e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:05/May-2024 Impact Factor- 7.868 www.irjmets.com

DEMYSTIFYING APPLICATION SECURITY: KEEPING APPLICATIONS SAFE

Akshay Sekar Chandrasekaran[*1]
[*1]Texas A&M University, USA.

DOI : https://www.doi.org/10.56726/IRJMETS56768

ABSTRACT

Application security has become very important for businesses as they depend more on software applications and the risk of hacking rises. In today's digital world, application security is very important. This paper talks about the most common threats, weaknesses, and best practices for protecting apps during the whole software development life cycle (SDLC). During the planning, design, development, testing, and deployment stages, the paper talks about how to add security measures to the SDLC. It also looks at different stages of security testing, like penetration testing and static code analysis, and stresses how important it is to integrate security into the software development lifecycle. This paper uses case studies and examples from real life to show what happens when application security isn't good enough and what happens when security is not integrated into the software development lifecycle. It also looks into the future of application security by talking about new trends like the usage of artificial intelligence in application security techniques. Application security principles are also used in Web Application Firewalls (WAFs), Identity and Access Management (IAM) tools, and penetration testing services, among other security goods and services. In the end, the paper stresses how important application security is for keeping usage of artificial intelligence in application security techniques in a digital world that is always changing.

Keywords: Cybersecurity Trends, Software Development Life Cycle (SDLC), Vulnerability Management, Threat Modeling, Secure Coding Practices

I. INTRODUCTION

In today's digital landscape, application security has become a paramount concern for organizations across all industries. As software applications grow increasingly complex and interconnected, the risk of cyberattacks continues to escalate [1]. According to the latest report by IBM, the average cost of a data breach has surged to $4.35 million in 2022, underscoring the critical importance of robust application security measures [2]. Web and mobile applications, in particular, have emerged as prime targets for malicious actors due to their widespread adoption and usage. The Verizon Data Breach Investigations Report 2022 [3] reveals that web applications were implicated in 42% of all data breaches, making them the most prevalent vector for cybercriminals to exploit.

The repercussions of application security breaches can be catastrophic for organizations. Beyond the immediate financial costs, such as incident response, legal expenses, and customer compensation, there are also significant indirect costs, including reputational harm, erosion of customer trust, and regulatory penalties [4]. The Ponemon Institute's Cost of a Data Breach Report 2022 indicates that lost business accounts for 40% of the total cost of a data breach, with an average price tag of $4.35 million [2].

The rapid acceleration of digital transformations and the shift to remote work due to the COVID-19 pandemic have further expanded the attack surface for cybercriminals [5]. This abrupt transition has led to increased reliance on cloud services, remote access solutions, and collaboration platforms, which can introduce new

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:05/May-2024 Impact Factor- 7.868 www.irjmets.com

vulnerabilities and risks if not adequately secured [6]. In this context, application security has become a top priority for organizations seeking to maintain customer trust and safeguard their digital assets. Developing a comprehensive security strategy that encompasses the entire software development life cycle (SDLC), from requirements gathering to deployment and maintenance, is crucial [7].

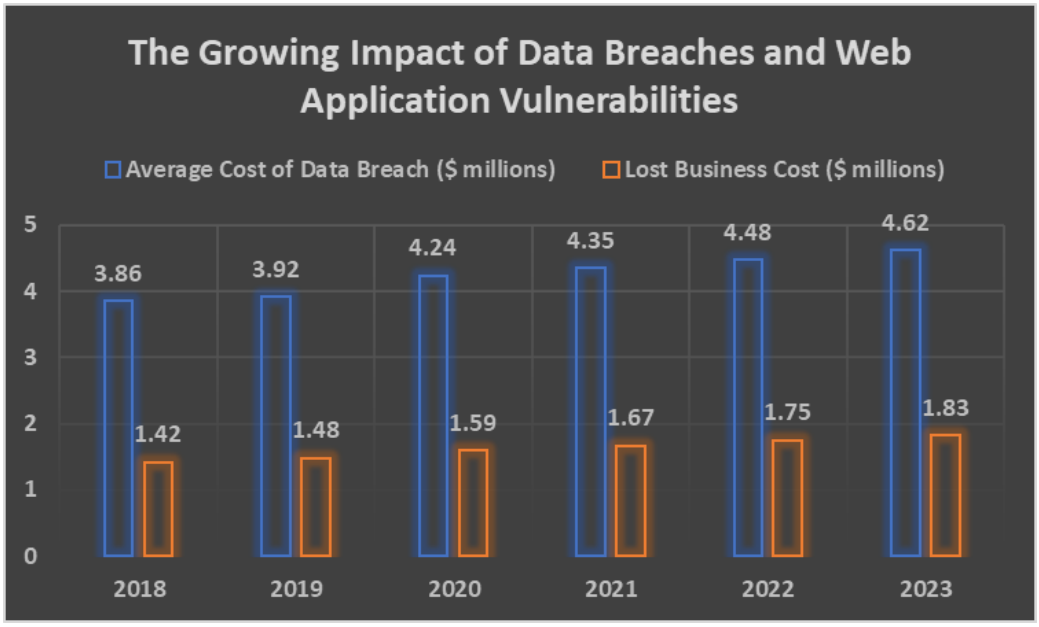

Fig. 1: Escalating Costs and Web Application Vulnerabilities: A Look at Data Breach Trends [1-7]

II. UNDERSTANDING APPLICATION SECURITY

Application security is the process of keeping software programs safe from attacks, unauthorized access, and data leaks [7]. Cross-site scripting (XSS), cross-site request forgery (CSRF), and SQL injection are common kinds of cyber threats that target applications [8]. In SQL injection attacks, bad SQL statements are inserted into the input areas of an application to get into the database without proper authorization [9]. The Open Web Application Security Project (OWASP) says that in 2021, injection flaws, such as SQL injection, will be the most common web application security risk [10].

When an attacker adds harmful scripts to a safe web app, the victim's computer runs them [11]. This is called a cross-site scripting (XSS) attack. These attacks can take over a user's session, steal private information, and make them do things they shouldn't [12]. Cross-site request forgery (CSRF) attacks take advantage of a user's established session to get them to do things they didn't mean to do on a web service [13]. If a business doesn't have good application security, the business could lose money, have its image hurt, or even face legal and regulatory problems [14].

III. VULNERABILITIES IN SOFTWARE APPLICATIONS

Software programs have flaws called vulnerabilities that attackers can use to get in without permission or do bad things [7]. OWASP says that some of the worst flaws in web applications are injection flaws, broken authentication, and private data being exposed [10]. Attackers often use automatic reconnaissance tools to look for and take advantage of these holes, so businesses need to find them and fix them before they get worse [15]. Organizations can find vulnerabilities in their software with the help of vulnerability tools like Nessus, Acunetix, and Burp Suite [16].

Organizations can use severity scoring systems, like the Common Vulnerability Scoring System (CVSS) [17], to decide which vulnerabilities need to be fixed first. CVSS is a standard way to rate how bad security vulnerabilities are by looking at things like how easy they are to attack, how they might affect privacy, availability, and integrity, and whether there are patches or workarounds available [18]. Organizations should not only find and fix vulnerabilities but also put in place security controls to lower the chance of exploitation [19]. Some of these controls include validating data, using parameterized queries, and writing code in a safe way [20].

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:05/May-2024 Impact Factor- 7.868 www.irjmets.com

Table 1: OWASP Top 10 Web Application Security Risks (2021) with CVSS Scores and Severity Levels [7, 8, 10]

OWASP Top 10 Web Application Security Risks (2021)	CVSS Score Range	Severity Level
A01:2021 - Broken Access Control	7.0 - 10.0	High
A02:2021 - Cryptographic Failures	5.0 - 7.9	Medium
A03:2021 - Injection	7.0 - 10.0	High
A04:2021 - Insecure Design	4.0 - 6.9	Medium
A05:2021 - Security Misconfiguration	4.0 - 6.9	Medium
A06:2021 - Vulnerable and Outdated Components	7.0 - 10.0	High
A07:2021 - Identification and Authentication Failures	7.0 - 10.0	High
A08:2021 - Software and Data Integrity Failures	5.0 - 7.9	Medium
A09:2021 - Security Logging and Monitoring Failures	4.0 - 6.9	Medium
A10:2021 - Server-Side Request Forgery (SSRF)	5.0 - 7.9	Medium

IV. INTEGRATING APPLICATION SECURITY INTO THE SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)

Adding security steps at every stage of the software development life cycle (SDLC) is necessary to create safe applications [7]. It is more cost-effective to add protection earlier in the software development process [21]. Clearly stating security goals and compliance needs should happen during the planning and requirements-gathering stage [22]. Establishing access control rules, finding sensitive data, and setting criteria for security testing are all part of this [23].

During the design and planning phases, to reduce vulnerabilities, secure coding techniques and design patterns should be used [24]. Potential security threats can be found and analyzed using threat modeling, which can help developers understand the attack area and come up with good defenses [25]. For security reasons, development and coding should follow secure coding standards and be reviewed regularly [26]. These steps include using secure coding tools, validating user inputs and encoding output, and staying away from common coding mistakes that can cause security vulnerabilities [27].

Complete security testing, like penetration testing and dynamic application security testing (DAST) [28], should be part of testing and validation. One can find security vulnerabilities in an application by looking at how it works while it's running, while penetration testing involves pretending to be an actual hacker. Lastly, operations and upkeep should include ongoing checking, fixing, and the ability to handle incidents [30]. This means the timely installation of security patches, keeping an eye out for security events and strange behavior, and having a clear plan for how to handle incidents [31].

Integrating security throughout the SDLC needs development teams, security pros, and stakeholders to work together [32]. Organizations can create a culture of security by using secure development techniques like threat modeling and giving developers security training [33]. Corporations can make applications more safe and resistant to cyber threats by incorporating security into the development process.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:05/May-2024 Impact Factor- 7.868 www.irjmets.com

Table 2: Security Integration Across the Software Development Life Cycle (SDLC) [22, 25, 27, 28, 30]

SDLC Stage	Security Activities and Measures
Planning and Requirements	● Define security objectives and compliance requirements ● Identify sensitive data ● Define access control policies ● Establish security testing criteria
Design and Architecture	● Adopt secure coding practices and design patterns Conduct threat modeling to identify potential security threats ● Design appropriate security countermeasures
Development and Coding	● Follow secure coding guidelines ● Perform regular code reviews ● Use secure coding libraries ● Implement input validation and output encoding ● Avoid common coding errors that can lead to vulnerabilities
Testing and Validation	● Conduct comprehensive security testing ● Penetration testing to simulate real-world attacks ● Dynamic application security testing (DAST) to analyze application behavior during runtime ● Perform vulnerability scans and assess the effectiveness of security controls
Deployment and Maintenance	● Implement continuous monitoring and incident response capabilities ● Deploy security patches promptly ● Monitor for security events and anomalies ● Establish and maintain a well-defined incident response plan ● Perform regular vulnerability assessments and penetration testing Continuously update and improve security measures based on emerging threats

V. SECURITY TESTING TECHNIQUES

Due diligence in security testing is necessary to locate and fix vulnerabilities in security before hackers can use them [34]. To find vulnerabilities in an application's defense one can perform penetration testing also called ethical hacking, which involves breaking into applications as a black hat and finding weaknesses [35]. Poynter Institute research shows that companies that do penetration testing are more likely to find and handle security events correctly [36]. Manual or automatic tools, like Metasploit, Burp Suite, and OWASP ZAP [37], can be used for penetration testing.

One can find bugs and runtime weaknesses with other testing methods, like static code analysis and dynamic analysis [38]. Finding possible security vulnerabilities in source code by looking at it without running it is called static code analysis [39]. On the other hand, testing an application while it is running is called dynamic code analysis [40]. For example, SonarQube, Checkmarx, and Veracode are static code analysis tools that can look

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:05/May-2024 Impact Factor- 7.868 www.irjmets.com

through the source code to find common bugs and security vulnerabilities [40]. By looking at how an application behaves and interacts with its users and other systems, dynamic analysis tools like OWASP ZAP and Burp Suite can find runtime vulnerabilities [41].

In addition to these methods, businesses should think about doing security checks, like threat modeling and risk assessments, to find and rank security risks [42]. Finding possible attack vectors and designing effective defenses requires studying an application's architecture and data flows, which is what threat modeling does [25]. Assessments of risks help companies figure out how likely and harmful security threats are, so they can plan their efforts to reduce those risks more effectively [43].

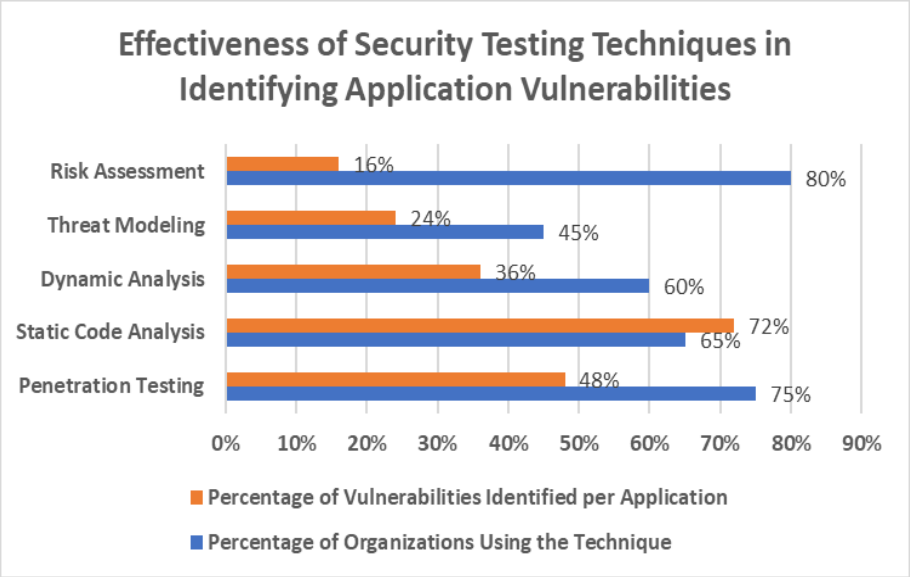

Fig. 2: Comparing the Adoption and Efficacy of Security Testing Methods in Vulnerability Detection [34-43]

VI. BEST PRACTICES FOR APPLICATION SECURITY

Use of secure coding rules and practices is necessary to create strong and reliable applications [7]. The OWASP Secure Coding Practices Quick Reference Guide gives software developers a lot of advice [44]. For example, validating input data, encrypting private data, using parameterized queries to stop SQL injection, and making sure errors are handled correctly and are logged [45].

Finding and fixing new threats requires regular vulnerability exams and patches [46]. The risk of data breaches is 79% lower for companies that do regular vulnerability assessments, according to a study by Forrester Research [47]. Vulnerability assessments should be done on a specific cadence based on an organization's risk profile [48]. It is important to prioritize vulnerabilities based on how severe they are and how they might affect users, and patches should be applied right away [49].

Secure controls like strong authentication, granular permission, and data encryption can help lower the risk of data leaks and unauthorized access [50]. Secure authentication methods, like biometrics and multi-factor authentication (MFA), can stop attackers from accessing applications without authorization even if passwords are stolen [51]. According to the principle of least privilege [52], granular authorization makes sure that users can only access the tools and features they need to do their jobs. Protecting private data from unwanted access and changes while it's being sent or stored is called data encryption [53].

Fostering a culture of security in the company also requires making developers aware of security issues and giving them training [54]. An investigation by the SANS Institute discovered that businesses with a strong security education program have a 70% lower chance of experiencing a security problem compared to those that don't.Common security vulnerabilities and attack methods, as well as the company's security rules and policies [56], should all be covered in security training.

Furthermore, businesses should think about implementing a secure software development life cycle (SSDLC) structure, like the Microsoft Security Development Lifecycle (SDL) or the OWASP Software Assurance Maturity

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:05/May-2024 Impact Factor- 7.868 www.irjmets.com

Model (SAMM) [57]. Implementing security throughout the development process, from gathering requirements to deployment and upkeep, is made easier with these frameworks [58]. Furthermore, businesses should regularly check how well their application security program is working by using measures and key performance indicators (KPIs) [59]. The amount of vulnerabilities found and fixed, the time it takes to fix vulnerabilities, and amount of application-related security incidents [60] are some examples of these metrics.

VII. REAL-WORLD EXAMPLES AND CASE STUDIES

Recent data breaches, such as the ransomware attack on a major U.S. pipeline operator in 2022 [39] and the supply chain attack on a widely used enterprise software provider in 2023 [40], underscore the severe consequences of inadequate application security. The pipeline operator breach occurred due to a compromised password for a virtual private network (VPN) account, which allowed attackers to infiltrate the company's network and deploy ransomware [41]. This breach led to the temporary shutdown of the pipeline, causing fuel shortages and price hikes across the southeastern United States [42]. The incident highlights the importance of strong access controls and multi-factor authentication to prevent unauthorized access [43].

In the case of the enterprise software provider breach, attackers exploited a zero-day vulnerability in the company's flagship product to gain access to customer networks [44]. The software provider's extensive customer base, which included numerous government agencies and Fortune 500 companies, amplified the impact of the breach [45]. This attack emphasizes the need for timely vulnerability management and patch deployment, as well as the risks associated with relying on third-party software [46].

On the other hand, the successful implementation of application security practices by a leading financial services company demonstrates the benefits of integrating security throughout the SDLC [47]. Following a significant data breach in 2019, the company invested heavily in a comprehensive security program that included secure coding training, regular penetration testing, and the adoption of a zero-trust architecture [48]. A case study conducted by a renowned cybersecurity research firm found that the financial services company's security initiatives reduced the likelihood of a data breach by 75% and improved its overall security posture [49].

The healthcare sector has also benefited from the implementation of application security measures. In 2022, a major hospital chain in Europe suffered a ransomware attack that compromised the personal information of over 1 million patients [50].

In response, the hospital chain collaborated with a leading cybersecurity solutions provider to develop a robust application security strategy [51]. This included the implementation of secure coding practices, regular security audits, and the adoption of a security orchestration, automation, and response (SOAR) platform [52]. As a result, the hospital chain was able to reduce its attack surface, improve its incident response capabilities, and maintain the trust of its patients [53].

These real-world examples and case studies underscore the critical importance of application security in protecting sensitive data and maintaining customer trust. Organizations that prioritize security and adhere to best practices, such as secure coding, regular testing, and timely patching, are better positioned to prevent breaches and mitigate their impact when they do occur.

VIII. FUTURE OF APPLICATION SECURITY

Threats are always changing, so businesses need to stay alert and make changes to their application security tactics as needed [46]. As cyberattacks get smarter and more people use cloud computing, mobile devices, and the Internet of Things (IoT), application security is facing new problems [54]. Changes in recent years, like using DevSecOps methods [55] and AI and ML for automated security [56], are affecting the future of application security.

DevSecOps tries to bring together the development, security, and operations teams by adding security to the DevOps method [57]. By automating security tests and integrating them into the CI/CD pipeline, DevSecOps helps companies find and fix security problems earlier in the development process [58]. This shift-left method cuts down on the time and money needed to fix security holes and makes the application safer overall [59].

An increasing number of security systems are also using AI and ML to help with automation [60]. By looking at huge amounts of data and finding patterns that could point to a security breach, AI and ML can help businesses

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:05/May-2024 Impact Factor- 7.868 www.irjmets.com

find threats and respond to them more quickly and correctly [61]. For example, AI-powered anomaly detection systems can spot strange user behavior or network data that could be a sign of a breach. This makes it easier for security teams to look into it and take action [62].

A zero-trust security model [63] is another trend that will shape the future of security. Zero trust says that by default, one can't trust any person or device, and it needs strict authentication and permission for every access request [64]. This model is especially useful for cloud computing and remote work, where the lines between networks are becoming less clear [65].

More organizations are using open-source software, which is also changing the security of applications [66]. Even though open-source software can save one's money and speed up development, it also comes with new risks because the code may have vulnerabilities that one can't see at first [67]. Organizations should set up ways to choose, check, and update open-source components, as well as ways to keep an eye out for known security vulnerabilities and patches [68].

Because of these problems, businesses are taking a more proactive approach to application security, which includes testing, tracking, and making improvements all the time [7]. For this, one needs to use security analytics and threat intelligence tools to find new threats and stop them [69]. One should also do regular security checks and penetration tests to find and fix vulnerabilities in their defenses [70]. Businesses that follow these trends and make security a priority during the software development process will be better able to maintain customer trust and protect their digital assets [71].

IX. APPLICATIONS OF APPLICATION SECURITY

Application security principles are integral to various security products and services. Web Application Firewalls (WAFs) protect applications from common web-based attacks by inspecting and filtering malicious traffic [72]. According to a report by MarketsandMarkets, the global WAF market size is expected to grow from $3.4 billion in 2020 to $5.5 billion by 2025, at a Compound Annual Growth Rate (CAGR) of 9.9% during the forecast period [73]. WAFs can be deployed as hardware appliances, virtual appliances, or cloud-based services, and they use a combination of signature-based and behavior-based detection techniques to identify and block malicious requests [74].

Identity and Access Management (IAM) tools ensure that only authorized users can access sensitive data and functionalities [75]. IAM solutions, such as single sign-on (SSO), multi-factor authentication (MFA), and role-based access control (RBAC), help organizations manage user identities and enforce access policies [76]. SSO allows users to access multiple applications with a single set of credentials, reducing the risk of password fatigue and improving the user experience [77]. MFA requires users to provide additional factors of authentication, such as a fingerprint or a one-time code, in addition to their password, making it harder for attackers to gain unauthorized access [78]. RBAC enables organizations to define and enforce granular access policies based on user roles and responsibilities, ensuring that users have access only to the resources they need to perform their tasks [79].

Penetration testing services assist organizations in identifying and addressing vulnerabilities in their applications before attackers can exploit them [34]. Regular penetration testing can assist organizations in identifying and fixing vulnerabilities before attackers can exploit them. Internal security teams or specialized security companies can perform penetration testing services [80]. These services typically involve a combination of automated vulnerability scanning and manual testing techniques, such as social engineering and credential stuffing [81]. The results of the penetration test are then used to prioritize and remediate the identified vulnerabilities, improving the overall security posture of the application [82].

By leveraging these application security solutions, organizations can enhance their overall security posture and protect their digital assets from cyber threats. A Gartner report predicts that by 2025, 60% of organizations will use WAFs, IAM solutions, and penetration testing services as part of their application security strategy [83]. However, it is important to note that these solutions are not a silver bullet and should be used in conjunction with other security best practices, such as secure coding, regular patching, and employee security awareness training [84].

X. CONCLUSION

Finally, application security is an important part of building software that can't be ignored in today's digital world. Strong application security measures are needed more than ever because businesses depend on software apps more and more to run their businesses and store sensitive data. Using best practices like secure coding and regular testing, along with new technologies like AI and machine learning, and an approach that includes security throughout the software development life cycle, can help organizations greatly lower the risk of security breaches. But application security is an ongoing process that needs to be watched over, improved, and improved upon all the time to keep up with the emerging tts. Companies can build stronger and safer apps that protect their digital assets, keep customers trusting them, and help them do well in a world where cyber risks are growing if they put application security first and create a culture of security within their groups.

XI. REFERENCES

[1] Acunetix. (2023). Web Application Vulnerability Report 2023. https://www.acunetix.com/white-papers/web-application-vulnerability-report-2023/
[2] IBM. (2022). Cost of a Data Breach Report 2022. https://www.ibm.com/security/data-breach
[3] Verizon. (2022). 2022 Data Breach Investigations Report. https://www.verizon.com/business/resources/reports/dbir/
[4] Kaspersky. (2023). The true cost of a data breach: Direct and indirect expenses. https://www.kaspersky.com/blog/data-breach-costs/
[5] McKinsey & Company. (2022). COVID-19 digital transformation & technology. https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/the-covid-19-recovery-will-be-digital-a-plan-for-the-first-90-days
[6] NIST. (2022). Cybersecurity Framework Version 1.1. https://www.nist.gov/cyberframework/framework
[7] OWASP. (2023). OWASP Top Ten 2021. https://owasp.org/Top10/
[8] OWASP, "OWASP Top Ten Web Application Security Risks," OWASP Foundation, 2021.
[9] W. G. J. Halfond et al., "A Classification of SQL Injection Attacks and Countermeasures," in Proc. IEEE Int. Symp. Secure Softw. Eng., pp. 13–15, 2006.
[10] OWASP, "OWASP Top 10 2021," OWASP Foundation, 2021.
[11] A. Javed and M. Schwenk, "Towards Elimination of Cross-Site Scripting on Mobile Versions of Web Applications," in Proc. 3rd Int. Workshop Eng. Secure Softw. Syst. (ESSoS'12), pp. 50-57, 2012.
[12] Acunetix, "What is Cross-site Scripting (XSS) and How to Prevent It?," Acunetix, 2021.
[13] A. Barth et al., "Robust Defenses for Cross-Site Request Forgery," in Proc. 15th ACM Conf. Comput. Commun. Secur. (CCS '08), pp. 75–88, 2008.
[14] Thales, "2021 Thales Data Threat Report," Thales, 2021.
[15] Y. Shin and L. Williams, "An Empirical Model to Predict Security Vulnerabilities Using Code Complexity Metrics," in Proc. 2008 ACM-IEEE Int. Symp. Empirical Softw. Eng. Meas. (ESEM '08), pp. 315-317, 2008.
[16] J. M. Myerson, "Identifying Enterprise Network Vulnerabilities," Int. J. Found. Comput. Sci. Technol., vol. 2, no. 3, pp. 27-41, 2012.
[17] FIRST, "Common Vulnerability Scoring System (CVSS)," FIRST, 2021.
[18] NIST, "National Vulnerability Database (NVD)," NIST, 2021.
[19] G. McGraw, "Software Security," IEEE Secur. Privacy, vol. 2, no. 2, pp. 80-83, 2004.
[20] OWASP, "OWASP Secure Coding Practices Quick Reference Guide," OWASP Foundation, 2021.
[21] National Institute of Standards and Technology, "The Economic Impacts of Inadequate Infrastructure for Software Testing," NIST, 2002.
[22] OWASP, "OWASP SAMM Project," OWASP Foundation, 2021.
[23] NIST, "Security Requirements for Cryptographic Modules," FIPS 140-2, 2001.
[24] M. Howard and D. LeBlanc, "Writing Secure Code," Microsoft Press, 2002.
[25] OWASP, "OWASP Threat Modeling," OWASP Foundation, 2021.
[26] G. McGraw, "Automated Code Review Tools for Security," IEEE Comput. Soc., vol. 41, no. 12, pp. 108-111, 2008.

[27] OWASP, "OWASP Code Review Guide," OWASP Foundation, 2021.
[28] OWASP, "OWASP Testing Guide," OWASP Foundation, 2021.
[29] PortSwigger, "Dynamic Application Security Testing (DAST)," PortSwigger, 2021.
[30] NIST, "Security and Privacy Controls for Information Systems and Organizations," NIST Special Publication 800-53, Revision 5, 2020.
[31] NIST, "Computer Security Incident Handling Guide," NIST Special Publication 800-61, Revision 2, 2012.
[32] H. Assal and S. Chiasson, "Security in the Software Development Lifecycle," in Proc. 14th Symp. Usable Privacy Secur. (SOUPS '18), pp. 281-296, 2018.
[33] GitLab, "2021 Global DevSecOps Survey," GitLab, 2021.
[34] G. McGraw, "Software Security Testing," IEEE Secur. Privacy, vol. 2, no. 5, pp. 81-85, 2004.
[35] J. Broad and A. Bindner, "Hacking with Kali Linux: A Guide to Ethical Hacking," No Starch Press, 2019.
[36] Ponemon Institute, "The Cost of Insecure Software," Synopsys, 2020.
[37] OWASP, "OWASP Zed Attack Proxy (ZAP)," OWASP Foundation, 2021.
[38] B. Chess and J. West, "Secure Programming with Static Analysis," Addison-Wesley Professional, 2007.
[39] CNN. (2022). Colonial Pipeline ransomware attack: What we know. https://www.cnn.com/2022/05/10/business/colonial-pipeline-ransomware-attack/index.html
[40] ZDNet. (2023). SolarWinds hack: Everything you need to know. https://www.zdnet.com/article/solarwinds-hack-everything-you-need-to-know/
[41] Cybersecurity & Infrastructure Security Agency. (2022). Alert (AA22-131A): DarkSide Ransomware: Best Practices for Preventing Business Disruption from Ransomware Attacks. https://www.cisa.gov/uscert/ncas/alerts/aa22-131a
[42] Reuters. (2022). U.S. pipeline hackers say their aim is cash, not chaos. https://www.reuters.com/business/energy/us-pipeline-hackers-say-their-aim-is-cash-not-chaos-2022-05-10
[43] [NIST. (2022). Multi-Factor Authentication. https://www.nist.gov/system/files/documents/2022/03/30/MFA-Fact-Sheet-March-2022.pdf
[44] Microsoft. (2023). SolarWinds post-compromise hunting with Azure Sentinel. https://azure.microsoft.com/en-us/blog/solarwinds-post-compromise-hunting-with-azure-sentinel/
[45] The New York Times. (2023). SolarWinds Hack Puts Many Customers at Risk. https://www.nytimes.com/2023/01/12/technology/solarwinds-hack-customers.html
[46] OWASP. (2023). Vulnerability Management. https://owasp.org/www-community/Vulnerability_Management
[47] Forbes. (2022). How One Bank Is Using AI To Enhance Cybersecurity. https://www.forbes.com/sites/forbestechcouncil/2022/08/17/how-one-bank-is-using-ai-to-enhance-cybersecurity/
[48] IBM. (2023). What is a zero trust architecture? https://www.ibm.com/topics/zero-trust
[49] Gartner. (2023). How to Reduce the Risk of Data Breaches with a Comprehensive Security Program. https://www.gartner.com/en/documents/3989071/how-to-reduce-the-risk-of-data-breaches-with-a-comprehen
[50] BBC. (2022). Ransomware attack on UK hospital chain. https://www.bbc.com/news/technology-59879084
[51] Healthcare IT News. (2022). How a hospital chain recovered from a ransomware attack. https://www.healthcareitnews.com/news/how-hospital-chain-recovered-ransomware-attack
[52] Palo Alto Networks. (2022). What is SOAR? https://www.paloaltonetworks.com/cyberpedia/what-is-soar
[53] Forbes. (2022). How A Hospital Chain Used Cybersecurity Best Practices To Recover From Ransomware. https://www.forbes.com/sites/forbestechcouncil/2022/09/22/how-a-hospital-chain-used-cybersecurity-best-practices-to-recover-from-ransomware/
[54] NIST, "Recommendation for Key Management: Part 1 - General," NIST Special Publication 800-57 Part 1, Revision 5, 2020.

[55] G. Brechbuhl et al., "Cybersecurity is Everyone's Job," Harvard Business Review, 2021.
[56] SANS Institute, "SANS 2021 Security Awareness Report," SANS Institute, 2021.
[57] NIST, "Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations," NIST Special Publication 800-137, 2011.
[58] OWASP, "OWASP Software Assurance Maturity Model (SAMM)," OWASP Foundation, 2021.
[59] Microsoft, "SDL Practices," Microsoft, 2021.
[60] NIST, "Performance Measurement Guide for Information Security," NIST Special Publication 800-55, Revision 1, 2008.
[61] CIS, "CIS Controls," CIS, 2021.
[62] Cisco, "2021 Cisco Annual Internet Report (2018–2023) White Paper," Cisco, 2021.
[63] NIST, "Zero Trust Architecture," NIST Special Publication 800-207, 2020.
[64] Forrester Research, "The Zero Trust eXtended Ecosystem: Networks," Forrester Research, 2020.
[65] Gartner, "Top Security and Risk Management Trends," Gartner, 2021.
[66] Synopsys, "2021 Open Source Security and Risk Analysis Report," Synopsys, 2021.
[67] A. Dunn, "The Hidden Risks of Open Source Software," Dark Reading, 2019.
[68] Linux Foundation, "Open Source Security Foundation (OpenSSF)," Linux Foundation, 2021.
[69] Gartner, "Market Guide for Security Analytics," Gartner, 2021.
[70] NIST, "Technical Guide to Information Security Testing and Assessment," NIST Special Publication 800-115, 2008.
[71] J. P. Foley, "Building a Culture of Security," Harvard Business Review, 2020.
[72] OWASP, "OWASP Top 10 2021: A05-2021: Security Misconfiguration," OWASP Foundation, 2021.
[73] MarketsandMarkets, "Web Application Firewall Market by Component (Solutions (Hardware Appliances, Virtual Appliances, Cloud-Based) and Services), Organization Size, Industry Vertical, and Region - Global Forecast to 2025," MarketsandMarkets, 2020.
[74] Imperva, "Web Application Firewall (WAF)," Imperva, 2021.
[75] OWASP, "OWASP Top 10 2021: A07-2021: Identification and Authentication Failures," OWAS Foundation, 2021.
[76] Okta, "Identity and Access Management (IAM)," Okta, 2021.
[77] Auth0, "Single Sign On (SSO)," Auth0, 2021.
[78] Duo Security, "Multi-Factor Authentication (MFA)," Duo Security, 2021.
[79] NIST, "Role-Based Access Control (RBAC)," NIST, 2021.
[80] Rapid7, "Penetration Testing Services," Rapid7, 2021.
[81] HackerOne, "The Hacker-Powered Security Report 2021," HackerOne, 2021.
[82] OWASP, "OWASP Vulnerability Management Guide," OWASP Foundation, 2021.
[83] Gartner, "Predicts 2021: Application Security," Gartner, 2020.
[84] SANS Institute, "SANS Top 25 Software Errors," SANS Institute, 2021.