SIL

At the edge of digital innovation

eGov Conference 2024

Smart Governance with GovTech

Enhancing Citizen Experience and Engagement

08th April 2024

www.sil.mu

Application-Level Security

Agenda

What is Application Level Security (ALS)

Importance of Application Level Security

Risks of not implementing Application Level Security

Application Security Tools and Solutions

What is Application-Level Security ?

- Application-level security refers to the security measures and protocols implemented within software applications to protect them from various security threats and vulnerabilities.

- Unlike network or system-level security, which focuses on securing the infrastructure and communication channels, application-level security is concerned with safeguarding the application itself and the data it processes

Key Aspects of Application-Level Security

- Authentication and Authorization: only authorized users can access sensitive data or perform specific actions within the application.
- Data Encryption: Encrypting data at rest and in transit.
- Input Validation and Sanitization: Input validation ensures that user input meets expected criteria, while sanitization removes potentially malicious content before processing.
- Session Management: Generating and validating session tokens, enforcing session timeouts.

Key Aspects of Application-Level Security

- Error Handling and Logging: Implementing robust error handling mechanisms to gracefully handle errors and exceptions within the application.

- Security Configuration: This includes configuring security settings, such as HTTP headers, CORS policies, and content security policies, to mitigate common web application security risks.

- Secure Development Practices: Implementing OWASP (Open Web Application Security Project) Security Principles.

Importance of Application Level Security

Importance of Application-Level Security

- **Protecting Sensitive Data**: Application-level security helps safeguard sensitive information, such as user credentials, personal data, and financial details, from unauthorized access and misuse.

- **Preventing Data Breaches**: By implementing security measures like encryption, input validation, and access controls, application-level security helps prevent data breaches and unauthorized disclosure of sensitive information.

- **Mitigating Security Risks**: Applications are often targeted by attackers looking to exploit vulnerabilities for malicious purposes. Application-level security helps identify and mitigate these risks, reducing the likelihood of successful cyber attacks.

Importance of Application-Level Security

- **Compliance with Regulations**: Many industries and regions have regulations and compliance requirements regarding data protection and security. Implementing robust application-level security measures helps organizations comply with these regulations and avoid potential penalties or legal consequences.

- **Preventing Service Disruption**: Security incidents can lead to service disruptions, downtime, and financial losses. Application-level security helps minimize the impact of security incidents and ensures the continuity of operations.

Risks of not implementing Application Level Security

Not implementing application-level security can lead to various pitfalls, leaving systems vulnerable to a range of cyber threats such as data breaches, unauthorized access, and manipulation.

Risks of not implementing Application-Level Security

Case 1 :

“In 2013, Adobe experienced a security breach where attackers gained unauthorized access to customer IDs, encrypted passwords, and payment card information of approximately 38 million users. The breach was attributed to weak password encryption mechanisms and inadequate access controls.”

***Unauthorized Access**: Lack of proper authentication and authorization mechanisms can lead to unauthorized users gaining access to sensitive resources within the application.*

Risks of not implementing Application-Level Security

Case 2 :

"The Equifax data breach in 2017 compromised the personal information of approximately 147 million people. It was discovered that attackers exploited a vulnerability in a web application to gain access to sensitive data."

Data Breaches*: Without proper application-level security measures, sensitive data stored within an application can be compromised. This can result in significant financial losses, reputational damage, and legal consequences.*

Risks of not implementing Application-Level Security

Case 3 :

"In 2014, eBay experienced a security breach where attackers gained unauthorized access to a database containing encrypted passwords and other personal information of approximately 145 million users. The breach was attributed to compromised employee credentials."

Insecure Authentication Mechanisms*: Weak or poorly implemented authentication mechanisms can be exploited by attackers to gain unauthorized access to user accounts.*

Risks of not implementing Application-Level Security

Case 4 :

“The LinkedIn data breach in 2012 involved attackers stealing approximately 6.5 million hashed passwords. It was found that LinkedIn had inadequate session management controls, which allowed attackers to access user accounts using compromised credentials.”

Inadequate Session Management*: Failure to properly manage user sessions can result in session hijacking or fixation attacks, where attackers take control of valid user sessions.*

Application Security Tools and Solutions by SIL

Application Security tools and Solutions by SIL

SIL uses the following tools and solutions to identify, mitigate, and manage security risks on all the software development project through out the whole lifecycle of a project.

- Static Application Security Testing (SAST) Tools: SAST tools analyze source code, byte code, or binary code to identify security vulnerabilities and coding errors without executing the application by using SonarQube.
- Dynamic Application Security Testing (DAST) Tools: DAST tools assess running applications by sending malicious payloads and analyzing responses to identify vulnerabilities such as injection flaws, broken authentication, and insecure configurations by using OWASP ZAP (Zed Attack Proxy).

Application Security tools and Solutions by SIL

- Web Application Firewalls (WAFs): WAFs protect web applications from a variety of attacks by inspecting and filtering HTTP traffic based on defined security rules and policies.by using Barracuda WAF.

- Runtime Application Self-Protection (RASP) Tools: RASP tools are embedded within applications or application runtimes to detect and prevent security threats in real-time, providing additional protection against attacks such as SQL injection, cross-site scripting, and code injection by using SonarQube.

Thank You

SIL

2, Saint Georges Street, Port Louis

Republic of Mauritius

207 8000 I silmail@sil.mu