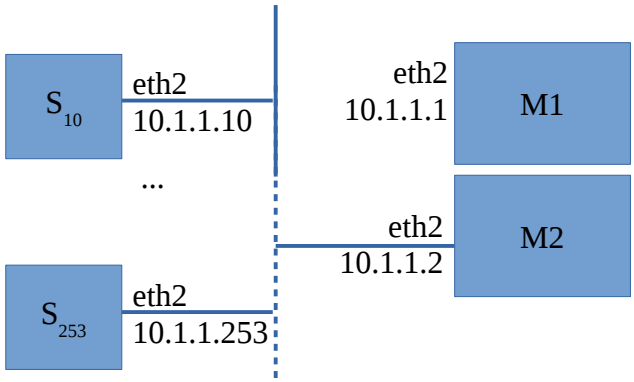
Laboratorio di Amministrazione di Sistemi T

Prova pratica del 29 giugno 2017

Descrizione generale del problema



Un sistema di monitoraggio ad alta disponibilità è costituito da una coppia di macchine “monitor” (M1 e M2).

In condizioni normali M1 effettua la rilevazione periodica di alcuni parametri di funzionamento dei server di un pool che può contenere fino a 244 macchine numerate da 10 a 253, non necessariamente tutte presenti o sempre accese.

M2 riceve continuamente sotto forma di log una copia dei dati rilevati da M1; se questo flusso si interrompe, si ipotizza che questo sia un sintomo certo del guasto di M1, e quindi si deve fare in modo che M2 ne prenda il posto (e che torni quiescente al riprendere del flusso).

Su entrambe le macchine le informazioni vengono memorizzate su di una directory LDAP locale, e usate poi per calcolare statistiche.

File da consegnare
(tutti gli script e i file di configurazione devono essere disponibili su entrambi i Monitor – normalmente sono in esecuzione su M1, fatta eccezione per track.sh che viene eseguito su M2, e fatta eccezione per le condizioni di guasto descritte in tale script)

detect.sh – Lo script esegue ogni 5 secondi una scansione dell'intervallo di indirizzi dei server per rilevare quali sono accesi (implementare il controllo in modo che tutte le potenziali risposte al *ping* siano ricevute nel giro di 1-2 secondi).
Per ogni nuovo server trovato rispetto all'iterazione precedente, invoca *init.sh* passando l'indirizzo come parametro, e al termine di questa sequenza scrive l'elenco completo degli indirizzi di tutti i server accesi nel file `/root/up.server.list`

init.sh – Lo script si collega alla macchina specificata come parametro per:

- installare lo script *counter.sh* copiandolo dalla cartella `/root` della macchina Monitor
- configurare il packet filter perché possano essere contati i byte in uscita da ognuna delle porte TCP specificate, una per riga, nel file `/root/ports.to.monitor` (presente sul Monitor)
- configurare l'agent SNMP in modo da consentire la rilevazione del traffico attraverso l'esecuzione di *counter.sh*

Indicare nei commenti cosa è necessario predisporre per consentire la connessione non interattiva.

counter.sh - Questo script, lanciato dall'agent SNMP di un server, individua le regole iptables inserite da *init.sh* per leggere (e azzerare contestualmente) il contatore dei relativi byte osservati, e produce un elenco di righe nel formato `<porta>:<bytes>`

traffic.schema.ldif – Definire i tipi di attributo `timestamp` (intero), `address` e `port_bytes` (stringhe) e una classe `traffic` che li contenga obbligatoriamente entrambi.

monitor.sh - Ogni volta che viene eseguito, legge le statistiche di traffico via SNMP da tutti i server accesi, e crea una nuova entry nella directory LDAP locale coi dati raccolti, identificandola per mezzo di un timestamp sufficientemente preciso da evitare duplicazioni. Lo stesso LDIF usato localmente per inserire la entry deve essere inviato all'altro monitor per mezzo di syslog.
Indicare nei commenti:

- come configurare i demoni syslog dei Monitor per consentire la scrittura sul file `/var/log/packets.log` di M2 dei messaggi ricevuti da M1;
- come garantire l'esecuzione automatica ogni 15 minuti dello script.

track.sh – Legge continuamente il file `/var/log/packets.log` e inserisce nella directory LDAP le entry che vi compaiono. Se rileva inattività (nessuna nuova entry nel file) per oltre 20 minuti, avvia un'istanza di *detect.sh* e configura il sistema per eseguire automaticamente *monitor.sh* ogni 15 minuti. Non appena dovesse comparire una nuova riga su `/var/log/packets.log`, lo script dovrà terminare *detect.sh* e arrestare l'esecuzione periodica di *monitor.sh.*