目录

介绍

本教程是一本关于 tinc VPN 的实操指南，记录了作者几年来使用 tinc 的经验、教训和建议，可作为 tinc 入门参考和快速指南。

您可以在这里发现本教程的最新版本：https://chanix.github.io/TincCookbook

作者序

谨以此文献给我的家人，感谢一直以来对我的容忍和支持。我不擅长表达，在这里说声谢谢，或许未来他们能够知道。

在工作和生活中，我有将不同地点和网络环境的设备连接在一起，形成虚拟专有网络（VPN）的需要：

```
希望能进行异地之间数据的交换（远程办公，远程访问家庭NAS，远程备份……）；
希望安全的连接，不用担心隐私和数据的泄露；
希望能简化安全和网络配置，不用考虑各台设备所处环境的区别；
希望尽可能的降低搭建和运维成本，最好是折腾完了就扔一边不用管；
希望尽可能的简单快捷，在下不是IT科班出身，太深奥的东西理解不了；
```

经过比较和测试，最终选择了 tinc 作为虚拟专有网络的实现工具。目前搭建的几个 tinc 网络已经稳定运行了超过3年+。在实际操作中，遇到了不少问题踩了不少坑，相对小众所以资料也不多。因此我将相关的经验和文档整理了一下，形成本教程，作为个人使用 tinc 的总结和备忘。同时也公开分享出来，如果能为您答疑解惑，将不胜荣幸。

本教程针对非专业人士，重点在于实操。作为读者，您不需要是专业IT人员，但需对网络有基本的了解并有一定的动手能力。本教程重点在于如何快捷的使用 tinc，而不是对其原理和源码深入的剖析和讲解。作者试图让您在读完本教程后，拥有快速搭建和管理 tinc 网络的能力。换句话说，如果完成一件事有很多种方法，本教程只会告诉你一个经过作者验证和总结的方法（可能最笨但确保实用），不会教你"茴香"的"茴"有九种写法。

本教程的目的在于让一个具有网络使用和电脑使用经验的普通用户在半小时内搭建完成自己的 tinc 网络。如果您遇到了困难和麻烦，那一定是本教程写的还不够清晰明了，请及时反馈，这是我逐步修改和完善的动力和方向。

相关内容

- 本书源码仓库：https://github.com/Chanix/TincCookbook
- tinc 官网：https://www.tinc-vpn.org
- tinc 源码仓库：https://github.com/gsliepen/tinc

转载注意事项

tinc 简介

本章节包含一些技术介绍。对于普通使用者而言不需要太过关心技术细节。可以跳过本章，直接阅读后继章节。

tinc 是一个组建虚拟专用网络（VPN）的工具，以 GNU 协议发布，通过隧道及加密技术在互联网上点与点之间创建专有网络。tinc 在网络层工作，因此无需对现有软件进行修改和配置。

使用 tinc，您可以配置专属的的低延迟、高带宽、可扩展的 P2P VPN。其中的数据通讯经过加密和压缩，避免敏感数据和隐私的泄露。无论何时何地，只要能连入互联网，就可以安全的访问 **VPN**。

具体技术细节和功能不多说了，参见官网。对于我来说，选择 tinc 有以下几个理由：

- 开源，截止目前还在不断更新完善；
- 分布式网状路由，避免单点高负载和故障；
- 可运行多个实例来接入多个VPN；
- 通过虚拟网卡通讯，无需对现有应用软件进行修改和配置；
- 通讯支持 加密/认证/压缩，并支持参数选择；
- 支持常见的操作系统和网络拓扑，适用场景广泛；

基础使用

对于普通使用者，快速的搭建起一个能用的安全强度适合的 VPN 是最大的需求。因此，本章节主要讲解基本概念和实践操作。配置内核、编译源代码、甚至交叉编译、高级定制等内容请参考本教程的进阶部分。

预备知识

- 每个 tinc VPN 必须有个名称，一个 VPN 可以包括很多主机；
- 每台主机必须有个名称，同时需要运行 tinc，一台主机可以通过运行多个 tinc 实例来加入多个 tinc VPN；
- tinc 启动时接受参数来指定要启动的网络，并定位到对应的网络配置目录读取配置；
- 启动后，读取网络配置目录中的主配置文件 tinc.conf，执行启动脚本（tinc-up），然后 ConnectTo 指定的主机，同时接受其他主机对本机的 ConnetTo；
- tinc 通过读取主机描述文件来获得主机信息，当前主机和 ConnectTo 的目标主机上都必须有双方的主机描述文件；
- ConnectTo 成功（认证通过），则加入 tinc VPN；
- 当 tinc 结束的时候，执行关闭脚本（tinc-down）；

tinc VPN 名称接受 a-z 0-9 _ 中的字符，主机名称也是一样。

组网步骤

搭建 tinc VPN 非常快捷，每台主机的设置仅需几步，所有主机重复这套步骤即可：

graph LR A(安装) --> B(配置和交换密钥); B(配置和交换密钥) --> C(运行); C(运行) --> D(检查);

其中的安装、运行和检查非常简单，各平台的操作大同小异，有个大致的概念就可以了，后继章节有详细的针对各个平台的说明，请参阅实例部分。

主要需要理解和掌握的是“配置和交换密钥”这部分。

如何配置和交换密钥

- 配置，就是在 **tinc** 主配置目录里，按照配置规范，建立好相应的子目录和相应的文件。
- 交换密钥，就是在网络配置目录中，包含通讯双方的主机描述文件。

tinc 的配置方式是通过一套目录和其中的文件（以下简称为配置主目录）来完成的，配置主目录中按规范存放一系列子目录和文件。配置主目录默认值为 /etc/tinc （Unix-like OS）或 C:\Program Files\tinc （Windows OS）。

配置主目录下，每个子目录是以该目录名为名称的 VPN 网络的配置目录（以下简称为 网络配置目录）。每个网络配置目录，指定了该网络的相关配置。tinc 实例启动时接受参数来指定要启动的网络，并定位到对应的网络配置目录读取配置。

每个网络配置目录中，有以下内容：

- tinc.conf

主配置文件，其中的内容指定了该网络下 tinc 的配置 。其中的 Name 说明本主机名称，ConnectTo 指定启动后要自动连接的主机（可以多个）。

- tinc-up、tinc-down

脚本文件，这两个脚本是在 tinc 启动和关闭该网络时被调用。一般用来设置虚拟网卡的IP、路由。　　如果是 Unix-like 系统，需要有运行权限，如果是 Windows 系统，则需要增加 “.bat”后缀，即 “tinc-up.bat” 和 “tinc-down.bat”。

- rsa_key.priv

RSA 私钥文件，存放本主机的 RSA 私钥；

- hosts 子目录

主机描述文件存放目录。其中的每一个文件描述了一台主机的信息，文件名与主机名保持一致。

这是我笔记本（notebook）上的配置主目录：

```
/etc/tinc               (配置主目录)
|
├── vpn_home            （第一个 VPN 的网络配置目录，目录名和网络名保持一致，为 vpn_home）
|   ├── hosts
```

```
│   │   ├── tinc_ecs    （主机 tinc_ecs 的描述文件）
│   │   └── notebook    （主机 notebook 的描述文件）
│   ├── rsa_key.priv    （本主机的 RSA 私钥）
│   ├── tinc.conf       （tinc 主配置文件）
│   ├── tinc-down       （当关闭 vpn_home 时，执行该脚本）
│   └── tinc-up            （当启动 vpn_home 时，执行该脚本）
│
└── vpn_work            （第二个 VPN 的网络配置目录，目录名和网络名保持一致，为 vpn_work）
    ├── hosts
    │   ├── server      （主机 server 的描述文件）
    │   └── notebook    （主机 notebook 的描述文件）
    ├── rsa_key.priv    （本主机的 RSA 私钥）
    ├── tinc.conf       （tinc 主配置文件）
    ├── tinc-down       （当关闭 vpn_work 时，执行该脚本）
    └── tinc-up         （当启动 vpn_work 时，执行该脚本）
```

在这个示例中，主机上共配置了两个VPN，*vpn_home* 和 *vpn_work*。每个网络的配置都是在配置主目录下的一个子目录，子目录名称和网络名称一致。

以 *vpn_home* 为例：

tinc.conf

```
# 说明本主机名称
Name = notebook

# 指定启动时自动连接的主机。
# 可以使用多个ConnectTo来自动连接多个主机。
# 也可以没有，等待其他主机发起连接。
ConnectTo = tinc_ecs
```

tinc-down

```
#!/bin/sh

# 关闭虚拟网卡
ifconfig $INTERFACE down
```

tinc-up

```
#!/bin/sh

# 启用虚拟网卡，并设置其 IP 为 10.0.0.100，子网掩码为 255.255.255.0
ifconfig $INTERFACE 10.0.0.100 netmask 255.255.255.0
```

hosts/tinc_ecs

```
# tinc_ecs 是一台公网机器，公网IP为 111.222.333.444
Address = 111.222.333.444

# tinc_ecs 的 VPN 内部 IP 为 10.0.0.1
# /32 说明其为一台机器而不是子网（普通用户直接用 /32 就可以了）
Subnet = 10.0.0.1/32

-----BEGIN RSA PUBLIC KEY-----
......
......
......
-----END RSA PUBLIC KEY-----
```

hosts/notebook

```
# notebook 是移动办公的笔记本，没有公网IP，所以没有 Address 这一行。
# notebook 的 VPN 内部 IP 为 10.0.0.100，这里要和 tinc-up 里面设定的一致。
# /32 说明其为一台机器而不是子网（普通用户直接用 /32 就可以了）
Subnet = 10.0.0.100/32

-----BEGIN RSA PUBLIC KEY-----
......
......
......
-----END RSA PUBLIC KEY-----
```

总结一下，其实就是本节开头的话：

- 所谓配置，就是按照上面的例子建立相应的目录，并在目录中建立相应的文件；
- 所谓交换密钥，就是确保通讯的双方主机有对方的公钥，而公钥存放在主机描述文件中。

文笔有限，可能看到这里您还是有点模糊。不要紧，有了基本的概念和知识，跟着我一起规划和搭建一次 tinc VPN。先不求甚解，依葫芦画瓢，实用优先，然后再求其索。

组网实例

上面说了很多理论的东西，任何说教都比不上实践。下面结合实例，跟我一起组建属于自己的tinc网络。 这是作者日常使用的tinc网络。详细说明了作者是如何开始设计和搭建一个tinc网络。包括了目前可以看到的大多数网络环境和应用场景，目前已经稳定运行了超过3年。读者可以按照自己的需要，从中截取一部分。

网络规划

加入VPN的每台主机上，都需要运行tinc，所以您必须有对目标主机的操作权限；

tinc 通过connectTo到主机，出于加密认证的需要，需要有对方主机的描述文件；

tinc 是分布式的，但是维护主机描述文件还是集中起来比较好，所以我设置了一个核心主机，这台主机上保存所有主机的描述文件。其他主机仅需要有本身和这台核心主机的描述文件就可以了。这样可以集中式的管理，降低维护量。增加或者删除主机只需要在这台核心主机上进行就可以了。主机可以在运行期间，自动发现和交换其他主机的描述信息。

我使用的平台比较杂，不用看全部，挑自己用的就好了。

尽量遵循不侵入的方式，尽量少的管理和尽量少自动化的操作。这样降低难度，比较方便一点。

开机就自动连上，每次去点很麻烦，尽量少操作；

普通的使用，用 router 模式即可。

我的网络规划和配置如下，箭头表示 ConnectTo 方向。基本上常见的网络环境都有了，读者可以按照顺序来读，或者直接跳到对应的章节截取需要的网络部分即可。我想，我这个环境覆盖了90%的使用场景，依葫芦画瓢就可以了。

稳定优先，可以接受一定的成本，但要尽可能的降低。

tinc支持多种模式，上述有。使用用交换机模式，可以将多个以太网段连成一个以太网，可以DHCP什么的。但需要转发很多广播包，效率上不如路由器模式。对于我的场景来说，没有这个必要，而且主机少，手工指定所有机器的IP配置并不麻烦，而且这样比较可控，通讯报文和效率都比较高。所以，选择默认的 router 路由模式。

Tinc 有两个大版本。1.0为稳定版，1.1开发版。核心功能1.0版都有了，具体的差别参见官网。本着生产系统稳定优先原则，选择1.0版。

0、最终的网络示意图，基本上常见的网络环境都有了，读者可以按照顺序来读，或者直接跳到对应的章节截取需要的网络部分即可。我想，我这个环境覆盖了90%的使用场景，读者照葫芦画瓢就可以了。

graph RL; L(台式机) --> Z(核心主机); W(笔记本) --> Z; M(开发机) --> Z; D(NAS) --> Z; R(路由器 NAT) --> Z; I(家人设备) --- R; J(临时访问设备) --- R;
1、建立核心tinc主机

1、安装一台核心tinc主机。这台主机需要具有公网ip，打开tinc使用的端口，外部能直接访问。对于tinc网络中的其他设备而言，只要能上网，能联通这台核心主机的对应端口，就可以联通tinc网络。按个人习惯，IP 定为 10.0.0.254，C类网络（可容纳255台机器），网络掩码为 255.255.255.0，CIDR 10.0.0.254/24。

可以购买云主机VPS 或者具有公网IP的家里的路由器也可以。出于简单稳定的需要，我选择云主机，阿里云。

我的选择，买一台云主机用以运行tinc。不需要太高的配置，个人普通使用1核1GB内存20G普通云盘就可以了。带宽方面按照自己的流量来选择，一般有按使用付费和固定带宽两种模式。我喜欢简单的付费模式，带宽要求也不大，因此选择的是1M固定带宽。这样的配置，目前阿里云的价格是人民币70左右，一次性时间长一点付费还有优惠。我是一次性付了N年，差不多涵盖了有生之年……扔那里慢慢用就好了。

阿里云提供99%以上的可靠性，负载不大，基本不可能出问题，因此不考虑核心主机的备份了。网络出问题忍一下不是什么大事，主机出了问题，只要配置文件备份好了，重新安装也就是半小时的事情。又不是什么大不了的环境，杀鸡不用牛刀。

安装：

sudo apt-get install tinc -y

ubuntu 的配置文件夹为 /etc/tinc

生成配置文件

生成秘钥

配置为默认运行

/etc/tinc/nets.boot

然后重新启动

sudo reboot

2、配置常用操作机器（win10）

先安装驱动

然后复制就好了。

同样的 配置

本机启动tinc

然后记得hosts里面加上核心机器的描述文件

然后将本机的描述文件复制到 核心机器上

等一下，或者重新启动 service tinc restart

然后 ping ……

搞定了。

3、配置常用开发机（mac）

4、配置常用测试机（LINUX）

5、配置虚拟机（virtualbox等）

6、配置路由器（openwrt）

7、配置阿里云专有网络，仅有内网IP的机器

8、配置安卓手机

9、配置苹果手机

10、常用便携设备（便携路由）

graph LR; A(光纤路由器) --> B(NETGEAR); B --> C(IPTV 机顶盒); B --> D(其他设备);

TODO

本章节尚未更新和完成，作者还没有更新到这里，只是占了个位置。

不要心急，不要心急，作者会慢慢逐步更新的。

advanced

还没写呢！

进阶

进一步的资料

HOST MAIN SCRIPT

tinc网络实际使用来做什么

远程控制，ssh 远程桌面 VNC

代理转发

个人有时候做些项目，需要。例如开发微信，给别人看产品样品，

通过代理访问受限资源

个人/小型公司的代码管理等等 连接多个以太网段形成1个网络

安全、管理和运维

秘钥长度的选择

压缩的选择

主机名用域名还是写死IP地址？

通过工具，例如 ansible 简化客户端管理

通过NAT，简化对客户端的干预

结束语

写完了，没啥说的。有啥意见和建议放issue里吧。

祝您好运。