EasyChair Preprint

№ 10435

Design Principles for Secure Systems

Taofeek O. Agboola

June 22, 2023

Design Principles for Secure Systems

Taofeek Agboola
Department of Computer Sceince
Stephen F. Austin State University
Nacogdoches, United States of America
agboolato@jacks.sfasu.edu

***Abstract*—Many designers mistakenly believe that strengthening security always reduces usability and vice versa. This paper presents useful principles for designing and building a secure system. The principles are useful to those whose aims are to design secure systems and to review existing ones. This paper explores the design principles of secure systems, which are essential for maintaining the confidentiality, integrity, and availability of data and systems. The principles of modularity, isolation, and secure-by-default are critical in ensuring systems are resilient to attacks and can recover securely from failures. The principle of least privilege is also discussed, as it is key in ensuring users only have the access they need, minimizing the risk of system compromise. Also, the principle of failing-securely is also considered. This principle ensures that systems can detect and respond to attacks or failures in a way that minimizes their impact. The implementation of these principles requires thorough understanding of the system architecture and potential threats. This paper emphasizes the importance of considering security from the beginning of the design process and continually throughout the system lifecycle. Overall, by following these principles, designers and architects can create secure systems that protect against a wide range of threats, resulting in boost of confidence among users and stakeholders. This study makes two significant contributions: first, it gives a model to help with thinking, and second, it offers actual advice in the form of seven interaction design principles for secure systems.**

Keywords—Security, Design Principles, Secure system, cybersecurity.

I. INTRODUCTION

Security is a quality aspect that constrains the behavior of application by imposing access and use restrictions on the data and other assets. Often time, security problems are associated with software errors like buffer overflows or weak cryptosystems. "A computer is secure if you can depend on it and its software to behave as you expect" [1]. A computing system regularly needs to store, process, and offer access to sensitive data in order to be useful. Unfortunately, this service makes them easy targets for attacks and the successful breach of these systems could lead to financial, embarrassing, and negative consequences.

In today's digital age, the security of information systems is of paramount importance. As organizations increasingly rely on technology to manage, store, transmit and retrieve sensitive data, there is a growing need for secure systems that can protect against a wide range of threats. Designing a secure system requires a holistic approach that considers all aspects of the system [1i]. It also requires a deep understanding of various serious risks, threat landscape and the potential vulnerabilities that could be exploited by attackers.

A. Security Concept

The security triad "CIA triad", refers to three essential concepts in information security: confidentiality, integrity, and availability. These three concepts are regarded as critical to the security and protection of information and systems.

a) Confidentiality: Confidentiality refers to the protection of sensitive data from unauthorized access or disclosure. This can be achieved using encryption, access controls, and other measures to limit access to data to authorized personnel only [2].

b) Integrity: Integrity ensures that data is not tampered with or altered in any way. This can be achieved through the use of digital signatures, hashing, and other cryptographic methods. [2]

c) Availability: Availability refers to the ability to access data and resources when needed. This can be achieved using redundant systems, backups, and disaster recovery plans. [2]

A secure information security framework is built on the three above-mentioned concepts. It is critical to balance these concepts based on an organization's specific demands and requirements. A banking institution, for example, may prioritize secrecy and integrity, whereas a healthcare company may favor availability and confidentiality. The security triad is frequently used as a framework for developing and implementing security controls and measures to protect against a variety of threats such as cyberattacks, data breaches, and other security incidents. Organizations can better defend themselves from the risks of unauthorized access, data loss, and other security concerns by ensuring that information privacy is maintained accurately, and available to authorized users at the authorized location on authorized devices when needed.

B. Design principles for secure systems

There are several design principles that can be used to ensure information systems, network and technologies are designed and built securely. One of the most important principles is the use of machine learning and artificial intelligence to improve the security of systems. This includes developing algorithms for detecting and mitigating security threats in real-time, as well as using machine learning to identify patterns and anomalies that may indicate security breach.

Buffer overflows, SQL injection, and cross-site scripting are examples of typical vulnerabilities that can be avoided by using safe coding techniques and tools. Utilizing secure coding practices and tools which involves creating guidelines for writing secure code and developing tools that can help identify and remediate security vulnerabilities during the development process is another important principle.

Designing secure systems also requires a thorough understanding of the threat landscape and the potential

vulnerabilities that could be exploited by attackers. This requires ongoing monitoring of security threats and vulnerabilities, as well as the development of effective threat intelligence capabilities.

The use of defense-in-depth, which involves implementing multiple layers of security controls to protect against a range of threats, includes implementing access controls, intrusion detection, response mechanisms, and encryption.

The absolute worst results of an attack can consistently be prevented if services are developed and run with security as top priority. In light of this, the paper will create a set of guidelines/principles to help build systems that are not only resistant to attack but also simpler to maintain and upgrade.

II. Design Principle Models

Before diving into the cybersecurity design concept, it is appropriate to first detail some guides that enhance decision making in designing secure systems. Applying these guides will necessitate some adaptation to the required situation. For instance, the requirements of an autonomous vehicle service may differ from the remote administration of an energy power station. In either case, it will guide decision making.

A. Establish the context

Before developing a secure system design, it is necessary to first determine the scope of the activity, including the objectives, understanding of the fundamentals and action to address any identified threats. Some of which are as follows:
Understand the acceptable risk, system's purpose, and what is required to run the system: For instance, a user may not be authorized access to view, modify or delete data. A system not being available to users for a period or at a particular location.

Understand the threat models: Using threat model approaches like attack trees to determine possibilities of an attack would help the design to examine the level of competence required for an adversary to carry out attack successfully. It will also help in mapping security controls, and build resilient measures to curb attacks.

Understand the systems end-to-end flow: This includes the understanding of the devices that will be used in accessing data, third party services that will access the system and appropriate security for every interaction. [3].

B. Making compromise difficult

Understanding crucial security controls: Low-skill attackers frequently give up easily after few efforts and move on to the next target if a system is designed to be tougher than average.

Untrusted Input should be validated, transformed, or rendered safely: External input should be well validated to ensure it conforms to the expected format and if not possible, it should be transformed or rendered safely to boost confidentiality.

Because attackers often target privileged users through social engineering, email phishing, replay, among others, designing of a different system/level of authentication is required. [3]

III. Cybersecurity design principles

Cybersecurity is the practice of protecting computers, servers, mobile devices, electronic systems, networks, and data from malicious attacks, theft, and damages

A. Principle: Defense-in-depth

The evidence that a single security product cannot completely protect a network and systems from every threat it might encounter serves as the foundation of a defense in depth approach. However, using a variety of security tools and procedures can help identify and stop attacks as they happen and allow organizations to successfully mitigate a variety of threats. As networks, systems, and user populations grow within businesses, this strategy assumes more and more significance [4]. Another advantage of layered security is redundancy. Other security measures can help limit the impact of an external attacker compromising one line of defense or an insider threat compromising a portion of an organization's network and mitigate the damage to the entire network. Using only one security solution, on the other hand, presents a single point of failure; if it is compromised, the entire network or system can be penetrated or harmed. Usage of multilevel[2] security measures should always be considered.

- Avoid single point of failure by building of defenses in multiple layers that back each other up thereby forcing attackers to defeat independent layers [5].
- Defense in depth execution must balance complexity, manageability, performance, and overall implementation cost [5].
- Capital and resource expenditures should never exceed the cost of the asset or the projected loss if there is a compromise [5].

An instance of defense in depth is through a composition of detectors.
Say you had two detectors, D_1 and D_2, which have false positive rates of FP_1 and FP_2 respectively, and false negative rates of FN_1 and FN_2, respectively. One way to use the two detectors would be to have them in parallel, meaning that either detector going off would trigger a response. This would increase the false positive rate and decrease the false negative rate. On the other hand, we could also have the detectors in series, meaning that both detectors have to alert in order to trigger a response. In this case, the false positive rate would decrease while the false negative rate would increase. [6]

B. Principle: Least Privilege

Least privilege: Least privilege is the principle of providing users with the minimum level of access required to perform a task. This can help to limit the impact of any security breaches or attacks [7].
For instance:

Consider a research building home to a team of scientists as well as other people hired to maintain the building (janitors, IT staff, kitchen staff, etc.) Some rooms with sensitive research data might be only accessible to trusted scientists. These rooms should not be accessible to the maintenance staff (e.g janitors). For best security practices, any one party should only have as much privilege as it needs to play its intended role [6].
Give a program the set of access privileges that it legitimately requires to execute its a certain operation. Trying to minimize

the level of privileges given each program and system component.

C. Principle: Seperation of Duties

Separation of duties is the idea that no one person should have too much power or influence over a system, hence it divides responsibilities among several people. By doing so, fraud, mistakes, and other security problems may be avoided [8]. For instance: The person in charge of making sales may also approve discounts in a financial system. That person would have an incentive to discount the software and may make poor discount decisions to increase sales. Instead, someone else with superior priority like a manager, may be assigned to approve a discount before the sale can be completed.

D. Principle: Secure by Default

Secure by default aims to ensure that all security features of a system are enabled by default, rather than requiring users to activate them manually. Here are some examples of how this principle can be applied in the design of secure systems.

- Two-factor authentication (2FA): 2FA is a security mechanism that requires users to provide two forms of identification before granting access to a system. By enabling 2FA by default, users are required to provide an additional layer of security beyond just a username and password. This can help to prevent unauthorized access to systems and data. Google, for example, has implemented 2FA by default for all users of their services [9].

- Encryption: Encryption is the process of encoding information so that it can only be read by authorized parties. By enabling encryption by default, organizations can ensure that their data is protected even if it is intercepted by unauthorized users. For example, Apple has implemented encryption by default for all data stored on its devices, including messages, photos, and other personal information [10].

- Firewall: A firewall is a network security device that monitors and restricts network traffic based on predefined security rules. Organizations can safeguard their networks from unwanted access and attacks by making a firewall the default setting. For example, Microsoft has implemented a firewall by default for all versions of its Windows operating system [11].

- Secure boot: Secure boot is a feature that ensures that only authorized software is loaded during the boot process of a system. By enabling secure boot by default, organizations can ensure that only trusted software is executed on their systems, which can help to prevent malware attacks [12].

E. Principle: Modularity

The design principle of modularity involves breaking down a system into smaller, independent components or modules to improve security. Some examples of how this principle can be applied in the design of secure systems include:

Microservices: Microservice is a modular method of software development in which a big application is divided and created in smaller chunks, and deployed independently. This technique can aid in security by decreasing the attack surface and limiting the effect of any security breaches that may occur. "Microservices architecture can support security practices by reducing the attack surface, limiting the impact of an attack, and promoting isolation" [13].

Modular hardware design: This entails breaking down a system into smaller, independent components that can be readily swapped out or upgraded. This can increase security by making it easier to identify and patch flaws in individual components. For example, a modular hardware design might include separate components for the processor, memory, and input/output interfaces. "Modularity in hardware design can be used to minimize the impact of vulnerabilities by constraining their effect to a limited part of the system" [14].

Virtualization: Virtualization is a technology that allows multiple operating systems or applications to run on a single physical machine. This can help to improve security by isolating different components or applications from each other, making it more difficult for attackers to move laterally through a system. For example, a company might use virtualization to run multiple virtual machines on a single physical server, with each virtual machine running a different application or service. Virtualization provides a level of isolation that can limit the impact of a compromise or prevent a successful attack from spreading.

F. Principle: Secure Failure

Even the most secure systems can fail. It is important to prepare for this possibility to minimize the impact, this principle is a crucial component of any successful cybersecurity strategy. No system can be totally secure, hence it's crucial to prepare for failure to lessen its effects. The principle of failing securely is a critical concept in cybersecurity that emphasizes the need for systems and applications to be designed in a way that minimizes the impact of security breaches and other failures [15].

Failing securely involves several key principles, including:

- Fail gracefully: Systems and applications should be built to fail gracefully, which means that they should work as much as feasible even if they fail. For example, a food website may temporarily disable customers the ability to place orders during a security breach, but customers can still be allowed to browse and check food menu.
- Limit access: Access to sensitive data and systems should be limited to only those who need it to perform their job duties. This helps to minimize the impact of any potential security breaches.
- Regularly test and update: Systems and apps should be tested and updated regularly to identify and address vulnerabilities. This includes performing frequent penetration testing and vulnerability scanning, as well

as updating software and systems with the most recent security patches.

- Encryption: Sensitive data should be encrypted both on transit and at rest to protect it from unauthorized access in the event of a security breach.

G. Principle: Isolation

Isolation is a major design approach in cybersecurity that includes separating various parts of a system or network to prevent unwanted access or malware distribution. This principle is also commonly employed in other areas of cybersecurity, such as network and cloud security. Organizations can limit the harm caused by security breaches and lower the danger of unauthorized access or data loss by isolating distinct components of a system or network. Here are some examples of how this principle can be applied in different areas of cybersecurity:

a) Network Isolation: Network isolation involves separating different parts of a network to prevent unauthorized access or the spread of malware. This can be accomplished through techniques such as firewalls, virtual private networks (VPNs), and network segmentation. For example, a company might use network segmentation to separate their finance department's network from the rest of the organization's network to protect sensitive financial information from unauthorized access.

b) Process Isolation: Process isolation involves separating different processes or applications running on a computer to prevent malware or other malicious code from spreading between them. This can be accomplished through techniques such as containerization and sandboxing. For example, a company might use containerization to isolate different applications running on a server to prevent a security breach in one application from spreading to others.

c) Data Isolation: Data isolation involves separating sensitive data from the rest of the system or network to prevent unauthorized access or data loss. This can be accomplished through techniques such as data encryption, access control, and secure storage. For example, a company might use encryption to protect sensitive customer data stored on their servers and restrict access to that data only to authorized personnel.

IV. CONCLUSION

In conclusion, the design principles of secure systems are critical to ensuring the confidentiality, integrity, and availability of data and systems. Through the above listed principles, designers can build systems that are resistant to attacks and recover from failures securely. The idea of least privilege ensures that users have only the access required to accomplish their obligations, decreasing the possibility of inadvertent or intentional system damage. Finally, the principle of failing securely ensures that the system can detect and respond to attacks or failures in a way that minimizes their impact.

REFERENCES

[1] G. S. Simson Garfinkel, Practical Unix and Internet securit, 2, Ed., Sebastopol, CA: O'Reilly & Associates, 1996.

[2] National Institute of Standards and Technology, "Security and Privacy Controls for Information Systems and Organizations," *NIST*, p. 492, 12 September 2020.

[3] National Cyber Security Center, "Secure design principles," National Cyber Security Center, 21 May 2019. [Online]. Available: Secure design principles. [Accessed 21 April 2023].

[4] M. D. S. Jerome H. Saltzer, "The Protection of Information in Computer System," Institute of Electrical and Electronics Engineers, [Online]. Available: http://web.mit.edu/Saltzer/www/publications/protection/. [Accessed 21 April 2023].

[5] Cybersecurity and Infrastructure Security Agency (CISA), "Improving Industrial Control System Cybersecurity with Defense-in-Depth Strategies," Red Hat, September 2016. [Online]. Available: https://www.cisa.gov/sites/default/files/recommended_practices/NCCIC_ICS-CERT_Defense_in_Depth_2016_S508C.pdf. [Accessed 23 April 2023].

[6] N. W. P. K. F. S. A. L. N. David Wagner, "Computer Security," 2023. [Online]. Available: https://textbook.cs161.org/#computer-security. [Accessed 30 April 2023].

[7] F. Schneider, "Least privilege and more," *IEEE Security & Privacy*, vol. 1, pp. 55 - 59, 14 October 2003.

[8] K. Q. G. A. W. Kevin M. Stine, "Framework for Improving Critical Infrastructure Cybersecurity,," National Institute of Standards and Technology, 19 February 2014. [Online]. Available: https://www.nist.gov/publications/framework-improving-critical-infrastructure-cybersecurity. [Accessed 30 April 2023].

[9] P. S. M. S. David Basin, Applied Information Security, Springer, Berlin, Heidelberg, 2011.

[10] D. Bourgeois, "INFORMATION SYSTEMS FOR BUSINESS AND BEYOND," Apple Platform Security, [Online]. Available: https://pressbooks.pub/bus206/chapter/chapter-6-information-systems-security/. [Accessed 30 April 2023].

[11] P. C. v. Oorschot, Computer Security and the Internet, 2, Ed., Springer Cham, 2021, p. 281–308.

[12] B. R. Richard Wilkins, "UEFI SECURE BOOT IN MODERN COMPUTER SECURITY SOLUTIONS," Microsoft, September 2013. [Online]. Available: https://media.kasperskycontenthub.com/wp-content/uploads/sites/63/2014/06/21032725/UEFI_Secure_Boot_in_Modern_Computer_Security_Solutions_2013.pdf. [Accessed 1 May 2023].

[13] Gartner, "Should Your Team Be Using Microservice Architectures?," *Information Technology Article : Gartner*, 20 August 2021.

[14] A. R. (NIST), "Platform Firmware Resiliency Guidelines," National Institute of Standard and Technology, May 2018. [Online]. Available: https://csrc.nist.gov/publications/detail/sp/800-193/final.

[15] The European Union Agency for Cybersecurity (ENISA), "Baseline Security Recommendations for IoT," 20 November 2017.

[i] The systems are a collection of digital components that are connected using communication technologies to perform a business function which include *hardware, software, and network infrastructure.*