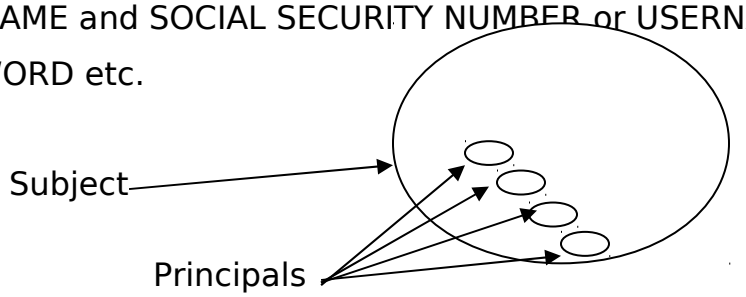
LIYANA ARACHCHIGE RANIL

SHORT NOTES ON JAVA SECURITY

1. JAAS stands for Java Authentication and Authorization Service
2. JAAS API can be used to Authenticate users as well as Authorization of users
3. Authorization must always be performed after a successful Authentication happened
4. For Authentication, JAAS provides Login Modules. Other than that users can develop proprietary login modules as well
5. Authentication can be used along with Servlet and JSP authentication mechanism where by entries in the web.xml help in authentication
6. web.xml can support BASIC , DIGEST , FORM and CLIENT-CERT authentication
7. BASIC , FORM authentication would transfer user credentials as plain text
8. BASIC authentication would provide a browser generated credential window while FORM based authentication would provide a custom developed form for authentication
9. FORM based authentication must make use , j_security_check , j_username , j_password fields and action URL for a successful login attempt
10. When user submits such a form , the application server in question would make use of it’s authorized roles and users for the authentication realm
11. Login Module contains usually UserNameCallBack , PasswordCallback as callbacks
12. Upon successful login , the module would return silently while failed login it would throw a security exception

13. DIGEST would encrypt the credentials with some encryption algorithm while BASIC one would use BASE-64 encoding
14. CLIENT-CERT authentication is based on PUBLIC KEY INFRASTRUCTURE authentication mechanism , in this case x.509 standard certificates would be used
15. When an application uses JAAS for authentication, it would create a SUBJECT which comprises of one or many PRINCIPALS. A PRINCIPAL is a representation of user's IDENTITY. Could be USERNAME and SOCIAL SECURITY NUMBER or USERNAME and PASSWORD etc.



16. Permission are granted to specific PRINCIPAL(s) in the java policy file
17. Permissions can be granted to a code source as well. It can be specified from where the code originates
18. JAAS allows developers to control what code runs and also who runs the code
19. After a user is AUTHENTICATED , the PRINCIPAL is associated with the CURRENT SECURITY ACCESS CONTROL CONTEXT
20. Authorization is performed based on who is running the code and what permission is granted that. All the configuration are done in the policy file
21. To associate the PRINCIPAL with the current ACCESS CONTROL CONTEX use static method of the SUBJECT class **doAs** method and pass the PriviledgeAction or PriviledgeExceptionAction

instance where the RUN method is defined which would be executed upon successful authorization.

22. doAsPriviledge method is exactly same as doAs , except one difference between the number of parameter passes. doAsPriviledge takes one more parameter which specifies the AccessControlContext.
23. doAsPriviledge allows you to pass the AccessControlContext against which the permissions must be checked
24. If a NULL AccessControlContex is passed to **doAsPriviledge** method then that means the subject passed along must be associated with a new AccessControlContext
25. NULL AccessControlContext is useful in server environment , when a server needs to do security checks for different principals irrespective of the security context of the server , it can use NULL AccessControlContext for that
26. login.config configures different login modules available for the use while security.policy configures what permission are granted to a PRINCIPLE or a CODE SOURCE etc