

VIRTUAL PRIVATE NETWORK

INDICE

INTRODUZIONE
TINC
PREPARAZIONE DEL SISTEMA
INSTALLAZIONE
CONFIGURAZIONE
RUNTIME DI TINC
CONCLUSIONI

INTRODUZIONE

- Opportunità per una azienda
- Aspetti principali delle VPN

INTRODUZIONE

Una VPN (Virtual Private Network) è una soluzione sempre più diffusa per collegare tra loro due reti private attraverso una rete pubblica in maniera sicura

Tale applicazione sfrutta i meccanismi di trasporto della rete pubblica (ad es. Internet) per collegare in maniera trasparente, ossia come se fossero collegate in maniera diretta, due reti remote.

QUALI OPPORTUNITA'?

In genere si ricorre ad una **Intranet**, per poter contare sulla possibilità di accesso da parte dei dipendenti aziendali anche dislocati in un'area geografica più o meno diffusa.

I problemi relativi a questa soluzione sono:

- Gestione dei database aziendali: è richiesto un aggiornamento costante e frequente
- Costi di connessione elevati: tale sistema richiede molte ore di collegamento (e quindi di telefonate)

QUALI OPPORTUNITA'?

In genere si ricorre ad una **Intranet**, per poter contare sulla possibilità di accesso da parte dei dipendenti aziendali anche dislocati in un'area geografica più o meno diffusa.

I problemi relativi a questa soluzione sono:

- Gestione dei database aziendali: è richiesto un aggiornamento costante e frequente
- Costi di connessione elevati: tale sistema richiede molte ore di collegamento (e quindi di telefonate)

Reti locali interne possedute da società private per permettere la comunicazione tra gli uffici. Sfrutta le tecnologie di Internet

QUALI OPPORTUNITA'?

Nel caso poi l'azienda, oltre che ai propri collaboratori interni, senta la necessità di poter condividere i propri database anche con partner, fornitori o altre figure esterne, non si servirà di una Intranet, bensì di una **Extranet**

Rete esterna posseduta da società private per permettere la comunicazione tra gli uffici. Sfrutta le tecnologie di Internet

PROBLEMI:

- estendere a tutti gli aggiornamenti dei database
- continua implementazione di tutti gli strumenti (software, protocolli, misure di sicurezza, server, ecc.)

7

QUALI OPPORTUNITA'?

Spesso si decide di ricorrere a **linee dedicate** tramite l'affitto di circuiti diretti numerici, Cdn

Grazie a tale soluzione, i costi non saranno più legati al tempo di connessione alla rete pubblica, ma dipenderanno dalla larghezza di banda di cui un'azienda necessita

PROBLEMI:

- Gestione diretta, per la fruizione dei servizi Ip
- Acquistare nuovi prodotti hardware
- Preoccuparsi dei vari problemi di sicurezza che derivano dai contatti fra la propria Intranet (o Extranet) ed Internet

8

QUALI OPPORTUNITA'?

Spesso si decide di ricorrere a **linee dedicate** tramite l'affitto di circuiti diretti numerici, Cdn

Grazie a tale soluzione, i costi non saranno più legati al tempo di connessione alla rete pubblica, ma dipenderanno dalla larghezza di banda di cui un'azienda necessita

Tipo di connessione realizzato tramite un cavo ad essa riservato. Questa connessione è indicata anche col nome di Cdn (Circuiti diretti numerici)

PROBLEMI:

- Gestione diretta, per la fruizione dei servizi Ip
- Acquistare nuovi prodotti hardware
- Preoccuparsi dei vari problemi di sicurezza che derivano dai contatti fra la propria Intranet (o Extranet) ed Internet

9

QUALI OPPORTUNITA'?

Una possibile soluzione alternativa, meno costosa, è rappresentata dalle **VPN (Virtual Private Network)**

Il termine stesso, Reti Private Virtuali, indica che la loro funzione è la medesima di una rete geografica aziendale di altro tipo: garantire un canale privilegiato per la trasmissione dei dati aziendali tra sedi fisiche differenti, o partner esterni

La differenza, tuttavia, è sostanziale. Anziché creare reti geografiche fisicamente private (con linee dedicate, ecc.), si utilizza, per il trasferimento dati tra due reti aziendali distanti fra loro, il canale pubblico, Internet

10

QUALI OPPORTUNITA'?

Wide Area Network, rete geografica. Rete di computer che ricopre un'area geografica molto estesa, e che interconnette sia singoli host che LAN o MAN, dislocati anche in continenti diversi (cfr. internetworking). Negli uffici pubblici spesso si creano delle WAN tra host, attraverso la normale linea telefonica, oppure per mezzo di linee dedicate, o via satellite.

In questo modo, creando comunque una **Wan**, macchine, che lavorano in qualunque parte del mondo, potranno essere connesse tra loro come in presenza di una linea diretta, ma a costi decisamente inferiori, operando quindi in remoto

Il principale vantaggio a livello economico, è proprio quello di affidare a terzi gli oneri riguardanti il trasferimento dei propri dati

11

SOLO VANTAGGI?

Il principale problema derivante dall'adozione di una simile soluzione è quello di poter trasferire dati su un canale pubblico garantendone **riservatezza e sicurezza** rispetto ad intrusioni esterne.

La sicurezza e la riservatezza dei dati che viaggiano attraverso una VPN sono garantite dal meccanismo che è alla base della stessa: il **tunneling**

12

TUNNELING

E' un insieme di tecniche che permettono di creare un collegamento per la trasmissione di dati tra sedi remote sfruttando la rete Open Internet generando ogni volta un percorso virtuale e privato (VPN)

Infatti il termine vuole proprio indicare che i dati, pur viaggiando in un canale pubblico, vengono comunque incanalati in una sorta di circuito (tunnel) preferenziale, e resi inaccessibili da chiunque non possieda l'autorizzazione

Il meccanismo del tunneling viene implementato attraverso due strumenti fondamentali: **autenticazione** e **cifratura**

13

CIFRATURA

I dati inviati attraverso una VPN vengono cifrati grazie ad algoritmi matematici nel momento in cui escono da una sede aziendale, spesso vengono anche compressi per rendere più rapida la loro trasmissione

Una volta giunti a destinazione verranno decifrati ed, eventualmente decompressi

14

CIFRATURA/2

In genere gli algoritmi di crittografia si appoggiano a schemi a chiave asimmetrica.

Tale schema prevede che ogni utente abbia una chiave **pubblica** ed una **privata**

Ogni utente cifra con la chiave pubblica del destinatario e decifra con la propria chiave privata

In una VPN, in genere, tali schemi vengono utilizzati per individuare, periodicamente, una chiave segreta unica e nota ad entrambi, essa consente sia di cifrare che decifrare un messaggio

15

CIFRATURA/3

Tale soluzione permette infatti di accelerare il processo di trasferimento dei dati che, con chiavi asimmetriche, verrebbe rallentato dagli stessi processi di cifratura e decifratura

Grazie all'utilizzo di algoritmi a chiave pubblica, la chiave segreta non è intellegibile da un esterno, garantendone così la massima riservatezza

16

AUTENTICAZIONE

In un sistema di telecomunicazioni come la VPN c'è bisogno di una procedura codificata (detta protocollo) con cui il sistema:

- autorizza l'accesso alla rete
- assicura che la trasmissione digitale dei dati sia indirizzata al destinatario corretto
- garantisce a quest'ultimo la certezza della sua provenienza.

17

FUNZIONAMENTO GENERALE DI UNA VPN

Gli utenti di una VPN, per comunicare tra loro, effettuano i seguenti tre passi:

1. si identificano seguendo lo schema dettato dal protocollo di autenticazione
2. effettuano la cifratura dei dati per garantire la riservatezza della propria comunicazione
3. spediscono i dati sfruttando la rete OpenInternet

18

TINC

- Caratteristiche principali
- Sicurezza di TINC
- Schema di autenticazione
- Funzionamento generale di TINC
- Versioni di TINC

TINC

TINC è un acronimo di There Is No Cabal dove Cabal era un'organizzazione nata per controllare il traffico di Internet. Poiché questo è ciò che una rete privata virtuale vuole evitare, il progetto della VPN è stato chiamato TINC

Processo server che generalmente opera in background e aderisce ad un servizio

Abbiamo scelto TINC perché è una soluzione software alquanto efficiente per l'implementazione di una VPN in quanto è un **demon** che ci permette di configurare una VPN in maniera semplice e con un numero di comandi limitato

MECCANISMO DEL TUNNELING

L'aspetto del **tunneling** implementato da TINC sfrutta, per la maggior parte dei dati, le proprietà del protocollo **UDP** che permette di incapsulare un pacchetto, compresso facoltativamente, e cifrarlo in un datagram IP

Tale tecnica consente di ridurre il traffico sulla rete, rispetto all'utilizzo esclusivo del protocollo **TCP**

Come già accennato precedentemente il meccanismo del **tunneling** è implementato tramite due strumenti fondamentali atti a garantire la sicurezza e la riservatezza dei dati:

CIFRATURA & AUTENTICAZIONE

MECCANISMO DEL TUNNELING

L'aspetto del **tunneling** implementato da TINC sfrutta, per la maggior parte dei dati, le proprietà del protocollo **UDP** che permette di incapsulare un pacchetto, compresso facoltativamente, e cifrarlo in un datagram IP

Tale tecnica consente di ridurre il traffico sulla rete, rispetto all'utilizzo esclusivo del protocollo **TCP**

Come già accennato precedentemente il meccanismo del **tunneling** è implementato tramite due strumenti fondamentali atti a garantire la sicurezza e la riservatezza dei dati:

CIFRATURA & AUTENTICAZIONE

MECCANISMO DEL TUNNELING: CIFRATURA

L'aspetto della **cifratura** è curato da TINC tramite le funzioni implementate nella libreria OpenSSL

Un pacchetto, opzionalmente compresso, verrà concatenato con un numero progressivo di 32 bit, dopodiché il risultato sarà cifrato utilizzando (per default) l'algoritmo di cifratura blowfish in modalità CBC con chiavi di 128 bit

MECCANISMO DEL TUNNELING: AUTENTICAZIONE

Per quanto riguarda l'**autenticazione**, TINC (dalla versione 1.0pre3 in poi) utilizza un protocollo che sfrutta lo schema a chiave pubblica RSA per lo scambio delle chiavi simmetriche

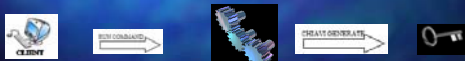
Tali chiavi simmetriche saranno poi utilizzate per la cifratura e decifratura dei dati spediti

VEDIAMO I PASSI DELLO SCHEMA DI AUTENTICAZIONE →

SCHEMA DI AUTENTICAZIONE PASSO I:GENERAZIONI DELLE CHIAVI RSA

Il client (server) esegue la procedura di configurazione durante la quale:

- genera il file relativo alle proprie variabili
- genera il file contenente le informazioni necessarie agli altri host per poter comunicare con esso
- lancia un comando specifico per generare la propria coppia di chiavi RSA, le quali saranno memorizzate in formato **PEM** ad un path di default o ad un specificato dall'utente



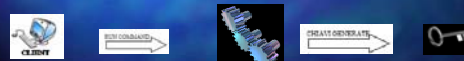
25

SCHEMA DI AUTENTICAZIONE PASSO I:GENERAZIONI DELLE CHIAVI RSA

Il client (server) esegue la procedura di configurazione durante la quale:

- genera il file relativo alle proprie variabili
- genera il file contenente le informazioni necessarie agli altri host per poter comunicare con esso
- lancia un comando specifico per generare la propria coppia di chiavi RSA, le quali saranno memorizzate in formato **PEM** ad un path di default o ad un specificato dall'utente

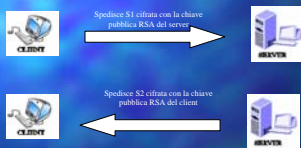
Privacy Enhanced Mail, è uno standard dall'Internet Activities Board) per fornire posta sicura su Internet e per la scrittura di chiavi e certificati in file ASCII. Usa due Encapsulation Block (EB):
-----BEGIN RSA KEY -----
-----END RSA KEY -----
Tra gli EB viene inserita la chiave codificata in BASE 64 secondo l'algoritmo descritto nell'RFC.



26

SCHEMA DI AUTENTICAZIONE PASSO II:SCAMBO DI CHIAVI (META_KEY)

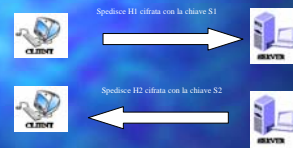
Il client genera S1, una stringa di bits totalmente casuale
Il server genera S2, una stringa di bits totalmente casuale



27

SCHEMA DI AUTENTICAZIONE PASSO III:RICHIESTA DI VERIFICA (CHALLENGE)

Il client genera una stringa di bits casuali H1
Il server genera una stringa di bits casuali H2

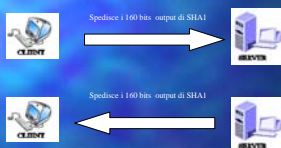


Il client (server) invia una stringa casuale al server (client) cifrata con la propria chiave simmetrica. Tale messaggio è utilizzato per richiedere una verifica di autenticazione

28

SCHEMA DI AUTENTICAZIONE PASSO IV:RISPOSTA ALLA VERIFICA (CHAL_REPLY)

Il client applica SHA1 su H2
Il server applica SHA1 su H1



Per il I passo il server (client) "autentico" riuscirà a rispondere alla richiesta di verifica inviategli dal client (server)

29

SCHEMA DI AUTENTICAZIONE PROPRIETA'

1. Tale schema permette di spedire i messaggi di autenticazione simultaneamente, non c'è bisogno di aspettare l'altro peer per spedirli. Questo significa che anche le operazioni di calcolo (cifratura e decifratura **RSA**) possono essere fatte in parallelo con la spedizione e ricezione dei messaggi

Rivest, Shamir, Adelman. Sistema di crittografia a chiave pubblica, il cui nome è formato dalle iniziali degli autori. Cifrato a chiave asimmetrica usato per la sicurezza Web, in cui vengono generate due chiavi, una pubblica usata per la cifratura ed una privata per la decifratura, con conseguente vantaggio rispetto ai cifrari simmetrici (unica chiave per cifrare e decifrare il messaggio) di non dover comunicare mai la chiave privata ma solamente quella pubblica. E' basato sulla difficoltà di calcolare funzioni con algoritmi discreti ma soprattutto di calcolare la fattorizzazione di numeri ottenuti moltiplicando numeri primi molto grandi

30

SCHEMA DI AUTENTICAZIONE PROPRIETA'

- Viene inviato un solo messaggio cifrato di RSA anziché due. Questo migliora la velocità e riduce il volume di informazioni che un attaccante potrebbe scoprire.
- Prima vengono scambiate le chiavi di cifratura e in seguito viene effettuata la verifica. Questo previene attacchi *meet-in-the-middle*.

Attacco che consiste nel "mettersi in mezzo" tra due host per captarne i segreti e/o cercare di spacciarsi per uno di loro. In dettaglio l'attaccante cerca di scoprire la chiave di cifratura e decifratura frapponendosi fra i due host e captando coppie di messaggi 'X' 'Y' che rappresentano rispettivamente il testo in chiaro e il testo cifrato. Un esempio di utilizzo di tale attacco è quello portato al DES doppio e triplicato che riduce le potenzialità di tali schemi di cifratura.

31

SCHEMA DI AUTENTICAZIONE PROPRIETA'

- Siccome lo scambio di chiavi è avvenuto in maniera sicura anche lo scambio di messaggi sarà tale. Infatti solo il vero destinatario di un pacchetto conoscerà la chiave privata per decifrarlo.
- La prima cosa spedita è una stringa totalmente random in modo da evitare attacchi di tipo *known plaintext*.

Attacco che consiste nel provare a indovinare la chiave privata. In particolare conoscendo le coppie testo in chiaro - testo cifrato si precomputa un certo numero di chiavi (naturalmente non tutto) e controlla se tra queste c'è quella utilizzata dall'algoritmo di cifratura.

32

FUNZIONAMENTO GENERALE DI TINC: AUTENTICAZIONE

Come per ogni VPN gli utenti di TINC effettueranno i seguenti tre passi:

- si identificano seguendo il protocollo di autenticazione

Tale protocollo utilizza lo schema a chiave pubblica RSA per lo scambio sicuro delle chiavi simmetriche usate in seguito per la cifratura - decifratura dei dati spediti

Il protocollo di autenticazione implementato da TINC garantisce:

- l'accesso alla rete
- che solo il legittimo destinatario riesca a leggere il messaggio
- l'autenticità del mittente di un messaggio

33

FUNZIONAMENTO GENERALE DI TINC: AUTENTICAZIONE/2

Come già accennato lo schema di autenticazione implementato da TINC garantisce l'accesso alla rete, ossia la possibilità di scambiarsi dati cifrati, ai soli utenti che hanno effettuato tutti i passi dello stesso

Un "utente malizioso", ovvero che non ha eseguito lo schema di autenticazione, potrà solo inviare messaggi cifrati con una chiave "simmetrica" del tutto casuale ad un altro host della VPN

Quest'ultimo non possedendo la chiave "simmetrica" suddetta, non sarà in grado di decifrarli

34

FUNZIONAMENTO GENERALE DI TINC: AUTENTICAZIONE/3

Se un "utente malizioso", invece, legge di nascosto un messaggio cifrato inviato sulla VPN, non sarà in grado di decifrarlo, non possedendo la chiave simmetrica scambiata durante lo schema di autenticazione tra mittente e destinatario

Questo garantisce, oltre all'accesso alla rete, anche che solo il legittimo destinatario riesca a leggere il messaggio

35

FUNZIONAMENTO GENERALE DI TINC: AUTENTICAZIONE/4a

L'autenticità del mittente di un messaggio cifrato è garantita al destinatario dal fatto che lo schema di autenticazione previsto da TINC impedisce attacchi di tipo meet-in-the-middle

VEDIAMO COME!!!

36

FUNZIONAMENTO GENERALE DI TINC: AUTENTICAZIONE/4b

Un utente malizioso che vorrebbe ingannare l'host **B**, spacciandosi per un host generico **A** non potrà farlo

L'unico modo per far credere a **B** di essere in comunicazione con **A** sarebbe quello di cifrare e decifrare con le medesime chiavi simmetriche che **A** si è scambiato con **B** all'atto dell'autenticazione

Poiché lo scambio delle chiavi simmetriche tra **A** e **B** è avvenuto in maniera sicura il **MITM** non ne è a conoscenza e non potrà dunque spacciarsi per **A**

man in the middle: colui che effettua un attacco di tipo meet-in-the-middle

37

FUNZIONAMENTO GENERALE DI TINC

2. cifrano i dati seguendo il processo di cifratura di TINC

Il processo di cifratura di default, descritto precedente, può essere anche modificato settando le relative variabili di configurazione

E' possibile, inoltre, evitare che il pacchetto subisca il processo di cifratura settando in modo opportuna la relativa variabile di configurazione (fortemente sconsigliato)

38

FUNZIONAMENTO GENERALE DI TINC

3. spediscono i dati sfruttando OpenInternet

Come già detto in precedenza l'utente di una VPN implementata da TINC spedisce i propri messaggi sfruttando le proprietà del protocollo UDP

Un utente può anche decidere di spedire i propri messaggi utilizzando il protocollo TCP, settando l'apposita variabile

In generale, comunque, viene usato il protocollo UDP, poichè offre maggiore efficienza in termini di traffico sulla rete.

39

VERSIONI DI TINC

Le versioni più importanti sono:

Tinc 1.0pre3 set 2000: introduce chiavi RSA, usa OpenSSL per la cifratura, supporta le reti virtuali multiple

Tinc 1.0pre5 feb 2002: aumento di sicurezza grazie all'aggiunta del numero progressivo e del codice di autenticazione messaggio ai pacchetti (MAC), contiene supporti preliminari per i pacchetti IPv6

Message Authentication Code. E' un algoritmo che prende in input un messaggio e una chiave segreta e computa un valore che servirà per autenticare il messaggio.

Tinc 1.0pre6 mar 2002: nuovo supporto per i pacchetti IPv6, introdotta la libreria zlib per una compressione opzionale

Internet Protocol (versione 6). Il termine può essere riferito al pacchetto al protocollo oppure all'indirizzo. Indicando rispettivamente: il formato, il checksum dell'header e l'indirizzo univoco assegnato all'host

40

VERSIONI DI TINC

Tinc 1.0 ago 2003: ripulisce il codice sorgente, raddoppia il rendimento e riduce l'inattività, supporto aggiuntivo per la compressione dovuto all'aggiunta della libreria lzo, supporto aggiuntivo per Windows 2000 e XP

Tinc 1.0.1 ago 2003: aggiorna la documentazione e ripara gli errori di compilazione dovuti a errori di scrittura

Tinc 1.0.2 set 2003: usa RSA, aggiunge preventivamente la sequenza di numeri e il MAC ai pacchetti, supporta le reti virtuali multiple, usa **SHA1** di default come algoritmo per la cifratura, usa **SHA1** come funzione hash e inoltre supporta il routing per i pacchetti con indirizzo di tipo IPv6, ripara un bug in meta-data che provocava l'aborto del daemon tinc

41

VERSIONI DI TINC

Tinc 1.0 ago 2003: ripulisce il codice sorgente, raddoppia il rendimento e riduce l'inattività, supporto aggiuntivo per la compressione dovuto all'aggiunta della libreria lzo, supporto aggiuntivo per Windows 2000 e XP

Tinc 1.0.1 ago 2003: aggiorna la documentazione e ripara gli errori di compilazione dovuti a errori di scrittura

cifrano a blocchi sviluppato da Bruce Schneier, opera su blocchi di 64 bit con chiavi di lunghezza variabile fino a 448 bits. Questo algoritmo utilizza varie tecniche tra le quali la rete Feistel, le S-box dipendenti da chiavi e funzioni non invertibili che lo rendono, forse, l'algoritmo più sicuro attualmente disponibile.

Secure Hash Algorithm, (1 indica lo shift nell'operazione dei blocchi)

Tinc 1.0.2 set 2003: usa RSA, aggiunge preventivamente la sequenza di numeri e il MAC ai pacchetti, supporta le reti virtuali multiple, usa **SHA1** di default come algoritmo per la cifratura, usa **SHA1** come funzione hash e inoltre supporta il routing per i pacchetti con indirizzo di tipo IPv6, ripara un bug in meta-data che provocava l'aborto del daemon tinc

42

SOFTWARE UTILIZZATO

Tinc 1.02 Ultima versione del software Tinc per implementare una VPN, è un software **GNU GPL** (General Public License) ossia è gratuito e si può ridistribuirlo e/o modificarlo, rispettando sempre i parametri della licenza, a proprio piacimento.

OpenSSL Usato per tutte le operazioni di cifratura

zlib Libreria utilizzata per la compressione dei pacchetti

lzo Libreria utilizzata per un'ulteriore compressione

43

SOFTWARE UTILIZZATO

Tinc 1.02 Ultima versione del software Tinc per implementare una VPN, è un software **GNU GPL** (General Public License) ossia è gratuito e si può ridistribuirlo e/o modificarlo, rispettando sempre i parametri della licenza, a proprio piacimento.

Si intende di software di sistema compatibile Unix sviluppato dalla *Free Software Foundation*. L'intento della fondazione è quello di produrre software *freeware*, ossia gratuiti.

General Public License. È la licenza con la quale viene rilasciato un software di tipo GNU. Tale licenza permette di modificare o ridistribuire tale software, ma non a scopo di lucro.

44

PIATTAFORME COMPATIBILI

Tinc può girare su diverse piattaforme, quali:

- Linux: per la quale è stata scritta
- FreeBSD
- OpenBSD
- NetBSD
- Solaris8 (SunOS 5.8)
- Darwin (MacOS/X)
- Windows

45

PREPARAZIONE DEL SISTEMA

- OpenSSL
- Zlib
- Lzo
- Configurazione kernel

PREPARAZIONE DEL SISTEMA

Prima di configurare e costruire Tinc sarà necessario installare le seguenti librerie:

1. **OpenSSL**
2. **zlib**
3. **lzo**

Tutte queste librerie rispettano i termini di GNU GPL

47

OpenSSL

Reperibile all'url <http://www.openssl.org/> verrà utilizzata per tutte le funzioni di crittografia

Per installare questa libreria basterà decomprimere il pacchetto scaricato e lanciare dalla stessa directory i comandi:

- `./configure`
- `make`
- `make check`
- `make test`
- `make install`

48

zlib

Reperibile all'url <http://www.gzip.org/zlib/> serve per la compressione opzionali di pacchetti UDP

Se zlib non è installato il ./configure di Tinc darà errore

Per installare questa libreria si dovrà decomprimere il pacchetto e lanciare dalla stessa directory i seguenti comandi:

- `./config`
- `make`
- `make check`
- `make install`

49

lzo

Reperibile all'url <http://www.oberhumer.com/opensource/lzo/> offre altre forme di compressione a Tinc

Se non è installata avremo errore alla configurazione di Tinc

Come le altre due librerie anche questa va decompressa e installata con i comandi:

- `./configure`
- `make`
- `make test`
- `make install`

50

CONFIGURAZIONE DEL KERNEL

Per far girare Tinc sotto Linux si dovrà configurare il kernel a seconda del distributore e della versione del sistema operativo

Se si usa Linux con una versione del kernel precedente alla 2.4.0 si dovrà creare il file di device ethertap, con i seguenti comandi:

```
mknod -m 600 /dev/tap0 c 36 16  
mknod -m 600 /dev/tap1 c 36 17  
...  
mknod -m 600 /dev/tapN c 36 N+16
```

Fino a un massimo di sedici file di ethertap

51

CONFIGURAZIONE DEL KERNEL

Se si usano i driver universali tun/tap si dovrà creare (a meno che non esista già) il seguente device file:

```
mknod -m 600 /dev/tun c 10 200
```

Se si usa Linux con una versione del kernel 2.4.0 o successiva, il dispositivo tun/tap sarà probabilmente generato automaticamente come /dev/net/tun

N.B. Diversamente dalla device ethertap non si avrà bisogno di più file device per far girare più Tinc

52

TUN/TAP

I dati che viaggiano sulla VPN sono prima letti dalla device della rete virtuale e poi spediti all'interfaccia della rete virtuale. Ci sono due tipi di "virtual network device"

1. Tun: è una device di tipo point-to-point (punto a punto) che supporta solo IPv4 e/o IPv6
2. Tap: è una device di tipo **Ethernet** e quindi ne supporta gli stessi formati

Se la device è di tipo Tap l'indirizzo MAC dovrà coincidere con l'interfaccia della rete virtuale

53

TUN/TAP

I dati che viaggiano sulla VPN sono prima letti dalla device della rete virtuale e poi spediti all'interfaccia della rete virtuale. Ci sono due tipi di "virtual network device"

1. Tun: è una device di tipo point-to-point (punto a punto) che supporta solo IPv4 e/o IPv6
2. Tap: è una device di tipo **Ethernet** e quindi ne supporta gli stessi formati

Sistema di reti sviluppato dalla INTEL, XEROX e DIGITAL nel 1979. È una rete a diffusione di tipo bus con un controllo operativo decentralizzato. Utilizza cavi UTP o coassiali per una comunicazione efficiente e veloce tra host che non devono superare la distanza di 1.5 km. In una rete di tipo Ethernet si possono trasmettere 10 Mbit al sec.

54

INSTALLAZIONE

- Installazione su architetture multiple
- Opzioni dello script *configure*
- Opzioni di controllo
- Valori condivisi
- File di sistema necessari

INSTALLAZIONE

Per la nostra implementazione abbiamo usato SuSE Linux 7.2 con la versione kernel 2.4.4

Dopo aver estratto tutti i file del pacchetto Tinc-1.0.2 in una directory a piacere (che chiameremo tinc) ad un path altrettanto simbolico (che per comodità chiameremo "sysconfdir"), bisognerà eseguire i seguenti comandi:

<i>./configure</i>	se usi csh o una vecchia versione di sistemV dovrai scrivere sh./configure
<i>make</i>	compila il pacchetto
<i>make check</i>	opzionale, testa il pacchetto stesso
<i>make install</i>	installa il programma, alcuni file dati e la documentazione

56

INSTALLAZIONE COMPILAZIONE PER ARCHITETTURE MULTIPLE

Si può compilare il pacchetto per diversi tipi di architetture, creando i file oggetto per ognuna delle architetture nelle proprie directory

Per fare ciò sarà necessario disporre di una versione di make che supporti la variabile VPATH, come ad esempio il make distribuito da GNU

Possiamo generare i file oggetto e gli eseguibili relativi ad ogni architettura ad un path specificato utilizzando VPATH. Metteremo il codice sorgente in ".." per far sì che *configure* lo trovi automaticamente

57

INSTALLAZIONE OPZIONI DELLO SCRIPT CONFIGURE

Per default *make install* installerà i vari file in "sysconfdir"/tinc/bin, "sysconfdir"/tinc/man etc; se si volesse specificare una destinazione basterà dare allo script *configure* l'opzione:

--prefix=PATH

Per specificare il luogo in cui *configure* dovrà cercare i file per l'installazione di programmi e librerie relative ad architetture specifiche o indipendenti, si dovrà usare l'opzione:

--exec-prefix=PATH

58

OPZIONI DELLO SCRIPT CONFIGURE/2

In generale, per vedere tutte le opzioni bisogna dare il comando:

--help

Per installare programmi extra, prima o dopo la configurazione si deve usare la seguente opzione:

--program-prefix=PREFIX

--program-suffix=SUFFIX

59

OPZIONI DELLO SCRIPT CONFIGURE/3

In alcuni pacchetti bisogna fare attenzione alle opzioni:

--enable-FEATURE

dove FEATURE indica una parte opzionale del pacchetto

--with-PACKAGE

dove PACKAGE è una parte che serve al pacchetto originale per poter essere compilato.

60

ANCORA SU 'CONFIGURE'

Per alcuni *configure* bisogna specificare il tipo di host sui quali Tinc girerà

Se *./configure* stampa un messaggio in cui dice che non può essere ospitato su tale tipo di host, bisogna usare l'opzione:

`--host=TYPE`

dove *TYPE* può essere sia un nome corto come 'sun4' o un nome standard composto di tre campi: CPU-COMPANY-SYSTEM

Per sapere quali valori dobbiamo assegnare a *TYPE* bisogna andare a leggere il file *config.sub*

61

OPZIONI DI CONTROLLO

Per controllare l'operato dello script *configure* si possono usare le seguenti opzioni:

`--cache-file=FILE`

Per salvare i risultati dei test in *FILE* invece che in *./config.cache*, utile per fare il debug in quanto basterà settare *FILE* con */dev/null* per disabilitare il caching

`--quiet` `--silent` `-q`

Non fanno stampare i messaggi di ricerca file di *configure*, ridirigendo il normale output in */dev/null*

62

OPZIONI DI CONTROLLO/2

`--srcdir=DIR`

Ricerca il pacchetto del codice sorgente nella directory *DIR*, anche se *configure* è in grado di determinarla automaticamente

`--version`

Stampa la versione di autoconf usato per la generazione dello script *configure*

63

FILE DI SISTEMA NECESSARI

Prima di passare alla configurazione di tinc abbiamo bisogno di aggiornare alcuni file di sistema:

`/etc/networks`

A questo file si potrà aggiungere una linea di comando in cui si indicherà il nome simbolico della VPN (Es. mynet 10.0.0.0)

`/etc/service`

A cui si potrà aggiungere una linea di comando per indicare il numero di porta su cui girerà la VPN

Per default il numero di porta usata è 655

64

CONFIGURAZIONE

- Configurazione per reti multiple
- File di configurazione
- Script di configure
- Procedura di configurazione
- Esempio i configurazione

CONFIGURAZIONE

Una volta deciso chiaramente come dovrà essere la VPN si potrà procedere con:

1. Editing dei file di configurazione (*tinc.conf* e i file di configurazione dell'host)
2. Generazione delle chiavi
3. Distribuzione dei file di configurazione ai vari hosts della VPN

66

CONFIGURAZIONE RETI MULTIPLE

Se si vuol far girare più Tinc su un solo computer, ad esempio un computer che fa parte di più VPN, si potrà assegnare un netname diverso alle varie VPN con il comando:

tincd -n netname

netname è il nome simbolico assegnato alla VPN e verrà creata una nuova directory chiamata "sysconfdir"/tinc/netname

N.B. anche se non è strettamente necessario che si dia un nome alla propria VPN è altamente consigliato poiché in tale modo è più facile interagire con essa

67

FILE DI CONFIGURAZIONE

Il file di configurazione principale è **tinc.conf** che si trova in "sysconfdir"/tinc/ oppure in "sysconfdir"/tinc/netname/ se si è usata l'opzione -n

Gli altri file di configurazione si trovano in "sysconfdir"/tinc/hosts/ oppure in "sysconfdir"/tinc/netname/host/ sempre se si usa l'opzione -n

Uno è il file di configurazione dell'host stesso e gli altri sono i file relativi agli altri host della VPN

68

PRINCIPALI VARIABILI DI CONFIGURAZIONE

AddressFamily = <ipv4|ipv6|opzionale> (qualsiasi valore)

Questa opzione interessa la famiglia di indirizzi da usare per le connessioni in ascolto e per i socket uscenti

BindToAddress = <indirizzo> [sperimentale]

Se il computer ha più di un indirizzo IPv4 o IPv6, Tinc per default ascolterà tutte le connessioni in arrivo

(*)=valori di default [*]=commenti

69

PRINCIPALI VARIABILI DI CONFIGURAZIONE/2

Device = <device> ('/dev/tap0', '/dev/net/tun' o altro per altre piattaforme)

Specifica la device da usare per la rete virtuale. Tinc rileverà automaticamente di che tipo di dispositivo si tratta

Hostnames = <si o no> (no)

Questa opzione controlla o meno che gli indirizzi IP (sia quello reale che quello della VPN) siano disponibili tramite il **DNS**

Domain Name System, sistema di denominazione del dominio. Un server dedicato con un database effettua il mapping del nome dell'host in un indirizzo IP, in altri termini lo sostituisce con il numero IP.

(*)=valori di default [*]=commenti

70

PRINCIPALI VARIABILI DI CONFIGURAZIONE/3

Interface = <interfaccia>

Definisce il nome dell'interfaccia che corrisponde alla device legata alla rete virtuale (es. eth0 o PPP0)

KeyExpire = <secondi> (3600)

Controlla il tempo di vita delle chiavi usate per la cifratura dei dati
E' pratica comune cambiare tali chiavi a intervalli regolari per rendere ancora più difficili attacchi di tipo **crack**

Programma che tenta di indovinare la password di un utente tramite ripetuti tentativi di invio di stringhe casuali, oppure tratta da un dizionario preparato dal mal intenzionato hacker, o meglio cracker

(*)=valori di default [*]=commenti

71

PRINCIPALI VARIABILI DI CONFIGURAZIONE/4

Name = <nome> [richiesto]

E' il nome simbolico della connessione

PingTimeout = <secondi> (60)

E' il numero di secondi dopo i quali Tinc interroga l'altra estremità del daemon per vedere se è ancora attivo

(*)=valori di default [*]=commenti

72

PRINCIPALI VARIABILI DI CONFIGURAZIONE/5

PriorityInheritance = <si/no> [no] [sperimentale]

Quando questa opzione è abilitata il valore del campo **TOS** dei pacchetti di tipo IPv4 sarà ereditato dai pacchetti di tipo UDP che saranno spediti

PrivateKey = <chiave> [obsoleta]

Questa è la chiave privata RSA per Tinc. Tuttavia, per ragioni di sicurezza è meglio memorizzare le chiavi private in altri file. Ciò previene attacchi di tipo **caevdropping** al momento dell'editing del file di configurazione

(*)=valori di default [*]=commenti

73

PRINCIPALI VARIABILI DI CONFIGURAZIONE/5

PriorityInheritance = <si/no> [no] [sperimentale]

Quando questa opzione è abilitata il valore del campo **TOS** dei pacchetti di tipo IPv4 sarà ereditato dai pacchetti di tipo UDP che saranno spediti

PrivateKey = <chiave> [obsoleta]

Questa è la chiave privata RSA per Tinc. Tuttavia, per ragioni di sicurezza è meglio memorizzare le chiavi private in altri file. Ciò previene attacchi di tipo **caevdropping** al momento dell'editing del file di configurazione

(*)=valori di default [*]=commenti

74

PRINCIPALI VARIABILI DI CONFIGURAZIONE/6

PrivateKeyFile=<path>("sysconfdir"/tinc/netname/rsa_key.priv)

Questo è il percorso completo della chiave privata RSA

N.B. Ci deve essere esattamente uno tra PrivateKey e PrivateKeyFile specificato nel file di configurazione

(*)=valori di default [*]=commenti

75

PRINCIPALI VARIABILI DI CONFIGURAZIONE/7

BindToInterface = <interfaccia> [sperimentale]

Se sul computer è presente più di una interfaccia di rete, Tinc per default ascolterà tutte le connessioni in arrivo

ConnectTo = <nome>

Con questa opzione è possibile specificare qual è l'altro Tinc a cui connettersi allo start up

Possono essere specificate multiple ConnectTo

(*)=valori di default [*]=commenti

76

ConnectTo IN DETTAGLIO

Se ci sono una o più "ConnectTo" nel file di configurazione il daemon tinc stimerà i valori dei puntatori ad altri tinc daemon e cercherà di connettersi

Qualunque sia il risultato di questo tentativo di connessione il daemon tinc resterà comunque in ascolto delle connessioni provenienti da altri host

Se il valore di ConnectTo è specificato, ma comunque non si ricevono risposte, il daemon tinc riproverà la connessione. Quindi il daemon è partito rimarrà in funzionamento finchè chi lo ha lanciato non lo fermerà esplicitamente

77

PRINCIPALI VARIABILI DI CONFIGURAZIONE/8

Mode = <router|switch|hub> (router)

Questa opzione seleziona la rotta che il pacchetto dovrà seguire per raggiungere gli altri tinc daemon

router: le variabili di sottorete saranno usate per formare una tabella di router. E' supportata la modalità unicast e il routing con indirizzi di tipo IPv4 e IPv6

switch: gli indirizzi MAC dei pacchetti saranno usati per creare dinamicamente una tabella di routing. E' supportato l'invio dei pacchetti in modalità unicast, multicast, broadcast e ogni tipo di protocollo supportato da una scheda ethernet

hub: simile allo switch, ma in questo caso è preferibile spedire sempre i pacchetti in modalità broadcast poiché non c'è nessuna gestione della tavola di routing

(*)=valori di default [*]=commenti

78

SETTAGGI DELLA VARIABILE MODE

Se *Mode* è settata con router (default) **ARP** (Address Resolution Protocol) non lavorerà, ma il daemon tinc lascerà il compito di risoluzione dell'indirizzo al ricevitore del pacchetto, che controllerà se il MAC address coincide con l'interfaccia della rete virtuale

Se *Mode* è settata in modalità *hub* o *switch* Tinc userà la modalità broadcast per spedire i pacchetti e ne dedurrà l'indirizzo dal MAC address

Ogni interfaccia dovrà avere un unico MAC address ed è per tale motivo che switch e hub non sono supportati da tutte le piattaforme

79

SETTAGGI DELLA VARIABILE MODE

Se *Mode* è settata con router (default) **ARP** (Address Resolution Protocol) non lavorerà, ma il daemon tinc lascerà il compito di risoluzione dell'indirizzo al ricevitore del pacchetto, che controllerà se il MAC address coincide con l'interfaccia della rete virtuale

Address Resolution Protocol, protocollo per la risoluzione degli indirizzi. E' un sistema del tipo "domanda - risposta". Dom. Di chi è l'indirizzo IP? Resp. E' di tale indirizzo fisico.

80

VARIABILI DI CONFIGURAZIONE DELL'HOST

Address = <indirizzo IP| hostname> [raccomandata]

Questa variabile è richiesta soltanto se ci si vuole collegare in maniera diretta ad un determinato host.

Cipher = <cifratura> (blowfish)

Tale variabile serve per specificare il tipo di cifratura simmetrica usata per criptare i pacchetti UDP

Per default la cifratura avviene tramite l'algoritmo blowfish in modalità **CBC**
none=nessuna cifratura

(*)=valori di default [*]=commenti

81

VARIABILI DI CONFIGURAZIONE DELL'HOST

Address = <indirizzo IP| hostname> [raccomandata]

Questa variabile è richiesta soltanto se ci si vuole collegare in maniera diretta ad un determinato host.

Cipher = <cifratura> (blowfish)

Tale variabile serve per specificare il tipo di cifratura simmetrica usata per criptare i pacchetti UDP

Per default la cifratura avviene tramite l'algoritmo blowfish in modalità **CBC**
none=nessuna cifratura

Cipher Block Chaining. E' una modalità di funzionamento di un algoritmo di cifratura. Il messaggio viene processato in blocchi e il risultato è dato dalla concatenazione delle cifrature intermedie

(*)=valori di default [*]=commenti

82

VARIABILI DI CONFIGURAZIONE DELL'HOST/2

Compression = <livello> (0)

Questa operazione setta il livello di compressione usato per i pacchetti UDP

0=nessuna compressione
1=compressione veloce con zlib
2-9=miglior compressione con zlib
10=compressione veloce di lzo
11=miglior compressione di lzo

Digest = <hash> (sha1)

Questa variabile setta la funzione hash da lanciare per l'autenticazione dei pacchetti UDP
none=nessuna identificazione

(*)=valori di default [*]=commenti

83

VARIABILI DI CONFIGURAZIONE DELL'HOST/3

IndirectData = <sjino> (no)

Questa opzione specifica se altri Tinc daemon possono fare una connessione diretta con il nostro Tinc daemon oltre a quella specificata in ConnectTo

MACLength = <bytes> (4)

Specifica la lunghezza dell'indirizzo MAC usato per autenticare i pacchetti

Può andare da 0 alla lunghezza dell'output della funzione hash usata per l'autenticazione

(*)=valori di default [*]=commenti

84

VARIABILI DI CONFIGURAZIONE DELL'HOST/4

Port = <porta> (655)

Questo è il numero di porta sulla quale il daemon è in ascolto. Si può usare un numero decimale o un nome simbolico

PublicKey = <chiave> [obsoleto]

Questa è la chiave pubblica RSA per questo host

PublicKeyFile = <path> [obsoleto]

Questo è il path completo dove trovare la chiave pubblica RSA

(*)=valori di default [*]=commenti

85

VARIABILI DI CONFIGURAZIONE DELL'HOST/5

Subnet = <indirizzo[lunghezza prefisso]>

Specifica la sottorete di cui Tinc si servirà. Tinc proverà a guardare, tramite una appropriata ricerca nella sottorete, se ci sono altri daemon per spedire pacchetti.

TCPonly = <si|no> (no) [sperimentale]

Se questa variabile è settata su si allora tutti i pacchetti verranno spediti tramite una connessione TCP invece che con una connessione UDP

(*)=valori di default [*]=commenti

86

SCRIPT DI CONFIGURE

Sono piccole porzioni di codice compilate ed eseguite da *configure*

/etc/tinc/netname/tinc-up

Sarà eseguito dopo che il tinc daemon sarà partito e connesso con la device della rete virtuale

E' usato per settare l'interfaccia della corrispondente rete

/etc/tinc/netname/tinc-down

Questo script parte dopo che il tinc daemon termina e serve per rilasciare l'interfaccia in modo da poter essere riutilizzata

87

SCRIPT DI CONFIGURE/2

/etc/tinc/netname/hosts/host-up

Questo script parte quando l'host, di cui abbiamo specificato il nome in ConnectTo diventa raggiungibile

Serve per aggiornare le variabili d'ambiente: NODE, REMOTEADDRESS, REMOTEPORT

/etc/tinc/netname/hosts/host-down

Questo script parte quando l'host, di cui abbiamo specificato il nome in ConnectTo diventa irraggiungibile

Aggiorna le stesse variabili ambiente di host-up ma, naturalmente, in modo inverso

88

SCRIPT DI CONFIGURE VARIABILI D'AMBIENTE

Gli script visti partono automaticamente e fanno uso di alcune variabili d'ambiente quali:

NETNAME, se netname è specificata questa variabile lo contiene

NAME, contiene il nome del nostro tinc daemon

DEVICE, contiene il nome della device di rete che tinc usa

INTERFACE, contiene il nome dell'interfaccia della rete virtuale che tinc usa

89

SCRIPT DI CONFIGURE VARIABILI D'AMBIENTE/2

NODE, quando un host diventa (in)raggiungibile, questo insieme viene aggiornato in base al suo nome

REMOTEADDRESS, quando un host diventa (in)raggiungibile, questo insieme viene aggiornato in base al suo indirizzo

REMOTEPORT, quando un host diventa (in)raggiungibile, questo insieme viene aggiornato in base al numero di porta che sta usando per comunicare con gli altri tinc daemon

90

PROCEDURA DI CONFIGURAZIONE

Passo1) Creazione del principale file di configurazione : *tinc.conf*

Le linee obbligatorie per tale file sono:

Name= nome simbolico

Device = /dev/net/tun valore usato nel nostro esempio

Una volta deciso a quale altro daemon connettersi, si può aggiungere la linea:

ConnectTo= nome simbolico dell'altro daemon

91

PROCEDURA DI CONFIGURAZIONE

Passo2) Creazione dell'host configuration file

Tale file dovrà chiamarsi "nome simbolico", ossia il valore che hai dato alla variabile **Name** in *tinc.conf* e dovrà contenere necessariamente le seguenti linee:

Address = tuo.reale.hostname.org

Subnet = 192.156.1.0/24

Address può anche usare l'indirizzo IP invece dell'host name.

Subnet specifica il range di indirizzi che sono locali solo per la tua parte di VPN

92

PROCEDURA DI CONFIGURAZIONE

Passo3) Generazione delle chiavi

Una volta creati i file di configurazione, si possono facilmente generare la chiave pubblica e la chiave privata per la cifratura e decifratura dei dati spediti sulla VPN

Per far ciò basterà lanciare il seguente comando:

tincd -K oppure **tincd -n netname -K**

93

PROCEDURA DI CONFIGURAZIONE

Lanciato il comando Tinc genererà le due chiavi e chiederà in che file memorizzarle. Si può immettere un path o premere INVIO per accettare le destinazioni di default che sono:

"*sysconfdir*"/*tinc/netname/hosts/nonehost* per la chiave pubblica e
"*sysconfdir*"/*tinc/netname/rsa_key.priv* per la chiave privata

94

PROCEDURA DI CONFIGURAZIONE

Passo 4) Configurazione dell'interfaccia di rete

Dopo aver scelto la device di rete virtuale a cui legare il daemon tinc bisogna decidere l'indirizzo IP da associargli e quale maschera di rete dovrà usare

Tinc aprirà una device per la rete virtuale (/dev/net/tun, /dev/tap0 o simili) con le quali potrà anche creare una interfaccia di rete chiamata per esempio tun0 o tap0

95

PROCEDURA DI CONFIGURAZIONE

Se si usa Linux e i driver universali per tun/tap, l'interfaccia di rete avrà lo stesso nome della rete. Si può configurare l'interfaccia di rete aggiungendo alcune linee nello script tinc-up, ad esempio:

```
#!/bin/sh ifconfig $INTERFACE 192.168.1.1 netmask 255.255.0.0  
ifconfig $INTERFACE 192.168.1.1 netmask 255.0.0.0
```

96

ESEMPIO DI CONFIGURAZIONE

Immaginiamo la seguente situazione. Nella rete VPN chiamata "my net" c'è: UfficioA vuole connettersi a tre uffici B,C,D usando Internet. Tutti e quattro gli uffici sono connessi 7 giorni su 7, 24 ore su 24

A funziona come il centro della rete. B e C si vogliono connettere ad A, mentre D vuole connettersi a C. Ad ogni ufficio dovrà essere assegnato il proprio indirizzo IP di rete, 10.x.0.0

97

ESEMPIO DI CONFIGURAZIONE

```
A: net 10.1.0.0 mask 255.255.0.0 gateway 10.1.54.1 Internet IP 1.2.3.4
B: net 10.2.0.0 mask 255.255.0.0 gateway 10.2.1.12 Internet IP 2.3.4.5
C: net 10.3.0.0 mask 255.255.0.0 gateway 10.3.69.254 Internet IP 3.4.5.6
D: net 10.4.0.0 mask 255.255.0.0 gateway 10.4.3.32 Internet IP 4.5.6.7
```

Il gateway rappresenta l'indirizzo IP della VPN e Internet IP è l'indirizzo IP del firewall (di cui si ha bisogno per far girare *tincd*)

NB Si deve anche specificare il numero di porta attraverso la quale avviene la spedizione dei pacchetti UDP e TCP (default 655)

98

ESEMPIO DI CONFIGURAZIONE

In questo esempio assumiamo che:

1. eth0 è l'interfaccia che porta all'interno (fisicamente) della LAN di un determinato ufficio, anche se tale interfaccia potrebbe essere la stessa utilizzata per affacciarsi su Internet

Local Area Network, rete privata limitata a 10 km per la condivisione di risorse (HD condivisi, stampanti, modem e altre periferiche). Si distingue dagli altri tipi di rete per: grandezza, tecnologia di trasmissione, topologia: a bus (Ethernet), ad anello (Token ring), a stella, od altro.

2. mynet è il nome simbolico della VPN che tutti i rami usano

99

ESEMPIO DI CONFIGURAZIONE CONFIGURAZIONE UFFICIO 'A'

```
Nel file "sysconfdir"/tinc/mynet/tinc.conf:
Name=UfficioA
PrivateKeyFile= "sysconfdir"/tinc/mynet/rsa_key.priv
Device=/dev/tap0
```

Mentre in tutti gli altri host(B,C,D) nel file "sysconfdir"/tinc/mynet/hosts/UfficioA conterranno le seguenti linee:

```
Subnet=10.1.0.0/16
Address=1.2.3.4
----BEGIN RSA PUBLIC KEY----
...
----END RSA PUBLIC KEY----
```

100

ESEMPIO DI CONFIGURAZIONE CONFIGURAZIONE UFFICIO 'A'

```
In più, in "sysconfdir"/tinc/mynet/tinc-up:
#interfaccia interna reale della VPN (gateway)
#ifconfig eth0 10.1.54.1 netmask 255.255.0.0
ifconfig SINTERFACE 10.1.54.1 netmask 255.0.0.0
```

L'indirizzo IP di eth0 e tap0 sono gli stessi. Ciò è possibile perché si hanno differenti netmask

101

ESEMPIO DI CONFIGURAZIONE CONFIGURAZIONE UFFICIO 'B'

```
Nel file "sysconfdir"/tinc/mynet/tinc.conf:
Name=UfficioB
ConnectTo=UfficioA
PrivateKeyFile= "sysconfdir"/tinc/mynet/rsa_key.priv
```

Mentre in tutti gli altri host(A,C,D) nel file "sysconfdir"/tinc/mynet/hosts/UfficioB sono specificate le seguenti linee:

```
Subnet=10.2.0.0/16
Address=2.3.4.5
----BEGIN RSA PUBLIC KEY----
...
----END RSA PUBLIC KEY----
```

102

ESEMPIO DI CONFIGURAZIONE CONFIGURAZIONE UFFICIO 'B'

```
Inoltre, in "sysconfdir"/tinc/mynet/tinc-up:
#interfaccia interna reale della VPN (gateway)
#ifconfig eth0 10.2.43.8 netmask 255.255.0.0
ifconfig $INTERFACE 10.2.1.12 netmask 255.0.0.0
```

L'indirizzo IP di eth0 non è lo stesso di tap0. In più c'è l'inizializzazione di ConnectTo

102

ESEMPIO DI CONFIGURAZIONE CONFIGURAZIONE UFFICIO 'C'

```
Nel file "sysconfdir"/tinc/mynet/tinc.conf:
Name=UfficioC
ConnectTo=UfficioA
PrivateKeyFile= "sysconfdir"/tinc/mynet/rsa_key.priv
Device=/dev/tap1
```

Poiché C ha un altro daemon che gira sulla porta 655 deve riservare un'altra porta a Tinc. Supposto che tale porta sia la numero 1000, gli altri host (A,B,D) nel file "sysconfdir"/tinc/mynet/hosts/UfficioC conterranno le seguenti linee:

```
Subnet=10.3.0.0/16
Address=3.4.5.6
Port=1000
----BEGIN RSA PUBLIC KEY----
...
----END RSA PUBLIC KEY----
```

104

ESEMPIO DI CONFIGURAZIONE CONFIGURAZIONE UFFICIO 'C'

```
Inoltre, in "sysconfdir"/tinc/mynet/tinc-up:
#interfaccia interna reale della VPN (gateway)
#ifconfig eth0 10.3.69.254 netmask 255.255.0.0
ifconfig $INTERFACE 10.3.69.254 netmask 255.0.0.0
```

105

ESEMPIO DI CONFIGURAZIONE CONFIGURAZIONE UFFICIO 'D'

```
Nel file "sysconfdir"/tinc/mynet/tinc.conf:
Name=UfficioD
ConnectTo=UfficioC
PrivateKeyFile= "sysconfdir"/tinc/mynet/rsa_key.priv
Device=/dev/tap0
```

Mentre in tutti gli altri host (A,B,C) nel file "sysconfdir"/tinc/mynet/hosts/UfficioD conterranno le seguenti linee:

```
Subnet=10.4.0.0/16
Address=4.5.6.7
----BEGIN RSA PUBLIC KEY----
...
----END RSA PUBLIC KEY----
```

106

ESEMPIO DI CONFIGURAZIONE CONFIGURAZIONE UFFICIO 'D'

```
In "sysconfdir"/tinc/mynet/tinc-up:
#interfaccia interna reale della VPN (gateway)
#ifconfig eth0 10.4.3.32 netmask 255.255.0.0
ifconfig $INTERFACE 10.4.3.32 netmask 255.0.0.0
```

NB Poiché D dovrà connettersi a C, e poiché quest'ultimo usa la porta 1000 per il daemon Tinc il ramo D dovrà andare a leggere tale numero nel file di configurazione relativo a C

107

RUNTIME DI TINC

- Opzioni di runtime
- Debug di TINC
- Messaggi di errori comuni
- Invio di pacchetti
- Daemon message

COME FAR GIRARE TINC

Il primo comando da dare per far girare Tinc è:

```
tincd oppure tincd -n netname
```

Tinc si staccherà dal terminale e continuerà a girare in background

Per controllare il corretto funzionamento, si può lanciare il *debug* per Tinc e vedere nel file di *syslog* (che per default è: "sysconfdir"/tinc/var/log/tinc.netnamelog) se e dove ci sono errori

109

OPZIONI DI RUNTIME

Oltre a eseguire i settaggi e l'operazioni contenute nei file di configurazione, Tinc accetta alcune opzioni da linea di comando:

```
-c, --config=path
```

Legge i file di configurazione al path immesso. Per default è "sysconfdir"/tinc/*netname*

```
-D, --no-detach
```

Non permette a Tinc di fare delle fork e inabiliterà il meccanismo automatico di riavvio dovuto a errori fatali

110

OPZIONI DI RUNTIME/2

```
-d, --debug=level
```

Regola il livello del debug

Più elevato sarà più informazioni saranno annotate nel file di *syslog*

```
-K, --kill[=signal]
```

Tenta di terminare Tinc con SIGTERM o un altro segnale specificato nell'opzione
Usandolo insieme con -n terminerà la rete specificata

111

OPZIONI DI RUNTIME/3

```
-n, --net=netname
```

Configura Tinc per la rete *netname*

```
-K, --generate-keys[=bits]
```

Genera la coppia di chiavi pubblica/privata lunga *bits* (default 1024). Inoltre chiederà dove immagazzinare i file con le chiavi (premendo INVIO si accettano i valori di default)

```
-L, --mlock
```

Limita l'uso della memoria concessa a Tinc solo alla memoria centrale
Ciò impedirà che i dati sensibili come le chiavi private siano scritte in file o partizioni condivise

112

OPZIONI DI RUNTIME/4

```
--logfile[=file]
```

Scriva le informazioni relative al funzionamento di Tinc nel file di *syslog* che può essere specificato in *file*

Se *file* è omissso, per default è:

```
sysconfdir/tinc/var/log/tinc.netnamelog
```

```
--pidfile=file
```

Scriva il PID in *file* anziché in

```
sysconfdir/tinc/var/run/tinc.netnamepid
```

113

OPZIONI DI RUNTIME/5

```
--bypass-security
```

Disabilita la crittografia e l'autenticazione
Utile solo per il debug

```
--help
```

Stampa le varie opzioni di questo runtime e termina

```
--version
```

Stampa la versione ed esce

114

RISOLUZIONE DEI PROBLEMI

Per trovare e risolvere problemi la prima cosa da fare è far partire il Tinc con l'opzione di debug al più alto livello, lanciando il comando:

```
tincd -d5 -D oppure tincd -n netname -d5 -D
```

115

RISOLUZIONE DEI PROBLEMI/2

Se tincd parte, ma non lavora come dovrebbe e il comando precedente non annota alcuni messaggi di errore, si potranno controllare:

- Lo script tinc-up: far sì che tale script abbia i giusti comandi
- Subnet: controllare che nel file di configurazione dell'host Subnet è inizializzata con la porzione di VPN che gli spetta
- Firewall e NAT: controllare che si ha un firewall oppure una NAT, se è così controllare che è permesso il traffico TCP e UDP sulla porta 655

116

RISOLUZIONE DEI PROBLEMI/2

Se tincd parte, ma non lavora come dovrebbe e il comando precedente non annota alcuni messaggi di errore, si potranno controllare:

- Lo script tinc-up: far sì che tale script abbia i giusti comandi
- Subnet: controllare che nel file di configurazione dell'host Subnet è inizializzata con la porzione di VPN che gli spetta
- Firewall e NAT: controllare che si ha un firewall oppure una NAT, se è così controllare che è permesso il traffico TCP e UDP sulla porta 655

Un firewall, letteralmente "muro di fuoco", è, come suggerisce il suo nome, un sistema progettato per impedire accessi non autorizzati a (e da) reti private. Esso può essere realizzato sia via software che via hardware (o anche con una combinazione delle due cose). Il suo utilizzo tipico è quello di impedire agli utenti provenienti da Internet l'accesso non autorizzato ad una Intranet.

Network Address Translator, traduzione di indirizzo di rete. Processo che fa corrispondere un indirizzo in un altro, di solito usato per assegnare un unico indirizzo a tutti gli utenti di una rete LAN.

117

MESSAGGI DI ERRORI COMUNI

Oltre ai messaggi di errori più comuni di seguito troverete le rispettive soluzioni

Alcuni dei seguenti messaggi saranno visibili solo se il livello del debug è abbastanza alto

Could not open /dev/tap0: No such device

- Compilare "Netlink device emulation" nel kernel

Can't write to /dev/net/tun: No such device*

- Compilare "Universal TUN/TUP driver" nel kernel.
- Controllare che la device Tun non sia in /dev/

118

MESSAGGI DI ERRORI COMUNI/2

Network address and prefix length do not match

- Se si usa un solo indirizzo IP, setta la netmask con /32

Error reading RSA key file *rsa_key.priv*: No such file or directory

- Specificare il pathname completo del file che contiene la chiave privata nella variabile `PrivateKeyFile`

119

MESSAGGI DI ERRORI COMUNI/3

Warning: insecure file permissions for RSA private key file *rsa_key.priv*

- Il file della chiave privata può essere letto anche da altri utenti. Usa chmod per correggere i permessi di lettura del file

Creating metsocket failed: Address family not supported

- Per default Tinc crea i socket sia per IPv4 che per IPv6. Poiché alcune piattaforme non supportano entrambi i tipi di socket basterà aggiungere l'inizializzazione di AddressFamily=ipv4 in tinc.conf

120

MESSAGGI DI ERRORI COMUNI/4

Cannot route packet: unknown IPv4 destination 1.2.3.4

- Basterà spedire il pacchetto in broadcast (indirizzo che termina con .255)

Cannot route packet: ARP request for unknown address 1.2.3.4

- Cambiare l'indirizzo della sottorete a cui si sta trasmettendo il traffico

121

MESSAGGI DI ERRORI COMUNI/5

Packet with destination 1.2.3.4 is looping back to us!

- Qualcosa non è configurato in modo corretto.

Generalmente l'errore è che si è data una lunghezza prefissata alla sottorete pari all'interfaccia della rete virtuale

Tale lunghezza dovrebbe essere più grande.

- C'è la possibilità che Subnet=...nel file di configurazione dell'host sia errata. Cambiare la sottorete in modo che sia accettata dalle altre interfacce, o se il problema non si risolve, provare cambiando la lunghezza prefissata con /32

Node foo (1.2.3.4) is not reachable

- Tale nodo non è più connesso, perché il relativo daemon Tinc non sta più girando o perché la connessione ad Internet è stata interrotta

122

MESSAGGI DI ERRORI COMUNI/6

Received UDP packet from unknown source 1.2.3.4 (port 12345)

- Se questo messaggio compare sporadicamente non è preoccupante
 - Altrimenti, significa che un nodo non è più raggiungibile, quindi un NAT sta cambiando l'indirizzi sorgente dei pacchetti UDP
- Per ovviare a ciò aggiungere TCPOOnly = sì nel file di configurazione

Got bad/bogus/unauthorized REQUEST from foo (1.2.3.4 port 12345)

- Un nodo non ha una corretta coppia di chiavi pubblica/privata, poiché un attaccante prova ad accedere alla vostra VPN oppure c'è stato un errore della rete ha causato la corruzione dei dati spediti dal nodo
- Generare e distribuire nuove coppie

123

INVIO DEI PACCHETTI

I pacchetti possono essere spediti solo se le due parti conoscono le chiavi per la cifratura e la connessione è attiva.

Schema di un pacchetto:



124

MESSAGGIO DI RICHIESTA CHIAVE

E' uno dei messaggi più importanti:



Le chiavi usate per la cifratura non sono mai spedite direttamente, poiché questo causerebbe un traffico eccessivo, ma può succedere che un daemon ne abbia bisogno

125

MESSAGGIO DI RISPOSTA A REQ-KEY



126

MESSAGGIO DI CAMBIO CHIAVI



127

DAEMON MESSAGE

Inoltre, dopo che sarà stata effettuata l'autenticazione tra due daemon, quest'ultimi potranno iniziare a scambiarsi informazioni relative alle sottoreti di cui hanno conoscenza, i messaggi per fare ciò sono:



ADD_EDGE informa gli altri daemon che esiste il collegamento tra node1 e node2

128

DAEMON MESSAGE

Inoltre, dopo che sarà stata effettuata l'autenticazione tra due daemon, quest'ultimi potranno iniziare a scambiarsi informazioni relative alle sottoreti di cui hanno conoscenza, i messaggi per fare ciò sono:



ADD_SUBNET informa gli altri daemon del proprietario della sottorete specificata

129

DAEMON MESSAGE



DEL_EDGE informa gli altri daemon che il collegamento tra node1 e node2 si è rotto. Gli altri daemon si calcoleranno un nuovo itinerario, altrimenti contrassegneranno come irraggiungibili uno dei due nodi

130

DAEMON MESSAGE



DEL_SUBNET informa che la sottorete specificata è stata eliminata

131

DAEMON MESSAGE



Tali messaggi vengono inviati a intervalli regolari (ad eccezione di traffico intenso), con l'aggiunta di alcuni bit casuali, per controllare se un daemon è ancora vivo

132

CONCLUSIONI

La crittografia è un "prodotto" difficile sia da realizzare e sia da valutare per quanto riguarda la qualità dello stesso

Infatti non si ha mai la certezza assoluta di essere al sicuro da eventuali attacchi

133

CONCLUSIONI

La VPN, nonostante ciò, offre una protezione sicuramente maggiore rispetto ad una rete pubblica (a meno di eventuali intrusi interni)

In particolare il pacchetto software che abbiamo utilizzato per costruire una VPN, non solo ha offerto prodotti di crittografia eccellenti ma è anche immune a vari tipi di attacchi (know plaintext, meet-in-the-middle, eavesdropping, etc)

134