Spring boot security

========

=> It is spring extension module that provides multiple readymade filters to enable security on spring boot MVC and spring boot rest applications.

Problem:

(and also on microservices)

websever /Application server

=>if the Security module is used in spring style then

it is called spring security

=> if the security module is used in spring boot style then it is called spring boot security

Solution::

browser window

browser window

Servlet container

to

Servlet

Filter comp

IOC container

Java class acting

as spring bean

Here spring bean Injection to ServletFilter

comp is not possible..becoz

ServletFiltercemp is not taken

as spring bean managed by IOC container and the Servlet filter comp

is managed by ServletContainer

=>In order inject Dependent spring bean to servlet filter comp, we need to take ServletFilter comp also as spring bean in IOC container

(1) request

ServletContainer

web server or application server

DelegatingFilterProxy

ready ServletFilter) (2)

[url pattern:/

filtername: f1

(3)

IOC container

java elass

ServletFilter

comp (as spring Jean(4)

bear id: f1

This time

as spring bean

(5)

Dependency Injection (@Autowired)

is possible

(Tagret spring bean) (Dependent spring bean)

Spring Security/Spring boot security

can be applied spring/spring boot

web applications and also non spring/spring boot web applications

from

if we take servlet filter comp as spring bean in IOC container then any another dependent spring bean object can be injected to that Servlet filter spring bean.. but that servlet filter can not trap and take requests given browser window.. For this we need to cfg special ready made ServletFilter comp given by spring security module that is "DelegatingFilterProxy" having url pattern "/" and logical name matching with ServletFilter comp's bean id.belonging to IOC container

w.r.t diagram

==========

(1) browser gives the request to web application

(2) DelegatingFilterProxy traps and takes the request

(3)DelegatingFilterProxy delegates/passes the request to ServletFilter comp that is acting as spring bean in IOC container (Searches for that ServletFilter comp whose (4) the logics in ServletFilter comp executes

(5) ServletFilter comp uses the injected dependent spring bean services through method calls.

=>Security in web application is nothing but enabling Authentication + Authorization on the web application

bean id matches with DelegatingFilterProxy's logical name)

Authentication:: checking the indentity of user using usernames, passwords, thumb impressions, iris, digital signatures and etc.. Authorization :: Checking the access permissions of the autheticated users on different resources of the Project.

(Here the roles of the user will be verified before allowing the user to access different resources of the Project) =>Roles are nothing but designations given to users.. based on roles of the users the access permissions on different resources will be decided.

=>All customers, employees of the bank must be authenticated to use bankApp..

=>The users having customer role will get less access peromission on the resoruces where as

the users having employee role will get more access peromission on the resoruces.

=> It is always recomanded to enable authorization of accessing the resoucers not based on the username.. It is recomanded to perform on the roles of the users.

=>during the authentication process.. get the roles of the user and use those roles for authorization.

resource1

enduser

401 error

Authentication

get roles from

realm

succeded

resource2

(Authorization) (access permissions

resource3

verification) gets

on resources based

the roles

on the roles

(uname, pwd and

(if authentication fails)

403 error (if authorization fails) |

The comps of Security implementation

a) Authentication provider/a) Authentication Info provider

b) Authentication and Authorization manager

a) Authentication provider/a) Authentication Info provider /Securty Realm

It is the small realm where usernames, passwords,roles are managed and will be used during both authentication and authorization. The possible places are

a) Properties file

b) xml file file

c) json file

d) Db s/w

e) LDAP server (best)

g) InMemory DB

and etc..

LDAP:: Lightweight Directory Access Protocol

realm means small DB s/w or repository where the usernames, passwords and roles are managed.

=>In all other Security Realms exception LDAP server,the password recovery is possible where as in LDAP server password recovery is not possible only Password Reset is possible

=>In all other Authentication providers we can get the password of the users

if they forgot the passwords.. where in LDAP server .. no proivision to get

password.. only resetting of password is possible. (Here pwd recovery is not possible only

b) Authentication and Authorization manager

note:: Security realm means it is the small storage place where the username, passwords and roles will be stored and managed

=>It is the component that verifies given username and password to perform authentication and gives 401 error if authenticatio fails. for

resetting is possible)

=>The same component collects the roles of authenticated user and performs Authorization activities accessing different resources.. gives 403 error if authroization fails.

=>The Authentication and Authorization manager can be arranged in two ways

a) Using Programatic approach (bad)

=>Here we need to develop the logics of Authentication and Authorization manager explicitly

by spending huge amounts of time and brain manually (Not recomanded)

b) Using Declarative approach (good)

=> Max Web server /App servers provide built-in security support by providing built-in Authentication and Authorization manager ... By adding entries web.xml file we can activate

that Authentication and Authorization manager (refer security in web application in servlket,jsp env..)

with

Limitations of Declaration approch for securiing web applications (working servlet container

====== supplied Authentication and Authorization manager

a) Only selected servers support this features.. some times we need to purchase License of costly web server or Application server to use this feature

b) As of now this facility possible by adding addtional entries in web.xml file .. i.e not suitable

in 100%ven cfgs, spring boot apps (where xml is totally eliminated)

c) Whenfiove from one server to another server these configuration in web.xml file may change

(web.xml entries related to security are not portable across the multiple servers)

d) No support for LDAP server as authentication info provider.

through web.xml entries)

and etc..

To overcome these problems we use spring security or spring boot security

Advantages of spring security/spring boot security

==========

==========

in

a) Can be used in spring MVC/spring rest/spring micros services apps and also non-spring based web applications like servlet,jsp web applications, JSF web applications and etc...

b) The security cfgs code are portable across the multiple servers

c) We can this security irrespective of wheather the underlying server supports the ServletContainer

level Declarative security service or not (spring security no way related Servlet container's security)

d) Supports different Authentication Info providers includng LDAP.

e) We need not to arrange costly servers only for security..

In spring env or spring boot env.. we can apply security on MVC apps or spring rest apps or micro services Apps in

3 approaches

a) Using Spring security/spring boot secuirty

the

i) Basic Authentication (browser generates dialog box asking username, password)

ii) Form based Authentication (readymade or user-defined form page will be there asking username, password) (best)

-> Using InMemory DB as authentication Info provider (RAM Level DB)

-> Using Properties file as authentication Info provider

-> Using DB s/w as authentication info provider

with the support of spring JDBC/ sprjing ORM/spring data JPA / UserDetails service ->Using LDAP server as authentication info provider.

b) Using JWT (JSON web tokens) (loosing its popularity)

c) Using Oauth 2.x (Open Authorization) / okt [we can use FB, google, twitter login

Spring Boot security

=========

starter

for $50 feature)

SSO --- Single Sign On

=>Once we add spring-boot-stater-security to spring MVC /spring Rest/ microservices project one ready made filter called "DelegatingFilterProxy" will be registered with "/" url pattern having logical name "springSecurityFilterChain" .. For this One class will be genearted exteing from AbstractSecurityWebApplicationInitializer(AC) class (Internal class)

=> we need to develop ConfigurationAdapter class as @Configuration class extending from WebSecurityConfigurerAdapter overriding two configure(-) methods having authentication info provider, authentication, authorization details. Every details we add in authentication and authorization seperate InMemory Filters will be generated as spring beans and they will be linkedelegatingFilterProxy filter i.e DelegatingFilterProxy traps the request and links these requests dynamically generate filters then passes to controller clases through DispatcherServlet.

@Configuration

@EnableWebSecurity

It is in Spring boot 2.x

==========

=====

public class SpringSecurityConfig extends WebSecurityConfigurerAdapter {

@Override

public void configure(AuthenticationManagerBuilder auth) throws Exception {

// provide logic for configuration Authentication Info provider like In MemoryDB, DB s/w and etc..

```
}
@Override
public void configure(HttpSecurity http) throws Exception {
// provide logic for Authentication and authorization and etc..
}
}
```

App (spring /spring boot MVC or Rest App /MicroService)

authentication

in Spring boot 3.x, we can take this Security configuration

class directly a s normal @Configuration class with out extending from other classes

browser

request

to controller1

(1)

Delegating FilterProxy (2) // url pattern)

info-provibler

0 0 0

spring securty

generated In

(3)

Memory Filters

Dispatcher(4) Servlet

based on our requirements

controller1

controller? controller3

cfg in configure(HttpSecrutiy http) method

One these filters even contains spring security

supplied Authentication manager.

Authorization levels in spring security/ Spring boot Security

a) permitAll() :: No authentication + No Authorization (no role checking)

eg:: home page, about us page, contact us page, terms and conditions page

b) authenticate() :: Only Authentication on the given request url resource(controller) and no authorization (no role checking)

c) hasARole() ::

d) hasAnyRole ::

eg:: main menu page,inbox page,send/compo'se mail page

Authentication + authorization (role checking will be there)

eg:: checking balance page, transfer money, withdraw/deposite money page,changing password deleting mails and etc..

Authentication +authorization (any one role should be there for user

in the list of given roles)

eg:: checking balance page, transfer money,withdraw/deposite money page,changing password deleting mails and etc..

controller classes request paths

/home /contactUs /aboutUs

/inbox -----

-----> permitAll

--> permitAll

--->permitAll

authenticated

/checkbalance -->hasAnyRole "USER", "MANAGER"

/transferMoney ----> hasARole "USER"

/depositeMoney ----> hasAnyRole "USER","MANAGER","VISITOR"