ICIT | Institute for Critical Infrastructure Technology
The Cybersecurity Think Tank

A Holistic Approach to Application Development and Digital Security for the Enterprise

AN ICIT FELLOW PERSPECTIVE
JULY 2020

DARREN DEATH

ICIT FELLOW

Introduction

Developing resilient code for business applications is critically important to the continued mission success of companies, as this defends against the exploitation of software vulnerabilities that attackers can leverage to disrupt organizational stability (Sinha, 2019). Ensuring that newly developed code is free from exploitable vulnerabilities contributes to resilience by reducing an organization's overall attack surface (Petratos & Faccia, 2019). By reducing the total attack surface across an enterprise's digital inventory, an organization is effectively making it more difficult for an attacker to exploit and retrieve sensitive organizational assets (Karanja, 2017). Thus, guaranteeing that cybersecurity requirements, principles, and tools are implemented across every application's System Development Life Cycle (SDLC) is a top priority for development teams looking to defend their organization from malicious adversaries (Josang, Odegaard, & Oftedal, 2015). Developers can implement mechanisms to ensure their code is free from exploitable vulnerabilities (Thomas, 2018). These mechanisms include integrating cybersecurity throughout the SDLC, applying best practices and standards, and static and dynamic code analysis (Reagin & Gentry, 2018).

> *"Developing resilient code for business applications is critically important to the continued mission success of companies, as this defends against the exploitation of software vulnerabilities that attackers can leverage to disrupt organizational stability."*
>
> \- Darren Death
> ICIT Fellow

The general business problem affecting organizations is that development teams are creating insecure applications that pose serious risks to the mission of their organizations. Specifically, cybersecurity requirements and best practices are not integrated and operationalized as part of the SDLC for software application projects (Josang, Odegaard, & Oftedal, 2015). This is likely because many development teams do not have personnel trained in cybersecurity best practices and thus security is regularly an afterthought in the development process. By adequately integrating and operationalizing cybersecurity requirements throughout the SDLC, organizations can develop secure software with a more reasonable risk profile.

Enterprise software often plays a critical role in the performance of a company, supporting functions that enable the organization's delivery of value to its customers (Porter, 2001). However, software that has inherent cybersecurity defects because of inadequate requirements could result in a loss of confidence from constituents, severely impacting an organization's survival (Golgeci & Ponomarov, 2013). An effective countermeasure to this risk is ensuring that cybersecurity requirements are integrated into any new software development activity at project initiation (Yayla & Hu, 2012). Additionally, by adopting cybersecurity requirements early in the development process, organizations accumulate less technical debt in newly developed software (Granneman, 2018). The cost to implement a security control is always more expensive later in a project, as reworking a project takes more time than designing intentionally to begin with (Brown et al., 2010).

Cybersecurity and Application Development Integration

Integrating cybersecurity into all phases of the SDLC ensures that cybersecurity capabilities are built into every new application. However, including a cybersecurity team in the development of technology projects is challenging (Kulkarni, 2018). In some cases, an organization's cybersecurity program is perceived as the group who denies access to technology, rather than enabling innovation. In other cases, working to implement cybersecurity requirements is seen as slowing down projects, resulting in missed deadlines. As a consequence of these views, cybersecurity leadership must work on creating a positive relationship between cybersecurity, application development, and other technical teams. Cybersecurity leadership must work closely with technical teams to provide the guidance, tools, and resources to help the developer community create secure code (Kulkarni, 2018). Utilizing industry best practices like the Open Web Application Security Project (OWASP) Top 10, developers can retrieve specific information on known vulnerabilities commonly coded into business applications (OWASP, 2017).

> *"Cybersecurity leadership must work closely with technical teams to provide the guidance, tools, and resources to help the developer community create secure code"*
>
> \- Darren Death
> ICIT Fellow

Application Development Cybersecurity Tools

Technical tools related to application code analysis should be utilized by development teams to ensure that new code is secure and free from vulnerabilities. In particular, both static and dynamic code analysis should be conducted. Static code analysis examines the source code of an application looking for security vulnerabilities (Thomas, 2018). This should be part of the quality-assurance testing within the application's SDLC (Williams, 2018). Doing so will increase efficiency by embedding some of the security checks into the existing development review processes (Morales, Yasar, & Volkmann, 2018). Dynamic code analysis differs from static code analysis in that it operates against a live production environment (Rahul, Kharvi, & Manu, 2019). Traditionally, dynamic code analysis mimics the actions of a user attempting to access an application. However, in this case, the dynamic scanning tool attempts to find potential vulnerabilities by mimicking a malicious actor. Like static code analysis, dynamic code analysis can be implemented as part of the quality assurance process (Williams, 2018). Unlike static code analysis, dynamic code analysis can also be used during the operations phase of the SDLC (Josang, Odegaard, & Oftedal, 2015) where it is conducted as part of regularly planned penetration testing exercises (Abdullah, 2020).

Application Host Server Cybersecurity Tools

Part of protecting an application environment is ensuring that the underlying operating system is secure (Ahmed & Al-Shaer, 2019). Just as cybersecurity should be integrated throughout the SDLC, the implementation of cybersecurity best practices and standards is entirely applicable to the implementation of information technology (IT) systems. For operating systems, one of the best sources of information is the Center for Internet Security (CIS) Critical Security Controls (CSC) (CIS, 2019). This set of best practices provides IT teams with a wide variety of operating system types and technologies to choose from, with a focus on delivering the configurations needed to operate those technologies securely. From an SDLC perspective, cybersecurity should be embedded in all server environments that will host business applications to ensure that best practices are applied to any new hosts (Josang, Odegaard, & Oftedal, 2015).

Where application development utilizes static and dynamic code analysis to verify security control effectiveness, host operating systems utilize automated vulnerability assessment tools to analyze and report on the vulnerabilities that may be present in the target system (Patel, 2019). An automated vulnerability assessment tool can be loaded with the organization's cybersecurity policies, which can be used to assess the host's systems security posture. The automated vulnerability scanner will report the security controls that are compliant with organizational policies and the configurations that fall outside the expected norms established by organizational policy. An automated vulnerability scanner also interrogates the host operating system for patch compliance. Organizations who do not implement security patches are vulnerable to malicious attackers looking to gain unauthorized access to sensitive organizational information (FBI, 2018). By effectively utilizing an automated vulnerability assessment tool, the threat surface of a host operating system is reduced, improving mission resiliency.

Managing Discovered Application and Host Vulnerabilities

It is also essential to look at all digital systems across an organization's environment, including ones internal to the organization or outsourced to a third party (Patel, 2019). Any internal vulnerability should be assessed across all deployed platforms to determine the severity of the potential breach (Ogutu, Bennett, & Olawoyin, 2018). Examples of deployed platforms include servers, server software, workstations, desktop applications (i.e., Application Integrated Development Environment (IDE)), networking equipment, Internet of Things (IoT) technologies, and web-based applications. Special care must be taken to implement tools and techniques that allow for the identification of vulnerabilities across all these platforms, including cloud technologies (Yimam & Fernandez, 2016). Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) implementations must be reviewed to ensure that all configurations initiated by the customer are applied correctly and do not expose the organization to unexpected risk.

An organization will find vulnerabilities throughout its collection of digital assets. As a result, it may be unreasonable to close them all at once. This is where the concept of triage comes into play, typically following one of two vulnerability mitigation strategies (Ogutu, Bennett, & Olawoyin, 2018). The first strategy focuses on vulnerabilities with known exploits that can be leveraged over a network connection, as these must be immediately mitigated. The second strategy is to methodically implement mitigation activities. This can be used for assets where physical access is required to exploit a vulnerability or robust segmentation exists on the network, thus protecting the asset (Granneman, 2018). Whichever strategy is used, the organization should continuously monitor themselves, ensuring previously discovered vulnerabilities are appropriately mitigated and new vulnerabilities are promptly removed. The best way to do this is through vulnerability and patch compliance tools, as most organizations have diverse technologies deployed throughout their enterprise (Cain, Edwards, & Still, 2018).

Authentication Considerations for Applications

Many models and frameworks exist to protect the authentication process. One of those frameworks is Multi-Factor Authentication (MFA), which is a security scheme that utilizes multiple methods to gather information about a user before allowing them to securely access an application (Ibrokhimov et al., 2019). Factors that are utilized as a part of a MFA system include knowledge, inherence, possession, and time, and Knowledge (Nguyen & Memon, 2018).

The easiest of the four factors to grasp is knowledge, as it includes stereotypical login identification like an individual's username and password. To increase the security of an underlying authentication process, this is usually combined with other factors to create a true MFA scheme (Nguyen & Memon, 2018).

Inherence is a physical quality about the user (Ometov et al., 2018), most commonly biometric characteristics like a retina or fingerprint. When a user attempts to login to an application he or she submits biometric data. The MFA system then compares a digitized sample of the biometric data with the data submitted by the user. If a match exists, the user is granted access to the application.

Possession requires an item that an individual owns, usually a physical authentication token or a user's smartphone. Typically, a randomly generated code is created using an algorithm that examines the time of day and data tied to the user. The item and the business application both use the same algorithm, so when the user enters the current code from the item, the business application can verify the user by ensuring the codes match (Agrawal et al., 2019). Critically, there is no need for the item and the business application to exchange information directly during this process.

Time refers to sending users a one-time code that needs to be entered within a specific window (Chang-Seop, 2018). This is usually combined with the possession factor, as the one-time code is sent to a device which is already linked to the account being accessed. Most commonly, this code is sent via a text message to a phone or email to a previously registered address. The user has a certain amount of time to use the code before it expires.

Application Security Management Considerations

> *"To effectively integrate the cybersecurity capabilities described, management support is essential"*
>
> - Darren Death
> ICIT Fellow

To effectively integrate the cybersecurity capabilities described above, management support is essential (Rothrock et al., 2018). A formal cybersecurity program, aligned with the organization's risk appetite, must be established to support business-aligned security control implementations (Ogutu, Bennett, & Olawoyin, 2018). The executive leadership must ensure that cybersecurity is prioritized within IT implementations and integrated into all digital projects at initiation. Doing so will allow every software application and digital technology to be evaluated for security ramifications as part of the design process. Empowering a cybersecurity team to implement the above-mentioned technical countermeasures will protect a company's assets from potential exploits related to insecure software development (Karanja & Rosso, 2017).

Conclusion

Developing secure code for business applications is critically important. Without it, companies cannot defend against the exploitation of their software vulnerabilities, leaving businesses open to attackers and the subsequent disruption of organizational stability (Sinha, 2019). By reducing the total attack surface across an enterprise's digital inventory, an organization is effectively making it more difficult for malicious actors to retrieve sensitive organizational assets. To do so, cybersecurity requirements, principles, and tools need to be a top priority across the SDLC for applications (Josang, Odegaard, & Oftedal, 2015). Developers can do this by implementing cybersecurity best practices, and static and dynamic code analysis.

For a new software project to be implemented successfully, the cybersecurity and software development teams must be well integrated, guaranteeing that cybersecurity requirements are communicated, implemented, and deployed (Death, 2017). An organization's leadership must be fully engaged to guard against the risk of insecurely developed software applications, including reminding all employees that cybersecurity requirements are, in fact, business requirements supported by the uppermost leaders of the organization (Rothrock, Kaplan, & Oord, 2018).

About the Author

Darren Death is a proven technology leader with over 20 years of experience deploying enterprise systems for large private and public organizations. Death has led, designed, and implemented large-scale, organizational-wide enterprise IT systems with far-reaching impact.

Death currently serves on the EC-Council International Advisory Board for TVM (Threat and Vulnerability Management) and as the CISO and serves as the InfraGard Maryland – Cyber Threat Special Interest Group Chief and American Council for Technology / Industry Advisory Council (ACT-IAC) – Cyber Security Community of Interest Program Chair. He serves on the Board of Advisors and as faculty for the Cyber Intelligence Initiative at the Institute of World Politics. Death holds a master's degree in Cybersecurity and Information Assurance and is currently working toward his Doctorate in Information Technology - Information Assurance and Cybersecurity.

This article aligns with Death's doctoral research in Cybersecurity, specifically related to the tools and techniques used by an organization in implementing a strategy that is organizationally aligned and focused on security.

About ICIT

The Institute for Critical Infrastructure Technology (ICIT) is a 501c(3) cybersecurity Think Tank with a mission to cultivate a cybersecurity renaissance that will improve the resiliency of our Nation's 16 critical infrastructure sectors, defend our democratic institutions, and empower generations of cybersecurity leaders.

References

Abdullah, H. S. (2020). Evaluation of open source web application vulnerability scanners. *Academic Journal of Nawroz University, 9*(1), 47-52.

Agrawal, V., Paliwal, R. K., Sharma, P., & Shrivastava, A. (2019). *Web security using user authentication methodologies: CAPTCHA, OTP and user behavior authentication.* OTP and User Behavior Authentication (2019).

Ahmed, M., & Al-Shaer, E. (2019). Measures and metrics for the enforcement of critical security controls: a case study of boundary defense. *In Proceedings of the 6th Annual Symposium on Hot Topics in the Science of Security* (pp. 1-3).

Brown , N., Cai, Y., Guo, Y., Kazman, R., Kim, M., Kruchten, P., & Sangwan, R. (2010). Managing technical debt in software-reliant systems. *In Proceedings of the FSE/SDP workshop on Future of software engineering research (pp. 47-52).*

Cain, A. A., Edwards, M. E., & Still, J. D. (2018). An exploratory study of cyber hygiene behaviors and knowledge. *Journal of Information Security and Applications, 42*, 36-45. doi:10.1016/j.jisa.2018.08.002

Center for Internet Security (CIS) (2019). *CIS controls.* Retrieved from https://www.cisecurity.org/controls/.

Chang-Seop, P. (2018). One-time password based on hash chain without shared secret and re-registration. *Computers & Security, 75*, 138–146.

Death, D. (2017). *Information security handbook: develop a threat model and incident response strategy to build a strong information security framework.* Birmingham, UK: Packt Publishing.

FBI (2018, May 07). *2017 internet crime report.* Retrieved from https://www.fbi.gov/news/stories/2017-internet-crime-report-released-050718

Golgeci, I., & Y. Ponomarov, S. (2013). Does firm innovativeness enable effective responses to supply chain disruptions? an empirical study. *Supply Chain Management: An International Journal, 18*(6), 604-617. doi:10.1108/SCM-10-2012-0331

Granneman, J. (2018). The business guide to improving information security. *The Journal of Equipment Lease Financing (Online), 36*(3), 1-9.

Ibrokhimov, S., Hui, K. L., Al-Absi, A. A., & Sain, M. (2019). Multi-Factor Authentication in Cyber Physical System: A State of Art Survey. In *2019 21st International Conference on Advanced Communication Technology (ICACT)* (pp. 279-284). IEEE.

Josang, A., Odegaard, M., & Oftedal, E. (2015). Cybersecurity Through Secure Software Development. *IFIP World Conference on Information Security Education (pp. 53-63).* Springer, Cham.

Karanja, E. (2017). The role of the chief information security officer in the management of IT security. *Information and Computer Security, 25*(3), 300-329. doi:10.1108/ICS-02-2016-0013.

Karanja, E., & Rosso, M. A. (2017). The chief information security officer: An exploratory study. *Journal of International Technology and Information Management, 26*(2), 23-47.

Kulkarni, M. (2018). IT governance in global banks: Emerging models. *Vinimaya, 39*(1), 11-21.

Morales, J. A., Yasar, H., & Volkmann, A. (2018). Weaving security into devops practices in highly regulated environments. *International Journal of Systems and Software Security and Protection (IJSSSP), 9*(1), 18-46.

Nguyen, T., & Memon, N. (2018). Tap-Based user authentication for smartwatches. *Computers & Security, 78*(2018), 174–186.

Ogutu, J., Bennett, M. R., & Olawoyin, R. (2018). Closing the gap: Between traditional & enterprise risk management systems. *Professional Safety, 63*(4), 42-47.

Ometov, A., Bezzateev, S., Mäkitalo, N., Andreev, S., Mikkonen, T., & Koucheryavy, Y. (2018). Multi-factor authentication: A survey. *Cryptography, 2*(1), 1. doi:10.3390/cryptography2010001

Open Web Application Security Project (OWASP) (2017). *OWASP top ten.* Retrieved from https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/

Patel, K. (2019). A Survey on Vulnerability assessment & penetration testing for secure communication. *In 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)* (pp. 320-325). IEEE.

Petratos, P., & Faccia, A. (2019). Accounting information systems and system of systems: assessing security with attack surface methodology. *In Proceedings of the 2019 3rd International Conference on Cloud and Big Data Computing* (pp. 100-105).

Porter, M. E. (2001). *The value chain and competitive advantage*. Understanding Business Processes, 50-66.

Rahul, B. S., Kharvi, P., & Manu, M. N. (2019). *Implementation of devsecops using open-source tools.*

Reagin, M. J., & Gentry, M. V. (2018). Enterprise cybersecurity: Building a successful defense program. *Frontiers of Health Services Management, 35*(1), 13-22. doi:10.1097/HAP.0000000000000037.

Rothrock, R. A., Kaplan, J., & Oord, F. (2018). The board's role in managing cybersecurity risks. *MIT Sloan Management Review, 59*(2), 12-15.

Sinha, S. (2019). *Bug bounty hunting for web security* (pp. 57-78). Apress, Berkeley, CA.

Thomas, T. W. (2018). *Security code review with static analysis techniques for the detection and remediation of security vulnerabilities.*

Williams, L. (2018). Continuously integrating security. *In Proceedings of the 1st International Workshop on Security Awareness from Design to Deployment* (pp. 1-2).

Yayla, A. A., & Hu, Q. (2012). The impact of IT-business strategic alignment on firm performance in a developing country setting: Exploring moderating roles of environmental uncertainty and strategic orientation. *European Journal of Information Systems: Special Issue: Information Systems Research, Education & Policy in the Mediterranean Region (MCIS), 21*(4), 373-387. doi:10.1057/ejis.2011.52

Yimam, D., & Fernandez, E. B. (2016). A survey of compliance issues in cloud computing. Journal of Internet Services and Applications, 7(1), 1-12. doi:10.1186/s13174-016-0046-8