2016 NOV 13

# DIY darknet for fun and profit (./diy-darknet-for-fun-and-profit.html)

> A darknet (or dark net) is an overlay network that can only be accessed with specific software, configurations, or authorization, often using non-standard communications protocols and ports.

Well known example of the darknet is the Onionland -- public darknet created with Tor software. This article will show you how to create your own darknet with Invisible Internet Protocol (I2P) for fun and profit. Unlike the Onionland, it will be fully distributed and suitable for any type of private and anonymous communications.

## How own darknet can be useful

**Online privacy and anonymity research**

You can run your own network to study how anonymous networks work, find their weaknesses and performance improvement possibilities.

**Filesharing software**

Anonymous network layer can be added to any filesharing software, for example, to torrent clients and apps like PopcornTime. Users will download and share content anonymously, torrent-trackers can work anonymously in these networks too.

**VPN/proxy providers**

Nextgen VPN/proxy providers can use this technology to create truly trustless service. Users will mix their traffic with each other and use regular Internet through provider's "exit nodes".

**and more...**

# I2P network overview and terminology

> I2P (Invisible Internet Protocol) is a universal anonymous network layer. All communications over I2P are anonymous and end-to-end encrypted, participants don't reveal their real IP addresses.
>
> I2P client is a software used for building and using anonymous I2P networks. Such networks are commonly used for anonymous peer-to-peer applications (filesharing, cryptocurrencies) and anonymous client-server applications (websites, instant messengers, chat-servers).

I2P uses special virtual addresses called **destinations** instead of IP address. Noone can track what real IP address owns a given destination.

Also, I2P network uses a specialized distributed database called netDb (https://geti2p.net/en/docs/how/network-database) for routing. Regular clients just store their netDb localy, but there is a special mode called **floodfill**, which makes the node to maintain and distribute netDb to others.

To join I2P network client needs to get some initial netBb items. This process is called "reseeding" (https://geti2p.net/en/docs/reseed). **Reseed server** is a public HTTPS server which hosts signed archive with netDb items. Actually, clients can reseed in many other ways (including manual reseed from friends), HTTPS reseed is just the easiest way.

So, **what needs to be done** to create fully-functioning I2P darknet:

- run some I2P clients in the "floodfill" mode
- create a public "reseed" server, which will help new clients to bootstrap into I2P network
- create I2P client bundle for new users

# Toolbox

For demonstration purposes I will set up test network using LXC virtualization on a single Ubuntu 16.04 host. You can do similar steps with real computers in the Internet/LAN/CJDNS/etc, if you really want.

As I2P client I will use lightweight C++ implementation i2pd (http://i2pd.website).

As a reseed server I will use pyseeder (https://github.com/PurpleI2P/pyseeder).

# Prepare host system

```
# Install LXC and build tools
sudo apt-get install lxc bridge-utils build-essential git

# Install python3 with utilities and dependencies for building
# python-cryptography (for pyseeder)
sudo apt-get install python3 python3-pip python3-virtualenv \
    libssl-dev libffi-dev python-dev

# Prepare a working directory for lxc-testnet project
# Note that I will use $WDIR variable in many other shells
export WDIR=$HOME/lxc-testnet
mkdir $WDIR

# Download my set of shell scripts for managing LXC testnet
git clone https://github.com/l-n-s/i2pd-testnet-framework.git $WDIR

# (Optional) For some reason, apparmor in my Ubuntu box breaks LXC, so I disabled
# You could try to find another solution
sudo systemctl stop apparmor && sudo systemctl disable apparmor
```

# Build static i2pd

Run `build_static_i2pd.sh` in the working directory.

```
cd $WDIR && ./build_static_i2pd.sh
```

It will build static i2pd binary with all dependancies included, so you can copy it to other machines (with the same architecture) and it will just work.

# Reseed server

```
# Download pyseeder and install it's requirements
cd $WDIR
git clone https://github.com/PurpleI2P/pyseeder.git
cd $WDIR/pyseeder
virtualenv --python=python3 venv
. venv/bin/activate
pip3 install -r requirements.txt
deactivate
```

**Open a new shell** to do all pyseeder operations there and activate virtual environment:

```
cd $WDIR/pyseeder
. venv/bin/activate
```

Generate a new keypair for the reseed server. Just for this example, **set "dadadada" as password**:

```
./pyseeder.py keygen --signer-id mark@mail.i2p
```

This will create files `data/mark_at_mail.i2p.crt` and `data/priv_key.pem`. It is reseed public certificate for distributing with i2pd and private key for creating SU3 files and run HTTPS server.

By default, Ubuntu host exposes to LXC network interface `lxcbr0` with IP address 10.0.3.1. So reseed URL will be:

```
export RESEED_URL="https://10.0.3.1:8443/"
```

# Create binary distributive

For demonstration purposes, I will only create a distributive for Linux boxes.

```
# Create distributive folder and copy static binary to it
mkdir -p $WDIR/dist
cp $BUILD_DIR/src/i2pd/i2pd $WDIR/dist

# Copy reseed certificate
mkdir -p $WDIR/dist/certificates/reseed
cp $WDIR/pyseeder/data/mark_at_mail.i2p.crt $WDIR/dist/certificates/reseed
```

You will need to distribute this package with configuration file `i2pd.conf` for your network. `./testnetctl re-install` command generates `i2pd.conf` and run/stop scripts from `templates/` folder, so it's all automated with LXC testnet.

Inspect `docs/i2pd.conf` config file in i2pd repository (https://github.com/PurpleI2P/i2pd) if you wish to create network with real computers.

# Final preparations

Create 12 virtual computers with LXC and copy i2pd distributive there.

```
# Create 12 LXC containers (5 of them will be floodfills)
sudo ./testnetctl prepare 12
# Install i2pd to LXC containers
sudo ./testnetctl re-install
```

Now it is time to run some floodfill nodes and collect their `router.info` files.

You can manually copy `router.info` files from your machines, but make sure you name it as `something-random.dat` , because pyseeder looks for .dat files.

```
# Collect reseed data: run and stop i2pd to generate router.info file, then copy
sudo ./testnetctl collect_reseed_data
```

Almost there! Create a reseed file (i2pseeds.su3) from netDb folder. **Swith to pyseeder shell** and run:

```
echo "dadadada" | ./pyseeder.py reseed --signer-id mark@mail.i2p --netdb $WDIR/ne
```

Run HTTPS reseed server (script will prompt for a password, it is still "dadadada"):

```
./pyseeder.py serve --host 0.0.0.0 --cert data/mark_at_mail.i2p.crt
```

# ACTION!

Run I2P network:

```
sudo ./testnetctl run
```

Notice in pyseeder shell that clients are downloading .su3 file. After that, nodes will start discovering each other.

Open a new shell to run monitoring script:

```
cd $WDIR/monitoring
./scanner.sh iplist.txt
watch -n1 ./i2pd_monitor.sh iplist.txt
```

This simple script will display real-time statistics of all nodes in the network.

If the task is to reinstall i2pd folder `dist` to all containers:

```
sudo ./testnetctl re-install
```

If the task is to reinstall just `dist/i2pd` binary:

```
sudo ./testnetctl re-install-binary
```

Stop the network:

```
sudo ./testnetctl stop
```

To clean host system after experiments (delete all containers):

```
sudo ./testnetctl clean
```

# So, what's next?

Try to analyze network traffic between nodes and you'll notice that it's all mixed and encrypted. It's impossible to know for sure if someone is hosting a website in I2P, or using torrents, or just routing transit traffic.

Look at I2P tech-intro (https://geti2p.net/en/docs/how/tech-intro) and specs (https://geti2p.net/spec) for more details on how the protocol works. Use i2pd docs (https://i2pd.readthedocs.io/) for more information about setting up hidden services with i2pd.

If you have any questions, feel free to ask them at Freenode IRC channel #i2pd-dev (https://webchat.freenode.net/?channels=i2pd-dev).

---

Posted at 2016 Nov 13 by Invisible Villain  Tags: i2p, i2pd, darknet, research

Back to top