(1) 控制php脚本能访问的目录  
　　使用open\_basedir选项能够控制PHP脚本只能访问指定的目录，这样能够避免PHP脚本访问  
　　不应该访问的文件，一定程度上限制了phpshell的危害，我们一般可以设置为只能访问网站目录：  
　　open\_basedir = D:/usr/www

(2) 打开magic\_quotes\_gpc来防止SQL注入  
　　SQL注入是非常危险的问题，小则网站后台被入侵，重则整个服务器沦陷，  
　　所以一定要小心。php.ini中有一个设置：  
　　magic\_quotes\_gpc = Off  
这个默认是关闭的，如果它打开后将自动把用户提交对sql的查询进行转换，  
　　比如把 ' 转为 \'等，这对防止sql注射有重大作用。所以我们推荐设置为：  
　　magic\_quotes\_gpc = On

(3) 错误信息控制  
　　一般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，所以一般服务器建议禁止错误提示：  
　　display\_errors = Off  
　　如果你却是是要显示错误信息，一定要设置显示错误的级别，比如只显示警告以上的信息：  
　　error\_reporting = E\_WARNING & E\_ERROR  
　　当然，我还是建议关闭错误提示。

; error\_reporting = E\_ALL & ~E\_NOTICE ; 显示所有的错误，除了提醒  
　　; error\_reporting = E\_COMPILE\_ERROR|E\_ERROR|E\_CORE\_ERROR ; 仅显示错误  
　　error\_reporting = E\_ALL & ~E\_NOTICE ; 显示所有的错误，除了提醒  
　　display\_errors = On ; 显示出错误信息(作为输出的一部分)  
(4) 错误日志  
　　建议在关闭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：  
log\_errors = On  
　　同时也要设置错误日志存放的目录，建议根apache的日志存在一起：  
　　error\_log = D:/usr/local/apache2/logs/php\_error.log  
　　注意：给文件必须允许apache用户的和组具有写的权限。

(5) 设置PHP的扩展库路径

extension\_dir=""  
(6) 在关闭display\_errors后开启PHP错误日志（路径在php-fpm.conf中配置）

log\_errors = On  
(7) 设置PHP的opcache和mysql动态库  
 zend\_extension=opcache.so extension=mysqli.so extension=pdo\_mysql.so  
(8) 设置PHP的时区  
 date.timezone = PRC  
(9) 开启opcache

zend\_extension = opcache

opcache.enable=1

opcache.enable\_cli=1

opcache.memory\_consumption=128

opcache.max\_accelerated\_files=2000

（10）设置PHP脚本允许访问的目录（需要根据实际情况配置）  
 open\_basedir = /usr/share/nginx/html;  
（11）会话安全  
 [session.use\_only\_cookies](https://www.php.net/manual/zh/session.configuration.php#ini.session.use-only-cookies)=1 【只使用 cookie 来记录会话 ID】

[session.use\_strict\_mode](https://www.php.net/manual/zh/session.configuration.php#ini.session.use-strict-mode)=1 【启用 session.use\_strict\_mode 是必不可少】