9/30日 14:00 -10/4日 08:00暂时无法发布内容!

烟雨平生

首页 | 博文目录 | 关于我



博客访问: 1828 博文数量: 2 博客积分: 0 博客等级: 民兵 技术积分: 20 用户组:普通用户

注册时间: 2015-08-13 10:02



文章分类

全部博文(2) **★** linux(1) openssI(1) 未分配的博文(0)

文章存档

± 2015年(2)

我的朋友



推荐博文

·在 SCO OpenServer 6.0 D2M1. ·activiti 获取当前任务流程图..

OPENSSL 验证证书链

分类: C/C++ 2015-08-14 11:15:13

步骤:

- 1)初始化环境
- a.新建证书存储区X509_STORE_new()
- b.新建证书校验上下文X509_STORE_CTX_new()
- 2)导入根证书
- a.读取CA证书, 从DER编码格式化为X509结构d2i X509()
- b.将CA证书导入证书存储区X509_STORE_add_cert()
- 3)导入要校验的证书test
- a.读取证书test, 从DER编码格式化为X509结构d2i X509()
- b.在证书校验上下文初始化证书test,X509_STORE_CTX_init()
- c.校验X509_verify_cert

点击(此处)折叠或打开

```
#include <stdio.h>
      #include <string.h>
      #include <stdlib.h>
 5. #include <openssl/evp.h>
      #include <openss1/x509.h>
      #include <openssl/pem.h>
 7.
      #define CERT_PATH "/home/ycg/demoCA"
      #define ROOTCA_CERT "rootca_cert.pem"
10
      #define CLASS2CA_CERT "class2ca_cert.pem"
11.
12.
      #define CLIENT_CERT "client_cert.pem"
13.
14.
      #define GET_ROOT_CA_CERT(str) sprintf(str, "%s/%s", CERT_PATH, ROOTCA_CERT)
      #define GET_CLASS2_CA_CERT(str) sprintf(str, "%s/%s", CERT_PATH, CLASS2CA_CERT)
16.
      #define GET_CLIENT_CERT(str, path, name) sprintf(str, "%s/%s", path, name)
17
18.
19.
      #define MAX_LEGTH 4096
20.
21.
      int my_load_cert(unsigned char *str, unsigned long *str_len,
22.
                   const char *verify_cert, const unsigned int cert_len)
23.
24.
         FILE *fp;
25.
          fp = fopen(verify_cert, "rb");
26.
          if ( NULL == fp)
27.
28.
              fprintf(stderr, "fopen fail\n");
29.
              return -1;
30.
31.
32.
          *str_len = fread(str, 1, cert_len, fp);
          fclose(fp);
33.
```



·【PyTorch基础教程7】多维特... ·旋转算法 ·openstack注入文件到虚机源码...

相关博文

```
·人脸识别主板能应用哪些产品...
整车控制单元(VCU)
嵌入式开发与纯软件什么区别...
启明云端分享:S系列1.54寸串...
Rhapsody — MBSE 开发工具...
IPD的前世今生
香蕉派 BPI-M5单板计算机, 采...
自助售货机主板要注意哪几个...
·TAITherm—专业热管理工具...
```

· 启明云端分享: ESP32-C3环境...

```
34.
          return 0;
35. }
36.
37.
      X509 *der_to_x509(const unsigned char *der_str, unsigned int der_str_len)
39.
          X509 *x509:
 40.
          x509 = d2i_X509(NULL, &der_str, der_str_len);
 41.
          if ( NULL == x509 )
 42.
              fprintf(stderr, "d2i_X509 fail\n");
43.
 11
              return NULL;
45.
 46
          return x509;
47. }
 48.
49. X509 *pem_to_x509(const char *pem_file)
50.
 51.
          X509 *x509;
5.2
          BIO *cert = NULL;
          if ((cert = BIO_new(BIO_s_file())) == NULL) {
53.
 54.
              goto end;
 55.
 56.
 57.
          if (BIO_read_filename(cert, pem_file) <= 0) {</pre>
 58.
              goto end;
59.
 60.
61.
          x509 = PEM_read_bio_X509(cert, NULL, NULL, NULL);
62.
          if ( NULL == x509 )
63.
64.
              fprintf(stderr, "PEM_read_bio_X509_AUX fail\n");
65.
              return NULL;
66.
 67.
          return x509;
68.
69.
          if (cert != NULL)
 70.
71.
              BIO_free(cert);
72.
73.
          return NULL;
74. }
75.
76.
      int x509_verify()
77. {
78.
          int ret;
79.
          char cert[MAX_LEGTH];
 80.
          X509 *user = NULL;
81
82.
          X509 *rootca = NULL;
83.
          X509 *class2ca = NULL;
84.
 85.
          X509_STORE *ca_store = NULL;
 86.
          X509_STORE_CTX *ctx = NULL;
          STACK_OF(X509) *ca_stack = NULL;
87
88.
 89.
          /* x509初始化 */
          ca store = X509_STORE_new();
90.
          ctx = X509_STORE_CTX_new();
91.
 92.
93.
          /* root ca*/
 94.
          GET_ROOT_CA_CERT(cert);
 95.
          rootca = pem_to_x509(cert);
96.
          /* 加入证书存储区 */
97.
          ret = X509_STORE_add_cert(ca_store, rootca);
98.
99.
100
              fprintf(stderr, "X509_STORE_add_cert fail, ret = %d\n", ret);
101.
              goto EXIT;
102.
103.
104.
          GET_CLASS2_CA_CERT(cert);
105.
          class2ca = pem_to_x509(cert);
106.
107.
          /* 加入证书存储区 */
108.
          ret = X509_STORE_add_cert(ca_store, class2ca);
109.
          if ( ret != 1 )
110.
111.
              fprintf(stderr, "X509_STORE_add_cert fail, ret = %d\n", ret);
112.
              goto EXIT;
113.
114.
115.
          /* 需要校验的证书 */
          GET_CLIENT_CERT(cert, CERT_PATH, CLIENT_CERT);
116.
117.
          user = pem_to_x509(cert);
118.
119.
          ret = X509_STORE_CTX_init(ctx, ca_store, user, ca_stack);
120.
121.
122
              fprintf(stderr, "X509_STORE_CTX_init fail, ret = %d\n", ret);
123.
```

```
124.
125.
           //openssl-1.0.1c/crypto/x509/x509_vfy.h
ret = X509_verify_cert(ctx);
126.
127.
128.
           if ( ret != 1 )
129.
130.
                fprintf(stderr, "X509_verify_cert fail, ret = %d, error id = %d, %s\n",
131.
                       ret, ctx->error, X509_verify_cert_error_string(ctx->error));
132.
133.
134. EXIT:
           X509_free(rootca);
X509_free(class2ca);
X509_free(user);
135.
136.
137.
138.
           X509_STORE_CTX_cleanup(ctx);
X509_STORE_CTX_free(ctx);
139.
140.
141.
142.
           X509_STORE_free(ca_store);
143.
144.
           return ret == 1 ? 0 : -1;
145. }
146.
147. int main()
148. {
           OpenSSL_add_all_algorithms();
x509_verify();
149.
150.
151.
           return 0;
152. }
阅读(1265) | 评论(0) | 转发(0) |
上一篇:Linux的nm查看动态和静态库中的符号
下一篇:没有了
```

给主人留下些什么吧!~~

关于我们 | 关于IT168 | 联系方式 | 广告合作 | 法律声明 | 免费注册

Copyright 2001-2010 ChinaUnix.net All Rights Reserved 北京皓辰网域网络信息技术有限公司. 版权所有

感谢所有关心和支持过ChinaUnix的朋友们

16024965号-6