Laboratorio di Amministrazione di Sistemi L-A

Prova pratica del 7 maggio 2012

Descrizione del problema – Si vuole realizzare un sistema di archiviazione in cui un pool di client collocati sulla rete 10.1.1.0/24 è disponibile all'utenza, che li impiega per richiedere file ad un pool di server (10.9.9.1 … 10.9.9.9). Per ipotesi i client sono workstation utilizzabili da un solo utente per volta. I file potrebbero non essere immediatamente disponibili (es. devono essere recuperati da un nastro) per cui la richiesta viene inoltrata dal client al server, che poi recapita il file in un momento successivo, secondo il procedimento descritto in seguito.

La comunicazione tra client e server avviene attraverso un router/firewall, che ha indirizzi 10.1.1.254 e 10.9.9.254 rispettivamente sulle due reti, ed ospita una directory LDAP.

La directory serve a conservare le informazioni di uso delle risorse da parte degli utenti: ad ogni trasferimento eseguito, il server aggiorna il computo di risorse impiegate dall'utente. Periodicamente, il contatore viene resettato. In base alle risorse consumate, il router/firewall decide se e come consentire ad ogni utente l'accesso ai server.

Si ipotizzi che gli account utente siano distribuiti centralmente, e quindi disponibili in modo del tutto omogeneo su tutti i client e su tutti i server, con accesso via ssh senza password già configurato.

Script da realizzare

snmpd.conf – modifiche da apportare al file di configurazione di default degli agent SNMP dei client per consentire la rilevazione dell'utente attivo, attraverso la mappatura sul sottoalbero 1.3.6.1.4.1.2021.60 dell'output del comando /bin/ps h -o user -C richiedi.sh

traffico.schema – schema LDAP che definisca gli attributi *utente* (stringa) e *traffico* (interi) e la classe *risorse* che li utilizzi obbligatoriamente entrambi.

richiedi.sh – script da utilizzare sul client per richiedere un file: accetta sulla riga di comando il nome del file, seleziona in modo pseudo-casuale un server del pool, ed invoca via ssh il comando *recupera.sh <nome_file>* sul server in modo non bloccante.

Se la connessione fallisce, lo script avvisa l'utente di riprovare dopo 10 minuti e termina.

Se la connessione ha successo, *recupera.sh* può impiegare un tempo arbitrario, mentre questo script deve proseguire immediatamente dopo la sua invocazione, per attendere la comparsa del file richiesto nella home dell'utente. In particolare, deve controllare ogni 10 secondi se il file è arrivato, avvertendo l'utente in caso positivo e terminando. Se dopo 5 minuti il file non è comparso, deve terminare avvisando l'utente del fallimento dell'operazione.

recupera.sh – script da utilizzare sul server per pianificare il recupero e la consegna del file.

Lo script lancia un comando "*get <nome_file>*" che si ipotizza esistere sul sistema e che ha l'effetto (in caso di successo) di collocare il file nella directory corrente. Dopo aver testato il successo di *get*, lo script copia il file via scp sul client che l'ha richiesto.

Al termine, lo script crea se non esiste (o aggiorna se esiste) l'entry LDAP dell'utente corrispondente, inizializzando (o incrementando) il valore di *traffico* con la dimensione del file trasferito.

gate.sh – script che ascolta permanentemente il traffico che fluisce sul router diretto dalla rete dei client verso la rete dei server sulla porta TCP/22, limitatamente ai pacchetti di inizio e fine connessione. Ad ogni tentativo di inizio connessione, verifica via SNMP l'utente che l'ha generato, ricava via LDAP il consumo dell'utente, ed abilita la connessione nel packet filter se *traffico* è minore di 20MB. Ad ogni fine connessione, rimuove la regola precedentemente inserita.

reset.sh – script eseguito ogni 10 minuti sul router (indicare nei commenti come ottenere questo effetto) che individua tutte le entry LDAP della classe *risorse* ed azzera il valore *traffico* per ogni entry trovata.

iptables-init.sh – script per la configurazione iniziale del packet filter del router. Deve bloccare tutto quanto non strettamente necessario con una modalità che notifichi esplicitamente via ICMP lo stato di chiusura delle porte a chi tenta di accedervi.