



中间件安全(一)

主讲: Gnosis

中间件安全基础是注册信息安全专业人员需要掌握的通用基础知识。

- 了解中间件的基本概念和加固方法
- 掌握主流中间件的权限配置，解析漏洞风险
- 掌握 JAVA 开发的中间件反序列化漏洞风险

知识体

主流的中间件



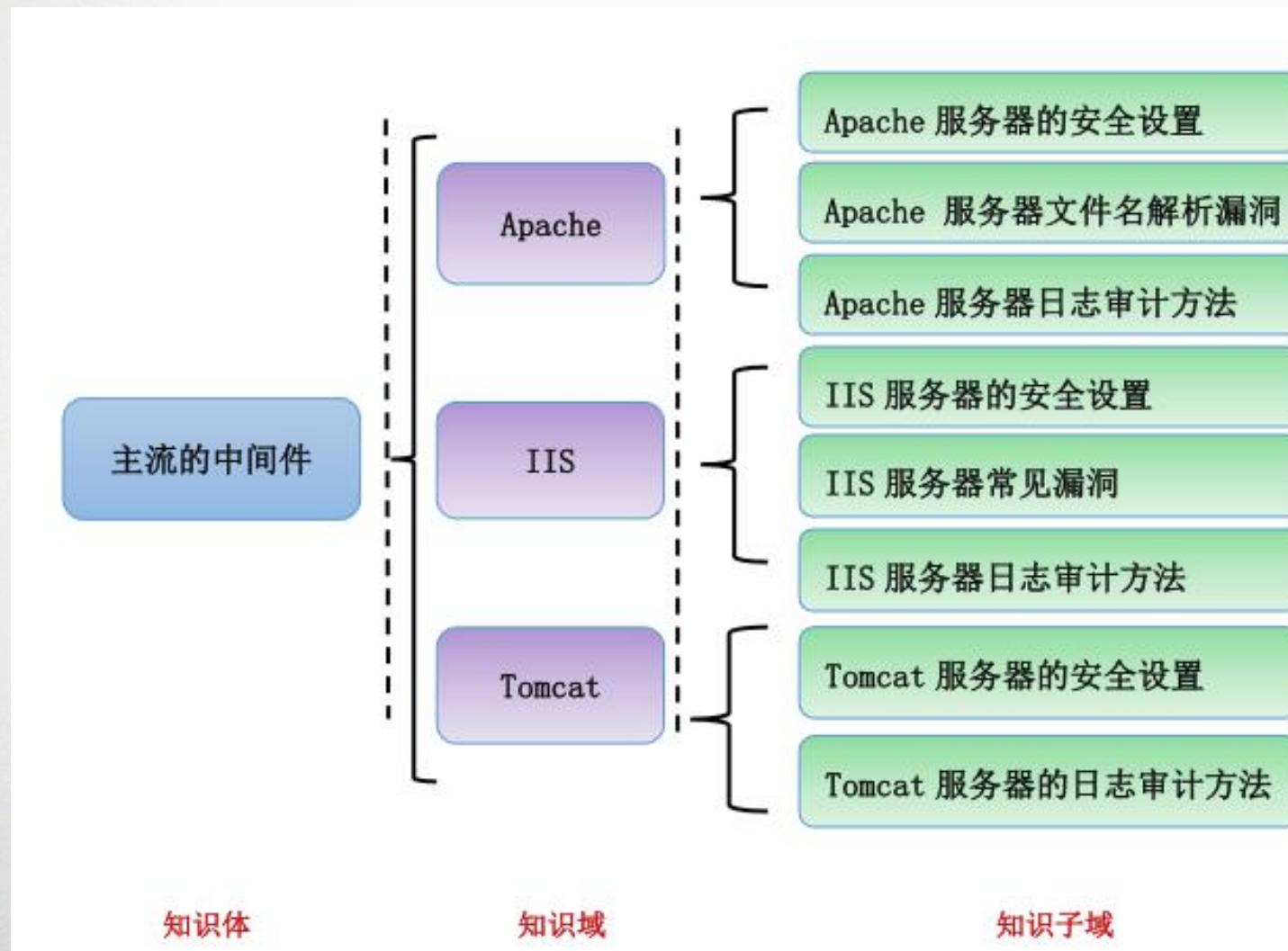
Apache Tomcat®



JAVA 开发的中间件



主流的中间件



Apache 服务器的安全设置

Apache 服务器文件名解析漏洞

Apache 服务器日志审计方法

Apache 服务器的安全设置

Apache是世界使用排名第一的Web[服务器](#)软件。它可以运行在几乎所有广泛使用的[计算机平台](#)上，由于其[跨平台](#)和安全性被广泛使用，是最流行的Web服务器端软件之一。它快速、可靠并且可通过简单的API扩充，将[Perl/Python](#)等[解释器](#)编译到服务器中

Apache 2 Test Page
powered by CentOS

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page it means that the Apache HTTP server installed at this site is working properly.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

If you are the website administrator:

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

You are free to use the images below on Apache and CentOS Linux powered HTTP servers. Thanks for using Apache and CentOS!

About CentOS:

The **Community ENTerprise Operating System** (CentOS) Linux is a community-supported enterprise distribution derived from sources freely provided to the public by Red Hat. As such, CentOS Linux aims to be functionally compatible with Red Hat Enterprise Linux. The CentOS Project is the organization that builds CentOS. We mainly change packages to remove upstream vendor branding and artwork.

For information on CentOS please visit the [CentOS website](#).

Note:

CentOS is an Operating System and it is used to power this website; however, the webserver is owned by the domain owner and not the CentOS Project. **If you have issues with the content of this site, contact the owner of the domain, not the CentOS Project.**

Unless this server is on the `centos.org` domain, the CentOS Project doesn't have anything to do with the content on this webserver or any e-mails that directed you to this site.

For example, if this website is www.example.com, you would find the owner of the `example.com` domain at the following WHOIS server:

<http://www.internic.net/whois.html>

Apache 服务器的安全设置

Apache 自身的安全性是很高的，但是人为的错误设置会导致 Apache 产生安全问题。

了解当前 Apache 服务器的运行权限

了解控制配置文件和日志文件的权限，防止未授权访问

了解设置日志记录文件、记录内容、记录格式

了解禁止 Apache 服务器列表显示文件的方法

了解修改 Apache 服务器错误页面重定向的方法

掌握设置 Web 目录的读写权限，脚本执行权限的方法

Apache 服务器的安全设置——了解当前 Apache 服务器的运行权限

➤ Apache配置文件详细说明

```
# This is the main Apache server configuration file. It contains the
# configuration directives that give the server its instructions.
# See <URL:http://httpd.apache.org/docs/2.2/> for detailed information.
# In particular, see
# <URL:http://httpd.apache.org/docs/2.2/mod/directives.html>
# for a discussion of each configuration directive.
#
# Do NOT simply read the instructions in here without understanding
# what they do. They're here only as hints or reminders. If you are unsure
# consult the online docs. You have been warned.
#
# The configuration directives are grouped into three basic sections:
# 1. Directives that control the operation of the Apache server process as a
#    whole (the 'global environment').
# 2. Directives that define the parameters of the 'main' or 'default' server,
#    which responds to requests that aren't handled by a virtual host.
#    These directives also provide default values for the settings
#    of all virtual hosts.
# 3. Settings for virtual hosts, which allow Web requests to be sent to
#    different IP addresses or hostnames and have them handled by the
#    same Apache server process.
```

Apache服务器主要配置文件

它包含服务器的影响服务器运行的配置指令

#1. 控制整个Apache服务器行为的部分（即全局环境变量）

#2. 定义主要或者默认服务参数的指令，也为所有虚拟主机提供默认的设置参数

#3. 虚拟主机的设置参数

Apache 服务器的安全设置——了解当前 Apache 服务器的运行权限

➤ Apache配置文件详细说明

```
# Configuration and logfile names: If the filenames you specify for many  
# of the server's control files begin with "/" (or "drive:/\" for Win32), the  
# server will use that explicit path. If the filenames do *not* begin  
# with "/", the value of ServerRoot is prepended -- so "logs/foo.log"  
# with ServerRoot set to "/etc/httpd" will be interpreted by the  
# server as "/etc/httpd/logs/foo.log".      配置和日志文件名  
#
```

➤ 安全问题1

默认地，服务器HTTP响应头会包含apache和php版本号。像下面的，这是有危害的，因为这会让黑客通过知道详细的版本号而发起已知该版本的漏洞攻击。

```
### Section 1: Global Environment  
#  
# The directives in this section affect the overall operation of Apache,  
# such as the number of concurrent requests it can handle or where it  
# can find its configuration files.  
#  
#  
# Don't give away too much information about all the subcomponents  
# we are running. Comment out this line if you don't mind remote sites  
# finding out what major optional modules you are running  
ServerTokens OS
```

Apache 服务器的安全设置——了解当前 Apache 服务器的运行权限

➤ 隐藏Apache 版本信息

为了阻止这个，需要在httpd.conf设置ServerTokens为Prod，这会在响应头中显示“Server:Apache”而不包含任何的版本信息。

1.# vi httpd.conf

2.ServerTokens Prod

下面是ServerTokens的一些可能的赋值：

ServerTokens Prod 显示 “Server: Apache”

ServerTokens Major 显示 “Server: Apache/2”

ServerTokens Minor 显示 “Server: Apache/2.2”

ServerTokens Min 显示 “Server: Apache/2.2.17”

ServerTokens OS 显示 “Server: Apache/2.2.17 (Unix)”

ServerTokens Full 显示 “Server: Apache/2.2.17 (Unix) PHP/5.3.5” (如果你这指定任何的值，这个是默认的返回信息)

```
# such as the number of concurrent requests it can handle or where it
# can find its configuration files.
#
#
# Don't give away too much information about all the subcomponents
# we're running. Comment out this line if you don't mind remote sites
# finding out what major optional modules you are running
ServerTokens Prod
#
```

Accept-Ranges: bytes
Connection: close
Content-Length: 4961
Content-Type: text/html; charset=UTF-8
~~Date: Sun, 20 Aug 2017 10:12:32 GMT~~
Server: Apache/2.2.15 (CentOS)

未更改

Accept-Ranges: bytes
Connection: close
Content-Length: 4961
Content-Type: text/html; charset=UTF-8
Date: Sun, 20 Aug 2017 10:18:56 GMT
Server: Apache

更改后

➤ Apache配置文件详细说明

ServerRoot用于指定守护进程httpd的运行目录，httpd在启动之后将自动将进程的当前目录改变为这个目录，因此如果设置文件中指定的文件或目录是相对路径，那么真实路径就位于这个ServerRoot定义的路径之下。

```
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
#           ServerRoot:指出服务器保存其配置、出错和日志文件等的根目录
# NOTE! If you intend to place this on an NFS (or otherwise network)
# mounted filesystem then please read the LockFile documentation
# (available at <URL:http://httpd.apache.org/docs/2.2/mod/mpm_common.html#lockfile>);
# you will save yourself a lot of trouble.
#
# Do NOT add a slash at the end of the directory path.
#
ServerRoot "/etc/httpd"
```

路径的结尾不要添加斜线



Apache 服务器的安全设置——了解当前 Apache 服务器的运行权限

➤ Apache配置文件详细说明

PidFile指定的文件将记录httpd守护进程的进程号，由于httpd能自动复制其自身，因此系统中有多个httpd进程，但只有一个进程为最初启动的进程，它为其他进程的父进程，对这个进程发送信号将影响所有的httpd进程。PidFILE定义的文件中就记录httpd父进程的进程号

```
# PidFile: The file in which the server should record its process
# identification number when it starts. Note the PIDFILE variable in
# /etc/sysconfig/httpd must be set appropriately if this location is
# changed.
#
PidFile run/httpd.pid          PidFile:记录服务器启动进程号的文件

#
# Timeout: The number of seconds before receives and sends time out.
#
Timeout 60                      Timeout:接收和发送前超时秒数

#
# KeepAlive: Whether or not to allow persistent connections (more than
# one request per connection). Set to "Off" for non-persistent connections.
#
KeepAlive Off                  KeepAlive是否允许稳固的连接（每个连接有多个请求），设为"Off"则停用。
```

➤ Apache配置文件详细说明

- MaxKeepAliveRequests:允许持久连接的请求数。设置为0，允许无限数量。建议把这个数字保持在最高，以达到最大的性能。
- KeepAliveTimeout:等待下一个请求的秒数相同的客户端在同一连接上。

- # #服务器池大小调节(MPM特定)
- # prefork MPM (多处理模块)
- StartServers:启动服务器进程的数量
- MinSpareServers:空闲的服务进程的最小数
- MaxSpareServers:空闲的最大服务进程数
- ServerLimit:对MaxClients生命周期最大值
- MaxClients:客户端进程的最大数
- MaxRequestsPerChild:服务器进程服务的请求的最大数量

```
# MaxKeepAliveRequests: The maximum number of requests to allow
# during a persistent connection. Set to 0 to allow an unlimited amount.
# We recommend you leave this number high, for maximum performance.
#
MaxKeepAliveRequests 100

#
# KeepAliveTimeout: Number of seconds to wait for the next request from the
# same client on the same connection.
#
KeepAliveTimeout 15
```

```
## Server-Pool Size Regulation (MPM specific)
## 

# prefork MPM
# StartServers: number of server processes to start
# MinSpareServers: minimum number of server processes which are kept spare
# MaxSpareServers: maximum number of server processes which are kept spare
# ServerLimit: maximum value for MaxClients for the lifetime of the server
# MaxClients: maximum number of server processes allowed to start
# MaxRequestsPerChild: maximum number of requests a server process serves
<IfModule prefork.c>
StartServers      8
MinSpareServers  5
MaxSpareServers  20
ServerLimit      256
MaxClients       256
MaxRequestsPerChild 4000
</IfModule>
```

Apache 服务器的安全设置——了解当前 Apache 服务器的运行权限

➤ Apache配置文件详细说明

侦听特定的IP地址或端口

```
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a DSO you
# have to place corresponding 'LoadModule' lines at this location so the
# directives contained in it are actually available _before_ they are used.
# Statically compiled modules (those listed by 'httpd -l') do not need
# to be loaded here.
#
# Example:
# LoadModule foo_module modules/mod_foo.so
#
LoadModule auth_basic_module modules/mod_auth_basic.so
LoadModule auth_digest_module modules/mod_auth_digest.so
LoadModule authn_file module modules/mod_authn_file.so
LoadModule authn_alias_module modules/mod_authn_alias.so
LoadModule authn_anon_module modules/mod_authn_anon.so
LoadModule authn_dbm_module modules/mod_authn_dbm.so
LoadModule authn_default_module modules/mod_authn_default.so
LoadModule authz_host_module modules/mod_authz_host.so
LoadModule authz_user_module modules/mod_authz_user.so
LoadModule authz_owner_module modules/mod_authz_owner.so
LoadModule authz_groupfile_module modules/mod_authz_groupfile.so
LoadModule authz_dbm_module modules/mod_authz_dbm.so
LoadModule authz_default_module modules/mod_authz_default.so
LoadModule ldap_module modules/mod_ldap.so
LoadModule authnz_ldap_module modules/mod_authnz_ldap.so
LoadModule include_module modules/mod_include.so
LoadModule log_config_module modules/mod_log_config.so
LoadModule logio_module modules/mod_logio.so
LoadModule env_module modules/mod_env.so
LoadModule ext_filter_module modules/mod_ext_filter.so
```

```
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, in addition to the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses (0.0.0.0)
#
#Listen 12.34.56.78:80
Listen 80
```

动态支持模块

Apache 服务器的安全设置——了解当前 Apache 服务器的运行权限

➤ Apache配置文件详细说明

```
root 41640 0.0 0.5 175400 5772 ? S Aug20 0:02 /usr/sbin/httpd  
apache 41648 0.0 0.3 175532 3184 ? S Aug20 0:00 /usr/sbin/httpd  
apache 41649 0.0 0.3 175532 3180 ? S Aug20 0:00 /usr/sbin/httpd  
apache 41650 0.0 0.3 175532 3200 ? S Aug20 0:00 /usr/sbin/httpd  
apache 41651 0.0 0.3 175532 3184 ? S Aug20 0:00 /usr/sbin/httpd  
apache 41652 0.0 0.3 175532 3116 ? S Aug20 0:00 /usr/sbin/httpd  
apache 41653 0.0 0.3 175532 3180 ? S Aug20 0:00 /usr/sbin/httpd  
apache 41654 0.0 0.3 175532 3188 ? S Aug20 0:00 /usr/sbin/httpd
```

当前 Apache 服务器的运行权限

运行httpd服务的用户和组，主要是给网站应用降权的。建议创建一个www用户。当然你可以使用apache这种用户，前提是存在。apache默认是用daemon来运行的，建议降权

```
# Load config files from the config directory "/etc/httpd/conf.d".  
#  
Include conf.d/*.conf  
从配置目录中加载配置文件" / etc / httpd / conf.d  
#  
# ExtendedStatus controls whether Apache will generate "full" status  
# information (ExtendedStatus On) or just basic information (ExtendedStatus  
# Off) when the "server-status" handler is called. The default is Off.  
#  
#ExtendedStatus On  
当调用“服务器状态”处理程序时。默  
认是关闭的  
#  
# If you wish httpd to run as a different user or group, you must run  
# httpd as root initially and it will switch.  
#  
# User/Group: The name (or #number) of the user/group to run httpd as.  
# . On SCO (ODT 3) use "User nouser" and "Group nogroup".  
# . On HPUX you may not be able to use shared memory as nobody, and the  
# suggested workaround is to create a user www and use that user.  
# NOTE that some kernels refuse to setgid(Group) or semctl(IPC_SET)  
# when the value of (unsigned)Group is above 60000;  
# don't use Group #-1 on these systems!  
#  
User apache  
Group apache
```

➤ Apache配置文件详细说明

➤ 禁止访问外部文件

安全基线项目名称	Apache 目录访问权限安全基线要求项
安全基线编号	编号 SBL-Apache-03-01-01 重要
安全基线项说明	禁止 Apache 访问 Web 目录之外的任何文件。
检测操作步骤	<p>1、参考配置操作</p> <p>编辑 httpd.conf 配置文件， <Directory /> + Order Deny,Allow + Deny from all + </Directory> </p> <p>2、补充操作说明</p> <p>设置可访问目录， <Directory /web> + Order Allow,Deny + Allow from all + </Directory> +</p> <p>其中/web 为网站根目录。</p>

```
#  
# Controls who can get stuff from this server.  
#  
# Order allow,deny  
# Allow from all  
#
```

控制谁可以从这个服务器
得到东西

基线符合性	1、判定条件
判定依据	无法访问 Web 目录之外的文件。
2、检测操作	访问服务器上不属于 Web 目录的一个文件，结果应无法显示。

➤ Apache配置文件详细说明

- 下面的行防止.htaccess和.htpasswd文件被Web客户查看。

```
#  
<Files ~ "^.ht">  
Order allow,deny  
Deny from all  
</Files>
```

```
# The following lines prevent .htaccess and .htpasswd files from being  
# viewed by Web clients.  
#  
<Files ~ "^\.\.ht">  
    Order allow,deny  
    Deny from all  
    Satisfy All  
</Files>  
  
#
```

Apache 服务器的安全设置——了解控制配置文件和日志文件的权限，防止未授权访问

- Apache配置文件详细说明
- 日志配置操作

安全基线项目名称	Apache 审核登录策略安全基线要求项	# ErrorLog: The location of the error log file. an ErrorLog directive within a <VirtualHost> blocks relating to that virtual host will be do* define an error logfile for a <VirtualHost> errors will be logged there and not here.
安全基线编号	编号 SBL-Apache-02-01-01 重要	
安全基线项目说明	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。	
检测操作步骤	<p>1、参考配置操作</p> <p>编辑 httpd.conf 配置文件，设置日志记录文件、记录内容、记录格式。</p> <pre>LogLevel notice ErrorLog logs/error_log LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Accept}i\" \"%{Referer}i\" \"%{User-Agent}i\"" combined CustomLog logs/access_log combined</pre> <p>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。</p> <p>CustomLog 指令设置访问日志的文件名和位置。访问日志中会记录服务器所处理的所有请求。</p> <p>LogFormat 设置日志格式。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。</p>	<p>number of messages logged to the error_log. e: debug, info, notice, warn, error, crit,</p> <p>es define some format nicknames for use with (see below).</p> <pre>%r %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" common > %U referer 1 agent</pre>

Apache 服务器的安全设置——了解控制配置文件和日志文件的权限，防止未授权访问

- Apache配置文件详细说明
- 日志配置操作
- 了解设置日志记录文件、记录内容、记录格式

基线符合性	1、判定条件
判定依据	查看 logs 目录中相关日志文件内容，记录完整。
	2、检测操作
	查看相关日志记录。
	3、补充说明

- # LogLevel:控制登录到error_log的消息的数量。可能的值包括:调试、信息、通知、警告、错误、临界(失败)、警戒(必须处理)、致命。

```
#  
# LogLevel: Control the number of messages logged to the error_log.  
# Possible values include: debug, info, notice, warn, error, crit,  
# alert, emerg.  
#  
LogLevel warn  
#
```

Apache 服务器的安全设置——了解禁止 Apache 服务器列表显示文件的方法

➤ Apache配置文件详细说明

➤ 隐藏敏感信息

```
#  
# Optionally add a line containing the server version and virtual host  
# name to server-generated pages (internal error documents, FTP directory  
# listings, mod_status and mod_info output etc., but not CGI generated  
# documents or custom error documents).  
# Set to "EMail" to also include a mailto: link to the ServerAdmin.  
# Set to one of: On | Off | EMail  
#  
ServerSignature On
```

ServerSignature将页脚配置在服务器生成的文档上。就像404示例错误页面一样。正常使用它更好地隐藏整个签名，并添加或修改httpd.conf文件或apache.conf文件的行：

ServerSignature Off

如果您有某些原因想要显示ServerSignature，请使用：

ServerSignature On

或者如果要显示mailto链接（示例管理员邮件），请使用：

ServerSignature Email

安全基线项目名称	Apache 隐藏敏感信息安全基线要求项
安全基线编号	编号 SBL-Apache-03-02-07 重要
安全基线项目说明	隐藏 Apache 的版本号及其它敏感信息。
检测操作步骤	1、参考配置操作 修改 httpd.conf 配置文件： ServerSignature Off ServerTokens Prod
基线符合性判定依据	1、判定条件 2、检测操作 检查配置文件。

Apache 服务器的安全设置——了解禁止 Apache 服务器列表显示文件的方法

➤ Apache配置文件详细说明

➤ 目录列表访问限制

安全基线项目名称	Apache 目录列表安全基线要求项
安全基线编号	编号 SBL-Apache-03-02-04 重要
安全基线项说明	禁止 Apache 列表显示文件
检测操作步骤	<p>1、参考配置操作</p> <p>(1) 编辑 <code>httpd.conf</code> 配置文件， <code><Directory "/web"></code> <code>Options FollowSymLinks</code> <code>AllowOverride None</code></p>

	<pre>Order allow,deny Allow from all </Directory></pre> <p>将 Options Indexes <code>FollowSymLinks</code> 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。</p> <p>(2) 设置 Apache 的默认页面，编辑%apache%\conf\httpd.conf 配置文件， <code><IfModule dir_module></code> <code>DirectoryIndex index.html</code> <code></IfModule></code></p> <p>其中 index.html 即为默认页面，可根据情况改为其它文件。</p> <p>(3) 重新启动 Apache 服务</p>
基线符合性 判定依据	<p>1、判定条件</p> <p>当 WEB 目录中没有默认首页如 index.html 文件时，不会列出目录内容</p> <p>2、检测操作</p> <p>直接访问 http://ip:8800/xxx (xxx 为某一目录)</p>

Apache 服务器的安全设置——掌握设置 Web 目录的读写权限，脚本执行权限的方法

- 我们首先设定网站目录和文件的所有者和所有组为centos, www, 如下命令:
- `chown -R centos:www /home/centos/web`

- 设置网站目录权限为750, 750是centos用户对目录拥有读写执行的权限, 这样centos用户可以在任何目录下创建文件, 用户组有有读执行权限, 这样才能进入目录, 其它用户没有任何权限。
- `find -type d -exec chmod 750 {} \;`

- 设置网站文件权限为640, 640指只有centos用户对网站文件有更改的权限, http服务器只有读取文件的权限, 无法更改文件, 其它用户无任何权限。
- `find -not -type d -exec chmod 640 {} \;`

- 针对个别目录设置可写权限。比如网站的一些缓存目录就需要给http服务有写入权限。例如discuz x2的/data/目录就必须要有写入权限。
- `find data -type d -exec chmod 770 {} \;`

➤ 脚本执行权限的方法

```
#  
#  
#       Order allow,deny  
#       Allow from all  
#       <FilesMatch ".\.(php|asp|jsp)$">  
#           Deny from all  
#       </FilesMatch>  
</Directory>  
  
#
```

- 配置内容中的DIR为需要限制执行脚本文件的目录，FilesMatch后的内容为需要限定的执行的脚本后缀名。例如：这里需要禁止测试站点uploads文件夹下的[PHP](#)，ASP，JSP脚本的运行

Apache 服务器的安全设置——防攻击管理

➤ 限制请求消息长度

安全基线项目名称	Apache 接收 HTTP 请求长度安全基线要求项
安全基线编号	编号 SBL-Apache-03-02-01 重要
安全基线项说明	限制 http 请求的消息主体的大小。
检测操作步骤	<p>1、参考配置操作</p> <p>编辑 httpd.conf 配置文件，修改为 102400Byte</p> <pre>LimitRequestBody 102400</pre> <p>2、补充操作说明</p> <p>上传文件超过 100K 将报错。</p>
基线符合性判定依据	<p>1、判定条件</p> <p>检查配置文件设置。</p> <p>2、检测操作</p> <p>上传文件超过 100K 将报错。</p> <p>3、补充说明</p>

Apache 服务器的安全设置——防攻击管理

➤ 更改默认端口

安全基线项目名称	Apache 运行端口安全基线要求项
安全基线编号	编号 SBL-Apache-03-02-02 重要
安全基线项目说明	更改 Apache 服务器默认端口
检测操作步骤	<p>1、参考配置操作</p> <p>(1) 修改 httpd.conf 配置文件，更改默认端口到 8080 Listen x.x.x.x:8080</p> <p>(2) 重启 Apache 服务</p> <p>2、补充操作说明</p>
基线符合性判定依据	<p>1、判定条件</p> <p>使用 8080 端口登陆页面成功</p> <p>2、检测操作</p> <p>登陆 http://ip:8080</p> <p>3、补充说明</p>

Apache 服务器的安全设置——防攻击管理

➤ 拒绝服务防范

安全基线项目名称	Apache 拒绝服务防范安全基线要求项
安全基线编号	编号 SBL-Apache-03-02-05 重要
安全基线项说明	拒绝服务防范。
检测操作步骤	<p>1、参考配置操作</p> <p>(1) 编辑 <code>httpd.conf</code> 配置文件， <code>Timeout 10 KeepAlive On</code> <code>KeepAliveTimeout 15</code> <code>AcceptFilter http data</code> <code>AcceptFilter https data</code></p> <p>(2)重新启动 Apache 服务</p>
基线符合性判定依据	<p>1、判定条件</p> <p>2、检测操作</p> <p>检查配置文件是否设置。</p>

Apache 服务器的安全设置——防攻击管理

➤ 删除无用文件

安全基线项目名称	Apache 无用文件安全基线要求项
安全基线编号	编号 SBL-Apache-03-02-06 重要
安全基线项说明	删除缺省安装的无用文件。
检测操作步骤	<p>1、参考配置操作</p> <p>删除缺省 HTML 文件：</p> <pre># rm -rf /usr/local/apache2/htdocs/*</pre> <p>删除缺省的 CGI 脚本：</p> <pre># rm -rf /usr/local/apache2/cgi-bin/*</pre> <p>删除 Apache 说明文件：</p> <pre># rm -rf /usr/local/apache2/manual</pre> <p>删除源代码文件：</p> <pre># rm -rf /path/to/httpd-2.2.4*</pre> <p>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。</p>
基线符合性判定依据	<p>1、判定条件</p> <p>2、检测操作</p> <p>检查对应目录。</p>

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IP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\n %>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，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

```
1. #LoadModule cgi_module modules/mod_cgi.so
2. #ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"
3. #
4. #AllowOverride None
5. # Options None
6. #Order allow,deny
7. #Allow from all
8. #
```

- 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

```
1. <Location />
2. <LimitExcept GET POST CONNECT OPTIONS>
3.   Order Allow,Deny
4.   Deny from all
5. </LimitExcept>
6. </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

Apache服务安全加固

➤ 使用mod_security和mod_evasive来保障Apache的安全

- “mod_security” 和 “mod_evasive” 是Apache在安全方面非常流行的两个模块。mod_security作为防火墙而运行，它允许我们适时地监视通信，还可以有助于我们保护网站或Web服务器免受暴力破解攻击。借助默认的包安装程序，我们可以轻松地把mod_security安装在服务器上。：

```
# yum install mod_security  
# /etc/init.d/httpd restart
```

- 另一个模块mod_evasive的工作效率很高，它只采用一个请求就可以很好地工作，可以防止DDoS攻击造成巨大危害。mod_evasive可以应对http暴力破解攻击和DoS(或DDoS)攻击。该模块可以在三种情况下检测攻击：一是在每秒钟内有太多请求到达同一个页面时,二是在任何子进程试图发出超过50个并发请求时，三是在任何地址已经被临时列入黑名单时它仍试图尝试新的请求

Apache服务安全加固

➤ DDoS攻击的防御和强化

- 你不可能完全阻止企业网站免受DDoS攻击。下面这些命令便于你进行控制。
- TimeOut指令用于设置在特定事件失效之前，服务器等待事件完成的时间长度。其默认值是300秒。对于容易遭受DDoS攻击的网站，把这个值降低很有好处。这个值的大小取决于网站上的请求种类。注意，对于某些CGI脚本，这个设置可能会产生问题。
- MaxClients：此指令允许用户设置服务器可同时服务的连接限制。每一个新连接都要根据这个限制进行排队。它适用于Prefork和Worker。其默认值为256。
- KeepAliveTimeout：在关闭连接之前，服务器随后的等待时间长度。默认值是5秒。
- LimitRequestFields：这个设置可以帮助我们限制可以接受的HTTP请求的头部字段数量。其默认值为100。有时，由于http的请求头部过多而导致发生DDoS攻击，用户不妨降低这个值。
- LimitRequestFieldSize：帮助我们设置HTTP请求头部的大小。

StartServers	8
MinSpareServers	5
MaxSpareServers	20
ServerLimit	256
MaxClients	256
MaxRequestsPerChild	4000

Apache服务安全加固

➤ 用ssl证书保障Apache的安全

- 你还可以用SSL证书用加密的方式保障信息传输的安全。在电子商务网站中，消费者为了买东西，有时需要提供账户或信用卡的细节，默认情况下，Web服务器用明文发送这些信息。配置服务器使其借助于SSL证书就可以为用户进行加密传输。
- **openssl genrsa -des3 -out example.com.key 1024**
- **openssl req -new -key example.com.key -out exmaple.csr**
- **openssl x509 -req -days 365 -in example.com.com.csr -signkey example.com.com.key -out example.com.com.crt**
- 在创建并签署了证书后，你需要在Apache配置中增加这个证书。用vim编辑器打开主配置文件，并增加下面的内容，然后重启服务：

```
SSLEngine on  
SSLCertificateFile /etc/pki/tls/certs/example.com.crt  
SSLCertificateKeyFile /etc/pki/tls/certs/example.com.key  
SSLCertificateChainFile /etc/pki/tls/certs/sf_bundle.crt  
ServerAdmin ravi.saive@example.com  
ServerName example.com  
DocumentRoot /var/www/html/example/  
ErrorLog /var/log/httpd/example.com-error_log  
CustomLog /var/log/httpd/example.com-access log common
```

Apache 服务器的安全设置

Apache 服务器文件名解析漏洞

Apache 服务器日志审计方法

Apache 服务器文件名解析漏洞——了解 Apache 服务器解析漏洞的利用方式

➤ Apache文件名解析特性

- Apache对于文件名的解析是从后往前解析的，直到遇见一个它认识的文件类型为止。因此，如果web目录下存在以类似webshell.php.test这样格式命名的文件，Apache在解析时因为不认识.test这个文件类型，所以会一直往前解析，当解析到.php时，它认识了，因此会将它解析为PHP文件
- Apache的这种解析特性经常被用来绕过Web应用的文件上传检测。当Web应用的文件上传功能在检测上传文件的合法性时，如果仅通过检测上传文件的扩展名来判断文件是否合法，就可以利用Apache的这种文件名解析特征绕过Web应用的检测
- 下面来看一个实例：目标网站后台存在一个上传图片的功能，只允许上传JPG和GIF图片格式的文件。但程序在验证上传文件合法性处存在漏洞，只是简单地通过上传文件扩展名来确定文件是否合法，这时我们就可以利用Apache的文件名解析特征来绕过这种检测。
将文件名修改为类似phpshell.php.jpg这样的格式上传，发现绕过了检测，文件被成功上传到目标网站

File Manager - Current disk free 9.49 G of 13.44 G (70.63%)

Current Directory (Writable, 0777) /var/www/cxpath/link/

[WebRoot](#) | [View Writable](#) | [Create Directory](#) | [Create File](#)

Filename	Last modified
Parent Directory	
1.jpg	2012-09-05 17:07:43
1.php	2012-09-05 17:07:43

Apache 服务器文件名解析漏洞——了解 Apache 服务器解析漏洞的利用方式

➤ Apache文件名解析特性

- 可以在httpd.conf配置文件中添加以下内容来阻止Apache解析这种文件。

```
<Files ~ "\.(php.)">  
Order Allow,Deny  
Deny from all  
</Files>
```

- 修改后需要重启Apache服务生效。
- 这样即使攻击者上传了类似phphshell.php.jpg这样格式的文件，Apache也不会将它解析为PHP文件了



- 网上说的“低版本的apache存在未知扩展名解析漏洞”的说法是错误的，正确的说法应该是使用module模式与php结合的所有版本 apache存在未知扩展名解析漏洞，使用fastcig模式与php结合的所有版本apache不存在此漏洞

➤ 安全加固

➤ Apache的安全加固我们主要从以下两点考虑：一是Apache Web Server本身是否安全，比如是否存在安全漏洞；二是Apache Web Server是否提供了可使用的安全功能，这部分主要是检查Apache的配置是否得当，在安全性、可用性、稳定性之间取得平衡。

➤ Apache版本的选择与安装注意事项

➤ 检查目前使用的Apache版本是否存在安全漏洞，如果存在，需要升级到新的安全版本。在选择Apache的版本时，我们一般选择最新的稳定版本。这样可以在安全性和稳定性之间取得一个很好的平衡。从低版本升级到高版本时，建议先在测试环境中测试通过后再进行升级，以避免由于兼容性带来的问题。

➤ 在安装时使用自定义的安装路径，并配置使用自定义的WEB目录。

Apache 服务器的安全设置

Apache 服务器文件名解析漏洞

Apache 服务器日志审计方法

Apache 服务器日志审计

➤ Apache 服务的日志文件，默认情况下主要有两种

/var/log/httpd/access_log 记录用户访问网站的记录信息

/var/log/httpd/error_log 记录用户错误请求的信息，包括 Web 服务启动或运行过程中的问题，比如网页找不到、文件权限设置不正确等

Apache 服务日志的分析

Apache 的访问日志默认存放在 Apache 安装目录的 logs 目录下，名称为 access.log，具体位置可以在 httpd.conf 或 Apache 安装目录下的 conf/vhosts/ 目录中的站点配置文件 *.conf 中进行指定

```
140.205.201.4 - - [21/Aug/2017:04:30:53 +0800] "GET /icons/apache_pb.gif HTTP/1.1" 200 2326 "http://cmdb.catroot.cn/" "Mozilla/4.0 (compatible; MSIE 8.0; Trident/4.0; Windows NT 6.1; SLCC2 2.5.5231; .NET CLR 2.0.50727; .NET CLR 4.1.23457; .NET CLR 4.0.23457; Media Center PC 6.0; MS-WK 8)"  
140.205.201.26 - - [21/Aug/2017:04:30:55 +0800] "GET /icons/apache_pb.gif HTTP/1.1" 200 2056 "http://cmdb.catroot.cn/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.117 Safari/537.36"  
  
66.249.65.216 - - [21/Aug/2017:17:42:47 +0800] "GET /dist/css/ace-rtl.min.css HTTP/1.1" 404 287 "http://cmdb.catroot.cn/site/login" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"  
66.249.65.212 - - [21/Aug/2017:17:42:48 +0800] "GET /dist/css/ace.min.css HTTP/1.1" 404 283 "http://cmdb.catroot.cn/site/login" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"  
66.249.65.214 - - [21/Aug/2017:17:42:48 +0800] "GET /dist/css/font-awesome.min.css HTTP/1.1" 404 292 "http://cmdb.catroot.cn/site/login" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"  
66.249.65.212 - - [21/Aug/2017:17:42:50 +0800] "GET /dist/js/jquery-2.0.3.min.js HTTP/1.1" 404 290 "http://cmdb.catroot.cn/site/login" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"
```

Apache 服务器日志审计

Apache 服务日志的分析

```
123.118.235.44 - - [21/Jan/2015:20:23:53 +0800] "GET /info.php?=PHPE9568F34-D428-11d2-A769-00AA001ACF42 HTTP/1.1" 200 2524
```

各字段说明

- 项目 样例
- 客户端IP **123.118.235.44**
- 由客户端 identd 进程判断的 RFC1413 身份(identity) -
- 注意：输出中的符号 “-” 表示此处的信息无效。除非在严格控制的内部网络中，此信息通常很不可靠，不应该被使用。只有在将 *IdentityCheck* 指令设为 *On* 时，Apache 才会试图得到这项信息。
- 记录用户HTTP的身份验证 -
- 服务器完成请求处理时的时间 **[21/Jan/2015:20:23:53 +0800]**格式:[日/月/年:时:分:秒 时区]
- 请求方式，请求资源，协议 **GET /info.php?=PHPE9568F34-D428-11d2-A769-00AA001ACF42 HTTP/1.1**
- 协议状态码 **200**
- 服务器向客户端发送的字节数 **2524**

Apache 服务器日志审计

Apache 服务日志的分析

以下列举一些命令便于快速分析日志：

获得访问前 10 位的 IP 地址

➤ `cat access_log | awk '{print $1}' |sort|uniq -c |sort -nr |head -10`

```
root@root httpd]# cat access_log | awk '{print $1}' |sort|uniq -c |sort -nr |head -10
 64 14.215.176.21
 61 14.215.176.20
 55 14.215.176.148
 54 14.215.176.149
 47 39.155.208.201
   6 221.204.19.123
   6 212.237.59.163
   6 212.237.26.152
   6 101.199.112.52
   4 39.155.208.203
```

Apache 服务器日志审计

Apache 服务日志的分析

以下列举一些命令便于快速分析日志：

- 访问次数最多的文件或页面,取前 10
- `cat access_log |awk '{print $11}'|sort |uniq -c| sort -nr | head -10`

```
root@root httpd]# cat access_log | awk '{counts[$(11)]+=1}; END {for(url in counts) print counts[url], url}'  
  
"http://catroot.cn/icons/poweredbypng"  
"http://114.115.206.60/phpmyadmin"  
"http://cmdb.catroot.cn/site/login"  
"http://cmdb.catroot.cn/icons/poweredbypng"  
"http://job.catroot.cn/site/login"  
"http://cmdb.catroot.cn/icons/apache_pb.gif"  
"http://114.115.206.60:80"  
9 "http://114.115.206.60/"  
"http://job.catroot.cn/"  
"http://job.catroot.cn/icons/apache_pb.gif"  
"http://job.catroot.cn/icons/poweredbypng"  
16 "-"  
"http://cmdb.catroot.cn/"  
"http://catroot.cn/icons/apache_pb.gif"  
"http://catroot.cn/"  
root@root httpd]#
```

Apache 服务器日志审计

Apache 服务日志的分析

以下列举一些命令便于快速分析日志：

- 统计此日志文件中所有的流量
- `cat access_log |awk '{sum+=$10} END {print sum/1024/1024/1024 "G"}'`

```
[root@root httpd]# cat access_log |awk '{sum+=$10} END {print sum/1024/1024/1024 "G"}'  
0.000501184G  
[root@root httpd]#
```

- 列出输出大于 200000byte (约200kb) 的 exe 文件以及对应文件发生次数
- `cat access_log |awk '($10 > 200000 && $7~/.exe/){print $7}'|sort -n|uniq -c|sort -nr|head -100`

Apache 服务器日志审计

Apache 服务日志的分析

以下列举一些命令便于快速分析日志：

如果日志最后一列记录的是页面文件传输时间，则有列出到客户端最耗时的页面

cat access_log |awk '\$7~/\.php/){print \$NF " " \$1 " " \$4 " " \$7}'|sort -nr|head -20

```
[root@root httpd]# cat access_log |awk '$7~/\.php/){print $NF " " $1 " " $4 " " $7}'|sort -nr|head -20
"ZmEu" 212.237.59.163 [20/Aug/2017:21:25:52 /MyAdmin/scripts/setup.php
"ZmEu" 212.237.59.163 [20/Aug/2017:21:25:52 /myadmin/scripts/setup.php
"ZmEu" 212.237.59.163 [20/Aug/2017:21:25:51 /pma/scripts/setup.php
"ZmEu" 212.237.59.163 [20/Aug/2017:21:25:51 /phpmyadmin/scripts/setup.php
"ZmEu" 212.237.59.163 [20/Aug/2017:21:25:50 /phpMyAdmin/scripts/setup.php
"Go-http-client/1.1" 139.162.88.63 [20/Aug/2017:21:37:34 http://clientapi.ipip.net/echo.php?info=1234567890
Firefox/3.5.1" 14.215.176.21 [20/Aug/2017:13:06:59 //templates/default/layout/footer.php
Firefox/3.5.1" 14.215.176.21 [20/Aug/2017:13:06:22 //hm_frontpage.php
Firefox/3.5.1" 14.215.176.21 [20/Aug/2017:13:06:22 //admin/privilege.php?act=login
Firefox/3.5.1" 14.215.176.21 [20/Aug/2017:13:06:21 //main.php
Firefox/3.5.1" 14.215.176.21 [20/Aug/2017:13:06:19 //admin.php?mod=phpcms&file=login
Firefox/3.5.1" 14.215.176.21 [20/Aug/2017:13:06:11 //discuz_version.php
Firefox/3.5.1" 14.215.176.21 [20/Aug/2017:13:06:10 //source/discuz_version.php
Firefox/3.5.1" 14.215.176.20 [20/Aug/2017:13:07:09 //uc_center/admin.php
Firefox/3.5.1" 14.215.176.20 [20/Aug/2017:13:07:00 ///rss.php
Firefox/3.5.1" 14.215.176.20 [20/Aug/2017:13:06:57 //akcms_inc.php?i=28
Firefox/3.5.1" 14.215.176.20 [20/Aug/2017:13:06:52 //login.php
Firefox/3.5.1" 14.215.176.20 [20/Aug/2017:13:06:28 //data/module/notexist_path.php
Firefox/3.5.1" 14.215.176.20 [20/Aug/2017:13:06:24 //news_manajemen/index.php
Firefox/3.5.1" 14.215.176.20 [20/Aug/2017:13:06:12 //admin.php
```

Apache 服务器日志审计

Apache 服务日志的分析

以下列举一些命令便于快速分析日志：

列出最耗时的页面(超过 60 秒的)的以及对应页面发生次数

```
cat access_log |awk '$NF > 60 && $7~/.php/){print $7}'|sort -n|uniq -c|sort -nr|head -2
```

```
[root@root httpd]# cat access_log |awk '$NF > 60 && $7~/.php/){print $7}'|sort -n|uniq -c|sort -nr|head -2
      1 //wp-login.php
      1 //uc_center/admin.php
```

Apache 服务器日志审计

Apache 服务日志的分析

以下列举一些命令便于快速分析日志：

列出传输时间超过 30 秒的文件

```
cat access_log |awk '($NF > 30){print $7}'|sort -n|uniq -c|sort -nr|head -20
```

```
[root@root httpd]# cat access_log |awk '($NF > 30){print $7}'|sort -n|uniq -c|sort -nr|head -20
47 /
20 /icons/poweredb.png
20 /icons/apache_pb.gif
2 /favicon.ico
1 //zb_system/script/common.js
1 //wtnews.xml
1 //wp-login.php
1 //wp-includes/wlwmanifest.xml
1 //wp-includes/images/crystal/license.txt
1 //wp-content/plugins/wooframework-tweaks/readme.txt
1 //wp-content/plugins/all-video-gallery/readme.txt
1 //wlwmanifest.xml
1 //Web.sitemap
1 //web-console/ServerInfo.jsp
1 //upfile/js/index.js
1 //ufinder/ufinder.js
1 //ufinder/ufinder.config.js
1 //ueditor/ueditor.config.js
1 //uc_center/images/admincp.css
1 //uc_center/admin.php
```

Apache 服务器日志审计

Apache 服务日志的分析

以下列举一些命令便于快速分析日志：

- 统计 404 的连接
- `awk '($9 ~/404/)' access_log | awk '{print $9,$7}' | sort |uniq -c |head -10`

```
[root@root httpd]# awk '($9 ~/404/)' access_log | awk '{print $9,$7}' | sort |uniq -c |head -10
1 404 //1.html?%201=2And%203=
1 404 //addons/theme/stv1/_static/js/core.js
1 404 ///admin/
1 404 //admin_aspcms/js/menu.js
1 404 //admin/cpstyle.css
1 404 //admin/default.aspx
1 404 //admin/editor/editor/dialog/fck_about.html
1 404 //administrator/manifests/files/joomla.xml
1 404 //admin.php
1 404 //admin.php?mod=phpcms&file=login
[root@root httpd]#
```

Apache 服务器日志审计

Apache 服务日志的分析

以下列举一些命令便于快速分析日志：

统计 HTTP Status

```
cat access_log |awk '{print $9}'|sort|uniq -c|sort -rn
```

```
[root@root httpd]# cat access_log |awk '{print $9}'|sort|uniq -c|sort -rn
 263 404
   72 403
   35 200
    6 304
    4 400
    2 "-"
  1 503
  1 500
  1 302
  1 301
  1 204
  1 200
  1 104
  1 103
  1 100
  1 004
  1 003
  1 000
```

Apache 服务器日志审计

Apache 服务日志的分析

以下列举一些命令便于快速分析日志：

- 蜘蛛分析查看是哪些蜘蛛来访问过
- `cat access_log |awk '{print $12}' | grep -iE 'bot|crawler|slurp|spider' |sort |uniq -c`

```
[root@aliyun ~]# cat access.log |awk '{print $12}' | grep -iE 'bot|crawler|slurp|spider' |sort |uniq -c|sort -rn
6376 "Baidu-YunGuance-SLABot(ce.baidu.com)"
3546 "YisouSpider"
204 "Googlebot-Image/1.0"
91 "Baiduspider-image(+http://www.baidu.com/search/spider.htm)"
48 "Baidu-YunGuance-RSBot(ce.baidu.com)"
36 "Googlebot/2.1
33 "spider-ads"
18 "msnbot-media/1.1
17 "Mozilla/5.0(compatible;+Sosospider/2.0;++http://help.soso.com/webspider.htm)"
16 "360spider-image"
13 "Baiduspider(+http://www.baidu.com/search/spider.htm)"
12 "360spider(http://webscan.360.cn)"
11 "compatible;Baiduspider/2.0;
10 "Googlebot/1.0
7 "spiderman"
7 "Googlebot-video/1.0"
6 "msnbot/2.0b
6 "Googlebot
5 "EtaoSpider"
3 "SinaweiboBot"
2 "tbot-nutch/Nutch-1.10"
2 "RSSingBot
1 "MediaVBot/1.0
1 "AdnormCrawler
```

Apache 服务器日志审计

那么你应当在web日志中寻找哪些蛛丝马迹来分析确定针对web服务器的恶意活动呢？

- 1) ' 单引号需要注意，因为经常这是SQL注入攻击的特征。
- 2) .. 和 .. 表示目录跳转。
- 3) 注意以下两个文件，如果其服务器对应的响应代码是200,很不幸，意味着你的系统已经被攻克了。

/etc/passwd
/etc/shadow

- 4) 以下各种路径如果在web日志中出现，也值得严重关切：

/bin/ksh
/bin/bash
/bin/id
/bin/cat

Apache 服务器日志审计

那么你应当在web日志中寻找哪些蛛丝马迹来分析确定针对web服务器的恶意活动呢？

在windows平台需要在web日志中注意如下内容：

msadc/.◆

scripts/..\\../winnt/system32/cmd.exe?/c+dir

cmd.exe

net.exe

netstat.exe

5) 如果攻击成功，通常会使用一些常用的命令，这里面可能包括如下的特征：

| 管道

< 输出重定向

; 分号

6) ASCII控制字符：

Apache 总结

- 知识子域：Apache 服务器的安全设置
 - 了解当前 Apache 服务器的运行权限
 - 了解控制配置文件和日志文件的权限，防止未授权访问
 - 了解设置日志记录文件、记录内容、记录格式
 - 了解禁止 Apache 服务器列表显示文件的方法
 - 了解修改 Apache 服务器错误页面重定向的方法
 - 掌握设置 Web 目录的读写权限，脚本执行权限的方法
-
- 知识子域：Apache 服务器文件名解析漏洞
 - 了解 Apache 服务器解析漏洞的利用方式
 - 掌握 Apache 服务器文件名解析漏洞的防御措施
 -
-
- 知识子域：Apache 服务器日志审计
 - 掌握 Apache 服务器日志审计方法

THANK YOU 感谢观看
FOR YOUR ATTENTION!

北京谷安天下科技有限公司

谷安天下公司主页：www.gooann.com

谷安培训教育网页：<http://px.gooann.com>

安全意识产品网页：<http://sectv.gooann.com>

产品解决方案网页：<http://product.gooann.com>

谷安信息安全商城：<http://gooannpx.taobao.com>