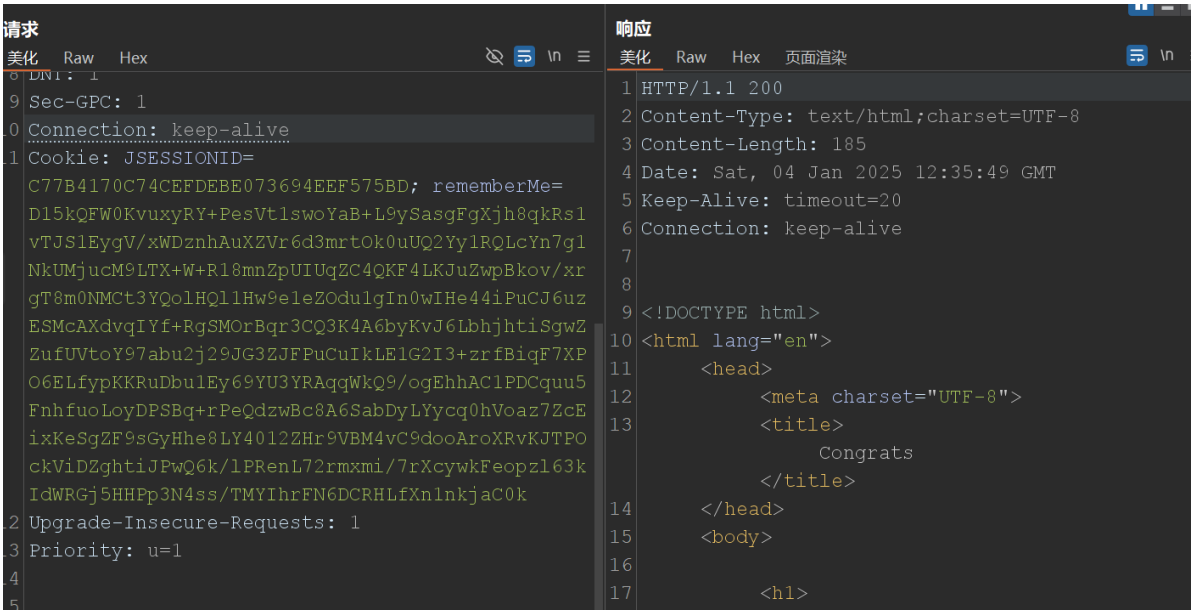
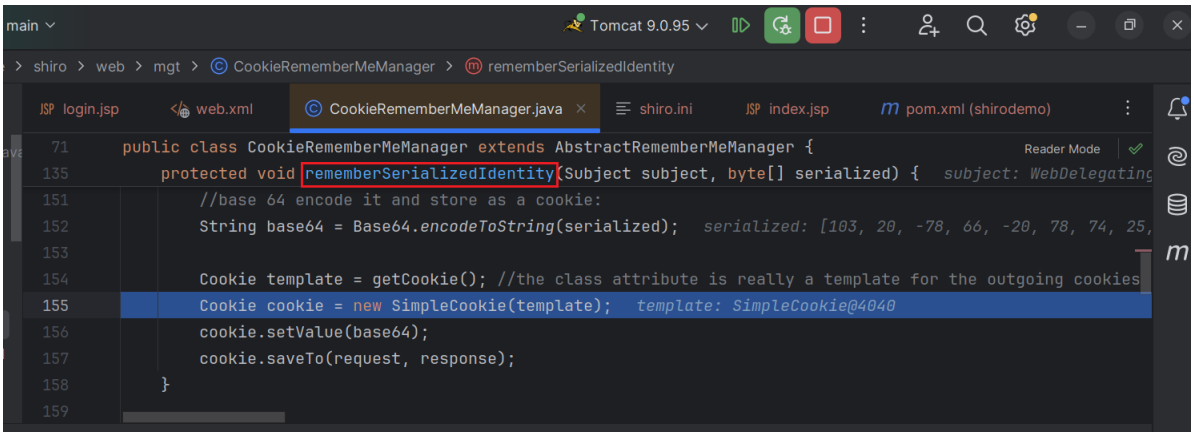
shiro <= 1.2.4

bp抓包可以看到这个cookie：



太长了！而且是加密后的。

这里搜索 rememberMe关键字，能找到这里：



调试也能断下。

往上findusage能回溯到AuthenticatingFilter

当然，我们关注的是加密和序列化的过程，不用那么回溯，看这里：

```
protected void rememberIdentity(Subject subject, PrincipalCollection accountPrincipals) {
    byte[] bytes = convertPrincipalsToBytes(accountPrincipals);
    rememberSerializedIdentity(subject, bytes);
}
```

```
protected byte[] convertPrincipalsToBytes(PrincipalCollection principals) {
    byte[] bytes = serialize(principals);
    if (getCipherService() != null) {
        bytes = encrypt(bytes);
    }
    return bytes;
}
```

这里就先序列化再加密了。

看加密：

```
JSP login.jsp  web.xml  CookieRememberMeManager.java  AbstractRememberMeManager.java  JSP index.jsp
64  public abstract class AbstractRememberMeManager implements RememberMeManager {   Reader Mode
            configured () cipher.
    protected byte[] encrypt(byte[] serialized) {
470     byte[] value = serialized;
471     CipherService cipherService = getCipherService();
472     if (cipherService != null) {
473         ByteSource byteSource = cipherService.encrypt(serialized, getEncryptionCipherKey());
474         value = byteSource.getBytes();
475     }
476     return value;
```

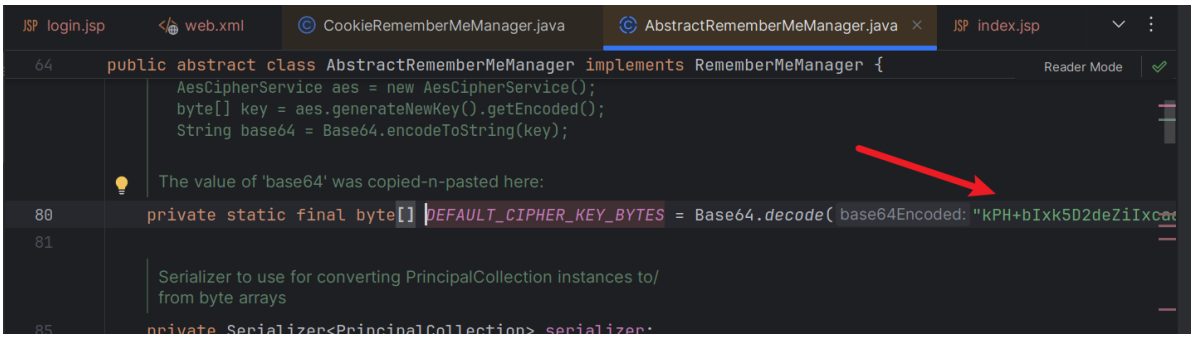
```
JSP login.jsp  web.xml  CookieRememberMeManager.java  AbstractRememberMeManager.java  JSP index.jsp
64  public abstract class AbstractRememberMeManager implements RememberMeManager {
    Returns the cipher key to use for encryption operations.
    Returns:  the cipher key to use for encryption operations.
    See Also: a description of the various {@code get/set*Key}
              methods.
181 public byte[] getEncryptionCipherKey() {
182     return encryptionCipherKey;
183 }
184
```

有意思，这个常量key。。。

逐步回溯可以找到这里：

```
JSP login.jsp  web.xml  CookieRememberMeManager.java  AbstractRememberMeManager.java  JSP index.jsp
64  public abstract class AbstractRememberMeManager implements RememberMeManager {   Reader
106 public AbstractRememberMeManager() {
107     this.serializer = new DefaultSerializer<PrincipalCollection>();
108     this.cipherService = new AesCipherService();
109     setCipherKey(DEFAULT_CIPHER_KEY_BYTES);
110 }
111
    Returns the Serializer used to serialize and deserialize
    PrincipalCollection instances for persistent remember me
```

这个 DEFAULT_CIPHER_KEY_BYTES



6。

然后这里的加密方法是AES。

现在的问题是，序列化的cookie在哪儿被反序列化导致漏洞的?

（待续）

2025年1月6日

继续来研究哪儿进行反序列化cookie的。

其实就是从

CookieRememberMeManager#getRememberedSerializedIdentity

find usages，找到所继承的AbstractRememberMeManager:

```
public PrincipalCollection getRememberedPrincipals(SubjectContext
subjectContext) {
    PrincipalCollection principals = null;
    try {
        byte[] bytes = getRememberedSerializedIdentity(subjectContext);
        //SHIRO-138 - only call convertBytesToPrincipals if bytes exist:
        if (bytes != null && bytes.length > 0) {
            principals = convertBytesToPrincipals(bytes, subjectContext);
        }
    } catch (RuntimeException re) {
        principals = onRememberedPrincipalFailure(re, subjectContext);
    }

    return principals;
}
```

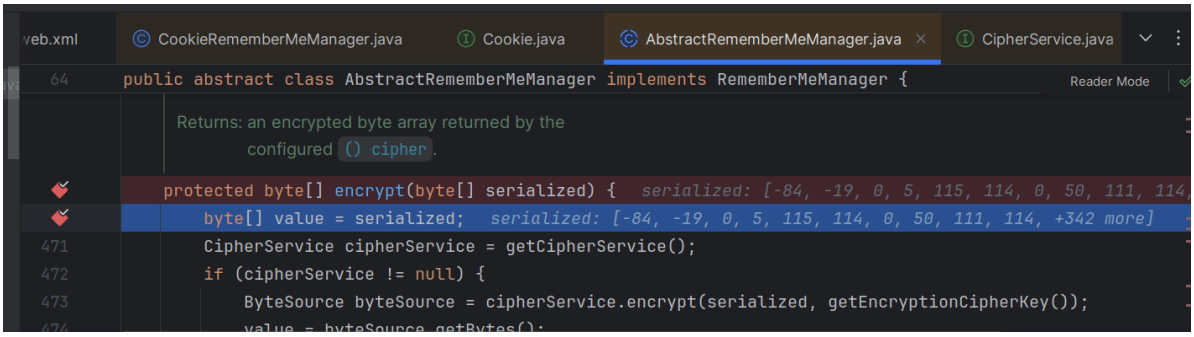
这里面的 `convertBytesToPrincipals`：

```
protected PrincipalCollection convertBytesToPrincipals(byte[] bytes,
SubjectContext subjectContext) {
    if (getCipherService() != null) {
        bytes = decrypt(bytes);
    }
    return deserialize(bytes);
}
```

先decrypt，然后反序列化~

可以debug再跟一遍流程。

我们输入正确用户名密码，勾选rememberMe后，先序列化 + 加密：



那我们怎么触发反序列化cookie呢?

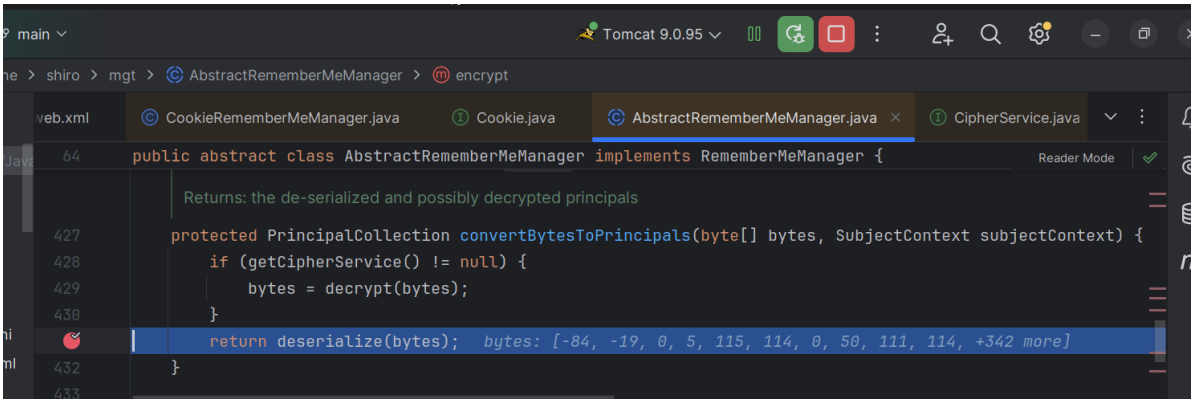
我们把cookie的JESSONID删掉。

请求

美化 Raw Hex

```
ed/login.jsp
8 DNT: 1
9 Sec-GPC: 1
10 Connection: keep-alive
11 Cookie: JSESSIONID=
6BA0BA832813BE6433D7B049876BFDB3; rememberMe=
6cvMGSSI37xC1dKl0havZSz40h01R8mVh3iu3Oxlm4V+MfD
WpQlGivxo84pDZDttZwMSVwgD02t4iy0ryvqVbMRqJkJYdJ
0vP4OhJxI9Wb+pxsBHvQq0AD0cSl7SIHPFP+wfE8NyVeMd/
tysPTPCkrDVlZ52cCyPWomPdu6zHTV0WR8oECfqGSKjWiiC
Q+BBrK6x6qYldxKLQmlGObPVOF0hOx4qSwKxwx8rOj2KNL0
9utsFYmfVgCqy71PVsQMVSRwGuPq1hVsCUB5PaVYObyX7oI
Bv1vCsrMDTc0jBpvKFYgKH/mR1HWql+UMv2Zky8sW94QiJn
QbOvmEnLqP7IVo3qVrazo4gn5ag0wRN3Obh0vI1U6zb+OPP
SCsI2+sZD2ml0RbDChnbDJ0QUX5JlBw+YQMFqdhqKzhxtQe
hDx6DlLflShDS/vVE1Wok6I60rhnM9Kp9GE3PYEH5dbCMF0
Kg3+RWhlvm94PMnib54GqjrQfWFiUrchl3hKIbjaJm
12 Upgrade-Insecure-Requests: 1
13 Priority: u=1
14
```

可以看到就进来了：



猜测是JESSONID在的时候不会反序列化rememberMe。

源码找找原因。

看下这个stack：

```
convertBytesToPrincipals:431, AbstractRememberMeManager
(org.apache.shiro.mgt)getRememberedPrincipals:396, AbstractRememberMeManager
(org.apache.shiro.mgt)getRememberedIdentity:604, DefaultSecurityManager
(org.apache.shiro.mgt)resolvePrincipals:492, DefaultSecurityManager
(org.apache.shiro.mgt)createSubject:342, DefaultSecurityManager
(org.apache.shiro.mgt)buildSubject:846, Subject$Builder
(org.apache.shiro.subject)buildWebSubject:148, WebSubject$Builder
(org.apache.shiro.web.subject)createSubject:292, AbstractShiroFilter
(org.apache.shiro.web.servlet)doFilterInternal:359, AbstractShiroFilter
(org.apache.shiro.web.servlet)doFilter:125, OncePerRequestFilter
(org.apache.shiro.web.servlet)internalDoFilter:168, ApplicationFilterChain
(org.apache.catalina.core)doFilter:144, ApplicationFilterChain
(org.apache.catalina.core)invoke:168, StandardWrapperValve
(org.apache.catalina.core)invoke:90, StandardContextValve
(org.apache.catalina.core)invoke:482, AuthenticatorBase
(org.apache.catalina.authenticator)invoke:130, StandardHostValve
(org.apache.catalina.core)invoke:93, ErrorReportValve
(org.apache.catalina.valves)invoke:660, AbstractAccessLogValve
(org.apache.catalina.valves)invoke:74, StandardEngineValve
(org.apache.catalina.core)service:346, CoyoteAdapter
(org.apache.catalina.connector)service:383, Http11Processor
(org.apache.coyote.http11)process:63, AbstractProcessorLight
(org.apache.coyote)process:937, AbstractProtocol$ConnectionHandler
(org.apache.coyote)doRun:1791, NioEndpoint$SocketProcessor
(org.apache.tomcat.util.net)run:52, SocketProcessorBase
(org.apache.tomcat.util.net)runWorker:1190, ThreadPoolExecutor
(org.apache.tomcat.util.threads)run:659, ThreadPoolExecutor$Worker
(org.apache.tomcat.util.threads)run:63, TaskThread$WrappingRunnable
(org.apache.tomcat.util.threads)run:745, Thread (java.lang)
```

emmm，没找着，嘛，无关紧要)