SSL证书卸载与SSI高级应用

http://netkiller.github.io/journal/ssi.html

Mr. Neo Chen (陈景峯), netkiller, BG7NYT

中国广东省深圳市龙华新区民治街道溪山美地
518131
+86 13113668890

<netkiller@msn.com>

文档出处:
http://netkiller.github.io
http://netkiller.sourceforge.net

微信扫描二维码进入 Netkiller 微信订阅号

QQ群：128659835 请注明“读者”

2014-09-17

摘要

目录

1. 什么是SSI(Server Side Include)

SSI是服务器端页面包含，SSI工作在web服务器上，web服务器可以在一个页面中包含另一个页面，在用户端看来是只有一个页面。

2. 为什么使用SSI

我们又很多个子站，所有网站的header与footer都相同，还有一些block区块也存在共用。所以我们将这个共用的部分拆分，然后使用SSI按需包含。

3. 谁来负责SSI制作

稍有经验的美工人员都可以灵活使用SSI，程序员也可在短时间内学会SSI.

4. 怎么处理SSI包含

4.1. SSI 目录规划

```
/www/example.com
  |-- inc.example.com
  |-- www.example.com
  |-- images.example.com
  |-- acc.example.com
```

inc.example.com 是SSI共用文件，存放shtml文件。

www.example.com 是主站，会用到inc.example.com中的公共模块。

acc.example.com 与 www.example.com 类似。

注意

/www/inc.example.com是公共目录，不需要配置nginx，不能通过浏览器访问到该目录.

为什么要独立公共文件，而不是放在/www/www.example.com目录下面呢？我是为了方便发布代码，分开的好处是我可以针对inc.example.com做发布，而不影响其他项目。

```
由于include作用于web服务器的$document_root目录，例如当前$document_root是/www/example.com/www.example.com


<!--#include file="/example.shtml"--> 会引用 /www/example.com/www.example.com/example.shtml 文件，而不是操作系统根目录。
```

所以我们无法引用与www.example.com同级别的inc.example.com公共文件。例如：

```
<!--#include file="/www/example.com/inc.example.com/example.shtml"--> 会引用 /www/example.com/www.example.com/www/example.com/inc.example.com/exam
<!--#include file="../inc.example.com/example.shtml"--> 会引用 也无法正常工作。
```

这是服务器限制，如果SSI可能包含$document_root之外的文件，将会带来安全问题，例如

```
<!--#include file="/etc/passwd"-->
```

怎样能突破限制呢？我想出了别名，通过别名/include引用/www/example.com/inc.example.com目录中的公文模块，例如：

```
    location /include/ {
        root   /www/example.com/inc.example.com;
    }
```

提示

Apache 与 Nginx 服务器的 SSI 实现稍有不同include file与include virtual也有差异。

4.2. www.example.com 静态内容服务器

```
# cat /etc/nginx/conf.d/www.example.com.conf

server {
    listen       80;
    server_name  www.example.com;

    charset utf-8;
    access_log  /var/log/nginx/www.example.com.access.log;
    error_log   /var/log/nginx/www.example.com.error.log;

    location / {
        root   /www/example.com/www.example.com;
        index  index.html;
    }

    location /include/ {
        root   /www/example.com/inc.example.com;
    }
    location /info/ {
        proxy_pass http://info.example.com/;
    }


    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }
}
```

4.3. acc.example.com 动态网页服务器

```
server {
    listen       80;
    server_name  acc.example.com;
    charset utf-8;
    access_log  /var/log/nginx/acc.example.com.access.log;
    error_log   /var/log/nginx/acc.example.com.error.log;

    set $X_FORWARDED_FOR $http_x_forwarded_for;
```

```
    location / {
        root   /www/example.com/acc.example.com/htdocs;
        index  index.php;

        try_files $uri $uri/ /index.php?/$request_uri;
    }

    location /include/ {
        root   /www/example.com/inc.example.com;
    }

    location ^~ /images/ {
        rewrite /images/(.+)$ /$1 break;
        proxy_pass http://images.example.com;
        break;
    }
    location ~ \.php$ {
        fastcgi_pass   127.0.0.1:9000;
        fastcgi_index  index.php;
        fastcgi_param  SCRIPT_FILENAME  /www/example.com/acc.example.com/htdocs/$fastcgi_script_name;
        include        fastcgi_params;
        fastcgi_param  DOCUMENT_ROOT /www/example.com/acc.example.com/htdocs;
    }
}
```

注意

该服务器不对外提供服务器，只允许下面的SSL卸载服务器通过反向代理连接

4.4. SSL卸载服务器

将SSL证书处理，机密与解密操作转移到该服务器，不让业务服务器处理证书的加密与解密操作，上面的HTTP对内访问，HTTPS对外访问，HTTPS通过反向代理连接HTTP服务器实现SSL证书卸载

```
upstream acc.example.com {
    server acc1.example.com;
    server acc2.example.com;
    server acc3.example.com;
}

server {
    listen       443;
    server_name  acc.example.com;

    ssl                  on;
    ssl_certificate      /etc/nginx/example.com/acc.example.com.pem;
    ssl_certificate_key  /etc/nginx/example.com/acc.example.com.key;

    ssl_session_timeout  5m;

    ssl_protocols  SSLv2 SSLv3 TLSv1;
    ssl_ciphers  HIGH:!aNULL:!MD5;
    ssl_prefer_server_ciphers   on;

    location / {
        proxy_pass http://acc.example.com;
        proxy_http_version 1.1;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        break;
    }
}
```

4.5. /www/inc.example.com 公共包含文件

/www/inc.example.com/include/cn/config.html

```
<!--#set var="HTML_HOST" value="http://www.example.com"-->
<!--#set var="INFO_HOST" value="http://info.example.com"-->
<!--#set var="NEWS_HOST" value="http://news.example.com"-->
<!--#set var="IMG_HOST" value="http://images.example.com"-->
<!--#set var="JS_HOST" value="http://images.example.com"-->
<!--#set var="CSS_HOST" value="http://images.example.com"-->

<!--#if expr="${X_FORWARDED_FOR}"-->

<!--#set var="ACC_HOST" value="https://myid.example.com"-->
<!--#set var="IMG_HOST" value="/images"-->
<!--#set var="JS_HOST" value="/images"-->
<!--#set var="CSS_HOST" value="/images"-->

<!--#else -->

<!--#set var="ACC_HOST" value="http://myid.example.com"-->
<!--#set var="IMG_HOST" value="http://images.example.com"-->
<!--#set var="JS_HOST" value="http://images.example.com"-->
```

```
<!--#set var="CSS_HOST" value="http://images.example.com"-->

<!--#endif -->
```

${X_FORWARDED_FOR} 用来判断用户是通过http还是https进入，由于images.example.com 没有SSL证书，需要有区分的载入图片的地址。/images 通过反向代理连接http://images.exampe.com.

4.6. 引用包含文件实例

```
<!--#include file="/include/cn/config.html"-->
<!DOCTYPE>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title></title>
<link rel="shortcut icon" href="<!--#echo var="IMG_HOST"-->/favicon.ico" type="image/x-icon" />
<link rel="stylesheet" href="<!--#echo var="CSS_HOST"-->/styles/common.css" />
<script type="text/javascript" src="<!--#echo var="JS_HOST"-->/scripts/jquery-1.7.1.min.js"></script>
</head>
<body>
<div id="homeNav"><!--#include virtual="/include/cn/header.html" --></div>
<a href="<!--#echo var='ACC_HOST'-->/register/" class="real">
        <h3><img src="<!--#echo var="IMG_HOST"-->/new/ico_real.png" />注册账户</h3>
</a>
</body>
</html>
```