### 安全规约

## nginx安全规范

### 下载位置

Nginx软件的下载，应该从Nginx官方提供的下载页面统一下载下载。Nginx官网下载地址为：<http://nginx.org/en/download.html>

需要下载最新稳定版本。注意不要下载beta版本

### 安装

Nginx安装时，不引入不必要的模块

### 配置

#### 消除目录浏览漏洞

nginx默认不允许目录浏览，请检查目录浏览的相关配置，确保没有目录浏览漏洞：

检查各个配置文件，确保autoindex的配置为off。

autoindex off

#### 开启访问日志

开启日志有助于发生安全事件后回溯分析整个事件的原因及定位攻击者

默认情况下，nginx已开启访问日志记录，请在nginx配置文件中确认已开启访问日志

access\_log /data/logs/nginx/local.yw.wiiqq.com\_access.log;

#### 目录安全配置

如果Nginx以nobody用户启动，则黑客通过网站漏洞入侵服务器后，将获得nginx的nobody权限，因此需要确认网站web目录和文件的属主与nginx启动用户不同，防止网站被黑客恶意篡改和删除。网站web目录和文件的属主可以设置为root等（非nginx启动用户）。Web目录权限统一设置为755，web文件权限统一设置为644。只有上传目录等需要可读可写权限的目录可以设置为777 。

但是，黑客可以在777权限目录中上传或者写入web木马，因此应该严格保证777权限的目录没有执行脚本的权限。安全配置如下：

1、对于使用php的业务，在nginx配置文件中添加如下配置，即可禁止log目录执行php脚本：

|  |
| --- |
| location ~\* ^/data/cgisvr/log/.\*\.(php|php5)$ {  deny all;  } |

2、对于使用非php的业务（如python、二进制cgi等），则需禁止外部访问777目录，配置用例如下：

|  |
| --- |
| Location ~ ^/log/ {  deny all;  } |

**如果是php环境，安装PHPIPS，** <http://km.oa.com/group/11796/articles/show/198451>

#### 管理目录安全配置

对于管理目录，需要做到只允许合法ip可以访问，nginx限制白名单ip访问的配置日下：

|  |
| --- |
| location ~ ^/private/ {  allow 192.168.1.0/24;  deny all;  } |

#### 删除默认页面

删除nginx默认首页index.html，业务可以自行创建默认首页代替之。

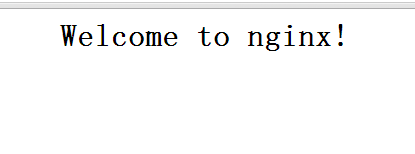
Nginx存在默认目录:



删除如下配置信息。

|  |
| --- |
| location /doc {  root /usr/share;  autoindex on;  allow 127.0.0.1;  deny all;  }  location /images {  root /usr/share;  autoindex off;  } |

删除首页index.html后，新建其他首页内容不允许出现如下首页内容：



## php安全规范

### 下载位置

PHP程序的下载，应该从PHP官方提供的下载页面下载：

PHP官网下载地址为：<http://www.php.net/downloads.php>

注意不要下载和使用beta版本。

### 安装

php安装时，建议不引入不必要的模块。

### 安全配置

#### 控制脚本访问权限

PHP默认配置允许php脚本程序访问服务器上的任意文件，为避免php脚本访问不该访问的文件，从一定程度上限制了php木马的危害，需设置php只能访问网站目录或者其他必须可访问的目录。

/usr/local/apache2/web/为网站根目录，打开php.ini，安全加固配置方式如下：

|  |
| --- |
| **open\_basedir = /usr/local/apache2/web/** |

需要多个目录时，以冒号隔开如：

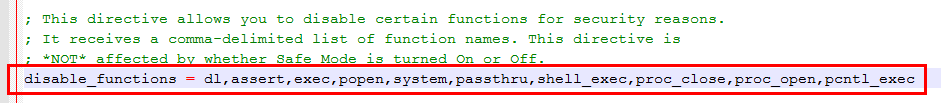
|  |
| --- |
| **open\_basedir = /usr/local/apache2/web/:/tmp/:/data/adc/** |

#### 禁止使用的PHP危险函数

Web木马程序通常利用php的特殊函数执行系统命令，查询任意目录文件，增加修改删除文件等。php木马程序常使用的函数为：dl,eval,assert,exec,popen,system,passthru,shell\_exec等。

打开php.ini，安全加固配置方式如下，禁止使用这些危险函数：

|  |
| --- |
| **disable\_functions = dl,assert,exec,popen,system,passthru,shell\_exec,proc\_close,proc\_open,pcntl\_exec** |



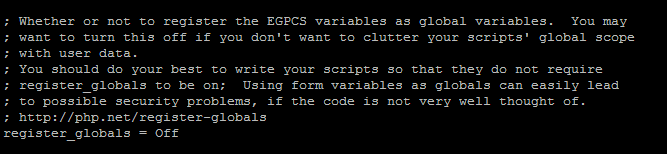
#### 关闭注册全局变量（PHP 5.3.\*与5.4.\*中已废弃）

在PHP中提交的变量，包括使用POST或者GET提交的变量，会自动注册为全局变量，能够直接访问，这是对服务器非常不安全的，所以不能让它注册为全局变量，就把注册全局变量选项关闭。

打开php.ini，安全加固配置方式如下，关闭注册全局变量设置：

|  |
| --- |
| **register\_globals = Off** |

**注意事项：**默认php配置文件该选项是关闭的。

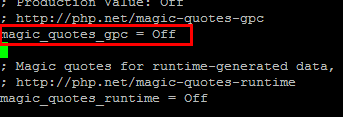


#### 开启magic\_quotes\_gpc(PHP 5.3.\*与5.4.\*中已废弃)

magic\_quotes\_gpc会把引用的数据中包含单引号'和双引号"以及反斜线 \自动加上反斜线，自动转译符号，确保数据操作的正确运行，magic\_quotes\_gpc的设定值将会影响通过Get/Post/Cookies获得的数据，可以有效的防止SQL注入漏洞。

打开php.ini，安全加固配置方式如下，打开magic\_quotes\_gpc设置：

|  |
| --- |
| **magic\_quotes\_gpc = On** |

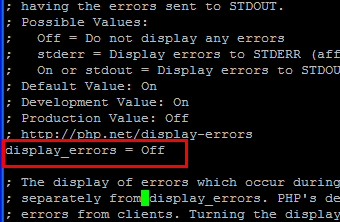


#### 关闭错误消息显示

php在没有连接到数据库或者其他情况下会有提示错误，一般错误信息中会包含php脚本当前的路径信息或者查询的SQL语句等信息，这类信息提供给黑客后，是不安全的，所以服务器建议禁止错误提示。

打开php.ini，安全加固配置方式如下，关闭错误信息显示设置：

|  |
| --- |
| **display\_errors = Off** |



#### 开启错误日志记录

在关闭display\_errors后能够把错误信息记录下来，便于查找服务器运行的原因,同时也要设置错误日志存放的目录，建议跟webserver的日志放在一起。

打开php.ini，安全加固配置方式如下，打开错误日志记录并设置错误日志存放路径：

|  |
| --- |
| **log\_errors = On**  **error\_log = /usr/local/nginx/logs/php\_error.log** |

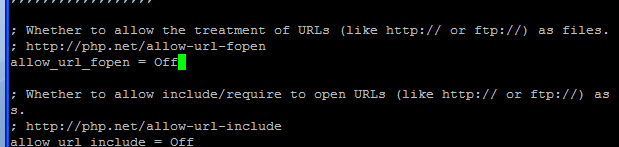
**注意事项**：该文件必须允许webserver的用户和组具有写的权限

#### 禁止访问远程文件

允许访问URL远程资源使得PHP应用程序的漏洞变得更加容易被利用，php脚本若存在远程文件包含漏洞可以让攻击者直接获取网站权限及上传web木马，一般会在php配置文件中关闭该功能，若需要访问远程服务器建议采用其他方式如libcurl库。

打开php.ini，安全加固配置方式如下，禁止访问远程文件:

|  |
| --- |
| **allow\_url\_fopen = Off**  **allow\_url\_include = Off** |

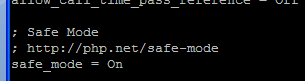


#### 开启php安全模式（PHP 5.3.\*与5.4.\*中已废弃）

php的安全模式是个非常重要的内嵌的安全机制，能够控制一些php中的函数，比如system()，同时把很多文件操作函数进行了权限控制，也不允许对某些关键文件的读取。

打开php.ini，开启safe\_mode:

|  |
| --- |
| **safe\_mode = On** |



**注意事项**：如果安全模式打开了，但是却是要执行某些程序的时候，可以指定要执行程序的主目录：safe\_mode\_exec\_dir = D:/usr/bin

#### nginx文件解析漏洞防御

如果webserver为nginx， 则须在PHP的配置文件php.ini中配置cgi.fix\_pathinfo = 0，防止nginx文件解析漏洞。

#### PHP(FastCGI)非root权限启动

如果使用FastCGI方式解析PHP，则需要非root权限运行。

1. 如果是启动php-cgi进程，则需要su切换到普通用户再启动。
2. 如果是启动php-fpm进程，默认配置已是非root运行，不能修改为root运行。

#### PHP(FastCGI)监听端口做访问控制

如果是单独启动php-cgi/php-fpm进程，监听IP也不是127.0.0.1，则需要做端口访问控制。如新增iptables规则（假设监听端口为9000，webserver来源IP为192.168.1.125）

iptables -A INPUT -i eth0 -p tcp -s 192.168.1.125 --dport 9000 -j ACCEPT

iptables -A OUTPUT -o eth0 -p tcp -d 192.168.1.125 --sport 9000 -j ACCEPT

iptables -A INPUT -i eth0 -p tcp --dport 9000 -j DROP

## mysql安全规范

下载