# GetShell 的姿势总结

## 0x00 什么是 WebShell

渗透测试工作的一个阶段性目标就是获取目标服务器的操作控制权限，于是 WebShell 便应运而生。Webshell 中的 WEB 就是 web 服务，shell 就是管理攻击者与操作系统之间的交互。

Webshell 被称为攻击者通过 Web 服务器端口对 Web 服务器有一定的操作权限，而 webshell 常以网页脚本的形式出现。常见的 WebShell 使用 asp、jsp 和 php 来编写，提供了如执行系统命令、文件上传下载、数据库管理等功能。

## 0x01 获取 WebShell 的方式

获取 WebShell 的动作又叫做 GetShell，是渗透测试各项能力的综合体现，也是渗透测试一个重要的阶段性目标。

GetShell 方式众多，常见如文件上传、SQL 注入、命令执行、文件包含、解析漏洞等等。有时候一个漏洞即可 GetShell，有时候则需要各种漏洞打一套组合拳方可。So，多交流，才能掌握更多 GetShell 骚姿势。

### 1. 文件上传漏洞 GetShell

通过利用任意文件上传漏洞可以最快获取 WebShell，一般常见有三种情况：（1）直接上传木马文件到目标服务器；（2）绕过防护（以下不包括绕过 WAF 防护，以后有时间再介绍绕过 WAF 的姿势）限制上传木马文件；（3）CMS 等的通用任意文件上传漏洞。在对目标进行渗透测试时，可从前后台头像修改、文件导入、图片上传等处寻找文件上传接口。

此外，还需要根据识别的站点指纹寻找是否存在文件上传漏洞。以下是针对不同情况下上传 WebShell 的方式。

**（1） 站点没有任何防护，且上传点未做安全校验，则可直接上传 WebShell 文件。**

**（2） 站点存在简单防护：**

- 前端校验文件后缀时，可先传允许的文件类型，然后抓包修改文件后缀。

- MIME 校验时，抓包修改 Conten-Type 为允许 MIME 类型。

### （3） 绕过黑名单的方式：

- 利用特殊文件后缀。
  如. php3、.php5、.php7、.phtml；asa、cer、cdx、aspx；jspx、 jsw、jsv、jspf 等，但不一定都能被解析。

- 配合 Windows/Linux 特性使用特殊字符，如上传. php::$DATA、“.php 空格”等后缀的文件到 Windows 服务器，由于 Windows 的文件后缀中不能包含一些特殊符号，使得这些文件在保存在 Windows 服务器上时后缀只剩下. php。

- Apache 1.x、2.2.x 版本文件解析漏洞，.php.xx。

- 后缀大小写，如 pHp。

- 在数据包中使用双 filename，如 filename=“1.jsp”; filename=“1.php”。

### （4）绕过白名单：

- 00 截断，要求 PHP<5.3.4 且 magic_quotes_gpc 为 OFF 状态。

- 配合解析漏洞（见解析漏洞 getshell）。

- 文件包含图片马（见文件包含 getshell）。

### （5）绕过文件内容检测：

- **使用文件头绕过，如图片中使用 GIF89a。**

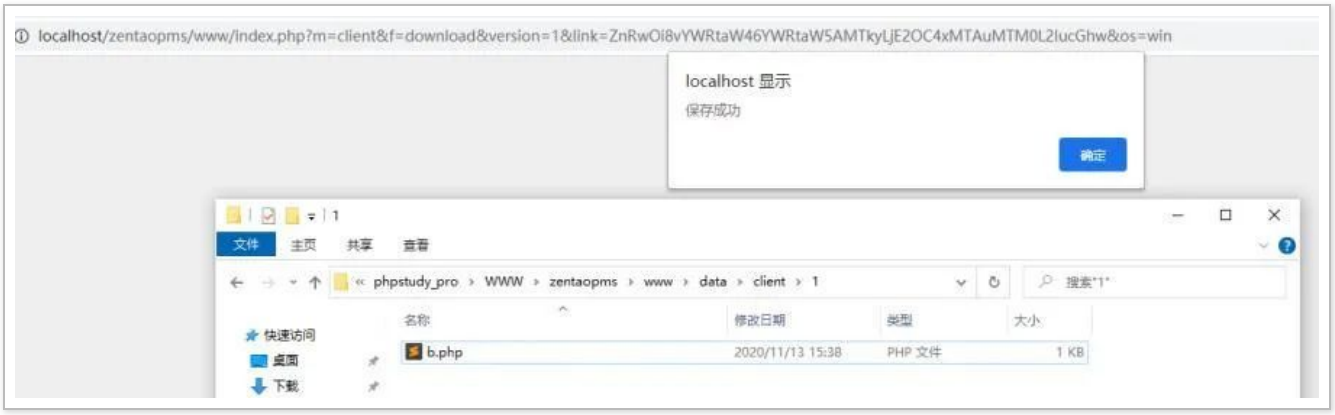
- **使用图片马，接着配合解析漏洞或者文件包含漏洞 getshell。**

- **使用元数据编辑器在图片的 EXIF 信息中插入一句话木马。**

### （6）CMS、框架等的文件上传漏洞。

如禅道 <=12.4.2 后台任意文件上传漏洞；编辑器漏洞、中间件的也不容忽视，如 CVE-2017-12615 可直接使用 put 写入木马文件。当然，“老洞”IIS6 开启 WebDAV 可直接 put 写入文件。

以下两张图是禅道 <=12.4.2 后台任意文件上传漏洞的利用截图：





## 2. 命令（代码）执行漏洞 GetShell

利用命令（代码）执行漏洞写入 WebShell 文件从而 GetShell。在此列举了以下四种常见情况作为说明。

（1）利用 CMS、框架通用漏洞，如 thinkPHP 命令执行漏洞，影响范围较广，如鲶鱼 CMS、ThinkCMF、yunCMS 等。Struts2 远程命令（代码）执行漏洞，如 S2–059、S2–057……

（2）中间件、架构通用漏洞，如：Jboss 远程命令执行、weblogic 未授权命令执行 CVE–2020–14882……

（3）应用程序命令执行漏洞，如 Redis4.x/5.x 命令执行漏洞、Zabbix 远程命令执行 CVE–2020–11800……

（4）命令注入漏洞写入 WebShell，根据 Linux 和 Windows 的不同采用不同的注入方式。如：


## 3. 解析漏洞 GetShell

利用解析漏洞将图片马等文件解析为恶意脚本文件从而 GetShell，本文主要介绍在 IIS、Nginx、Apache 的解析漏洞。一般，该漏洞会配合文件上传漏洞等来获取 WebShell。

### （1）IIS 解析漏洞。

- IIS 5.x/6.0 解析漏洞，其中文件名解析漏洞利用形式为 *.asp;.jpg；目录解析漏洞利用形式为 / asp/1.jpg。
- IIS 7.0/7.5 解析漏洞。其实该漏洞为 cgi.fix_pathinfo 的配置不当引起的解析漏洞，利用形式为 x.jpg%20\0.php。

### （2） Nginx 解析漏洞

- **Nginx 漏洞版本 0.8.41 ~ 1.4.3 / 1.5.0 ~ 1.5.7 默认配置导致解析漏洞，利用形式为 x.jpg/.php。**
- **Nginx 漏洞版本 0.5.、0.6.、0.7 <= 0.7.65、0.8 <= 0.8.37，利用形式为 x.jpg%00.php。**
- CVE-2013-4547，利用形式为 x.jpg[非空编码]\0.php
- PHPStudy v8.1.0.7 默认 Nginx 配置解析漏洞，利用形式为 x.jpg/.php。

### （3） Apache 解析漏洞

- 文件后缀 x.php.ss.ss2 会被解析为 php 文件
- 利用配置文件。上传. htaccess 配置文件，就会将. jpg 解析为 PHP 文件，内容如下：

```
AddType application/x-httpd-php xxx
```
SecIN@WiHat

或者

```
<FilesMatch "shell.jpg">
SetHandler application/x-httpd-php
</FilesMatch>
```
SecIN@WiHat

## 4.SQL 注入漏洞 GetShell

在 MySQL 注入中，可以利用 SQL 注入获取 WebShell，要写入 WebShell 文件到服务器中，需要满足以下条件：

1. 网站物理路径；
2. 文件写入的权限；
3. secure_file_priv 条件没有设置为 NULL；
   要求 mysql 数据库的配置中，没有将 secure_file_priv 条件没有设置为 NULL，即 secure_file_priv=NULL 时，无法导入导出文件；而当设置为空时，即 secure_file_priv = 时，则导入导出文件不受限制；如果设置为某个文件路径，如 secure_file_priv=/mysql / 时，则导入导出必须要在跟文件目录下完成。利用形式如下。

(1) 直接使用 into outfile

```
select '<?php phpinfo();?>' into outfile '/var/www/html/shell.php'

```
SecIN@WiHat

(2) 直接使用 into downfile

```
select '<?php phpinfo();?>' into downfile '/var/www/html/shell.php'

```
SecIN@WiHat

(3) 意为将数据导出到文件 shell.php 时，每个数据以一句话木马作为分割符

```
into outfile '/var/www/html/shell.php' lines terminated by '<?php phpinfo();?>'
2
3
                                                                    SecIN@WiHat
```

而在写入时，也可以将分隔符，即一句话木马做 hex 编码如：

```
1  union select 1,2,3 into outfile 'C:/2.php' fields terminated by 0x3c3f706870207068706696e666f28293b203f3e%23
2
3
                                                                    SecIN@WiHat
```

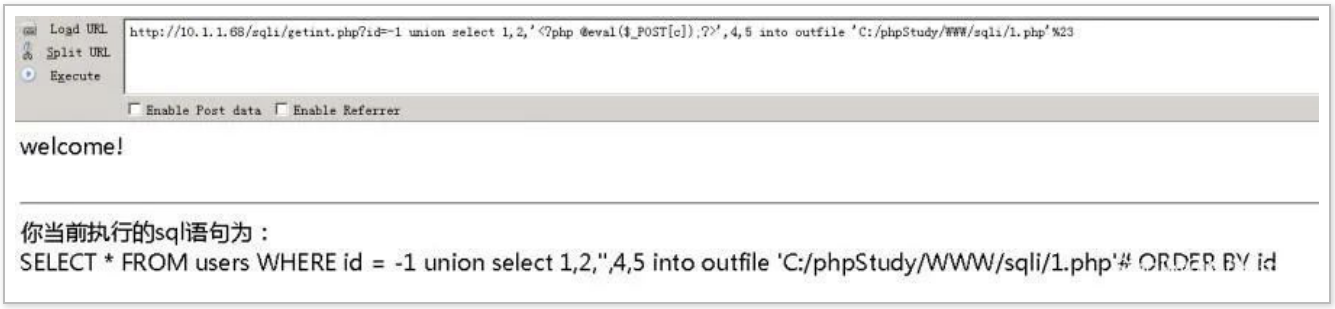
（4） 设置每行数据开头为一句话木马

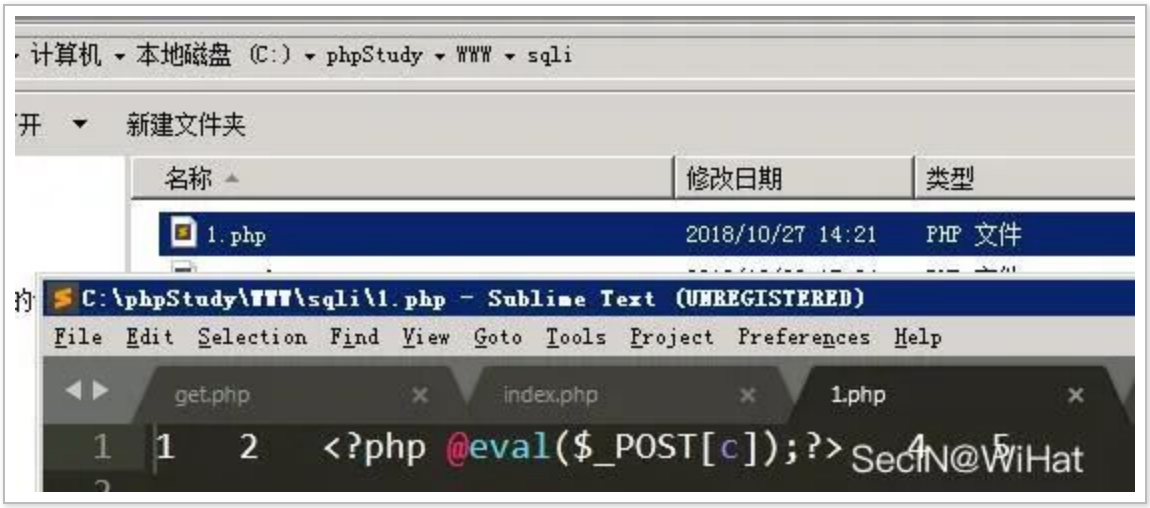
```
1  into outfile '/var/www/html/shell.php' lines starting by '<?php phpinfo();?>'
2
3
                                                                    SecIN@WiHat
```

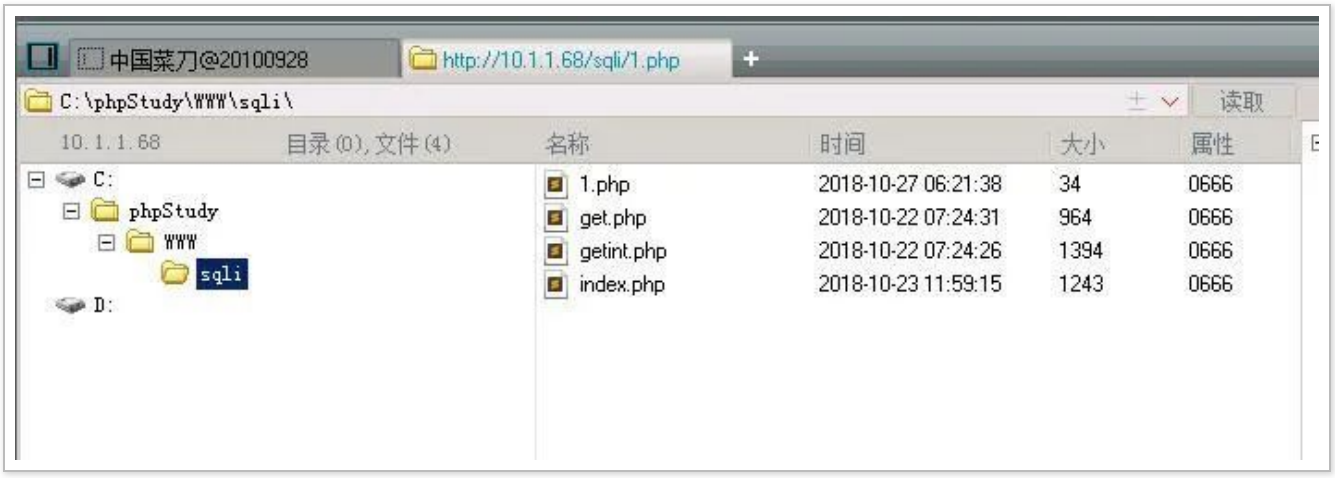
（5） 在字段之间的以一句话木马作为分隔符。

```
1  into outfile '/var/www/html/shell.php' fields terminated by '<?php phpinfo();?>'
2
3
                                                                    SecIN@WiHat
```

举例如下图所示：






## 5. 文件包含漏洞 GetShell

利用文件包含漏洞 GetShell，由于面比较大，主要介绍 PHP 文件包含，其分为本地文件包含和远程文件包含。本地文件包含中，需要 PHP 的配置文件项 allow_url_include 和 allow_url_fopen 均设置为 On。一般配合文件上传等漏洞 GetShell，最常见的如先传图片马，再包含之。

远程文件包含中，需要 PHP 的配置文件项 allow_url_fopen 设置为 On，利用方式是在远程服务器上存在 shell.txt，内容为 webshell，然后远程包含这个文件即可。笔者在文章 https://sec-in.com/article/80 中已经对文件包含漏洞做了详细的介绍，读者可前往阅读。

## 6. 其他方式 GetShell

将其命名为“其他方式”是因为不便将其再分类，笔者便都将其归纳至此。

### （1） 0day、nday GetShell

0day 自己挖，如一些 cms 黑白盒渗透、代码审计等；Nday 靠收集。

### （2） 后台创建模板写入 WebShell

创建、修改模板时写入 webshell 文件。

### （3） 修改网站配置插入 WebShell

在网站设置如“网站名称”中插入一句话，但需要注意闭合，稍有不慎会将网站插坏的。

### （4） 后台数据库备份 GetShell

数据库备份时修改备份文件后缀，在数据中插入一句话。

### （5） PHPMyadmin GetShell

- **开启日志文件，将日志文件的路径设置为网站的物理路径，如：**

```
SET GLOBAL general_log_file='C:/phpStudy/www/xxx.php'
```

SecIN@WiHat

最后在执行的 SQL 语句中带有一句话木马，这样，日志文件内容中就包含了 webshell。

- 导出数据时，将 webshell 导出到文件中，详见前文 SQL 注入 GetShell。

### （6） 组合漏洞 GetShell

如 DedeCMS CSRF + 任意文件写入。

### （7） 后台修改文件上传类型

添加上传文件类型，如白名单中添加 PHP。

### （8） ……

以上。本次就先总结到这里，扛不住这眼睛乏涩，望大佬们不吝赐教。

## 0x02 写在最后

很早前就想着将 GetShell 的方式总结一下，但无奈拖延症晚期的我，愣是拖了半年多。半年的时间内，也不知道自己做了啥。对了，很希望能和各位大佬多交流。最后，漏洞永远都会存在，就像蝗虫从未被我们消灭。