

## 0x01 基本任务

### 环境准备

实体机: Windows 10

虚拟机: Ubuntu 16.04

Nginx: 1.17.1

PHP: 5.6.40-8

MySQL: 5.7.26

### Nginx搭建

#### 1. 安装Nginx的依赖包

```
# 查看zlib是否安装
dpkg -l | grep zlib
# 解决依赖包openssl安装
sudo apt-get install openssl libssl-dev
# 解决依赖包pcre安装
sudo apt-get install libpcre3 libpcre3-dev
# 解决依赖包zlib安装
sudo apt-get install zlib1g-dev
```

#### 2. 下载Nginx

```
# 下载Nginx
wget http://nginx.org/download/nginx-1.17.1.tar.gz
# 解压Nginx
tar -xzvf nginx-1.17.1.tar.gz
# 重命名文件夹
mv nginx-1.17.1 nginx
# 移动文件夹到ubuntu常见软件目录下
mv nginx/ /usr/local/
```

#### 3. 安装Nginx

```
# 配置Nginx
cd /usr/local/nginx
sudo ./configure --prefix=/usr/local/nginx --conf-path=/usr/local/nginx/nginx.conf
# 编译Nginx
sudo make
# 安装Nginx
sudo make install
```

#### 4. 检查Nginx是否安装成功

```
cd /usr/local/nginx/sbin
./nginx -t

# 成功标志
# root@k1ea4c:/usr/local/nginx/sbin# ./nginx -t

# nginx: the configuration file /usr/local/nginx/nginx.conf syntax is ok

# nginx: configuration file /usr/local/nginx/nginx.conf test is successful
```

#### 5. 修改监听端口（防止跟后面apache的80端口冲突）

```
vim /usr/local/nginx/nginx.conf
将 listen 80 修改为 listen 8080
```

#### 6. 配置Nginx的运行用户

```
# 添加www组
groupadd www
# 创建nginx运行账户www并加入到www组，不允许www用户直接登录系统
useradd -g www www -s /bin/false
# 修改nginx运行用户
vim /usr/local/nginx/nginx.conf
在首部插入 user www www;
```

#### 7. 启动Nginx

```
cd /usr/local/nginx/sbin
./nginx
```

### PHP安装

```
add-apt-repository ppa:ondrej/php


apt-get -y update
apt-get -y install php5.6 php5.6-mcrypt php5.6-mbstring php5.6-curl php5.6-cli
php5.6-mysql php5.6-gd php5.6-xml php5.6-fpm
```

### 配置PHP+Nginx

```
vim /etc/php/5.6/fpm/pool.d/www.conf
```

 5d33e92af0a7271171

```
vim /usr/local/nginx/nginx.conf
```

 5d33e9380769c79870 5d33e9424996c78078

## MySQL搭建

### 1. 安装MySQL

```
sudo apt-get install mysql-server  
sudo apt-get install mysql-client  
sudo apt-get install libmysqlclient-dev
```

### 2. 设置登录密码

```
# 使用MySQL自带的命令mysqladmin设置root密码  
  
mysqladmin -u root password "root"  
  
# 使用mysqladmin命令更改root密码(备用)  
  
mysqladmin -u root -proot password "toor"
```

### 3. 配置远程访问

```
vim /etc/mysql/mysql.conf.d/mysqld.cnf  
  
注释掉 bind-address = 127.0.0.1
```

```
# 进入mysql服务, 执行授权命令  
grant all on *.* to root@'%' identified by '你的密码' with grant option;  
flush privileges;
```

```
# 重启mysql服务  
service mysql restart
```

## 0x02 扩展任务

### 环境准备

实体机: Windows 10

虚拟机: Windows Serv 2012

虚拟机: Ubuntu 16.04

Apache: 2.4.18

Tomcat: 9.0.22

### Apache搭建

```
sudo apt install apache2
```


### Tomcat 搭建

#### 1. 下载tomcat

```
# 下载安装包  
wget https://mirrors.tuna.tsinghua.edu.cn/apache/tomcat/tomcat-  
9/v9.0.22/bin/apache-tomcat-9.0.22.tar.gz  
# 解压  
tar -zxvf apache-tomcat-9.0.22.tar.gz  
# 移动目录  
mv apache-tomcat-9.0.22 /opt
```


#### 2. 修改服务端口

```
# vim /opt/apache-tomcat-9.0.22/conf/server.xml
```

5d3c03a5c325856205


#### 3. 启动或终止服务

```
/opt/apache-tomcat-9.0.22/startup.sh  
/opt/apache-tomcat-9.0.22/shutdown.sh
```

 5d3c04089bd7110142

## IIS搭建

[参考链接](<https://blog.csdn.net/KamRoseLee/article/details/79270454>)

 5d396b74b4b7746031

## 0x03 安全加固

### Linux加固

1. 禁用无用号，查口令文件，确认服务账号的sh应为/sbin/nologin

```
cat /etc/passwd | grep bin.bash
awk -F: '($2==""){print $1}' /etc/shadow
awk -F: '($2==""){print $1}' /etc/passwd
```

2. 添加口令策略

```
# 方式一
# vim /etc/login.defs

PASS_MAX_DAYS 90 #新建用户的密码最长使用天数
PASS_MIN_DAYS 0 #新建用户的密码最短使用天数
PASS_WARN_AGE 7 #新建用户的密码到期提前提醒天数

# 方式二
chage -m 0 -M 90 -E 2019-09-01 -W 7 <用户名>
# 表示将此用户的密码最长使用天数设为30，最短使用天数设为0，密码2019年9月1日过期，过期前七天警告用户。

# 设置连续输错三次密码，账号锁定五分钟
# vim /etc/pam.d/common-auth
添加 auth required pam_tally.so onerr=fail deny=3 unlock_time=300
```

3. 限制用户su

```
# vim vi /etc/pam.d/su
# 只允许k1ea4c组用户su到root
添加 auth required pam_wheel.so group=k1ea4c
```

4. SSH安全配置

```
# vim /etc/ssh/sshd_config

# 不允许root账号直接登录系统。

设置 PermitRootLogin 的值为 no

# 修改SSH使用的协议版本。

设置 Protocol 的版本为 2

# 修改允许密码错误次数（默认6次）。

设置 MaxAuthTries 的值为 3
```

## 5. 设置umask值

```
# vim /etc/profile

添加 umask 027
source /etc/profile
```

## Windows Serv 2012加固

### 1. 账号及安全策略


```
# 运行 - secpol.msc （管理工具）

# 账号策略

密码必须符合复杂性要求：启用
密码长度最小值 8个字符
密码最长使用期限： 30天
强制密码历史： 3个记住的密码
# 账号锁定

帐户锁定阈值： 3次无效登陆
帐户锁定时间： 30分钟
复位帐户锁定计数器： 30分钟之后


# cmd - gpupdate /force （立即生效）
```

5d3be5761bad768696

5d3be57dc8db561335

### 2. 禁用Guest账户权限

```
# 运行 - compmgmt.msc （计算机管理）  
  
# 快捷方式: net user guest /active:no
```

 5d3be602f307719357

### 3. Administrator账号、组重命名

```
# 运行 - compmgmt.msc （计算机管理）  
# 本地用户和组 - 用户 - 重命名Administrator  
# 本地用户和组 - 组 - 重命名Administrators
```


### 4. 日志及审核策略（方便寻找问题产生的根源）

审核策略更改 成功，失败  
审核登陆事件 成功，失败  
审核对象访问 失败  
审核过程跟踪 无审核  
审核目录服务访问 失败  
审核特权使用 失败  
审核系统事件 成功，失败  
审核账户登陆事件 成功，失败  
审核帐户管理 成功，失败

 5d3be6be9cf2497995



### 5. 调整事件日志的大小及覆盖策略（避免容量过小导致重要日志泄露）

```
# 运行 - eventvwr.msc （事件查看器）  
  
日志类型 日志大小  
  
应用程序 80000KB  
  
安全日志 80000KB  
  
系统日志 80000KB
```

 5d3be822ef2e089188


### 6. 安全选项策略设置

```
# 运行 - secpol.msc - 本地策略 - 安全选项设置
```

 5d3becfaa964873775 5d3bed025256291018 5d3bed08b1b2540314

## 7. 删除系统默认共享

```
# cmd - net share 查看默认共享 - net share <共享名> /del
```

 5d3bef282acd091726

## 8. 修改默认3389远程端口


```
# 运行 - regedit (注册表)
```

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal  
Server\Wds\rdpwd\Tds\tcp\PortNumber
```

 5d3bf090d671117150

## 9. 禁用135, 139, 445端口

```
# 管理工具 - 高级安全Windows防火墙
```

 5d3bf728a27a777821

## Nginx加固



1. Nginx默认是不允许列出整个目录的, 但是还需确认Nginx是不允许列出整个目录的, 以免被拉取整个网站的代码

```
# autoindex为off或者并未添加  
http {  
    autoindex off;  
}
```

2. Nginx默认是会在返回的数据包中显示版本号, 为了防止攻击者针对版本进行攻击, 因此对其版本号进行隐藏

```
http {  
    server_tokens off;  
}
```




5d33fb64e256b207225d33fc168648924488

### 3. 自定义缓存，限制缓冲区溢出攻击

```
http{
... ..
    server{
... ..
        client_body_buffer_size 16K;
        client_header_buffer_size 1k;
        client_max_body_size 1m;
        large_client_header_buffers 4 8k;
```

### 4. 设置timeout设低来防御DOS攻击

```
http {
... ..
    client_body_timeout 10;
    client_header_timeout 30;
    keepalive_timeout 30 30;
    send_timeout 10;
```

5d33ff22613da48792

### 5. 在目前的应用系统中值使用到POST和GET方法，所以除了它们外，其他方式的请求均可拒绝

```
server{
... ..
    if($request_method !~ ^(GET|HEAD|POST)$) {return404;}
... ..
```

### 6. 限制访问IP或者IP段

```
location/ {
    deny 192.168.1.1;
    allow 192.168.1.0/24;
    allow 10.1.1.0/16;
    allow 2001:0db8::/32;
    deny all;
}
```

# 注：规则按照顺序依次检测，直到匹配到第一条规则。 在这个例子里，  
# IPv4的网络中只有 10.1.1.0/16 和 192.168.1.0/24允许访问，但 192.168.1.1

```
# 除外，对于IPv6的网络，只有2001:0db8::/32允许访问。
```

## 6. 日志配置

```
http {
    .....
    log_format main '$remote_addr - $remote_user [$time_local] "$request" '$status
    $body_bytes_sent
    "$http_referer" '$http_user_agent' "$http_x_forwarded_for";
    access_log logs/ access.log main;
```

## 7. 限制并发和速度

```
location / {
    .....
    limit_conn one 1;
    limit_rate 20k;
    .....
}
```

# Apache加固

## 1. 低权限用户启动

```
# vim /etc/apache2/envvars


User www-data

Group www-data
```

## 2. 隐藏Apache banner信息

```
# vim /etc/apache2/conf-enabled/security.conf



# 在出现错误页的时候不显示服务器操作系统的名称
ServerTokens OS 修改为: ServerTokens Prod
# 不回显apache版本信息
ServerSignature On 修改为: ServerSignature Off
```

 5d36b3e29469443729 5d36b3eb4593725830

### 3. 禁止目录遍历

```
# vim /etc/apache2/apache2.conf


<Directory /var/www/html/k1ea4c>
    Options Indexes FollowSymLinks 修改为 Options FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

5d36b7930628c964785d36b799c0d0c57598

### 4. 限制IP访问

```
# vim /etc/apache2/apache2.conf

<Directory /var/www/html/k1ea4c>
    Options FollowSymLinks
    AllowOverride None
    Require all granted
    Order deny,allow
    Deny from 192.168.44.1
</Directory>
```

5d36bdf854710165425d36bdfd96fc851091

### 5. 限制禁止访问的文件夹，例如后台目录

```
# vim /etc/apache2/apache2.conf

<Directory /var/www/html/k1ea4c>
    Deny from all
</Directory>
```

### 6. 如果网站需要上传目录，应限制上传目录无脚本执行权限

```
# vim /etc/apache2/apache2.conf

<Directory "/var/www/html/upload">
    AllowOverride None
    <Files ~ "\.php">
        Order Allow,Deny
```

```
        Deny from all
    </Files>
    ....
</Directory>
```

## 7. 自定义错误页面

```
# vim /etc/apache2/conf-enabled/localized-error-pages.conf

ErrorDocument 400 /custom400.html
ErrorDocument 401 /custom401.html
ErrorDocument 403 /custom403.html
ErrorDocument 404 /custom404.html
ErrorDocument 405 /custom405.html
ErrorDocument 500 /custom500.html # Customxxx.html 为要设置的错误页面
```

## 8. 关闭 Trace

*TRACE和TRACK是用来调试web服务器连接的HTTP方式。支持该方式的服务器存在跨站脚本漏洞，通常在描述各种浏览器缺陷的时候，把"Cross-Site-Tracing"简称为XST。攻击者可以利用此漏洞欺骗合法用户并得到他们的私人信息。*

[XST学习资料]([https://blog.csdn.net/ocean\\_001/article/details/95474281](https://blog.csdn.net/ocean_001/article/details/95474281))

```
# vim /etc/apache2/conf-enabled/security.conf

TraceEnable Off
```

## 9. 禁止apache解析index.php.jpg文件，防止apache的文件解析漏洞（[Apache多后缀解析漏洞] ([https://github.com/vulhub/vulhub/tree/master/httpd/apache\\_parsing\\_vulnerability](https://github.com/vulhub/vulhub/tree/master/httpd/apache_parsing_vulnerability))

```
# vim /etc/apache2/apache2.conf

<Directory /var/www/html/k1ea4c/>
    # 如果运维人员给.php后缀增加了处理器，在有多个后缀的情况下，只要一个文件含有.php
    后缀的文件即将被识别成PHP文件，没必要是最后一个后缀。利用这个特性，将会造成一个可以绕过上
    传白名单的解析漏洞。
    AddHandler application/x-httpd-php .php

    Options FollowSymLinks
    AllowOverride None
    Require all granted
    Order deny,allow
    Deny from 192.168.44.1
</Directory>


# 添加以下内容
```


```
<FilesMatch "(.php|.php3|.php4|.php5.)">

    Order Deny,Allow

    Deny from all

</FilesMatch>
```

 5d3830e51594649461

 5d3830f38c8b928058

## 10. 拒绝服务防范

```
# vim /etc/apache2/apache2.conf
# 此处为一建议值，具体的设定需要根据现实情况

Timeout 10      #客户端与服务器端建立连接前的时间间隔
KeepAlive On
KeepAliveTimeout 15 # 限制每个 session 的保持时间是 15 秒
```

## 11. 禁用CGI

```
# vim /etc/apache2/apache2.conf

<Directory /var/www/html/k1ea4c/>
    AddHandler application/x-httpd-php .php
    # 我们在Indexes前面加上了'-' ,意思就是禁止目录遍历,防止敏感文件泄露,此项非常重要,另外,关闭CGI,SSI,以及Follow Symbolic Links
    Options -Indexes -Includes -ExecCGI -FollowSymLinks
    AllowOverride None
    Require all granted
    Order deny,allow
    Deny from 192.168.44.1
</Directory>
```

## 12. php.ini 配置

```
magic_quotes_gpc = Off 改为 magic_quotes_gpc = On # 防止SQL注入
allow_url_fopen = off    # 防止远程包含漏洞
```

[参考资料I](<https://klionsec.github.io/2017/11/26/apache-sec/>), [参考资料II](<https://www.alibabacloud.com/help/zh/faq-detail/52981.htm?spm=a2c63.q38357.a3.1.6cc952eaDXdWoN>)

## MySQL加固

### 1. 删除默认数据库和数据库用户

```
mysql> show databases;
mysql> drop database test; //删除数据库test
mysql> use mysql;
mysql> delete from db; //删除存放数据库的表信息，因为还没有数据库信息。
mysql> delete from user where not (user='root'); // 删除初始非root的用户
mysql> delete from user where user='root' and password=''; //删除空密码的root尽量重复操作
mysql> flush privileges; //强制刷新内存授权表。
```

### 2. 改变默认管理员的名字（root）

```
mysql> update mysql.user set user='admin' where user='root';
```

### 3. 使用独立用户运行MySQL

```
# vim /etc/mysql/mysql.conf.d/mysqld.cnf

[mysqld]
user=mysql
```

### 4. 禁止远程连接数据库

```
# vim /etc/mysql/mysql.conf.d/mysqld.cnf

取消注释 bind-address = 127.0.0.1
```

### 5. 限制用户连接量

```
# vim /etc/mysql/mysql.conf.d/mysqld.cnf

max_connections = 2
```

### 6. 数据库备份

```
mysqldump -u 用户名 -p 密码 数据库名 > back.sql # 备份指定数据库

mysql -u 用户名 -p 密码 数据库名 < bak.sql # 还原指定数据库
```



## Tomcat加固

### 1. 删除tomcat应用的文档和实例程序


```
cd /opt/apache-tomcat-9.0.22/webapps/  
rm -rf docs/ examples/
```

### 2. 为所有tomcat用户设置复杂密码


```
# 小知识  
manager-gui  
允许访问html接口(即URL路径为/manager/html/*)  
manager-script  
允许访问纯文本接口(即URL路径为/manager/text/*)  
manager-jmx  
允许访问JMX代理接口(即URL路径为/manager/jmxproxy/*)  
manager-status  
允许访问Tomcat只读状态页面(即URL路径为/manager/status/*)  
  
# vim /opt/apache-tomcat-9.0.22/conf/tomcat-users.xml
```

5d3c0d59ac009449015d3c0d8dedce9826175d3c0da838b4f59595

### 3. 设置SHUTDOWN字符串，防止恶意用户进行暴力破解


5d3c0e1c6b453210675d3c0e4086f5540911

```
# vim /opt/apache-tomcat-9.0.22/conf/server.xml
```

5d3c0ec13153191049


### 4. 以普通用户启动tomcat

```
useradd tomcat  
  
chown -R tomcat.tomcat /opt/apache-tomcat-9.0.22/  
su -c /opt/apache-tomcat-9.0.22/bin/starter.sh tomcat
```

5d3c36602243922472

## 0x04 最终目标

```
// 测试php连接MySQL
<?php
    $uid = $_GET['id'];
    $sql = "SELECT * FROM userinfo where id = $uid";
    $conn = mysql_connect('localhost', 'root', 'root');
    mysql_select_db("test", $conn);
    $result = mysql_query($sql, $conn);
    print_r('当前语句: '.$sql.'<br />结果: ');
    print_r(mysql_fetch_row($result));
    mysql_close()
?>
```

5d33f6119553789735