GLOBALRAIN

Practices for Secure Software Report

Table of Contents

Document Revision History

Version	Date	Author	Comments
1.0	04/16/2023	alfredo	

Client

ARTEMIS
FINANCIAL

Instructions

Submit this completed practices for secure software report. Replace the bracketed text with the relevant information. You must document your process for writing secure communications and refactoring code that complies with software security testing protocols.

- Respond to the steps outlined below and include your findings.
- Respond using your own words. You may also choose to include images or supporting materials. If you include them, make certain to insert them in all the relevant locations in the document.
- Refer to the Project Two Guidelines and Rubric for more detailed instructions about each section of the template.

Developer
Alfredo Romero

1. Algorithm Cipher

There are several encryption algorithms available, each with its own strengths and weaknesses. Therefore, the selection of an appropriate encryption algorithm cipher depends on the specific security needs and requirements of the system.

One widely used and secure encryption algorithm cipher is the Advanced Encryption Standard (AES). It is a symmetric key block cipher that was adopted by the National Institute of Standards and Technology (NIST) in 2001 as a replacement for the outdated Data Encryption Standard (DES).

AES uses a 128-bit block size and supports key lengths of 128, 192, or 256 bits. The cipher operates on a fixed number of rounds depending on the key size: 10 rounds for a 128-bit key, 12 rounds for a 192-bit key, and 14 rounds for a 256-bit key. AES uses a substitution-permutation network to encrypt and decrypt data.

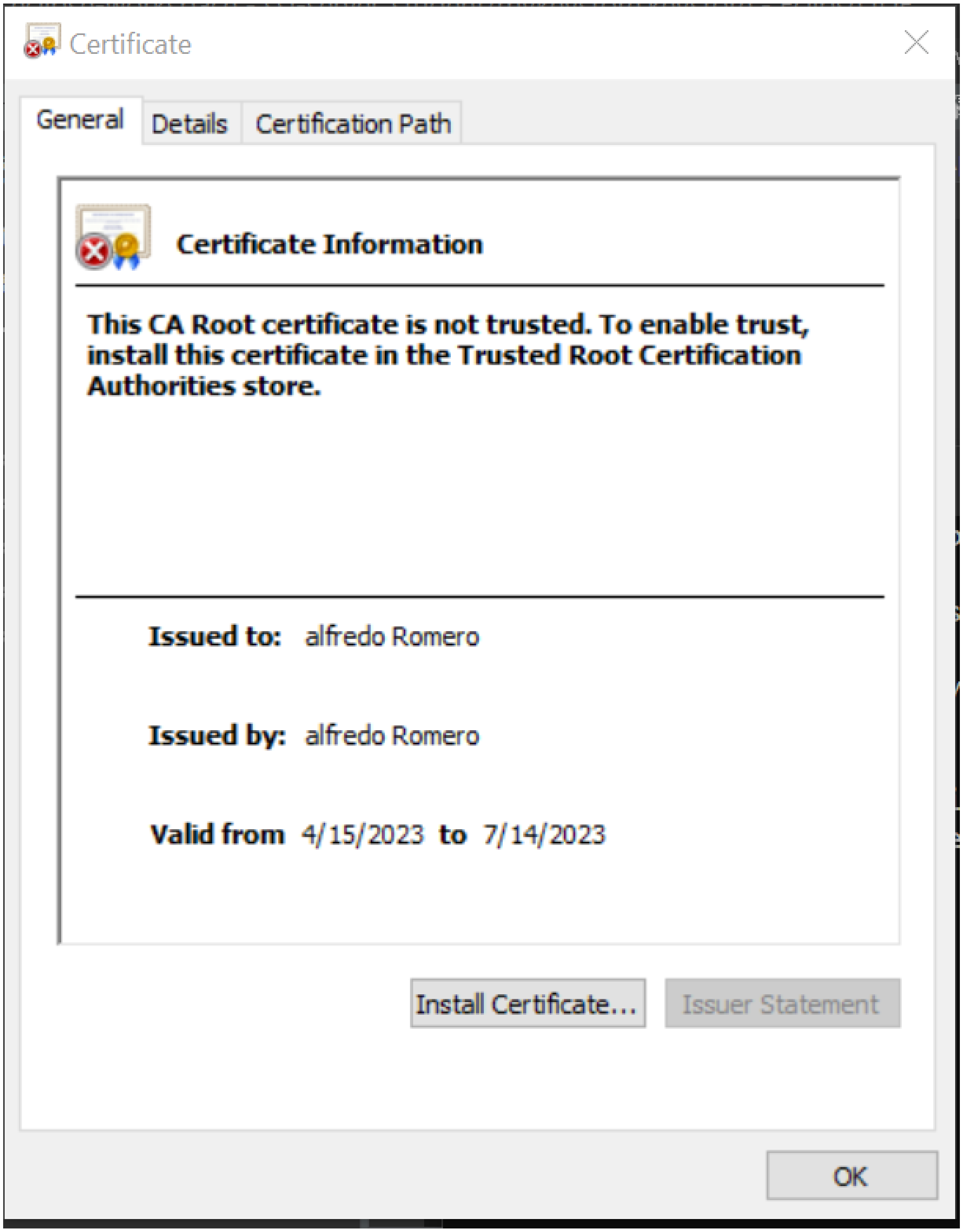
AES also incorporates hash functions such as SHA-256 for key derivation, which is used to derive a key from a password or passphrase, making it difficult for attackers to obtain the original key. Moreover, AES employs random numbers to enhance the security of the encryption process.

Symmetric key algorithms use the same key for both encryption and decryption. In contrast, non-symmetric key algorithms (also known as public-key cryptography) use two different keys: a public key for encryption and a private key for decryption. Symmetric key algorithms are faster and more efficient than non-symmetric key algorithms, but the latter provides better security in certain scenarios, such as key exchange over insecure channels.

The history of encryption algorithms dates back to the ancient times, where simple substitution ciphers were used to conceal messages. Over time, more complex encryption algorithms were developed to keep up with advancements in technology and cryptography research. In recent years, several encryption algorithms have been compromised, leading to the development of stronger and more secure ciphers such as AES.

In conclusion, AES is a widely adopted and secure encryption algorithm cipher that provides strong encryption and hash functions. It also uses random numbers to enhance security and can support different key sizes for varying levels of security. Therefore, AES is a suitable choice for most encryption scenarios, but the key size and other parameters should be chosen based on the specific security requirements of the system.

2. Certificate Generation



Certificate Information

This CA Root certificate is not trusted. To enable trust, install this certificate in the Trusted Root Certification Authorities store.

Issued to: alfredo Romero

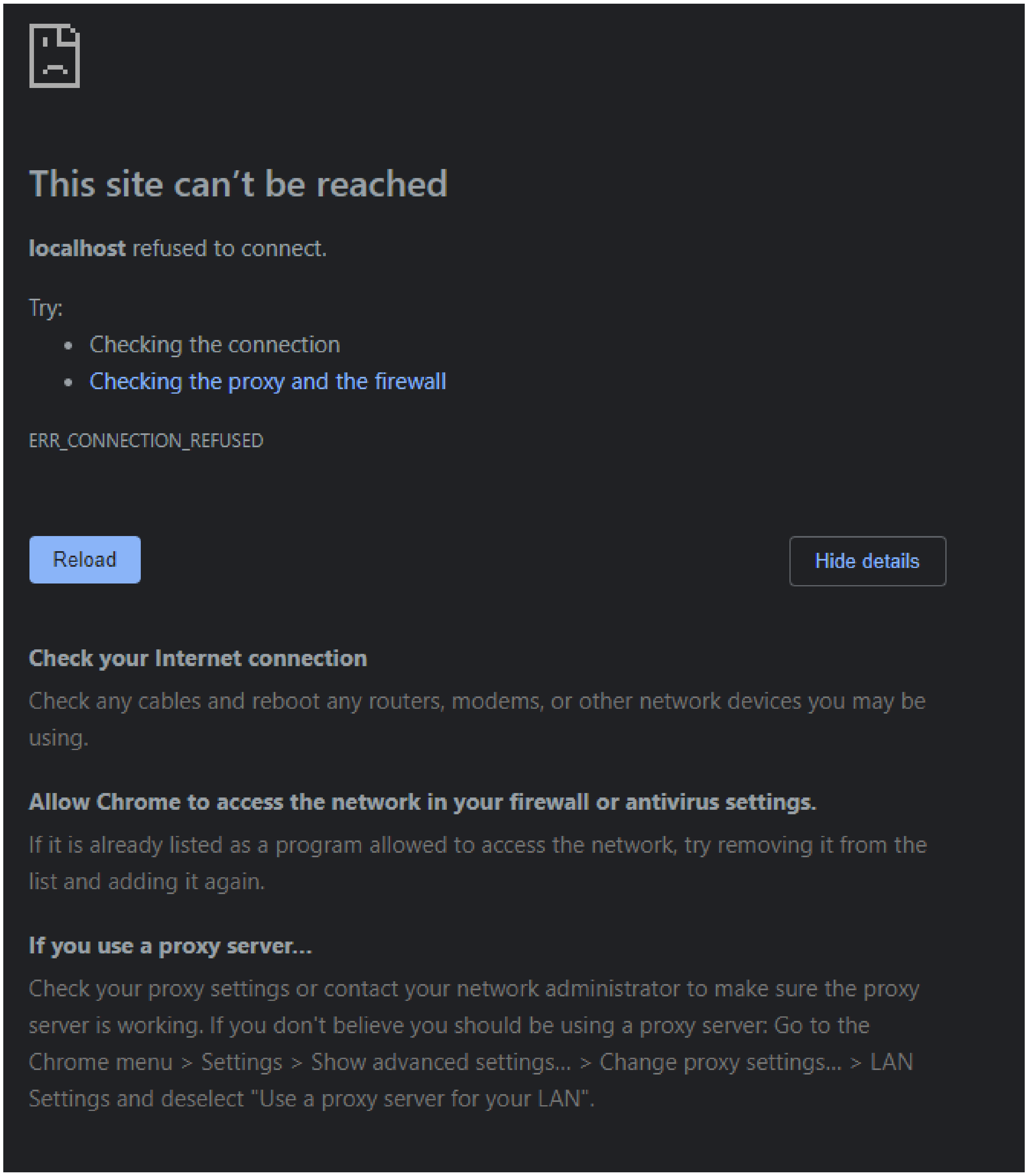
Issued by: alfredo Romero

Valid from 4/15/2023 **to** 7/14/2023

3. Deploy Cipher
Insert a screenshot below of the checksum verification.

[Insert screenshots here.]

4. Secure Communications



This site can't be reached

localhost refused to connect.

Try:

- Checking the connection
- Checking the proxy and the firewall

ERR_CONNECTION_REFUSED

Check your Internet connection

Check any cables and reboot any routers, modems, or other network devices you may be using.

Allow Chrome to access the network in your firewall or antivirus settings.

If it is already listed as a program allowed to access the network, try removing it from the list and adding it again.

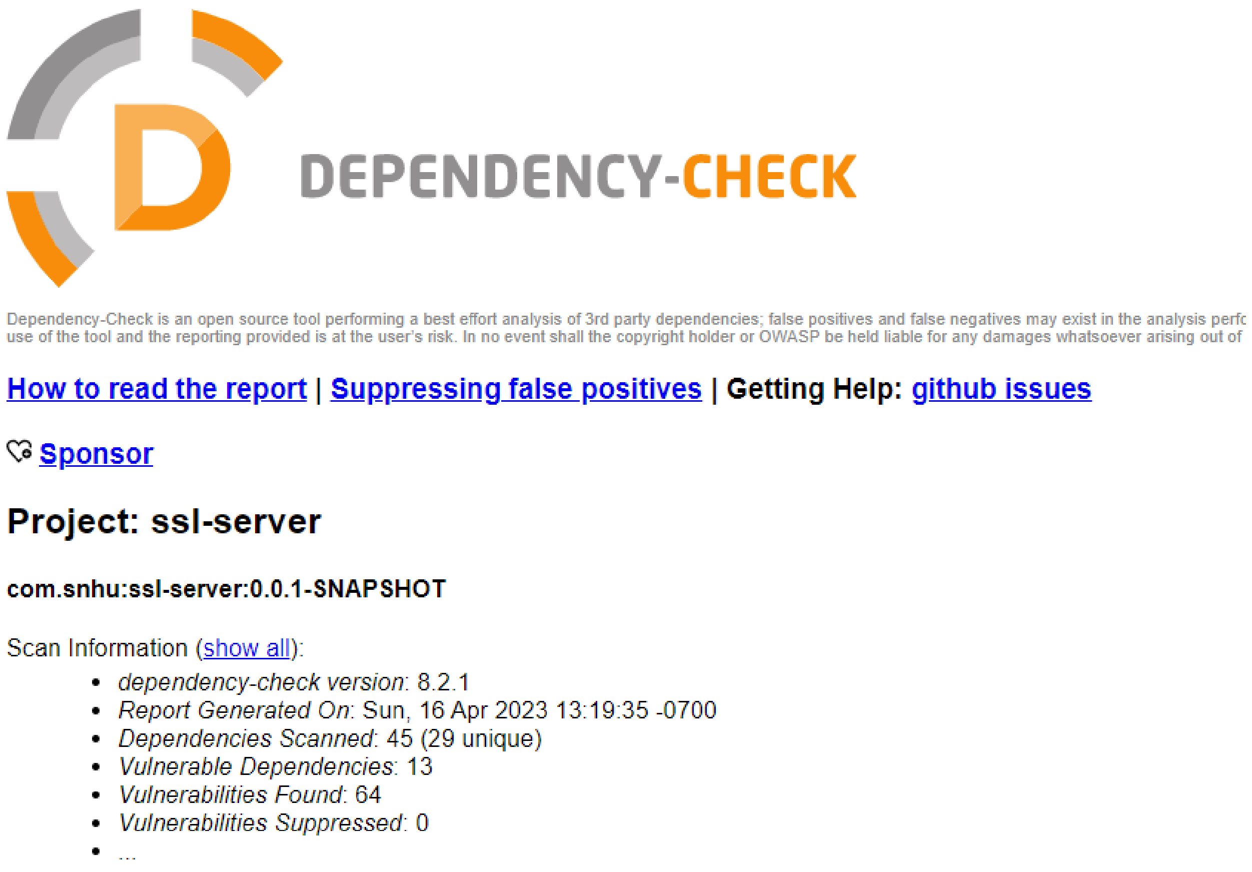
If you use a proxy server…

Check your proxy settings or contact your network administrator to make sure the proxy server is working. If you don't believe you should be using a proxy server: Go to the Chrome menu > Settings > Show advanced settings… > Change proxy settings… > LAN Settings and deselect "Use a proxy server for your LAN".

[Insert screenshots here.]

5. Secondary Testing

Insert screenshots below of the refactored code executed without errors and the dependency-check report.



DEPENDENCY-CHECK

Dependency-Check is an open source tool performing a best effort analysis of 3rd party dependencies; false positives and false negatives may exist in the analysis perfo
use of the tool and the reporting provided is at the user's risk. In no event shall the copyright holder or OWASP be held liable for any damages whatsoever arising out of

How to read the report | Suppressing false positives | Getting Help: github issues

Sponsor

Project: ssl-server

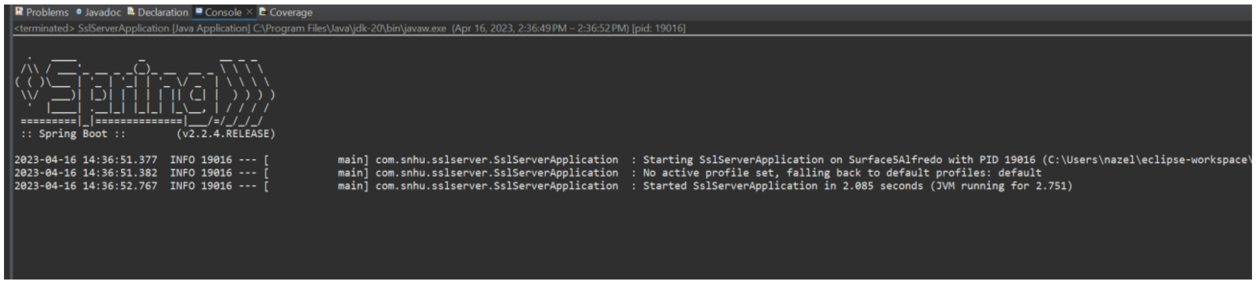
com.snhu:ssl-server:0.0.1-SNAPSHOT

Scan Information (show all):

- *dependency-check version*: 8.2.1
- *Report Generated On*: Sun, 16 Apr 2023 13:19:35 -0700
- *Dependencies Scanned*: 45 (29 unique)
- *Vulnerable Dependencies*: 13
- *Vulnerabilities Found*: 64
- *Vulnerabilities Suppressed*: 0
- ...

6. Functional Testing

Insert a screenshot below of the refactored code executed without errors.



```
<terminated> SslServerApplication [Java Application] C:\Program Files\Java\jdk-20\bin\javaw.exe (Apr 16, 2023, 2:36:49 PM – 2:36:52 PM) [pid: 19016]

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v2.2.4.RELEASE)

2023-04-16 14:36:51.377  INFO 19016 --- [           main] com.snhu.sslserver.SslServerApplication  : Starting SslServerApplication on Surface5Alfredo with PID 19016 (C:\Users\nazel\eclipse-workspace\
2023-04-16 14:36:51.382  INFO 19016 --- [           main] com.snhu.sslserver.SslServerApplication  : No active profile set, falling back to default profiles: default
2023-04-16 14:36:52.767  INFO 19016 --- [           main] com.snhu.sslserver.SslServerApplication  : Started SslServerApplication in 2.085 seconds (JVM running for 2.751)
```

[Insert screenshots here.]

7. Summary

we discussed how to generate a self-signed certificate using the Java Keytool. We outlined the steps involved in creating a new key pair, exporting the certificate, and importing it back into the keystore as a trusted certificate. We also addressed a common error message that can occur when trying to create a keystore in a directory without the necessary permissions. Overall, the Java Keytool provides a simple and effective way to generate self-signed certificates for use in testing and development environments. It is important to ensure that the necessary permissions are in place to avoid errors and ensure successful certificate creation.

we discussed various topics related to web application development using Spring Boot. We talked about the meaning and significance of different log messages that are typically generated during the development and deployment of a Spring Boot application, and how they can be used to identify issues and troubleshoot problems. We also touched upon the concept of URLs and endpoints in web development and what they signify. Overall, our conversation highlighted the importance of logging and monitoring in web application development, and how they can help ensure the smooth functioning of an application in production.

troubleshooting connection issues with a local server, specifically the error message "ERR_CONNECTION_REFUSED." We explored various steps to diagnose the problem, including checking network settings, firewall configurations, and ensuring that the server is actually running. We also used command line tools such as netstat and Get-NetTCPConnection to gather information about active connections. In conclusion, connection issues can have many different causes, but by systematically checking various settings and using diagnostic tools, it is often possible to identify and resolve the issue.

8. Industry Standard Best Practices

There are many industry standard best practices that can apply to various industries and domains, but here are a few general ones:

1. Use strong and unique passwords or passphrase and enable multi-factor authentication.
2. Keep software and systems up to date with the latest security patches.
3. Backup important data and perform regular disaster recovery testing.
4. Implement access controls to ensure that only authorized personnel can access sensitive data and systems.
5. Implement encryption and secure communication protocols to protect sensitive data in transit.
6. Perform regular security assessments and vulnerability testing to identify and remediate any security weaknesses.
7. Provide security awareness training to employees to help them recognize and respond to potential threats.
8. Have an incident response plan in place and test it regularly to ensure preparedness for any security incidents.

These practices can help organizations maintain a strong security posture and reduce the likelihood of security breaches and data loss.

best practices for certificate management:

1. Use trusted certificate authorities: Always obtain certificates from trusted certificate authorities (CA) rather than self-signing them. This ensures that the certificate can be verified by any client application that trusts the CA.

2. Use strong encryption algorithms and key sizes: Use encryption algorithms and key sizes that are currently considered to be strong and secure. This helps ensure that the certificate and the data it protects are not easily compromised.

3. Use certificate revocation lists (CRL) or Online Certificate Status Protocol (OCSP): Use CRLs or OCSP to revoke certificates in case they are compromised or are no longer needed.

4. Use separate certificates for different purposes: Use different certificates for different purposes, such as server authentication, client authentication, and code signing.

5. Rotate certificates regularly: Regularly rotate certificates to ensure that they are up-to-date and that their private keys have not been compromised.

6. Securely store and protect private keys: Private keys should be securely stored and protected from unauthorized access or theft.

7. Monitor certificates: Regularly monitor certificates for expiration and for any unusual activity that may indicate a compromise.

By following these industry standard best practices, organizations can ensure that their certificates are secure, trusted, and effective in protecting their data and applications.

Implementing security measures such as encryption, authentication, and authorization helps to protect user data and prevent unauthorized access to the application. Designing the application to be scalable ensures that it can handle a large number of users and requests without slowing down or crashing. Optimizing performance involves reducing database queries, minimizing network latency, and caching frequently accessed data.

Maintainable code should be modular and easy to understand, making it simpler to fix and update as needed. Testing the application through unit, integration, and end-to-end testing helps to identify and eliminate bugs. Logging and monitoring are crucial to track performance, identify issues and errors, and make necessary improvements.

Providing comprehensive documentation such as user guides and API documentation helps users and developers to effectively understand and use the application. By following these best practices, a web application can be made more reliable, robust, and efficient.

APA reference

1. Long, J. (n.d.). Dependency-Check Maven Plugin. Retrieved from https://jeremylong.github.io/DependencyCheck/dependency-check-maven/index.html

2. SSL Shopper. (n.d.). How to Create a Self-Signed Certificate Using Java Keytool. Retrieved from https://www.sslshopper.com/article-how-to-create-a-self-signed-certificate-using-java-keytool.html