


# Fun With Spring Security

Burt Beckwith

# Grails Spring Security Core Plugin

| | |
|---|---|
| **Version** | 3.0.3 |
| **Grails Version** | 3.0.0 > * |
| **Author** | Burt Beckwith |

---

## Current Documentation

- User guide
- User guide PDF
- User guide EPUB
- User guide MOBI (Kindle)

## Documentation (version 2.0.x)

- User guide
- User guide PDF

---

## Download Source



You can download this project in either zip or tar formats.

You can also clone the project with Git by running:

```
$ git clone git://github.com/grails-plugins/grails-spring-security-core
```

# It Doesn't Work … Please Help

- Step one is always to enable debug logging
- Grails 2

```
log4j = {
   debug 'org.springframework.security',
   trace 'grails.plugin.springsecurity'

   // optional
   debug 'org.hibernate.SQL'
   trace 'org.hibernate.type.descriptor.sql.BasicBinder'
}
```

- Grails 3+

```
logger 'org.springframework.security', DEBUG
logger 'grails.plugin.springsecurity', TRACE

// optional
logger 'org.hibernate.SQL', DEBUG
logger 'org.hibernate.type.descriptor.sql.BasicBinder', TRACE
```

```
269 DE 09:24:56.838 o.s.s.a.h.RoleHierarchyImpl                    - buildRolesReachableInOneOrMoreStepsMap() - From role ROLE_ADMIN one
    can reach [ROLE_USER] in one or more steps.
270 TR 09:24:56.840 g.p.s.SpringSecurityCoreGrailsPlugin           - Using SecurityContextHolder strategy MODE_THREADLOCAL
271 TR 09:24:56.843 g.p.s.SpringSecurityUtils                      - Ordered filters: {-2147483638=grails.plugin.springsecurity.web.
    SecurityRequestHolderFilter@60c83f3e, 300=org.springframework.security.web.context.SecurityContextPersistenceFilter@7cbb1658, 400=grails
    .plugin.springsecurity.web.authentication.logout.MutableLogoutFilter@5c446f51, 800=hacking.extralogin.ui.OrganizationFilter@1c4d7c58,
    1400=org.springframework.security.web.servletapi.SecurityContextHolderAwareRequestFilter@3ddd8120, 1500=grails.plugin.springsecurity.web
    .filter.GrailsRememberMeAuthenticationFilter@19f04915, 1600=grails.plugin.springsecurity.web.filter.
    GrailsAnonymousAuthenticationFilter@66b1839d, 1800=org.springframework.security.web.access.ExceptionTranslationFilter@11698caf, 1900=org
    .springframework.security.web.access.intercept.FilterSecurityInterceptor@40eccb23}
272 IN 09:24:56.845 g.p.s.w.GrailsSecurityFilterChain              - Creating filter chain: Ant [pattern='/assets/**'], []
273 IN 09:24:56.846 g.p.s.w.GrailsSecurityFilterChain              - Creating filter chain: Ant [pattern='/**/js/**'], []
274 IN 09:24:56.846 g.p.s.w.GrailsSecurityFilterChain              - Creating filter chain: Ant [pattern='/**/css/**'], []
275 IN 09:24:56.846 g.p.s.w.GrailsSecurityFilterChain              - Creating filter chain: Ant [pattern='/**/images/**'], []
276 IN 09:24:56.846 g.p.s.w.GrailsSecurityFilterChain              - Creating filter chain: Ant [pattern='/**/favicon.ico'], []
277 IN 09:24:56.847 g.p.s.w.GrailsSecurityFilterChain              - Creating filter chain: Ant [pattern='/**'], [grails.plugin.
    springsecurity.web.SecurityRequestHolderFilter@60c83f3e, org.springframework.security.web.context.
    SecurityContextPersistenceFilter@7cbb1658, grails.plugin.springsecurity.web.authentication.logout.MutableLogoutFilter@5c446f51, hacking.
    extralogin.ui.OrganizationFilter@1c4d7c58, org.springframework.security.web.servletapi.
    SecurityContextHolderAwareRequestFilter@3ddd8120, grails.plugin.springsecurity.web.filter.
    GrailsRememberMeAuthenticationFilter@19f04915, grails.plugin.springsecurity.web.filter.GrailsAnonymousAuthenticationFilter@66b1839d, org
    .springframework.security.web.access.ExceptionTranslationFilter@11698caf, org.springframework.security.web.access.intercept.
    FilterSecurityInterceptor@40eccb23]
278 TR 09:24:56.847 g.p.s.SpringSecurityCoreGrailsPlugin           - Filter chain: [[Ant [pattern='/assets/**'], []], [Ant [pattern='/**/
    js/**'], []], [Ant [pattern='/**/css/**'], []], [Ant [pattern='/**/images/**'], []], [Ant [pattern='/**/favicon.ico'], []], [Ant [
    pattern='/**'], [grails.plugin.springsecurity.web.SecurityRequestHolderFilter@60c83f3e, org.springframework.security.web.context.
    SecurityContextPersistenceFilter@7cbb1658, grails.plugin.springsecurity.web.authentication.logout.MutableLogoutFilter@5c446f51, hacking.
    extralogin.ui.OrganizationFilter@1c4d7c58, org.springframework.security.web.servletapi.
    SecurityContextHolderAwareRequestFilter@3ddd8120, grails.plugin.springsecurity.web.filter.
    GrailsRememberMeAuthenticationFilter@19f04915, grails.plugin.springsecurity.web.filter.GrailsAnonymousAuthenticationFilter@66b1839d, org
    .springframework.security.web.access.ExceptionTranslationFilter@11698caf, org.springframework.security.web.access.intercept.
    FilterSecurityInterceptor@40eccb23]]]
279 TR 09:24:56.849 g.p.s.SpringSecurityCoreGrailsPlugin           - AccessDecisionVoters: [org.springframework.security.access.vote.
    AuthenticatedVoter@3514237f, org.springframework.security.access.vote.RoleHierarchyVoter@520ee6b3, grails.plugin.springsecurity.web.
    access.expression.WebExpressionVoter@15f11bfb, grails.plugin.springsecurity.access.vote.ClosureVoter@16a499d1]
280 TR 09:24:56.850 g.p.s.SpringSecurityCoreGrailsPlugin           - AuthenticationProviders: [hacking.extralogin.auth.
    OrganizationAuthenticationProvider@3a12f3e7, grails.plugin.springsecurity.authentication.GrailsAnonymousAuthenticationProvider@84a9f65,
    org.springframework.security.authentication.RememberMeAuthenticationProvider@70700b8a]
281 TR 09:24:56.852 g.p.s.SpringSecurityCoreGrailsPlugin           - LogoutHandlers: [org.springframework.security.web.authentication.
    rememberme.TokenBasedRememberMeServices@5d2e65bd, org.springframework.security.web.authentication.logout.
    SecurityContextLogoutHandler@254513e8]
```

# Determining The Filter Chain

```
Checking match of request : '/secure/index'; against '/assets/**'
Checking match of request : '/secure/index'; against '/**/js/**'
Checking match of request : '/secure/index'; against '/**/css/**'
Checking match of request : '/secure/index'; against '/**/images/**'
Checking match of request : '/secure/index'; against '/**/favicon.ico'
Request '/secure/index' matched by universal pattern '/**'
```

# Walking The Filter Chain

```
FilterChainProxy - /secure/index at position 1 of 9 in additional
filter chain; firing Filter: 'SecurityRequestHolderFilter'
FilterChainProxy - /secure/index at position 2 of 9 in additional
filter chain; firing Filter: 'SecurityContextPersistenceFilter'
HttpSessionSecurityContextRepository - No HttpSession currently
exists
HttpSessionSecurityContextRepository - No SecurityContext was
available from the HttpSession: null. A new one will be created.
FilterChainProxy - /secure/index at position 3 of 9 in additional
filter chain; firing Filter: 'MutableLogoutFilter'
AntPathRequestMatcher - Checking match of request : '/secure/index';
against '/logoff'
FilterChainProxy - /secure/index at position 4 of 9 in additional
filter chain; firing Filter: 'OrganizationFilter'
FilterChainProxy - /secure/index at position 5 of 9 in additional
filter chain; firing Filter:
'SecurityContextHolderAwareRequestFilter'
```

# Walking The Filter Chain (cont.)

```
FilterChainProxy - /secure/index at position 6 of 9 in additional
filter chain; firing Filter: 'GrailsRememberMeAuthenticationFilter'
FilterChainProxy - /secure/index at position 7 of 9 in additional
filter chain; firing Filter: 'GrailsAnonymousAuthenticationFilter'
GrailsAnonymousAuthenticationFilter - Populated
SecurityContextHolder with anonymous token:
'grails.plugin.springsecurity.authentication.GrailsAnonymousAuthenti
cationToken@dc4337e: Principal:
org.springframework.security.core.userdetails.User@dc730200:
Username: __grails.anonymous.user__; Password: [PROTECTED]; Enabled:
false; AccountNonExpired: false; credentialsNonExpired: false;
AccountNonLocked: false; Granted Authorities: ROLE_ANONYMOUS;
Credentials: [PROTECTED]; Authenticated: true; Details:
org.springframework.security.web.authentication.WebAuthenticationDet
ails@957e: RemoteIpAddress: 127.0.0.1; SessionId: null; Granted
Authorities: ROLE_ANONYMOUS'
FilterChainProxy - /secure/index at position 8 of 9 in additional
filter chain; firing Filter: 'ExceptionTranslationFilter'
FilterChainProxy - /secure/index at position 9 of 9 in additional
filter chain; firing Filter: 'FilterSecurityInterceptor'
```

# Failed Login

```
FilterSecurityInterceptor - Secure object: FilterInvocation: URL:
/secure/index; Attributes: [ROLE_USER]

FilterSecurityInterceptor - Previously Authenticated:
grails.plugin.springsecurity.authentication.GrailsAnonymousAuthentic
ationToken

RoleHierarchyImpl - getReachableGrantedAuthorities() - From the
roles [ROLE_ANONYMOUS] one can reach [ROLE_ANONYMOUS] in zero or
more steps.

WebExpressionVoter - No WebExpressionConfigAttribute found

ClosureVoter - No ClosureConfigAttribute found

ExceptionTranslationFilter - Access is denied (user is anonymous);
redirecting to authentication entry point
```

# Failed Login (cont.)

```
ProviderManager - Authentication attempt using
hacking.extralogin.auth.OrganizationAuthenticationProvider

TokenBasedRememberMeServices - Interactive login attempt was
unsuccessful.

TokenBasedRememberMeServices - Cancelling cookie

AjaxAwareAuthenticationFailureHandler - Redirecting to
/login/authfail?login_error=1

GrailsRedirectStrategy - Redirecting to '/login/authfail?
login_error=1'

HttpSessionSecurityContextRepository - SecurityContext is empty or
contents are anonymous - context will not be stored in HttpSession.

SecurityContextPersistenceFilter - SecurityContextHolder now
cleared, as request processing completed
```

# Successful Login

```
AntPathRequestMatcher - Request '/login/authenticate' matched by
universal pattern '/**'

FilterChainProxy - /login/authenticate at position 1 of 9 in
additional filter chain; firing Filter:
'SecurityRequestHolderFilter'

FilterChainProxy - /login/authenticate at position 2 of 9 in
additional filter chain; firing Filter:
'SecurityContextPersistenceFilter'

HttpSessionSecurityContextRepository - HttpSession returned null
object for SPRING_SECURITY_CONTEXT

HttpSessionSecurityContextRepository - No SecurityContext was
available from the HttpSession:
org.apache.catalina.session.StandardSessionFacade@232c542c. A new
one will be created.

FilterChainProxy - /login/authenticate at position 3 of 9 in
additional filter chain; firing Filter: 'MutableLogoutFilter'
```

# Successful Login (cont.)

```
AntPathRequestMatcher - Checking match of request :
'/login/authenticate'; against '/logoff'

FilterChainProxy - /login/authenticate at position 4 of 9 in
additional filter chain; firing Filter: 'OrganizationFilter'

ProviderManager - Authentication attempt using
hacking.extralogin.auth.OrganizationAuthenticationProvider

RoleHierarchyImpl - getReachableGrantedAuthorities() - From the
roles [ROLE_USER] one can reach [ROLE_USER] in zero or more steps.

SessionFixationProtectionStrategy - Invalidating session with Id
'59DB715075D9716460E931ACEB6D60FB' and migrating attributes.

SessionFixationProtectionStrategy - Started new session:
5B4C50B452A7FA2D5E7253AF68EC9E15

TokenBasedRememberMeServices - Did not send remember-me cookie
(principal did not set parameter 'remember-me')
```

# Successful Login (cont.)

```
TokenBasedRememberMeServices - Remember-me login not requested.

AjaxAwareAuthenticationSuccessHandler - Redirecting to
DefaultSavedRequest Url: http://localhost:8080/secure/index

GrailsRedirectStrategy - Redirecting to
'http://localhost:8080/secure/index'

HttpSessionSecurityContextRepository - SecurityContext
'org.springframework.security.core.context.SecurityContextImpl@bbd64
3f3: Authentication:
org.springframework.security.authentication.UsernamePasswordAuthenti
cationToken@bbd643f3: Principal:
grails.plugin.springsecurity.userdetails.GrailsUser ... Granted
Authorities: ROLE_USER' stored to HttpSession

HttpSessionRequestCache - Removing DefaultSavedRequest from session
if present

SecurityContextPersistenceFilter - SecurityContextHolder now
cleared, as request processing completed
```

# Demo Apps

- Source is available at

  https://github.com/burtbeckwith/FunWithSpringSecurity

# Auto-assigning Roles

- You don't have to store all granted roles – some can be inferred
- Use existing attributes, a user class hierarchy, etc. in a custom

  `UserDetailsService`
- See the “autorole” demo app

# Not Using Roles

- You don't have to store role information at all; they can be *entirely* inferred
- See the “noroles” demo app

# Hacking New Delhi

- To demonstrate creating a custom authentication process, I updated the demo app from Greach 2015 (“hacking_madrid”) which was an update of the demo app from GGX 2011 (“hacking_london”)
- Updated to Grails 3, spring-security-core 3.0.3
- Adds an “organization” drop-down to the login page in addition to username and password
- See the “hacking_newdelhi” demo app

# Locking After Multiple Auth Fails

- Spring Security generates events for several scenarios
- One is a failed login because of an incorrect or missing password
- To slow down attackers attempting to hack your app by brute force, you can listen for `AuthenticationFailureBadCredentialsEvent` and lock the user's account after N failures
- See the “lockout” demo app

# X.509

- Spring Security supports using browser certificates to authenticate
- There aren't a lot of steps required to configure this, but it can be confusing
- See the “x509” demo app

# X.509 Chained

- I've been mulling over the idea of adding support for using more than one AuthenticationProvider to authenticate
- X.509 seems like a good candidate, since if you have physical access to a user's browser you can perform actions with their account; adding a second authentication phase (e.g. typical form auth) would be more secure
- See the “x509chained” demo app