

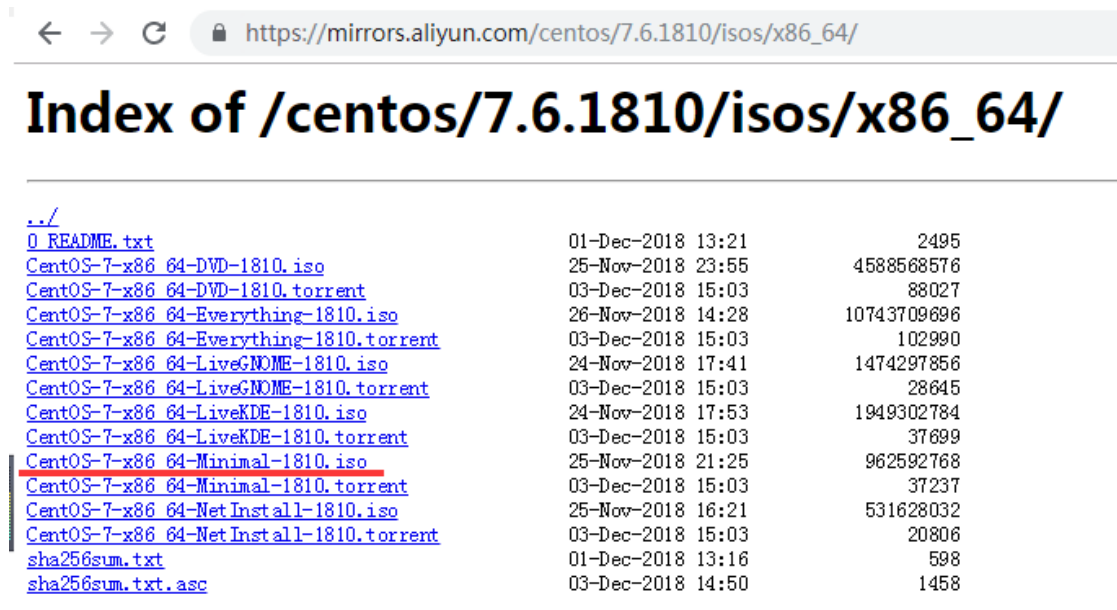
1 任务目标

1. 在Linux上搭建Nginx+php/php-fpm+mysql
2. 对linux主机和应用进行加固

2 完成过程-搭建

3.1 安装CentOS虚拟机

1. 下载ISO文件，从[阿里镜像站](https://mirrors.aliyun.com/centos/7.6.1810/isos/x86_64/)中下载CentOS 7镜像。这里我选择 Minimal（最小运行版），只包含一些操作系统最基本的软件，也没有图形化界面。

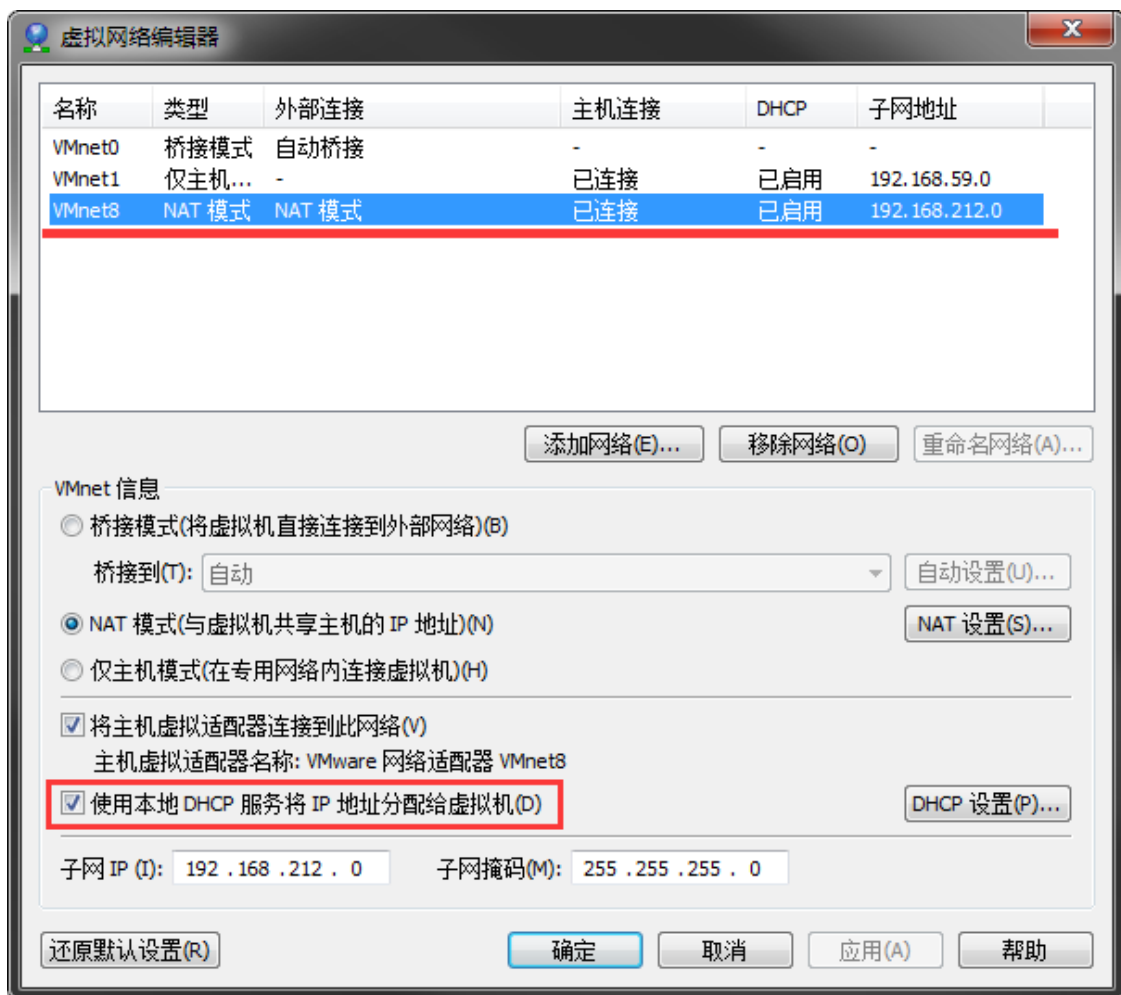


← → ↻ https://mirrors.aliyun.com/centos/7.6.1810/isos/x86_64/		
Index of /centos/7.6.1810/isos/x86_64/		
<hr/>		
../		
0 README.txt	01-Dec-2018 13:21	2495
CentOS-7-x86_64-DVD-1810.iso	25-Nov-2018 23:55	4588568576
CentOS-7-x86_64-DVD-1810.torrent	03-Dec-2018 15:03	88027
CentOS-7-x86_64-Everything-1810.iso	26-Nov-2018 14:28	10743709696
CentOS-7-x86_64-Everything-1810.torrent	03-Dec-2018 15:03	102990
CentOS-7-x86_64-LiveGNOME-1810.iso	24-Nov-2018 17:41	1474297856
CentOS-7-x86_64-LiveGNOME-1810.torrent	03-Dec-2018 15:03	28645
CentOS-7-x86_64-LiveKDE-1810.iso	24-Nov-2018 17:53	1949302784
CentOS-7-x86_64-LiveKDE-1810.torrent	03-Dec-2018 15:03	37699
CentOS-7-x86_64-Minimal-1810.iso	25-Nov-2018 21:25	962592768
CentOS-7-x86_64-Minimal-1810.torrent	03-Dec-2018 15:03	37237
CentOS-7-x86_64-NetInstall-1810.iso	25-Nov-2018 16:21	531628032
CentOS-7-x86_64-NetInstall-1810.torrent	03-Dec-2018 15:03	20806
sha256sum.txt	01-Dec-2018 13:16	598
sha256sum.txt.asc	03-Dec-2018 14:50	1458

2. 虚拟机使用的是VMware workstation 14，新建一个虚拟机，按照步骤，将下载好的ISO导进去，然后都是傻瓜式操作，安装结束后重启，登录操作系统。由于安装的是Minimal版本，所以刚装好时没有网络（因为没有配置IP）、ifconfig也用不了。

```
[root@localhost reven]# ifconfig
bash: ifconfig: command not found
[root@localhost reven]# ping www.baidu.com
ping: www.baidu.com: Name or service not known
[root@localhost reven]# _
```

3. 可以手动先修改配置文件，配置IP、网关、DNS（[参考文章](#)）。但我的虚拟机使用NAT的上网方式，可以使用虚拟局域网中的DHCP来为虚拟主机配置IP地址。



所以，使用 dhclient 就可以自动获取网络配置。

```
[root@localhost reven]# dhclient
[root@localhost reven]# ping www.baidu.com
PING www.a.shifen.com (163.177.151.109) 56(84) bytes of data:
64 bytes from 163.177.151.109 (163.177.151.109): icmp_seq=1 ttl=128 time=37.6 ms
^C
```

4. 虽然有了网络，但是没有ifconfig和netstat是很不方便的，所以我们可以使用yum进行下载。但直接下载提示找不到包。

```
[root@localhost reven]# yum install ifconfig
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
* base: mirrors.163.com
* extras: mirrors.cn99.com
* updates: mirrors.cn99.com
No package ifconfig available.
Error: Nothing to do
[root@localhost reven]#
```

原来ifconfig、netstat等网络工具都是包含在net-tools中，用yum search 去搜索就能知道下载这个工具需要安装的包名。

```
[root@localhost reven]# yum search ifconfig
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
* base: mirrors.163.com
* extras: mirrors.cn99.com
* updates: mirrors.cn99.com
base                                     | 3.6 kB  00:00:00
extras                                 | 3.4 kB  00:00:00
updates                                | 3.4 kB  00:00:00
(1/4): base/7/x86_64/group_gz         | 166 kB  00:00:00
(2/4): extras/7/x86_64/primary_db     | 205 kB  00:00:00
(3/4): base/7/x86_64/primary_db       | 6.0 MB  00:00:03
(4/4): updates/7/x86_64/primary_db    | 6.5 MB  00:00:03
===== Matched: ifconfig =====
net-tools.x86_64 : Basic networking tools
[root@localhost reven]#
```

接下来直接下载net-tools：yum install net-tools。

```
Installed:
  net-tools.x86_64 0:2.0-0.24.20131004git.el7

Complete!

[root@localhost reven]# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.212.137 netmask 255.255.255.0 broadcast 192.168.212.255
    ether 00:0c:29:d3:70:41 txqueuelen 1000 (Ethernet)
    RX packets 11449 bytes 14534410 (13.8 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4392 bytes 271696 (265.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 344 bytes 29864 (29.1 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 344 bytes 29864 (29.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

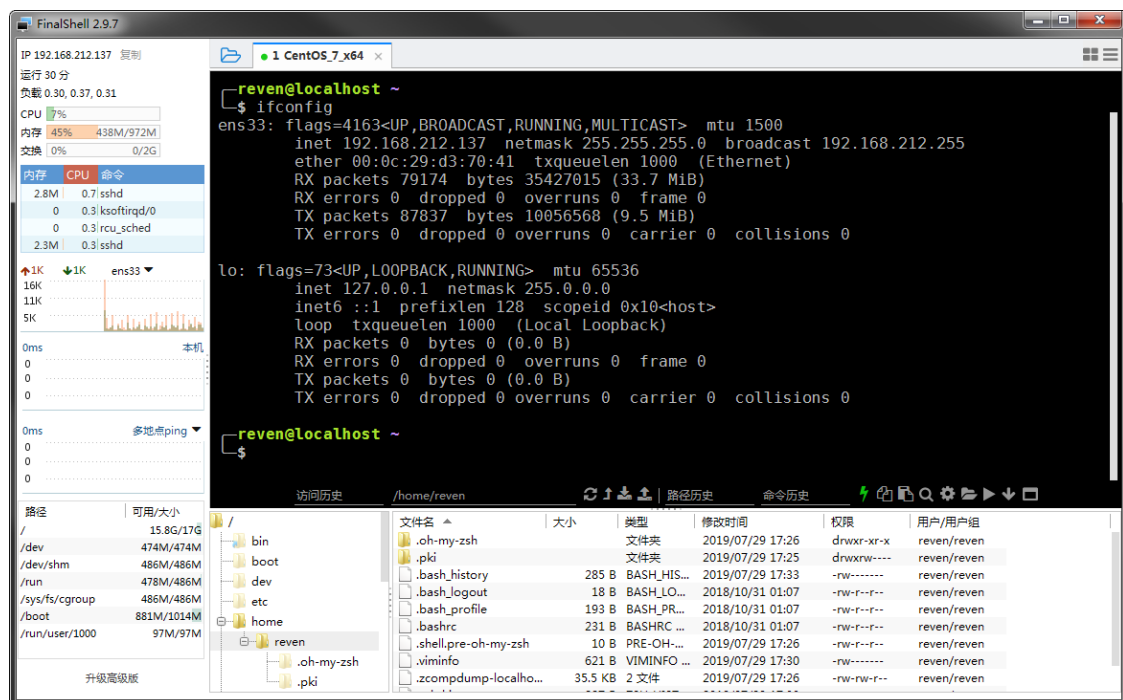
[root@localhost reven]# _
```

5. 由于我这个虚拟机没有图形化界面，直接在上面操作很不方便，所以我将通过ssh去连接服务器进行管理。而ssh服务在操作系统安装完成时已经自动开启了。

```
[root@localhost reven]# netstat -pantl | grep 22
tcp        0      0 0.0.0.0:22          0.0.0.0:*           LISTEN      6901/sshd
tcp6       0      0 :::22              :::*                 LISTEN      6901/sshd

[root@localhost reven]# _
```

我使用外部终端FinalShell进行连接（XShell、Putty都可以）



6. 但是当虚拟机重启之后出现无法自动获取IP地址的情况，需要重新使用dhclient去获取IP。最后通过修改配置文件解决了这个问题（[参考文章](#)）。编辑以下配置文件，将ONBOOT的值修改为yes。

```
vim /etc/sysconfig/network-scripts/ifcfg-eng33
```

```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=dhcp
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=ens33
UUID=6555cd7a-d9ab-47af-b24c-ed5765290d4f
DEVICE=ens33
ONBOOT=yes
```

7. 系统环境搭建好之后还可以使用yum update将系统中的软件和内核更新到最新版。

8. 其他可能会用到，但默认没有安装的工具或组件：

vim、git、wget、redhat-lsb

3.2 安装Nginx

1. 由于CentOS的源中没有收录CentOS，所以直接使用yum进行安装是无法找到安装包的。通过查阅官方文档，找到了CentOS的安装方法（[参考文档](#)）

应先安装yum-utils

```
yum install yum-utils
```

然后创建文件/etc/yum.repos.d/nginx.repo，并写入以下内容：

```
[nginx-stable]
name=nginx stable repo
baseurl=http://nginx.org/packages/centos/$releasever/$basearch/
gpgcheck=1
enabled=1
gpgkey=https://nginx.org/keys/nginx_signing.key

[nginx-mainline]
name=nginx mainline repo
baseurl=http://nginx.org/packages/mainline/centos/$releasever/$basearch/
gpgcheck=1
enabled=0
gpgkey=https://nginx.org/keys/nginx_signing.key
```

默认情况下，nginx-stable这个仓库会处于enable状态，而 nginx-mainline 则处于disable状态。所以需要执行以下命令，启用 nginx-mainline。

```
yum-config-manager --enable nginx-mainline
```

然后就可以安装Nginx了

```
yum install nginx
```

```
reven@localhost ~
$ sudo yum install nginx
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
* base: mirrors.163.com
* extras: mirrors.163.com
* updates: mirrors.163.com
nginx-mainline | 2.9 kB 00:00:00
nginx-stable | 2.9 kB 00:00:00
(1/2): nginx-stable/7/x86_64/primary_db | 46 kB 00:00:01
(2/2): nginx-mainline/7/x86_64/primary_db | 150 kB 00:00:02
Resolving Dependencies
--> Running transaction check
--> Package nginx.x86_64 1:1.17.2-1.el7.ngx will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package Arch Version Repository Size
=====
Installing:
nginx x86_64 1:1.17.2-1.el7.ngx nginx-mainline 767 k
=====
Transaction Summary
=====
Install 1 Package
```

2. 安装官方文档的做法确实成功安装了Nginx，但是它描述中的“默认情况下” nginx-mainline 处于 disable 状态，明显是因为nginx-mainline的enabled参数为0，把这个值改为1，就不用脱裤子肠胃出气了。（既然不需要将nginx-mainline启用，也就不需要yum-config-manager，所以yum-utils也就不是必要的步骤了）
3. 然而最后发现，最简单的方式其实是直接添加nginx官方提供的源就可以直接安装Nginx了（[参考文档](#)）。

```
rpm -Uvh http://nginx.org/packages/centos/7/noarch/RPMS/nginx-release-centos-7-0.el7.noarch.rpm
```

```
root@localhost /home/reven
# rpm -Uvh http://nginx.org/packages/centos/7/noarch/RPMS/nginx-release-centos-7-0.el7.noarch.rpm
Retrieving http://nginx.org/packages/centos/7/noarch/RPMS/nginx-release-centos-7-0.el7.noarch.rpm
Preparing... ##### [100%]
package nginx-release-centos-7-0.el7.noarch is already installed
root@localhost /home/reven
# yum install nginx 1
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
* base: mirrors.163.com
* extras: mirrors.163.com
* updates: mirrors.163.com
Resolving Dependencies
--> Running transaction check
--> Package nginx.x86_64 1:1.16.0-1.el7.ngx will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package Arch Version Repository Size
=====
Installing:
nginx x86_64 1:1.16.0-1.el7.ngx nginx 766 k
=====
```

4. 验证安装是否成功。运行Nginx查看版本为1.16.0，查看Nginx运行占用端口为80端口，尝试访问本机的80端口，返回Nginx的默认页面，说明Nginx安装成功。

```

root@localhost /home/reven
# nginx -v
nginx version: nginx/1.16.0
root@localhost /home/reven
# netstat -ntlp | grep nginx
tcp        0      0 0.0.0.0:*               LISTEN      15047/nginx: master
root@localhost /home/reven
# curl "http://localhost:80"
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>

```

5. 设置开机自启

```
systemctl enable nginx
```

3.3 安装php、php-fpm

1. 纠结了一下到底装php5还是7，记得以前搭过一套php靶机，装了php7无法解析，后来换成php5才成功，原因当时没有深究，只觉得php7有坑。但这次是抱着学习的态度来搭建环境，所以还是要尝试新版本，在这个过程中遇到困难，解决困难才能学到知识。
2. 如果直接使用yum install php，其源中的版本是php5.4，为了下载php7，我们首先得添加几个源。（[参考文章](#)）

```

rpm -Uvh https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
rpm -Uvh https://mirror.webtatic.com/yum/el7/webtatic-release.rpm

```

```

root@localhost /home/reven
# rpm -Uvh https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
Retrieving https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
warning: /var/tmp/rpm-tmp.D9eFmJ: Header V3 RSA/SHA256 Signature, key ID 352c64e5: NOKEY
Preparing...
Updating / installing...
 1:epel-release-7-11
root@localhost /home/reven
# rpm -Uvh https://mirror.webtatic.com/yum/el7/webtatic-release.rpm
Retrieving https://mirror.webtatic.com/yum/el7/webtatic-release.rpm
warning: /var/tmp/rpm-tmp.IT8icJ: Header V4 RSA/SHA1 Signature, key ID 62e74ca5: NOKEY
Preparing...
Updating / installing...
 1:webtatic-release-7-3

```

3. 然后安装php7及其部分基本组件。

```

yum -y install php70w.x86_64 php70w-cli.x86_64 php70w-common.x86_64 php70w-
gd.x86_64 php70w-ldap.x86_64 php70w-mbstring.x86_64 php70w-mcrypt.x86_64
php70w-mysql.x86_64 php70w-pdo.x86_64

```

```

Installed:
php70w.x86_64 0:7.0.33-1.w7          php70w-cli.x86_64 0:7.0.33-1.w7          php70w-common.x86_64 0:7.0.33-1.w7
php70w-gd.x86_64 0:7.0.33-1.w7       php70w-ldap.x86_64 0:7.0.33-1.w7        php70w-mbstring.x86_64 0:7.0.33-1.w7
php70w-mcrypt.x86_64 0:7.0.33-1.w7   php70w-mysql.x86_64 0:7.0.33-1.w7       php70w-pdo.x86_64 0:7.0.33-1.w7

Dependency Installed:
apr.x86_64 0:1.4.8-3.el7_4.1          apr-util.x86_64 0:1.5.2-6.el7           httpd.x86_64 0:2.4.6-89.el7.centos
httpd-tools.x86_64 0:2.4.6-89.el7.centos  libX11.x86_64 0:1.6.5-2.el7           libX11-common.noarch 0:1.6.5-2.el7
libXau.x86_64 0:1.0.8-2.1.el7          libXpm.x86_64 0:3.5.12-1.el7           libjpeg-turbo.x86_64 0:1.2.90-6.el7
libmcrypt.x86_64 0:2.5.8-13.el7        libtool-libs.x86_64 0:2.4.2-22.el7_3    libxcb.x86_64 0:1.13-1.el7
mailcap.noarch 0:2.1.41-2.el7

Complete!
[root@localhost /home/reven]# php -v
PHP 7.0.33 (cli) (built: Dec 6 2018 22:30:44) ( NTS )
Copyright (c) 1997-2017 The PHP Group
Zend Engine v3.0.0, Copyright (c) 1998-2017 Zend Technologies
[root@localhost /home/reven]#

```

4. 再安装php-fpm

```
yum -y install php70w-fpm php70w-opcache
```

安装完成后启动php-fpm，并设置开机自启

```
systemctl start php-fpm
systemctl enable php-fpm
```

```

Installed:
php70w-fpm.x86_64 0:7.0.33-1.w7          php70w-opcache.x86_64 0:7.0.33-1.w7

Complete!
[root@localhost /home/reven]# systemctl start php-fpm
[root@localhost /home/reven]#

```

5. 检测php能否与Nginx配合使用。

a. 查看Nginx的配置文件/etc/nginx/conf.d/default.conf，找到Nginx指向的web根目录

```

[root@localhost /home/reven]# cat /etc/nginx/conf.d/default.conf
server {
    listen      80;
    server_name localhost;

    #charset koi8-r;
    #access_log /var/log/nginx/host.access.log  main;

    location / {
        root    /usr/share/nginx/html;
        index  index.html index.htm;
    }

    #error_page  404              /404.html;

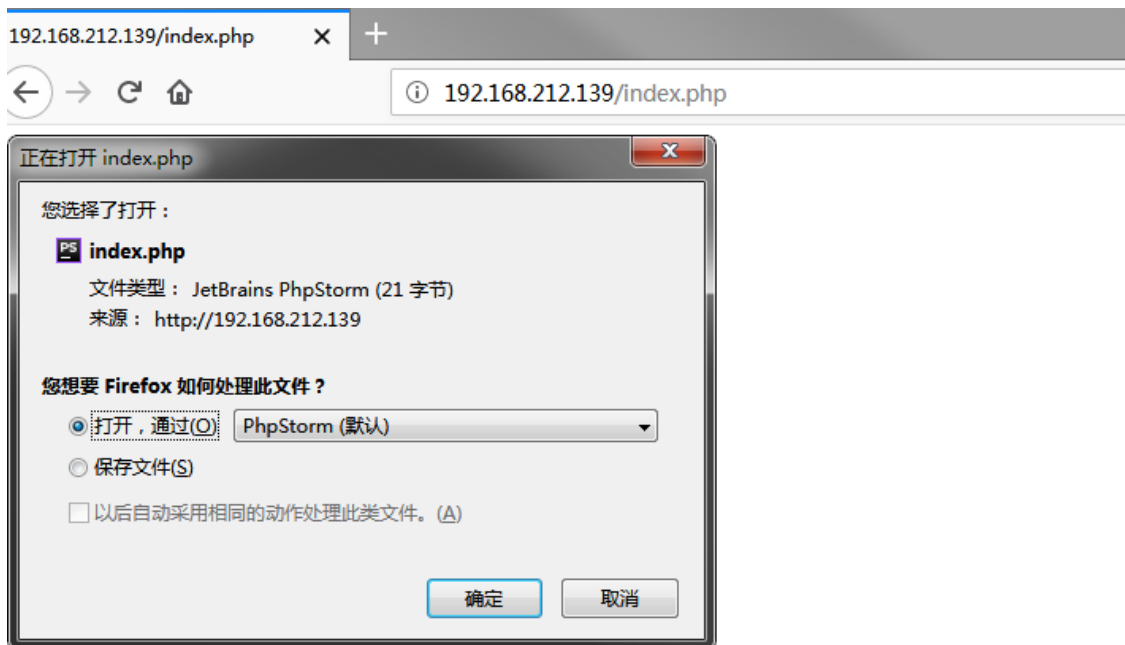
    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root    /usr/share/nginx/html;
    }
}

```

b. 进入web根目录，新建一个文件名为index.php，内容如下：

```
<?php
    phpinfo();
?>
```

c. 直接在浏览器中访问<http://192.168.212.139/index.php>，此时Nginx只当它是一个普通文件，直接返回文本数据，浏览器做下载处理。



d. 此时还需修改Nginx的配置文件，内容如下：

```
location ~ \.php$ {
    root          /usr/share/nginx/html;
    fastcgi_pass  127.0.0.1:9000;
    fastcgi_index index.php;
    fastcgi_param SCRIPT_FILENAME
    $document_root$fastcgi_script_name;
    include       fastcgi_params;
}
```

（按照参考文章中的配置操作，其实还是出现了找不到文件的问题，后来参考了[另一篇文章](#)：因为\$document_root的参数是由root html那一行定义的，默认是在/usr/share/nginx/html/ 所以把html换成站点根目录就正常了。）

修改完成后，重启Nginx，并设置开机自启

```
systemctl restart nginx
```

再次访问即可成功解析php

192.168.212.139/index.php

PHP Version 7.0.33

System	Linux localhost.localdomain 3.10.0-957.el7.x86_64 #1 SMP Thu Nov 8 23:39:32 UTC 2018 x86_64
Build Date	Dec 6 2018 22:32:48
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/bc2.ini, /etc/php.d/calendar.ini, /etc/php.d/ctype.ini, /etc/php.d/curl.ini, /etc/php.d/exif.ini, /etc/php.d/fileinfo.ini, /etc/php.d/ftp.ini, /etc/php.d/gd.ini, /etc/php.d/gettext.ini, /etc/php.d/gmp.ini, /etc/php.d/iconv.ini, /etc/php.d/json.ini, /etc/php.d/ldap.ini, /etc/php.d/mbstring.ini, /etc/php.d/mcrypt.ini, /etc/php.d/mysql.ini, /etc/php.d/openssl.ini, /etc/php.d/pdo.ini, /etc/php.d/pdo_mysql.ini, /etc/php.d/pdo_sqlite.ini, /etc/php.d/phar.ini, /etc/php.d/shmop.ini, /etc/php.d/simplexml.ini, /etc/php.d/sockets.ini, /etc/php.d/sqlite3.ini, /etc/php.d/tokenizer.ini, /etc/php.d/xml.ini, /etc/php.d/zip.ini
PHP API	20151012
PHP Extension	20151012
Zend Extension	320151012
Zend Extension Build	API320151012.NTS
PHP Extension Build	API20151012.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled

3.4 安装Mysql

1. 如果直接使用yum install mysql，CentOS 默认会指向一个叫mariadb的数据库，简单了解了一下发现其为mysql的一个变种，或者说是分支，完全兼容mysql，并且拥有一些mysql企业版才有的功能。

```
# yum install mysql
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
* base: mirrors.cn99.com
* epel: mirrors.njupt.edu.cn
* extras: mirrors.cn99.com
* updates: mirrors.163.com
* webtatic: uk.repo.webtatic.com
Resolving Dependencies
--> Running transaction check
--> Package mariadb.x86_64 1:5.5.60-1.el7_5 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                Arch                Version              Repository            Size
=====
Installing:
mariadb                x86_64              1:5.5.60-1.el7_5     base                  8.9 M
Transaction Summary


```

2. 但是我还是打算安装纯种的mysql，这里安装mysql 5.7 版本。（[参考文章](#)）

- a. 首先依然是添加mysql的源

```
rpm -Uvh http://dev.mysql.com/get/mysql57-community-release-el7-8.noarch.rpm
```

```
root@localhost /usr/share/nginx/html
# rpm -Uvh http://dev.mysql.com/get/mysql57-community-release-el7-8.noarch.rpm
Retrieving http://dev.mysql.com/get/mysql57-community-release-el7-8.noarch.rpm
warning: /var/tmp/rpm-tmp.coakvN: Header V3 DSA/SHA1 Signature, key ID 5072e1f5: NOKEY
Preparing...
Updating / installing...
 1:mysql57-community-release-el7-8
root@localhost /usr/share/nginx/html
```

- b. 安装好后，默认是5.7的版本，如果想安装5.6或5.5版本，可以修改配置文件/etc/yum.repos.d/mysql-community.repo，将对应版本的enabled值为1，其他版本置为0。

```
# Enable to use MySQL 5.5
[mysql55-community]
name=MySQL 5.5 Community Server
baseurl=http://repo.mysql.com/yum/mysql-5.5-community/el/7/$basearch/
enabled=0
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

# Enable to use MySQL 5.6
[mysql56-community]
name=MySQL 5.6 Community Server
baseurl=http://repo.mysql.com/yum/mysql-5.6-community/el/7/$basearch/
enabled=0
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

[mysql57-community]
name=MySQL 5.7 Community Server
baseurl=http://repo.mysql.com/yum/mysql-5.7-community/el/7/$basearch/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-mysql
```

- c. 安装mysql

```
yum install mysql-community-server
```

这下安装的就是mysql 5.7的版本了。接下来是个漫长的等待过程。

Dependencies Resolved

Package	Arch	Version	Repository	Size
Installing:				
mysql-community-libs	x86_64	5.7.27-1.el7	mysql57-community	2.2 M
replacing mariadb-libs.x86_64 1:5.5.60-1.el7_5				
mysql-community-libs-compat	x86_64	5.7.27-1.el7	mysql57-community	2.0 M
replacing mariadb-libs.x86_64 1:5.5.60-1.el7_5				
mysql-community-server	x86_64	5.7.27-1.el7	mysql57-community	165 M
Installing for dependencies:				
mysql-community-client	x86_64	5.7.27-1.el7	mysql57-community	24 M
mysql-community-common	x86_64	5.7.27-1.el7	mysql57-community	275 k
Transaction Summary				
Install 3 Packages (+2 Dependent packages)				
Total download size: 194 M				
Is this ok [y/d/N]: y				

d. 安装完成后，启动mysql服务。

```
systemctl start mysqld
```

3306端口已经开启，说明mysql服务已经在运行。

```
Complete!
[root@localhost /usr/share/nginx/html]# systemctl start mysqld
[root@localhost /usr/share/nginx/html]# netstat -pant | grep mysql
tcp6      0      0      0 :::3306          :::*              LISTEN      96958/mysqld
[root@localhost /usr/share/nginx/html]#
```

e. 设置开机自启

```
systemctl enable mysqld
systemctl daemon-reload
```

3. 修改root本地密码

a. 查看密码：5.7以前的版本默认是空密码，5.7版本为root用户随机生成了一个密码，保存在日志中，可以执行以下命令查看。

```
grep 'temporary password' /var/log/mysqld.log
```

```
[root@localhost /usr/share/nginx/html]# grep 'temporary password' /var/log/mysqld.log
2019-07-30T19:25:10.078381Z 1 [Note] A temporary password is generated for root@localhost: rhJdZ)_d)4?s
[root@localhost /usr/share/nginx/html]#
```

b. 登录mysql

```
mysql -uroot -p
```

登录之后必须马上修改临时密码才能执行其他语句，并且密码策略要求密码最小长度为8，至少含有1个数字，1个小写，1个大写字母，1个特殊字符（[参考文章](#)）。

```
set password for 'root'@'localhost'=password('MysqlP4ssword!');
```

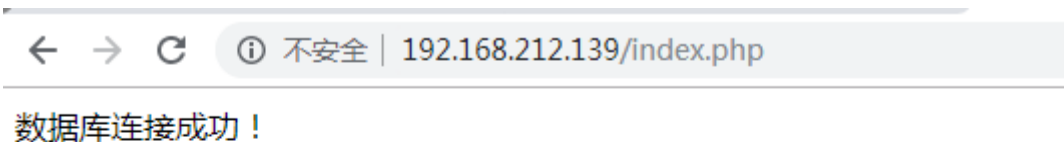
```
mysql> SHOW VARIABLES LIKE 'validate_password%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| validate_password_check_user_name | OFF |
| validate_password_dictionary_file | |
| validate_password_length | 8 |
| validate_password_mixed_case_count | 1 |
| validate_password_number_count | 1 |
| validate_password_policy | MEDIUM |
| validate_password_special_char_count | 1 |
+-----+-----+
7 rows in set (0.01 sec)
```

4. 检测PHP是否能与mysql互通

a. 将web根目录(/usr/share/nginx/html)下的index.php修改为如下内容：

```
<?php
//创建连接
$test = mysqli_connect('localhost','root','MysqlP4ssword!');//数据库服务器
地址、用户名、密码
//检测
if(!$test){
    echo "连接失败，请检查mysql服务以及账号密码";
}else{
    echo "数据库连接成功！";
}
?>
```

b. 再次访问<http://192.168.212.139/index.php>，显示“数据库连接成功！”



5. 执行mysql命令（这步是加固操作后补上的，所以用户名不同）

a. 将index.php修改为如下内容：

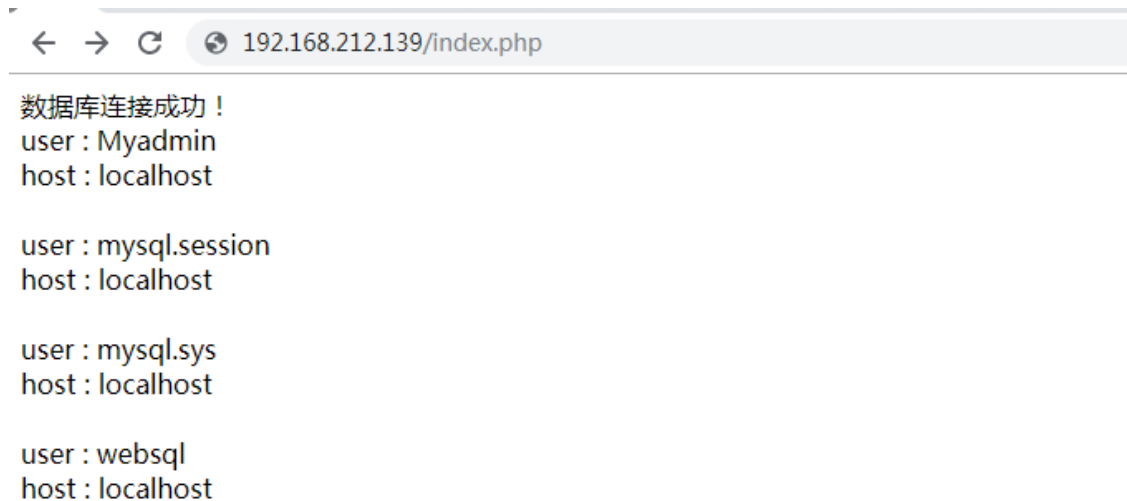
```
<?php
//创建连接
$test =
mysqli_connect('localhost','Myadmin','MysqlP4ssword!','mysql');//数据库服务器
地址、用户名、密码、数据库名称
//检测
if(!$test){
    echo "连接失败，请检查mysql服务以及账号密码";
}else{
    echo "数据库连接成功！<br/>";
    $query = "select user,host from user";
    $result = $test->query($query);
```

```

        while ($obj = mysqli_fetch_object($result)){
            foreach ($obj as $key => $value){
                echo "$key : $value";
                echo "<br/>";
            }
            echo "<br/>";
        }
    }
?>

```

b. 再次访问<http://192.168.212.139/index.php> , 显示查询到的内容



3 完成过程-加固

3.1 主机加固

1. 设置密码策略

#编辑配置文件 /etc/login.defs

将密码最长保质期设为90天
 口令更改最小间隔天数设为7天
 口令最小长度设置为8
 口令过期前警告天数设为7天

```

18 # Password aging controls:
19 #
20 #     PASS_MAX_DAYS   Maximum number of days a password may be used.
21 #     PASS_MIN_DAYS   Minimum number of days allowed between password changes.
22 #     PASS_MIN_LEN     Minimum acceptable password length.
23 #     PASS_WARN_AGE   Number of days warning given before a password expires.
24 #
25 PASS_MAX_DAYS   90
26 PASS_MIN_DAYS   7
27 PASS_MIN_LEN     8
28 PASS_WARN_AGE   7

```

#编辑/etc/pam.d/system-auth

添加一条策略: 至少8位, 包含1位大写字母, 1位小写字母和1位数字

```

password requisite pam_cracklib.so difok=3 minlen=8 ucredit=-1
lcredit=-1 dcredit=1

```

```

11 account    required    pam_unix.so
12 account    sufficient pam_localuser.so
13 account    sufficient pam_succeed_if.so uid < 1000 quiet
14 account    required    pam_permit.so
15
16 password    requisite    pam_pwquality.so try_first_pass local_users_only retry=3 authtok_type=
17 password    sufficient pam_unix.so sha512 shadow nullok try_first_pass use_authtok
18 password    required    pam_deny.so
19 password    required    pam_cracklib.so difok=3 minlen=8 ucredit=-1 lcredit=-1 dcredit=1
20

```

加固必要性：提升口令爆破难度

2. 限制root远程登录

```

#编辑配置文件/etc/ssh/sshd_config
    将PermitRootLogin的值设为 no
    将MaxAuthTries 的值设为 6

```

```

35 # Authentication:
36
37 #LoginGraceTime 2m
38 PermitRootLogin no
39 #StrictModes yes
40 MaxAuthTries 6
41 #MaxSessions 10
42
43 #PubkeyAuthentication yes
44

```

加固必要性：root用户允许远程登录有被爆破口令的风险。最大尝试次数设置为6可有效防止ssh暴力破解。

3. 设置账号锁定策略

```

#编辑配置文件：/etc/pam.d/system-auth
    添加一条策略：登录失败10次，锁定账号300秒
    auth required pam_tally.so onerr=fail deny=10 unlock_time=300

```

```

1  #%PAM-1.0
2  # This file is auto-generated.
3  # User changes will be destroyed the next time authconfig is run.
4  auth      required    pam_env.so
5  auth      required    pam_faildelay.so delay=2000000
6  auth      sufficient   pam_unix.so nullok try_first_pass
7  auth      requisite    pam_succeed_if.so uid >= 1000 quiet_success
8  auth      required    pam_deny.so
9  auth      required    pam_tally.so onerr=fail deny=10 unlock_time=300
10

```

加固必要性：防止系统用户口令被暴力破解

4. 设置登录超时

```

#编辑配置文件/etc/profile
    添加一条设置：登录后超过300秒未进行任何操作，则超时退出
    export TMOUT=300
#重载配置文件
    source /etc/profile

```

```

70         . "$i" >/dev/null
71     fi
72 fi
73 done
74
75 unset i
76 unset -f pathmunge
77 export TMOUT=300

```

3.2 Nginx加固

1. 删除默认页面及默认错误页面

```
rm /usr/share/nginx/html/50x.html index.html
```

```

root@localhost /usr/share/nginx/html
# ls
50x.html index.html index.php
root@localhost /usr/share/nginx/html
# rm 50x.html index.html
root@localhost /usr/share/nginx/html
# ls
index.php
root@localhost /usr/share/nginx/html
#

```

加固必要性：删除默认页面可以防止攻击者根据这些页面判断服务器中间件类型及版本。

2. 更改404和400返回页面

```

#在web根目录下创建自己的404.html页面
    只需要将原本的404页面复制，将版本号换掉即可。
#在server定义区域中加入：error_page 404 /404.html
    默认将该行注释，只要将#删除即可。
#重启nginx
systemctl restart nginx

```

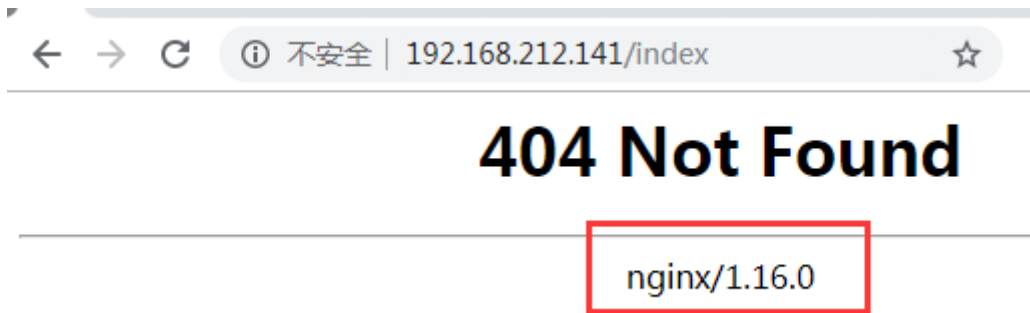
```

1 server {
2     listen      80;
3     server_name localhost;
4
5     #charset koi8-r;
6     #access_log /var/log/nginx/host.access.log  main;
7
8     location / {
9         root    /usr/share/nginx/html;
10        index   index.html index.htm;
11    }
12
13    error_page 404          /404.html;
14    error_page 400          /400.html;
15

```

加固必要性：防止Nginx版本信息泄漏

加固前：



```
HTTP/1.1 400 Bad Request
Server: nginx/1.16.0
Date: Sat, 03 Aug 2019 09:30:37 GMT
Content-Type: text/html
Content-Length: 157
Connection: close

<html>
<head><title>400 Bad Request</title></head>
<body>
<center><h1>400 Bad Request</h1></center>
<hr><center>nginx/1.16.0</center>
</body>
</html>
```

加固后：



3. 隐藏http返回头部中的版本号

```
#编辑配置文件/etc/nginx/nginx.conf
    在http定义区域中加入: server_tokens off;
#重启nginx
systemctl restart nginx
```



```

12
13
14 http {
15     include        /etc/nginx/mime.types;
16     default_type   application/octet-stream;
17
18     log_format main '$remote_addr - $remote_user [$time_local] "$request" '
19                     '$$status $body_bytes_sent "$http_referer" '
20                     '"$http_user_agent" "$http_x_forwarded_for"';
21
22     access_log /var/log/nginx/access.log main;
23     server_tokens off;
24     sendfile      on;
25     #tcp_nopush    on;

```

加固必要性：防止版本号泄漏

加固前：

Request

Raw Headers Hex

GET /index.php HTTP/1.1
Host: 192.168.212.141
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.131 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
Accept-Language: zh-CN,zh;q=0.9
Connection: close

Response

Raw Headers Hex

HTTP/1.1 200 OK
Server: nginx/1.16.0
Date: Sat, 03 Aug 2019 08:10:46 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Content-Length: 216

数据库连接成功!
user : Myadmin
host : localhost

user : mysql.session
host : localhost

user : mysql.sys
host : localhost

user : websql
host : localhost

加固后：

Request

Raw Headers Hex

GET /index.php HTTP/1.1
Host: 192.168.212.141
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.131 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
Accept-Language: zh-CN,zh;q=0.9
Connection: close

Response

Raw Headers Hex

HTTP/1.1 200 OK
Server: nginx
Date: Sat, 03 Aug 2019 09:43:47 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Content-Length: 216

数据库连接成功!
user : Myadmin
host : localhost

user : mysql.session
host : localhost

user : mysql.sys
host : localhost

user : websql
host : localhost

3.3 Mysql加固

1. 禁用mysql命令历史记录

```

#删除现有的.mysql_history文件
rm ~/.mysql_history
#创建.mysql_history的软链接指向黑洞 (/dev/null)
ln -s /dev/null ~/.mysql_history

```

```

root@localhost /home/reven
# ls -la ~/.mysql_history
-rw-----. 1 root root 227 Aug  1 09:27 /root/.mysql_history
root@localhost /home/reven
# rm ~/.mysql_history
root@localhost /home/reven
# ln -s /dev/null ~/.mysql_history
root@localhost /home/reven
# ls -la ~/.mysql_history
lrwxrwxrwx. 1 root root 9 Aug  2 07:40 /root/.mysql_history -> /dev/null
root@localhost /home/reven

```

加固必要性：虽然mysql_history只有root权限才有读写权限，但攻击者依然有可能通过提权漏洞获取到root权限，并且历史命令均以明文形式存储，若被攻击者直接读取，将可能造成更大的损失。

2. 重命名root账号

```
#切换到mysql数据库
use mysql;
#更新user表中用户名为root的用户名为“Myadmin”
update user set user="Myadmin" where user="root";
#刷新权限
flush privileges;
```

```
mysql> use mysql;
Database changed
mysql> select user,host from user;
+-----+-----+
| user          | host          |
+-----+-----+
| mysql.session | localhost     |
| mysql.sys     | localhost     |
| root         | localhost     |
+-----+-----+
3 rows in set (0.00 sec)

mysql> update user set user="Myadmin" where user="root";
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select user,host from user;
+-----+-----+
| user          | host          |
+-----+-----+
| Myadmin       | localhost     |
| mysql.session | localhost     |
| mysql.sys     | localhost     |
+-----+-----+
3 rows in set (0.00 sec)
```

加固必要性：mysql最高管理员用户名默认为root，攻击者一般会针对root用户进行爆破，修改默认用户名可以降低攻击者的成功率，也增加攻击成本。

3. 创建一个新用户

```
#创建一个用户
create user 'websql'@'localhost' identified by 'Mysql_f0r_Web';
#授予权限，只授予增删改查的权限（webdata为数据库名）
grant insert,delete,update,select on webData.* to 'websql'@'localhost';
#刷新权限
flush privileges;
```

```
mysql> create user 'websql'@'localhost' identified by 'Mysql_f0r_Web';
Query OK, 0 rows affected (0.00 sec)

mysql> grant insert,delete,update,select on webData.* to 'websql'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

加固必要性：一般的web应用程序只需要增删改查的权限，而不需要有其他数据库管理权限。根据最小权限原则，故需要创建一个符合业务需求权限的新用户给应用使用。

3.4 PHP加固

1. 隐藏HTTP响应头部中的PHP版本信息

```
#编辑/etc/php.ini
    将expose_php的值修改为off
#重启php-fpm
systemctl restart php-fpm
```

```
353
354 ; Decides whether PHP may expose the fact that it is installed on the server
355 ; (e.g. by adding its signature to the Web server header). It is no security
356 ; threat in any way, but it makes it possible to determine whether you use PHP
357 ; on your server or not.
358 ; http://php.net/expose-php
359 expose_php = Off
360
```

加固必要性：隐藏php版本信息，防止攻击者根据php版本信息找到系统的脆弱点。

加固前：



The screenshot shows the 'Response' tab of a web browser's developer tools. The 'X-Powered-By' header is highlighted in orange and reads 'PHP/7.0.33'. The response body shows a successful database connection message: '数据库连接成功!
user : Myadmin
host : localhost
user : mysql.session
host : localhost
user : mysql.sys
host : localhost
user : websql
host : localhost
'.

加固后：



The screenshot shows the 'Response' tab of a web browser's developer tools. The 'X-Powered-By' header is no longer present. The response body shows the same successful database connection message: '数据库连接成功!
user : Myadmin
host : localhost
user : mysql.session
host : localhost
user : mysql.sys
host : localhost
user : websql
host : localhost
'.

2. 关闭上传目录的php执行权限

```
#编辑/etc/php.ini,添加以下内容
<Directory "/usr/share/nginx/html/upload">
    php_admin_value engine = off
</Directory>
#重启php-fpm
systemctl restart php-fpm
```

加固必要性：如果允许上传目录执行php，则有可能被攻击者上传木马，getshell。

(网站还未搭建，还未有上传目录，此项暂不做具体操作)

3. 检查是否关闭远程文件包含功能

```
#编辑/etc/php.ini
allow_url_include = off #允许远程包含文件，默认关闭
```

加固必要性：攻击者利用文件包含漏洞，可以读取系统中的文件内容，也可以包含php木马，getshell。

其实可以加固的项还有很多很多，这里没有一一列出。

参考文章：

[linux安全基线加固](#)

[安全基线检查与配置浅析](#)