Apusic 金蝶天燕

生成时间：2026-05-19


# 安全扫描说明

金蝶Apusic应用服务器V10

## 版权声明

本文档所涉及的软件著作权、版权等知识产权已依法进行了注册，由金蝶天燕云计算股份有限公司合法拥有。受《中华人民共和国著作权法》《计算机软件保护条例》《知识产权保护条例》和相关国际版权条约、法律、法规以及其它知识产权法律和条约的保护。未经授权许可，不得非法使用。

## 免责声明

本文档包含的版权信息由金蝶天燕云计算股份有限公司合法拥有，受法律的保护，金蝶天燕云计算股份有限公司对本文档可能涉及到的非金蝶天燕云计算股份有限公司的信息不承担任何责任。在法律允许的范围内，您可以查阅并仅能够在《中华人民共和国著作权法》规定的合法范围内复制和打印本文档。任何单位和个人未经金蝶天燕云计算股份有限公司书面授权许可，不得使用、修改、再发布本文档的任何部分和内容，否则将被视为侵权，金蝶天燕云计算股份有限公司有依法追究其责任的权利。

本文档如有更新，不另行通知。对本文档中的问题您可向金蝶天燕云计算股份有限公司告知或查询。未经本公司明确授予的任何权利均予保留。

## 商标声明

Apusic是深圳市金蝶天燕云计算股份有限公司向中华人民共和国国家商标局申请注册的注册商标，注册商标专用权由金蝶天燕合法拥有，受法律保护。未经金蝶天燕的书面许可，任何单位及个人不得以任何方式或理由对该商标的任何部分进行使用、复制、修改、传播、抄录或与其它产品捆绑使用销售。凡侵犯金蝶天燕商标权的，金蝶天燕将依法追究其法律责任。本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

# 目录

# 1 基础介绍

在强安全环境下安装使用，或使用APPSCAN、绿盟等安全扫描工具前，需要对金蝶Apusic应用服务器部分配置作修改。本文档面向对象为性能开发工程师、安全开发工程师、测试工程师、技术运维工程师等。

# 2 安全加固配置

通常情况下，金蝶Apusic应用服务器分为两大部分的安全配置，一部分为应用服务器的管控平台，一部分为应用配置。下面根据这两部分配置安全加固说明。

## 2.1 应用服务器的管控平台常见安全配置

配置应用服务器的管控配置安全性，通常是在检测应用服务器本身，或检测应用服务器的监听端口（默认为6848）时配置。

### 2.1.1 加密会话（SSL）Cookie 中缺少 Secure 属性

1、解决"**加密会话（SSL）Cookie 中缺少 Secure 属性**"问题，进入金蝶Apusic应用服务器安装目录 `lib/install/applications/_admingui/WEB-INF/aas-web.xml` ，添加以下部分内容：

```
        <cookie-properties>
            <property name="SameSite" value="Strict"/>
        </cookie-properties>
```

```
18      <session-config>
19          <session-manager>
20              <manager-properties>
21                  <property name="sessionFilename" value=""/>
22              </manager-properties>
23          </session-manager>
24          <cookie-properties>
25          <property name="SameSite" value="Strict"/>
26      </cookie-properties>
27      </session-config>
28
```

2、解决"**加密会话（SSL）Cookie 中缺少 Secure 属性**"问题，修改 安装目录 `/lib/install/applications/_admingui/WEB-INF/web.xml` ；将 `<secure>false</secure>` 设置为true

```
<session-config>
    <cookie-config>
        <http-only>true</http-only>
        <secure>true</secure>
    </cookie-config>
</session-config>
```

### 2.1.2 跨站脚本攻击、跨站请求伪造

解决"**跨站脚本攻击、跨站请求伪造**"问题，修改 安装目录 `/lib/install/applications/_admingui/WEB-INF/web.xml` ，取消以下部分的注释：

```
<filter>
        <filter-name>XSSFilter</filter-name>
        <filter-class>com.sun.webui.jsf.util.XSSFilter</filter-class>
    </filter>
    <filter>
        <filter-name>CSRFFilter</filter-name>
        <filter-class>com.sun.webui.jsf.util.CSRFFilter</filter-class>
        <init-param>
            <param-name>enabled</param-name>
            <param-value>true</param-value>
        </init-param>
    </filter>

<filter-mapping>
        <filter-name>XSSFilter</filter-name>
        <url-pattern>/*</url-pattern>
    </filter-mapping>
    <filter-mapping>
```

```
    <filter-name>CSRFFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

```
        <init-param>
            <param-name>maxSize</param-name>
            <param-value>-1</param-value>
        </init-param>
    </filter>
    <!-- 安全测试时放开这段代码
    <filter>
        <filter-name>XSSFilter</filter-name>
        <filter-class>com.sun.webui.jsf.util.XSSFilter</filter-class>
    </filter>
    <filter>
        <filter-name>CSRFFilter</filter-name>
        <filter-class>com.sun.webui.jsf.util.CSRFFilter</filter-class>
        <init-param>
            <param-name>enabled</param-name>
            <param-value>true</param-value>
        </init-param>
    </filter>
    -->



    <filter-mapping>
        <filter-name>UploadFilter</filter-name>
        <servlet-name>FacesServlet</servlet-name>
    </filter-mapping>
    <!-- 安全测试时放开这段代码
    <filter-mapping>
        <filter-name>XSSFilter</filter-name>
        <url-pattern>/*</url-pattern>
    </filter-mapping>
    <filter-mapping>
        <filter-name>CSRFFilter</filter-name>
        <url-pattern>/*</url-pattern>
    </filter-mapping>
    -->
```

#### 2.1.3 unix文件参数变更问题

修改 安装目录 `/lib/install/applications/_admingui/WEB-INF/web.xml`，添加以下部分，解决unix文件参数变更问题。

```
<context-param>
    <param-name>enableUnixFileFilter</param-name>
    <param-value>true</param-value>
</context-param>

<filter>
    <filter-name>UnixFileFilter</filter-name>
    <filter-class>org.apache.catalina.filters.UnixFileFilter</filter-class>
</filter>
<filter-mapping>
    <filter-name>UnixFileFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

```
    <context-param>
        <param-name>enableUnixFileFilter</param-name>
        <param-value>true</param-value>
    </context-param>

    <filter>
        <filter-name>UnixFileFilter</filter-name>
        <filter-class>org.apache.catalina.filters.UnixFileFilter</filter-class>
    </filter>
    <filter-mapping>
        <filter-name>UnixFileFilter</filter-name>
        <url-pattern>/*</url-pattern>
    </filter-mapping>

   <servlet>
      <servlet-name>FacesServlet</servlet-name>
      <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
      <load-on-startup>1</load-on-startup>
   </servlet>
   <servlet>
      <servlet-name>ThemeServlet</servlet-name>
      <servlet-class>com.sun.webui.theme.ThemeServlet</servlet-class>
      <load-on-startup>2</load-on-startup>
   </servlet>
   <servlet>
   <servlet-name>DownloadServlet</servlet-name>
   <servlet-class>com.apusic.aas.admingui.common.servlet.DownloadServlet</servlet-class>
```

#### 2.1.4 应用服务器管控平台验证码任意输入

设置验证码任意输入，修改 安装目录 `/lib/install/applications/_admingui/WEB-INF/web.xml` ，将以下值设置为true。

```
<context-param>
      <param-name>com.apusic.CAPTCHA_DISABLED</param-name>
      <param-value>false</param-value>
   </context-param>
```

#### 2.1.5 解决header安全问题

##### 2.1.5.1 管控设置

解决"**解决header安全问题**"，如果是扫描AAS管控平台，修改【sec-admin-listener】。

1）在【sever-config】-【网络配置】-【协议配置】-【sec-admin-listener】添加以下http请求头

| | | |
|---|---|---|
| ☐ | X-Content-Type-Options | nosniff |
| ☐ | X-XSS-Protection | 1;mode=block |
| ☐ | Cache-Control | no-cache |
| ☐ | Cache-Control | no-store |
| ☐ | Pragma | no-cache |
| ☐ | Strict-Transport-Security | max-age=31 |

##### 2.1.5.2 配置文件domain.xml设置

在配置文件 `domain.xml` 中对应的协议， `sec-admin-listener` ，添加以下属性。

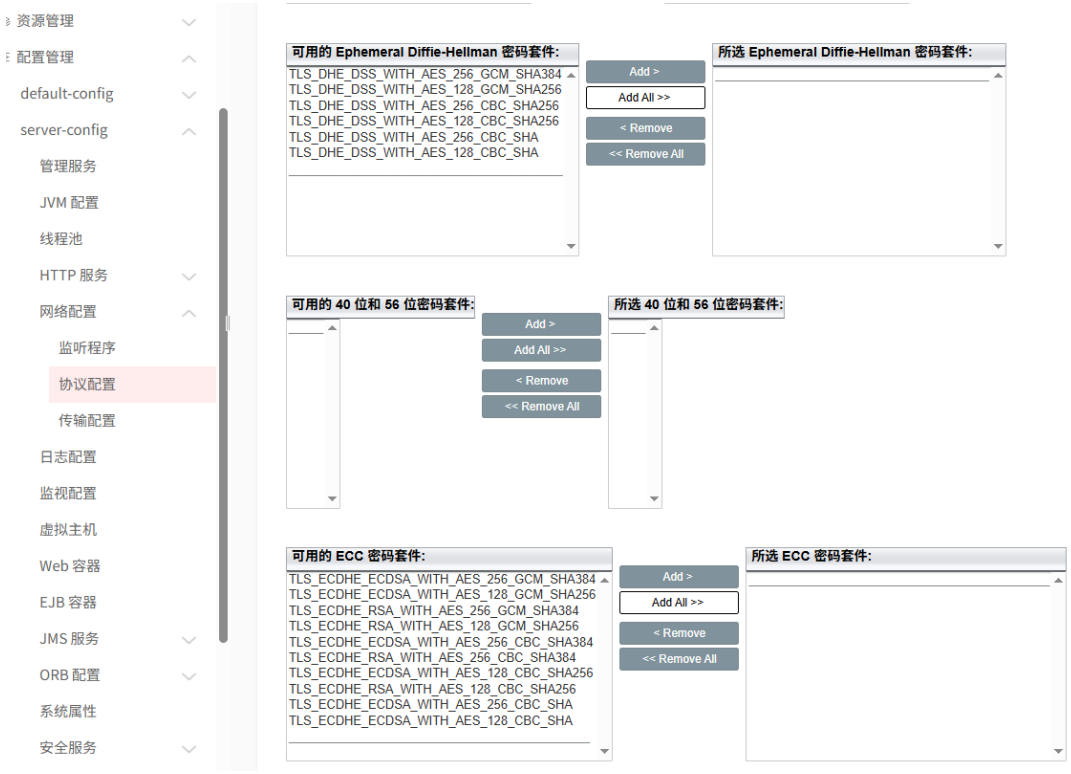
```
<protocol name="sec-admin-listener" security-enabled="true">
<http encoded-slash-enabled="true" default-virtual-server="__asadmin">
  <property name="X-XSS-Protection" value="1;mode=block"></property>
```

```
  <property name="Cache-Control" value="no-store"></property>
  <property name="Pragma" value="no-cache"></property>
  <property name="Strict-Transport-Security" value="max-age=3156000"></property>
  <property name="X-Content-Type-Options" value="nosniff"></property>
  <property name="X-Frame-Options" value="SAMEORIGIN"></property>
  <file-cache></file-cache>
</http></protocol>
```

### 2.1.6 解决不安全的ssl密码套件问题

#### 2.1.6.1 管控设置

**解决不安全的ssl密码套件问题**，选择合适的ssl密码套件（左边是表示过时的）。默认情况下不需要调整，如果选择了以下的密码套件还扫出有其他存在风险的密码套件，可根据实际再调整。




#### 2.1.6.2 配置文件domain.xml设置

在配置文件 `domain.xml` 中对应的协议， `sec-admin-listener` 中的 `ssl` 配置，设置 `ssl3-tls-ciphers` 。默认情况下不需要调整，如果选择了以下的密码套件还扫出有其他存在风险的密码套件，可根据实际再调整。

```
<protocol name="sec-admin-listener" security-enabled="true">
<http encoded-slash-enabled="true" default-virtual-server="__asadmin">
  <property name="X-XSS-Protection" value="1;mode=block"></property>
  <property name="Cache-Control" value="no-store"></property>
  <property name="Pragma" value="no-cache"></property>
  <property name="Strict-Transport-Security" value="max-age=3156000"></property>
  <property name="X-Content-Type-Options" value="nosniff"></property>
  <property name="X-Frame-Options" value="SAMEORIGIN"></property>
  <file-cache></file-cache>
</http>
<ssl tls-enabled="false" classname="com.sun.enterprise.security.ssl.ApusicSSLImpl" client-auth="want"
tls11-enabled="false" cert-nickname="kaas"
ssl3-tls-
ciphers="+TLS_DHE_DSS_WITH_AES_256_CBC_SHA256,+TLS_DHE_DSS_WITH_AES_256_CBC_SHA,+TLS_DHE_DSS_WITH_AES_128_CB
+TLS_DHE_DSS_WITH_AES_256_GCM_SHA384,+TLS_DHE_DSS_WITH_AES_128_GCM_SHA256,+TLS_DHE_DSS_WITH_AES_128_CBC_SHA2
```

```
+TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384,+TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384,+TLS_ECDH_ECDSA_WITH_AES_256_C
+TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256,+TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256,+TLS_ECDH_ECDSA_WITH_AES_256_C
+TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384,+TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256,+TLS_ECDH_RSA_WITH_AES_128_GCM
+TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,+TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,+TLS_ECDHE_ECDSA_WITH_AES_25
+TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,+TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,+TLS_ECDHE_ECDSA_WITH_AES_12
+TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,+TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,+TLS_ECDHE_RSA_WITH_AES_25
+TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,+TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA"></ssl></protocol>
```

### 2.1.7 有安全风险的TLS协议

#### 2.1.7.1 管控设置

**解决有安全风险的TLS协议问题**，只选择TLS1.2。


#### 2.1.7.2 配置文件domain.xml设置

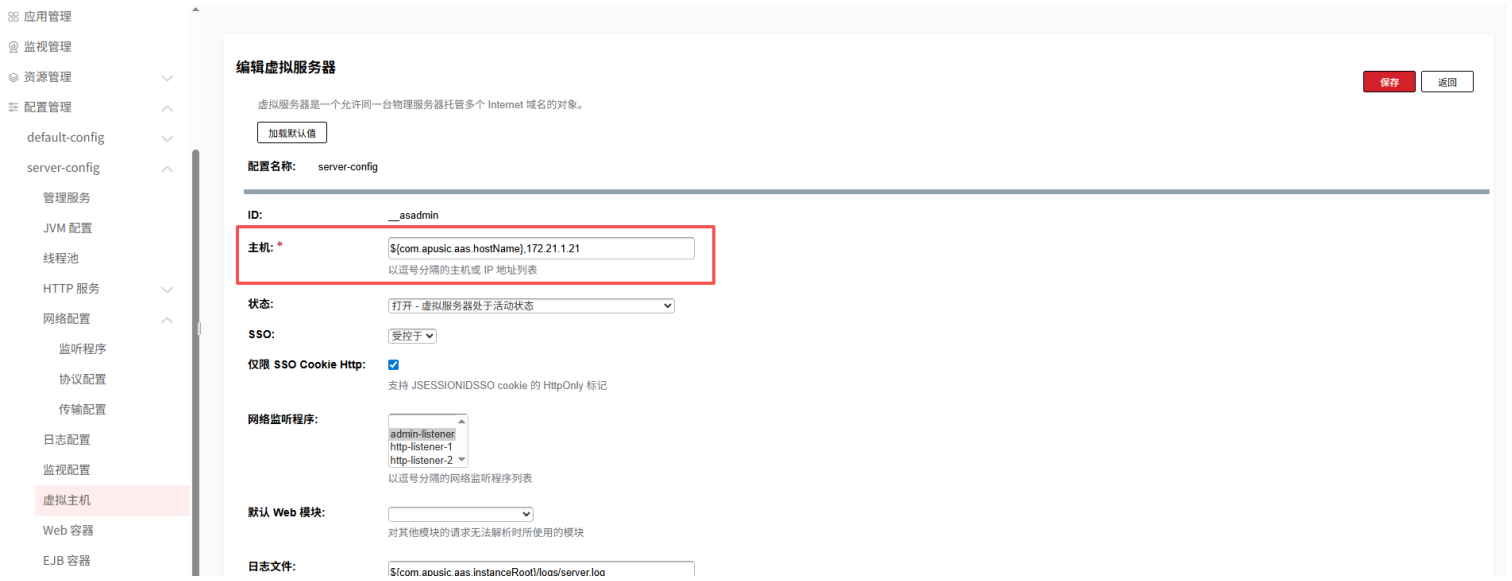
在配置文件 `domain.xml` 中对应的协议， `sec-admin-listener` 中的 `ssl` 配置，设置 `tls-enabled="false"` ， `tls11-enabled="false"` 。默认情况下不需要调整，如果设置之后还扫出有其他存在风险，可根据实际再调整。

```
<protocol name="sec-admin-listener" security-enabled="true">
<http encoded-slash-enabled="true" default-virtual-server="__asadmin">
  <property name="X-XSS-Protection" value="1;mode=block"></property>
  <property name="Cache-Control" value="no-store"></property>
  <property name="Pragma" value="no-cache"></property>
  <property name="Strict-Transport-Security" value="max-age=3156000"></property>
  <property name="X-Content-Type-Options" value="nosniff"></property>
  <property name="X-Frame-Options" value="SAMEORIGIN"></property>
  <file-cache></file-cache>
</http>
<ssl tls-enabled="false" classname="com.sun.enterprise.security.ssl.ApusicSSLImpl" client-auth="want"
tls11-enabled="false" cert-nickname="kaas"
ssl3-tls-
ciphers="+TLS_DHE_DSS_WITH_AES_256_CBC_SHA256,+TLS_DHE_DSS_WITH_AES_256_CBC_SHA,+TLS_DHE_DSS_WITH_AES_128_CB
+TLS_DHE_DSS_WITH_AES_256_GCM_SHA384,+TLS_DHE_DSS_WITH_AES_128_GCM_SHA256,+TLS_DHE_DSS_WITH_AES_128_CBC_SHA2
+TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384,+TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384,+TLS_ECDH_ECDSA_WITH_AES_256_C
+TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256,+TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256,+TLS_ECDH_ECDSA_WITH_AES_256_C
+TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384,+TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256,+TLS_ECDH_RSA_WITH_AES_128_GCM
+TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,+TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,+TLS_ECDHE_ECDSA_WITH_AES_25
+TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,+TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,+TLS_ECDHE_ECDSA_WITH_AES_12
+TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,+TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,+TLS_ECDHE_RSA_WITH_AES_25
+TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,+TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA"></ssl></protocol>
```

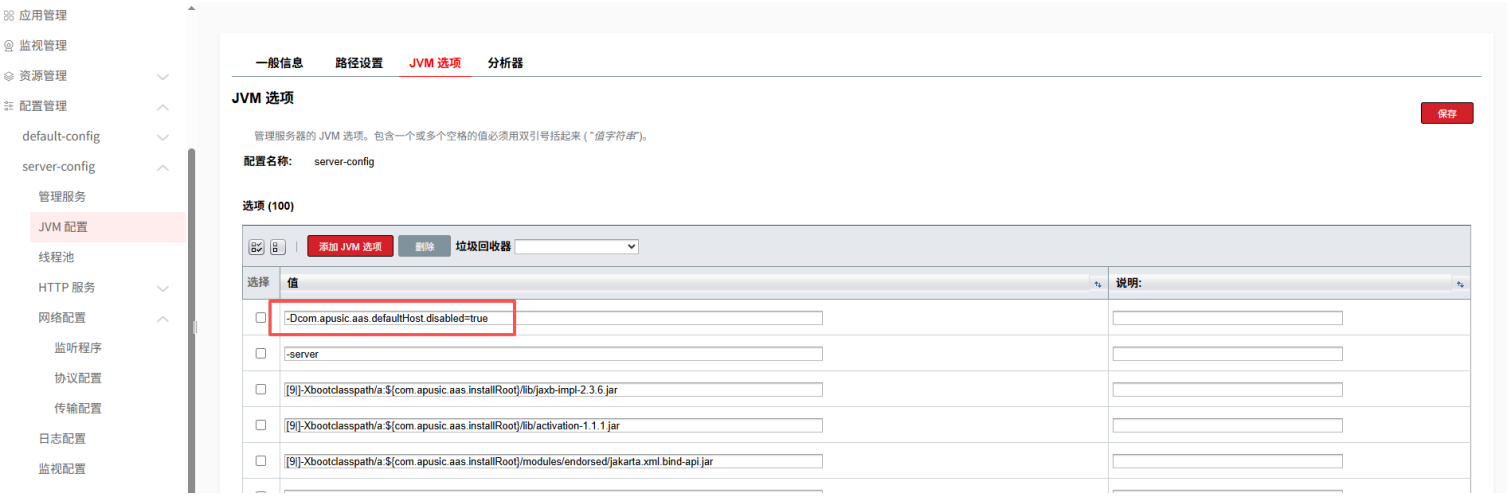
### 2.1.8 配置host 请求头攻击防御方法

### 2.1.8.1 管控设置

1.进入【虚拟主机】，进入 `--asadmin` 编辑页面，在"主机"中添加应用服务器的IP或域名。




2.添加JVM选项 `-Dcom.apusic.aas.defaultHost.disabled=true` 。




### 2.1.8.2 配置文件domain.xml设置

1.在 `__asadmin` 虚拟主机中添加hosts 的值有应用服务器的ip或域名

```
<config name="server-config">
  <system-property name="JMS_PROVIDER_PORT" description="Port Number that JMS Service will listen for remote clients connection." value="6876"></system-property>
  <http-service>
    <access-log></access-log>
    <virtual-server network-listeners="http-listener-1,http-listener-2" id="server" docroot="${com.apusic.aas.instanceRoot}/ROOT"></virtual-server>
    <virtual-server network-listeners="admin-listener" hosts="${com.apusic.aas.hostName},172.21.1.21" id="__asadmin"></virtual-server>
  </http-service>
  <iiop-service>
    <orb use-thread-pool-ids="thread-pool-1"></orb>
    <iiop-listener address="127.0.0.1" port="6837" lazy-init="true" id="orb-listener-1"></iiop-listener>
    <iiop-listener address="127.0.0.1" port="6838" id="SSL" security-enabled="true">
```

2.添加JVM选项 `-Dcom.apusic.aas.defaultHost.disabled=true`

```
</security-service>
<java-config classpath-suffix="" debug-options="-agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=8000" system-classpath="">
  <jvm-options>-Dcom.apusic.aas.defaultHost.disabled=true</jvm-options>
  <jvm-options>-server</jvm-options>
  <jvm-options>[9|]-Xbootclasspath/a:${com.apusic.aas.installRoot}/lib/jaxb-impl-2.3.6.jar</jvm-options>
  <jvm-options>[9|]-Xbootclasspath/a:${com.apusic.aas.installRoot}/lib/activation-1.1.1.jar</jvm-options>
  <jvm-options>[9|]-Xbootclasspath/a:${com.apusic.aas.installRoot}/modules/endorsed/jakarta.xml.bind-api.jar</jvm-options>
  <jvm-options>[9|]--add-opens=java.base/sun.net.www=ALL-UNNAMED</jvm-options>
  <jvm-options>[9|]--add-opens=java.base/sun.security.util=ALL-UNNAMED</jvm-options>
  <jvm-options>[9|]--add-opens=java.base/sun.security.provider=ALL-UNNAMED</jvm-options>
```

#### 2.1.9 http请求中存在请求头漏洞问题

解决**http请求中存在请求头漏洞问题**，在配置文件 `domain.xml` 中关闭http请求自动跳转至https。

```
  </protocol>
  <protocol name="sec-admin-listener" security-enabled="true">
    <http http2-enabled="false" encoded-slash-enabled="true" default-virtual-server="__asadmin">
      <file-cache></file-cache>
    </http>
    <ssl classname="com.sun.enterprise.security.ssl.ApusicSSLImpl" client-auth="want" cert-nickname="kaas"></ssl>
  </protocol>
  <protocol name="admin-http-redirect">
    <http-redirect secure="true"></http-redirect>
  </protocol>
  <protocol name="pu-protocol">
    <port-unification>
      <protocol-finder protocol="sec-admin-listener" classname="com.apusic.aas.grizzly.config.portunif.HttpProtocolFinder" name="http-finder"></protocol-finder>
      <protocol-finder protocol="admin-http-redirect" classname="com.apusic.aas.grizzly.config.portunif.HttpProtocolFinder" name="admin-http-redirect"></protocol-finder>
    </port-unification>
  </protocol>
</protocols>
<network-listeners>
  <network-listener protocol="http-listener-1" port="6888" name="http-listener-1" thread-pool="http-thread-pool" transport="tcp"></network-listener>
  <network-listener protocol="http-listener-2" port="6887" name="http-listener-2" thread-pool="http-thread-pool" transport="tcp"></network-listener>
```

## 2.2 应用常见安全配置

配置应用的安全性，通常是在用户部署在金蝶Apusic应用服务器中的应用程序，或检测应用程序的监听端口（默认为6887/6888）时配置。金蝶Apusic应用服务器出厂带有的监听程序 `http-listener-1` （默认监听端口6888）是没有开启"安全性"功能的，即没有使用https协议；因而通常建议使用监听程序 `http-listener-2` （默认监听端口6887）。

#### 2.2.1 加密会话（SSL）Cookie 中缺少 Secure 属性

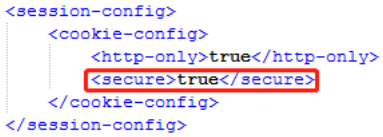
1、解决"**加密会话（SSL）Cookie 中缺少 Secure 属性**"问题，在应用程序中 `WEB-INF` 下添加 `apusic-web.xml` 文件，或在金蝶Apusic应用服务器 `${DOMAIN_HOME}/config/` 下添加 `default-apusic-web.xml` 文件（表示全局设置），文件内容如下：

```
    <?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE aas-web-app PUBLIC "-//Apusic.com//DTD Apusic Application Server 10.1 Servlet 3.0//EN"
"http://apusic.com/dtds/apusic-web-app_3_0-1.dtd">
<aas-web-app error-url="">
    <session-config>
    <cookie-properties>
        <property name="SameSite" value="Strict"/>
        </cookie-properties>
    </session-config>
</aas-web-app>
```

```
18    <session-config>
19        <session-manager>
20            <manager-properties>
21                <property name="sessionFilename" value=""/>
22            </manager-properties>
23        </session-manager>
24        <cookie-properties>
25        <property name="SameSite" value="Strict"/>
26    </cookie-properties>
27    </session-config>
28
```

2、解决"**加密会话（SSL）Cookie 中缺少 Secure 属性**"问题，应用程序的目录 `/WEB-INF/web.xml` ，或在金蝶Apusic应用服务器 `${DOMAIN_HOME}/config/default-web.xml` （表示全局设置），设置 `<cookie-config>` ，例如：

```
  <session-config>
    <cookie-config>
        <http-only>true</http-only>
        <secure>true</secure>
    </cookie-config>
  </session-config>
```




#### 2.2.2 跨站请求伪造

解决"**跨站请求伪造**"问题，应用程序中的 `web.xml` ，或在金蝶Apusic应用服务器 `${DOMAIN_HOME}/config/default-web.xml` （全局配置），配置CsrfFilter：

```
<filter>
    <filter-name>CsrfFilter</filter-name>
    <filter-class>org.apache.catalina.filters.CsrfFilter</filter-class>
    <init-param>
      <param-name>ignoreMethods</param-name>
      <param-value>GET</param-value>
    </init-param>
    <init-param>
      <param-name>ignoreUrls</param-name>
      <param-value>/,/common/index.jsf</param-value>
    </init-param>
    <init-param>
      <param-name>allowHostPattern</param-name>
      <param-value>127\.0\.0\.1|localhost</param-value>
    </init-param>
</filter>
<filter-mapping>
    <filter-name>CsrfFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

#### 2.2.3 跨站脚本攻击

解决"**跨站脚本攻击**"问题，应用程序中的 `web.xml` ，或在金蝶Apusic应用服务器 `${DOMAIN_HOME}/config/default-web.xml` （全局配置），配置XssFilter：

```
<filter>
    <filter-name>XssFilter</filter-name>
    <filter-class>org.apache.catalina.filters.XssFilter</filter-class>
    <init-param>
      <param-name>ignoreUrls</param-name>
      <param-value>/</param-value>
    </init-param>
</filter>
<filter-mapping>
    <filter-name>XssFilter</filter-name>
```

```
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

#### 2.2.4 unix文件参数变更问题

在金蝶Apusic应用服务器 `${DOMAIN_HOME}/config/default-web.xml` ，添加以下部分内容，解决unix文件参数变更问题。

```
    <context-param>
        <param-name>enableUnixFileFilter</param-name>
        <param-value>true</param-value>
    </context-param>

    <filter>
        <filter-name>UnixFileFilter</filter-name>
        <filter-class>org.apache.catalina.filters.UnixFileFilter</filter-class>
    </filter>
    <filter-mapping>
        <filter-name>UnixFileFilter</filter-name>
        <url-pattern>/*</url-pattern>
    </filter-mapping>
```

#### 2.2.5 解决header安全问题

##### 2.2.5.1 管控设置

解决“解决header安全问题”，如果是扫描应用程序，修改【http-listener-2】；因为【http-listener-1】是没有配置SSL的，所以不建议使用【http-listener-1】做安全扫描。

1）在【sever-config】-【网络配置】-【协议配置】-【http-listener-2】添加以下http请求头.

| | | |
|---|---|---|
| ☐ | X-Content-Type-Options | nosniff |
| ☐ | X-XSS-Protection | 1;mode=block |
| ☐ | Cache-Control | no-cache |
| ☐ | Cache-Control | no-store |
| ☐ | Pragma | no-cache |
| ☐ | Strict-Transport-Security | max-age=31 |

##### 2.2.5.2 配置文件domain.xml设置

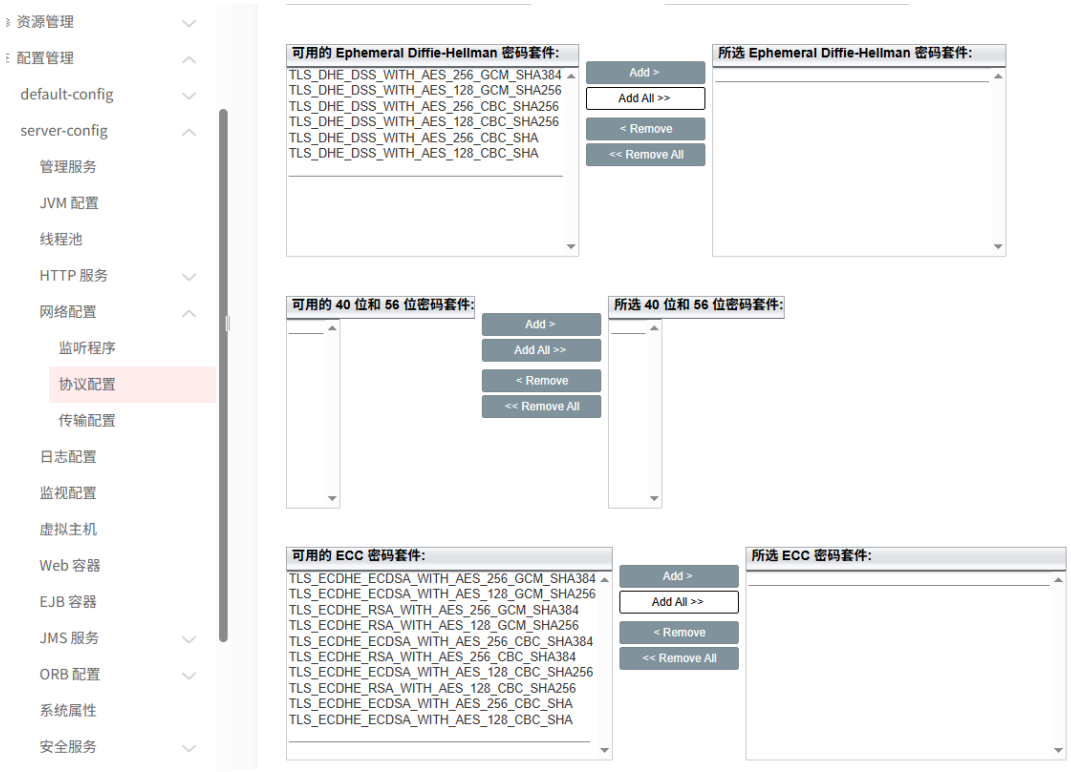
在配置文件 `domain.xml` 中对应的协议， `http-listener-2` ，添加以下属性。

```
<protocol name="http-listener-2" security-enabled="true">
<http encoded-slash-enabled="true" default-virtual-server="server">
  <property name="X-XSS-Protection" value="1;mode=block"></property>
  <property name="Cache-Control" value="no-store"></property>
  <property name="Pragma" value="no-cache"></property>
  <property name="Strict-Transport-Security" value="max-age=3156000"></property>
  <property name="X-Content-Type-Options" value="nosniff"></property>
  <property name="X-Frame-Options" value="SAMEORIGIN"></property>
  <file-cache></file-cache>
</http></protocol>
```

#### 2.2.6 解决不安全的ssl密码套件问题

##### 2.2.6.1 管控设置

**解决不安全的ssl密码套件问题**，选择合适的ssl密码套件（左边是表示过时的）。默认情况下不需要调整，如果选择了以下的密码套件还扫出有其他存在风险的密码套件，可根据实际再调整。




#### 2.2.6.2 配置文件domain.xml设置

在配置文件 `domain.xml` 中对应的协议， `http-listener-2` 中的 `ssl` 配置，设置 `ssl3-tls-ciphers` 。默认情况下不需要调整，如果选择了以下的密码套件还扫出有其他存在风险的密码套件，可根据实际再调整。

```
<protocol name="http-listener-2" security-enabled="true">
<http encoded-slash-enabled="true" default-virtual-server="server">
  <property name="X-XSS-Protection" value="1;mode=block"></property>
  <property name="Cache-Control" value="no-store"></property>
  <property name="Pragma" value="no-cache"></property>
  <property name="Strict-Transport-Security" value="max-age=3156000"></property>
  <property name="X-Content-Type-Options" value="nosniff"></property>
  <property name="X-Frame-Options" value="SAMEORIGIN"></property>
  <file-cache></file-cache>
</http>
<ssl tls-enabled="false" classname="com.sun.enterprise.security.ssl.ApusicSSLImpl" client-auth="want"
tls11-enabled="false" cert-nickname="kaas"
ssl3-tls-
ciphers="+TLS_DHE_DSS_WITH_AES_256_CBC_SHA256,+TLS_DHE_DSS_WITH_AES_256_CBC_SHA,+TLS_DHE_DSS_WITH_AES_128_CB
+TLS_DHE_DSS_WITH_AES_256_GCM_SHA384,+TLS_DHE_DSS_WITH_AES_128_GCM_SHA256,+TLS_DHE_DSS_WITH_AES_128_CBC_SHA2
+TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384,+TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384,+TLS_ECDH_ECDSA_WITH_AES_256_C
+TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256,+TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256,+TLS_ECDH_ECDSA_WITH_AES_256_C
+TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384,+TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256,+TLS_ECDH_RSA_WITH_AES_128_GCM
+TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,+TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,+TLS_ECDHE_ECDSA_WITH_AES_25
+TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,+TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,+TLS_ECDHE_ECDSA_WITH_AES_12
+TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,+TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,+TLS_ECDHE_RSA_WITH_AES_25
+TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,+TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA"></ssl></protocol>
```

### 2.2.7 有安全风险的TLS协议

#### 2.2.7.1 管控设置

**解决有安全风险的TLS协议问题**，只选择TLS1.2。


### 2.2.7.2 配置文件domain.xml设置

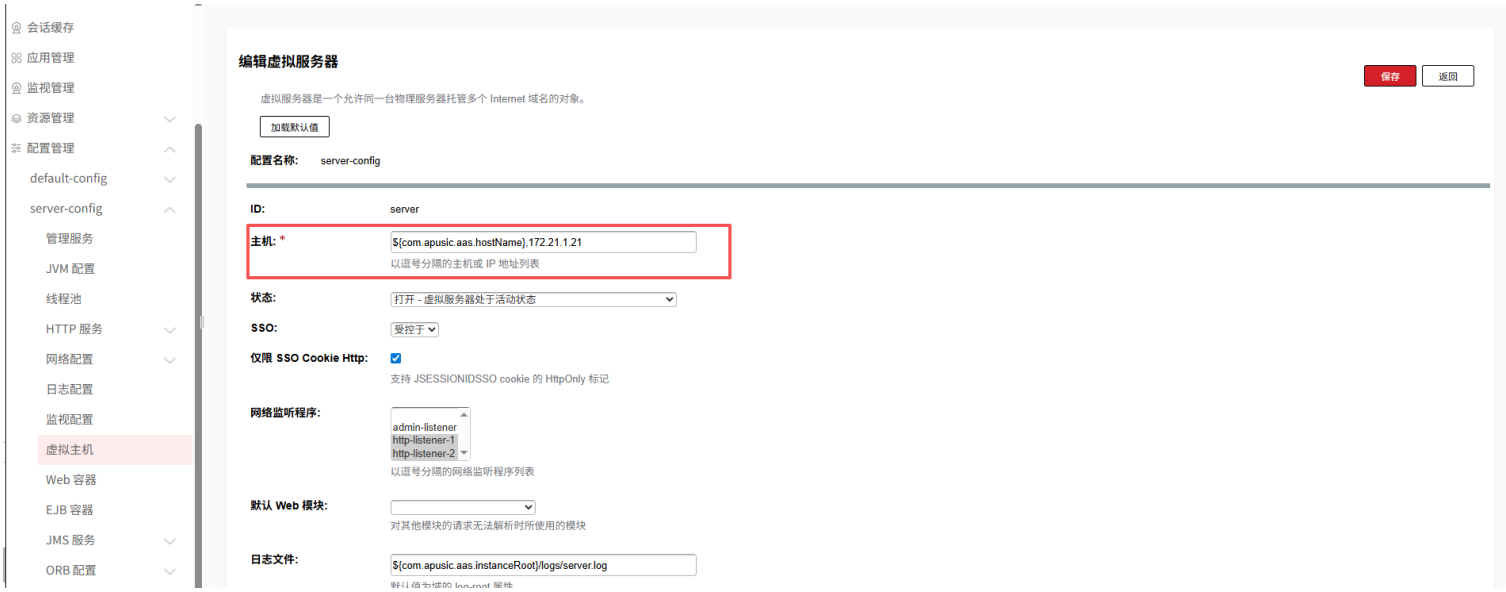
在配置文件 `domain.xml` 中对应的协议， `http-listener-2` 中的 `ssl` 配置，设置 `tls-enabled="false"` ， `tls11-enabled="false"` 。默认情况下不需要调整，如果设置之后还扫出有其他存在风险，可根据实际再调整。

```
<protocol name="http-listener-2" security-enabled="true">
<http encoded-slash-enabled="true" default-virtual-server="server">
  <property name="X-XSS-Protection" value="1;mode=block"></property>
  <property name="Cache-Control" value="no-store"></property>
  <property name="Pragma" value="no-cache"></property>
  <property name="Strict-Transport-Security" value="max-age=3156000"></property>
  <property name="X-Content-Type-Options" value="nosniff"></property>
  <property name="X-Frame-Options" value="SAMEORIGIN"></property>
  <file-cache></file-cache>
</http>
<ssl tls-enabled="false" classname="com.sun.enterprise.security.ssl.ApusicSSLImpl" client-auth="want"
tls11-enabled="false" cert-nickname="kaas"
ssl3-tls-
ciphers="+TLS_DHE_DSS_WITH_AES_256_CBC_SHA256,+TLS_DHE_DSS_WITH_AES_256_CBC_SHA,+TLS_DHE_DSS_WITH_AES_128_CB
+TLS_DHE_DSS_WITH_AES_256_GCM_SHA384,+TLS_DHE_DSS_WITH_AES_128_GCM_SHA256,+TLS_DHE_DSS_WITH_AES_128_CBC_SHA2
+TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384,+TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384,+TLS_ECDH_ECDSA_WITH_AES_256_C
+TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256,+TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256,+TLS_ECDH_ECDSA_WITH_AES_256_C
+TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384,+TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256,+TLS_ECDH_RSA_WITH_AES_128_GCM
+TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,+TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,+TLS_ECDHE_ECDSA_WITH_AES_25
+TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,+TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,+TLS_ECDHE_ECDSA_WITH_AES_12
+TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,+TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,+TLS_ECDHE_RSA_WITH_AES_25
+TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,+TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA"></ssl></protocol>
```

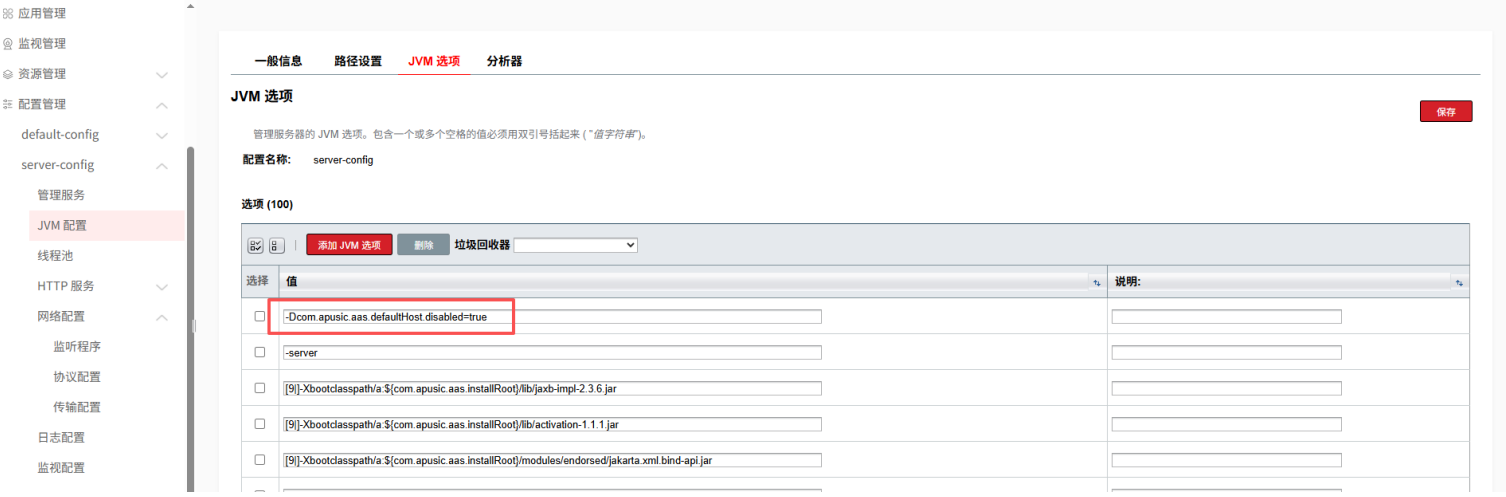
### 2.2.8 配置host 请求头攻击防御方法

### 2.2.8.1 管控设置

1.进入【虚拟主机】，进入 `server` 编辑页面，在"主机"中添加应用服务器的IP或域名。




2.添加JVM选项 `-Dcom.apusic.aas.defaultHost.disabled=true` 。




### 2.2.8.2 配置文件domain.xml设置

1.在 `server` 虚拟主机中添加hosts 的值有应用服务器的ip或域名。

```
<config name="server-config">
  <system-property name="JMS_PROVIDER_PORT" description="Port Number that JMS Service will listen for remote clients connection." value="6876"></system-property>
  <http-service>
    <access-log></access-log>
    <virtual-server network-listeners="http-listener-1,http-listener-2" hosts="${com.apusic.aas.hostName},172.21.1.21" id="server" docroot="${com.apusic.aas.instanceRoot}/ROOT"></virtual-server>
    <virtual-server network-listeners="admin-listener"  id="__asadmin"></virtual-server>
  </http-service>
  <iiop-service>
    <orb use-thread-pool-ids="thread-pool-1"></orb>
    <iiop-listener address="127.0.0.1" port="6837" lazy-init="true" id="orb-listener-1"></iiop-listener>
```

2.添加JVM选项 `-Dcom.apusic.aas.defaultHost.disabled=true`

```
    </security-service>
    <java-config classpath-suffix="" debug-options="-agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=8000" system-classpath="">
      <jvm-options>-Dcom.apusic.aas.defaultHost.disabled=true</jvm-options>
      <jvm-options>-server</jvm-options>
      <jvm-options>[9|]-Xbootclasspath/a:${com.apusic.aas.installRoot}/lib/jaxb-impl-2.3.6.jar</jvm-options>
      <jvm-options>[9|]-Xbootclasspath/a:${com.apusic.aas.installRoot}/lib/activation-1.1.1.jar</jvm-options>
      <jvm-options>[9|]-Xbootclasspath/a:${com.apusic.aas.installRoot}/modules/endorsed/jakarta.xml.bind-api.jar</jvm-options>
      <jvm-options>[9|]--add-opens=java.base/sun.net.www=ALL-UNNAMED</jvm-options>
      <jvm-options>[9|]--add-opens=java.base/sun.security.util=ALL-UNNAMED</jvm-options>
      <jvm-options>[9|]--add-opens=java.base/sun.security.provider=ALL-UNNAMED</jvm-options>
```

## 2.3 其他配置

### 2.3.1 防止信息泄露

金蝶Apusic应用服务器支持防止信息泄露，支持错误页面自定义，隐藏中间件类型和版本信息。

在【JVM设置】中设置JVM选项，开启自定义错误页面。需要设置 `-Dapusic.http.errorpage.custom.enabled=true` 开启自定义错误页面功能，设置为 `true` 后，需要添加 `-Dapusic.http.errorpage.custom.file` ，否则还是返回默认页面。

`-Dapusic.http.errorpage.custom.file` 后面加 `.[错误编码]` ，可以针对某一类型错误指定返回页面，例如 `-Dapusic.http.errorpage.custom.file.404` 表示404错误返回的页面。

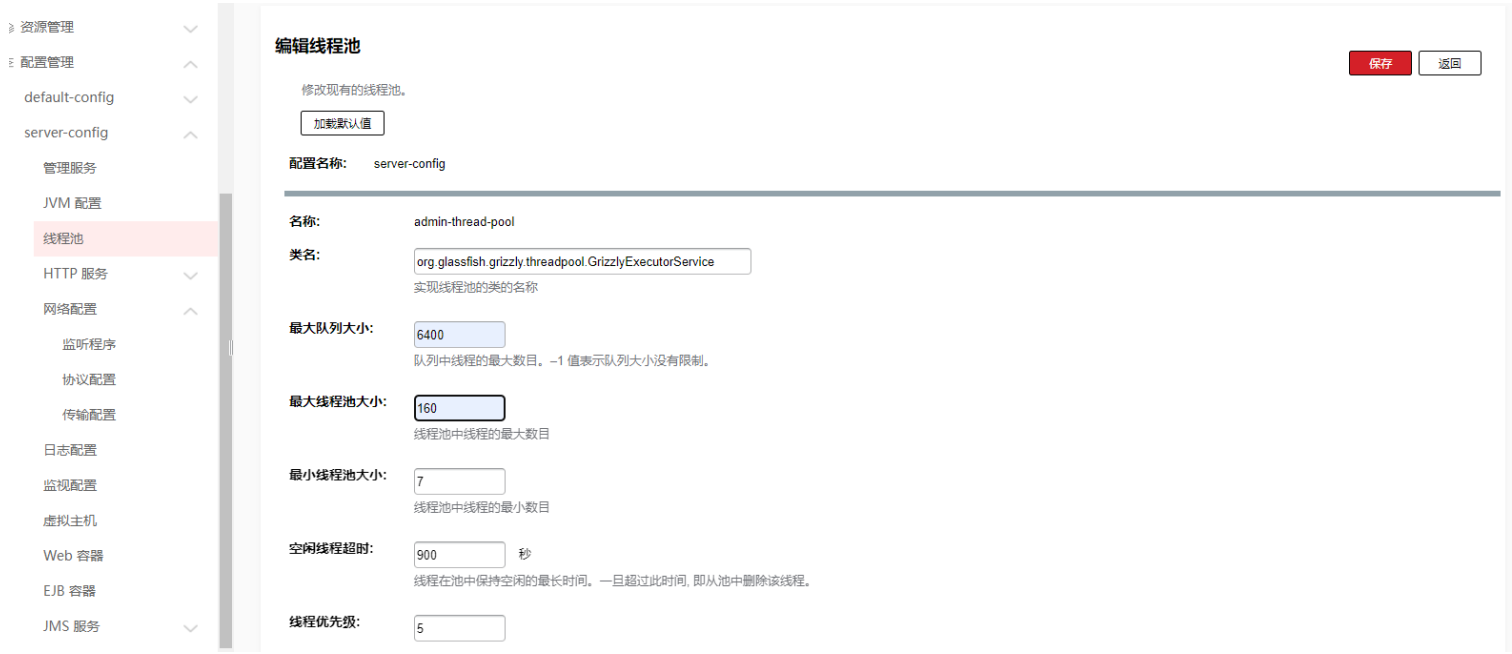
### 2.3.2 慢攻击检测

金蝶Apusic应用服务器支持开启慢攻击检测功能。

进入【传输配置】，进入编辑页面，勾选“慢攻击检测”，根据需要设置“慢攻击日志”、“连续写数据次数”、“写最小数据量”。

# 3 设置线程数

V10进行安全扫描时，由于网络延时、机器性能等，扫描时发送的大量线程数会无法及时处理，进而造成管控台无法通讯。需要进入管控台，例如扫描管控平台时设置【server-config】-【线程池】-【admin-thread-pool】，修改“最大队列数”为6400，修改“最大线程池”为160



# 4 客户端设置

在扫描前，例如使用Appscan，需要对Appscan进行配置

AAS管控台使用https，且包含动态验证码，因而设置【登录管理】-【会话标识】时，取消勾选"JSESSIONID"。

设置【通讯和代理】-【线程数】，设置为8；【超时】设置为90；勾选【不使用代理】

全国统一服务热线

4008-555-800

金蝶天燕云计算股份有限公司(简称“金蝶天燕云”)成立于2000年，前身为“金蝶中间件公司”，是金蝶集团旗下新一代软件基础云平台服务商，云计算国家标准制定企业，国家信创产业核心软件企业。金蝶天燕是国家863重点研发计划与核高基重大专项承接企业，也是“两网一站四库十二金”国家重点工程的基础平台提供商，产品广泛应用于政府、军工、金融、能源等关键行业，累计服务客户总数超过10万家。

Apusic 金蝶天燕

云计算国家标准制定企业

金蝶集团旗下基础软件企业

信息技术应用创新核心企业

官网：www.apusic.com