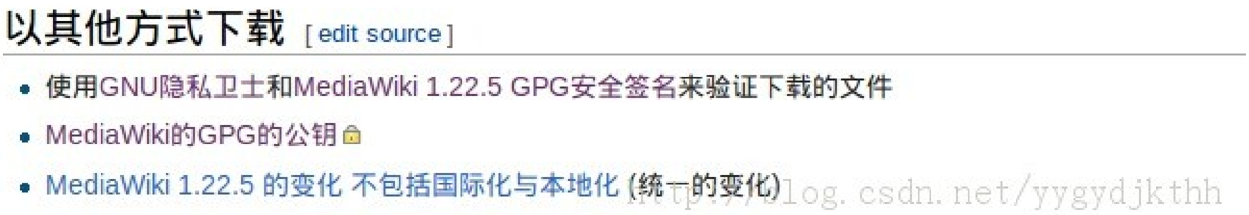
一、服务最小安装（WIKI 所必需的服务）：

```
test@debian:~/$：apt-get install mysql-dev apache2 php5 php5-mysql mysql-server
```

注意：请在进行系统时登录用户为非 root 账号，防止被直接利用 root 权限进行破坏性操作；

二、对 mediawiki 安装包进行数字签名的验证：

数字签名：保证信息传输的完整性、发送者的身份认证、防止交易中的抵赖发生；



以其他方式下载 [edit source]

- 使用GNU隐私卫士和MediaWiki 1.22.5 GPG安全签名来验证下载的文件
- MediaWiki的GPG的公钥
- MediaWiki 1.22.5 的变化 不包括国际化与本地化 (统一的变化)

以我下载的 mediawiki-1.22.5.tar.gz 包为例：

得到 MediaWiki 1.22.5 GPG 安全签名文件：mediawiki-1.22.5.tar.gz.sig

能过上图的 MediaWiki 的 GPG 的公钥 得到 pubkey 文件，并保存到文件 mediawiki_pubkey.txt 中；

导入公钥到当前系统中：

```
fuckids@debian-IDS:~/wiki$ gpg --import mediawiki_pubkey.txt
gpg: /home/fuckids/.gnupg/trustdb.gpg: trustdb created
gpg: key 7F901A30: public key "Mark A. Hershberger <mah@everybody.org>" imported
gpg: Total number processed: 1
gpg:               imported: 1  (RSA: 1)
gpg: no ultimately trusted keys found
```

//列出当前的公钥，检查 是否导入 成功

```
fuckids@debian-IDS:~/wiki$ gpg -k
/home/fuckids/.gnupg/pubring.gpg
--------------------------------
pub   2048R/7F901A30 2009-07-01 [expires: 2019-06-29]
uid                  Mark A. Hershberger <mah@everybody.org>
sub   2048R/84896BEA 2009-07-01 [expires: 2019-06-29]
```

//使用签名文件对下载的文件进行验证：

```
fuckids@debian-IDS:~/wiki$ gpg --verify mediawiki-1.22.5.tar.gz.sig mediawiki-1.22.5.tar.gz
gpg: Signature made Fri 28 Mar 2014 08:21:11 AM CST using RSA key ID 7F901A30
gpg: Good signature from "Mark A. Hershberger <mah@everybody.org>"
gpg: WARNING: This key is not certified with a trusted signature!
```

```
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 3CEF 8262 806D 3F0B 6BA1  DBDD 7956 EE47 7F90 1A30
```

其中 gpg: Good signature from "Mark A. Hershberger <mah@everybody.org>" 这一行表示签名文件是正确的，表示下载的包并没有被中间修改过；

三、mysql 安全加固：

1、执行 mysql 安全加固程序：

```
test@debian:~/$：/usr/bin/mysql_secure_installation
```

此程序主要完成以下几个功能：

（1）、为 root 用户设置密码；

（2）、删除匿名账号；

（3）、取消 root 用户远程登录；

（4）、删除 test 库和对 test 库的访问权限；

（5）、刷新授权表使修改生效；

通过这几项加固设置能够提高 mysql 库的安全性。

执行例子：（注意：其中//开头的行是人为添加的注释）

```
test@debian:~/$:~/Autosnort - Debian$ /usr/bin/mysql_secure_installation




NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MySQL
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!


In order to log into MySQL to secure it, we'll need the current
password for the root user.  If you've just installed MySQL, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MySQL
root user without the proper authorisation.
```

```
You already have a root password set, so you can safely answer 'n'.
//为 root 用户设置密码；
Change the root password? [Y/n] n
 ... skipping.

By default, a MySQL installation has an anonymous user, allowing anyone
to log into MySQL without having to have a user account created for
them.  This is intended only for testing, and to make the installation
go a bit smoother.  You should remove them before moving into a
production environment.
//删除匿名账号；
Remove anonymous users? [Y/n] y
 ... Success!

Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the
network.
//取消 root 用户远程登录；
Disallow root login remotely? [Y/n] n
 ... skipping.

By default, MySQL comes with a database named 'test' that anyone can
access.  This is also intended only for testing, and should be removed
before moving into a production environment.
//删除 test 库和对 test 库的访问权限；
Remove test database and access to it? [Y/n] y
 - Dropping test database...
ERROR 1008 (HY000) at line 1: Can't drop database 'test'; database doesn't exist
 ... Failed!  Not critical, keep moving...
 - Removing privileges on test database...
 ... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.
//刷新授权表使修改生效；
Reload privilege tables now? [Y/n] n
 ... skipping.

Cleaning up...
```

```
All done!  If you've completed all of the above steps, your MySQL
installation should now be secure.

Thanks for using MySQL!
```

2、创建wiki数据库，创建专用于wiki访问数据库的用户并分配给此用户只能本地访问wiki数据库的权限；创建专用于wiki访问数据库的原因是尽量少用root账号进行库的操作，防止权限的过大化；

进行数据库命令为：

```
mysql -u root -p
```

输入root密码后登入如下：

```
mysql>
```

（1）、创建wiki数据库；

```
mysql> create database zlsl_wiki;
```

（2）、创建专用于wiki访问的数据库用户，'wiki_user'@'localhost'表示用户名为wiki_user，且此用户只能够在本地登录， 防止被远程访问利用，登录密码为：fucksec2013；

```
mysql> CREATE USER 'wiki_user'@'localhost' IDENTIFIED BY 'fucksec2013';
Query OK, 0 rows affected (0.00 sec)
```

（3）、grant命令的格式：grant 执行动作 on 数据库.* to 用户名@登录主机 identified by "密码" ，以下完成的功能则为用户wiki_user只能本地登录且只能对wiki数据库zlsl_wiki进行所有操作；

```
mysql> GRANT ALL ON zlsl_wiki.* TO 'wiki_user'@'localhost' IDENTIFIED BY
'fucksec2013';
Query OK, 0 rows affected (0.00 sec)
mysql> exit
```

重启数据库服务：

```
/etc/init.d/mysql restart
```

四、安装mediawiki:

解压mediawiki-1.22.5.tar.gz：

```
fuckids@debian-IDS $ su
fuckids@debian-IDS $ tar zxvf mediawiki-1.22.5.tar.gz
```

移动mediawiki目录到apache web服务器目录下：

```
fuckids@debian-IDS $ mv   mediawiki-1.22.5 /var/www/mediawiki
```

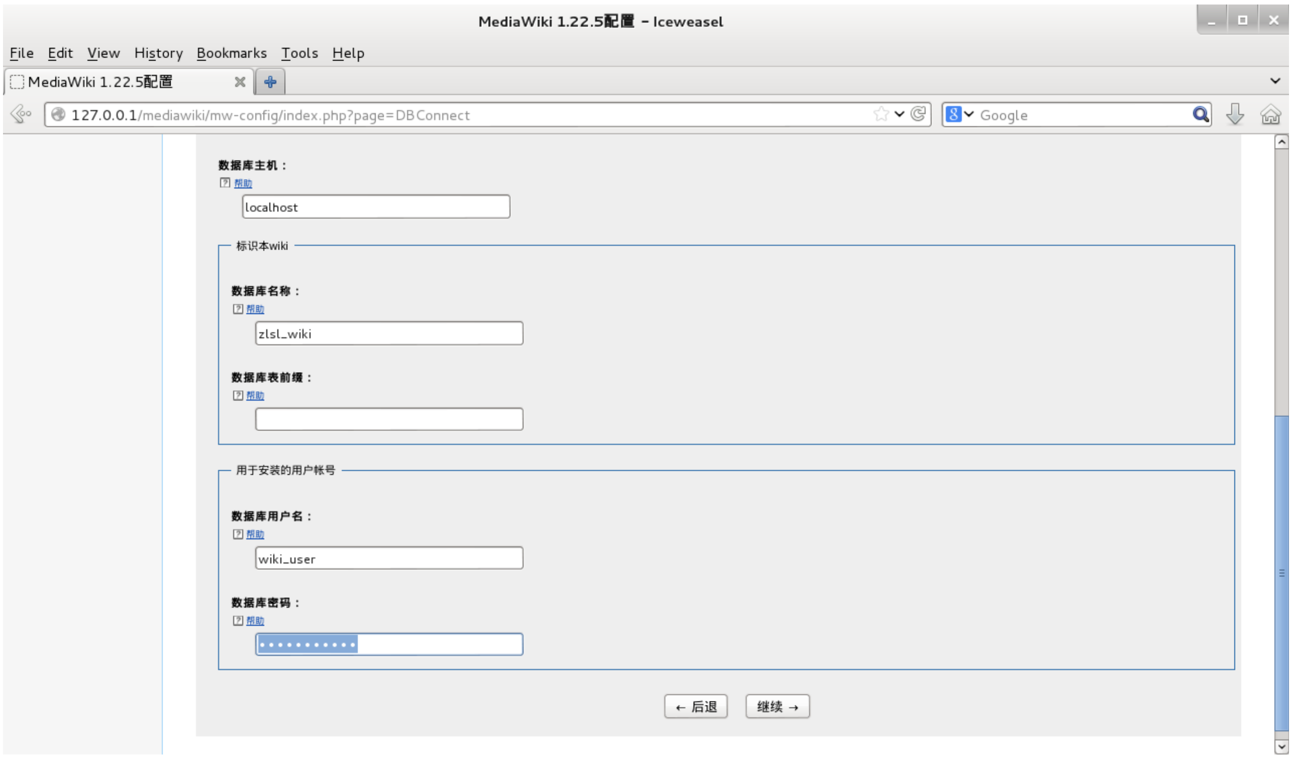
然后在浏览器中输入：

http://127.0.0.1/mediawiki/index.php

配置mediawiki的安装过程：

wiki使用的数据库名为创建的数据库的名称，用户名及密码和前面创建的用户名

密码一致；


MediaWiki 1.22.5配置 - Iceweasel
File Edit View History Bookmarks Tools Help
MediaWiki 1.22.5配置
127.0.0.1/mediawiki/mw-config/index.php?page=DBConnect
Google
帮助
← 后退
继续 →

数据库主机：
localhost

标识本wiki

数据库名称：
zlsl_wiki

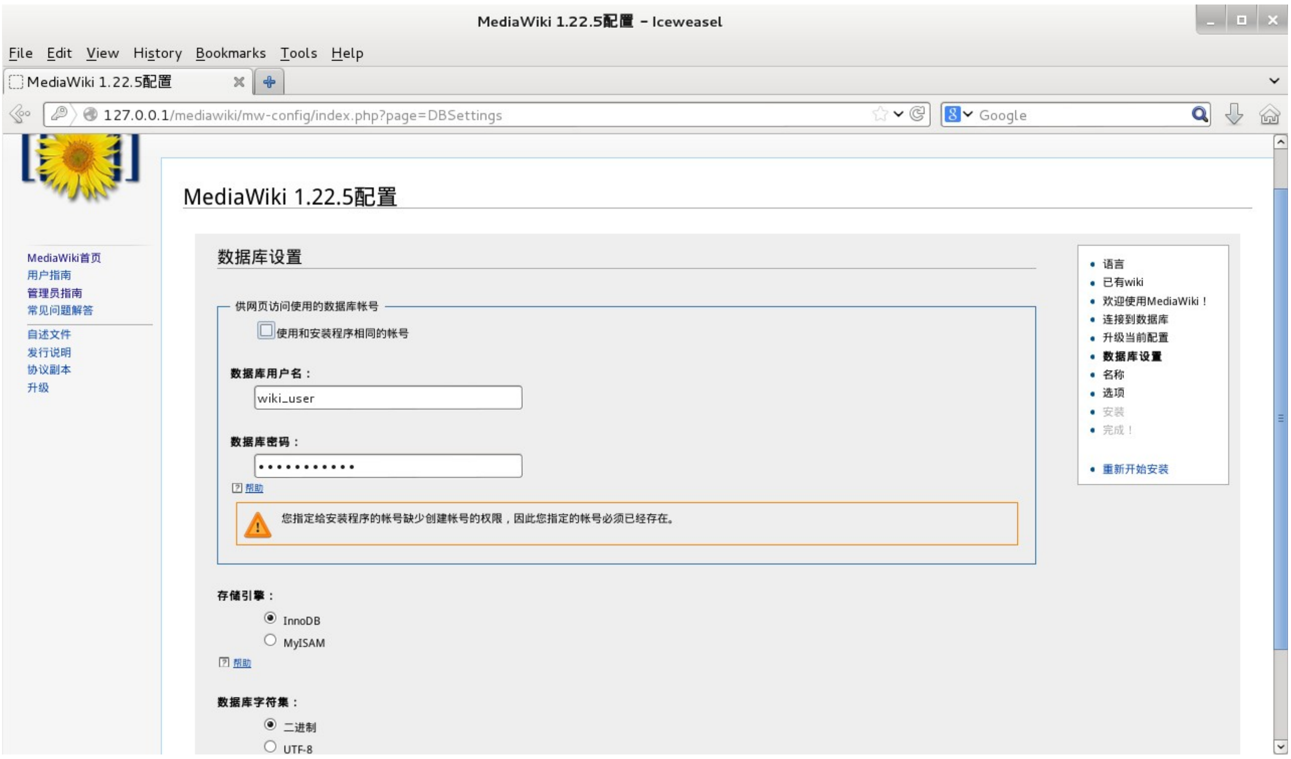
数据库表前缀：

用于安装的用户帐号

数据库用户名：
wiki_user

数据库密码：
••••••••••••

如果上一图中的用于安装的用户帐号和数据库访问的用户不一致时，在下图这个步骤中的数据库用户名和密码处填写相应的用户名/密码；



MediaWiki 1.22.5配置

数据库设置

供网页访问使用的数据库帐号

□ 使用和安装程序相同的帐号

数据库用户名：
wiki_user

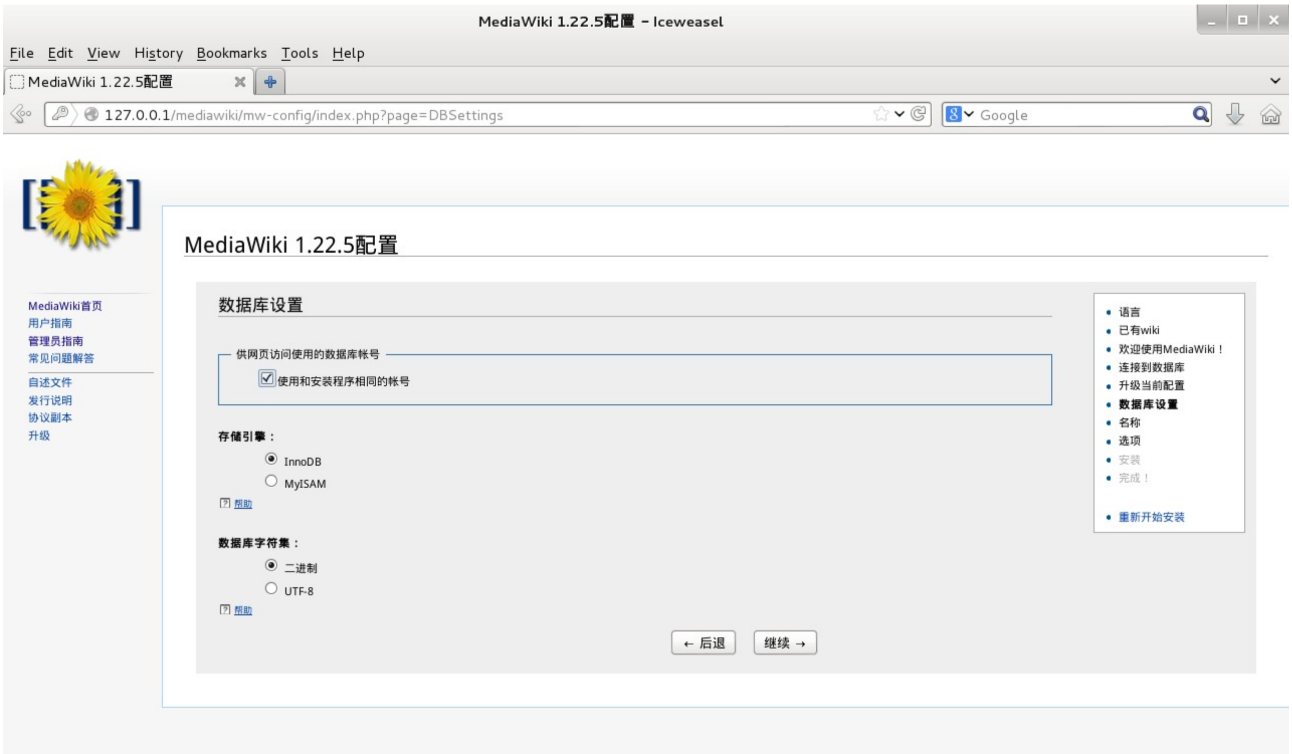
数据库密码：
•••••••••••

您指定给安装程序的帐号缺少创建帐号的权限，因此您指定的帐号必须已经存在。

存储引擎：
● InnoDB
○ MyISAM

数据库字符集：
● 二进制
○ UTF-8

如果用于安装的用户帐号和数据库访问的用户一致时，在下图中直接勾选 使用和安装程序相同的帐号；



MediaWiki 1.22.5配置

数据库设置

供网页访问使用的数据库帐号

☑ 使用和安装程序相同的帐号

存储引擎：

● InnoDB

○ MyISAM

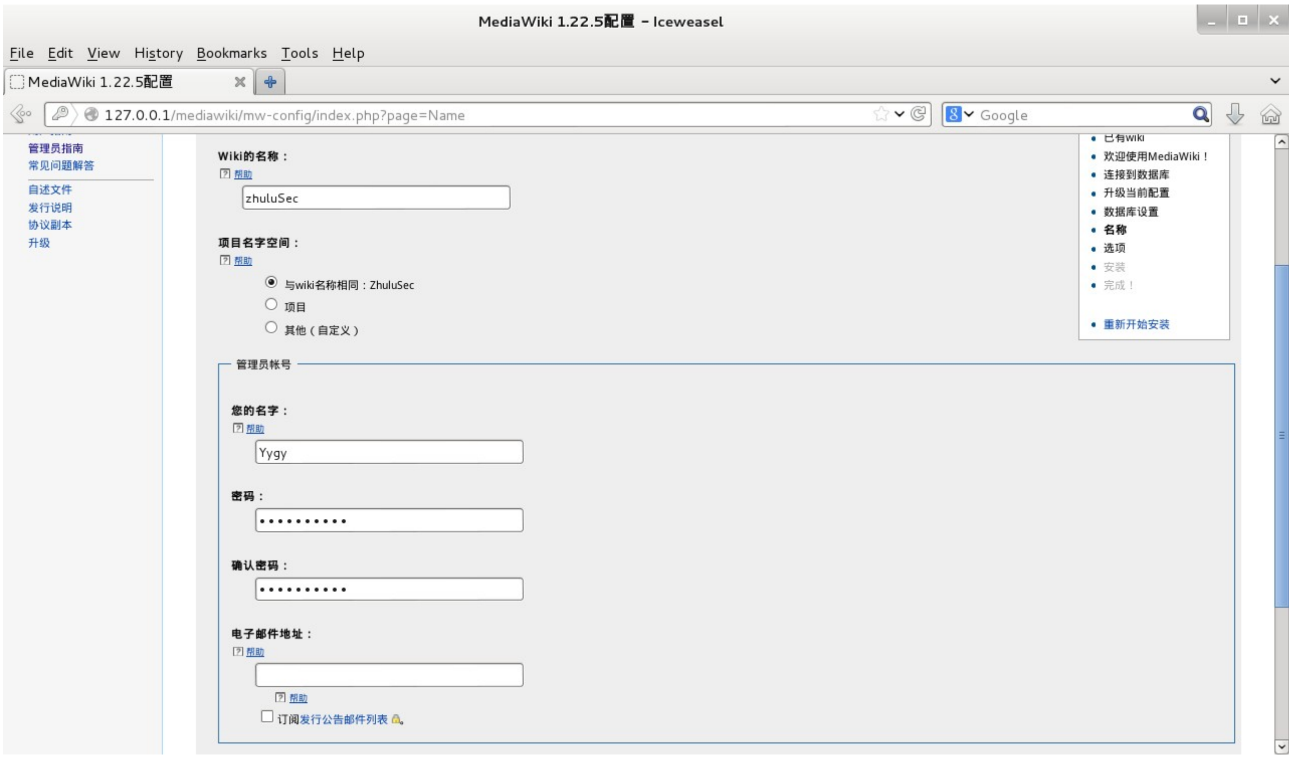
帮助

数据库字符集：

● 二进制

○ UTF-8

帮助



Wiki的名称：

帮助

zhuluSec

项目名字空间：

帮助

● 与wiki名称相同：ZhuluSec

○ 项目

○ 其他（自定义）

管理员帐号

您的名字：

帮助

Yygy

密码：

•••••••••••

确认密码：

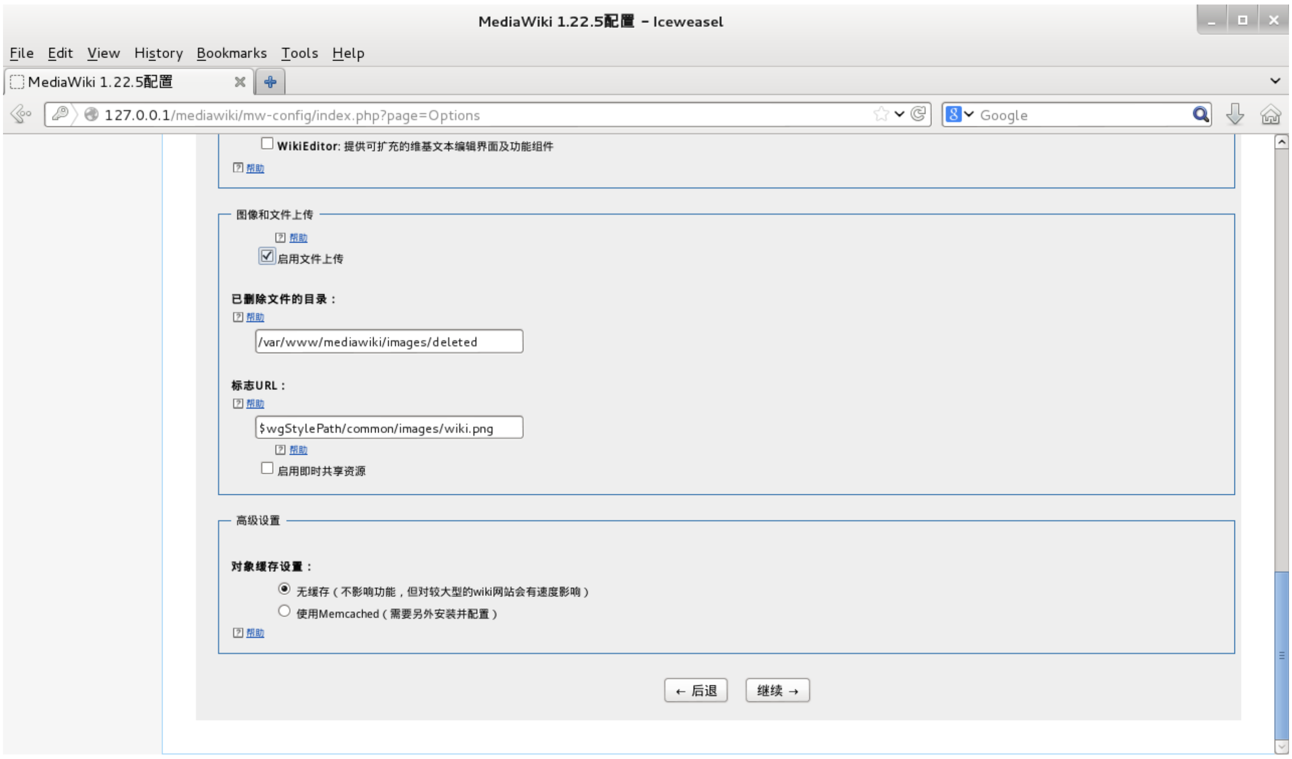
•••••••••••

电子邮件地址：

帮助

帮助

□ 订阅发行公告邮件列表


MediaWiki 1.22.5配置 – Iceweasel
File Edit View History Bookmarks Tools Help
MediaWiki 1.22.5配置
127.0.0.1/mediawiki/mw-config/index.php?page=Options
Google

□ WikiEditor: 提供可扩充的维基文本编辑界面及功能组件

帮助

图像和文件上传

帮助

☑ 启用文件上传

已删除文件的目录：

帮助

/var/www/mediawiki/images/deleted

标志URL：

帮助

$wgStylePath/common/images/wiki.png

帮助

□ 启用即时共享资源

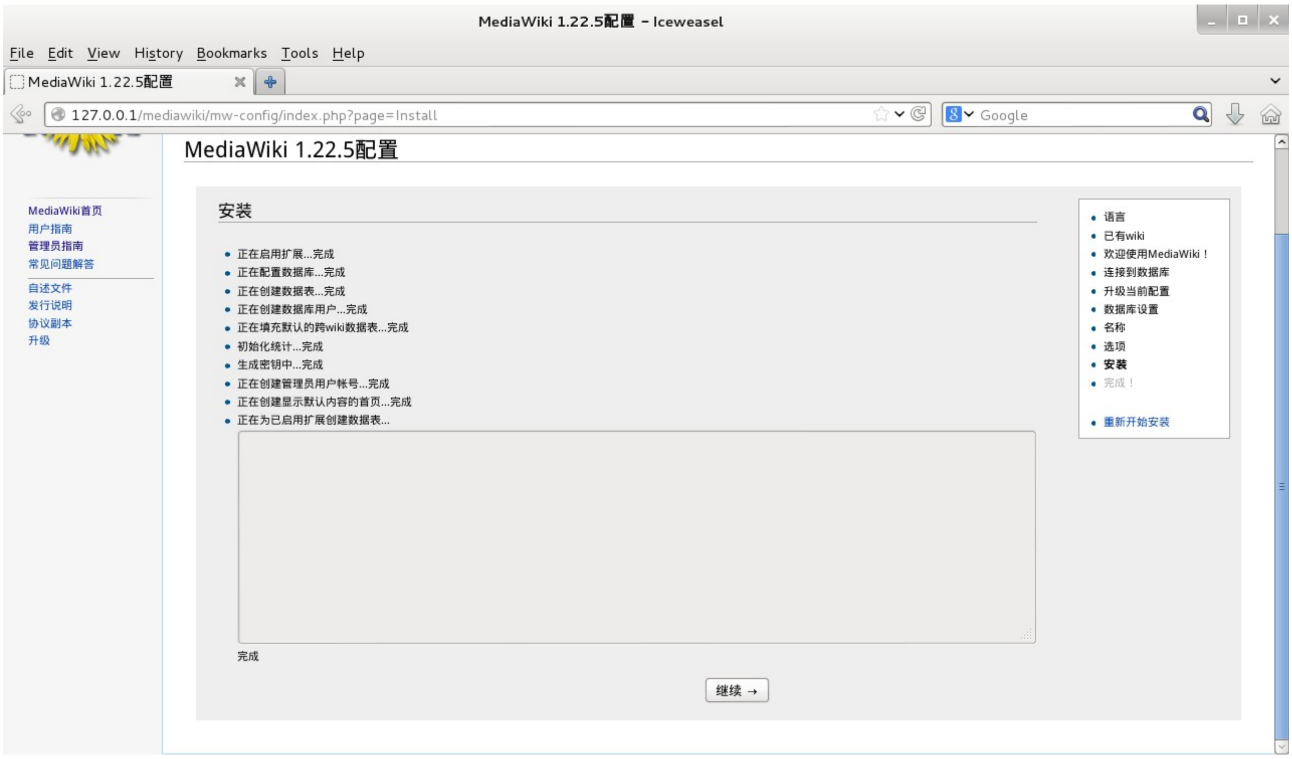
高级设置

对象缓存设置：

● 无缓存（不影响功能，但对较大型的wiki网站会有速度影响）

○ 使用Memcached（需要另外安装并配置）

帮助

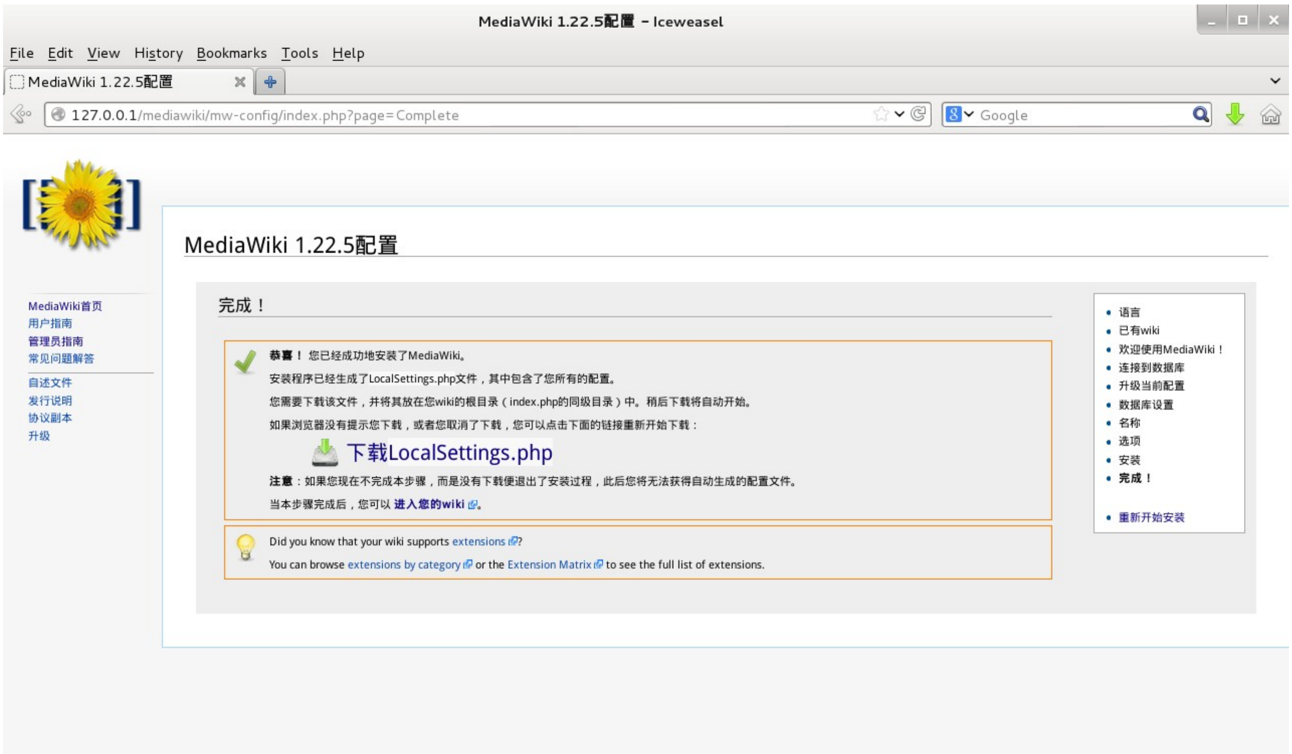

MediaWiki 1.22.5配置 – Iceweasel
File Edit View History Bookmarks Tools Help
MediaWiki 1.22.5配置
127.0.0.1/mediawiki/mw-config/index.php?page=Install
Google
MediaWiki首页
用户指南
管理员指南
常见问题解答
自述文件
发行说明
协议副本
升级

MediaWiki 1.22.5配置

安装

- 正在启用扩展...完成
- 正在配置数据库...完成
- 正在创建数据表...完成
- 正在创建数据库用户...完成
- 正在填充默认的跨wiki数据表...完成
- 初始化统计...完成
- 生成密钥中...完成
- 正在创建管理员用户帐号...完成
- 正在创建显示默认内容的首页...完成
- 正在为已启用扩展创建数据表...

完成



MediaWiki 1.22.5配置 - Iceweasel
File Edit View History Bookmarks Tools Help
127.0.0.1/mediawiki/mw-config/index.php?page=Complete
Google
MediaWiki首页
用户指南
管理员指南
常见问题解答
自述文件
发行说明
协议副本
升级

MediaWiki 1.22.5配置

完成！

恭喜！ 您已经成功地安装了MediaWiki。

安装程序已经生成了LocalSettings.php文件，其中包含了您所有的配置。

您需要下载该文件，并将其放在您wiki的根目录（index.php的同级目录）中。稍后下载将自动开始。

如果浏览器没有提示您下载，或者您取消了下载，您可以点击下面的链接重新开始下载：

下载LocalSettings.php

注意：如果您现在不完成本步骤，而是没有下载便退出了安装过程，此后您将无法获得自动生成的配置文件。

当本步骤完成后，您可以 **进入您的wiki**。

Did you know that your wiki supports extensions?

You can browse extensions by category or the Extension Matrix to see the full list of extensions.

将生成的配置文件 LocalSettings.php 下载保存到/var/www/mediawiki 目录下，此时再次输入以上网址若可以进行 wiki 的访问，则表示 wiki 安装成功;

五、php 安全加固:
//php harden fuc
ref:
http://www.waitalone.cn/php-web-security-for-linux.html
http://www.itokit.com/2012/1006/74782.html

//change file php.ini of debian os
/etc/php5/apache2/php.ini

1、改变 wiki 文件目录和文件属性，禁止写入，在 wiki 根目录下执行:

```
find -type f -name \*.php -exec chmod 444 {} \;
find -type d -exec chmod 555 {} \;
```

2、修改 php 配置文件/etc/php5/apache2/php.ini:

```
vim /etc/php5/apache2/php.ini
```

设置:

```
//设置模式为安全模式，此值直接影响 disable_functions 的命令是否生效;
[SQL]
; http://php.net/sql.safe-mode
sql.safe_mode = On    //设置了后会有问题？？？需要再测试
//禁用不安全的函数
disable_functions = system, show_source, symlink, exec, dl, shell_exec,
passthru, phpinfo, escapeshellarg, escapeshellcmd

//避免暴露 php 信息
expose_php = Off
//关闭错误信息提示
```

```
display_errors = Off
//不允许调用dl
enable_dl = Off
//避免远程调用文件
allow_url_include = Off
```

六、apache加固：

生成数字证书进行双向认证进行https控制访问：

生成密钥和证书方法：

ret:http://citypw.blogspot.tw/2013/11/how-to-set-up-apache2-with-ssltls.html

Generate CA certificates:

```
-------------------------------------------------------------------------------
root@d6-test:/opt/ssl# cp /usr/lib/ssl/misc/CA.sh .
root@d6-test:/opt/ssl# ./CA.sh -newca
CA certificate filename (or enter to create)

Making CA certificate ...
Generating a 2048 bit RSA private key
..................................+++
..............................................+++
writing new private key to './demoCA/private/./cakey.pem'
.............................................
.............................................
Country Name (2 letter code) [AU]:CN
State or Province Name (full name) [Some-State]:Shanghai
Locality Name (eg, city) []:Shanghai
Organization Name (eg, company) [Internet Widgits Pty Ltd]:MOT
Organizational Unit Name (eg, section) []:MOT
Common Name (e.g. server FQDN or YOUR name) []:hardened-shit
Email Address []:info@hardened-shit.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from /usr/lib/ssl/openssl.cnf
Enter pass phrase for ./demoCA/private/./cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number:
            c0:81:0e:bc:52:d0:19:5a
        Validity
            Not Before: Nov 19 02:08:14 2013 GMT
            Not After : Nov 18 02:08:14 2016 GMT
        Subject:
            countryName               = CN
            stateOrProvinceName       = Shanghai
            organizationName          = MOT
            organizationalUnitName    = MOT
            commonName                = hardened-shit
            emailAddress              = info@hardened-shit.com
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                D5:38:4C:2F:FE:CF:E5:19:E9:AC:C5:03:6E:81:6A:D9:15:8F:A8:63
            X509v3 Authority Key Identifier:
                keyid:D5:38:4C:2F:FE:CF:E5:19:E9:AC:C5:03:6E:81:6A:D9:15:8F:A8:63

            X509v3 Basic Constraints:
                CA:TRUE
Certificate is to be certified until Nov 18 02:08:14 2016 GMT (1095 days)

Write out database with 1 new entries
Data Base Updated
----------------------------------------------------------------------------
```

Copy intermediate key and certificate:

```
----------------------------------------------------------------------------
root@d6-test:/opt/ssl# cp demoCA/private/cakey.pem ca.key
root@d6-test:/opt/ssl#
root@d6-test:/opt/ssl# cp demoCA/cacert.pem ca.crt
----------------------------------------------------------------------------
```

Generate server key:

```
----------------------------------------------------------------------------
root@d6-test:/opt/ssl# openssl genrsa -des3 -out server.key 2048
Generating RSA private key, 2048 bit long modulus
...+++
................+++
e is 65537 (0x10001)
Enter pass phrase for server.key:
Verifying - Enter pass phrase for server.key:
----------------------------------------------------------------------------
```

Generate server CSR(Certificate Signing Request) with server key:

```
----------------------------------------------------------------------------
root@d6-test:/opt/ssl# openssl req -new -key server.key -out server.csr
.........................................
....................................
-----
Country Name (2 letter code) [AU]:CN
State or Province Name (full name) [Some-State]:Shanghai
Locality Name (eg, city) []:Shanghai
Organization Name (eg, company) [Internet Widgits Pty Ltd]:MOT
Organizational Unit Name (eg, section) []:MOT
Common Name (e.g. server FQDN or YOUR name) []:hardened-shit
Email Address []:info@hardened-shit.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
----------------------------------------------------------------------------
```

Genrate server certificate:

```
---------------------------------------------------------------------------------
root@d6-test:/opt/ssl# openssl req -x509 -days 2048 -key server.key -in
server.csr > server.crt
Enter pass phrase for server.key
---------------------------------------------------------------------------------
```

You can check out the cert or verify it:

```
openssl x509 -noout -text -in server.crt
openssl verify -CAfile ca.crt server.crt
```

Generate client's key:

```
---------------------------------------------------------------------------------
root@d6-test:/opt/ssl# openssl genrsa -des3 -out client.key 2048
Generating RSA private key, 2048 bit long modulus
...................................................................................................................
...........+++
........+++
e is 65537 (0x10001)
Enter pass phrase for client.key:
Verifying - Enter pass phrase for client.key:
---------------------------------------------------------------------------------
```

Client's CSR:

```
---------------------------------------------------------------------------------
root@d6-test:/opt/ssl# openssl req -new -key client.key -out client.csr
..................................................
..........................................
-----
Country Name (2 letter code) [AU]:CN
State or Province Name (full name) [Some-State]:Shanghai
Locality Name (eg, city) []:Shanghai
Organization Name (eg, company) [Internet Widgits Pty Ltd]:MOT
Organizational Unit Name (eg, section) []:MOT
Common Name (e.g. server FQDN or YOUR name) []:hardened-shit
Email Address []:info@hardened-info.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Generate client's certificate with CA certificate's signature:

```
root@d6-test:/opt/ssl# openssl ca -in client.csr -out client.crt
Using configuration from /usr/lib/ssl/openssl.cnf
Enter pass phrase for ./demoCA/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number:
            c0:81:0e:bc:52:d0:19:5c
        Validity
            Not Before: Nov 19 02:28:13 2013 GMT
            Not After : Nov 19 02:28:13 2014 GMT
        Subject:
            countryName               = CN
            stateOrProvinceName       = Shanghai
            organizationName          = MOT
            organizationalUnitName    = MOT
            commonName                = hardened-shit
            emailAddress              = info@hardened-info.com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                A6:A5:D7:7C:C7:A8:C3:24:C7:90:14:76:84:15:43:D0:2C:0C:31:66
            X509v3 Authority Key Identifier:
                keyid:D5:38:4C:2F:FE:CF:E5:19:E9:AC:C5:03:6E:81:6A:D9:15:8F:A8:63

Certificate is to be certified until Nov 19 02:28:13 2014 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
--------------------------------------------------------------------------------
```

Convert to pkcs12 format, which can be identified by firefox:

```
--------------------------------------------------------------------------------
root@d6-test:/opt/ssl# openssl pkcs12 -export -clcerts -in client.crt
-inkey client.key -out client.pfx
Enter pass phrase for client.key:
Enter Export Password:
Verifying - Enter Export Password:
--------------------------------------------------------------------------------
```

Enable SSL/TLS support in Apache2:

```
--------------------------------------------------------------------------------
root@hardened-shit:/opt# mv ssl /etc/ssl/hardened-shit

root@hardened-shit:/etc/apache2# a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
  service apache2 reload
root@hardened-shit:/etc/apache2# a2enmod ssl
Module ssl already enabled
//change file /etc/apache2/sites-enabled/default-ssl
SSLCertificateFile    /etc/ssl/hardened-shit/server.crt
       SSLCertificateKeyFile /etc/ssl/hardened-shit/server.key
```

```
        SSLCertificateChainFile /etc/ssl/hardened-shit/ca.crt

        SSLCACertificatePath /etc/ssl/hardened-shit/
        SSLCACertificateFile /etc/ssl/hardened-shit/ca.crt

        #SSLVerifyClient optional

        SSLVerifyClient require
        SSLVerifyDepth 10
```

//修改文件 /etc/apache2/ports.conf 以禁用 http:80 端口；

```
#NameVirtualHost *:80
#Listen 80
```

//若需要修改最大连接数等，修改文件 /etc/apache2/apache2.conf

```
StartServers          5
   MinSpareServers       5
   MaxSpareServers      10
   MaxClients          150
   MaxRequestsPerChild   0
```

注意：
以上配置完成后，需要进行服务的重启：

```
service apache2 restart
```

此时要输入服务端私钥的保护密码；

//复制客户端证书及 CA 证书并在需要对 wiki 进行访问的设备上的浏览器端进行导入这两个证书；

```
cp /etc/ssl/hardened-shit/client.pfx /media/Ubuntu\ prec/
cp /etc/ssl/hardened-shit/ca.crt /media/Ubuntu\ prec/
```

修改/var/www/Localsettings.php 文件，若是 80 端口开放的话则下面的为 http，若是开放的是 443 端口的话，那么下面的 http 就要修改为 https。否则页面的布局和格式将出现问题；

```
## The protocol and server name to use in fully-qualified URLs
#$wgServer = "http://127.0.0.1";
$wgServer = "http://111.111.111.120";

//set uploadfile
chmod -R 777 images/
//added this set to Localsettings.php
$wgUploadPath="/var/www/mediawiki/images";
```

3、删除在错误页面中显示服务器信息的功能：

Disallow server to print out any Linux and Apache information on error pages.

Change the following lines at /etc/apache2/conf.d/security

```
sudo nano /etc/apache2/conf.d/security
```

Change the following lines as following:

```
ServerToken Prod
ServerSignature Off
```

4、抗DOS、DDOS Module安装：

In order to avoid HTTTP DoS, DDoS, install the mod_evasive module.

http://www.ansoncheunghk.info/article/two-important-modules-secure-your-apache#mod_evasive

```
apt-get install libapache2-mod-evasive
sudo mkdir /var/log/mod_evasive

sudo chown www-data:www-data /var/log/mod_evasive/


cd /etc/apache2/mods-available/
vim mod-evasive.conf
```

In order to add custom configuration, we need to create mod-evasive.conf file in /etc/apache2/mods-available folder.

```
<ifmodule mod_evasive20.c>

  DOSHashTableSize 3097

  DOSPageCount  2

  DOSSiteCount  50

  DOSPageInterval 1

  DOSSiteInterval  1

  DOSBlockingPeriod  3600

  DOSLogDir   /var/log/mod_evasive

  DOSEmailNotify  info@yourdomain.com

  DOSWhitelist   127.0.0.1

</ifmodule>
```

To let you understand the meaning of each parameter, here are description of each parameter:

DOSHashTableSize: Size of the hash table used to store the IPs.

DOSPageCount: Number of pages allowed per DOSPageInterval.

DOSPageInterval: Time in seconds used by DOSPageCount.

DOSSiteCount: Number of objects allowed per DOSSiteInterval.

DOSSiteInterval: Time in seconds used by DOSSiteCount.

DOSBlockingPeriod: Time in seconds that IPs will be banned. If an IP tries to access the server within this period, the count will be restarted.

DOSLogDir: Optional. Directory to store the logs. If not specified, /tmp will be used.

DOSEmailNotify: Optional. Mail where notifications will be sent.

DOSWhitelist: Optional. List of IPs which won't be blocked.

Once you understand it, now you can changes the value to optimize for your server.

In order to activate the module, we have to inform Apache to enable mod_evasive. To activate the changes, we need to restart the Apache as well.

```
sudo a2enmod mod-evasive

sudo /etc/init.d/apache2 restart
```

Okay, mod_evasive module can guard again DoS or DDoS attack now. How are we going to secure Apache again SQL injection or Code Injection attack??? Other than, providing proper input validation or input filtering. We can install mod_security module to act as the first level web application security.

5、WAF—mod-security install:

```
apt-get install libapache-mod-security
```

download and install the latest OWASP ModSecurity Core Rule Set from the project website. Click here for more information.
https://github.com/SpiderLabs/owasp-modsecurity-crs

```
tar -zxvf SpiderLabs-owasp-modsecurity-crs.tar.gz
mkdir /etc/apache2/mods-security/rules/
cp -R SpiderLabs-owasp-modsecurity-crs-*/* /etc/apache2/mods-security/rules/
cd /etc/apache2/mods-security/rules/
mv modsecurity_crs_10_setup.conf.example modsecurity_crs_10_setup.conf
```

modify modsecurity_crs_10_setup.conf:

```
vi /etc/apache2/mod_security/modsecurity_crs_10_setup.conf
```

Add the following line to the end of the file and save :

```
SecDataDir  /var/www
```

change follow file:

```
vim /etc/apache2/mods-security/mod-security.conf
```

```
<       ifmodule mod_security2.c>
        LoadFile /usr/lib/libxml2.so
        Include /etc/apache2/mods-security/rules/*.conf
        Include /etc/apache2/mods-security/rules/activated_rules/*.conf
        Include /etc/apache2/mods-security/rules/experimental_rules/*.conf
        Include /etc/apache2/mods-security/rules/base_rules/*.conf
        Include /etc/apache2/mods-security/rules/slr_rules/*.conf
        Include /etc/apache2/mods-security/rules/optional_rules/*.conf

        # Turn the filtering engine On or Off

   SecFilterEngine On

   # Make sure that URL encoding is valid

   SecFilterCheckURLEncoding On

   # Unicode encoding check

   SecFilterCheckUnicodeEncoding Off

   # Only allow bytes from this range

   SecFilterForceByteRange 0 255

   # Only log suspicious requests

   SecAuditEngine RelevantOnly

   # The name of the audit log file

   SecAuditLog /var/log/apache2/audit_log

   # Debug level set to a minimum

   SecFilterDebugLog /var/log/apache2/modsec_debug_log

   SecFilterDebugLevel 0

   # Should mod_security inspect POST payloads

   SecFilterScanPOST On

   # By default log and deny suspicious requests
   # with HTTP status 500

   SecFilterDefaultAction "deny,log,status:500"

</IfModule>
```

```
#enable security mod
a2enmod mod-security
#restart apache2 server
service apache2 restart
```

Check if everything is working.

Open the Terminal Window and enter :

```
tail -f /var/log/apache2/error_log
```

The output should look something like this :

```
[Wed Apr 09 02:47:29 2014] [notice] ModSecurity for Apache/2.6.6
(http://www.modsecurity.org/) configured.
[Wed Apr 09 02:47:29 2014] [notice] ModSecurity: APR compiled
version="1.4.6"; loaded version="1.4.6"
[Wed Apr 09 02:47:29 2014] [notice] ModSecurity: PCRE compiled
version="8.30"; loaded version="8.30 2012-02-04"
```