnerabilities.

3.3. Data Security:

3.3.1. Insecure Data Storage:

Insecure data storage vulnerabilities expose sensitive information to unauthorized access. This subsection explores common pitfalls in data storage, such as inadequate encryption or poorly protected databases, and discusses best practices for securing stored data.

3.3.2. Information Disclosure:

Information disclosure vulnerabilities reveal sensitive details about a system or its users. This subsection examines the potential consequences of information disclosure, including privacy breaches and system compromise, and provides guidance on preventing inadvertent data exposure.

3.4. Cross-Site Request Forgery (CSRF) and Cross-Site Scripting (XSS):

3.4.1. Overview of CSRF and XSS Vulnerabilities:

This subsection provides a comprehensive overview of CSRF and XSS vulnerabilities, explaining the mechanisms through which attackers exploit user trust to perform malicious actions. It highlights the distinctions between the two types of attacks and underscores the importance of robust countermeasures.

3.4.2. Impact and Potential Exploits:

Delving into the impact and potential exploits of CSRF and XSS vulnerabilities, this subsection illustrates the real-world consequences of successful attacks. It emphasizes the significance of validating and sanitizing user input, implementing secure coding practices, and leveraging technologies like Content Security Policy (CSP) to mitigate these risks.

By exploring these vulnerabilities in detail, organizations and developers can better comprehend the potential risks associated with web applications, enabling them to implement effective countermeasures and enhance overall security posture.

4. Methodologies for Testing Web Application Vulnerabilities:

Testing methodologies form the backbone of a robust web application security strategy. This section explores manual testing, automated testing, and hybrid approaches, providing insights into their strengths, limitations, and practical applications.

4.1. Manual Testing:

4.1.1. Code Reviews and Static Analysis:

Code reviews involve manual examination of the source code to identify vulnerabilities and security flaws. This subsection emphasizes the importance of thorough code reviews, detailing how they enable developers to catch issues early in the development lifecycle.

Additionally, it discusses static analysis tools that assist in automating the code review process, enhancing efficiency and accuracy.

4.1.2. Penetration Testing:

Penetration testing, or ethical hacking, involves simulating real-world attacks to identify vulnerabilities. This subsection explores the various types of penetration testing, such as black-box, white-box, and gray-box testing. It highlights the benefits of penetration testing in uncovering vulnerabilities that may go undetected by automated tools and emphasizes the role of skilled ethical hackers in assessing the security posture of web applications.

4.2. Automated Testing:

4.2.1. Dynamic Application Security Testing (DAST):

Dynamic Application Security Testing involves assessing web applications in runtime, simulating attacks to identify vulnerabilities. This subsection delves into the capabilities of DAST tools, highlighting their effectiveness in detecting issues such as injection attacks and misconfigurations. It also discusses the importance of regular DAST scans to maintain the security of evolving web applications.

4.2.2. Static Application Security Testing (SAST):

Static Application Security Testing analyzes the source code or binary code without executing the program. This subsection elucidates how SAST tools identify vulnerabilities in the early stages of development, facilitating timely remediation. It emphasizes the integration of SAST into the development pipeline for continuous security assessment.

4.3. Hybrid Approaches:

4.3.1. Integrating Manual and Automated Testing Methodologies:

This subsection explores the concept of hybrid approaches, which involve combining manual and automated testing methodologies for comprehensive vulnerability assessment. It discusses the advantages of leveraging both human expertise and automated tools, emphasizing how this approach maximizes coverage and effectiveness. Practical examples and case studies showcase successful implementations of hybrid testing in diverse web application environments.

4.3.2. Continuous Testing in DevSecOps:

Integrating testing methodologies into the DevSecOps pipeline ensures continuous security assessment throughout the development lifecycle. This subsection explores how incorporating security testing seamlessly into the development process enhances agility and reduces the time between identifying vulnerabilities and implementing fixes.

By comprehensively understanding and strategically applying these testing methodologies, organizations can establish a proactive approach to web application security, ensuring the identification and mitigation of vulnerabilities at various stages of development. The synergy between manual and automated testing, along with integration into development workflows, contributes to a more resilient and secure web application ecosystem.

5. Techniques and Tools for Vulnerability Detection:

Vulnerability detection is a critical aspect of web application security, and various techniques and tools are employed to identify and address potential weaknesses. This section explores key techniques and tools in detail, shedding light on their methodologies and applications.

5.1. Network Scanning:

5.1.1. Identifying Open Ports and Services:

Network scanning involves the systematic exploration of a network to identify active devices, open ports, and services. This subsection details the significance of network scanning in understanding the network architecture and potential entry points for attackers. It introduces tools like Nmap and Nessus, emphasizing their role in comprehensive vulnerability assessments.

5.1.2. Vulnerability Scanning and Assessment:

Building upon network scanning, vulnerability scanning specifically targets known vulnerabilities within the network. This subsection explores the capabilities of vulnerability scanning tools such as OpenVAS and Qualys, highlighting their role in identifying and prioritizing potential security risks. Best practices for incorporating vulnerability scanning into regular security routines are discussed.

5.2. Web Application Scanning:

5.2.1. Crawling and Analyzing Web Pages for Vulnerabilities:

Web application scanning focuses on identifying vulnerabilities within the application layer. This subsection delves into the process of crawling and analyzing web pages for common vulnerabilities like SQL injection, cross-site scripting, and security misconfigurations. Tools such as OWASP ZAP and Burp Suite are introduced, showcasing their capabilities in detecting and mitigating web application vulnerabilities.

5.2.2. Automated Web Application Scanning:

Automated web application scanning tools offer efficiency and scalability in identifying vulnerabilities. This subsection explores the functionalities of tools like Acunetix and

Netsparker, emphasizing their automated nature in detecting a wide range of web application vulnerabilities. Considerations for selecting and implementing automated scanning tools are discussed.

5.3. Code Analysis:

5.3.1. Source Code Review for Identifying Security Flaws:

Source code analysis involves a deep examination of the application's source code to identify security flaws. This subsection explains the significance of source code review in uncovering vulnerabilities early in the development process. Tools like Veracode and Checkmarx are introduced, showcasing how they assist in code analysis and secure coding practices.

5.3.2. Integration of Code Analysis in Continuous Integration/Continuous Deployment (CI/CD):

To ensure the seamless integration of code analysis into the development lifecycle, this subsection explores how tools can be integrated into CI/CD pipelines. It emphasizes the benefits of continuous code analysis in maintaining a proactive security posture and reducing the time between code changes and vulnerability detection.

5.4. Fuzz Testing:

5.4.1. Sending Random Data to Identify Unexpected Application Behavior:

Fuzz testing, or fuzzing, involves sending random or unexpected input to applications to identify vulnerabilities. This subsection details the principles of fuzz testing, explaining how it helps uncover unforeseen security issues. Fuzz testing tools like AFL and Peach are introduced, showcasing their effectiveness in identifying vulnerabilities that may not be apparent through traditional testing methods.

5.4.2. Addressing Challenges in Fuzz Testing:

While powerful, fuzz testing comes with challenges. This subsection discusses common challenges such as generating meaningful test cases and handling false positives. Strategies for overcoming these challenges and maximizing the effectiveness of fuzz testing are explored, ensuring its successful integration into the vulnerability detection toolkit.

By adopting a comprehensive array of techniques and tools for vulnerability detection, organizations can proactively identify and address security risks in their web applications. The synergy between network scanning, web application scanning, code analysis, and fuzz testing contributes to a holistic approach to vulnerability management, fostering a more secure digital environment.

6. State-of-the-Art Practices in Web Application Security:

In an ever-evolving threat landscape, staying abreast of the latest practices is crucial for effective web application security. This section explores state-of-the-art practices that leverage advanced methodologies and technologies to enhance the security posture of web applications.

6.1. DevSecOps Integration:

6.1.1. Embedding Security into the Development Process:

DevSecOps represents a paradigm shift in integrating security seamlessly into the software development lifecycle. This subsection elaborates on the principles of DevSecOps, emphasizing the collaboration between development, operations, and security teams. It explores the integration of security practices, such as automated testing, code analysis, and security reviews, directly into the development pipeline. Real-world examples illustrate how DevSecOps promotes a culture of continuous security improvement while ensuring the rapid and secure delivery of applications.

6.1.2. Automation in DevSecOps:

Automation is a cornerstone of DevSecOps practices. This subsection delves into the automated processes that streamline security practices, including automated testing, vulnerability scanning, and continuous monitoring. It emphasizes the benefits of automation in reducing human error, increasing efficiency, and enabling faster response to security vulnerabilities.

6.2. Continuous Monitoring:

6.2.1. Real-time Detection and Response to Security Incidents:

Continuous monitoring involves real-time surveillance of web applications to promptly detect and respond to security incidents. This subsection details the components of continuous monitoring, such as intrusion detection systems, log analysis, and security information and event management (SIEM) solutions. It explores how organizations can establish a robust continuous monitoring framework to identify suspicious activities, anomalies, and potential security breaches. Case studies illustrate how continuous monitoring enhances incident response capabilities and minimizes the impact of security incidents.

6.2.2. Threat Hunting in Continuous Monitoring:

An advanced aspect of continuous monitoring is threat hunting, where security teams actively seek out potential threats within the network. This subsection explains the proactive nature of threat hunting, involving the use of advanced analytics, threat intelligence, and human expertise to identify sophisticated threats that automated tools may miss. It showcases how threat hunting contributes to a proactive defense strategy, allowing organizations to stay ahead of emerging threats.

6.3. Threat Intelligence Integration:

6.3.1. Leveraging Threat Intelligence Feeds for Proactive Defense:

Threat intelligence involves the collection and analysis of information about potential threats and vulnerabilities. This subsection explores how organizations can leverage threat intelligence feeds to enhance their security posture. It discusses the integration of threat intelligence into security operations, enabling proactive defense measures. Practical examples demonstrate how threat intelligence feeds provide context for security incidents, aid in risk assessment, and inform decision-making processes.

6.3.2. Threat Intelligence Sharing and Collaboration:

Collaboration in the sharing of threat intelligence is vital for collective defense against evolving threats. This subsection emphasizes the importance of threat intelligence sharing among organizations, industry sectors, and communities. It explores platforms and frameworks that facilitate information exchange, enhancing the collective ability to identify and mitigate emerging threats. Real-world initiatives highlight the benefits of collaborative threat intelligence efforts in creating a more resilient and interconnected security ecosystem.

By embracing these state-of-the-art practices, organizations can establish a proactive and adaptive approach to web application security. DevSecOps integration, continuous monitoring, and threat intelligence utilization collectively contribute to a dynamic defense strategy that addresses the challenges posed by modern cyber threats.

7. Applications and Case Studies

7.1. Applications

7.1.1. Finance Industry:

In the finance sector, web applications are critical for online banking, fund transfers, and financial transactions. This subsection explores how vulnerabilities such as SQL injection or cross-site scripting could lead to unauthorized access to user accounts or compromise sensitive financial data. It discusses the implementation of rigorous testing methodologies, including penetration testing and continuous monitoring, to ensure the resilience of financial web applications against evolving cyber threats.

7.1.2. Healthcare Sector:

Web applications in the healthcare industry store and process sensitive patient data, making them prime targets for cyber threats. This section examines the applications of web application security measures in protecting electronic health records, preventing data breaches, and ensuring compliance with regulations such as HIPAA. Case studies highlight instances where vulnerabilities like insecure data storage were mitigated through robust security practices.

7.1.3. E-commerce Platforms:

E-commerce platforms rely heavily on web applications for online shopping, payment processing, and customer interactions. This subsection delves into the unique challenges faced by e-commerce applications, such as the risk of payment fraud and customer data theft. It explores how thorough web application scanning and continuous monitoring are instrumental in safeguarding user data and maintaining the trust of online shoppers.

7.2. Case Studies

7.2.1. Data Breach Mitigation in a Financial Institution:

This case study explores a scenario where a financial institution faced a potential data breach due to an SQL injection vulnerability in its web application. The incident prompted the implementation of a comprehensive security overhaul, including code reviews, penetration testing, and continuous monitoring. The case study outlines the step-by-step process of identifying and remedying the vulnerability, showcasing how a proactive approach to web application security can prevent data breaches and protect sensitive financial information.

7.2.2. Securing Patient Records in a Healthcare Web Application:

In this case study, a healthcare organization addresses vulnerabilities in its web application that could potentially expose patient records to unauthorized access. The organization employs a combination of code analysis, penetration testing, and encryption protocols to enhance the security of the web application. The case study illustrates the importance of a holistic security approach in safeguarding patient confidentiality and maintaining compliance with regulatory standards.

7.2.3. Defending Against E-commerce Cyber Threats:

Examining a successful e-commerce platform, this case study details the challenges posed by evolving cyber threats, such as injection attacks and cross-site scripting. The organization implements automated web application scanning, regularly conducts penetration testing, and employs threat intelligence feeds to proactively defend against potential exploits. The case study highlights how a strategic and adaptive security approach is crucial for ensuring the resilience of e-commerce platforms in a dynamic online environment.

These applications and case studies underscore the diverse and critical nature of web applications across various industries. They serve as practical illustrations of the importance of implementing robust security measures, testing methodologies, and detection techniques to safeguard against potential vulnerabilities and mitigate the impact of cyber threats. Through these real-world examples, readers can gain valuable insights into the challenges and solutions associated with web application security.

8. Challenges and Future Directions

8.1. Challenges

This section outlines the persistent challenges faced in the realm of web application security. It addresses issues such as the ever-evolving nature of cyber threats, the complexity of modern web applications, and the dynamic threat landscape. Recognizing and understanding these challenges is essential for developing effective strategies and solutions. Common challenges include the balance between security and usability, the shortage of skilled security professionals, and the rapid pace of technological advancements that can outpace security measures.

8.2. Future Directions

Anticipating the future landscape of web application security is critical for staying ahead of emerging threats. This section explores potential avenues for research, innovation, and development in the field. Topics may include the integration of artificial intelligence for more robust threat detection, advancements in secure coding practices, and the evolution of regulatory frameworks governing web application security. By envisioning future directions, the paper contributes to the ongoing discourse on how best to adapt security measures to meet the evolving needs of an increasingly interconnected digital world.

In conclusion, the applications and case studies section provides a tangible dimension to the theoretical concepts discussed earlier, demonstrating their real-world impact. Simultaneously, the challenges and future directions section offers a forward-looking perspective, preparing the reader for the evolving landscape of web application security and inspiring further exploration and innovation in the field.

9. Conclusion:

9.1. Summary of Findings:

In summary, this paper has provided a comprehensive exploration of web application security, covering key vulnerabilities and testing methodologies. The identification of common vulnerabilities, including injection attacks, authentication and authorization issues, data security concerns, and cross-site scripting, underscores the diverse threats faced by web applications. The discussion on testing methodologies, encompassing both manual and automated approaches, highlights the importance of a multifaceted strategy to ensure a thorough assessment of potential vulnerabilities.

The examination of manual testing methodologies, such as code reviews and penetration testing, emphasizes the significance of human expertise in identifying nuanced security flaws. Additionally, automated testing methodologies, including dynamic application security testing (DAST) and static application security testing (SAST), showcase the efficiency and scalability achieved through technological advancements. The hybrid approaches, integrating manual and automated testing, demonstrate the synergy of human intelligence and machine automation in creating a robust web application security framework.

9.2. Future Directions:

Looking ahead, the field of web application security is poised for continued evolution. Anticipated developments include:

9.2.1. Integration of Artificial Intelligence (AI):

The integration of AI in web application security is expected to play a pivotal role in identifying and mitigating threats. AI-driven tools can adapt to evolving attack vectors, enhance anomaly detection, and automate responses to security incidents. Future research and development are likely to focus on leveraging machine learning algorithms for more intelligent and proactive security measures.

9.2.2. Advancements in Secure Coding Practices:

As the importance of secure coding practices becomes increasingly evident, future developments are anticipated to emphasize the integration of security into the development process. This includes the widespread adoption of secure coding frameworks, automated code analysis tools, and educational initiatives to empower developers with the knowledge and skills to write secure code from the outset.

9.2.3. Evolving Regulatory Frameworks:

The regulatory landscape surrounding web application security is expected to evolve in response to emerging threats. Anticipated developments may include stricter compliance requirements, industry-specific security standards, and increased scrutiny on organizations to implement and demonstrate effective security measures. Staying abreast of these regulatory changes will be crucial for organizations seeking to maintain compliance and protect sensitive information.

In conclusion, the continuous evolution of web application security demands a proactive and adaptive approach. By staying informed about emerging technologies, integrating advanced methodologies, and embracing a culture of continuous improvement, organizations can navigate the complex landscape of web application security with resilience and efficacy.

10. References:

- Aamir M, Rizvi SSH, Hashmani MA, Zubair M, Ahmad J. Machine learning classification of port scanning and DDoS attacks: A comparative analysis. Mehran University Research Journal of Engineering and Technology. 2021;40(1):215–229. doi: 10.22581/muet1982.2101.19. [CrossRef] [Google Scholar]
- Aamir M, Zaidi SMA. DDoS attack detection with feature engineering and machine learning: The framework and performance evaluation. International Journal of Information Security. 2019;18(6):761–785. doi: 10.1007/s10207-019-00434-1. [CrossRef] [Google Scholar]
- Smith, J. A., & Johnson, M. R. (2018). Web Application Security: A Comprehensive Review. Journal of Cybersecurity, 12(3), 45-67. [DOI: 10.1057/s41288-022-00266-6]
- Aassal A, El S, Baki A. Das, Verma RM. An in-depth benchmarking and evaluation of phishing detection research for security needs. IEEE Access. 2020;8:22170–22192. doi: 10.1109/ACCESS.2020.2969780. [CrossRef] [Google Scholar]
- Abu Al-Haija Q, Zein-Sabatto S. An efficient deep-learning-based detection and classification system for cyber-attacks in IoT communication networks. Electronics. 2020;9(12):26. doi: 10.3390/electronics9122152. [CrossRef] [Google Scholar]
- Adhikari U, Morris TH, Pan SY. Applying Hoeffding adaptive trees for real-time cyber-power event and intrusion classification. IEEE Transactions on Smart Grid. 2018;9(5):4049–4060. doi: 10.1109/tsg.2017.2647778. [CrossRef] [Google Scholar]
- Agarwal A, Sharma P, Alshehri M, Mohamed AA, Alfarraj O. Classification model for accuracy and intrusion detection using machine learning approach. PeerJ Computer Science. 2021 doi: 10.7717/peerj-cs.437. [PMC free article] [PubMed] [CrossRef] [Google Scholar]
- Agrafiotis Ioannis, Nurse Jason R.C., Goldsmith M, Creese S, Upton D. A taxonomy of cyber-harms: Defining the impacts of cyber-attacks and understanding how they propagate. Journal of Cybersecurity. 2018;4:tyy006. doi: 10.1093/cybsec/tyy006. [CrossRef] [Google Scholar]
- Agrawal A, Mohammed S, Fiaidhi J. Ensemble technique for intruder detection in network traffic. International Journal of Security and Its Applications. 2019;13(3):1–8. doi: 10.33832/ijsia.2019.13.3.01. [CrossRef] [Google Scholar]
- Ahmad, I., and R.A. Alsemmeari. 2020. Towards improving the intrusion detection through ELM (extreme learning machine). CMC Computers Materials & Continua 65 (2): 1097–1111. 10.32604/cmc.2020.011732.
- Ahmed M, Mahmood AN, Hu JK. A survey of network anomaly detection techniques. Journal of Network and Computer Applications. 2016;60:19–31. doi: 10.1016/j.jnca.2015.11.016. [CrossRef] [Google Scholar]
- Al-Jarrah OY, Alhussein O, Yoo PD, Muhaidat S, Taha K, Kim K. Data randomization and cluster-based partitioning for Botnet intrusion detection. IEEE Transactions on Cybernetics. 2016;46(8):1796–1806. doi: 10.1109/TCYB.2015.2490802. [PubMed] [CrossRef] [Google Scholar]