Table of Contents

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。

作者序

谨以此教程献给我的家人，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文科生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。

注：

- tinc 目前两大版本：1.0为稳定版，1.1开发版。本着稳定优先原则，本教程讲解和使用 1.0 版。
- tinc 是一种 VPN 实现技术，VPN 是虚拟专有网络的简称；
- tinc 相对小众，如果选择它，请记得最终需要负责的，是您 :) ；

相关内容

- 本书仓库：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

tinc 是一个组建虚拟专用网络（VPN）的工具，以 GNU 协议发布并开源，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。使用 tinc 组网，无论何时何地，只要能连入互联网，就可以直接访问操作 VPN 中的设备，同时网络通讯经过加密和压缩，安全性也有提高。

tinc VPN 是对等的网络，没有 server 和 client 之分，网络中每个连接的设备都视为一个主机，这些主机上都运行 tinc（路由器 NAT 除外，后面会有讲到）。安装、配置和运行很简单，后期增加/删除主机也很方便（修改配置重启即可），但需要您对接入的主机有操作权限。

tinc 支持多种模式，包括 router、switch、hub，可以想象成为虚拟的 路由器、交换、集线器，默认为 router 即路由器模式，一般使用默认的路由器模式即可。

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，避免传统 VPN 方案里中心服务器高负载和单点故障；
- 占用的系统资源小，适用资源受限的设备；
- 对于应用来说不需要额外关注，能感受到的就是系统里多了个网络设备(虚拟网卡）；
- 通讯支持加密、压缩，支持参数选择；
- 内网穿透（NAT）简单方便；
- 跨平台：支持常见的操作系统、网络协议（ipv4+ipv6），适用广泛；
- 支持桥接以太网；

分布式网状路由

与传统VPN方案比较，tinc最大的特点是分布式、基于网状结构的网络路由。基于P2P的技术，可以实现流量直接到达目标机器，而不像传统VPN 那样必须经过中间的服务端。

跨平台

支持大部分常见平台与操作系统。包括 Windows、Mac OS X、Linux、安卓、iOS、OpenWrt等。详细支持列表可以参见官网，点这里打开 https://www.tinc-vpn.org。支持ipv4、ipv6，适用广泛。

加密/认证/压缩

tinc 网络中数据通讯加密可靠，同时，可以通过配置的方式按需进行定制。

内网穿透（NAT）

支持各种常见网络环境下的内网穿透。

多种模式

支持 router（默认）、switch、hub 三种模式。其中 router 模式应用的较为广泛，router 模式对应路由器，仅转发IP报文，因此效率相对也较高。switch 模式对应交换机一般应用于将多个以太网络桥接。hub 一般不使用了。

```
$TINC_CONFIG_DIR
     └── NETNAME_1 （tinc 网络的名称，在启动 tinc 的时候作为标识）
            ├── hosts （放置本节点要连接到的各节点的基本配置信息）
            │   ├── HOST_1 （主机1的配置文件）
            │   ├── HOST_2 （主机2的配置文件）
            │   ├── ...
            │   └── HOST_n （节点n的配置文件）
            ├── rsa_key.priv （由 tincd 生成的本主机使用的非对称私钥，加密通讯使用）
            ├── tinc.conf （tinc 网络的配置文件）
            ├── tinc-down （关闭该网络时，会调用执行的脚本，可选）
            └── tinc-up （启动该网络时，会调用执行的脚本，可选）
```

```
Name = node_010_034_034_102
AddressFamily = ipv4
```

```
Cipher = aes-256-cbc
Digest = SHA512
#Interface = /dev/tap0
Device = /dev/tap0

Port = 34

Compression = 11

ConnectTo = node_010_034_000_001
```

```
#!/bin/sh

ifconfig $INTERFACE down
```

```
#!/bin/sh

#ifconfig $INTERFACE up
ifconfig $INTERFACE 10.34.34.102 netmask 255.255.0.0
```

-
  -
-
-
-
-

版本和模式：

复制代码

注：某些系统可能会有点差异，但差不多就这个意思，万变不离其宗，各位自己融会贯通一下。

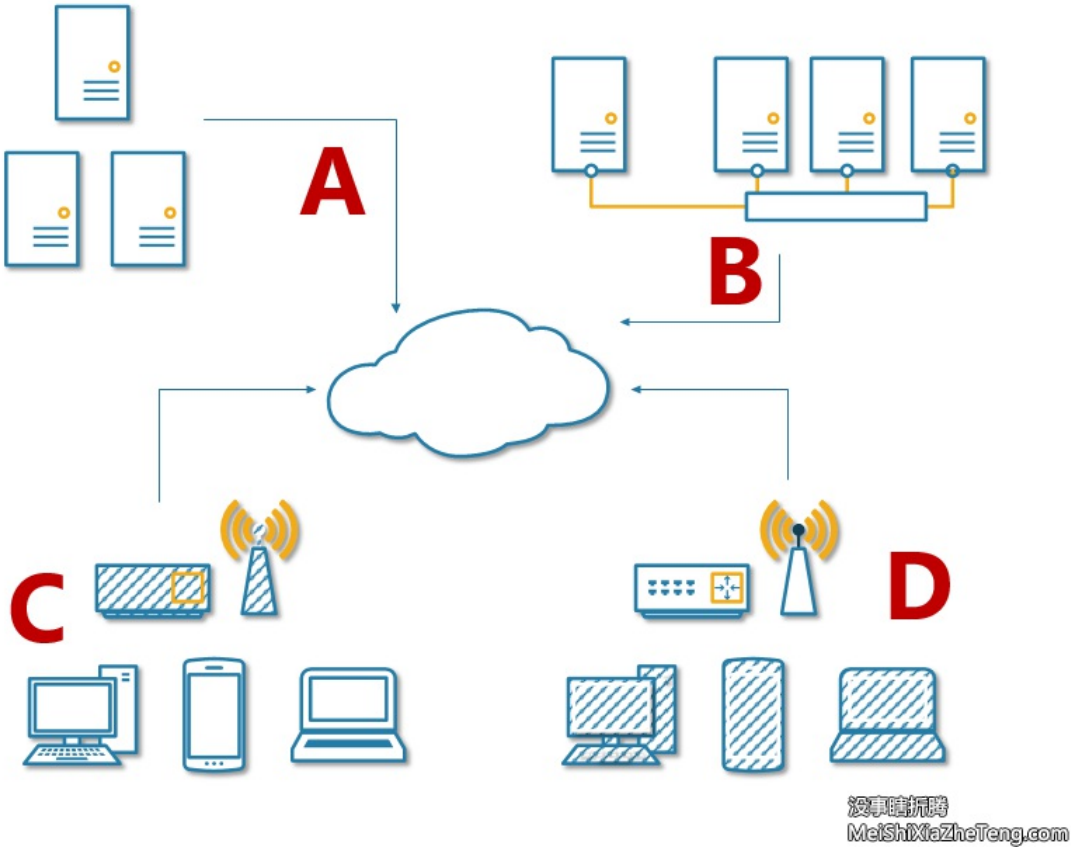
实际操作：

基本需求：

将目前常用的机器用 tinc 组成一 个VPN，虚拟专用网络命名为 vnet34。VPN 网络地址为 10.10.0.0/255.255.0.0（我贪方便用的B类地址，其实一般来说C类够用了，255个呢）。

网络拓扑：

我整理了一下我需要处理的情况，如下图：



A 互联网上的机器，绑定了公网IP，可以直接访问公网。tinc 网络里的其他设备需能直接访问这些设备，需要在每台设

备上安装 tinc。

B 互联网上的机器，没有绑定公网IP。可以访问公网、但外部无法直接访问该机器。tinc 网络里的其他设备需能直接访问这些设备，需要在每台设备上安装 tinc。

C 位于路由器后面的设备，tinc 网络里的设备需能直接访问这些设备，需要在每台设备上安装 tinc。

D 位于路由器后面的设备，tinc 网络里的设备无需直接访问这些设备，设备上不需安装，只需在路由上安装 tinc 即可。

C 和 D 有点象，在实际操作中，可以是混用的。比如，我的工作机器上跑了不少虚拟机需要被直接访问，就在这些机器上都安装了 tinc。而对于有些人来说，他们只需要能访问 tinc 网络就可以了，还特喜欢瞎折腾机器（时不时的重装一下啥的）。我要是在他们的机器设备上安装 tinc 那真是烦透了。这种情况，在路由器上装上，让他们通过路由透明代理到 tinc 网络就可以了。

基础知识

本手册并不想详细说明，作为 tinc 的使用者，您只需要知道下面的大概就可以了。

原理，怎么用，什么步骤

技术实现

虚拟网卡，操作系统中的一种TAP/TUN设备，是模拟出来的一种网卡。可以类比您机器上安装的实体网卡。tinc基于虚拟网卡进行通讯。对于上层软件应用来说，虚拟网卡和实体网卡并没有什么不同，所以上层软件不需要任何修改和适配。

所以要使用，首先您的OS要支持TAP/TUN（目前常见的系统都支持）。第二要运行tinc。第三，画个图说明tinc网络中数据是怎么传输的，是怎么建立连接的。

可以跳过本章节，直接看实例，然后反过头来看这个。下面说的比较详细，普通用户可以不用看，看后面实例就足够了。

tinc是怎么运行的？各种脚本的执行如何按次序执行？都支持哪些脚本？

How2use

如何使用

tinc 是个对等的网络，没有 server 和 client 之分，每个连接的设备都视为一个主机。各台机器上的配置文件都差不多。对于普通使用者来说，tinc的使用和其他软件并没有太大的不同。一般下面几个步骤就足够了：

安装，配置，运行

1、安装

tinc 是跨平台的应用，一般常见的平台都有现成的预编译软件可以直接使用。对于普通用户来说使用预编译的版本即可，即使不是最新版本作为个人使用也是足够的。

如果您有需要自己编译，那么需要具备配置内核、编译软件等相关知识。您可以参阅官网或者本手册的进阶部分以获得更进一步的信息。

2、配置

tinc有着自己的配置文件结构、可配置项非常多。不用担心，只要了解最基本的就可以了，大部分使用默认参数即可。

本节说明最基本的配置文件，如果有进一步的需求，请参阅官网或者本手册的进阶部分以获得更进一步的信息。

tinc 基本配置文件说明：

tinc 的配置文件一般是一个目录，由一系列文件组成。一个最简单的配置目录结构是酱紫的（括号里面的是注释和说明）：

注意：脚本需要可执行的权限。对于 Unix Like 系统，需要 chmod+x，对于windows系统，需要增加 “.bat”后缀名。

```
$TINC_CONFIG_DIR
     └── NETNAME_1 （tinc 网络的名称，在启动 tinc 的时候作为标识）
            ├── hosts （放置本节点要连接到的各节点的基本配置信息）
            │   ├── HOST_1 （主机1的配置文件）
            │   ├── HOST_2 （主机2的配置文件）
            │   ├── ...
            │   └── HOST_n （节点n的配置文件）
```

```
        ├── rsa_key.priv （由 tincd 生成的本主机使用的非对称私钥，加密通讯使用）
        ├── tinc.conf （tinc 网络的配置文件）
        ├── tinc-down （关闭该网络时，会调用执行的脚本，可选）
        └── tinc-up （启动该网络时，会调用执行的脚本，可选）
```

最简单的配置文件是酱紫的

tinc.conf

```
Name = node_010_034_034_102
AddressFamily = ipv4
Cipher = aes-256-cbc
Digest = SHA512
#Interface = /dev/tap0
Device = /dev/tap0

Port = 34

Compression = 11

ConnectTo = node_010_034_000_001
```

rsa_key.priv

tinc-down

```
#!/bin/sh

ifconfig $INTERFACE down
```

tinc-up

```
#!/bin/sh

#ifconfig $INTERFACE up
ifconfig $INTERFACE 10.34.34.102 netmask 255.255.0.0
```

/hosts/自己节点的描述文件和要connectto的主机描述文件。

- 配置文件

  tinc.conf 配置文件说明

  - host 配置项目说明
- 生成秘钥，传递公钥
- 运行软件
- 检查生效
- 完成

实例

这里放实例：

1、我的家庭网络

2、公司办公

3、服务器互联

```
* [Windows](todo.md)
* [Ubuntu 14.04](todo.md)
* [Ubuntu 16.04](todo.md)
* [Ubuntu 18.04](todo.md)
* [Mac OS X](todo.md)
* [CentOS](todo.md)
* [群辉](todo.md)
* [OpenWrt](todo.md)
```

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。

规划思路和前提

支持windows mac linux（Ubuntu/centOS）openwrt 群辉。

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。

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。

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65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。

安全一点，两台做个备份好一些。但实际使用中，本身这台机器的负载就不大，而且阿里云提供99%以上的可靠性。基本不可能出问题，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，就不杀鸡用牛刀了。

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）

advanced

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？

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。