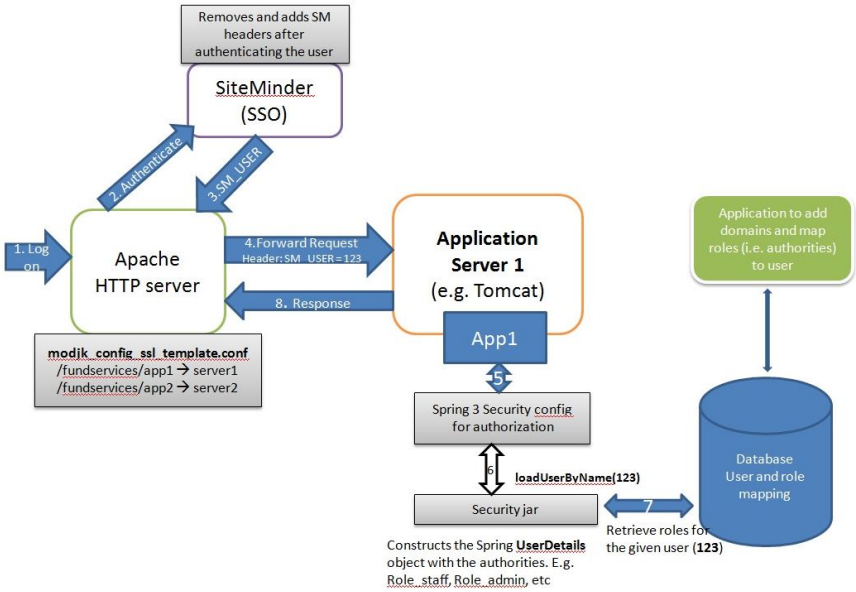
15 Security key area interview Q&A for Java developers

Posted on September 8, 2014 by Arulkumaran Kumaraswamipillai — No Comments ↓

Q1. Can you provide a high level overview of the “access control security” in a recent application you had worked?
A1. As shown below, SiteMinder is configured to intercept the calls to authenticate the user. Once the user is authenticated, a HTTP header “SM_USER” is added with the authenticated user name. For example “123”. The user header is passed to Spring 3 security. The “Security.jar” is a custom component that knows how to retrieve user roles for a given user like 123 from a database or LDAP server. This custom component is responsible for creating a UserDetails Spring object that contains the roles as authorities. Once you have the authorities or roles for a given user, you can restrict your

application URLs and functions to provide proper access control.



Q2. Can you provide a high level overview of the "access control security" in a recent application you had worked?

A2. **Single sign-on (SSO)** is a session/user authentication process that permits a user to enter one name and password in order to access multiple applications. The process authenticates the user for all the applications they have been given rights to and eliminates further prompts when they switch applications during a particular session. For example, SiteMinder, TivoliAccessManager (i.e. TAM), etc provides SSO. As shown in the diagram above SiteMinder authenticates the user and adds the SM_USER HTTP header to the application. It removes all the "SM" headers and add them after authenticating the user. This prevents amy malicious headers being injected via the browser with plugins like "Firefox Modify headers".

Q3. How will you go about implementing authentication and authorization in a web application?

A3. Use SSO application like Siteminder or Tivoli Access Manager to authenticate users, and Spring security 3 for authorization as described in the following Spring 3 security tutorials. Spring security pre-authentication scenario assumes that a valid authenticated user is available via either Single Sign On (SSO) applications like Siteminder, Tivoli, etc or a

X509 certification based authentication. The Spring security in this scenario will only be used for authorization.

Q4. What tools do you use to test your application for security holes?
A4. These tests are known as PEN (i.e. penetration) testing or security vulnerability testing. There are tools like

— **SkipFish** (web application security scanner) from Google.
— **Tamper data** from Firefox.

Q5. What is a two factor authentication?
A5. **Two-factor authentication** is a security process in which the user provides two means of identification. This includes

— something you have and something you know. For example, a bank card is which something you have and a PIN (i.e. Personal Identification Number) is something you know.

— two forms of identification like password and a biometric data like finger print or voice print. Some security procedures now require three-factor authentication, which involves possession of a physical token and a password, used in conjunction with biometric data.

Q6. What are the different layers of security?
A6. **Application-Layer Security:** For example, Spring 3 Security, JAAS (Java Authentication and Authorization) that provides a set of APIs to provide authentication and authorization (aka access control), etc. JAAS provides pluggable and extendable framework for programmatic user authentication and authorization at the JSE level (NOT JEE level). JAAS provides security at the JVM level (e.g. classes, resources). JAAS is the the core underlying technology for JEE Security. Spring security tackles security at the JEE level (e.g. URLs, Controller methods, service methods, etc)

Transport-Layer Security: Java Secure Sockets Extension (JSSE) provides a framework and an implementation for a Java version of the Secure Sockets Layer (SSL) and

16 Technical Key Areas

80+ step by step Java Tutorials

Transport Layer Security (TLS) protocols and includes functionality for data encryption, server authentication, message integrity, and optional client authentication to enable secure Internet communications. (TLS) 1.0 / (SSL) 3.0, is the mechanism to provide private, secured and reliable communication over the internet between the client and the server. It is the most widely used protocol that provides HTTPS for internet communications between the client (web browsers) and web servers.

Message-Layer Security: In message-layer security, security information is contained within the SOAP message and/or SOAP message attachment, which allows security information to travel along with the message or attachment. For example, the credit card number is signed by a sender and encrypted for a particular receiver to decrypt. Java Generic Security Services (Java GSS-API) is a token-based API used to securely exchange messages between communicating applications. The GSS-API offers application programmers uniform access to security services on top of a variety of underlying security mechanisms, including Kerberos. The advantage of this over point to point transport layer security is that the security stays with the message over all hops and after the message arrives at its destination. So, it can be used with intermediaries over multiple hops and protocols (e.g. HTTP, JMS, etc). The major disadvantage is that it is more complex to implement and requires more processing.

Note: Simple Authentication and Security Layer (SASL) is a framework for authentication and data security in Internet protocols. SASL is an Application-Layer security that supports TLS to compliment the services offered SASL.

Q7. What do you understand by the terms trusstores and keystores in Java?
A7. You generally need a **truststore** that points to a file containing trusted certificates, no matter whether you are implementing the server or the client side. You may or may not need a keystore. The **keystore** points to a file containing private key. You need a keystore if

JEE Tutorials (19)
Scala Tutorials (1)
Spring & HIbernate Tu
Tools Tutorials (19)
Other Tutorials (45)

100+ Java pre-interview coding tests

open all | close all
Can you write code? (
♦ Complete the given
Converting from A to I
Designing your classe
Java Data Structures
Passing the unit tests
What is wrong with th
Writing Code Home A
Written Test Core Jav
Written Test JEE (1)

How good are your .....?

open all | close all
Career Making Know-
Job Hunting & Resum

1) you are implementing the server side of the protocol, or
2) you are implementing the client side and you need to authenticate yourself to the server.

The **keystore** will be used for encrypting/signing some thing with your private key while the trust stores will be used mostly to authenticate remote servers. You can use the command line based "**keytool**" application that is shipped with your JDK to workwith keystores/trustores and certificates. For example,

To import parent certificate into your keystore

```
keytool -importcert -alias myservices -file mydom
```

To view the certificates

```
keytool -list -keystore truststore.jks
```

A Java client like RESTful web service client using 2 way SSL can configure the truststore in the JVM as shown below

```
java -Djavax.net.ssl.trustStore=C:\whatever\trust
```

Q8. How do you go about resolving any SSL related installation issues?
A8. There are several SSL tools that are available that can help you determine SSL problems and get your servers running SSL properly. OpenSSL is an open source implementation of the SSL protocol, and by far the most versatile SSL tool.

Q9. What is a one-way SSL? What is 2-way SSL?
A9. **One way SSL** just means that the server does not validate the identity of the client. The client generates a random key, encrypts it so that only the server can decrypt it, and sends it to the server. The server and client now have a shared secret that can be used to encrypt and validate the communications in both directions.

In two-way SSL authentication, the SSL client application verifies the identity of the SSL server application, and then the SSL server application verifies the identity of the SSL-client application. Two-way SSL authentication is also referred to as **client authentication** because the application (e.g. RESTful Web service client) acting as an SSL client presents its certificate to the SSL server after the SSL server authenticates itself to the SSL client.

Q10. How will you go about accomplshing password based encryption and decryption in Java?
A10. Password-Based Encryption/decryption can be accomplished using the PBEParameterSpec class. For example, if you are storing passwords for your database or RESTful web service connections in your properties file, they need to be encrypted and base64 encoded. You can't have clear text passwords lying around in properties files, URLs, or database tables. Base64 encoding helps you compactly transport binary data.

Q11. How can you perform role checks in a JEE application?
A11. IsUserInRole(), IsCallerInRole(), etc.

Q12. How can the current user be obtained in a JEE application?
A12. getUserPrincipal, getCallerPrincipal, etc.

Q13. What are some security issues in Internet based applications?
A13. authentication breach, authorization breach, data encryption flaws, denial of service attacks, xss(cross-site scripting) attacks, SQL injection attacks, etc.

Q14. How will you prevent SQL injection attacks in Java?
A14. By using **PreparedStatements** over normal Statements.

Q15. What are some of the things you will keep in mind to write a more secured Java applications?
A15. By using PreparedStatements over normal Statements.

#1. States are problematic from a security point of view due to sharing the state between the client and server. There are methods around this problem using encryption and other techniques, but again can complicate your solution. But favor stateless services where possible.

#2. Favoring immutable objects where applicable. String class was intentionally made immutable in Java for security reasons.

#3. Perform proper input validation to prevent any rogue characters. Both client side and server side validation.

#4. Always use Prepared or Callable statements. Stay away from ordinary statements.

#5. When using third-party frameworks, heed the warnings and best practices recommended by documentation.

#6. Handle exceptions properly, and don't let internal details like server names, database table names, etc to be displayed on the screen. Show generic exceptions where applicable like "An error has occurred, please contact support"

#7. Encrypt all sensitive information including the passwords in the .properties files.

#8. Subject your application to PEN testing before deploying to production.

Popular Posts

001A: ♦ 7+ Java integration styles & patterns interview questions & answers

389 views

01b: ♦ 13 Spring basics Q8 – Q13 interview questions & answers

296 views

♦ 7 Java debugging interview questions & answers

293 views

01: ♦ 15 Ice breaker questions asked 90% of the time in Java job interviews with hints

286 views

♦ 10 ERD (Entity-Relationship Diagrams) Interview Questions and Answers

279 views

♦ Q24-Q36: Top 50+ Core on Java classes, interfaces and generics interview questions & answers

240 views

001B: ♦ Java architecture & design concepts interview questions & answers

202 views

Bio | **Latest Posts**

Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

About Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job

interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

‹ 4 FAQ Performance tuning in Java interview Q&As

♦ 12 Java design patterns interview questions & answers ›

Posted in Java Key Area Essentials, member-paid, Security

Tags: Architect FAQs

Leave a Reply

Logged in as geethika. Log out?

Comment

Post Comment

Empowers you to open more doors, and fast-track

Technical Know Hows

☀ Java generics in no time ☀ Top 6 tips to transforming your thinking from OOP to FP ☀ How does a HashMap internally work? What is a hashing function?
☀ 10+ Java String class interview Q&As ☀ Java auto un/boxing benefits & caveats ☀ Top 11 slacknesses that can come back and bite you as an experienced Java developer or architect

Non-Technical Know Hows

☀ 6 Aspects that can motivate you to fast-track your career & go places ☀ Are you reinventing yourself as a Java developer? ☀ 8 tips to safeguard your Java career against offshoring ☀ My top 5 career mistakes

Prepare to succeed

☀ Turn readers of your Java CV go from “Blah blah” to “Wow”? ☀ How to prepare for Java job interviews? ☀ 16 Technical Key Areas ☀ How to choose from multiple Java job offers?

Select Category ▼

© Disclaimer