INFN

ISTITUTO NAZIONALE DI FISICA NUCLEARE

Sezione di Trieste

INFN/TC-04/3
18 Marzo 2004

UNA STRUTTURA DI AUTENTICAZIONE CENTRALIZZATA PER DIVERSI GRUPPI DI SISTEMI LINUX, REALIZZATA CON NIS E IPSEC

Tullio Macorini, Claudio Strizzolo, Lucio Strizzolo

INFN, Sezione di Trieste

Sommario

Il presente documento descrive una struttura di autenticazione centralizzata per sistemi Linux, che consente di gestire l'accesso a diversi insiemi di macchine da parte di gruppi distinti di utenti. La struttura in oggetto è in produzione per il riconoscimento degli utenti su vari sistemi di calcolo della Sezione di Trieste, dislocati in siti geografici differenti.

L'architettura è stata realizzata per mezzo di un unico dominio NIS, impiegando alcune funzionalità particolari offerte dai moduli di autenticazione di Linux, e utilizzando il protocollo IPsec per garantire la sicurezza delle comunicazioni tra i server NIS.

Il documento presenta le caratteristiche principali della struttura ed alcune modalità operative per la sua realizzazione.

Published by ***SIS–Pubblicazioni***
Laboratori Nazionali di Frascati

1 INTRODUZIONE

La Sezione di Trieste è geograficamente dislocata presso tre siti distinti:

- il Dipartimento di Fisica dell'Università;
- l'Area di Ricerca;
- il Dipartimento di Fisica Teorica di Miramare.

Questa suddivisione logistica ha gradualmente portato, nel corso del tempo, ad una replicazione dei sistemi di autenticazione per i vari siti, sia per questioni di affidabilità in caso di problemi di comunicazione tra le sedi, sia per motivi di sicurezza, dal momento che le connessioni di rete tra i siti non sono "private" e sono quindi a rischio di potenziali intercettazioni da parte di terzi.

Inoltre, per motivi storici, alcuni gruppi di utenti ed alcuni servizi particolari, come il mail server, hanno da sempre utilizzato sistemi di autenticazione autonomi.

Tutte le architetture di autenticazione sono state fino ad ora costruite su domini NIS[1] a sé stanti. I limiti di una situazione come questa sono evidenti:

- difficoltà nel tenere traccia degli accessi autorizzati ai diversi sistemi;
- necessità di definire gli account separatamente su ogni dominio NIS, per gli utenti che hanno accesso a diversi domini. Analogamente: necessità di mantenere password separate per ogni dominio da parte degli utenti;
- per ogni dominio NIS ci deve essere un master server e possibilmente almeno uno slave server da utilizzare per la ridondanza in caso di problemi sul master: al crescere del numero di domini NIS indipendenti, anche il numero di nodi configurati come server diventa elevato.

Pertanto si è deciso di realizzare una nuova struttura che consenta una gestione più semplice e razionale da parte del Servizio Calcolo e Reti della Sezione, ed anche maggiore comodità per gli utenti.

La struttura realizzata ha consentito di unificare i sistemi di autenticazione per i sistemi di calcolo centrali basati su Unix per due sedi della Sezione (Dipartimento di Fisica e Area di Ricerca).

Questo documento descrive la struttura realizzata e gli strumenti utilizzati per la messa in opera.

2 LA SITUAZIONE ESISTENTE

La situazione dei server di autenticazione nelle due sedi interessate dalla ristrutturazione, prima dell'intervento, era la seguente:

Dipartimento di Fisica: (due domini indipendenti)

- Dominio 1:

[1] Network Information Service, servizio conosciuto anche con il nome Yellow Pages.

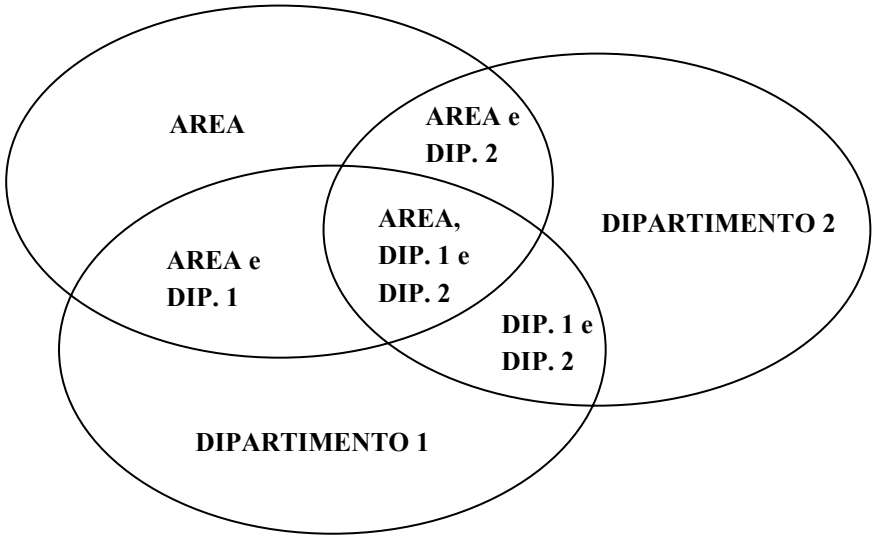
  - Utilizzo: autenticazione per il server centrale del Dipartimento, con utenza INFN e Dip. di Fisica, compresi gli studenti, e per alcuni client.
  - Master server: Alpha station con sistema operativo Tru64.
  - Slave server: nessuno.
  - Client: alcuni PC con Linux.
  - File server: home directory degli utenti condivise via NFS dal nodo facente funzioni di master server.
- Dominio 2:
  - Utilizzo: autenticazione per un gruppo di macchine ad uso esclusivo di un gruppo di ricerca.
  - Master server: Alpha server con sistema operativo Tru64.
  - Slave server: nessuno.
  - Client: alcuni nodi con sistema operativo Tru64, in via di dismissione, ed un numero crescente di PC con Linux.
  - File server: home directory degli utenti condivise via NFS dal nodo facente funzioni di master server.

Area di Ricerca: (un solo dominio)

- Dominio 3:
  - Utilizzo: autenticazione per tutti i sistemi gestiti dal servizio di Calcolo e Reti presso l'Area di Ricerca.
  - Master server: Alpha server con sistema operativo Tru64.
  - Slave server: Alpha station con sistema operativo Tru64.
  - Client: alcuni nodi con Tru64 e circa 25 PC con Linux.
  - File server: home directory degli utenti condivise via NFS da un server NAS (Network Attached Storage). Il NAS allo stato attuale prevede solo NIS quale sistema per l'autenticazione ed il riconoscimento degli utenti.

Data l'obsolescenza di gran parte dei sistemi Tru64 in uso, era già stata pianificata la loro sostituzione con PC con sistema operativo Linux.

Alcuni utenti hanno diritto di accesso solo ad uno dei tre domini NIS; altri, invece, a più di uno:



3 REQUISITI DELLA NUOVA STRUTTURA

- Si vuole realizzare una struttura di autenticazione unica per i sistemi di calcolo centrali con sistema operativo Unix della Sezione, presso il Dipartimento di Fisica e l'Area di Ricerca.
- Il servizio deve essere basato su server in ambiente Linux, vista la sempre maggiore diffusione di tale piattaforma all'interno dell'INFN.
- La struttura di autenticazione deve regolare l'accesso ai sistemi Linux, e solo facoltativamente a quelli con altri sistemi operativi (come Tru64), dal momento che questi ultimi sono prevalentemente in via di dismissione, o comunque poco utilizzati. In alternativa, i sistemi diversi da Linux possono continuare ad utilizzare le architetture di autenticazione esistenti, per i pochi utenti che ancora necessitino realmente di una piattaforma diversa da Linux.
- Qualora la scelta di supportare una singola piattaforma consenta un livello di prestazioni o di sicurezza superiore rispetto ad una soluzione multipiattaforma, si privilegerà la prima alternativa.
- Deve essere garantita la compatibilità con i sistemi di condivisione dei dati esistenti, ed in particolare con il server NAS (Network Attached Storage).
- Per quanto possibile, dev'essere garantita l'autenticazione degli utenti anche in caso di problemi di comunicazione tra le sedi.
- Deve essere possibile definire dei raggruppamenti di utenti distinti con possibilità di accedere a diversi gruppi di nodi rispettando di fatto la situazione esistente, con le intersezioni viste nello schema precedente.
- Le home directory degli utenti nella nuova struttura devono essere mantenute separate per i vari gruppi di nodi: ognuno di essi fa riferimento ad un file server distinto.
- Per quanto possibile, la trasmissione dei dati tra i server, in particolare quelli relativi al sistema di autenticazione, deve essere resa sicura, utilizzando ad esempio opportuni protocolli crittati.
- La struttura deve permettere l'aggiunta, in futuro, di nuove funzionalità basate sul medesimo schema di autenticazione, ad esempio il controllo degli accessi a servizi offerti tramite il Web.

4 LA SOLUZIONE SCELTA

La scelta dell'architettura di autenticazione è stata fortemente vincolata dalla necessità di garantire la compatibilità con il Network Attached Storage, il quale supporta solo l'autenticazione tramite NIS.

A differenza di altri metodi presi in considerazione (LDAP, Kerberos), devono essere valutate alcune limitazioni rilevanti proprie di tale servizio, ed in particolare:

- basso livello di sicurezza: scambio di dati non crittato tra i server;
- basso livello di scalabilità.

È stato stabilito di realizzare la struttura interamente su piattaforma Linux, e di mantenere i preesistenti sistemi di autenticazione per i nodi di tipo diverso (Tru64), al fine di poter attivare alcune funzionalità aggiuntive disponibili solo su Linux, oppure non compatibili tra sistemi Unix diversi, come ad esempio le shadow password e la codifica MD5. In questa decisione ha influito l'utilizzo ormai molto ridotto dei sistemi Tru64, ed il limitato numero di utenti che necessitano di tale piattaforma.

Tutti i sistemi coinvolti nella ristrutturazione sono basati sul sistema operativo Linux Red Hat, versioni 7.3 e 9.

Per ovviare al basso livello di sicurezza offerto da NIS, che comprta ad esempio la trasmissione in chiaro delle mappe tra master server e slave server, con possibile rischio di intercettazioni, si è deciso di utilizzare il protocollo IPsec per creare un tunnel tra gli slave ed il master, in modo che le comunicazioni tra di essi siano crittate.

5 LA NUOVA STRUTTURA NIS

5.1 Un dominio NIS unico

La nuova struttura prevede l'attivazione di un unico dominio NIS valido per tutti i sistemi coinvolti, con un unico archivio degli utenti. All'interno di tale dominio devono essere gestiti dei metagruppi distinti di utenti che avranno la possibilità di collegarsi ad un solo sottoinsieme di nodi, oppure a più di uno. Questa situazione può essere controllata per mezzo di opportune policy previste dal sistema di autenticazione standard di Linux, basato sui moduli PAM (Pluggable Authentication Modules), ed in particolare su uno di essi denominato pam_access, come verrà descritto in seguito.

5.2 Master e slave server

Il master server del dominio NIS è installato su un server Linux, presso l'Area di Ricerca. Questo nodo gestisce anche altri servizi (ad es. un server DNS), ma non è accessibile direttamente dagli utenti, garantendo in questo modo maggiore sicurezza e stabilità.

Presso l'Area è stato inoltre installato uno slave server su un secondo nodo, accessibile dagli utenti. I client in Area fanno riferimento sia al master che allo slave, in modo da garantire la ridondanza in caso di problemi su uno dei server.

In Dipartimento di Fisica sono attivi due slave server, cui fanno riferimento tutti i client del Dipartimento. In questo modo si garantisce l'autenticazione anche in caso di problemi sulla rete tra Area e Dipartimento, e quindi di irraggiungibilità del master.

6 INSTALLAZIONE E CONFIGURAZIONE DEI SERVER NIS

L'installazione dei server NIS consiste nella sola aggiunta dei pacchetti RPM necessari (ypserv, yp-tools, ypbind). La configurazione è del tutto intuitiva, anche se leggermente diversa a seconda che si tratti del master server o di uno slave.

Di seguito verranno presentati alcuni appunti utili. Per maggiori informazioni consultare la documentazione relativa al NIS.

6.1 Alcuni appunti per la configurazione dei server

- Nel file /etc/sysconfig/network inserire la linea:

  ```
  NISDOMAIN=nome-del-dominio-NIS
  ```

- Nel file /etc/yp.conf inserire la linea:

  ```
  domain nome-del-dominio-NIS server localhost
  ```

  Questo è utile soprattutto sugli slave, ma si può fare anche sul master. In questo modo ogni server fa riferimento a se stesso per le autenticazioni degli utenti, invece di contattare il master server.
- In /var/yp/securenets inserire la lista delle reti o, meglio, dei client autorizzati che il server NIS deve riconoscere:

  ```
  255.255.255.255 140.105.221.10
  255.255.255.255 140.105.221.15
  255.255.255.255 140.105.221.16
  (…)
  ```

- Solo sul master server, modificare il file /var/yp/Makefile:
  - Sulla linea che definisce le mappe da propagare, in corrispondenza del tag "all" inserire solo quelle necessarie:

    ```
    all: passwd shadow group hosts netgrp
    ```

  - Assegnare alla variabile NOPUSH il valore "false". In questo modo il master server viene autorizzato a propagare le mappe agli slave.
- L'inizializzazione dei server viene effettuata con il comando ypinit:
  - Master server:

    ```
    $ /usr/lib/yp/ypinit –m
    ```

  - Slave server:

    ```
    $ /usr/lib/yp/ypinit –s masterserver
    ```

- Definire i servizi da far partire allo startup:

  - Master server: ypserv, yppasswdd, ypbind. Anche se gli utenti non avranno diritto di accesso al master server, ypbind va comunque attivato. Il blocco degli accessi può essere imposto agendo opportunamente sui PAM, come descritto in seguito.
  - Slave server: ypserv e ypbind.

- Sul master server, inserire gli indirizzi degli slave server nel file /var/yp/ypservers, uno per riga. La lista contenuta in questo file viene utilizzata in fase di propagazione delle mappe dal master agli slave, ogni volta che le stesse vengono aggiornate o modificate.

6.2 La propagazione delle mappe dal master server agli slave

Ogni volta che viene effettuata una modifica nelle mappe gestite dal server NIS, ad esempio quando viene aggiunto un utente, oppure quest'ultimo cambia la propria password per mezzo del comando yppasswd, il master server si occupa di propagare le mappe agli slave tramite il comando yppush. In questo modo le mappe vengono allineate, e ogni slave può fornire ai client i dati aggiornati.

Il comando yppush ha effetto solo se nel Makefile del master server è stato assegnato il valore "false" al parametro NOPUSH. La lista degli slave verso i quali propagare le mappe viene fornita dal file /var/yp/ypservers.

Nel caso in cui, al momento della propagazione delle mappe da parte del master, qualche slave risulti irraggiungibile, il tentativo di propagazione fallisce, e gli slave in questione si ritroveranno delle mappe non aggiornate.

È consigliabile pertanto l'attivazione di un meccanismo che consenta agli slave server di richiedere periodicamente un aggiornamento delle mappe al master, qualora queste risultassero non allineate. Questo può essere fatto tramite il comando ypxfr. Generalmente è consigliabile definire un'entry in crontab in modo da eseguire ypxfr periodicamente per le mappe che si desidera aggiornare. La directory /usr/lib/yp contiene alcuni esempi di script realizzate per questo scopo: ypxfr_1perday, ypxfr_1perhour, ypxfr_2perday.

Sugli slave server della Sezione di Trieste viene eseguita ogni ora una script denominata /root/ypxfr/ypxfr_sync, creata sulla falsariga di quelle sopra indicate in modo da scaricare solo le mappe necessarie.

7 INSTALLAZIONE E CONFIGURAZIONE DEI CLIENT NIS

I client NIS non necessitano di particolari accorgimenti. Dopo aver installato gli RPM necessari (ypbind e yp-tools) è sufficiente completare la configurazione:

- Nel file /etc/sysconfig/network inserire la linea:

  `NISDOMAIN=`*nome-del-dominio-NIS*

- Nel file /etc/yp.conf inserire le linee:

  `domain` *nome-del-dominio-NIS* `server` *server1*
  `domain` *nome-del-dominio-NIS* `server` *server2*

Al posto di server1 e server2 inserire i nomi dei server master o slave da contattare. L'ordine è ininfluente, in quanto la richiesta di connessione al dominio da parte del client viene comunque inviata a tutti i server elencati, senza meccanismi che regolino la priorità. In questo modo il client sarà in grado di interpellare un secondo server in caso di problemi su quello contattato inizialmente, dopo un certo tempo di timeout.

- Sui server è necessario includere il client in /var/yp/securenets.

8 GESTIONE DEGLI ACCESSI A LIVELLO DI PAM

Il controllo degli accessi a diversi sottoinsiemi di macchine da parte di gruppi distinti di utenti viene gestito interamente tramite i PAM (Pluggable Authentication Modules) di Linux. In particolare, su tutti i nodi (server e client) viene forzato l'uso del modulo pam_access. Tale configurazione può essere operata sui singoli moduli di autenticazione definiti in /etc/pam.d oppure direttamente in /etc/pam.d/system-auth in modo da agire su tutti i moduli che richiamano il PAM system-auth:

```
(…)
account     required      /lib/security/$ISA/pam_access.so
account     required      /lib/security/$ISA/pam_unix.so
(…)
```

Il file /etc/security/access.conf deve contenere le regole per l'accesso al sistema, che verranno lette da pam_access.

Nel nostro caso il file access.conf utilizzato sul master server, che deve consentire l'accesso solo a root, è necessariamente diverso da quello in uso su tutti gli altri nodi, sui quali determinati metagruppi di utenti devono poter eseguire il login.

8.1 access.conf per il master server

Il file /etc/security/access.conf per il master server non necessita di particolari accorgimenti. È sufficiente disabilitare l'accesso a tutti gli utenti escluso root da qualsiasi terminale:

```
(…)
-:ALL EXCEPT root:ALL
```

8.2 access.conf per slave server e client

Per gestire l'accesso ai diversi insiemi di macchine da parte di distinti raggruppamenti di utenti del dominio NIS, è necessario definire prima di tutto i metagruppi di utenti, da utilizzare in seguito all'interno del file access.conf.

Inizialmente il progetto prevedeva la creazione di tanti metagruppi quanti sono i gruppi di sistemi da gestire: ad es. "usrarea" per raggruppare tutti gli utenti autorizzati ad accedere ai sistemi in Area di Ricerca; "usrdip" per quelli autorizzati ad entrare sui sistemi del Dipartimento di Fisica, e così via. In questo modo all'interno del file /etc/group si formano

delle entry molto lunghe, in quanto ogni metagruppo, contenente anche un numero elevato di utenti, viene definito in una singola riga del file.

Purtroppo il NIS non è in grado di gestire entry di lunghezza superiore a 1024 caratteri. Esiste la possibilità di forzare questo limite, ma ciò va in contrasto con alcune funzionalità del servizio NIS, ed è quindi sconsigliato.

Abbiamo quindi suddiviso i metagruppi in un certo numero di "spezzoni", ad es. usrarea0, usrarea1, usrarea2, …, inserendo gli utenti in questi nuovi metagruppi in modo tale da restare sotto il limite di 1024 caratteri per ognuno di essi. I GID (numeri di identificazione dei gruppi) dei metagruppi sono tutti oltre il valore 500, in modo da renderli esportabili dal NIS.

Una volta definiti i metagruppi, diventa banale limitare l'accesso agli insiemi di nodi ai soli utenti autorizzati, creando un file /etc/security/access.conf appropriato. Ad esempio, per i sistemi del Dipartimento di Fisica, cui devono avere accesso solo gli utenti raggruppati nei metagruppi usrdip*:

```
(…)
+:root:ALL
+:usrdip0 usrdip1 usrdip2:ALL
-:ALL:ALL
```

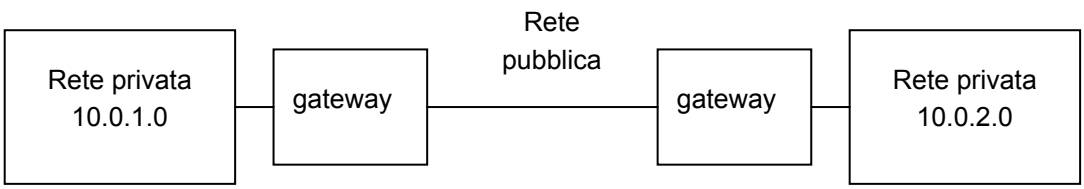
Fatto questo, sarà sufficiente includere un nuovo utente in uno qualsiasi dei metagruppi usrdip* per autorizzarlo ad eseguire il login sui nodi aventi un file access.conf come quello sopra.

9 IPSEC

Uno dei principali limiti del NIS consiste nel non prevedere un sistema sicuro per la trasmissione dei dati tra i server master e slave: ogni volta che una mappa viene aggiornata, essa viene inviata dal master agli slave in chiaro. Questo può essere di solito accettabile se i due nodi risiedono nella stessa LAN, eventualmente protetta da interferenze esterne con un firewall o altri sistemi. Nel caso della Sezione di Trieste, però, esiste la necessità di effettuare il trasferimento tra server che risiedono in siti geografici distinti.

Una soluzione a questo problema consiste nell'instaurare un tunnel con protocollo IPsec tra i server, in modo che tutte le comunicazioni tra gli stessi vengano crittate.

Il software FreeS/WAN permette di definire un tunnel IPsec tra due sottoreti distinte, collegate tra loro tramite due gateway:



La nostra situazione è un caso particolare e semplificato dello schema precedente, in quanto si richiede di attivare il tunnel solo fra due nodi.

I paragrafi seguenti presentano qualche informazione utile per l'installazione e la configurazione di FreeS/WAN nel caso descritto. Per ulteriori informazioni si consiglia di consultare la documentazione sul sito di FreeS/WAN.

9.1 Installazione

Sul sito di FreeS/WAN sono disponibili gli RPM necessari, precompilati per le varie versioni del kernel di Linux. Sono necessari due pacchetti: freeswan-module e freeswan-userland. È importante ricordare di eseguire nuovamente l'installazione dei moduli di freeswan aggiornati ogni volta che si esegue l'aggiornamento del kernel di Linux.

L'installazione e la configurazione del servizio devono essere effettuate sulle coppie di sistemi che si desidera comunichino tra loro in modo crittato.

La procedura è molto semplice:

1. Installare gli RPM. Il servizio ipsec viene automaticamente inserito nello startup del sistema, ai livelli opportuni.
2. Generare la chiave di autenticazione per il nodo:

```
$ ipsec newhostkey --output /etc/ipsec.secrets --hostname hostname
$ chmod 600 /etc/ipsec.secrets
```

3. Attivare il servizio:

```
$ service ipsec start
```

La prima volta che il servizio viene attivato, vengono visualizzati alcuni messaggi di warning. Essi si riferiscono al fatto che non sono disponibili le chiavi necessarie al programma, che vengono generate nel corso della prima attivazione.

Per verificare che l'installazione sia corretta:

```
$ ipsec verify
```

Il sistema dovrebbe rispondere OK come minimo ai seguenti controlli, mentre si possono di solito ignorare warning di altro tipo, almeno in questa fase:

```
Version check and ipsec on-path                                 [OK]
Checking for KLIPS support in kernel                            [OK]
Checking for RSA private key (/etc/ipsec.secrets)               [OK]
Checking that pluto is running                                  [OK]
```

9.2 Configurazione

Il file di configurazione di IPsec è /etc/ipsec.conf.

Per ogni coppia di host da far comunicare tramite IPsec bisogna definire un "left host" ed un "right host": è del tutto ininfluente quale host si definisca "left" e quale "right". Per definire il tunnel host-to-host è sufficiente inserire nel file ipsec.conf un blocco come il seguente:

```
conn serverA-serverB
```

```
type=tunnel
left=192.168.1.101
leftnexthop=%defaultroute
leftrsasigkey=0sAQOO13........WTC/IKouojlkj
right=192.168.2.102
rightnexthop=%defaultroute
rightrsasigkey=0sAQN2l........gUBYeCUSGose6
auto=start
```

Il blocco va definito allo stesso modo nell'ipsec.conf di entrambi gli host: non si deve cambiare assolutamente nulla.

In dettaglio:

- conn serverA-serverB
  Definisce un'etichetta che identifica la connessione. Il nome è del tutto arbitrario.
- type=tunnel
  Il tipo di connessione è "tunnel" per default: questa linea è stata inserita per maggiore chiarezza.
- left=192.168.1.101
  Indirizzo IP del nodo definito "left".
- leftnexthop=%defaultroute
  Percorso per la trasmissione dei dati dal nodo left verso il nodo right: gateway, router, ecc. Usando la stringa "%defaultroute" viene utilizzata la default route.
- leftrsasigkey=0sAQOO13.....WTC/Ikouojlkj
  Chiave pubblica di IPsec per il nodo "left". Si ottiene con il comando:

```
$ ipsec showhostkey --left
```

- right=192.168.2.102
  Indirizzo IP del nodo definito "right".
- rightnexthop=%defaultroute
  Percorso per la trasmissione dei dati dal nodo right verso il nodo left.
- rightrsasigkey=0sAQN2l.....gUBYeCUSGose6
  Chiave pubblica di IPsec per il nodo "right". Si ottiene con il comando:

```
$ ipsec showhostkey --right
```

- auto=start
  Abilita l'attivazione del tunnel in modo automatico allo startup del servizio ipsec.

Dopo aver configurato il tunnel su entrambi gli host, è necessario far ripartire il servizio:

```
$ service ipsec restart
```

Il comando `route` dovrebbe a questo punto mostrare alcune entry aggiuntive nella tabella di routing, relative ad un'interfaccia virtuale chiamata ipsec0:

```
# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
serverA.ts.infn trieste-gw2.ts. 255.255.255.255 UGH   0      0        0 ipsec0
192.168.1.0     *               255.255.255.0   U     0      0        0 eth0
192.168.1.0     *               255.255.255.0   U     0      0        0 ipsec0
169.254.0.0     *               255.255.0.0     U     0      0        0 eth0
127.0.0.0       *               255.0.0.0       U     0      0        0 lo
default         trieste-gw2.ts. 128.0.0.0       UG    0      0        0 ipsec0
128.0.0.0       trieste-gw2.ts. 128.0.0.0       UG    0      0        0 ipsec0
default         trieste-gw2.ts. 0.0.0.0         UG    0      0        0 eth0
```

Un'ulteriore verifica è possibile usando il comando `ipsec whack`, che dovrebbe mostrare alcune linee contenenti il nome del tunnel che è stato attivato:

```
# ipsec whack --status
(…)
000 #4: "serverB_serverA" STATE_QUICK_R2 (IPsec SA established);
EVENT_SA_REPLACE in 21379s; newest IPSEC; eroute owner
000 #4: "serverB_serverA" esp.3b5b970d@192.168.2.102
esp.b4396049@192.168.1.101 tun.1004@192.168.2.102 tun.1003@192.168.1.101
000 #6: "serverB_serverA" STATE_MAIN_R3 (sent MR3, ISAKMP SA
established); EVENT_SA_REPLACE in 1607s; newest ISAKMP
000 #2: "serverB_serverA" STATE_QUICK_I2 (sent QI2, IPsec SA
established); EVENT_SA_REPLACE in 20831s
000 #2: "serverB_serverA" esp.3b5b970c@192.168.2.102
esp.b4396048@192.168.1.101 tun.1002@192.168.2.102 tun.1001@192.168.1.101
(…)
```

Oppure si può eseguire un ping da uno dei due host verso l'altro, monitorando nel contempo la situazione sull'host da cui si esegue il ping, per mezzo del comando tcpdump:

```
# tcpdump -i eth0 | grep serverA
(…)
13:30:01.009766 serverB.ts.infn.it > serverA.ts.infn.it:
ESP(spi=0x3b5b970d,seq=0x417)
13:30:01.009804 serverB.ts.infn.it > serverA.ts.infn.it:
ESP(spi=0x3b5b970d,seq=0x418)
(…)
```

Il grep serve solo ad isolare i pacchetti relativi all'host bersagliato dal ping. La stringa "ESP" indica che la connessione è crittata.

10 STRUMENTI PER LA GESTIONE DEGLI ACCOUNT

La struttura realizzata comporta un aumento della complessità nelle operazioni di ordinaria amministrazione degli account: creazione, rimozione, definizione o modifica della data di scadenza.

La gestione degli account deve essere effettuata inevitabilmente sul master server del dominio NIS. Devono però essere considerate le problematiche legate al fatto che le home directory degli utenti vengono esportate da file server distinti per i vari gruppi di nodi. Inoltre deve essere gestita in modo appropriato l'assegnazione degli utenti ai metagruppi usrdip*, usrarea*, ecc., in modo da consentire l'accesso ai soli sottoinsiemi di nodi desiderati.

Al fine di semplificare la gestione sono state realizzate alcune script destinate all'esecuzione dei compiti più comuni:

- Creazione di un account.
- Rimozione di un account.
- Assegnazione/modifica della data di scadenza.

In futuro la disponibilità di strumenti potrebbe essere incrementata, realizzando ulteriori script per l'esecuzione di compiti meno frequenti, i quali possono tuttavia essere effettuati anche direttamente per mezzo dei comandi del sistema operativo (es. useradd, usermod, ecc.).

10.1 Creazione di un account

La script CreaUtente.pl effettua la creazione di un account.

In fase di immissione dati vengono richiesti:

- username;
- UID e GID di default;
- descrizione, ad es. nome e cognome;
- gruppi di macchine sulle quali deve essere attivato l'account (metagruppi);
- path della home directory;
- shell di default;
- data di scadenza;
- password;
- gruppi di utenti aggiuntivi nei quali deve essere inserito l'utente, esclusi il gruppo "privato" corrispondente allo username e i metagruppi.

La script effettua quindi le seguenti operazioni:

- creazione dell'utente, senza home directory;
- inserimento dell'utente nei metagruppi desiderati e negli eventuali gruppi aggiuntivi, all'interno del file /etc/group;
- rebuild delle mappe del NIS e trasmissione delle stesse agli slave server;

- creazione delle home directory sui file server dei gruppi di macchine ai quali l'utente ha accesso. La creazione delle home directory viene effettuata tramite chiamate ssh verso i file server interessati;
- definizione delle disk quota per l'utente sui file server. Anche questa operazione viene effettuata tramite ssh.

10.2 Rimozione di un account

La script CancellaUtente.pl esegue la rimozione di un account.
La script effettua le seguenti operazioni:

- salvataggio dei dati dell'utente (solo /home/*username*) in formato .tar.gz. Questa operazione deve essere eseguita singolarmente sui file server sui quali l'utente ha una home directory, e viene quindi effettuata per mezzo di chiamate ssh;
- rimozione delle home directory;
- cancellazione dell'account sul master server NIS;
- rebuild delle mappe del NIS in modo da eliminare l'account dal database e trasmissione delle stesse agli slave server.

10.3 Definizione della data di scadenza

La script CambiaDataScadenza.pl permette di definire o modificare la data di scadenza per un account.

11 ESPANDIBILITÀ

Con gli stessi criteri visti sopra, sarà possibile definire in futuro altri insiemi di macchine sui quali l'accesso potrà essere limitato a determinati gruppi di utenti. Ad esempio è possibile pensare ad un mail server (eventualmente in cluster per questioni di fault-tolerance) definito come slave server NIS in modo da autenticare con lo stesso database di password del dominio NIS, ma comunque in grado di "sopravvivere" ad un'eventuale mancanza di connessione con il master server.

Analogamente si può pensare di utilizzare lo stesso dominio NIS per fornire l'autenticazione ad un server Web, ad esempio per consentire l'accesso a determinati servizi riservati tramite interfaccia Web.

12 RIFERIMENTI

1. The Linux NIS(YP)/NYS/NIS+ HOWTO (http://www.tldp.org/HOWTO/NIS-HOWTO/index.html).
2. Linux FreeS/WAN (http://freeswan.org/)