第七次实验报告

防火墙和SSL实验

郭裕彬 2114052 物联网工程

实验要求

1. 防火墙实验

   防火墙实验在虚拟仿真环境下完成，要求如下：

   b. 了解包过滤防火墙的基本配置方法、配置命令和配置过程。

   c. 利用标准ACL，将防火墙配置为只允许某个网络中的主机访问另一个网络。

   d. 利用扩展ACL，将防火墙配置为拒绝某个网络中的某台主机访问网络中的Web服务器。

   e. 将防火墙配置为允许内网用户自由地向外网发起TCP连接，同时可以接收外网发回的TCP应答数据包。但是，不允许外网的用户主动向内网发起TCP连接。

2. SSL实验（选做）

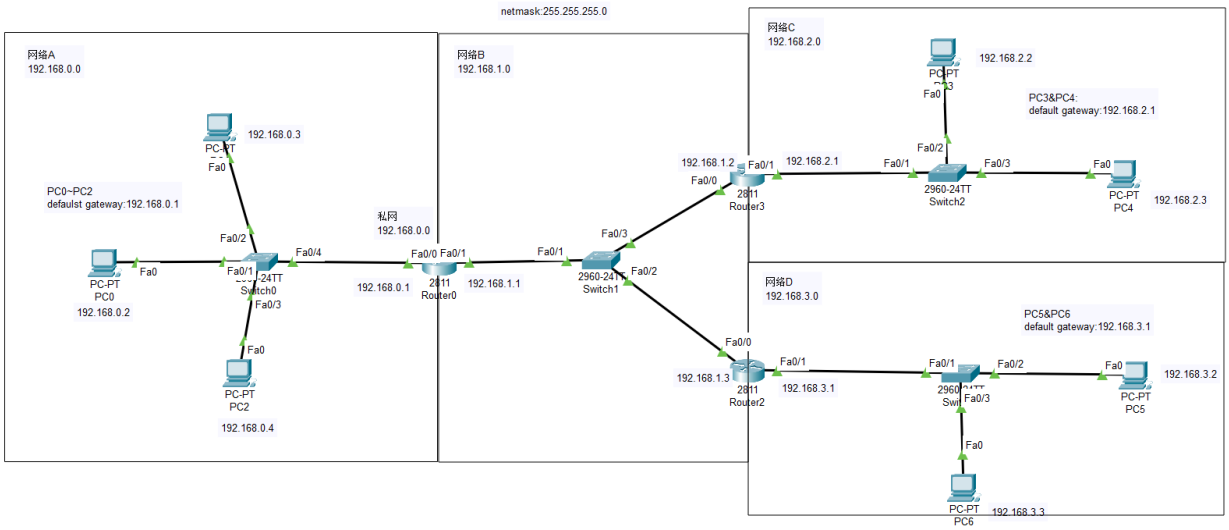
   SSL实验在实体环境下完成，要求如下：

   a. 完成Web服务器的证书生成、证书审批、证书安装、证书允许等整个过程。

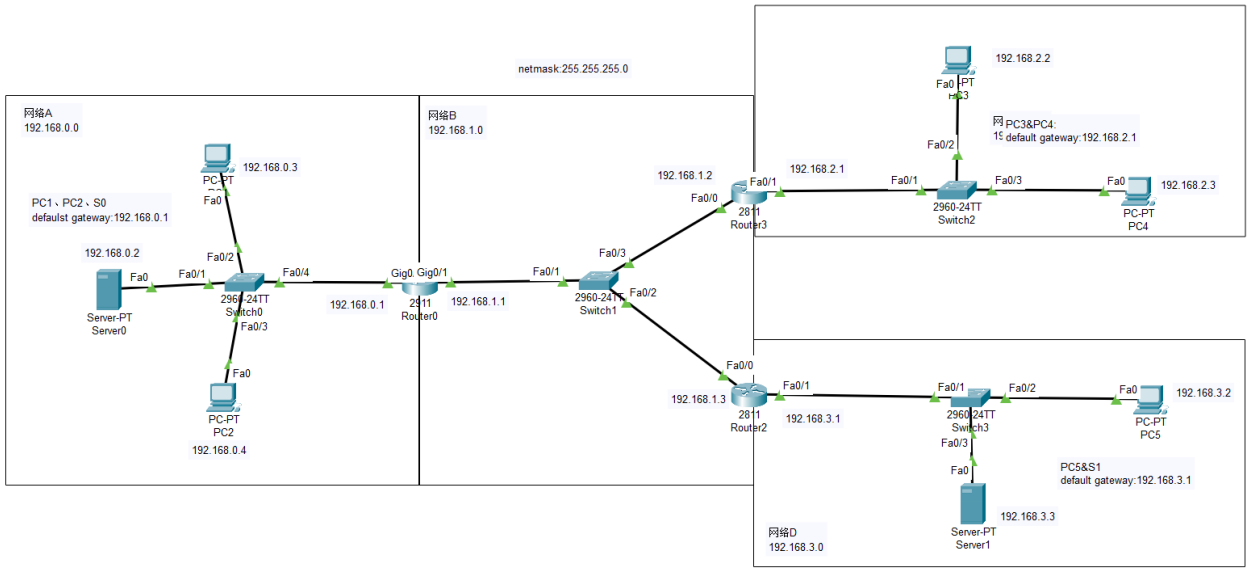
   b. 实现浏览器与Web服务器的安全通信。

实验设计

要求1.b网络拓扑



要求1.c、1.d网络拓扑

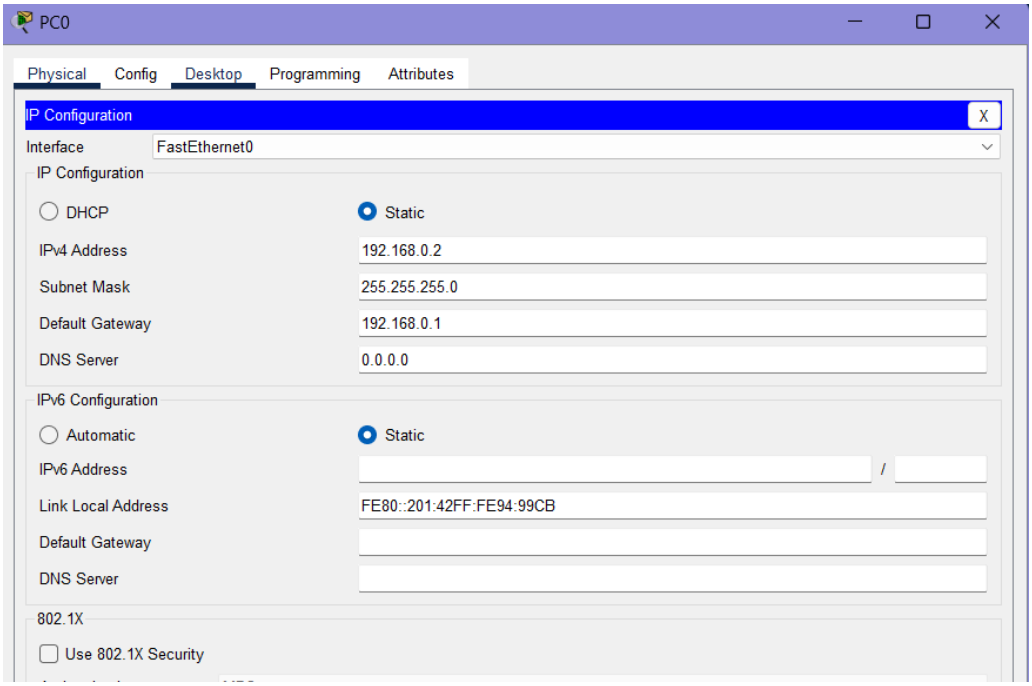


实验过程

防火墙实验

- 通过课程PPT和相关实验视频，学习ACL语句的编写和配置的过程
- 按照实验设计的网络拓扑图，配置各个设备的网络环境

- 各个主机和服务器的IP地址配置方法如之前的实验，在Desktop选项中的IP Configuration应用中进行配置，不再过多叙述。



- 路由器的IP地址配置

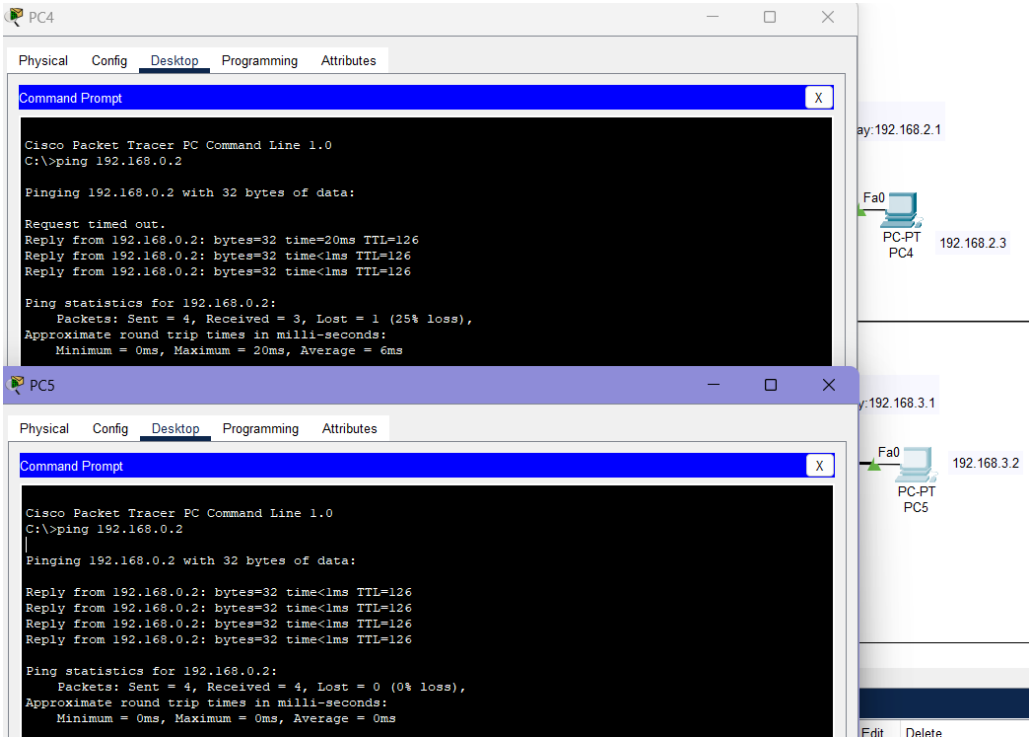
  按照实验设计分配，需要在一个设备上配置多个接口，因此在路由器的CLI中使用如下命令进行配置：

```
enable                                  //特权模式
config terminal                         //全局配置模式
interface fa0/0                  //接口fa0/0
ip address 192.168.0.1 255.255.255.0    //配置该接口IP
地址为192.168.0.1，子网掩码为255.255.255.0
no shutdown                             //开启接口
exit                                    //返回上一级
interface fa0/1                  //接口fa0/1
ip address 192.168.1.1 255.255.255.0    //配置该接口IP
地址为192.168.1.1，子网掩码为255.255.255.0
no shutdown                             //开启接口
exit                                    //返回上一级

router rip                              //开启rip
version 2                               //版本为2
network 192.168.0.0                     //分配网络
network 192.168.1.0                     //分配网络
```

  使用一样的方法按照设计配置R1和R2。

经过测试，在没有配置防火墙时，右侧的网络C和网络D中的主机均可以与网络A中主机正常通信



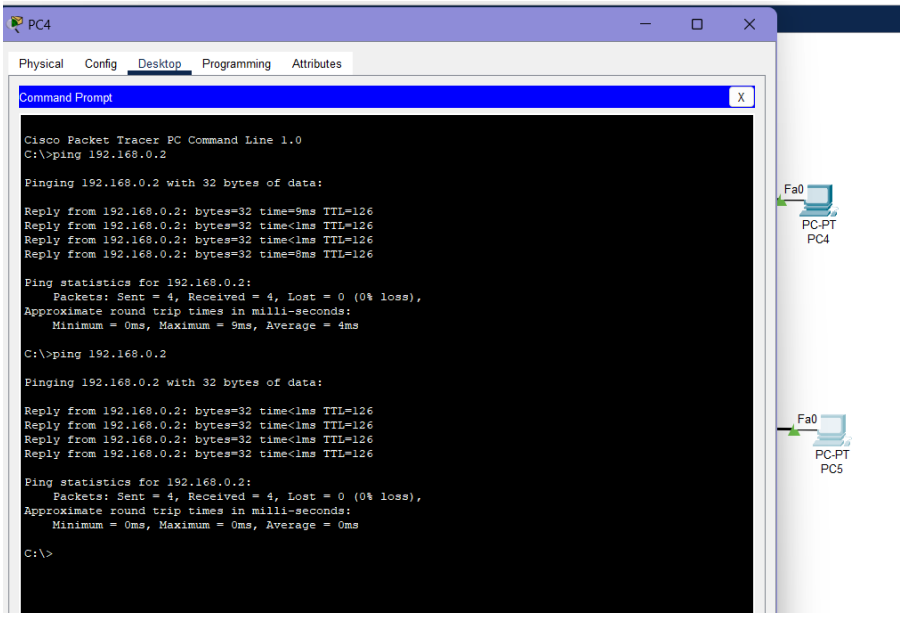
- 使用标准ACL配置防火墙，只允许网络C中的主机访问网络A，而不允许网络D中主机访问。

  在路由器R0的CLI中使用如下命令：

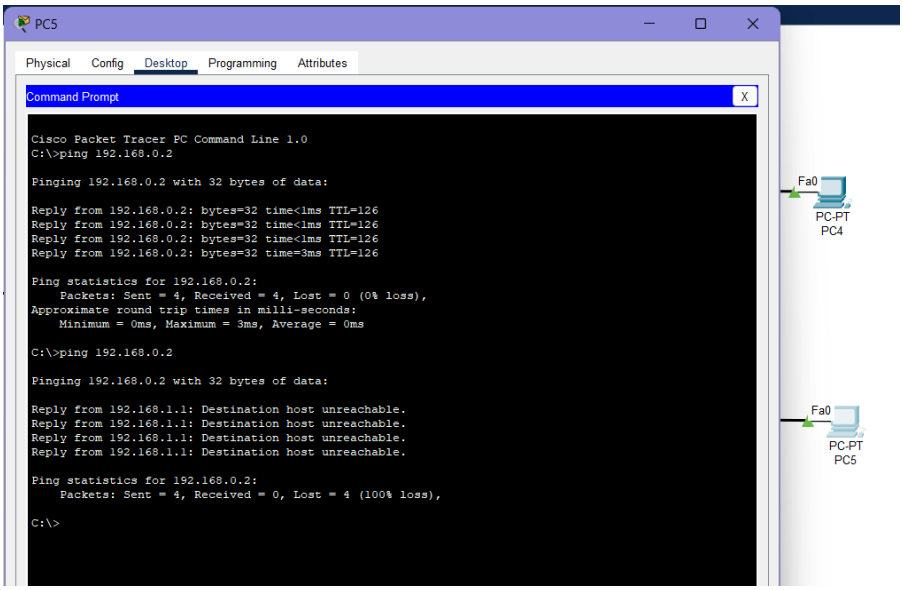
```
enable                                  //特权模式
config terminal                         //全局配置
access-list 1 permit 192.168.2.0 0.0.0.255  //创建ACL1，允许通配符为0.0.0.255的192.168.2.0网络的流量通过
access-list 1 deny any                  //阻止其余的网络（CISCO默认，可省略）

interface fa0/1                         //进入到R0与外部网络连接的接口
ip access-group 1 in                    //绑定ACL1到fa0/1的入站
```

  - 再对连通性进行测试
    - 网络C中主机PC4与网络A通信，可以看到仍能够正常通信

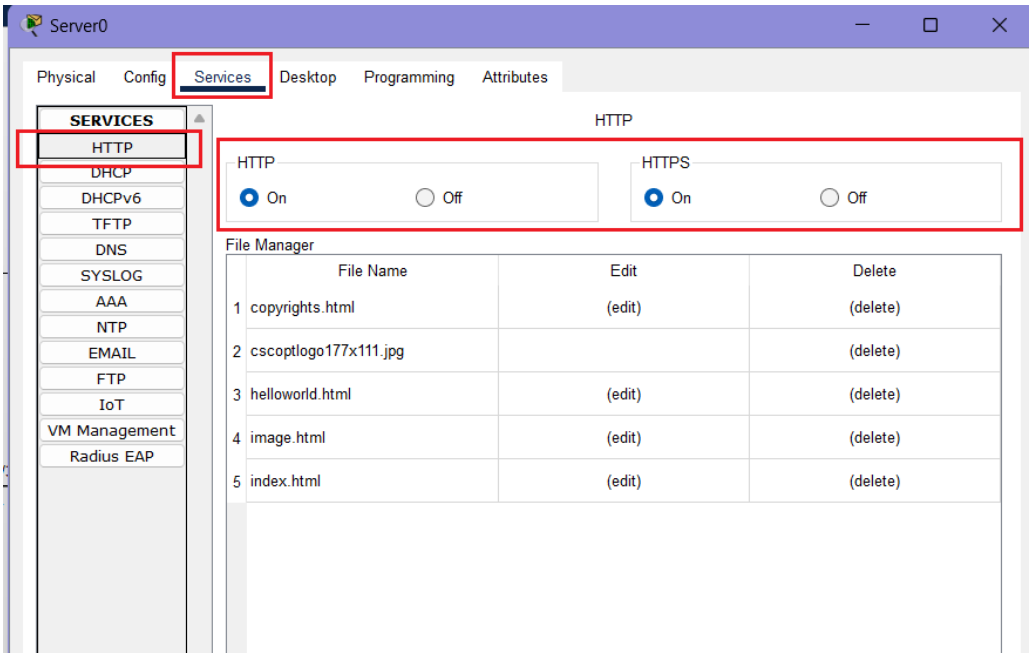


- 网络D中主机PC5与网络A通信，可以看出通信不能正常进行



以上结果说明，ACL配置成功，来自网络D的流量被路由器丢弃

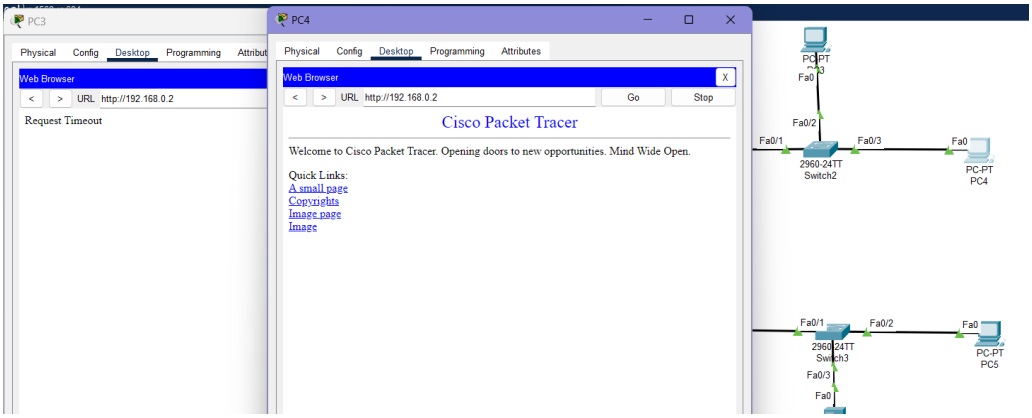
- 用扩展ACL，将防火墙配置为拒绝网络C中的主机PC3访问网络A中的Web服务器，而允许其他主机访问。
    - 开启服务器的HTTP服务，确保其可用



- 在路由器R0的CLI中使用如下命令：

```
access-list 102 deny tcp host 192.168.2.2 host
192.168.0.2 eq www        //拒绝192.168.2.2主机使用TCP流
量访问192.168.0.2的80端口
access-list 102 permit ip any any    //允许其他任何IP的
流量
interface gig0/1              //应用到gig0/1
ip access-group 102 in        //将ACL102绑定到gig0/1的入
站上
```

- 访问结果



从上图可以看出，PC3的访问超时，而同一网络下的PC4访问内网中的WEB服务器能够正常实现

- 将防火墙配置为允许内网A用户自由地向外网B、C、D发起TCP连接，同时可以接收外网发回的TCP应答数据包。但是，不允许外网的用户主动向内网发起TCP连接

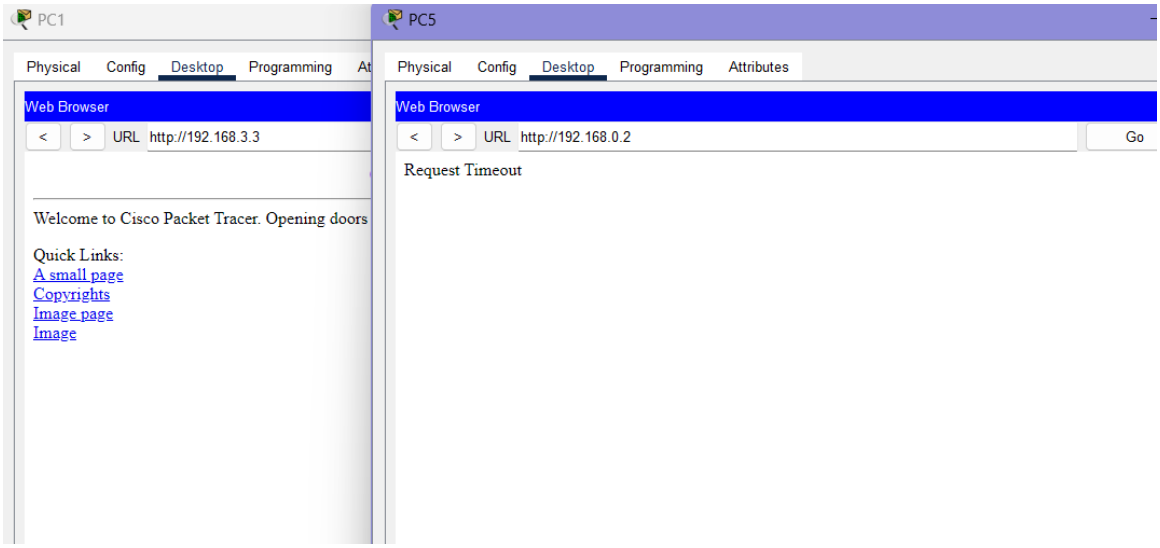
在一般情况下，我们使用ACL自反来实现这种防火墙配置，例如

```
ip access-list extended r0out
permit tcp any any reflect ctrltcp
ip access-list extended r0in
evaluate ctrltcp

interface gig0/1
ip access-group r0out out
ip access-group r0in in
```

这样就能将由出站ACL r0out生成的自反列表ctrltcp应用到入站的ACL r0in中，从而实现题给要求，但Cisco packet tracer不提供reflect这样的自反指令，因此我们只能通过简单地过滤非ACK包来阻止外网发起的TCP第二次握手，而允许由内网发出第一次握手请求的第二次握手响应包以及之后的正常数据传递包的通过。

```
access-list 101 permit tcp any any established
interface gig0/1
ip access-group 101 in
```



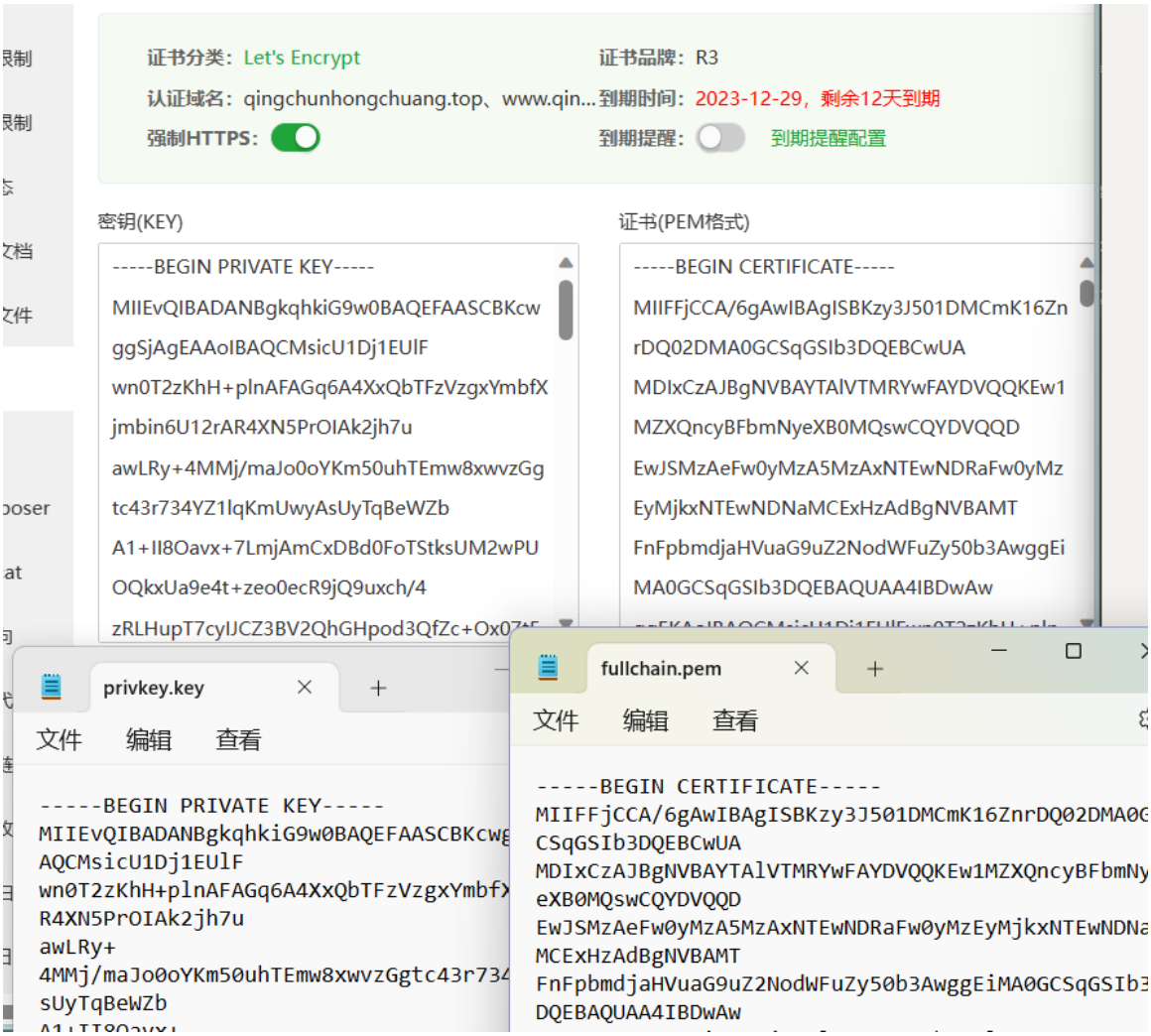
可以看到，网络A中的主机PC1能够正常访问网络C中的服务器S1，并得到服务器返回的应答数据包，而同一个网络C（外网）中的主机PC5却不能主动访问网络A中的服务器S0。

SSL实验

参加学校创新创业项目时搭建了对应的网站来进行实践，现项目完成，域名备案已经注销，无法正常通过网址访问，因此无法重新实现一次SSL证书生成、安装、允许的过程，仅能稍作演示

- 网站通过宝塔面板进行管理，可以看出之前申请的免费SSL证书还有十余天到期

- 使用宝塔面板提供的Let's Encrypt免费证书生成密钥和PEM格式证书，下载下载后，在当前证书选项卡中贴入对应的内容，保存后即实现了网站证书的配置



- 如果通过Nginx配置文件的修改方式来实现证书的安装，访问如下路径可以打开对应网站的nginx配置文件

之后通过添加如下语句读取保存在服务器中下图路径的key和pem文件，达到证书的安装效果


```
ssl on;
#fullchain证书路径
ssl_certificate /.../fullchain.pem;
#privkey证书路径
ssl_certificate_key /.../privkey.pem;
ssl_session_timeout 5m;
ssl_protocols SSLv3 TLSv1 TLSv1.1 TLSv1.2;
ssl_ciphers "HIGH:!aNULL:!MD5 or HIGH:!aNULL:!MD5:!3DES";
ssl_prefer_server_ciphers on;
```

- 在同一配置文件中开启403端口，允许HTTP到HTTPS的转换，之后便可以通过HTTPS安全访问网站。

```
listen 443 ssl http2;

#HTTP_TO_HTTPS_START
if ($server_port !~ 443){
    rewrite ^(/.*)$ https://$host$1 permanent;
}
```

- 之后我们便可以通过HTTPS访问，域名已经取消备案，故我们使用对应的服务器IP地址访问，可以看出虽然由于使用的证书来源是网站名而非IP地址报警不安全，但这个访问已经是https了。