目录

一、题目

1、源码

```php
class LoginManager {
    private $em;
    private $user;
    private $password;

    public function __construct($user, $password) {
        $this->em = DoctrineManager::getEntityManager();
        $this->user = $user;
        $this->password = $password;
    }

    public function isValid() {
        $user = $this->sanitizeInput($this->user);
        $pass = $this->sanitizeInput($this->password);

        $queryBuilder = $this->em->createQueryBuilder()
            ->select("COUNT(p)")
            ->from("User", "u")
            ->where("user = '$user' AND password = '$pass'");
        $query = $queryBuilder->getQuery();
        return boolval($query->getSingleScalarResult());
    }

    public function sanitizeInput($input, $length = 20) {
        $input = addslashes($input);
        if (strlen($input) > $length) {
            $input = substr($input, 0, $length);
        }
        return $input;
    }
}

$auth = new LoginManager($_POST['user'], $_POST['passwd']);
if (!$auth->isValid()) {
    exit;
}
```

2、知识点

知识点	说明
addslashes()	返回在预定义的字符（' " \ NUL）前添加反斜杠的字符串

3、解读

1）第33行，实例化LoginManager()对象，通过POST方法接收user和passwd的值作为对象参数，赋值给$auth。

2）第1行，实例化对象LoginManager()时，会执行构造器__construct()，初始化了两个参数为变量user和password。

3）第34行，对象创建成功后，调用函数isValid()对$auth进行判断。

4）第12行，通过调用函数sanitizeInput()对对象中的变量user和password进行过滤。

5）第24行，将变量传入函数sanitizeInput()，通过函数addslashes()进行过滤。如果传入的值长度大于20，就去前20位，否则就正常返回。

4、分析

1）这里使用了函数addslashes()对POST中user和passwd的特殊字符进行转义。也就代表着不能够闭合引号进行注入了。

2）但是后面使用了函数substr()进行截断处理，只取前20位。那么这时候在第20个字符处放一个会被转义的字符，让其触发转义字符\，让SQL语句变成这样：

```
select count(p) from user u where user = '1234567890123456789\' AND password = '$pass'
```

3）那么此时在passwd中传入or 1=1#，SQL语句就会变成这样：

```
select count(p) from user u where user = '1234567890123456789\' AND password = 'or 1=1#'
```

4）经过转义和注释掉的引号被消除，变成了这样：

```
select count(p) from user u where user = '1234567890123456789 AND password = 'or 1=1#
```

5）此时就成功绕过SQL注入的防御，顺利通过验证。

5、利用

```
user=1234567890123456789'
passwd=or 1=1#
```

二、CMS

1、源码-苹果CMS视频分享程序 8.0

```
inc\common\template.php
```

```php
if (!empty($lp['wd'])){
    $where .= ' AND ( instr(a_name,\''.$lp['wd'].'\')>0
    or instr(a_subname,\''.$lp['wd'].'\')>0 ) ';
}
```

```
inc\module\vod.php
```

```php
elseif($method=='search')
{
    $tpl->C["siteaid"] = 15;
    $wd = trim(be("all", "wd")); $wd = chkSql($wd);
    if(!empty($wd)){ $tpl->P["wd"] = $wd; }
```

```
inc\common\function.php
```

```php
function be($mode,$key,$sp=',')
{
    ini_set("magic_quotes_runtime", 0);
    $magicq= get_magic_quotes_gpc();
    switch($mode)
    {
        case 'post':
            $res=isset($_POST[$key]) ? $magicq?$_POST[$key]
            :@addslashes($_POST[$key]) : '';
            break;
        case 'get':
            $res=isset($_GET[$key]) ? $magicq?$_GET[$key]
            :@addslashes($_GET[$key]) : '';
            break;
        case 'arr':
            $arr =isset($_POST[$key]) ? $_POST[$key] : '';
            if($arr==""){
                $value="0";
            }
            else{
                for($i=0;$i<count($arr);$i++){
                    $res=implode($sp,$arr);
                }
            }
            break;
        default:
            $res=isset($_REQUEST[$key]) ? $magicq ? $_REQUEST[$key]
              : @addslashes($_REQUEST[$key]) : '';
            break;
    }
    return $res;
}
```

inc\common\360_safe3.php

```php
function chkSql($s)
{
    global $getfilter;
    if(empty($s)){
        return "";
    }
    $d=$s;
    while(true){
        $s = urldecode($d);
        if($s==$d){
            break;
        }
        $d = $s;
    }
    StopAttack(1,$s,$getfilter);
    return htmlEncode($s);
}
```

```php
function StopAttack($StrFiltKey,$StrFiltValue,$ArrFiltReq)
{
    $errmsg = "<div style=\"position:fixed;top:0px;width:100%;
    height:100%;background-color:white;color:green;font-weight:
    bold;border-bottom:5px solid #999;\"><br>您的提交带有不合法参数,
    谢谢合作!<br>操作IP: ".$_SERVER["REMOTE_ADDR"]."<br>操作时间: "
    .strftime("%Y-%m-%d %H:%M:%S")."<br>操作页面:".
    $_SERVER["PHP_SELF"]."<br>提交方式: "
    .$_SERVER["REQUEST_METHOD"]."</div>";
    $StrFiltValue=arr_foreach($StrFiltValue);
    $StrFiltValue=urldecode($StrFiltValue);

    if(preg_match("/".$ArrFiltReq."/is",$StrFiltValue)==1){
        print $errmsg;
        exit();
    }
    if(preg_match("/".$ArrFiltReq."/is",$StrFiltKey)==1){
        print $errmsg;
        exit();
    }
}
```

```php
//get拦截规则
$getfilter = "\\<.+javascript:window\\[.{1}\\\\x|<.*=(&#\\d+?;?)+?>|<.*(data|src)=data:text\\/html.*>|\\b(alert\\
(|confirm\\(|expression\\(|prompt\\(|benchmark\s*?\(.*\)|sleep\s*?\(.*\)|\\b(group_)?concat[\\s\\/\\*]*?\\
([^\\)]+?\\)|\bcase[\s\/\*]*?when[\s\/\*]*?\([^\)]+?\)|load_file\s*?\\()|<[a-z]+?\\b[^>]*?\\bon([a-z]{4,})\s*?
=|^\\+\\/v(8|9)|\\b(and|or)\\b\\s*?([\\(\\)'\"\\d]+?=[\\(\\)'\"\\d]+?|[\\(\\)'\"a-zA-Z]+?=[\\(\\)'\"a-zA-Z]+?|>|
<|\s+?[\\w]+?\\s+?\\bin\\b\\s*?\(|\\blike\\b\\s+?[\"']|\\/\\*.*\\*\\/|<\\s*script\\b|\\bEXEC\\b|UNION.+?
SELECT\s*(\(.+\)\s*|@{1,2}.+?\s*|\s+?.+?|(`|'|\").*?(`|'|\")\s*)|UPDATE\s*(\(.+\)\s*|@{1,2}.+?\s*|\s+?.+?|
(`|'|\").*?(`|'|\")\s*)SET|INSERT\\s+INTO.+?VALUES|(SELECT|DELETE)@{0,2}(\\(.+\\)|\\s+?.+?\\s+?|(`|'|\").*?
(`|'|\"))FROM(\\(.+\\)|\\s+?.+?|(`|'|\").*?(`|'|\"))|(CREATE|ALTER|DROP|TRUNCATE)\\s+(TABLE|DATABASE)";
//post拦截规则
$postfilter = "<.*=(&#\\d+?;?)+?>|<.*data=data:text\\/html.*>|\\b(alert\\(|confirm\\(|expression\\(|prompt\\
(|benchmark\s*?\(.*\)|sleep\s*?\(.*\)|\\b(group_)?concat[\\s\\/\\*]*?\\([^\\)]+?\\)|\bcase[\s\/\*]*?
when[\s\/\*]*?\([^\)]+?\)|load_file\s*?\\()|<[^>]*?
\\b(onerror|onmousemove|onload|onclick|onmouseover)\\b|\\b(and|or)\\b\\s*?([\\(\\)'\"\\d]+?=[\\(\\)'\"\\d]+?|[\\
(\\)'\"a-zA-Z]+?=[\\(\\)'\"a-zA-Z]+?|>|<|\s+?[\\w]+?\\s+?\\bin\\b\\s*?\(|\\blike\\b\\s+?[\"'])|\\/\\*.*\\*\\/|
<\\s*script\\b|\\bEXEC\\b|UNION.+?SELECT\s*(\(.+\)\s*|@{1,2}.+?\s*|\s+?.+?|(`|'|\").*?(`|'|\")\s*)|UPDATE\s*(\
(.+\)\s*|@{1,2}.+?\s*|\s+?.+?|(`|'|\").*?(`|'|\")\s*)SET|INSERT\\s+INTO.+?VALUES|(SELECT|DELETE)(\\
(.+\)|\\s+?.+?\\s+?|(`|'|\").*?(`|'|\"))FROM(\\(.+\\)|\\s+?.+?|(`|'|\").*?(`|'|\"))|
(CREATE|ALTER|DROP|TRUNCATE)\\s+(TABLE|DATABASE)";
//cookie拦截规则
$cookiefilter = "benchmark\s*?\(.*\)|sleep\s*?\(.*\)|load_file\s*?\\(|\\b(and|or)\\b\\s*?([\\(\\)'\"\\d]+?=[\\
(\\)'\"\\d]+?|[\\(\\)'\"a-zA-Z]+?=[\\(\\)'\"a-zA-Z]+?|>|<|\s+?[\\w]+?\\s+?\\bin\\b\\s*?\(|\\blike\\b\\s+?
[\"'])|\\/\\*.*\\*\\/|<\\s*script\\b|\\bEXEC\\b|UNION.+?SELECT\s*(\(.+\)\s*|@{1,2}.+?\s*|\s+?.+?|(`|'|\").*?
(`|'|\")\s*)|UPDATE\s*(\(.+\)\s*|@{1,2}.+?\s*|\s+?.+?|(`|'|\").*?(`|'|\")\s*)SET|INSERT\\s+INTO.+?VALUES|
(SELECT|DELETE)@{0,2}(\\(.+\\)|\\s+?.+?\\s+?|(`|'|\").*?(`|'|\"))FROM(\\(.+\\)|\\s+?.+?|(`|'|\").*?(`|'|\"))|
(CREATE|ALTER|DROP|TRUNCATE)\\s+(TABLE|DATABASE)";
```

inc\common\function.php

```php
function htmlEncode($str)
{
    if (!isN($str)){
        $str = str_replace(chr(38), "&#38;",$str);
        $str = str_replace(">", "&gt;",$str);
        $str = str_replace("<", "&lt;",$str);
        $str = str_replace(chr(39), "&#39;",$str);
        $str = str_replace(chr(32), " ",$str);
        $str = str_replace(chr(34), "&quot;",$str);
        $str = str_replace(chr(9), "    ",$str);
        $str = str_replace(chr(13), "<br />",$str);
        $str = str_replace(chr(10), "<br />",$str);
    }
    return $str;
}
```

inc\common\template.php

```php
if (!empty($lp['wd'])){
    $where .= ' AND ( instr(a_name,\''.$lp['wd'].'\')>0
    or instr(a_subname,\''.$lp['wd'].'\')>0 ) ';
}
```

```php
case 'vod':
    ...
    }

   if(!empty($this->P["order"])){ $lp['order'] = $this->P["order"];
      $this->P["auto"] = true; }
   if(!empty($this->P["by"])){ $lp['by'] = $this->P["by"];
      $this->P["auto"] = true; }
   if(!empty($lp['pagesize'])){

   ...

       if(!empty($this->P["wd"])){ $lp['wd']  = $this->P["wd"];
          $this->P["auto"] = true; }
   ...
    }
```

2、知识点

知识点	说明
get_magic_quotes_gpc()	递归去除转义字符（PHP 4 5 7）

3、解读

1）图1，漏洞位置在此。

2）图2，跟进变量wd到vod.php文件中，第1行，当$method=search时，进入第4行，使用函数be()对"all" "wd"进行处理，两头去空后赋值给$wd，然后调用函数chkSql()再对$wd处理一次。那么这里涉及了两个函数，一个个跟进。

3）图3，跟进第一个函数be()到function.php中，对第一个参数进行了switch case判断，传入的是"all"，这里没有匹配项，就会默认进入default中。也就是通过REQUEST方法接收第二个参数wd，并使用函数addslashes()进行处理。

4）图4，跟进第二个函数chkSql()到360_safe3.php中，可以看到这里对传入的实参$wd进行了while循环urldecode（URL解码）。然后在第15行调用了函数StopAttack()，以及16行的函数htmlEncode()，这里也是涉及了两个函数，一个个跟进。

5）图5，跟进第一个函数StopAttack()，同样在360_safe3.php中，使用了函数preg_match()对传入的实参进行了处理。根据参数匹配，第13行的preg_match变成：preg_match("/".$getfilter."/is",$s)；第17行的preg_match变成：preg_match("/".$getfilter."/is",1)

6）图6，跟进变量$getfilter，同样在360_safe3.php中，这些函数主要用于检测GET、POST、COOKIE中的恶意数据。

7）图7，跟进第二个函数htmlEncode()到function.php中，可以看到这里针对关键字符（&、'、空格、"、TAB、回车、换行、大于小于行等符号）进行了实体编码。

8）图8，继续回到template.php漏洞触发点，这里看一下$lp['wd']的获取方式。

9）图9，同样在template.php中，第13行，当P['wd']不为空时，$lp['wd']从P['wd']中获取数据。

4、分析

1）图1中，SQL语句是拼接的字符型变量，需要单引号闭合。

2）图7中，对关键字符进行了实体编码，没有过滤引号；但是在函数htmlEncode()又对但单引号进行了处理。

3）图4中，对数据进行了循环url解码，所以可以考虑双层URL编码进行绕过。

5、利用

网上给了Payload但没能理解，基础还是差了点，先放Payload吧

```
wd=))||if((select%0b(select(m_name)``from(mac_manager))regexp(0x5e61)),(`sleep`(3)),0)#%25%35%63
```

6、修复方案

新增对反斜杠的处理，ascii码位92。

7、参考链接

https://github.com/hongriSec/PHP-Audit-Labs/blob/master/Part1/Day13/files/README.md