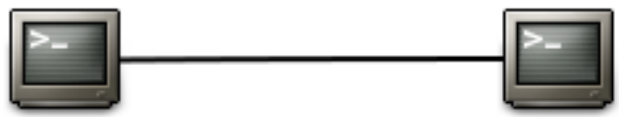
Sicher durch den Dschungel - Open Source VPN Lösungen

Daniel Ehlers

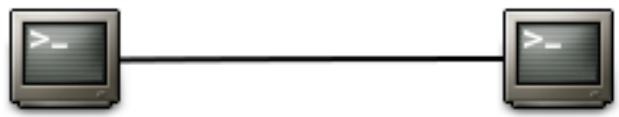
2. Oktober 2009

VPN ?



Netz mit Vertrauen

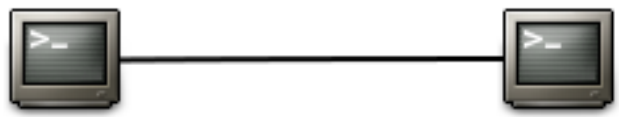
VPN ?



Netz mit Vertrauen

- Geräte in privatem Netz können Dienste nutzen

VPN ?



Netz mit Vertrauen

- Geräte in privatem Netz können Dienste nutzen
- Sensible Daten sind geschützt

VPN ?

Netz ohne Vertrauen

VPN ?

- Jeder kann Dienste nutzen

VPN ?

- Jeder kann Dienste nutzen
- Erreichbarkeit durch Router (NAT/Firewall) erschwert

VPN ?

- Jeder kann Dienste nutzen
- Erreichbarkeit durch Router (NAT/Firewall) erschwert
- Sensible Daten können mitgelesen werden

VPN ?

Was bietet ein VPN

- Nur Teilnehmer köennen auf Dienste zugreifen

VPN ?

Was bietet ein VPN

- Nur Teilnehmer köennen auf Dienste zugreifen
- Firewall/NAT Probleme werden umgangen

VPN ?

Was bietet ein VPN

- Nur Teilnehmer köennen auf Dienste zugreifen
- Firewall/NAT Probleme werden umgangen
- Daten werden verschlüsselt übertragen

VPN ?

Was bietet ein VPN

- Nur Teilnehmer köennen auf Dienste zugreifen
- Firewall/NAT Probleme werden umgangen
- Daten werden verschlüsselt übertragen
- Keine Anpassung an restlicher Software nötig

Implementierungsansätze

IPSec

- OpenSwan
- strongSWan
- KAME

TLS (SSL)

- OpenVPN
- cloudvpn
- tinc

Alternative

- n2n

OpenVPN

OpenVPN

Eckdaten

- Maintainer: OpenVPN Technologies Inc.

OpenVPN

Eckdaten

- Maintainer: OpenVPN Technologies Inc.
- WebSeite: http://www.openvpn.net/
- Status: Stable, 2.0.9 (1. Okt. 2009)

OpenVPN

Eckdaten

- Maintainer: OpenVPN Technologies Inc.
- WebSeite: http://www.openvpn.net/
- Status: Stable, 2.0.9 (1. Okt. 2009)
- Lizenz: GPLv2

OpenVPN

Eckdaten

- Maintainer: OpenVPN Technologies Inc.
- WebSeite: http://www.openvpn.net/
- Status: Stable, 2.0.9 (1. Okt. 2009)
- Lizenz: GPLv2
- Platformen: Windows (2000,XP,Vista), Linux,Open,Net,Free.. BSD, Mac OS X, Solaris

OpenVPN

Eckdaten

- Maintainer: OpenVPN Technologies Inc.
- WebSeite: http://www.openvpn.net/
- Status: Stable, 2.0.9 (1. Okt. 2009)
- Lizenz: GPLv2
- Platformen: Windows (2000,XP,Vista), Linux,Open,Net,Free.. BSD, Mac OS X, Solaris
- Verschlüsselung:
  - Grundlage liefert OpenSSL Bibliothek

OpenVPN

Eckdaten

- Maintainer: OpenVPN Technologies Inc.
- WebSeite: http://www.openvpn.net/
- Status: Stable, 2.0.9 (1. Okt. 2009)
- Lizenz: GPLv2
- Platformen: Windows (2000,XP,Vista), Linux,Open,Net,Free.. BSD, Mac OS X, Solaris
- Verschlüsselung:
  - Grundlage liefert OpenSSL Bibliothek
  - Hybrides Kryptosystem mit Zertifizierungsstelle (X.509)

OpenVPN

Eckdaten

- Maintainer: OpenVPN Technologies Inc.
- WebSeite: http://www.openvpn.net/
- Status: Stable, 2.0.9 (1. Okt. 2009)
- Lizenz: GPLv2
- Platformen: Windows (2000,XP,Vista), Linux,Open,Net,Free.. BSD, Mac OS X, Solaris
- Verschlüsselung:
  - Grundlage liefert OpenSSL Bibliothek
  - Hybrides Kryptosystem mit Zertifizierungsstelle (X.509)
  - oder Verifikation mit statischem Schlüssel

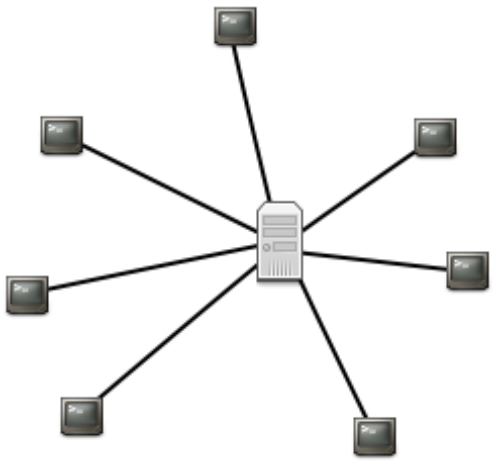
OpenVPN

Eckdaten

- Maintainer: OpenVPN Technologies Inc.
- WebSeite: http://www.openvpn.net/
- Status: Stable, 2.0.9 (1. Okt. 2009)
- Lizenz: GPLv2
- Platformen: Windows (2000,XP,Vista), Linux,Open,Net,Free.. BSD, Mac OS X, Solaris
- Verschlüsselung:
  - Grundlage liefert OpenSSL Bibliothek
  - Hybrides Kryptosystem mit Zertifizierungsstelle (X.509)
  - oder Verifikation mit statischem Schlüssel
  - oder Schlüssel und Benutzer/Passwort Abfrage

OpenVPN

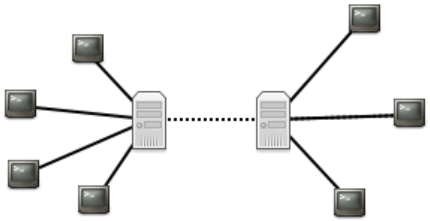
Topologie



- klassisches Server-Client Model

OpenVPN

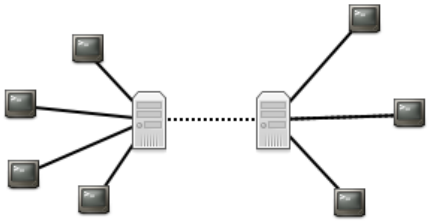
Topologie



- klassisches Server-Client Model
- mehrere Server über Netzwerk Brücken

OpenVPN

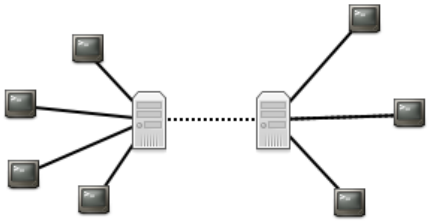
Topologie



- klassisches Server-Client Model
- mehrere Server über Netzwerk Brücken
  1. lokal auf einem System

OpenVPN

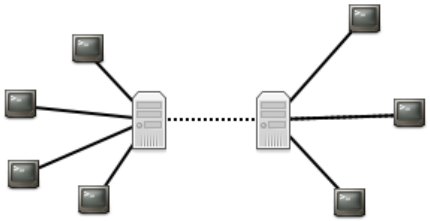
Topologie



- klassisches Server-Client Model
- mehrere Server über Netzwerk Brücken
  1. lokal auf einem System
  2. verbindung über vertrauenswürdige Netze

OpenVPN

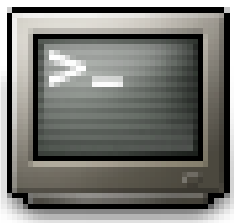
Topologie



- klassisches Server-Client Model
- mehrere Server über Netzwerk Brücken
  1. lokal auf einem System
  2. verbindung über vertrauenswürdige Netze
  3. über VPN Verbindung

OpenVPN

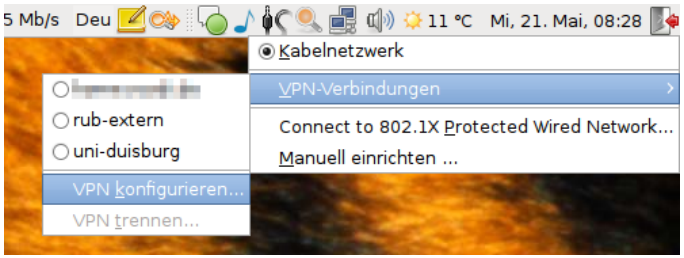
User Interfaces



- Konsolen Tools

OpenVPN

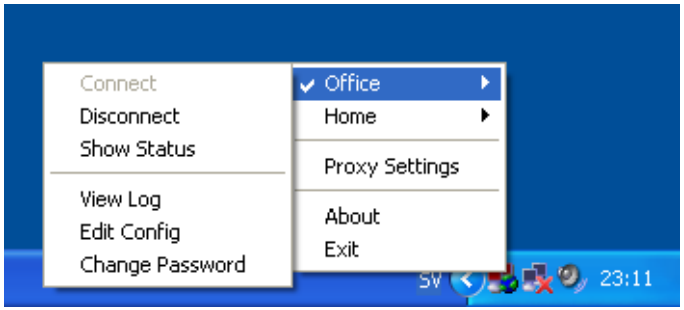
User Interfaces



- Konsolen Tools
- NetworkManager Plugin

OpenVPN

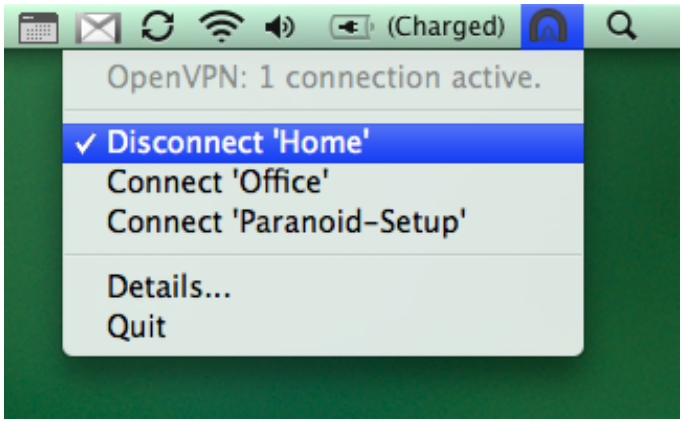
User Interfaces



- Konsolen Tools
- NetworkManager Plugin
- Windows GUI

OpenVPN

User Interfaces



- Konsolen Tools
- NetworkManager Plugin
- Windows GUI
- OS X (Tunnelblick)

OpenVPN

Konfiguration - Schluesselerstellung

```
1 #ca-host >. ./vars
2
3 #ca-host >./clean-all
4
5 #ca-host >./build-ca
6
7 #ca-host >./build-key-server server
8
9 #ca-host >./build-dh
10
11 #ca-host >./build-key client
```

- Skripts erleichtern den Prozess

OpenVPN

Konfiguration - Schluesselerstellung

```
1 #ca-host >. ./vars
2
3 #ca-host >./clean-all
4
5 #ca-host >./build-ca
6
7 #ca-host >./build-key-server server
8
9 #ca-host >./build-dh
10
11 #ca-host >./build-key client
```

- Skripts erleichtern den Prozess
- „vars„ ermöglicht Grundeinstellungen und minimiert die Formalismen

OpenVPN

Konfiguration - Server 1/2

```
1 # server type settings
2 local 85.10.215.47
3 port 1194
4 proto udp
5 dev tap0
6 comp-lzo
7
8 # Keys
9 ca   /etc/openvpn/keys/ca.crt
10 cert /etc/openvpn/keys/server.crt
11 key  /etc/openvpn/keys/server.key  # This file should be kept secret
12
13 dh /etc/openvpn/keys/dh2048.pem
14
15 # Maintain a record of client <-> virtual IP address
16 ifconfig-pool-persist ipp.txt
17
18 # Configure server mode for ethernet bridging.
19 server-bridge 10.100.100.1 255.255.255.0 10.100.100.2 10.100.100.255
20
21 # Allow clients to be able to "see" each other.
22 client-to-client
23
24 # Periodes of ping like messages
25 keepalive 5 120
26
27 # Select a cryptographic cipher.
28 cipher AES-256-CBC   # AES
```

Listing 1: server.conf

OpenVPN

Konfiguration - Server 2/2

```
30 # reduce daemon's privileges after initialization.
31 user nobody
32 group nogroup
33 # prevent accessing keys when privileges are dropped
34 persist-key
35 persist-tun
36
37 # Output a short status file
38 status /etc/openvpn/openvpn-status.log
39
40 # General logging
41 log-append  /var/log/openvpn.log
42 verb 6
43 mute 20
```

Listing 2: server.conf

OpenVPN

Konfiguration - Client

```
1 # Mark this as a client config
2 client
3
4 #Server Type
5 dev tap0
6 proto udp
7 remote vpn.plagis.de 1194
8 nobind
9 persist-key
10 persist-tun
11 comp-lzo
12
13 #Key Settings
14 ca keys/ca.crt
15 cert keys/client.crt
16 key keys/client.key
17
18 #Crypto
19 cipher AES-256-CBC
20 ns-cert-type server
```

Listing 3: client.conf

OpenVPN

Pro

- Getestete/Erprobte Software

OpenVPN

Pro

- Getestete/Erprobte Software
- Verwendet getestete/erprobte Komponenten

OpenVPN

Pro

- Getestete/Erprobte Software
- Verwendet getestete/erprobte Komponenten
- Breite Community Akzeptanz

OpenVPN

Pro

- Getestete/Erprobte Software
- Verwendet getestete/erprobte Komponenten
- Breite Community Akzeptanz
- GUI für alle Platformen

OpenVPN

Contra

- Kein Konzept zur Lastverteilung

cloudvpn

cloudvpn

Eckdaten

- Maintainer: [exa]

cloudvpn

Eckdaten

- Maintainer: [exa]
- Status: Stable, 1.99.8 (12 Sep. 2009)

cloudvpn

Eckdaten

- Maintainer: [exa]
- Status: Stable, 1.99.8 (12 Sep. 2009)
- Lizenz: GPLv3

cloudvpn

Eckdaten

- Maintainer: [exa]
- Status: Stable, 1.99.8 (12 Sep. 2009)
- Lizenz: GPLv3
- Plattformen: Linux, BSD, Win32, OSX (laut git)

cloudvpn

Eckdaten

- Maintainer: [exa]
- Status: Stable, 1.99.8 (12 Sep. 2009)
- Lizenz: GPLv3
- Plattformen: Linux, BSD, Win32, OSX (laut git)
- Verschlüsselung:
  - Grundlage liefert die gnuTLS Bibliothek

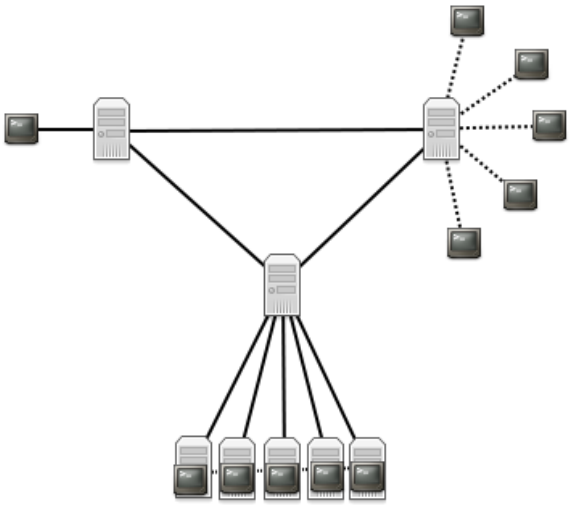
cloudvpn

Eckdaten

- Maintainer: [exa]
- Status: Stable, 1.99.8 (12 Sep. 2009)
- Lizenz: GPLv3
- Plattformen: Linux, BSD, Win32, OSX (laut git)
- Verschlüsselung:
  - Grundlage liefert die gnuTLS Bibliothek
  - Hybrides Kryptosystem mit Zertifizierungsstelle

cloudvpn

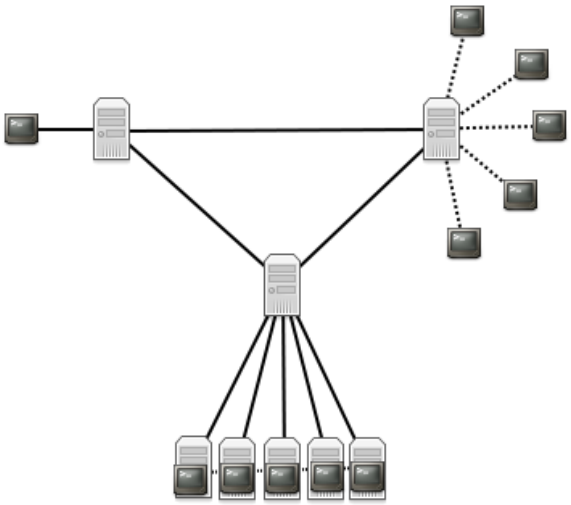
Topologie



- Erlaubt beliebige Topologien

cloudvpn

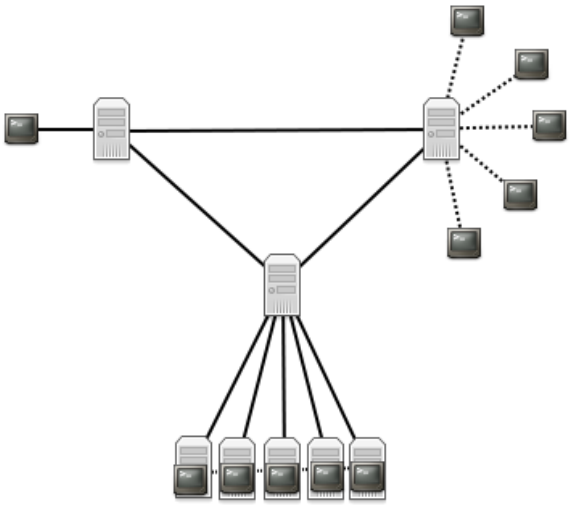
Topologie



- Erlaubt beliebige Topologien
- Knoten können als load balancer agieren

cloudvpn

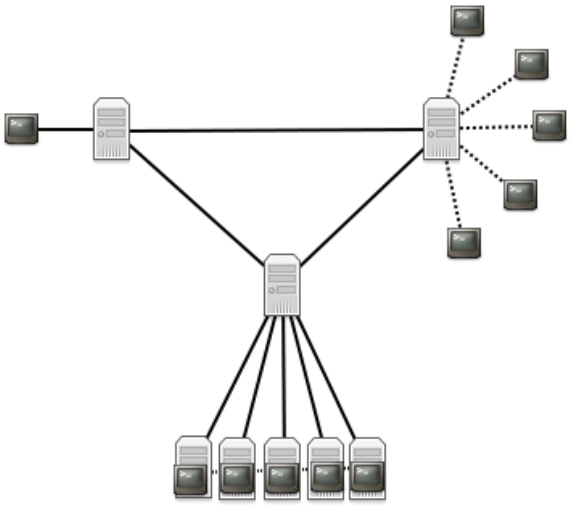
Topologie



- Erlaubt beliebige Topologien
- Knoten können als load balancer agieren
- Die eigentlichen Clients verbinden sich zum Mesh durch sog. Gates.

cloudvpn

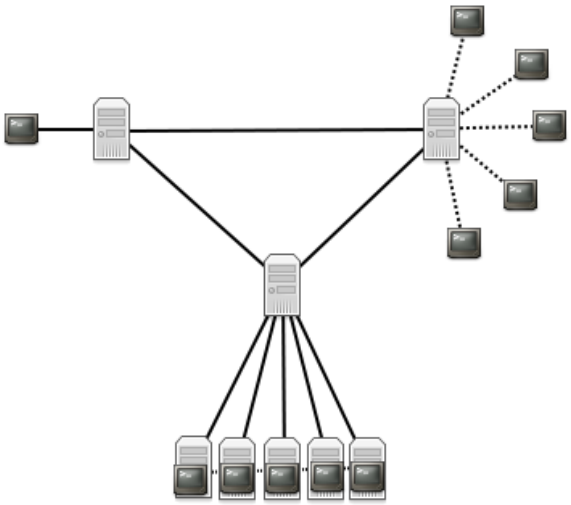
Topologie



- Erlaubt beliebige Topologien
- Knoten können als load balancer agieren
- Die eigentlichen Clients verbinden sich zum Mesh durch sog. Gates.
  - lokal über Sockets

cloudvpn

Topologie



- Erlaubt beliebige Topologien
- Knoten können als load balancer agieren
- Die eigentlichen Clients verbinden sich zum Mesh durch sog. Gates.
  - lokal über Sockets
  - oder über vertrauenswuerdiges Netz

cloudvpn

Konfiguration - MeshKnoten

```
1 #!/usr/bin/cloud -@include
2
3 ca ./ca.crt
4 key ./ssl.key
5 cert ./ssl.crt
6 dh ./dh1024.pem
7
8 listen 0.0.0.0 15135
9
10 status-file ./status.txt
11 status-interval 10000000
12
13 gate ./gate.sock
```

Listing 4: host1/darknet

```
1 #!/usr/bin/cloud -@include
2
3 ca ./ca.crt
4 key ./ssl2.key
5 cert ./ssl2.crt
6 dh ./dh1024.pem
7
8 connect 172.19.3.42 15135
9
10 heartbeat 10000000
11
12 status-file ./status.txt
13 status-interval 10000000
14
15 gate ./gate.sock
```

Listing 5: host2/darknet

cloudvpn

Konfiguration - Client

Die Clients verbinden sich zum Mesh.

```
1  ether -gate /path/to/socket &
2  ifconfig tap0 10.10.10.1
```

cloudvpn

Pro

- Sehr individual gestaltbares Netz
  - Ausfallsicherheit
  - Selbstreparatur

cloudvpn

Pro

- Sehr individual gestaltbares Netz
  - Ausfallsicherheit
  - Selbstreparatur
- “lightwight„ Lösung (folgt KISS Prinzip)

cloudvpn

Contra

- Keine Konzepte zum konfigurieren des Netzwerks

cloudvpn

Contra

- Keine Konzepte zum konfigurieren des Netzwerks
- Nur sehr grundlegend Dokumentiert

cloudvpn

Contra

- Keine Konzepte zum konfigurieren des Netzwerks
- Nur sehr grundlegend Dokumentiert
- Schlüsselverwaltung mit „certtool“ sehr aufwändig
- Keine GUI

tinc

tinc - There Is No Cabal

Eckdaten

- Maintainer: Guus Sliepen

tinc - There Is No Cabal

Eckdaten

- Maintainer: Guus Sliepen
- WebSeite: http://www.tinc-vpn.org/

tinc - There Is No Cabal

Eckdaten

- Maintainer: Guus Sliepen
- WebSeite: http://www.tinc-vpn.org/
- Status: stable, 1.0.9 (26 Dez. 2008)

tinc - There Is No Cabal

Eckdaten

- Maintainer: Guus Sliepen
- WebSeite: http://www.tinc-vpn.org/
- Status: stable, 1.0.9 (26 Dez. 2008)
- Lizenz: GPLv2

tinc - There Is No Cabal

Eckdaten

- Maintainer: Guus Sliepen
- WebSeite: http://www.tinc-vpn.org/
- Status: stable, 1.0.9 (26 Dez. 2008)
- Lizenz: GPLv2
- Plattformen: Linux, BSD, Windows (2000,XP,Vista,7) , OSX

tinc - There Is No Cabal

Eckdaten

- Maintainer: Guus Sliepen
- WebSeite: http://www.tinc-vpn.org/
- Status: stable, 1.0.9 (26 Dez. 2008)
- Lizenz: GPLv2
- Plattformen: Linux, BSD, Windows (2000,XP,Vista,7) , OSX
- Verschlüsselung:
  - Grundlage liefert OpenSSL library

tinc - There Is No Cabal

Eckdaten

- Maintainer: Guus Sliepen
- WebSeite: http://www.tinc-vpn.org/
- Status: stable, 1.0.9 (26 Dez. 2008)
- Lizenz: GPLv2
- Plattformen: Linux, BSD, Windows (2000,XP,Vista,7) , OSX
- Verschlüsselung:
  - Grundlage liefert OpenSSL library
  - Hybrides Kryptosystem (Blowfish-128-CBC)

tinc

Topologie

- Erlaubt beliebige Topologien

tinc

Topologie

- Erlaubt beliebige Topologien
- Unterscheidet sich von cloudvpn lediglich im Routing

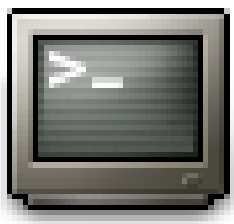
tinc

Topologie

- Erlaubt beliebige Topologien
- Unterscheidet sich von cloudvpn lediglich im Routing
- Jeder Knoten im Netzwerk ist auch Client

tinc

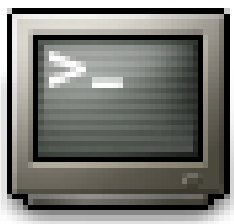
Control Tools



- (Unix) Konsolen Tools

tinc

Control Tools



- (Unix) Konsolen Tools
- (Windows) Konsolen Tools + Service

tinc

Konfiguration - Grundstruktur

Für jedes tinc Netz wird ein Unterordner in /etc/tinc/ angelegt.

```
/etc/tinc/darknet/
/etc/tinc/darknet/rsa_key.priv
/etc/tinc/darknet/tinc-up
/etc/tinc/darknet/tinc-down
/etc/tinc/darknet/tinc.conf
/etc/tinc/darknet/hosts/
```

tinc

Konfiguration - Host (Alpha)

```
1 Name = alpha
2
3 ConnectTo = beta
4
5 Device = /dev/net/tun
```

Listing 6: darknet/tinc.conf

```
1 # The real IP address of this tinc host. Can be used by other tinc hosts.
2 Address = 172.19.3.42
3
4 # Portnumber for incoming connections. Default is 655.
5 Port = 655
6
7 # Subnet on the virtual private network that is local for this host.
8 Subnet = 192.168.1.0/24
9
10 # The public key generated by 'tincd -n darknet -K' is stored here
11 -----BEGIN RSA PUBLIC KEY-----
12 ...
13 -----END RSA PUBLIC KEY-----
```

Listing 7: darknet/hosts/alpha

```
1 #!/bin/sh
2 ifconfig $INTERFACE 192.168.1.1 netmask 255.255.0.0
```

Listing 8: darknet/tinc-up

tinc

Konfiguration - Host (Beta)

```
1 Name = beta
2
3 ConnectTo = alpha
4
5 Device = /dev/net/tun
```

Listing 9: darknet/tinc.conf

```
1 # The real IP address of this tinc host. Can be used by other tinc hosts.
2 Address = 172.19.3.27
3
4 # Portnumber for incoming connections. Default is 655.
5 Port = 6500
6
7 # Subnet on the virtual private network that is local for this host.
8 Subnet = 192.168.2.0/24
9
10 # The public key generated by 'tincd -n example -K' is stored here
11 -----BEGIN RSA PUBLIC KEY-----
12 ....
13 -----END RSA PUBLIC KEY-----
```

Listing 10: darknet/hosts/beta

```
1 #!/bin/sh
2 ifconfig $INTERFACE 192.168.2.1 netmask 255.255.0.0
```

Listing 11: darknet/tinc-up

tinc

Konfiguration - Betrieb

- Host Dateien verteilen

tinc

Konfiguration - Betrieb

- Host Dateien verteilen
- Daemon starten

```
#alpha > tincd -n darknet
#beta > tincd -n darknet
```

tinc

Konfiguration - Betrieb

- Host Dateien verteilen
- Daemon starten

```
#alpha > tincd -n darknet
#beta > tincd -n darknet
```

- Daten austauschen

```
#alpha > ping 192.168.2.1
```

tinc

Pro

- Einfach gehaltenes Konfigurationsprinzip

tinc

Pro

- Einfach gehaltenes Konfigurationsprinzip
- Individuell gestallbares Netz
  - Ausfallsicherheit
  - Selbstreparatur

tinc

Pro

- Einfach gehaltenes Konfigurationsprinzip
- Individuell gestallbares Netz
  - Ausfallsicherheit
  - Selbstreparatur
- Verwendet getestete/erprobte Komponenten

tinc

Pro

- Einfach gehaltenes Konfigurationsprinzip
- Individuell gestallbares Netz
  - Ausfallsicherheit
  - Selbstreparatur
- Verwendet getestete/erprobte Komponenten

tinc

Pro

- Einfach gehaltenes Konfigurationsprinzip
- Individuell gestallbares Netz
  - Ausfallsicherheit
  - Selbstreparatur
- Verwendet getestete/erprobte Komponenten
- Gut dokumentiert

tinc

Contra

- Keine GUI

n2n

n2n

Eckdaten

- Maintainer: ntop

n2n

Eckdaten

- Maintainer: ntop
- WebSeite: http://www.ntop.org/n2n/

n2n

Eckdaten

- Maintainer: ntop
- WebSeite: http://www.ntop.org/n2n/
- Status: 1.3.3

n2n

Eckdaten

- Maintainer: ntop
- WebSeite: http://www.ntop.org/n2n/
- Status: 1.3.3
- Lizenz: GPLv3

n2n

Eckdaten

- Maintainer: ntop
- WebSeite: http://www.ntop.org/n2n/
- Status: 1.3.3
- Lizenz: GPLv3
- Platformen: Windows, Linux, OS X, FreeBSD

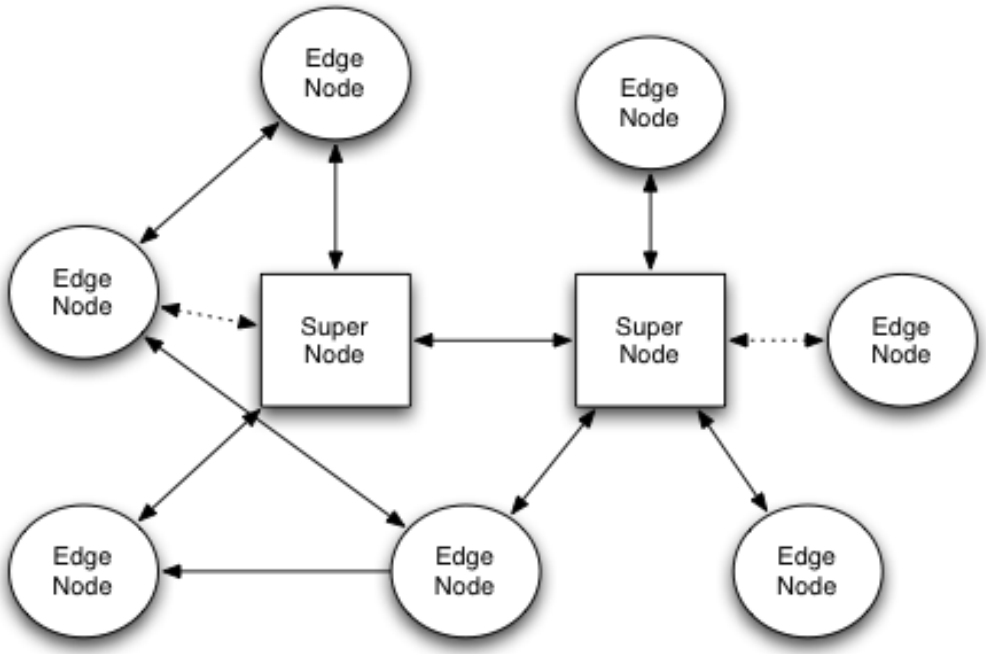
n2n

Eckdaten

- Maintainer: ntop
- WebSeite: http://www.ntop.org/n2n/
- Status: 1.3.3
- Lizenz: GPLv3
- Platformen: Windows, Linux, OS X, FreeBSD
- Verschlüsselung: Symmetrisches Kryptosystem (TwoFish)

n2n

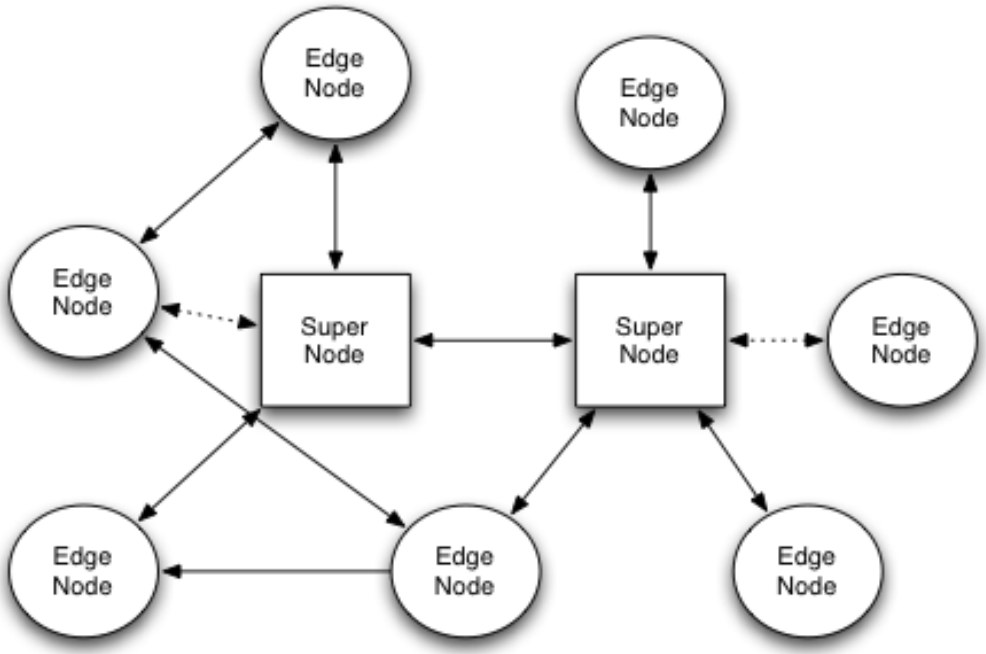
Topologie



- Benötigt min. einen SuperNode (Server)

n2n

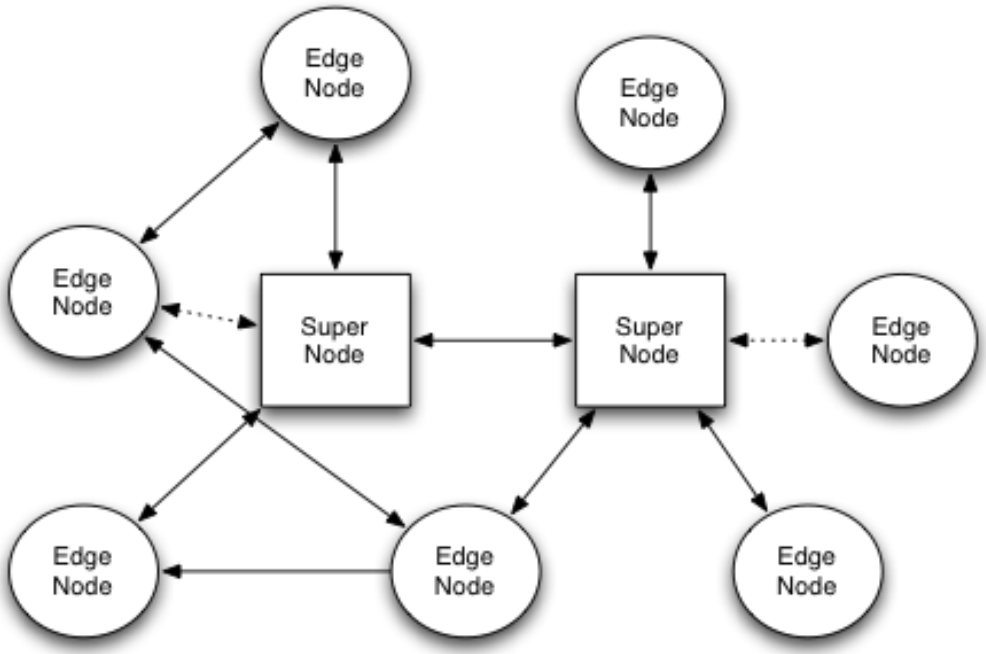
Topologie



- Benötigt min. einen SuperNode (Server)
- Edges stellen direkte Verbindung zu kommunikations Partnern her.

n2n

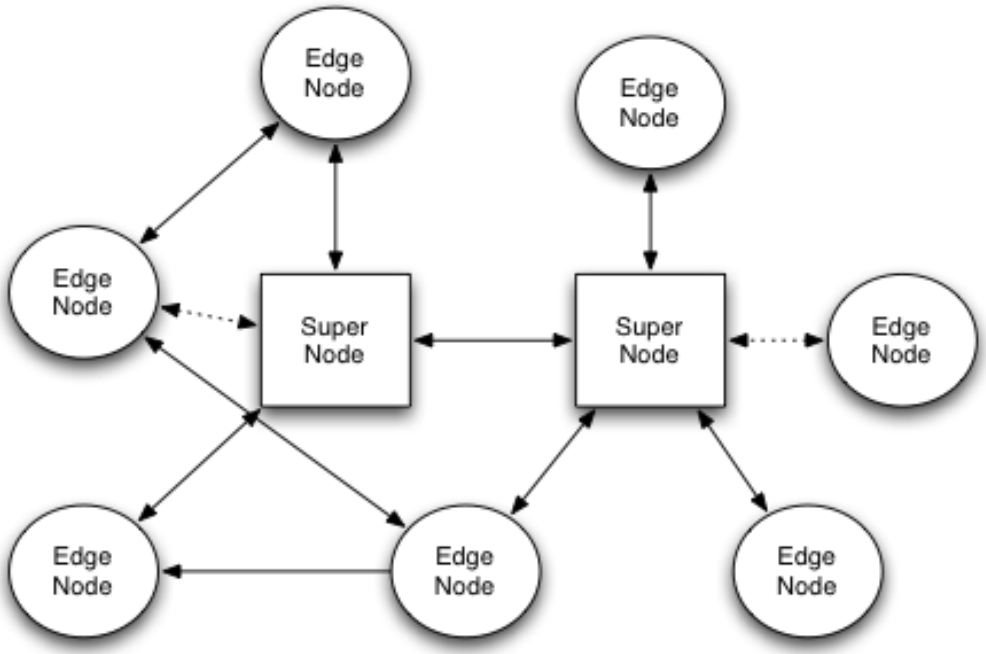
Topologie



- Benötigt min. einen SuperNode (Server)
- Edges stellen direkte Verbindung zu kommunikations Partnern her.
- NATs werden umschifft.

n2n

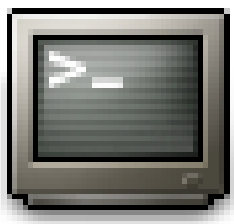
Topologie



- Benötigt min. einen SuperNode (Server)
- Edges stellen direkte Verbindung zu kommunikations Partnern her.
- NATs werden umschifft.
- Traffic läuft nur im Notfall über SuperNode

n2n

User Interfaces



- Konsolen Tools

n2n

User Interfaces

- Konsolen Tools
- Windows GUI

n2n

Konfiguration

- Starten des SuperNodes

  `#supernode > supernode -l <port>`

n2n

Konfiguration

- Starten des SuperNodes

```
#supernode > supernode -l <port>
```

- Verbinden der Edge Knoten

```
#edge node1> edge -a 10.1.2.1 -c linuxTage -k <key> \
             -l <supernode>:<port>
#edge node2> edge -a 10.1.2.2 -c linuxTage -k <key> \
             -l <supernode>:<port>
```

n2n

Konfiguration

- Starten des SuperNodes

```
#supernode > supernode -l <port>
```

- Verbinden der Edge Knoten

```
#edge node1> edge -a 10.1.2.1 -c linuxTage -k <key> \
             -l <supernode>:<port>
#edge node2> edge -a 10.1.2.2 -c linuxTage -k <key> \
             -l <supernode>:<port>
```

- Daten austauschen

```
#edge node1> ping 10.1.2.2
```

n2n

Pro

- Daten werden direkt zwischen Partnern versendet

n2n

Pro

- Daten werden direkt zwischen Partnern versendet
- Minimaler Konfigurationsaufwand

n2n

Pro

- Daten werden direkt zwischen Partnern versendet
- Minimaler Konfigurationsaufwand
- NATs werden umgangen.

n2n

Contra

- Lediglich statischer (pre Shared) Schlüssel unterstützt

n2n

Contra

- Lediglich statischer (pre Shared) Schlüssel unterstützt
- Keine authentifizierungs Konzepte

Fragen ?

Vielen Dank für Ihre Aufmerksamkeit