培训教育
Training Services

WEB应用服务器

配置及安全应用

主讲：高天

课程大纲

- Apache服务器安全加固
- Nginx 服务器安全加固
- Tomcat安全加固
- Web应用服务器加固思路

Apache服务安全加固

一.账号设置

- 以专门的用户帐号和组运行 Apache。
- 根据需要为 Apache 创建用户、组
- 参考配置操作 如果没有设置用户和组，则新建用户，并在 Apache 配置文件中指定
- (1) 创建 apache 组：groupadd apache
- (2) 创建 apache 用户并加入 apache 组：useradd apache –g apache
- (3) 将下面两行加入 Apache 配置文件 httpd.conf 中

```
1. User apache
2. Group apache
```

- 检查 httpd.conf 配置文件。 检查是否使用非专用账户(如 root)运行 apache
- 默认一般符合要求，Linux下默认apache或者nobody用户，Unix默认为daemon用户

Apache服务安全加固

授权设置

- 严格控制Apache主目录的访问权限，非超级用户不能修改该目录中的内容
- Apache 的 主目录对应于 Apache Server配置文件 httpd.conf 的Server Root控制项中应为：

```
1.    “Server Root /usr/local/apache”
```

- 判定条件
- 非超级用户不能修改该目录中的内容
- 检测操作
- 尝试修改，看是否能修改
- 一般为/etc/httpd目录，默认情况下属主为root:root，其它用户不能修改文件，默认一般符合要求
- 严格设置配置文件和日志文件的权限，防止未授权访问。
- chmod 600 /etc/httpd/conf/httpd.conf”设置配置文件为属主可读写，其他用户无权限。
- 使用命令”chmod 644 /var/log/httpd/*.log”设置日志文件为属主可读写，其他用户只读权限。
- /etc/httpd/conf/httpd.conf默认权限是644，可根据需要修改权限为600。
- /var/log/httpd/*.log默认权限为644，默认一般符合要求。

Apache服务安全加固

日志设置

- 设备应配置日志功能，对运行错误、用户访问等进行记录，记录内容包括时间，用户使用的 IP 地址等内容。
- 编辑 httpd.conf 配置文件，设置日志记录文件、记录内容、记录 格式。　　其中，错误日志：

```
1. LogLevel notice #日志的级别
2. ErrorLog /…/logs/error_log #日志的保存位置(错误日志)
3. 访问日志：
4. LogFormat %h %l %u %t \"%r\" %>s %b "%{Accept}i\"%{Referer}i\" \"%{User-Agent}i\""
5. combined
6. CustomLog /…/logs/access_log combined (访问日志)
```

- ErrorLog 指令设置错误日志文件名和位置。错误日志是最重要的 日志文件，
- Apache httpd 将在这个文件中存放诊断信息和处理请 求中出现的错误。
- 若要将错误日志送到 Syslog，则设置： ErrorLog syslog。
- CustomLog 指令指定了保存日志文件的具体位置以及日志的格式。访问日志中会记录服务器所处理的所有请求。
- LogFormat 设置日志格式，建议设置为 combined 格式。
- LogLevel 用于调整记录在错误日志中的信息的详细程度，建议设置为notice。
- 日志的级别，默认是warn，notice级别比较详细，在实际中由于日志会占用大量硬盘空间，一般没有设置

Apache服务安全加固

➢ 禁止访问外部文件

- ➢ 禁止 Apache 访问 Web 目录之外的任何文件。
- ➢ 参考配置操作
- ➢ 编辑 httpd.conf 配置文件：

```
1.    Order Deny,Allow
2.    Deny from all
```

- ➢ 设置可访问目录

```
1.    Order Allow,Deny
2.    Allow from all
```

- ➢ 其中/web 为网站根目录
- ➢ 默认配置如下:

```
1.    Options FollowSymLinks
2.    AllowOverride None
```

- ➢ 一般可根据业务需要设置

Apache服务安全加固

禁止目录列出

目录列出会导致明显信息泄露或下载，禁止 Apache 列表显示文件,编辑 httpd.conf 配置文件：

```
1.    #Options Indexes FollowSymLinks #删掉Indexes
2.      Options FollowSymLinks
3.    AllowOverride None
4.    Order allow,deny
5.    Allow from all
```

- 将Options Indexes FollowSymLinks 中的 Indexes 去掉，就可以禁 止 Apache 显示该目录结构。 Indexes 的作用就是当该目录下没 有 index.html 文件时，就显示目录结构。
- 重新启动 Apache 服务
- 可以设置 /etc/httpd/httpd.conf 段中删除Options的Indexes设置
- 一般可根据业务需要设置

Apache服务安全加固

错误页面重定向

Apache 错误页面重定向功能防止敏感信息泄露

修改 httpd.conf 配置文件：

```
1.        ErrorDocument 400 /custom400.html
2.    ErrorDocument 401 /custom401.html
3.    ErrorDocument 403 /custom403.html
4.    ErrorDocument 404 /custom404.html
5.    ErrorDocument 405 /custom405.html
6.    http://www.013188.com
7.    ErrorDocument 500 /custom500.html
8.  注：Customxxx.html 为要设置的错误页面。
```

- 重新启动 Apache 服务
- 此项需要应用系统设有错误页面，或者不在httpd中设置完全由业务逻辑实现，可根据业务需求加固。

Apache服务安全加固

拒绝服务防范

- 根据业务需要，合理设置 session 时间，防止拒绝服务攻击
- 编辑 httpd.conf 配置文件:

```
1.      Timeout 10 #客户端与服务器端建立连接前的时间间隔
2.      KeepAlive On
3.      KeepAliveTimeout 15 #限制每个 session 的保持时间是 15 秒 注：此处为一建议值，具体的设定需要根据现实情况。
```

- 重新启动 Apache 服务
- 默认Timeout 120 KeepAlive Off，KeepAliveTimeout 15，该项设置涉及性能调整，一般不做。

Apache服务安全加固

- **隐藏 Apache 的版本号**
- 隐藏 Apache 的版本号及其它敏感信息。

- 1.配置操作

- 修改 httpd.conf 配置文件：

```
1. ServerSignature Off ServerTokens Prod
```

Apache服务安全加固

➢ 关闭 TRACE功能

- 关闭 TRACE，防止 TRACE 方法被访问者恶意利用。
- 配置修改vim /etc/httpd/conf/httpd.conf

```
1.    添加 “TraceEnable Off”
2.    注：适用于 Apache 2.0 以上版本
```

Apache服务安全加固

➢ 禁用 CGI

- ➢ 如果服务器上不需要运行 CGI 程序，建议禁用 CGI
- ➢ 1.修改配置vim /etc/httpd/conf/httpd.conf，把 cgi-bin 目录的配置和模块都注释掉

```
#LoadModule cgi_module modules/mod_cgi.so
#ScriptAlias /cgi-bin/ “/var/www/cgi-bin/”
#
#AllowOverride None
# Options None
#Order allow,deny
#Allow from all
#
```

- ➢ .根据需要设置，如果没有CGI程序，可以关闭
- ➢

Apache服务安全加固

- **监听地址绑定**
- 服务器有多个 IP 地址时，只监听提供服务的 IP 地址
- 1.使用命令查看是否绑定IP地址

```
1. cat /etc/httpd/conf/httpd.conf|grep Listen
```

- 修改配置vim /etc/httpd/conf/httpd.conf 修改

```
1.     Listen x.x.x.x:80
```

- 默认设置是Listen 80监听所有地址，如果服务器只有一个IP地址可不做该项设置，如果有多个IP可以按照需要设

Apache服务安全加固

➢ 删除缺省安装的无用文件

➢ 删除缺省安装的无用文件.

参考配置操作删除缺省 HTML 文件

```
# rm -rf /usr/local/apache2/htdocs/*
```

删除缺省的CGI脚本

```
# rm -rf /usr/local/apache2/cgi-bin/*
```

删除 Apache 说明文件

```
# rm -rf /usr/local/apache2/manual
```

删除源代码文件

```
1.          # rm -rf /path/to/httpd-2.2.4*
2.          根据安装步骤不同和版本不同，某些目录或文件可能不存在或位置不同。
```

删除CGI

可根据实际情况删除，一般是 /var/www/html /var/www/cgi-bin 默认就是空的

Apache服务安全加固

➢ 禁用非法 HTTP 方法

- ➢ 禁用PUT、DELETE等危险的HTTP 方法.
- ➢ 编辑 httpd.conf 文件,只允许 get、post 方法

```
<Location />
<LimitExcept GET POST CONNECT OPTIONS>
  Order Allow,Deny
  Deny from all
</LimitExcept>
</Location>
```

- ➢ 根据需要可设置，如果没有不需要用到put delete HTTP 方法的话，加在/etc/httpd/conf/httpd.conf的段中。
- ➢

Nginx安装及服务控制3-1

- 编译安装nginx(使用CentOS 5.8)
  - 依赖包：pcre-devel、zlib-devel
  - 默认以nobody身份运行，建议使用专有账号nginx

```
[root@localhost ~]# yum -y install pcre-devel zlib-devel
[root@localhost ~]# useradd -M -s /sbin/nologin nginx

[root@localhost ~]# tar zxf nginx-1.0.8.tar.gz
[root@localhost ~]# cd nginx-1.0.8
[root@localhost nginx-1.0.8]# ./configure --prefix=/usr/local/nginx --
user=nginx --group=nginx --with-http_stub_status_module
[root@localhost nginx-1.0.8]# make
[root@localhost nginx-1.0.8]# make install

[root@localhost nginx-1.0.8]# ln -s /usr/local/nginx/sbin/nginx
/usr/local/sbin/
```

启用状态统计模块

建立符号链接

Nginx安装及服务控制3-2

➢ Nginx的运行控制

```
[root@localhost ~]# nginx -t
nginx: the configuration file /usr/local/nginx/conf/nginx.conf syntax is ok
nginx: configuration file /usr/local/nginx/conf/nginx.conf test is successful

[root@localhost ~]# nginx

[root@localhost ~]# netstat -anpt | grep nginx
tcp    0    0 0.0.0.0:80      0.0.0.0:*       LISTEN      26088/nginx: master

[root@localhost ~]# cat /usr/local/nginx/logs/nginx.pid
22378
[root@localhost ~]# kill -s QUIT 22378
```

等同于 killall -s QUIT nginx

Nginx安装及服务控制3-3

- 添加nginx服务

```
[root@localhost ~]# vi /etc/init.d/nginx
#!/bin/bash
# chkconfig: - 99 20
……
case "$1" in
  start)
    /usr/local/nginx/sbin/nginx
    ;;
  stop)
    kill -s QUIT $(cat /usr/local/nginx/logs/nginx.pid)
    ;;
……

[root@localhost ~]# chmod +x /etc/init.d/nginx
[root@localhost ~]# chkconfig --add nginx
```

nginx.conf配置文件 2-1

- 全局配置、I/O事件配置

```
[root@localhost ~]# vi /usr/local/nginx/conf/nginx.conf
……
#user  nobody;
worker_processes  1;

#error_log  logs/error.log;
#pid        logs/nginx.pid;
……

events {
    use epoll;
    worker_connections  4096;
}
……
```

工作进程数量

日志文件位置

I/O事件模型

每进程连接数

nginx.conf配置文件 2-2

- HTTP监听配置

```
[root@localhost ~]# vi /usr/local/nginx/conf/nginx.conf
……
http {
    access_log  logs/access.log  main;
    sendfile        on;
    keepalive_timeout  65;
    ……
    server {
        listen       80;
        server_name  www.benet.com;
        charset utf-8;
        location / {
            root   html;
            index  index.html index.php;
        }
    }
}
```

监听地址及端口

网站根目录位置

Nginx的访问状态统计

- 启用统计页面

```
location ~ /status {
        stub_status    on;
        access_log off;
}
```

- 访问统计页面
  - http://YourServer/status

小结

- 请思考：
  - Nginx与Apache相比，具有哪些优势?
  - 如何启动、关闭Nginx服务程序?
  - 如何修改Nginx服务器的监听地址、网站目录?

虚拟Web主机2-1

- 基于域名的虚拟主机

```
http {
    ……
    server {
        server_name  www.benet.com;
        location / {
            root   /var/www/benet;
            index  index.html index.php;
        }
    }
    server {
        server_name  www.accp.com;
        location / {
            root   /var/www/accp;
            index  index.html index.php;
        }
    }
}
```

第1个虚拟主机

第2个虚拟主机

虚拟Web主机2-2

- 基于IP地址的虚拟主机

```
http {
    ……
    server {
        listen        192.168.4.11:80;
        server_name  www.benet.com;
        ……
    }

    server {
        listen        192.168.4.22:80;
        server_name  www.accp.com;
        ……
    }
}
```

如果直接通过IP地址访问，各虚拟主机的域名也可以相同

Nginx 服务器安全加固

- **默认配置文件和Nginx端口**
- /usr/local/nginx/conf/ – Nginx配置文件目录，/usr/local/nginx/conf/nginx.conf是主配置文件
- /usr/local/nginx/html/ – 默认网站文件位置
- /usr/local/nginx/logs/ – 默认日志文件位置
- Nginx HTTP默认端口 : TCP 80
- Nginx HTTPS默认端口: TCP 443
- 你可以使用以下命令来测试Nginx配置文件准确性。
- /usr/local/nginx/sbin/nginx -t
- 将会输出:
  the configuration file /usr/local/nginx/conf/nginx.conf syntax is ok
  configuration file /usr/local/nginx/conf/nginx.conf test is successful
  执行以下命令来重新加载配置文件。
- /usr/local/nginx/sbin/nginx -s reload
- 执行以下命令来停止服务器。
- /usr/local/nginx/sbin/nginx -s stop
-

Nginx 服务器安全加固

- **一、配置SELinux**
- **注意：对于云服务器 ECS，参阅** *ECS 使用须知* **，基于兼容性、稳定性考虑，请勿开启 SELinux。**
- 安全增强型 Linux（SELinux）的是一个Linux内核的功能，它提供支持访问控制的安全政策保护机制。它可以大部分的攻击。下面我们来看如何启动基于centos/RHEL系统的SELinux。
- **安装SELinux**
- rpm -qa | grep selinux
- libselinux-1.23.10-2
  selinux-policy-targeted-1.23.16-6
  如果没有返回任何结果，代表没有安装 SELinux，如果返回了类似上面的结果，则说明系统安装了SELinux。
- **布什值锁定**
  运行命令getsebool -a来锁定系统。
- getsebool -a | less
- getsebool -a | grep off
- getsebool -a | grep o

Nginx 服务器安全加固

- **通过分区挂载允许最少特权**
- 服务器上的网页/html/php文件单独分区。例如，新建一个分区/dev/sda5(第一逻辑分区)，并且挂载在/nginx。确保 /nginx是以noexec, nodev and nosetuid的权限挂载。以下是我的/etc/fstab的挂载/nginx的信息：
  LABEL=/nginx /nginx ext3 defaults,nosuid,noexec,nodev 1 2
  注意：你需要使用fdisk和mkfs.ext3命令创建一个新分区。

Nginx 服务器安全加固

配置/etc/sysctl.conf强化Linux安全

- 你可以通过编辑/etc/sysctl.conf来控制和配置Linux内核、网络设置。
- # Avoid a smurf attack
- net.ipv4.icmp_echo_ignore_broadcasts = 1
- # Turn on protection for bad icmp error messages
- net.ipv4.icmp_ignore_bogus_error_responses = 1
- # Turn on syncookies for SYN flood attack protection
- net.ipv4.tcp_syncookies = 1
- # Turn on and log spoofed, source routed, and redirect packets
- net.ipv4.conf.all.log_martians = 1
- net.ipv4.conf.default.log_martians = 1
- # No source routed packets here
- net.ipv4.conf.all.accept_source_route = 0
- net.ipv4.conf.default.accept_source_route = 0

Nginx 服务器安全加固

配置/etc/sysctl.conf强化Linux安全

- 你可以通过编辑/etc/sysctl.conf来控制和配置Linux内核、网络设置。
- # Make sure no one can alter the routing tables
- net.ipv4.conf.all.accept_redirects = 0
- net.ipv4.conf.default.accept_redirects = 0
- net.ipv4.conf.all.secure_redirects = 0
- net.ipv4.conf.default.secure_redirects = 0
- # Don’ t act as a router
- net.ipv4.ip_forward = 0
- net.ipv4.conf.all.send_redirects = 0
- net.ipv4.conf.default.send_redirects = 0
- # Turn on execshild
- kernel.exec-shield = 1
- kernel.randomize_va_space = 1

Nginx 服务器安全加固

配置/etc/sysctl.conf强化Linux安全

- 你可以通过编辑/etc/sysctl.conf来控制和配置Linux内核、网络设置。
- # Turn on execshild
- kernel.exec-shield = 1
- kernel.randomize_va_space = 1
- # Tuen IPv6
- net.ipv6.conf.default.router_solicitations = 0
- net.ipv6.conf.default.accept_ra_rtr_pref = 0
- net.ipv6.conf.default.accept_ra_pinfo = 0
- net.ipv6.conf.default.accept_ra_defrtr = 0
- net.ipv6.conf.default.autoconf = 0
- net.ipv6.conf.default.dad_transmits = 0
- net.ipv6.conf.default.max_addresses = 1
- # Optimization for port usefor LBs
- # Increase system file descriptor limit
- fs.file-max = 65535

Nginx 服务器安全加固

配置/etc/sysctl.conf强化Linux安全

- 你可以通过编辑/etc/sysctl.conf来控制和配置Linux内核、网络设置。
- # Allow for more PIDs (to reduce rollover problems); may break some programs 32768
- kernel.pid_max = 65536
- # Increase system IP port limits
- net.ipv4.ip_local_port_range = 2000 65000
- # Increase TCP max buffer size setable using setsockopt()
- net.ipv4.tcp_rmem = 4096 87380 8388608
- net.ipv4.tcp_wmem = 4096 87380 8388608
- # Increase Linux auto tuning TCP buffer limits
- # min, default, and max number of bytes to use
- # set max to at least 4MB, or higher if you use very high BDP paths
- # Tcp Windows etc
- net.core.rmem_max = 8388608
- net.core.wmem_max = 8388608
- net.core.netdev_max_backlog = 5000
- net.ipv4.tcp_window_scaling = 1
-

Nginx 服务器安全加固

- **删除所有不需要的Nginx模块**
- 你需要直接通过编译Nginx源代码使模块数量最少化。通过限制只允许web服务器访问模块把风险降到最低。你可以只配置安装nginx你所需要的模块。例如，禁用SSL和autoindex模块你可以执行以下命令：
- ./configure –without-http_autoindex_module –without-http_ssi_module
- make
- make install
- 通过以下命令来查看当编译nginx服务器时哪个模块能开户或关闭：
- ./configure –help | less
- 禁用你用不到的nginx模块。
  （可选项）更改nginx版本名称。
  编辑文件/http/ngx_http_header_filter_module.c：
- vi +48 src/http/ngx_http_header_filter_module.c
- 找到行：
- static char ngx_http_server_string[] = “Server: nginx” CRLF;
- static char ngx_http_server_full_string[] = “Server: ” NGINX_VER CRLF;
- 按照以下行修改：
- static char ngx_http_server_string[] = “Server: Ninja Web Server” CRLF;
- static char ngx_http_server_full_string[] = “Server: Ninja Web Server” CRLF;
- 保存并关闭文件。现在你可以编辑服务器了。增加以下代码到nginx.conf文件来关闭nginx版本号的显示。
- server_tokens off

Nginx 服务器安全加固

- **使用mod_security(只适合后端Apache服务器)**
- mod_security为Apache提供一个应用程序级的防火墙。为后端Apache Web服务器安装mod_security，这会阻止很多注入式攻击。

- **安装SELinux策略以强化Nginx Web服务器**
- 默认的SELinux不会保护Nginx Web服务器，但是你可以安装和编译保护软件。
  1、安装编译SELinux所需环境支持
- yum -y install selinux-policy-targeted selinux-policy-devel
- 2、下载SELinux策略以强化Nginx Web服务器。
- cd /opt
- wget ‘http://downloads.sourceforge.net/project/selinuxnginx/se-ngix_1_0_10.tar.gz?use_mirror=nchc’
- 3、解压文件
- tar -zxvf se-ngix_1_0_10.tar.gz

Nginx 服务器安全加固

安装SELinux策略以强化Nginx Web服务器

- 4、编译文件
- cd se-ngix_1_0_10/nginx
- make
- 将会输出如下：
  Compiling targeted nginx module
  /usr/bin/checkmodule: loading policy configuration from tmp/nginx.tmp
  /usr/bin/checkmodule: policy configuration loaded
  /usr/bin/checkmodule: writing binary representation (version 6) to tmp/nginx.mod
  Creating targeted nginx.pp policy package
- rm tmp/nginx.mod.fc tmp/nginx.mod
- 5、安装生成的nginx.pp SELinux模块：
- /usr/sbin/semodule -i nginx.pp

Nginx 服务器安全加固

➢ 基于Iptables防火墙的限制

- 下面的防火墙脚本阻止任何除了允许：
- 来自HTTP(TCP端口80)的请求
- 来自ICMP ping的请求
- ntp(端口123)的请求输出
- smtp(TCP端口25)的请求输出
- #!/bin/bash
- IPT=”/sbin/iptables”
- #### IPS ######
- # Get server public ip
- SERVER_IP=$(ifconfig eth0 | grep ‘inet addr:’ | awk -F’inet addr:’ ‘{ print $2}’ | awk ‘{ print $1}’)
- LB1_IP=”204.54.1.1”
- LB2_IP=”204.54.1.2”
- # Do some smart logic so that we can use damm script on LB2 too
- OTHER_LB=”"
- SERVER_IP=”"
- [["$SERVER_IP" == "$LB1_IP"]] && OTHER_LB=”$LB2_IP” || OTHER_LB=”$LB1_IP”
- [["$OTHER_LB" == "$LB2_IP"]] && OPP_LB=”$LB1_IP” || OPP_LB=”$LB2_IP”
- ### IPs ###
- PUB_SSH_ONLY=”122.xx.yy.zz/29”
- #### FILES #####
- BLOCKED_IP_TDB=/root/.fw/blocked.ip.txt

Nginx 服务器安全加固

➢ 基于Iptables防火墙的限制

- ### IPs ###
- PUB_SSH_ONLY=" 122.xx.yy.zz/29"
- #### FILES #####
- BLOCKED_IP_TDB=/root/.fw/blocked.ip.txt
- SPOOFIP=" 127.0.0.0/8 192.168.0.0/16 172.16.0.0/12 10.0.0.0/8 169.254.0.0/16 0.0.0.0/8 240.0.0.0/4 255.255.255.255/32 168.254.0.0/16 224.0.0.0/4 240.0.0.0/5 248.0.0.0/5 192.0.2.0/24"
- BADIPS=$([[-f ${BLOCKED_IP_TDB}]] && egrep -v " ^#|^$" ${BLOCKED_IP_TDB})
- ### Interfaces ###
- PUB_IF=" eth0" # public interface
- LO_IF=" lo" # loopback
- VPN_IF=" eth1" # vpn / private net
- ### start firewall ###
- echo "Setting LB1 $(hostname) Firewall..."
- # DROP and close everything
- $IPT -P INPUT DROP
- $IPT -P OUTPUT DROP
- $IPT -P FORWARD DROP
- # Unlimited lo access
- $IPT -A INPUT -i ${LO_IF} -j ACCEPT
- $IPT -A OUTPUT -o ${LO_IF} -j ACCEPT
- ### Unlimited vpn / net access

Nginx 服务器安全加固

➢ 基于Iptables防火墙的限制

- # Unlimited lo access
- $IPT -A INPUT -i ${LO_IF} -j ACCEPT
- $IPT -A OUTPUT -o ${LO_IF} -j ACCEPT
- # Unlimited vpn / pnet access
- $IPT -A INPUT -i ${VPN_IF} -j ACCEPT
- $IPT -A OUTPUT -o ${VPN_IF} -j ACCEPT
- # Drop sync
- $IPT -A INPUT -i ${PUB_IF} -p tcp ! –syn -m state –state NEW -j DROP
- # Drop Fragments
- $IPT -A INPUT -i ${PUB_IF} -f -j DROP
- $IPT -A INPUT -i ${PUB_IF} -p tcp –tcp-flags ALL FIN,URG,PSH -j DROP
- $IPT -A INPUT -i ${PUB_IF} -p tcp –tcp-flags ALL ALL -j DROP
- # Drop NULL packets
- $IPT -A INPUT -i ${PUB_IF} -p tcp –tcp-flags ALL NONE -m limit –limit 5/m –limit-burst 7 -j LOG –log-prefix ” NULL Packets “
- $IPT -A INPUT -i ${PUB_IF} -p tcp –tcp-flags ALL NONE -j DROP
- $IPT -A INPUT -i ${PUB_IF} -p tcp –tcp-flags SYN,RST SYN,RST -j DROP
- # Drop XMAS
- $IPT -A INPUT -i ${PUB_IF} -p tcp –tcp-flags SYN,FIN SYN,FIN -m limit –limit 5/m –limit-burst 7 -j LOG –log-prefix ” XMAS Packets “
- $IPT -A INPUT -i ${PUB_IF} -p tcp –tcp-flags SYN,FIN SYN,FIN -j DROP

Nginx 服务器安全加固

➢ 基于Iptables防火墙的限制

- # Drop XMAS
- $IPT -A INPUT -i ${PUB_IF} -p tcp –tcp-flags SYN,FIN SYN,FIN -m limit –limit 5/m –limit-burst 7 -j LOG –log-prefix " XMAS Packets "
- $IPT -A INPUT -i ${PUB_IF} -p tcp –tcp-flags SYN,FIN SYN,FIN -j DROP
- # Drop FIN packet scans
- $IPT -A INPUT -i ${PUB_IF} -p tcp –tcp-flags FIN,ACK FIN -m limit –limit 5/m –limit-burst 7 -j LOG –log-prefix " Fin Packets Scan "
- $IPT -A INPUT -i ${PUB_IF} -p tcp –tcp-flags FIN,ACK FIN -j DROP
- $IPT -A INPUT -i ${PUB_IF} -p tcp –tcp-flags ALL SYN,RST,ACK,FIN,URG -j DROP
- # Log and get rid of broadcast / multicast and invalid
- $IPT -A INPUT -i ${PUB_IF} -m pkttype –pkt-type broadcast -j LOG –log-prefix " Broadcast "
- $IPT -A INPUT -i ${PUB_IF} -m pkttype –pkt-type broadcast -j DROP
- $IPT -A INPUT -i ${PUB_IF} -m pkttype –pkt-type multicast -j LOG –log-prefix " Multicast "
- $IPT -A INPUT -i ${PUB_IF} -m pkttype –pkt-type multicast -j DROP
- $IPT -A INPUT -i ${PUB_IF} -m state –state INVALID -j LOG –log-prefix " Invalid "
- $IPT -A INPUT -i ${PUB_IF} -m state –state INVALID -j DROP
- # Log and block spoofed ips
- $IPT -N spooflist
- for ipblock in $SPOOFIP
- do
- $IPT -A spooflist -i ${PUB_IF} -s $ipblock -j LOG –log-prefix " SPOOF List Block "

Nginx 服务器安全加固

➢ 基于Iptables防火墙的限制

- # Log and block spoofed ips
- $IPT -N spooflist
- for ipblock in $SPOOFIP
- do
- $IPT -A spooflist -i ${PUB_IF} -s $ipblock -j LOG –log-prefix ” SPOOF List Block “
- $IPT -A spooflist -i ${PUB_IF} -s $ipblock -j DROP
- done
- $IPT -I INPUT -j spooflist
- $IPT -I OUTPUT -j spooflist
- $IPT -I FORWARD -j spooflist
- # Allow ssh only from selected public ips
- for ip in ${PUB_SSH_ONLY}
- do
- $IPT -A INPUT -i ${PUB_IF} -s ${ip} -p tcp -d ${SERVER_IP} –destination-port 22 -j ACCEPT
- $IPT -A OUTPUT -o ${PUB_IF} -d ${ip} -p tcp -s ${SERVER_IP} –sport 22 -j ACCEPT
- done
- # allow incoming ICMP ping pong stuff
- $IPT -A INPUT -i ${PUB_IF} -p icmp –icmp-type 8 -s 0/0 -m state –state NEW,ESTABLISHED,RELATED -m limit –limit 30/sec -j ACCEPT
- $IPT -A OUTPUT -o ${PUB_IF} -p icmp –icmp-type 0 -d 0/0 -m state –state ESTABLISHED,RELATED -j ACCEPT
- # allow incoming HTTP port 80

Nginx 服务器安全加固

➢ 基于Iptables防火墙的限制

- # allow incoming ICMP ping pong stuff
- $IPT -A INPUT -i ${PUB_IF} -p icmp –icmp-type 8 -s 0/0 -m state –state NEW,ESTABLISHED,RELATED -m limit –limit 30/sec -j ACCEPT
- $IPT -A OUTPUT -o ${PUB_IF} -p icmp –icmp-type 0 -d 0/0 -m state –state ESTABLISHED,RELATED -j ACCEPT
- # allow incoming HTTP port 80
- $IPT -A INPUT -i ${PUB_IF} -p tcp -s 0/0 –sport 1024:65535 –dport 80 -m state –state NEW,ESTABLISHED -j ACCEPT
- $IPT -A OUTPUT -o ${PUB_IF} -p tcp –sport 80 -d 0/0 –dport 1024:65535 -m state –state ESTABLISHED -j ACCEPT
- # allow outgoing ntp
- $IPT -A OUTPUT -o ${PUB_IF} -p udp –dport 123 -m state –state NEW,ESTABLISHED -j ACCEPT
- $IPT -A INPUT -i ${PUB_IF} -p udp –sport 123 -m state –state ESTABLISHED -j ACCEPT
- # allow outgoing smtp
- $IPT -A OUTPUT -o ${PUB_IF} -p tcp –dport 25 -m state –state NEW,ESTABLISHED -j ACCEPT
- $IPT -A INPUT -i ${PUB_IF} -p tcp –sport 25 -m state –state ESTABLISHED -j ACCEPT
- ### add your other rules here ####
- ########################
- # drop and log everything else
- $IPT -A INPUT -m limit –limit 5/m –limit-burst 7 -j LOG –log-prefix ” DEFAULT DROP “
- $IPT -A INPUT -j DROP
- exit 0

Nginx 服务器安全加固

- **控制缓冲区溢出攻击**
- 编辑nginx.conf，为所有客户端设置缓冲区的大小限制。
- vi /usr/local/nginx/conf/nginx.conf
- 编辑和设置所有客户端缓冲区的大小限制如下：
- ## Start: Size Limits & Buffer Overflows ##
- client_body_buffer_size 1K;
- client_header_buffer_size 1k;
- client_max_body_size 1k;
- large_client_header_buffers 2 1k;
- ## END: Size Limits & Buffer Overflows ##

Nginx 服务器安全加固

➢ 控制缓冲区溢出攻击

解释：

1、client_body_buffer_size 1k-（默认8k或16k）这个指令可以指定连接请求实体的缓冲区大小。如果连接请求超过缓存区指定的值，那么这些请求实体的整体或部分将尝试写入一个临时文件。

2、client_header_buffer_size 1k-指令指定客户端请求头部的缓冲区大小。绝大多数情况下一个请求头不会大于1k，不过如果有来自于wap客户端的较大的cookie它可能会大于 1k，Nginx将分配给它一个更大的缓冲区，这个值可以在large_client_header_buffers里面设置。

3、client_max_body_size 1k-指令指定允许客户端连接的最大请求实体大小，它出现在请求头部的Content-Length字段。

如果请求大于指定的值，客户端将收到一个”Request Entity Too Large” (413)错误。记住，浏览器并不知道怎样显示这个错误。

4、large_client_header_buffers-指定客户端一些比较大的请求头使用的缓冲区数量和大小。请求字段不能大于一个缓冲区大小，如果客户端发送一个比较大的头，nginx将返回”Request URI too large” (414)同样，请求的头部最长字段不能大于一个缓冲区，否则服务器将返回”Bad request” (400)。缓冲区只在需求时分开。默认一个缓冲区大小为操作系统中分页文件大小，通常是4k或8k，如果一个连接请求最终将状态转换为keep- alive，它所占用的缓冲区将被释放。

你还需要控制超时来提高服务器性能并与客户端断开连接

Nginx 服务器安全加固

- 按照如下编辑：
- ## Start: Timeouts ##
- client_body_timeout　10;
- client_header_timeout 10;
- keepalive_timeout　　5 5;
- send_timeout　　10;
- ## End: Timeouts ##
- 1、client_body_timeout 10;-指令指定读取请求实体的超时时间。这里的超时是指一个请求实体没有进入读取步骤，如果连接超过这个时间而客户端没有任何响应，Nginx将返回一个”Request time out” (408)错误。
  2、client_header_timeout 10;-指令指定读取客户端请求头标题的超时时间。这里的超时是指一个请求头没有进入读取步骤，如果连接超过这个时间而客户端没有任何响应，Nginx将返回一个”Request time out” (408)错误。
  3、keepalive_timeout 5 5; – 参数的第一个值指定了客户端与服务器长连接的超时时间，超过这个时间，服务器将关闭连接。参数的第二个值（可选）指定了应答头中Keep-Alive: timeout=time的time值，这个值可以使一些浏览器知道什么时候关闭连接，以便服务器不用重复关闭，如果不指定这个参数，nginx不会在应 答头中发送Keep-Alive信息。（但这并不是指怎样将一个连接“Keep-Alive”）参数的这两个值可以不相同。
  4、send_timeout 10; 指令指定了发送给客户端应答后的超时时间，Timeout是指没有进入完整established状态，只完成了两次握手，如果超过这个时间客户端没有任何响应，nginx将关闭连接。

Nginx 服务器安全加固

➢ 控制并发连接

- 你可以使用NginxHttpLimitZone模块来限制指定的会话或者一个IP地址的特殊情况下的并发连接。编辑nginx.conf:
- ### Directive describes the zone, in which the session states are stored i.e. store in slimits. ###
- ### 1m can handle 32000 sessions with 32 bytes/session, set to 5m x 32000 session ###
- limit_zone slimits $binary_remote_addr 5m;
- ### Control maximum number of simultaneous connections for one session i.e. ###
- ### restricts the amount of connections from a single ip address ###
- limit_conn slimits 5;
- 上面表示限制每个远程IP地址的客户端同时打开连接不能超过5个。

Nginx 服务器安全加固

- **只允许我们的域名的访问**
- 如果机器人只是随机扫描服务器的所有域名，那拒绝这个请求。你必须允许配置的虚拟域或反向代理请求。你不必使用IP地址来拒绝。
- ## Only requests to our Host are allowed i.e. nixcraft.in, images.nixcraft.in and www.nixcraft.in
- if ($host !~ ^(nixcraft.in|www.nixcraft.in|images.nixcraft.in)$) {
- return 444;
- }
- ##

Nginx 服务器安全加固

➢ 限制可用的请求方法

- GET和POST是互联网上最常用的方法。 Web服务器的方法被定义在RFC 2616。如果Web服务器不要求启用所有可用的方法，它们应该被禁用。下面的指令将过滤只允许GET，HEAD和POST方法：
- ## Only allow these request methods ##
- if ($request_method !~ ^(GET|HEAD|POST)$) {
- return 444;
- }
- ## Do not accept DELETE, SEARCH and other methods ##
- 更多关于HTTP方法的介绍
- GET方法是用来请求，如文件http://www.moqifei.com/index.php。
- HEAD方法是一样的，除非该服务器的GET请求无法返回消息体。
- POST方法可能涉及到很多东西，如储存或更新数据，或订购产品，或通过提交表单发送电子邮件。这通常是使用服务器端处理，如PHP，Perl和Python等脚本。如果你要上传的文件和在服务器处理数据，你必须使用这个方法。

Nginx 服务器安全加固

- **如何拒绝一些User-Agents?**
- 你可以很容易地阻止User-Agents,如扫描器，机器人以及滥用你服务器的垃圾邮件发送者。
- ## Block download agents ##
- if ($http_user_agent ~* LWP::Simple|BBBike|wget) {
- return 403;
- }
- ##
- 阻止Soso和有道的机器人：
- ## Block some robots ##
- if ($http_user_agent ~* Sosospider|YodaoBot) {
- return 403;
- }

Nginx 服务器安全加固

- **如何防止图片盗链**
- 图片或HTML盗链的意思是有人直接用你网站的图片地址来显示在他的网站上。最终的结果，你需要支付额外的宽带费用。这通常是在论坛和博客。我强烈建议您封锁，并阻止盗链行为。
- # Stop deep linking or hot linking
- location /images/ {
- valid_referers none blocked www.example.com example.com;
- if ($invalid_referer) {
- return 403;
- }
- }
- 例如：重定向并显示指定图片
- valid_referers blocked www.example.com example.com;
- if ($invalid_referer) {
- rewrite ^/images/uploads.*.(gif|jpg|jpeg|png)$ http://www.examples.com/banned.jpg last
- }

Nginx 服务器安全加固

目录限制

- 你可以对指定的目录设置访问权限。所有的网站目录应该一一的配置，只允许必须的目录访问权限。
  通过IP地址限制访问
  你可以通过IP地址来限制访问目录/admin/:
- location /docs/ {
- ## block one workstation
- deny 192.168.1.1;
- ## allow anyone in 192.168.1.0/24
- allow 192.168.1.0/24;
- ## drop rest of the world
- deny all;
- }

Nginx 服务器安全加固

- **通过密码保护目录**

  首先创建密码文件并增加“user”用户：

- mkdir /usr/local/nginx/conf/.htpasswd/
- htpasswd -c /usr/local/nginx/conf/.htpasswd/passwd user
- 编辑nginx.conf,加入需要保护的目录：
- ### Password Protect /personal-images/ and /delta/ directories ###
- location ~ /(personal-images/.|*delta/.*) {
- auth_basic “Restricted”；
- auth_basic_user_file /usr/local/nginx/conf/.htpasswd/passwd;
- }
- 一旦密码文件已经生成，你也可以用以下的命令来增加允许访问的用户：
- htpasswd -s /usr/local/nginx/conf/.htpasswd/passwd userName

Nginx 服务器安全加固

- **Nginx SSL配置**
- HTTP是一个纯文本协议，它是开放的被动监测。你应该使用SSL来加密你的用户内容。
  创建SSL证书
  执行以下命令：
- cd /usr/local/nginx/conf
- openssl genrsa -des3 -out server.key 1024
- openssl req -new -key server.key -out server.csr
- cp server.key server.key.org
- openssl rsa -in server.key.org -out server.key
- openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt

Nginx 服务器安全加固

- **Nginx SSL配置**
- HTTP是一个纯文本协议，它是开放的被动监测。你应该使用SSL来加密你的用户内容。
  编辑nginx.conf并按如下来更新：
- server {
- server_name example.com;
- listen 443;
- ssl on;
- ssl_certificate /usr/local/nginx/conf/server.crt;
- ssl_certificate_key /usr/local/nginx/conf/server.key;
- access_log /usr/local/nginx/logs/ssl.access.log;
- error_log /usr/local/nginx/logs/ssl.error.log;
- }
- 重启nginx:
- /usr/local/nginx/sbin/nginx -s reload

Nginx 服务器安全加固

Nginx与PHP安全建议

- PHP是流行的服务器端脚本语言之一。如下编辑/etc/php.ini文件：
- # Disallow dangerous functions
- disable_functions = phpinfo, system, mail, exec
- ## Try to limit resources ##
- # Maximum execution time of each script, in seconds
- max_execution_time = 30
- # Maximum amount of time each script may spend parsing request data
- max_input_time = 60
- # Maximum amount of memory a script may consume (8MB)
- memory_limit = 8M
- # Maximum size of POST data that PHP will accept.
- post_max_size = 8M
- # Whether to allow HTTP file uploads.
- file_uploads = Off

Nginx 服务器安全加固

Nginx与PHP安全建议

- # Maximum allowed size for uploaded files.
- upload_max_filesize = 2M
- # Do not expose PHP error messages to external users
- display_errors = Off
- # Turn on safe mode
- safe_mode = On
- # Only allow access to executables in isolated directory
- safe_mode_exec_dir = php-required-executables-path
- # Limit external access to PHP environment
- safe*mode_allowed_env_vars = PHP*
- # Restrict PHP information leakage
- expose_php = Off
- # Log all errors
- log_errors = On

Nginx 服务器安全加固

Nginx与PHP安全建议

- # Do not register globals for input data
- register_globals = Off
- # Minimize allowable PHP post size
- post_max_size = 1K
- # Ensure PHP redirects appropriately
- cgi.force_redirect = 0
- # Disallow uploading unless necessary
- file_uploads = Off
- # Enable SQL safe mode
- sql.safe_mode = On
- # Avoid Opening remote files
- allow_url_fopen = Off

Nginx 服务器安全加固

- **如果可能让Nginx运行在一个chroot监狱**
- 把nginx放在一个chroot监狱以减小潜在的非法进入其它目录。你可以使用传统的与nginx一起安装的chroot。如果可能，那使用FreeBSD jails，Xen，OpenVZ虚拟化的容器概念。
- **在防火墙级限制每个IP的连接数**
- 网络服务器必须监视连接和每秒连接限制。PF和Iptales都能够在进入你的nginx服务器之前阻止最终用户的访问。
  Linux Iptables:限制每次Nginx连接数
  下面的例子会阻止来自一个IP的60秒钟内超过15个连接端口80的连接数。
- /sbin/iptables -A INPUT -p tcp –dport 80 -i eth0 -m state –state NEW -m recent –set
- /sbin/iptables -A INPUT -p tcp –dport 80 -i eth0 -m state –state NEW -m recent –update –seconds 60 –hitcount 15 -j DROP
- service iptables save
- 请根据你的具体情况来设置限制的连接数。

Nginx 服务器安全加固

- **配置操作系统保护Web服务器**
- 像以上介绍的启动SELinux.正确设置/nginx文档根目录的权限。Nginx以用户nginx运行。但是根目录（/nginx或者/usr /local/nginx/html）不应该设置属于用户nginx或对用户nginx可写。找出错误权限的文件可以使用如下命令：
- find /nginx -user nginx
- find /usr/local/nginx/html -user nginx
- 确保你更所有权为root或其它用户，一个典型的权限设置 /usr/local/nginx/html/
- ls -l /usr/local/nginx/html/
- 示例输出：
- -rw-r–r– 1 root root 925 Jan 3 00:50 error4xx.html
- -rw-r–r– 1 root root 52 Jan 3 10:00 error5xx.html
- -rw-r–r– 1 root root 134 Jan 3 00:52 index.html
- 你必须删除由vi或其它文本编辑器创建的备份文件：
- find /nginx -name ‘.?’ *-not -name .ht* -or -name ‘~’ *-or -name* ‘.bak’ *-or -name* ‘.old*’
- find /usr/local/nginx/html/ -name ‘.?’ *-not -name .ht* -or -name ‘~’ *-or -name* ‘.bak’ *-or -name* ‘.old*’
- 通过find命令的-delete选项来删除这些文件。

Nginx 服务器安全加固

- **限制Nginx连接传出**

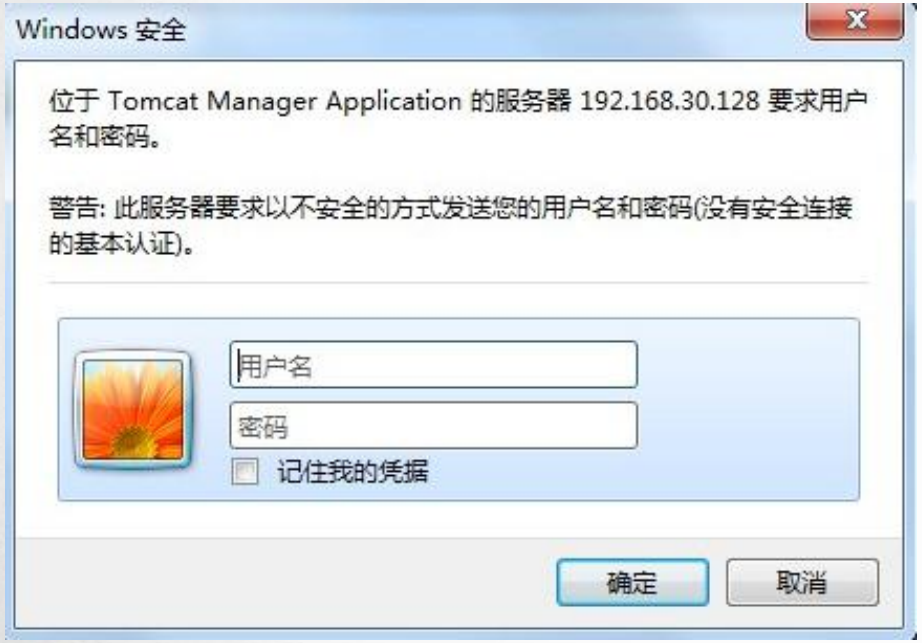
- 黑客会使用工具如wget下载你服务器本地的文件。使用Iptables从nginx用户来阻止传出连接。ipt_owner模块试图匹配本地产生的数据包的创建者。下面的例子中只允许user用户在外面使用80连接。

- /sbin/iptables -A OUTPUT -o eth0 -m owner –uid-owner vivek -p tcp –dport 80 -m state –state NEW,ESTABLISHED -j ACCEPT

- 通过以上的配置，你的nginx服务器已经非常安全了并可以发布网页。可是，你还应该根据你网站程序查找更多的安全设置资料。例如，wordpress或者第三方程序。

Tomcat服务安全加固

- 默认安装Tomcat自带启用了管理后台功能，该后台可直接上传war对站点进行部署和管理，通常由于运维人员的疏忽，导致管理后台空口令或者弱口令的产生，使得黑客或者不法分子利用该漏洞直接上传WEBSHELL导致服务器沦陷。
  通常访问Tomcat后台管理地址为:http://iP:8080/manager/html/, 具体如下图所示：



- **漏洞危害：**
- 通过猜解到的口令等Tomcat管理后台后，可以上传webshell导致服务器被入侵。

Tomcat服务安全加固

- **修复方案：**
- 由于此类型漏洞对业务系统造成比较严重的危害，建议针对tomcat管理后台作如下整改：
- **1.网络访问控制**
- 如果您的业务不需要使用tomcat管理后台管理业务代码，建议您使用安全组防火墙或直接将部署tomcat目录下webapps下的manager、host-manager文件夹全部删除；
- 注释Tomcat目录下conf下的tomcat-users.xml中的所有代码，如下：
- 若业务系统需要使用tomcat管理后台进行业务代码发布和管理，建议为Tomcat管理后台配置强口令，修改默认admin用户，且密码长度不低于10位，必须包含大写字母、特殊符号、数字组合。

Tomcat服务安全加固

- 修复方案：
- 开启Tomcat的访问日志

```
1. 独立运行的tomcat，修改conf/server.xml，取消注释
2. <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
3. prefix="localhost_access_log." suffix=".txt" pattern="common" resolveHosts="false"/>
4. 启用access_log后，重启tomcat，在tomcat_home/logs中可以看到访问日志。
```

Tomcat服务安全加固

- 修复方案：
- Tomcat默认帐号安全

```
1. 修改安装目录conf下的tomcat-user.xml文件，重新设置复杂口令后并保存文件。重启tomcat后新口令生效。
```

```
1.  <!--
2.                      <role rolename="tomcat"/>
3.                      <role rolename="role1"/>
4.                      <user username="tomcat" password="tomcat" roles="tomcat"/>
5.                      <user username="both" password="tomcat" roles="tomcat,role1"/>
6.                      <user username="role1" password="tomcat" roles="role1"/>
7.                       -->
```

Tomcat服务安全加固

- 修复方案：
- 修改默认访问端口

```
1. conf/server.xml把8080改成任意端口
```

Tomcat服务安全加固

- **修复方案：**
- **重定向错误页面**
- 修改访问tomcat错误页面的返回信息，conf/web.xml在倒数第1行之前加

```
<error-page>
                <error-code>401</error-code>
                <location>/401.htm</location>
        </error-page>
        <error-page>
                <error-code>404</error-code>
                <location>/404.htm</location>
        </error-page>
        <error-page>
                <error-code>500</error-code>
                <location>/500.htm</location>
         </error-page>
```

- 然后在webapps\manger目录中创建相应的401.html\404.htm\500.htm文件

Tomcat服务安全加固

- **禁止列目录**
- 防止直接访问目录

- 时由于找不到默认页面而列出目录下的文件

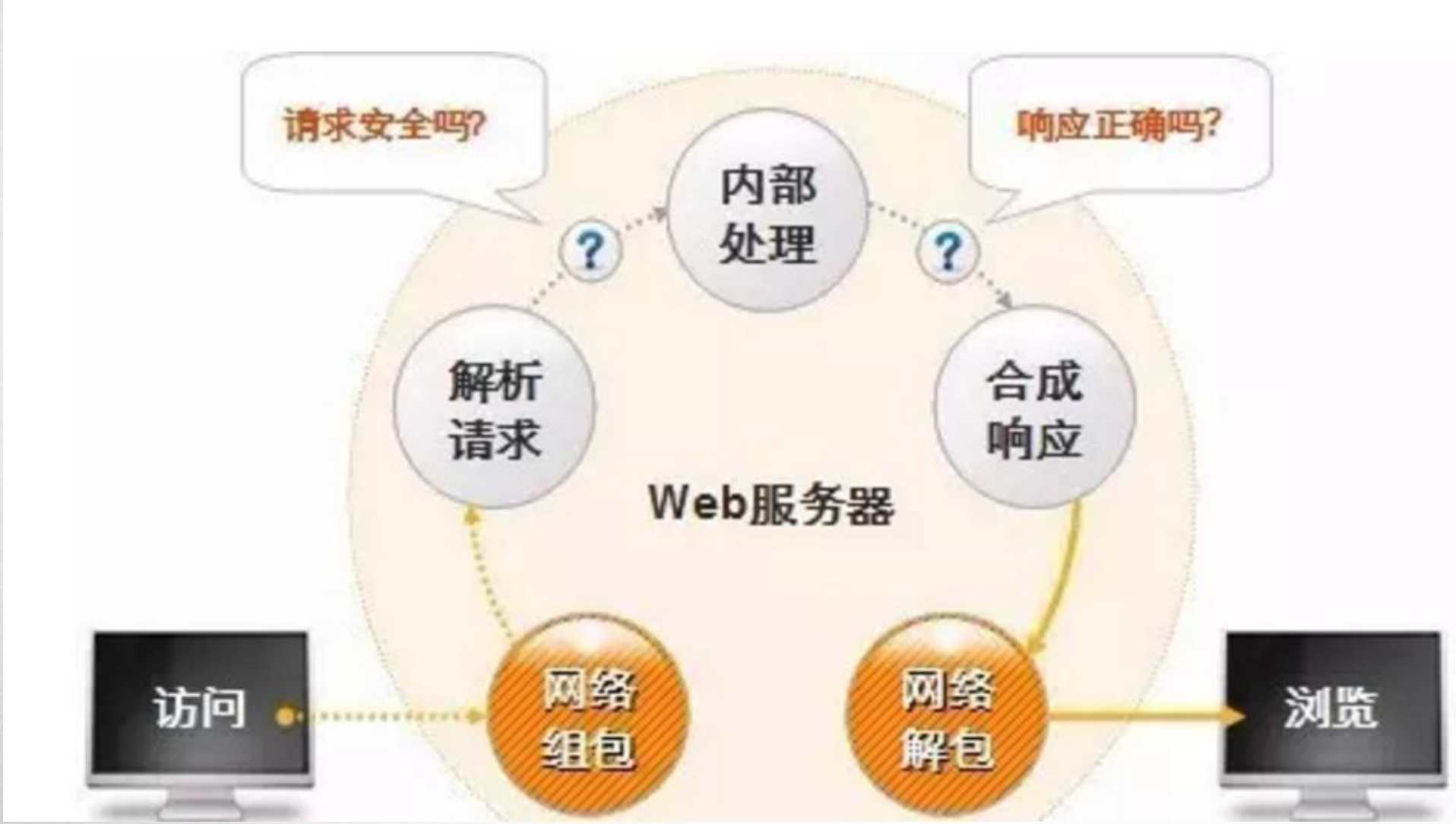
```
1. 打开web.xml，将<param-name>listings</param-name> 改成<param-name>false</param-name>
```

Tomcat服务安全加固

- **删除文档和示例程序**

- 删除webapps/docs、examples、manager、ROOT、host-manager
- 。

Web应用服务器加固思路



Web应用服务器加固思路

- **Web安全分为两大类：**
- · Web服务器的安全性(Web服务器本身安全和软件配置)。
- · Web应用程序的安全性(在Web服务器上运行的Java、 ActiveX、PHP、ASP代码的安全)。
- **Web服务器面临的攻击**
- Web服务器攻击利用Web服务器软件和配置中常见的漏洞。这些漏洞包括：
- · 缓冲区溢出
- · 文件目录遍历
- · 脚本权限
- · 文件目录浏览
- · Web服务器软件默认安装的示例代码
- · Web服务器上运行的其他软件中的漏洞，例如SQL数据库软件
- 让我们对上诉漏洞依个进行深入地探讨。

Web应用服务器加固思路

- **缓冲区溢出**
- 缓冲区溢出允许恶意代码注入到应用程序，它损坏应用程序的堆栈——内存中存储应用程序代码的一个地方——并用不同的代码代替原始代码的一部分来实现攻击者的目的，例如运行特洛伊木马程序或远程控制应用程序。以下是缓冲区溢出漏洞的一个简单示例代码，使用C语言编写：
- char aTmp[100];
- scanf("%s",aTmp);
- 在第一行中，程序员声明一个长度为100的数组aTmp。在第二行中，scanf方法从控制台读取数据存到aTmp数组。代码不会检查%s 变量是否能够容纳输入数据的大小。因为程序员编码过程不对输入字符串的大小进行检查，如果给定的输入超过100个字符，就会造成缓冲区溢出。一个精心构造构的输入中可能包含汇编代码，这部分汇编代码能够获得源程序一样的运行权限。

Web应用服务器加固思路

- **目录遍历**
- 目录遍历是指访问到了不是原先设想或允许的目录(或文件夹)。例如，微软IIS Web站点的默认文件夹为C:\inetpub，攻击者可使用的目录遍历漏洞，在该文件夹之外去读取他们本不该访问的文件。详细来说，假如有一个网址为“www.bad.com”的网站，其服务器代码中包含目录遍历漏洞。攻击者通过输入以下URL就可以利用该漏洞：
- http://www.bad.com/../autoexec.bat
- URL中的“.../”告诉服务器上溯一个目录，也就是“C:\”目录(Web 服务器可以将斜杠转换为反斜杠)。所以如果IIS服务器默认目录为“c:\inetpub”，那么该URL会转到“C:\”目录，攻击者将能够看到“c:\autoexec.bat”文件。除非将服务器配置好了避免目录遍历，不然所有目录可能都是可访问的。这种情况下，Web服务器将显示“autoexec.bat”文件的内容，或者攻击者选择的任何其他文件。
- 值得注意的是:我们已经使用 IIS 作为示例;但是，此漏洞的利用不是针对IIS服务器的，在其他的Web 服务器上也有目录遍历漏洞。

Web应用服务器加固思路

- **脚本权限**
- **目录浏览**
- 通常情况下，目录浏览是禁用的，但是如果启用它，则它显示该目录中的所有文件，并允许浏览的子目录。有时知道一个文件存在可以帮助攻击者利用Web 服务器上文件和程序的漏洞。为此，不建议启用Web 服务器上的目录浏览。
- **5.默认示例**
- 默认示例是包含在Web 服务器软件中并在服务器软件安装时默认安装的应用程序。一些默认安装的示例包含安全漏洞。针对这些漏洞保护的最佳办法是不要安装示例，如果已经安装了，最好把它们删除掉。

Web应用服务器加固思路

- **其他服务**
- 攻击者可以通过攻击在Web服务器上运行的其他服务来攻陷Web服务器。这些服务包括FTP、SMTP、POP3、SQL服务器和NetBIOS服务。防止此类攻击的最佳方法是减少“受攻击面”。关闭所有运行在Web服务器操作系统上不必要的服务并对剩下的服务进行安全地配置。最佳做法是使 Web服务器只有一个Web服务程序，而没有其他的服务。运行数据库和其他的软件应部署在单独的服务器上，这样服务器受防火墙保护，只有Web服务器易受Web攻击。如果攻击者设法利用其他服务的漏洞来攻击服务器，他们也能够干扰或攻陷Web站点。

Web应用服务器加固思路

- **Web服务器软件的固有漏洞**
- 每个Web服务器软件，包括IIS和Apache，由于缺乏安全的编码技术，该软件的程序员已经提供了内置漏洞。例如，IIS的.htr漏洞，允许攻击者看到驻留在服务器上的文件的内容。几乎每周都会发布主要的Web服务器软件平台中的新漏洞。

Web应用服务器加固思路

Web服务器的保护

- 针对上述漏洞最佳做法是遵循以下建议搭建并运行Web服务器。采取下列措施将提高Web服务器的安全性。
- · 给Web服务器服务或守护程序配置能够使它正常运行最少的权限。这样，即使攻击者控制了Web 服务器，他们只能获得运行该软件对应的用户账户的权限。这样，攻击计算机或网络上的其他软件可行方案就极为有限了。
- · 安装最新的安全补丁并时刻关注漏洞的最新动态。
- · 删除默认示例并避免安装类似的示例。
- · 通过删除不需要的应用程序，安全配置同一台计算机上的其他网络服务，确保操作系统已安装最新的安全补丁来保证承载Web服务器的计算机的安全。
- · 确保只给需要执行的脚本单独的目录运行的权限。
- · 在Web服务器上每个目录中，都提供一个index.html文件，以避免需要目录浏览。

Web应用服务器加固思路

- **第三方安全产品**
- 商业和免费的产品也可以帮助抵御与Web服务器相关的不同漏洞。主要有以下产品：
- · 软硬件防火墙
- · Web应用防火墙(WAFs)
- · 病毒防御软件
- · 基于ISAPI的安全产品
- · 安全日志
- · 反馈分析软件
- · 入侵检测系统和入侵检测防御系统
- · 漏洞扫描软件
- · 输入验证