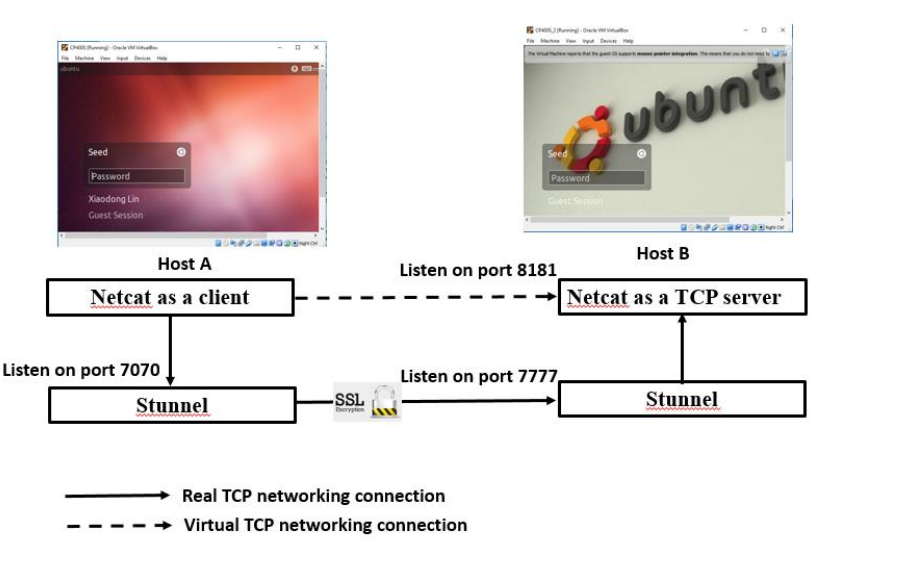
Secure Communication with Stunnel

58119125 JiangZhuoyang

**Introduction:**

Stunnel is an open source program that allows you to encrypt arbitrary TCP connections inside SSL (Secure Sockets Layer) available on both Unix and Windows. Stunnel can allow you to secure non-SSL aware daemons and protocols (like POP, IMAP, LDAP, etc) by having Stunnel provide the encryption, requiring no changes to the daemon's code.



In this lab, we are required to establish a secure channel by using Stunnel, which are shown in the above Figure. In the end, you can use Netcat to listen on a TCP port 8181 on one Ubuntu 16.04 virtual machine (referred to as Host B). Then, you can use Netcat to connect to the Listening TCP port 8181 from another Ubuntu 16.04 virtual machine (referred to as Host A), but over a secure SSL/TLS channel through Stunnel, as shown in the Figure above.

**The lab environment:**

* Ubuntu 16.04 virtual machine HostA as client
* Ubuntu 16.04 virtual machine HostB as sever
* Stunnel and OpenSSL

**Setup**

|  |  |
| --- | --- |
| **Step** | **Procedure** |
| 1. | 1. Install the Stunnel -- Universal SSL Wrapper onto two Ubuntu 16.04 virtual machines. 2. To install Stunnel on Ubuntu, type:   ***sudo apt-get install stunnel***   1. The result after downloading.   01-下载Stunnel |
| 2. | 1. Create two VM as the experiment machine 2. Set VM’s identity: 3. Host A(Client)：10.0.2.6 4. Host B(Server): 10.0.2.7 |

**Public key encryption with OpenSSL**

|  |  |
| --- | --- |
| **Step** | **Procedure** |
| 1. | 1. Generate private key and certificate for stunnel server-side proxy using OpenSSL   ***openssl req -newkey rsa:2048 -nodes -keyout stunnel.pem -x509 -days 365 -out stunnel.pem***   1. Input the certificate information as below:   02-在服务机上创建证书 |
| 2. | 1. View private key information   03-查看私钥信息   1. View certificate information   03-查看证书信息 |

Stunnel configuration on Server VM

|  |  |
| --- | --- |
| **Step** | **Procedure** |
| 1. | 1. create a "stunnel.conf" file in the “/etc/stunnel” directory:   ***sudo vi /etc/stunnel/stunnel.conf***   1. Modify the server VM’s "stunnel.conf" file:   04-编写StunnelServerConfigurationFile   1. Configure port information: |
| 2. | 1. Enabling Stunnel by modifying the /etc/default/stunnel4 file，firstly open the file:   ***sudo vi stunnel4***   1. Modify the server VM’s "stunnel4" file:   05-修改Stunnel4文件信息   1. Set ENABLED to 1: |

Stunnel configuration on Client VM

|  |  |
| --- | --- |
| **Step** | **Procedure** |
| 1. | 1. create a "stunnel.conf" file in the “/etc/stunnel” directory:   ***sudo vi /etc/stunnel/stunnel.conf***   1. Modify the client VM’s "stunnel.conf" file:   07-编写StunnelClientConfigurationFile   1. Configure port information: |
| 2. | 1. Enabling Stunnel by modifying the /etc/default/stunnel4 file，firstly open the file:   ***sudo vi stunnel4***   1. Modify the client VM’s "stunnel4" file:   08-修改Stunnel4文件信息   1. Set ENABLED to 1: |

Test TCP information transmission under stunnel agent

|  |  |
| --- | --- |
| **Step** | **Procedure** |
| 1. | 1. Test run stunnel on the server VM   06-尝试运行stunnel   1. Test run stunnel on the client VM   09-尝试运行stunnel |
| 2. | 1. Establish TCP connection with stunnel on the client and connect to the server:      1. Using stunnel to establish the TCP connection with the client on the server:      1. The connection is successful. Start the communication test. The test results are as follows:   10-stunnel通信结果 |

**Analyze our results:**

We were puzzled when setting the port number of the stunnel configuration file. However, after deeply understanding the principle of the stunnel agent, we understood the true meaning of accept and connect, and successfully set the port. Finally, we ran stunnel on two virtual machines at the same time, established a TCP connection between the client and the server, and successfully completed secure communication.