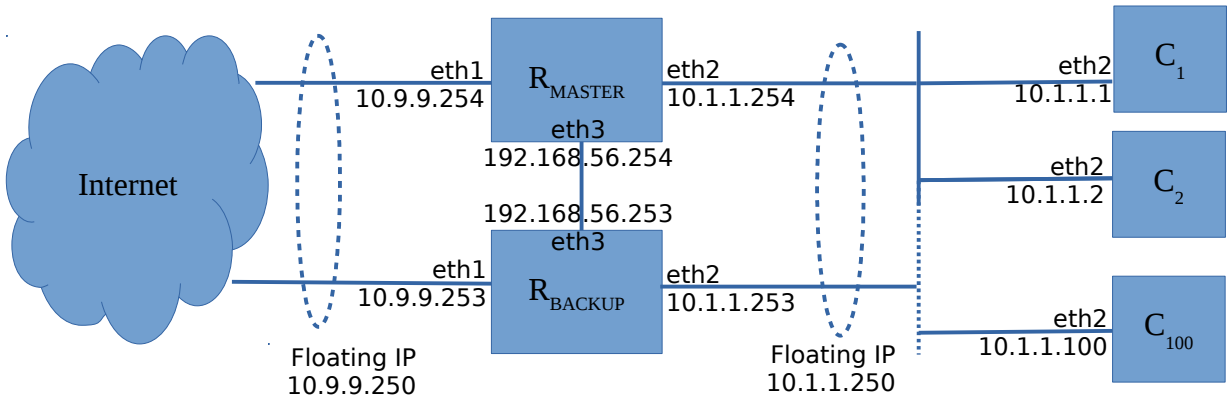
## Laboratorio di Amministrazione di Sistemi T

# Prova pratica del 30 ottobre 2018

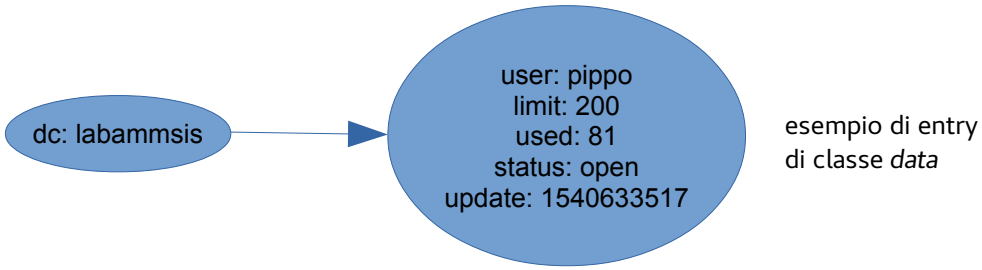
### Descrizione generale del problema




Una rete che ospita 100 client è connessa a Internet attraverso una coppia di macchine ($R_{MASTER}$ e $R_{BACKUP}$) che cooperano per svolgere il ruolo di router-firewall in alta disponibilità.

A tal fine, le due macchine hanno propri indirizzi IP, configurati staticamente (in figura: quelli con byte finale 254 o 253); inoltre quella che in un dato momento è attiva come router-firewall provvederà (in qualche modo non oggetto dell'esame) a configurare dinamicamente sul lato Internet l'indirizzo di servizio 10.9.9.250, e sul lato client l'indirizzo di servizio 10.1.1.250, che i client devono usare come default gateway e come indirizzo per comunicare col router attivo in ogni modo necessario (es. syslog, LDAP, ecc).

Ogni utente del sistema può usare qualsiasi client (in ogni istante un client può essere utilizzato da un solo utente) e ha a disposizione una quantità di traffico registrata nella directory LDAP replicata sulle due macchine e organizzata come nell'esempio di DIT qui riportato:




Le entry di classe *data* contengono il massimo quantitativo di dati (*limit*) che un utente (*user*) può complessivamente inviare o ricevere attraverso il router-firewall prima di essere disconnesso, il quantitativo già usato (*used*), lo stato del firewall nei confronti dell'utente (*status*); infine l'attributo *update* indica l'ultimo aggiornamento della entry.

*limit* e *used* sono espressi in MB (milioni di byte).

**File da consegnare**

(tra parentesi quadre la collocazione – R significa che deve essere disponibile su entrambi i router, C su tutti i client – e il punteggio indicativo; non fanno parte del nome)

**data.schema.ldif [R, 3]** – Definire i tipi di attributo *user*, *status* (stringhe), *limit*, *used, update* (interi), e la classe *data* che li utilizzi come in figura.

**connect.sh [C,11]** – Lo script viene usato da un utente per richiedere al RF di abilitare la connessione verso Internet. Per fare ciò, invia al router attraverso rsyslog un messaggio contenente l'indirizzo del client e il nome dell'utente, e si pone in attesa che *status* della relativa entry LDAP diventi *open*. Se ciò non avviene entro 5 secondi, stampa un messaggio d'errore e termina. Non appena ciò accade stampa un messaggio di successo ed entra in un ciclo in cui ogni 10 secondi controlla nuovamente *status*; se rileva che tale attributo assume un valore diverso da open, stampa un messaggio che dichiari il nuovo valore e termina.
Indicare nei commenti come configurare rsyslog sui client e sui router perché i messaggi inviati dai client vengano scritti nel file **`/var/log/reqs`** di $R_{MASTER}$ e $R_{BACKUP}$.

**snmpuser.sh [R, 6] –** Questo script ricava via SNMP quale utente sta eseguendo *connect.sh* sul client specificato come parametro; restituisce una stringa vuota se *connect.sh* non è in esecuzione.
Indicare nei commenti come configurare l'agent SNMP dei client per consentire il controllo.

**fwmanage.sh [R, 5]** - Questo script accetta come parametri un carattere e un indirizzo IP.

- Se il carattere è **`I`** abilita il traffico diretto dall'indirizzo IP verso Internet, nascondendo il reale IP sorgente
- Se il carattere è **`D`** rimuove le regole inserite in precedenza.

**ldapmod.sh [R, 5] –** Questo script accetta tre parametri: uno username, il nome di un attributo, e il relativo valore. Modifica su entrambi i router la entry LDAP relativa allo username impostando il nuovo valore per l'attributo indicato, e aggiorna sempre *update* al tempo corrente.

**check.sh [R, 8]** – Questo script osserva senza sosta il file `/var/log/reqs`, e per ogni riga corrispondente a un messaggio generato da *connect.sh* controlla che l'utente restituito da *snmpuser.sh co*incida con l'utente dichiarato nel messaggio, e in tal caso che l'utente abbia status=**`closed`**.
Se i controlli sono superati, lancia *fwmanage.sh* coi parametri appropriati per autorizzare il traffico dal client verso Internet, e lancia *ldapmod.sh* per aggiornare la entry dell'utente ponendo *status*=**`open`**.

**traffic.sh [R, 15] -** Questo script per prima cosa verifica di essere in esecuzione sul router correntemente attivo, altrimenti termina immediatamente; se prosegue, rileva tramite *snmpuser.sh,* per ogni client abilitato a connettersi a Internet, l'utente attivo su tale client.

- Se non risulta attivo nessun utente, usa *fwmanage.sh* per rimuovere l'autorizzazione dal packet filter relativa al client;
- Se risulta un utente valido, determina il traffico complessivo gnerato dall'ultima esecuzione dello script, legge ed aggiorna il valore di *used* per incrementarlo del traffico misurato, e se il nuovo valore di *used* supera *limit*:
  - usa *fwmanage.sh* rimuovere l'autorizzazione al traffico dal client verso Internet,
  - lancia *ldapmod.sh* per aggiornare la entry dell'utente ponendo *status*=**`locked`**.
  - fa in modo che dopo 30 minuti *used* venga azzerato e *status* venga posto a **`closed`**

Indicare nei commenti come far eseguire automaticamente lo script ogni 5 minuti.

**fwinit.sh [R, 11] –** Questo script configura i packet filter dei router perché blocchino tutto il traffico non strettamente indispensabile agli script, mantenendo inoltre abilitato il traffico SSH tra i due router (su tutte le interfacce)