Commit

[security-http] Check owner of persisted remember-me cookie

Browse files

7.2
v7.2.0-RC1 ··· v5.4.47

jderusse committed last week Verified 1 parent 2c7c4ba commit 81354d3

Showing **2 changed files** with **48 additions** and **5 deletions**.

Whitesp... Ignore whitespace Split Unifi...

Filter changed files

- src/Symfony/Component...
  - RememberMe
    - PersistentRemem...
  - Tests/RememberMe
    - PersistentRemem...

19

src/Symfony/Component/Security/Http/RememberMe/PersistentRem...

```
66  66              throw new AuthenticationException('The cookie is incorrectly formatted.');
67  67          }
68  68
69      -       [$series, $tokenValue] = explode(':', $rememberMeDetails->getValue());
    69  +       [$series, $tokenValue] = explode(':', $rememberMeDetails->getValue(), 2);
70  70          $persistentToken = $this->tokenProvider->loadTokenBySeries($series);
71  71
    72  +       if ($persistentToken->getUserIdentifier() !== $rememberMeDetails->getUserIdentifier() || $persistentToken->getClass() !== $rememberMeDetails->getUserFqcn()) {
    73  +           throw new AuthenticationException('The cookie\'s hash is invalid.');
    74  +       }
    75  +
    76  +       // content of $rememberMeDetails is not trustable. this prevents use of this class
    77  +       unset($rememberMeDetails);
    78  +
72  79          if ($this->tokenVerifier) {
73  80              $isTokenValid = $this->tokenVerifier->verifyToken($persistentToken, $tokenValue);
74  81          } else {
78  85              throw new CookieTheftException('This token was already used. The account is possibly
```

```
compromised.');
79  86          }
80  87
81      -       if ($persistentToken->getLastUsed()->getTimestamp() + $this->options['lifetime'] < time()) {
    88  +           $expires = $persistentToken->getLastUsed()->getTimestamp() + $this->options['lifetime'];
    89  +           if ($expires < time()) {
82  90                  throw new AuthenticationException('The cookie has expired.');
83  91          }
84  92
85      -       return parent::consumeRememberMeCookie($rememberMeDetails->withValue($persistentToken->getLastUsed()->getTimestamp().':'.$rememberMeDetails->getValue().':'.$persistentToken->getClass()));
    93  +       return parent::consumeRememberMeCookie(new RememberMeDetails(
    94  +           $persistentToken->getClass(),
    95  +           $persistentToken->getUserIdentifier(),
    96  +           $expires,
    97  +           $persistentToken->getLastUsed()->getTimestamp().':'.$series.':'.$tokenValue.':'.$persistentToken->getClass()
    98  +       ));
86  99      }
87  100
88  101     public function processRememberMe(RememberMeDetails $rememberMeDetails, UserInterface $user): void
```

34 ■■■■□

src/Symfony/Component/Security/Http/Tests/RememberMe/Persist…

```
80  80          $this->tokenProvider->expects($this->any())
81  81              ->method('loadTokenBySeries')
82  82              ->with('series1')
83      -           ->willReturn(new PersistentToken(InMemoryUser::class, 'wouter', 'series1', 'tokenvalue', new \DateTime('-10 min')))
    83  +           ->willReturn(new PersistentToken(InMemoryUser::class, 'wouter', 'series1', 'tokenvalue', $lastUsed = new \DateTime('-10 min')))
84  84          ;
85  85
86  86          $this->tokenProvider->expects($this->once())->method('updateToken')->with('series1');
```

```
98  98
99  99          $this->assertSame($rememberParts[0], $cookieParts[0]); // class
100 100         $this->assertSame($rememberParts[1], $cookieParts[1]); // identifier
101             -         $this->assertSame($rememberParts[2], $cookieParts[2]); // expire
    101         +         $this->assertEqualsWithDelta($lastUsed->getTimestamp() + 31536000, (int) $cookieParts[2], 2); // expire
102 102         $this->assertNotSame($rememberParts[3], $cookieParts[3]); // value
103 103         $this->assertSame(explode(':', $rememberParts[3])[0], explode(':', $cookieParts[3])[0]); // series
104 104     }
105 105
    106         +     public function testConsumeRememberMeCookieInvalidOwner()
    107         +     {
    108         +         $this->tokenProvider->expects($this->any())
    109         +             ->method('loadTokenBySeries')
    110         +             ->with('series1')
    111         +             ->willReturn(new PersistentToken(InMemoryUser::class, 'wouter', 'series1', 'tokenvalue', new \DateTime('-10 min')))
    112         +         ;
    113         +
    114         +         $rememberMeDetails = new RememberMeDetails(InMemoryUser::class, 'jeremy', 360, 'series1:tokenvalue');
    115         +
    116         +         $this->expectException(AuthenticationException::class);
    117         +         $this->expectExceptionMessage('The cookie\'s hash is invalid.');
    118         +         $this->handler->consumeRememberMeCookie($rememberMeDetails);
    119         +     }
    120         +
    121         +     public function testConsumeRememberMeCookieInvalidValue()
    122         +     {
    123         +         $this->tokenProvider->expects($this->any())
    124         +             ->method('loadTokenBySeries')
    125         +             ->with('series1')
    126         +             ->willReturn(new PersistentToken(InMemoryUser::class, 'wouter',
```

```
                'series1', 'tokenvalue', new \DateTime('-10 min')))
127 +           ;
128 +
129 +           $rememberMeDetails = new RememberMeDetails(InMemoryUser::class, 'wouter', 360, 'series1:tokenvalue:somethingelse');
130 +
131 +           $this->expectException(AuthenticationException::class);
132 +           $this->expectExceptionMessage('This token was already used. The account is possibly compromised.');
133 +           $this->handler->consumeRememberMeCookie($rememberMeDetails);
134 +       }
135 +
106 136        public function testConsumeRememberMeCookieValidByValidatorWithoutUpdate()
107 137        {
108 138            $verifier = $this->createMock(TokenVerifierInterface::class);
```

2 comments on commit `81354d3`

alexandre-le-borgne commented on `81354d3` yesterday

@jderusse If I understand correctly, everyone can simply enable "remember me", modify the cookie to add anyone's user id and log into their account?

jderusse commented on `81354d3` yesterday

Member Author

> everyone can simply enable "remember me"

No, the remember me behavior has to be enabled in the security configuration file. So it's not everyone, but developers.

Also, the vulnerability is only about **persisted** remember me feature (The default remember me implementation uses hash signature verification and is safe)

> modify the cookie to add anyone's user id and log into their account

correct

Please sign in to comment.