**主机头漏洞**

## 安全威胁

远程攻击者可以利用漏洞绕过限制访问受限资源。

攻击所需条件：攻击者必须访问透明代理服务器。

漏洞信息：HTTP Host头字段在RFC 2616中规范定义，用于web服务器以允许多个WEB站点共享一个IP地址。

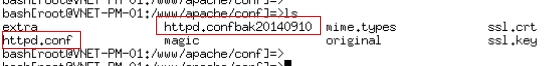
当作为透明代理时，应用程序依赖HTTP "Host:"头字段数据存在安全问题，可导致访问受限WEB站点或通过发送伪造的HTTP "Host:"头字段的HTTP请求可绕过浏览器的安全上下文保护机制。

## 整改建议

Web应用程序应该使用SERVER\_NAME而不是主机头。它也应该创建一个虚拟的虚拟主机，捕捉具有无法识别的主机标头的所有要求。Nginx配置下，通过利用非通配符：ServerName和转向UseCanonicalName指令对这个也可以做通过指定一个非通配符SERVER\_NAME，并在Apache下。

## 案例参考

**一、在Httpd.conf开启Include conf /extra/httpd-vhosts.conf 虚拟主机服务**



**二、根据现网环境修改正则**



默认下httpd-vhosts.conf内容如下：

<VirtualHost \*:80>

ServerAdmin webmaster@dummy-host.hleclerc-PC.ingenidev

DocumentRoot "C:/Dev/Projets/WampServer2-64b/install\_files\_wampserver2/bin/apache/Apache2.2.17/docs/dummy-host.hleclerc-PC.ingenidev"

ServerName dummy-host.hleclerc-PC.ingenidev

ServerAlias www.dummy-host.hleclerc-PC.ingenidev

ErrorLog "logs/dummy-host.hleclerc-PC.ingenidev-error.log"

CustomLog "logs/dummy-host.hleclerc-PC.ingenidev-access.log" common

</VirtualHost>

<VirtualHost \*:80>

ServerAdmin webmaster@dummy-host2.hleclerc-PC.ingenidev

DocumentRoot "C:/Dev/Projets/WampServer2-64b/install\_files\_wampserver2/bin/apache/Apache2.2.17/docs/dummy-host2.hleclerc-PC.ingenidev"

ServerName dummy-host2.hleclerc-PC.ingenidev

ErrorLog "logs/dummy-host2.hleclerc-PC.ingenidev-error.log"

CustomLog "logs/dummy-host2.hleclerc-PC.ingenidev-access.log" common

</VirtualHost>



修改成：

<VirtualHost \*:80>

ServerAdmin webmaster@163.com

DocumentRoot "/www/apache/htdocs"

ServerName portalman.gd.chinavent.com

ErrorLog "logs/dummy-host.localhost-error.log"

CustomLog "logs/dummy-host.localhost-access.log" combined

<Directory /www/apache/htdocs >

SetEnvIf Host "portalman.gd.chinavnet.com" gd=1

Order allow,deny

Allow from env=gd

</Directory>

</VirtualHost>

根据现网环境需要修改红色标志内容

<VirtualHost \*:80>

网站服务端口

DocumentRoot "/www/apache/htdocs"

网站目录绝对路径

ServerName portalman.gd.chinavent.com

网站域名

<Directory /www/apache/htdocs >

网站目录绝对路径

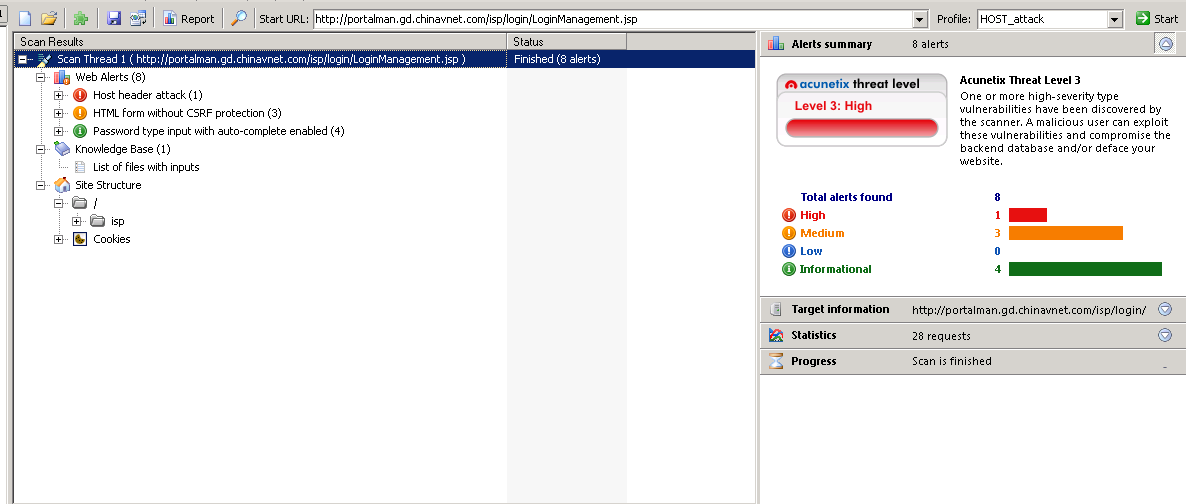
SetEnvIf Host "portalman.gd.chinavnet.com" gd=1

网站域名

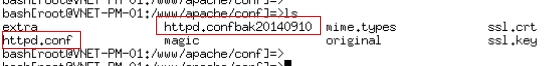
配置完毕 重启WEB服务即可。

## 主机头漏洞加固指南

存在主机头漏洞服务器：



一、在Httpd.conf开启Include conf /extra/httpd-vhosts.conf 虚拟主机服务





二、根据现网环境修改正则



默认下httpd-vhosts.conf内容如下：

<VirtualHost \*:80>

ServerAdmin webmaster@dummy-host.hleclerc-PC.ingenidev

DocumentRoot "C:/Dev/Projets/WampServer2-64b/install\_files\_wampserver2/bin/apache/Apache2.2.17/docs/dummy-host.hleclerc-PC.ingenidev"

ServerName dummy-host.hleclerc-PC.ingenidev

ServerAlias www.dummy-host.hleclerc-PC.ingenidev

ErrorLog "logs/dummy-host.hleclerc-PC.ingenidev-error.log"

CustomLog "logs/dummy-host.hleclerc-PC.ingenidev-access.log" common

</VirtualHost>

<VirtualHost \*:80>

ServerAdmin webmaster@dummy-host2.hleclerc-PC.ingenidev

DocumentRoot "C:/Dev/Projets/WampServer2-64b/install\_files\_wampserver2/bin/apache/Apache2.2.17/docs/dummy-host2.hleclerc-PC.ingenidev"

ServerName dummy-host2.hleclerc-PC.ingenidev

ErrorLog "logs/dummy-host2.hleclerc-PC.ingenidev-error.log"

CustomLog "logs/dummy-host2.hleclerc-PC.ingenidev-access.log" common

</VirtualHost>



修改成：

<VirtualHost \*:80>

ServerAdmin webmaster@163.com

DocumentRoot "/www/apache/htdocs"

ServerName portalman.gd.chinavent.com

ErrorLog "logs/dummy-host.localhost-error.log"

CustomLog "logs/dummy-host.localhost-access.log" combined

<Directory /www/apache/htdocs >

SetEnvIf Host "portalman.gd.chinavnet.com" gd=1

Order allow,deny

Allow from env=gd

</Directory>

</VirtualHost>

根据现网环境需要修改红色标志内容

<VirtualHost \*:80>

网站服务端口

DocumentRoot "/www/apache/htdocs"

网站目录绝对路径

ServerName portalman.gd.chinavent.com

网站域名

<Directory /www/apache/htdocs >

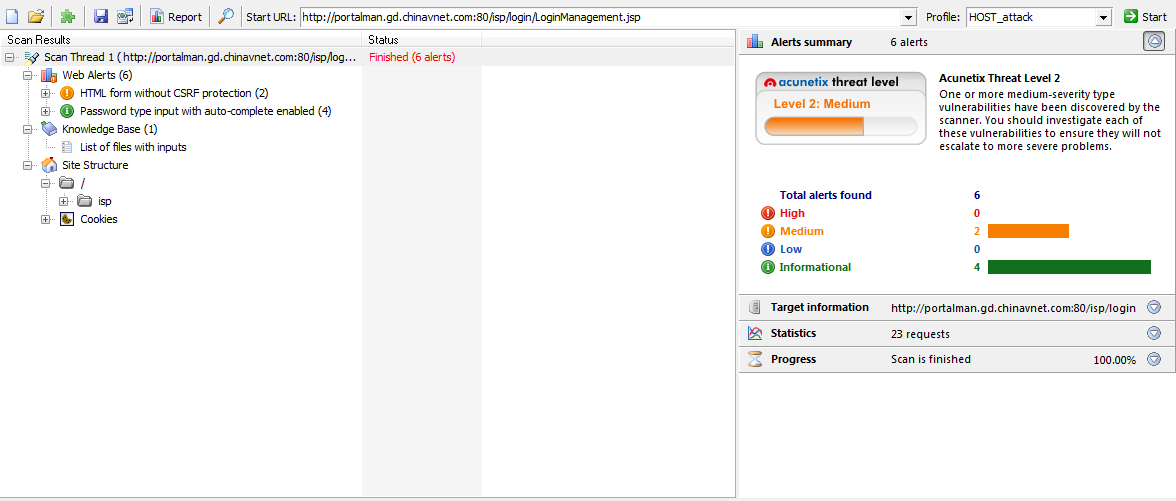
网站目录绝对路径

SetEnvIf Host "portalman.gd.chinavnet.com" gd=1

网站域名

配置完毕 重启WEB服务即可。

## 加固后复扫结果



## 参考：关于Apache服务器的访问控制

Apache的访问控制指对任何资源的任何方式的访问控制。  
一、基于主机或者IP地址的控制  
这种访问控制基于访问者的主机名或者IP地址，通过使用 Deny 和 Allow 指令，实现允许或者禁止某个主机访问我们的服务器资源。通常 Order 指令也会一起使用，来定义 Deny 和 Allows 指令起作用的顺序。如果不使用 Order 指令，默认的顺序为 Deny, Allow， 就相当于 Order Deny,Allow。Order 指令和 Allow，Deny 指令配合使用，实现了一个三步控制系统。  
第一步：根据Order的顺序，将顺序在前的所有的Allow指令或者Deny指令应用于当前请求；也就是说如果是 Order allow,deny ，那么第一步就是将所有的Allow指令应用于当前访问，如果有匹配，那么就允许该访问；如果沒有匹配到一条Allow指令，那么就禁止其访问；  
第二步：将剩下的另外一个指令的所有语句跟当前请求匹配，如果有匹配就执行相应的访问控制；  
第三步：如果当前请求没有在前两步匹配到任何指令，就执行 Order 指令中后面的那个指令。  
下面，通过一个例子理解一下：  
使指定目录下的资源只让本地访问  
     <Directory "/server">  
        Order allow,deny  
        Allow from 127.0.0.1  
    </Directory>  
 第一步： 本地的访问匹配了 Allow from 127.0.0.1 这条指令，所以被允许；其他的主机没有匹配到任何一条Allow指令，所以被禁止访问；  
第二步：所有访问都没有匹配到 Deny 指令；  
第三步：没有匹配到任何指令的访问，也就是不是本地的访问，按照 Order 指令，执行后面的 deny 的指令，所以被禁止访问；  
二、根据环境变量的访问控制  
这种方式可以通过 Allow from env= 或者 Deny from env= 语法实现，比如只允许使用火狐浏览器的客户访问：  
     <Directory "/server">  
        SetEnvIf User-Agent "Firefox" ff=1  
        Order allow,deny  
        Allow from env=ff  
    </Directory> 三、使用 mod\_rewrite 实现访问控制  
通过 mod\_rewrite 指令的 [F] 标志，可以基于任何的标准对一个资源实现访问控制。  
比如，我希望早八点到晚六点的时候才能访问任何资源，我就可以这样：  
 RewriteEngine On  
RewriteCond %{TIME\_HOUR} >20 [OR]  
RewriteCond %{TIME\_HOUR} <07  
RewriteRule ^/fridge - [F]  这样的话，晚上八点以后到早上7点，都会返回 403