

# Forum:[Hack Forums](#)

Thread: [\[Socialites\] Algo VPN - Set Up a Personal IPSEC VPN in the Cloud](#)

SUI-3183372 - April 17, 2017, 10:32 pm

Download: [https://mega.nz/#!V5wgCI7S!o7K\\_dlw8H8Zhex8UfEYqQGSeG898nV75ZZI4TJ6Tkkl](https://mega.nz/#!V5wgCI7S!o7K_dlw8H8Zhex8UfEYqQGSeG898nV75ZZI4TJ6Tkkl)

## Algo VPN

Algo VPN is a set of Ansible scripts that simplify the setup of a personal IPSEC VPN. It uses the most secure defaults available, works with common cloud providers, and does not require client software on most devices. See our release announcement for more information.

FeaturesSupports only IKEv2, with a single cipher suite: AES-GCM, HMAC-SHA2, and P-256 DH

Generates Apple profiles to auto-configure iOS and macOS devices

Includes helper scripts to add and remove users

Blocks ads with a local DNS resolver and HTTP proxy (optional)

Sets up limited SSH users for tunneling traffic (optional)

Based on current versions of Ubuntu and strongSwan

Installs to DigitalOcean, Amazon EC2, Google Compute Engine, Microsoft Azure, or your own server

Anti-featuresDoes not support legacy cipher suites or protocols like L2TP, IKEv1, or RSA

Does not install Tor, OpenVPN, or other risky servers

Does not depend on the security of [TLS](#)

Does not require client software on most platforms

Does not claim to provide anonymity or censorship avoidance

Does not claim to protect you from the [FSB](#), [MSS](#), [DGSE](#), or [FSM](#)

## Deploy the Algo Server

The easiest way to get an Algo server running is to let it set up a new virtual machine in the cloud for you.

Setup an account on a cloud hosting provider. For example, DigitalOcean (most user friendly), Amazon EC2, Google Compute Engine, and Microsoft Azure.

[Download Algo](#) and unzip it in a convenient location on your local machine.

Install Algo's core dependencies. Open the Terminal. The python interpreter you use to deploy Algo must be python2. If you don't know what this means, you're probably fine. cd into the algo-master directory where you unzipped Algo, then run:

macOS:

Code \$ python -m ensurepip --user

\$ python -m pip install --user --upgrade virtualenv

Linux (deb-based):

Code \$ sudo apt-get update && sudo apt-get install \\\nbuild-essential \\\n

```
libssl-dev \
libffi-dev \
python-dev \
python-pip \
python-setuptools \
python-virtualenv -y
```

Windows:

Windows instructions in spoiler ([Click to View](#))

#### Start Windows Instructions

Before run Algo, you have to have: Windows 10 (Anniversary update or later version)

64-bit installation (can't run on 32-bit systems)

Once you verify your system is 64-bit and up to date, you have to do a few manual steps to enable the 'Windows Subsystem for Linux':

Open 'Settings'

Click 'Update & Security', then click the 'For developers' option on the left.

Toggle the 'Developer mode' option, and accept any warnings Windows pops up.

Wait a minute for Windows to install a few things in the background (it will eventually let you know a restart may be required for changes to take effect—ignore that for now). Next, to install the actual Linux Subsystem, you have to jump over to 'Control Panel', and do the following:

Click on 'Programs'

Click on 'Turn Windows features on or off'

Scroll down and check 'Windows Subsystem for Linux (Beta)', and then click OK.

The subsystem will be installed, then Windows will require a reboot. Reboot, then open up the start menu and enter 'bash' (to open up 'Bash' installation in a new command prompt). Fill out all the questions (it will have you create a separate user account for the Linux subsystem), and once that's all done (it takes a few minutes to install), you will finally have Ubuntu running on your Windows laptop, somewhat integrated with Windows.

Install additional packages:

```
Code sudo apt-get update && sudo apt-get install python-pip python-setuptools build-essential
libssl-dev libffi-dev python-dev python-virtualenv git -y
```

Clone the Algo repository:

```
Code git clone https://github.com/trailofbits/algo && cd algo
```

End Windows Instructions

Install Algo's remaining dependencies for your operating system. Use the same terminal window as the previous step and run:

```
Code $ python -m virtualenv env && source env/bin/activate && python -m pip install -U pip &&
python -m pip install -r requirements.txt
```

On macOS, you may be prompted to install cc. You should press accept if so.

Open config.cfg in your favorite text editor. Specify the users you wish to create in the users list.

Start the deployment. Return to your terminal. In the Algo directory, run ./algo and follow the instructions. There are several optional features available. None are required for a fully functional VPN server. These optional features are described in greater detail in [ansible-roles.md](#). That's it! You will get the message below when the server deployment process completes. You now have an Algo server on the internet. Take note of the p12 (user certificate) password in case you need it later.

You can now setup clients to connect it, e.g. your iPhone or laptop. Proceed to Configure the VPN Clients below.

Note: If you want to run Algo again at any point in the future, you must first "reactivate" the dependencies for it. To reactivate them, open your terminal, use `cd` to navigate to the directory with Algo, then run `source env/bin/activate`.

Advanced users who want to install Algo on top of a server they already own or want to script the deployment of Algo onto a network of servers, please see the [Advanced Usage documentation](#).

### Configure the VPN Clients

Distribute the configuration files to your users, so they can connect to the VPN. Certificates and configuration files that users will need are placed in the configs directory. Make sure to secure these files since many contain private keys. All files are saved under a subdirectory named with the IP address of your new Algo VPN server.

#### Apple Devices

Find the corresponding mobileconfig (Apple Profile) for each user and send it to them over AirDrop or other secure means. Apple Configuration Profiles are all-in-one configuration files for iOS and macOS devices. On macOS, double-clicking a profile to install it will fully configure the VPN. On iOS, users are prompted to install the profile as soon as the AirDrop is accepted.

#### Android Devices

You need to install the [strongSwan VPN Client for Android 4 and newer](#) because no version of Android supports IKEv2. Import the corresponding user.p12 certificate to your device.

#### Windows

Windows clients have a more complicated setup than most others. Follow the steps below to set one up:

Copy the CA certificate (cacert.pem), user certificate (\$user.p12), and the user PowerShell script (windows\_\$user.ps1) to the client computer.

Import the CA certificate to the local machine Trusted Root certificate store.

Open PowerShell as Administrator. Navigate to your copied files.

If you haven't already, you will need to change the Execution Policy to allow unsigned scripts to run.

Code `Set-ExecutionPolicy Unrestricted -Scope CurrentUser` In the same PowerShell window, run the included PowerShell script to import the user certificate, set up a VPN connection, and activate stronger ciphers on it.

After you execute the user script, set the Execution Policy back before you close the PowerShell window.

Code `Set-ExecutionPolicy Restricted -Scope CurrentUser` Your VPN is now installed and ready to use.

If you want to perform these steps by hand, you will need to import the user certificate to the Personal certificate store, add an IKEv2 connection in the network settings, then activate stronger ciphers on it via the following PowerShell script:

Code `Set-VpnConnectionIPsecConfiguration -ConnectionName "Algo"`  
`-AuthenticationTransformConstants SHA256128 -CipherTransformConstants AES256`  
`-EncryptionMethod AES256 -IntegrityCheckMethod SHA256 -DHGroup Group14 -PfsGroup none`

Linux Network Manager Clients (e.g., Ubuntu, Debian, or Fedora Desktop)

Network Manager does not support AES-GCM. In order to support Linux Desktop clients, please choose the "compatible" cryptography and use at least Network Manager 1.4.1. See [Issue #263](#) for more information.

Linux strongSwan Clients (e.g., OpenWRT, Ubuntu Server, etc.)

Install strongSwan, then copy the included `ipsec_user.conf`, `ipsec_user.secrets`, `user.crt` (user certificate), and `user.key` (private key) files to your client device. These will require customization based on your exact use case. These files were originally generated with a point-to-point OpenWRT-based VPN in mind.

Ubuntu Server 16.04 example

1. `sudo apt-get install strongswan strongswan-plugin-openssl`: install strongSwan
2. `/etc/ipsec.d/certs`: copy `user.crt` from `algo-master/configs/<name>/pki/certs`
3. `/etc/ipsec.d/private`: copy `user.key` from `algo-master/configs/<name>/pki/private`
4. `/etc/ipsec.d/cacerts`: copy `cacert.pem` from `algo-master/configs/<name>/cacert.pem`
5. `/etc/ipsec.secrets`: add your `user.key` to the list, e.g. `xx.xxx.xx.xxx : ECDSA user.key`
6. `/etc/ipsec.conf`: add the connection from `ipsec_user.conf` and update `leftcert` to match the `user.crt` filename
7. `sudo ipsec restart`: pick up config changes
8. `sudo ipsec up <conn-name>`: start the ipsec tunnel
9. `sudo ipsec down <conn-name>`: shutdown the ipsec tunnel

One common use case is to let your server access your local LAN without going through the VPN. Set up a passthrough connection by adding the following to `/etc/ipsec.conf`. Replace `192.168.1.1/24` with the subnet your LAN uses:

```
Code conn lan-passthrough
leftsubnet=192.168.1.1/24
rightsubnet=192.168.1.1/24
authby=never # No authentication necessary
type=pass # passthrough
auto=route # no need to ipsec up lan-passthrough
Other Devices
```

Depending on the platform, you may need one or multiple of the following files.`cacert.pem`: CA Certificate

`user.mobileconfig`: Apple Profile

`user.p12`: User Certificate and Private Key (in PKCS#12 format)

`user.sswan`: Android strongSwan Profile

`ipsec_user.conf`: strongSwan client configuration

`ipsec_user.secrets`: strongSwan client configuration

`windows_user.ps1`: Powershell script to help setup a VPN connection on Windows

Setup an SSH Tunnel

If you turned on the optional SSH tunneling role, then local user accounts will be created for each user in `config.cfg` and an SSH `authorized_key` files for them will be in the `configs` directory (`user.ssh.pem`). SSH user accounts do not have shell access, cannot authenticate with a password, and only have limited tunneling options (e.g., `ssh -N` is required). This is done to ensure that SSH users have the least access required to tunnel through the server and can perform no other actions.

Use the example command below to start an SSH tunnel by replacing `user` and `ip` with your own. Once the tunnel is setup, you can configure a browser or other application to use `127.0.0.1:1080` as a SOCKS proxy to route traffic through the Algo server.

Code `ssh -D 127.0.0.1:1080 -f -q -C -N user@ip -i configs/ip_user.ssh.pem`

### Adding or Removing Users

Algo's own scripts can easily add and remove users from the VPN server.

Update the users list in your config.cfg

Open a terminal, cd to the algo directory, and activate the virtual environment with `source env/bin/activate`

Run the command: `./algo update-users` The Algo VPN server now contains only the users listed in the config.cfg file.

Additional Documentation [Advanced Usage](#) describes how to deploy an Algo VPN server directly from Ansible.

[FAQ](#) includes answers to common questions.

[Roles](#) includes a description of optional Algo VPN server features.

[Troubleshooting](#) includes answers to common technical issues.

Download: [https://mega.nz/#!V5wgCI7S!o7K\\_dlw8H8Zhex8UfEYqQGSeG898nV75ZZI4TJ6TkKk](https://mega.nz/#!V5wgCI7S!o7K_dlw8H8Zhex8UfEYqQGSeG898nV75ZZI4TJ6TkKk)

Source: [TrailOfBits](#)

**Mr. Robot** - April 18, 2017, 12:55 am

Hey man thanks for the great contribution.

**mr robots hungry child** - April 18, 2017, 9:02 pm

This is extremely interesting and detailed. Well done!

**'OverSight** - April 19, 2017, 3:21 pm

Wonderful. I didn't read it all but would this be essentially logless?

**SUI-3183372** - April 19, 2017, 3:35 pm

Yes.

Then anything additional you don't want stored just point to /dev/null

**SUI-3183372** - October 17, 2017, 12:37 am

I've begun working with the Socialites usergroup to better maintain my privacy, security, and anonymity threads. This will help me push out thread updates sooner and include more content. This thread will be reviewed and new content should be added in the upcoming days.

**Mi2414** - October 18, 2017, 3:40 pm

I know this [article](#) is old, but does this still mean if someone has the PSKs then you are screwed?

**SUI-3183372** - October 18, 2017, 4:12 pm

(Oct 18, 2017 03:40 PM)Amidst Wrote: [CITE](#) I know this [article](#) is old, but does this still mean if someone has the PSKs then you are screwed?

Idk much about the NSA's attack capabilities and whether or not anything has changed since the time period this article talks about (2010), however that article does mention IPSec, which is the protocol this VPN is based off of:

Quote:But in 2010, the NSA had already developed tools to attack the most commonly used VPN encryption schemes: Secure Shell (SSH), Internet Protocol Security (IPSec), and Secure Socket Layer (SSL) encryption.

