RELAZIONE DI TIROCINIO

SICUREZZA DELLE APPARECCHIATURE INFORMATICHE DI UNA RETE LOCALE

Laureando: Fabrizio GRANZA
Relatore: prof. Francesco BOMBI

Università degli studi di Padova
Facoltà di Ingegneria
Laurea triennale in Ingegneria Informatica

23 Settembre 2010

Indice

Capitolo 1

Introduzione

Questa relazione descrive i risultati del tirocinio svolto presso l'azienda *Gavia systems s.r.l.* con lede operativa a Villanova del Ghebbo (RO). L'azienda opera prevalentemente nel campo delle telecomunicazioni, fornendo servizi di consulenza, installazione, gestione e manutenzione di apparati di telecomunicazione, concentrandosi in modo particolare sulla sicurezza informatica. L'attività di questa azienda arriva a toccare la telefonia, il networking, Intenet.

L'obiettivo del tirocinio consiste nella progettazione del prototipo di un sistema hardware e software per la sicurezza delle apparecchiature informatiche di una rete locale. Il progetto nasce dalla necessità di prevenire furti di PC, stampanti e altri strumenti informatici appartenenti a reti locali installate in edifici molto estesi (ad esempio gli ospedali); per la continuità oraria dei servizi offerti, un complesso del genere non può trovare nell'installazione di sistemi di antifurto agli infissi una soluzione al problema esposto. L'idea è quella di rilevare lo disconnessione fisica di un dispositivo della rete, sfruttando il protocollo SNMP (Simple Network Management Protocol), ed interpretarlo come un potenziale atto di furto; tuttavia il sistema richiede una gestione più accurata affinché operazioni di manutenzione, che causassero eventi analoghi, non siano interpretati come azioni illecite. Nel momento in cui il sistema dovesse rilevare un potenziale pericolo, all'amministratore giungerà un messaggio tramite sms e/o tramite e-mail, contenente informazioni riguardo il probabile furto in atto.

In linea di massima il sistema di sicurezza ha un principio di funzionamento ed una struttura analoghe ad un classico sistema di anti-intrusione: tutte le apparecchiature da sorvegliare sono monitorate da dei dispositivi che fanno capo ad un nodo centrale. Infatti in una rete locale computer e stampanti sono collegati a degli switch; il sistema prevede

che questi ultimi monitorino i vari dispositivi e comunichino con un server che svolge il ruolo di nodo centrale. Come in un sistema anti-intrusione l'apertura di una porta o di una finestra apre il contatto tra due terminali, attivando così l'allarme; anche nel sistema di sicurezza per le apparecchiature informatiche l'interruzione del collegamento tra computer e switch deve far scattare l'allarme. L'evento di disconnessione fisica è di tipo hardware e mette in moto il sistema software che provvederà a fornire tutti i dettagli al nodo centrale, da cui scatterà il piano contro il furto. L'evento di tipo hardware deve essere opportunamente gestito ed interpretato affinché sia possibile generare dei messaggi da inoltrare al nodo centrale; come anticipato, questo ruolo è svolto da tutti gli switch che appartengono alla rete locale. È naturale pensare che i dispositivi di una rete provengano da costruttori diversi, per cui alcuni dati possono essere personalizzati da costruttore a costruttore. Il sistema software che gestisce i messaggi deve poter interpretare in maniera univoca le informazioni senza che ci siano margini di ambiguità; quindi i messaggi devono seguire uno standard, o meglio il sistema software deve elaborare solo le informazioni standard. A fornire queste regole c'è il protocollo SNMP (Simple Network Management Protocol) la cui conoscenza e il cui utilizzo rappresentano la chiave del progetto: è necessario sapere come e quando opera, chi coinvolge, quale standard è utilizzato nella scrittura dei messaggi generati.

Lo scopo principale del sistema è di avvisare l'amministratore di un potenziale furto in atto; ciò avviene generando e inviandogli un messaggio tramite i servizi di sms e di posta elettronica. Questi sono due servizi tempestivi, ma l'efficienza del sistema richiede che l'amministratore mantenga accesi e "in ascolto" rispettivamente un cellulare ed un software di gestione di posta elettronica su un computer o su un dispositivo di mobilità. L'amministratore, alla lettura del messaggio, saprà come interpretare la situazione: potrebbe esserci effettivamente un furto in atto, è in corso una manutenzione, dei dipendenti stanno svolgendo del lavoro straordinario. L'amministratore risulta essere la figura che gestisce la maggior parte delle azioni e che prende le decisioni sulle contromisure da attuare. Un'altra figura, di supporto all'amministratore, si rintraccia in colui che monitora la rete: svolge un ruolo analogo all'addetto alla visione delle immagini riprodotte dalle telecamere di sicurezza in un sistema di sorveglianza. Questa figura ha il ruolo di controllare la situazione in tempo reale ed intervenire in caso di anomalia del sistema.

Capitolo 2

Basi di partenza del tirocinio

Le linee guida per lo sviluppo del progetto si possono tradurre in un processo prototipale. Questa metodologia appare la più adatta in quanto nella fase iniziale i requisiti sono molto pochi e soprattutto non si è in possesso di un quadro preciso riguardante le tecnologie che dovranno supportare ed eseguire il prodotto richiesto; è possibile quindi che il modello che man mano viene creato debba subire delle modifiche anche importanti. Il lavoro si svolge in un team ristretto a poche unità che eseguono insieme le varie fasi che il piano di progetto prevede.

2.1 Piano di progetto

Una prima pianificazione di massima prevede tre iterazioni principali; la prima ha come scopo lo sviluppo del prototipo in un ambiente ristretto ad un solo calcolatore che è in grado di simulare il comportamento di una rete composta di più dispositivi. In una seconda iterazione il prototipo si adatta ad una rete reale di dimensioni contenute; infine, nella terza iterazione si testa il prototipo nella rete in cui dovrà essere installato il sistema. Ogni iterazione è composta dei seguenti punti:

- analisi dei (nuovi) requisiti,
- sviluppo del prototipo,
- collaudo e test.

2.2 Specifiche di progetto

Il sistema software deve essere in grado di rilevare le disconnessioni fisiche delle apparecchiature della rete, che sono le azioni propedeutiche ad un furto. Pertanto deve essere possibile constatare se i computer e le stampanti di rete, che rappresentano i dispositivi maggiormente soggetti ai furti in quanto si trovano nei locali di lavoro, sono connesse alla rete oppure no. Per quanto riguarda switch e router, essendo dei nodi centrali della rete, la loro collocazione avviene già in locali sicuri; tuttavia il sistema deve curare anche questi dispositivi in maniera analoga. Il protocollo SNMP (Simple Network Management Protocol) è in grado di fornire gli strumenti necessari affinché l'evento della disconnesione fisica possa essere trasformato in una serie di segnalazioni che permettano la gestione della situazione. Tale protocollo trova una valida implementazione nella suite di applicativi Net-SNMP, un prodotto open source che ha avuto origine da un progetto universitario.

Poiché è plausibile che gli eventuali furti avvengano in locali in cui non vi è personale, è molto probabile che le apparecchiature presenti siano spente; per cui è necessario che i dispositivi spenti siano continuamente monitorati dal sistema. Questa situazione pone una questione importante: i dispositivi spenti, dal punto di vista del sistema di sicurezza, appariranno come dispositivi disconnessi dalla rete oppure connessi nonostante il loro stato di inattività? Nel primo caso il funzionamento del sistema dipende dallo stato in cui si trovano le apparecchiature da monitorare, mentre nel secondo caso il sistema può operare a prescindere dallo stato in cui si trovano i dispositivi.

Il protocollo SNMP utilizza l'indirizzo IP e i numeri di porta degli switch per identificare i collegamenti interrotti, al cui estremo opposto è presente un dispositivo che è stato disconneso dalla rete. Diventa pertanto necessaria la presenza di un archivio che permetta la conoscenza di tutti i collegamenti tra gli switch e le altre apparecchiature presenti nella rete. L'archivio deve anche consentire l'associazione tra un dispositivo ed il locale in cui si trova fisicamente; considerando complessi molto estesi, la localizzazione si dovrebbe allargare considerando anche gli edifici e i piani.

L'azione della disconnessione fisica non sempre è dovuta ai furti; è possibile che alcuni dispositivi siano disconnessi dalla rete a causa di manutenzioni autorizzate, oppure perché alcuni dipendenti collegano alla rete il proprio portatile all'inizio del turno di lavoro e lo rimuovono al termine. Queste situazioni causerebbero dei falsi positivi per il sistema e, data la frequenza decisamente inferiore dei furti rispetto alle altre azioni lecite di disconnessione dei dispositivi, la maggior parte delle segnalazioni che il sistema

dovrebbe produrre non corrisponde a situazioni illecite. Ogni dispositivo quindi si può trovare in diversi stati a seconda dell'utilizzo; la conoscenza dello stato serve al sistema per capire se un dispositivo debba essere controllato o meno. Si deduce che l'archivio debba essere espanso affinché possa associare ad ogni dispositivo della rete lo stato in cui esso si trova: attivo, inattivo o in manutenzione. Poiché la maggior parte dei dispositivi è soggetta ad un uso periodico e regolare, ad ognuno di essi può essere associato l'orario in cui si trova in stato di attività, cosicché il sistema può commutare da uno stato all'altro automaticamente osservando l'orario di lavoro; tuttavia le azioni di manutenzione devono essere gestite manualmente in quanto solitamente non sono pianificate con regolarità.

Tutte le funzionalità del sistema devono essere disponibili in un software applicativo che viene realizzato sotto forma di web application e installato in un computer identificato come server SNMP.

2.3 Strumenti necessari

Lo sviluppo del prototipo avviene in ambiente Linux, sfruttando la distribuzione Ubuntu nella verisone 9.04; la conoscenza del software di base è fondamentale per la scelta, sulla base della compatibilità, dei software di ambiente che dovranno essere utilizzati per realizzare le varie componenti del sistema di sicurezza.

Come accennato precedentemente il protocollo SNMP fornisce gli strumenti per la gestione della rete; nella suite di applicativi Net-SNMP si riscontra una completa implementazione di SNMP. Per quanto riguarda l'archivio è intuitiva la progettazione di una base di dati relazionale, da realizzare mediante l'ausilio di un RDBMS (Relational DataBase Management System); durante la fase di sviluppo del prototipo il RDBMS scelto per realizzare la base di dati è MySQL. L'utilizzo di una web application richiede la presenza di un server web, identificato nella piattaforma Apache. La stesura del software avviene utilizzando i linguaggi HTML (HyperText Markup Language) e PHP (Php: Hypertext Processor).

L'aggiornamento automatico dello stato dei dispositivi, sulla base dell'orario di lavoro, prevede la creazione di operazioni pianificate; queste, in ambiente Linux, si realizzano mediante il comando crontab, che permette di mandare in esecuzione periodicamente dei comandi.

Capitolo 3

Tecnologie utilizzate

Il prototipo prevede soprattutto lo sviluppo di un sistema software che deve essere progettato sulla base del protocollo SNMP, di cui si utilizza l'implementazione fornita dalla suite di applicativi Net-SNMP. La progettazione hardware non prevede l'installazione di alcun componente fisico aggiuntivo, mentre è importante individuare i diversi dispositivi che compongono la rete perché ognuno di essi occupa un ruolo diverso all'interno del sistema di sicurezza.

3.1 Apparecchiature informatiche

Una rete locale è composta di diversi dispositivi, ognuno dei quali svolge un ruolo diverso all'interno della rete. Per poter simulare e testare il funzionamento del sistema è necessario avere a disposizione una rete reale, composta dei dispositivi che solitamente si trovano in una rete, cioé: desktop computer, computer portatili, periferiche, hub, switch, bridge, router. Per apprendere informazioni precise e dettagliate su queste apparecchiature e sul loro funzionamento si rimanda alle seguenti letture: "Reti di calcolatori" di A. S. Tanenbaum e "Reti di calcolatori" di L. R. Peterson e B. S. Davie. Tuttavia di seguito saranno trattate delle caratteristiche particolari di alcuni dispositivi che risultano essere importanti per il funzionamento del sistema.

3.1.1 Computer e periferiche

I desktop computer rappresentano le postazioni di lavoro, i quali si trovano negli uffici e pertanto rimangono fissi in questi locali dove il personale li utilizza durante i turni di

lavoro; poiché sono lo strumento di lavoro più utilizzato e si trovano in stanze la cui sicurezza contro le intrusioni non autorizzate non è garantita, il controllo di questi dispositivi è l'obiettivo principale del sistema. La situazione è completamente diversa per quanto concerne i computer portatili; solitamente questi viaggiano insieme al proprietario, il quale giunto alla propria postazione di lavoro, collega il proprio portatile alla rete. Terminata la sessione di lavoro, il dispositivo viene disconnesso dalla rete e, insieme al suo proprietario, lascia la postazione di lavoro. Mentre per un desktop computer la disconnessione fisica dalla rete è un evento straordinario (dovuto ad un furto o ad una manutenzione), per un computer portatile si tratta di un evento ordinario.

I dispositivi descritti sino ad ora, nel momento in cui entrano a far parte di una rete locale, acquisiscono un'identità all'interno della rete stessa attraverso l'assegnazione dell'indirizzo IP. Diversa è la situazione per le periferiche, tra le quali: monitor, scanner, stampanti, tastiere, ecc. Queste sono collegate ai diversi computer, ma non hanno un'identità all'interno della rete, ovvero un indirizzo; fanno eccezione alcune stampanti, chiamate stampanti di rete, che al contrario ne hanno uno. Ovviamente le perifieriche si trovano nei medesimi posti in cui si trovano i computer e pertanto, analogamente a quanto succede per i desktop computer, non ricevono garanzie di sicurezza dai locali in cui sono ospitati.

3.1.2 Switch

I moderni switch sono dotati di una porzione di software, chiamato agent SNMP (si veda la sottosezione 3.2.1), che consente di effettuare operazioni di controllo sulla sezione di rete collegata allo switch stesso; all'agent, quindi anche allo switch, è associato un indirizzo IP nonostante esso lavori ad un livello in cui non si utilizzano tali indirizzi. Lo switch è in grado di sapere quali delle sue porte sono collegate ad un altro dispositivo e quali no; in termini strettamente più tecnici lo stato di una porta collegata si dice "up", mentre quello di una porta non collegata si dice "down". La disconnessione fisica di un dispositivo da una porta dello switch, definita tecnicamente "link down", modifica lo stato della porta da "up" a "down"; viceversa la connessione fisica di un dispositivo ad una porta dello switch, chiamata anche "link up", modifica lo stato della porta da "down" ad "up". L'agent è in grado di rilevare queste situazioni e predisporre un messaggio da spedire al dispositivo designato come gestore della rete. Si richiama però a proposito una questione accennata precedentemente riguardo al riconoscimento della presenza di un dispositivo spento ad una porta; per rispondere a questo problema si introduce lo standard Wake on LAN, uno standard Ethernet che consente di avviare o meglio risvegliare un computer

in standby da una postazione remota. Il supporto Wake on LAN è implementato sulla scheda madre e sulla scheda di rete dei computer ed è indipendente dal sistema operativo presente. Questo standard ha tra le conseguenze il fatto che uno switch sia in grado di rilevare la presenza di un dispositivo su una data porta, anche se questo risulta spento. Se non ci fosse questa caratteristica, un dispositivo spento potrebbe essere disconnesso senza che lo switch rilevi l'evento; quindi non verrebbe generato alcun messaggio di allarme e di conseguenza il sistema verrebbe eluso. Poiché sono proprio i dispositivi spenti i sorvegliati speciali del sistema, il supporto Wake on LAN si mostra indispensabile per fornire la funzionalità che garantisce l'affidabilità del sistema.

3.2 Protocollo SNMP

Il modello TCP/IP, o suite di protocolli Internet, è un insieme di protocolli adottati per rendere possibile la comunicazione tra le reti di calcolatori che si presentano molto diverse ed eterogenee le une dalle altre. Il modello TCP/IP è il più famoso (insieme al modello ISO/OSI) ed è alla base del funzionamento di Internet, esso si presenta diviso in quattro livelli:

- Host to link,
- Internet,
- Trasporto,
- Applicazione.

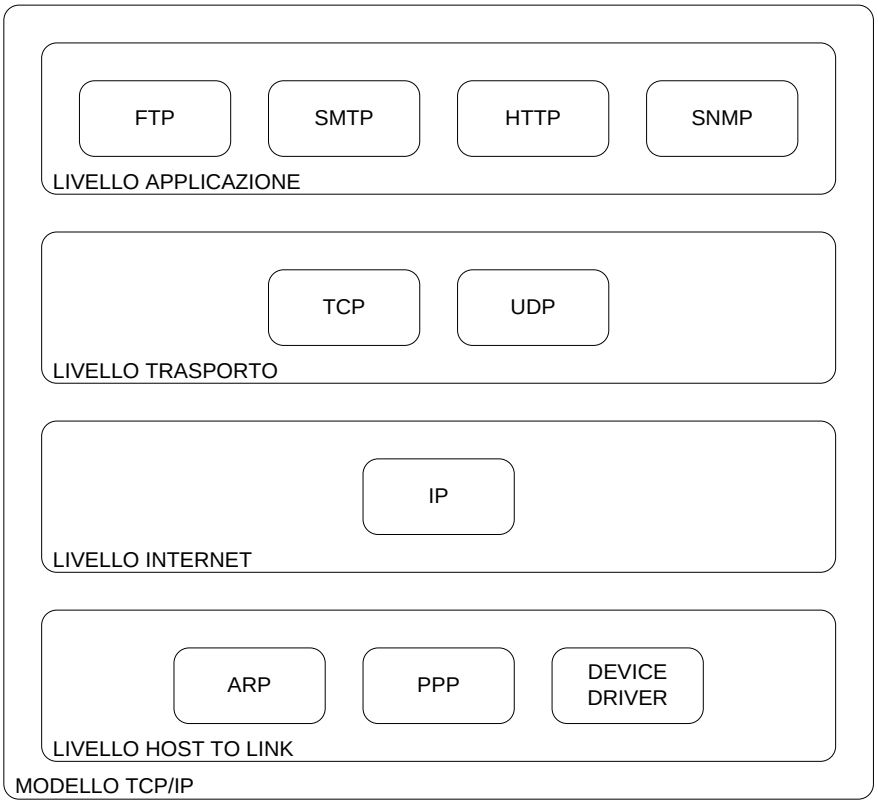

Figura 3.1: Modello TCP/IP

Le letture suggerite precedentemente forniscono molte informazioni riguardo al modello TCP/IP e ai livelli che lo compongono.

Il protocollo SNMP (Simple Network Management Protocol) appartiene alla suite di protocolli Internet situato nel livello applicazione, il quale si occupa della gestione dei dispositivi di una rete. L'IETF (Internet Engineering Task Force) definisce gli standard dei protocolli Internet, tra cui anche SNMP, di cui sono state stilate tre versioni:

- SNMPv1,
- SNMPv2,
- SNMPv3.

SNMPv1 è la versione primordiale, pienamente compatibile con gli standard dell'IETF. Per garantire la sicurezza i dispositivi da gestire sono associati a delle community, che altro non sono che password; tuttavia la sicurezza non è garantita pienamente e ciò ha condotto alla necessità di una versione successiva. SNMPv2 ha introdotto miglioramenti in performance, sicurezza e comunicazione, introducendo tra l'altro nuovi comandi

per garantire ciò. SNMPv3 è attualmente la versione standard; essa fornisce tre servizi importanti: autenticazione, privacy e controllo di accesso.

3.2.1 Ruoli in SNMP

Lavorando con SNMP è possibile individuare due diversi ruoli:

- manager,
- agent.

Un manager è un server che esegue processi di gestione per la rete, anche sotto il controllo diretto dell'operatore; spesso ci si riferisce ad esso come NMS (Network Management Station). L'agent è una porzione di software che è eseguito sui dispositivi di rete da gestire; può essere un programma a sè stante oppure incorporato all'interno del sistema operativo. Tra i dispositivi che si possono gestire tramite il protocollo SNMP si trovano router, switch, server, workstation e stampanti. Un agent possiede una serie di oggetti di cui deve tenere traccia. I comportamenti di questi oggetti rappresentano le informazioni che la NMS può utilizzare per determinare lo stato globale della rete e del dispositivo su cui risiede l'agent che è stato interrogato. La SMI (Structure of Management Information) fornisce un modo per definire tali oggetti e i comportamenti che essi possono assumere. Gli oggetti di cui un agent deve tenere traccia sono contenuti nel MIB (Management Information Base), una sorta di database che fornisce la definizione degli oggetti stessi, utilizzando la sintassi dettata dalla SMI; per un agent, il MIB appare come un dizionario. In linea generale un agent può implementare più MIB.

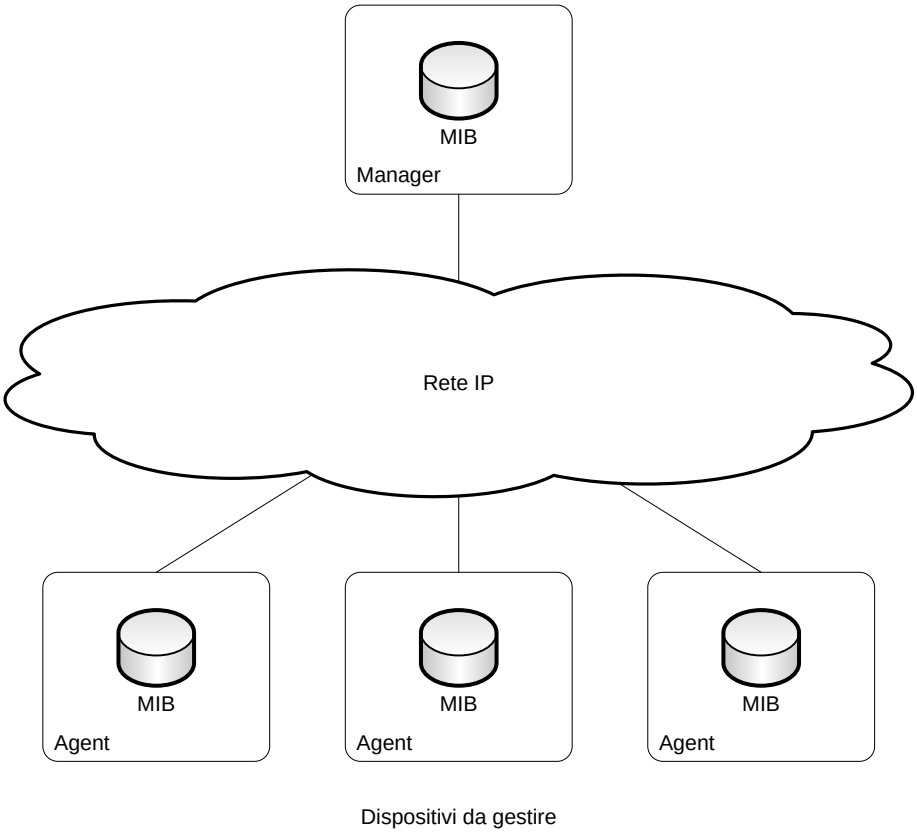

Figura 3.2: Esempio di rete

3.2.2 Comunicazione in SNMP

Una NMS deve far fronte a due diverse tipologie di comunicazione con gli agent:

- polling,
- trap.

Il polling è un'azione di interrogazione a rotazione degli agent in cui il manager richiede determinate informazioni, le quali saranno utilizzate per determinare lo stato dei dispositivi di rete e della rete stessa.

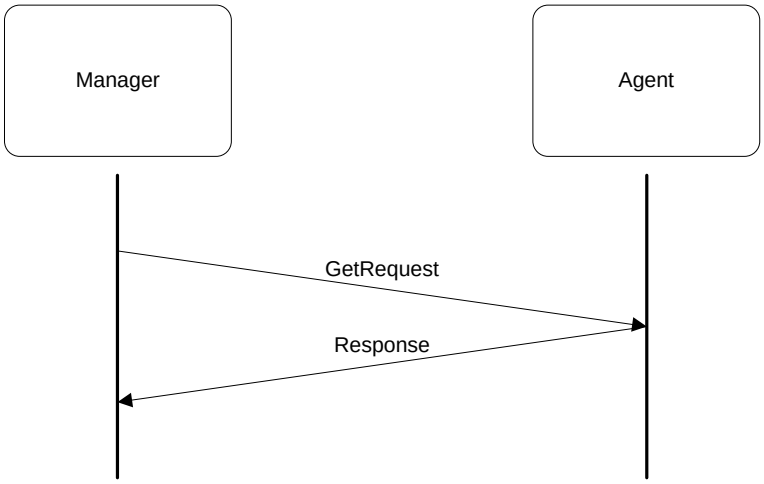

Figura 3.3: Polling

La trap è un messaggio spedito dall'agent al manager, non sollecitato da quest'ultimo, quindi non in risposta ad una interrogazione del manager, per comunicare l'avvenimento di un particolare evento; anche in questo caso il manager dovrà gestire, in maniera "straordinaria", le informazioni ricevute.

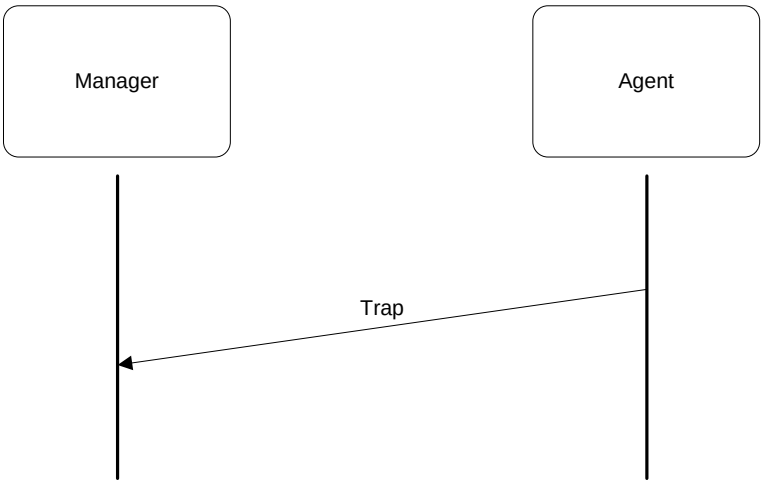

Figura 3.4: Trap

Per passare i dati tra manager e agent a livello trasporto, SNMP usa il protocollo UDP. Quest'ultimo è definito non è affidabile in quanto durante una trasmissione non c'è alcun riscontro di una eventuale perdita del messaggio. È compito dell'applicazione SNMP quindi determinare se un messaggio è stato perso o danneggiato e quindi in tali

circostanze ritrasmetterlo; il tutto è consentito grazie all'utilizzo di un timeout. Quando il manager deve interrogare un agent gli spedisce una richiesta, cioé un messaggio, e si pone in attesa della risposta; se questa non perviene entro lo scadere del timeout, l'applicazione SNMP del manager sospetta una possibile perdita del messaggio contenente la risposta dell'agent, per cui il manager stesso ritrasmette la richiesta. La mancata ricezione della risposta potrebbe anche dipendere dal fatto che il messaggio contenente la richiesta sia stato perso prima di giungere all'agent, per cui quest'ultimo, non essendo interrogato, non spedisce alcuna risposta; l'applicazione SNMP del manager cura questa situazione in modo analogo alla precedente ritrasmettendo la richiesta allo scadere del timeout. Le situazioni descritte si verificano nel caso di un polling, ma nella comunicazione tramite trap l'inaffidabilità del protocollo UDP potrebbe causare la perdita effettiva dei messaggi. Se un agent spedisce una trap che viene persa durante il trasporto, il manager non ha modo di sapere che ne è stata spedita una, tanto meno di avvisare l'agent del fatto che tale trap non gli è pervenuta; purtroppo nemmeno l'agent può avvertire la perdita della trap, poiché in caso di corretta ricezione il manager non è tenuto a comunicargli il riscontro positivo.

Nelle reti i calcolatori eseguono contemporaneamente più processi e se avvengono degli scambi di messaggi lungo la rete, sono necessarie più connessioni contemporanee da e verso altri calcolatori. Per consentire ciò, assicurandosi che i messaggi vengano indirizzati al processo del calcolatore destinatario che li sta attendendo, si utilizzano le porte ovvero i numeri identificatori delle connessioni a livello di trasporto. I processi dei destinatari devono porsi in ascolto alla porta su cui devono giungere i messaggi che stanno attendendo. SNMP utilizza la porta 161 del protocollo UDP per spedire le richieste e ricevere le risposte (nel polling), la porta 162 per ricevere le trap dagli agent.

Per garantire la sicurezza nelle comunicazioni tra manager ed agent, le prime due versioni di SNMP utilizzano il concetto di community. Esistono tre nomi diversi di community:

- read-only,
- read-write,
- trap.

Un agent è configurato con uno di questi nomi che altro non è che una password. Le community controllano differenti tipi di attività, le quali sono intuibili dai nomi stessi

delle community: la community read-only consente di leggere i valori dei dati (ovvero degli oggetti che possiede un agent), ma non di modificarli, quella read-write, invece, consente sia la lettura che la modifica dei valori dei dati, infine la community trap permette di ricevere le trap dagli agent.

3.2.3 Oggetti da gestire

Come citato precedentemente la SMI fornisce un modo per definire gli oggetti che dovranno essere gestiti dal manager. La definizione degli oggetti può essere suddivisa in tre attributi:

- nome,
- tipo e sintassi,
- codifica.

Il nome, indicato anche con la sigla OID (Object IDentifier), definisce in maniera univoca un oggetto; esso può apparire sia in forma numerica che in forma testuale. Gli oggetti da gestire sono organizzati in una gerarchia ad albero che rappresenta lo schema base per nominare gli oggetti. Un OID è formato da una sequenza di numeri separati da punti che identificano i vari nodi percorsi dalla radice per raggiungere la posizione in cui si trova l'oggetto a cui è associato tale OID.

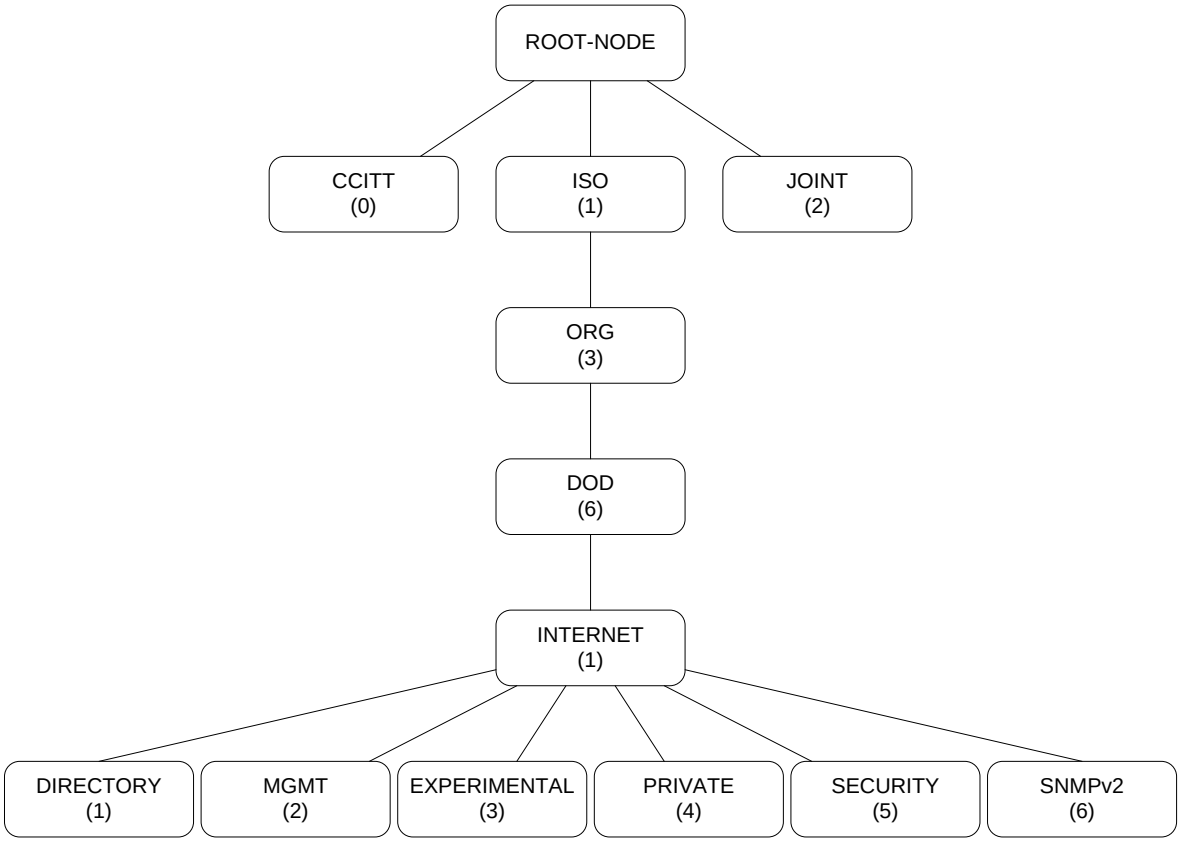

Figura 3.5: Albero degli oggetti

Come anticipato precedentemente, un oggetto può essere identificato con un nome ben preciso anziché con una sequenza di numeri. Ad esempio, osservando la figura 3.5, il nome "private" è sufficiente ad identificare univocamente quel nodo; in alternativa lo si può indicare tramite la sequenza 1.3.6.1.4.
Nel contesto del protocollo SNMP il tipo e la sintassi definiscono come i dati sono rappresentati e trasmessi tra manager e agent; questa notazione è indipendente dal tipo di macchina. Qui sono definiti i diversi tipi di dato, cioé l'insieme di valori che un oggetto può assumere, tra i quali:

- integer: rappresenta un numero intero, da -2^{31} a $2^{31}-1$, espresso attraverso 32 bit;
- counter: rappresenta un numero intero incrementante, da 0 a $2^{32}-1$, espresso attraverso 32 bit;
- gauge: rappresenta un numero intero incrementante e decrementante, da 0 a $2^{32}-1$, espresso attraverso 32 bit;

- stringa di byte: rappresenta una stringa di testo;
- object identifier: rappresenta una sequenza di numeri decimali separati da punti;
- indirizzo IP: rappresenta un indirizzo IP, espresso attraverso 32 bit;
- indirizzo di rete: simile all'indirizzo IP, rappresenta diversi tipi di indirizzi di rete;
- time ticks: rappresenta un tempo in centesimi di secondo, espresso attraverso 32 bit.

Una successiva estensione ha introdotto i seguenti tipi di dato:

- integer32: analogo al precedente integer;
- counter32: analogo al precedente counter;
- gauge32: analogo al precedente gauge;
- unsigned32: rappresenta un numero intero, da 0 a $2^{32} - 1$, espresso attraverso 32 bit;
- counter64: rappresenta un numero intero incrementante, da 0 a $2^{64} - 1$, espresso attraverso 64 bit.

La codifica definisce come gli oggetti sono codificati e decodificati cosicché dopo essere stati trasportati attraverso un mezzo fisico di trasmissione possano essere tradotti correttamente nei corrispettivi valori.

3.2.4 Operazioni SNMP

Le operazioni definite dal protocollo SNMP sono:

- get,
- get-next,
- get-bulk,
- set,
- get-response,

- trap,
- notification,
- inform,
- report.

Operazione get

L'operazione get è una richiesta effettuata dal manager e spedita ad un agent, il quale una volta ricevuta cerca di soddisfarla, rispondendo con le informazioni che il manager vuole ottenere. Non sembra essere chiaro però, come il manager specifichi l'informazione che gli interessa; infatti nella richiesta è contenuta una variabile obbligatoria che contiene una lista degli oggetti di cui il manager ne vuole conoscere i valori.

Operazione get-next

L'operazione get-next è simile all'operazione get; è una richiesta in cui il manager desidera ottenere dall'agent l'oggetto successivo a quello specificato nella richiesta stessa. Grazie all'albero degli oggetti ed alla notazione numerica degli OID è possibile stabilire una relazione d'ordine tra gli oggetti, tale per cui ha senso parlare di oggetto successivo; l'ordine che sussiste tra gli oggetti è di tipo lessicografico. Osservando la notazione numerica, è immediato stabilire quale oggetto viene prima di un altro.

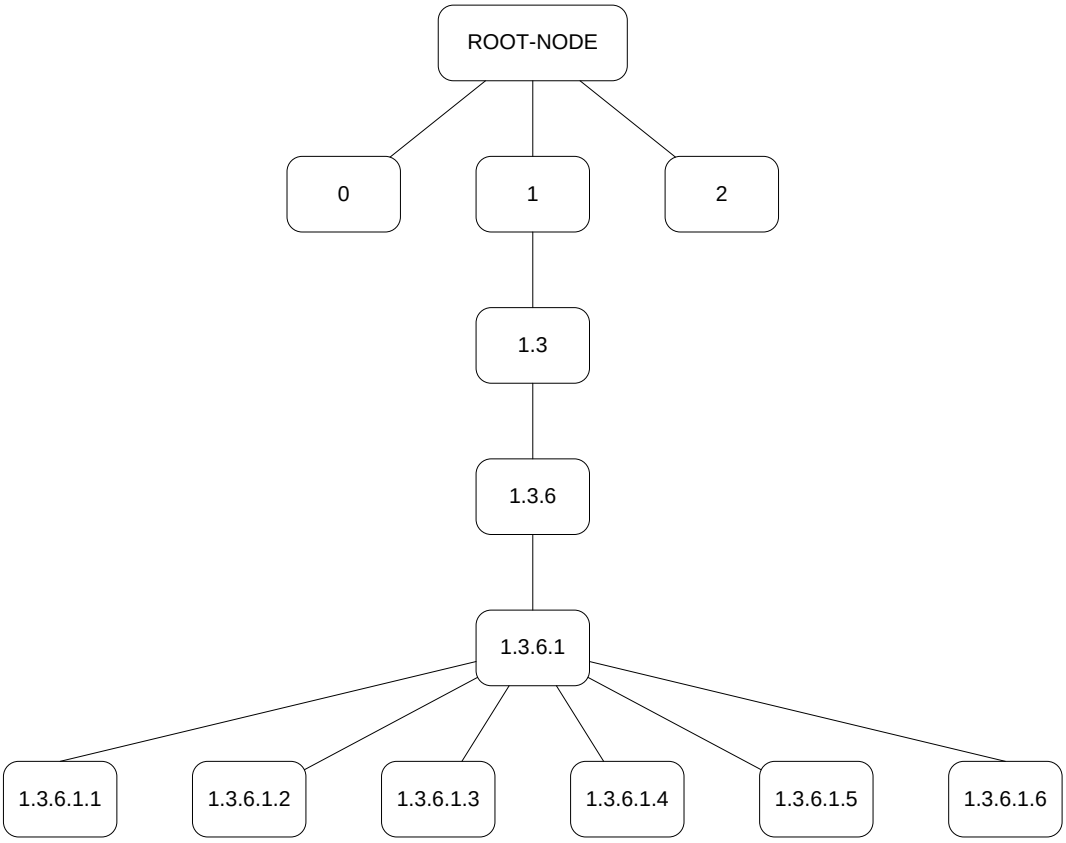

Figura 3.6: Relazione d'ordine nell'albero degli oggetti

Operazione get-bulk

L'operazione get-bulk (introdotta a partire da SNMPv2) consente di effettuare una richiesta per ottenere grandi quantità di dati; per fare ciò utilizza più volte l'operazione get-next. Oltre ai nomi degli oggetti, devono essere specificati due campi: "non-ripetenti" e "massime-ripetizioni"; il primo indica quanti oggetti possono essere restituiti con una sola chiamata get-next, mentre il secondo indica quanti oggetti devono essere restituiti singolarmente ognuno con una chiamata get-next.

Operazione set

L'operazione set è utilizzata dal manager per cambiare il valore di un oggetto o per crearne di nuovi. Solo gli oggetti di tipo read-write o write-only possono essere modificati o creati mediante tale comando. Quando si vuole impostare un valore ad un oggetto bisogna indicare l'oggetto da modificare, il tipo di dato e il nuovo valore che assumerà. L'operazione

set permette di modificare più oggetti con una singola chiamata, con lo svantaggio che un singolo fallimento di scrittura compromette anche tutti gli altri.

Operazione get-response

Ognuna delle operazioni descitte fino ad ora stimola l'agent a generare un messaggio di risposta tramite l'operazione get-response. Nel caso in cui l'agent riesca a soddisfare il comando del manager, la comunicazione si conclude con successo; purtroppo è possibile che l'agent riscontri un errore nel soddisfare le richieste del manager, a cui viene segnalato l'eventuale errore. Nel caso di risposta alle operazioni get, get-next e set il messaggio di errore può essere:

- noError (0): non c'è stato alcun errore nell'esecuzione della richiesta;
- tooBig (1): la risposta alla richiesta era troppo grande per essere spedita in una sola risposta;
- noSuchName (2): l'oggetto richiesto all'agent (per un get o per un set) non esiste;
- badValue (3): il valore a cui impostare l'oggetto è inconsistente;
- readOnly (4): l'oggetto a cui si vuole effettuare un set non può essere modificato;
- genErr (5): l'errore non rientra in nessuna delle cause precedenti.

Questo elenco di errori, stilato per SNMPv1, è stato ampliato per la versione successiva, nella quale è stata introdotta l'operazione get-bulk e soprattutto è stata migliorata la consistenza dei messaggi di errore. I nuovi tipi di errore sono:

- noAccess (6): l'oggetto a cui si vuole effettuare un set non è accessibile;
- wrongType (7): il valore a cui impostare l'oggetto non è concorde con il tipo di dato previsto;
- wrongLength (8): il valore a cui impostare l'oggetto supera i limiti di lunghezza previsti;
- wrongEncoding (9): l'operazione set sta tentando di utilizzare una codifica errata per l'oggetto da modificare;
- wrongValue (10): il valore a cui impostare l'oggetto non è comprensibile;

- noCreation (11): l'oggetto a cui si vuole effettuare un set non esiste;
- inconsistentValue (12): il MIB si trova in uno stato inconsistente, per cui non accetta modifiche;
- resourceUnavailable (13): nessuna risorsa di sistema è disponibile per eseguire un set;
- commitFailed (14): è l'errore generico nel caso di un set;
- undoFailed (15): un set è fallito e l'agent non è in grado di ripristinare l'oggetto al valore precedente il fallimento;
- authorizationError (16): non è stato possibile autenticarsi; la community è errata;
- notWritable (17): una variabile non accetta un set, sebbene sia modificabile;
- inconsistentName (18): un set è fallito perché la variabile si trova in uno stato inconsistente.

Trap

Le operazioni descritte sino ad ora implementano una comunicazione sincrona, ovvero una comunicazione in cui il manager richiede un'operazione e l'agent risponde per soddisfare tale richiesta; quindi sembrerebbe che l'agent non inizi mai la comunicazione. Invece tramite le trap l'agent è in grado di comunicare con il manager senza che quest'ultimo lo abbia sollecitato con una richiesta. Ciò serve all'agent per poter comunicare al manager la manifestazione di un evento straordinario. Alcune delle cause che potrebbero provocare la generazione di trap sono:

- l'interfaccia di rete di un dispositivo si disattiva o si riattiva,
- una chiamata al modem non è in grado di stabilire una connessione,
- si verifica un problema sullo switch o sul router.

Quando il manager riceve la trap, deve sapere come interpretarla e quali informazioni porta con sé. Una trap è identificata da un numero, chiamato generic trap number, che può assumere un valore da 0 a 6 per altrettanti significati:

- coldStart (0): l'agent è stato ricaricato (rebooted) e quindi tutte le variabili da gestire sono state azzerate, in particolare quelle di tipo incrementale (counter e gauge);
- warmStart (1): l'agent si è reinizializzato, ma nessuna variabile è stata azzerata;
- linkDown (2): un'interfaccia di un dispositivo si è disattivata;
- linkUp (3): un'interfaccia di un dispositivo si è attivata;
- authentificationFailure (4): qualcuno ha tentato di interrogare un agent con una community non corretta; questo tipo di trap è utile nel determinare se qualcuno sta tentando di ottenere un accesso non autorizzato ad un dispositivo;
- egpNeighborLoss (5): un protocollo vicino si è disattivato;
- enterpriseSpecific (6): la trap è specificata dall'impresa; i venditori e gli utenti SNMP definiscono le proprie trap da collocare in un'apposita area dell'albero degli oggetti.

Notification-type

Il PDU (Protocol Data Unit) è il formato dei messaggi che i manager e gli agent utilizzano per spedire e ricevere informazioni. Il formato PDU delle trap nella versione SNMPv1 è diversa da quello delle operazioni get e set. Nella versione SNMPv2 è stato definito un particolare tipo, chiamato notification-type, il cui formato PDU per le trap è identico a quello delle operazioni get e set.

Operazione inform

SNMPv2 fornisce un meccanismo, chiamato inform, che consente una comunicazione tra due manager. Questa operazione può esssere utile quando è necessaria la presenza di più manager per gestire la rete. Quando è spedito un inform da un manager ad un altro, il ricevente spedisce una risposta al mittente per avvisare della corretta ricezione dell'inform stesso. L'inform può anche essere utilizzato da un agent per spedire una trap ad un manager; in tal caso l'agent, al contrario di quanto accade nella trasmissione delle trap, ottiene in risposta dal manager un messaggio che conferma la ricezione dell'inform contenente la trap.

Operazione report

L'operazione di report permette ad un motore SNMP di comunicare con un altro, principalmente per rapportare problemi con i messaggi SNMP.

3.2.5 Formato delle trap

Le trap sono dei messaggi che contengono dei dati, alcuni dei quali sono standard, mentre altri sono a discrezione dell'agent che genera la trap; i dati standard sono quelli di interesse per le comunicazioni con il manager. Nell'introdurre i diversi dispositivi che compongono una rete si sono citati hub, switch e bridge; purtroppo negli hub non si riscontra la presenza di un agent, per cui i dispositivi di maggiore interesse diventano switch e bridge che al contrario ne possiedono uno. Tra i diversi dati contenuti in una trap emerge su tutti il tipo di trap, riconducibile ad uno dei sette elencati precedentemente. Si possono leggere quindi l'indirizzo IP dell'agent, la porta dello switch (o del bridge) sulla quale è stato rilevato l'evento, la community, il tempo trascorso dall'accensione dell'agent (chiamato anche up time) e altre informazioni.

3.3 Suite di applicativi Net-SNMP

Net-SNMP è una suite di applicativi che implementa il protocollo SNMP; è distribuito in diversi sistemi operativi tra cui Linux, Solaris, Mac OS X e altri. La storia del progetto è abbastanza recente e se ne trova una dettagliata descrizione nel sito web di Net-SNMP.
La suite comprende:

- una libreria,
- una serie di applicazioni a riga di comando,
- un agent.

Le applicazioni consentono di implementare le operazioni definite dal protocollo SNMP; l'agent permette alla macchina che lo ospita di essere gestita da un manager al pari degli altri agent: può quindi essere interrogato e può generare trap.

3.3.1 Applicazioni di Net-SNMP

Le applicazioni di Net-SNMP sono:

- snmptranslate: traduce un OID dalla forma numerica a quella testuale e viceversa;
- snmpget: implementa l'operazione get del protocollo SNMP;
- snmpgetnext: implementa l'operazione get-next del protocollo SNMP;
- snmpbulkget: implementa l'operazione get-bulk del protocollo SNMP;
- snmpwalk: recupera un insieme ordinato di valori da gestire utilizzando le richieste get-next;
- snmpbulkwalk: recupera un insieme ordinato di valori da gestire utilizzando le richieste get-bulk;
- snmpset: implementa l'operazione set del protocollo SNMP;
- snmptest: implementa una semplice operazione di richiesta del protocollo SNMP.

Questo primo insieme di applicativi contiene operazioni che rappresentano richieste effettuate da un manager verso un agent; infatti, tra i vari parametri da specificare nella riga di comando, si trova il nome oppure l'indirizzo IP del dispositivo ospitante l'agent destinatario della richiesta (si ricorda a proposito la compatibilità tra il protocollo SNMP e quello IP, passando attraverso il protocollo UDP). L'agent interrogato risponde con le informazioni richieste dal manager.

Con gli applicativi descritti si implementa la comunicazione di tipo polling. Per quanto riguarda l'implementazione delle comunicazione di tipo trap la situazione è leggermente più complessa. Ricordando che tale tipo di comunicazione è asincrona, l'agent, non appena ne ha la necessità, invia un messaggio al manager; poiché l'invio di una trap è sollecitato da un evento straordinario, la gestione di questi particolari messaggi deve essere tempestiva, il che condiziona il manager a posizionarsi in un continuo stato di ascolto per essere in grado di percepire all'istante l'arrivo delle trap. L'applicazione snmptrapd, eseguita sul manager, lo posiziona nello stato di ascolto in cui è in grado di rilevare e gestire le trap non appena giungono; con opportuni parametri è possibile destinare ad un file di log i messaggi relativi alla rilevazione ed alla gestione delle trap. Con tale descrizione dell'applicazione snmptrapd si è definita solamente la rilevazione delle trap; per quanto riguarda la loro gestione è fondamentale il ruolo di un file di configurazione chiamato snmptrapd.conf, che deve essere specificato tra i parametri della riga di comando quando si invoca l'applicazione snmptrapd. Nel file di configurazione si elencano i diversi tipi di

trap che si intendono gestire (quelli non specificati non vengono gestiti); ognuno di essi deve essere preceduto dal comando traphandle e seguito da un'applicazione a riga di comando. Il comando traphandle lancerà l'applicazione che segue il tipo di trap rilevato; se quel tipo non è presente nell'elenco, nessuna applicazione sarà lanciata. L'applicazione lanciata può essere l'esecuzione di un file eseguibile (tipo php o perl); il file eseguibile è chiamato solitamente handler.

Come visto in precedenza, una delle componenti della suite Net-SNMP è rappresentata da un agent; quindi il dispositivo su cui è installata tale suite può assumere anche il ruolo di agent. Per quanto riguarda la comunicazione di tipo polling, i messaggi di risposta sono generati dall'agent in occasione della ricezione delle richieste da parte del manager; diventa ben più interessante capire come Net-SNMP realizzi l'operazione di generazione delle trap negli agent. L'applicazione snmptrap implementa l'operazione trap del protocollo SNMP, permettendo all'agent di spedire un messaggio in modo asincrono; tra i parametri devono comparire il nome oppure l'indirizzo IP del dispositivo su cui risiede l'agent che ha spedito la trap ed il tipo di trap. Il manager che si trova in ascolto di eventuali trap ne riceve una e leggendo il tipo di trap è in grado di scegliere quale sequenza di operazioni attuare per la gestione della trap; il nome o l'indirizzo IP ed eventuali altri parametri rappresentano le informazioni che permettono all'handler lanciato di portare a compimento la gestione della trap. In alternativa all'applicazione snmptrap si può utilizzare snmpimport concordemente a quanto afferma il protocollo SNMP.

3.4 Operazioni pianificate

Nei sistemi operativi Linux il comando crontab consente la pianificazione di comandi, i quali vengono mandati in esecuzione periodicamente. Ogni linea di un file crontab segue un formato particolare, composta di cinque campi, seguiti dal comando da eseguire. I cinque campi si riferiscono nell'ordine:

- al minuto (da 0 a 59),
- all'ora (da 0 a 23),
- al giorno del mese (da 1 a 31),
- al mese (da 1 a 12),

- al giorno della settimana (da 0 a 6, dove 0 corrisponde alla domenica e gli altri seguono nell'ordine).

In alternativa ai valori indicati, per ogni campo è anche possibile associare un asterisco ("*") col significato di tutti i valori; ad esempio se nel campo relativo all'ora compare l'asterisco, il comando viene eseguito ad ogni ora. Altri due simboli speciali consentono di aumentare le indicazioni per la pianificazione temporale pur mantenendo una sola riga per ogni comando che si vuole eseguire; questi sono: la virgola (",") che consente di separare i valori quando si inserisce più di un valore per campo e il trattino ("-") che consente di specificare un intervallo di valori per campo evidenziandone solamente gli estremi.

Capitolo 4

Sviluppo del sistema

Il sistema di sicurezza deve essere installato in una rete qualsiasi, progettata secondo lo standard Ethernet. Nella rete devono essere presenti, oltre ai vari computer, un server ed almeno uno switch. Pertanto la rete si deve presentare come in figura 4.1.

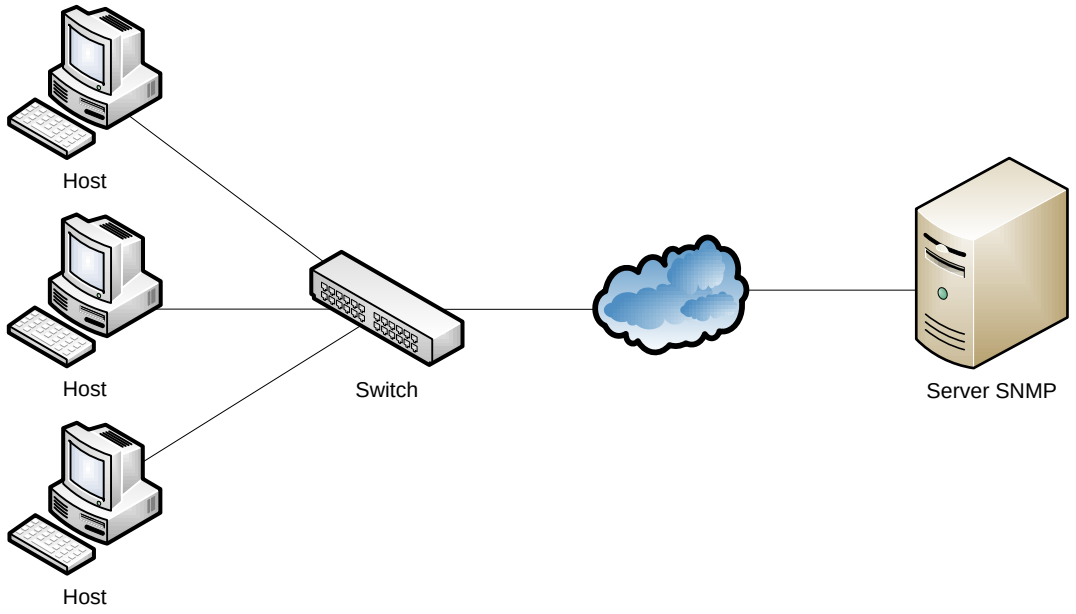

Figura 4.1: Rete

Il sistema di sicurezza identifica nel manager la stazione centrale dove confluiscono tutti i messaggi atti a segnalare gli eventi critici. Alla base del sistema vi è quindi la comunicazione tra il manager e gli agent; sfruttando le impostazioni del protocollo Telnet,

ad ogni agent si imposta l'indirizzo IP del manager a cui dovranno essere inoltrate le trap. Negli switch risiedono gli agent, mentre il computer sulla quale viene installata la suite di applicativi Net-SNMP svolge il ruolo del manager e viene identificato nella rete come server SNMP; In quest'ultimo è anche presente un database per l'archiviazione dei dati.

4.1 Progettazione del database

Nel computer che assume il ruolo del manager si deve progettare un database affinché sia possibile memorizzare dati riguardo i dispositivi che compongono la rete, i locali che li ospitano e le trap catturate. Come descritto nell'introduzione, ci possono essere più figure che interagiscono con il sistema attraverso l'applicazione e ad ognuna di esse possono corrispondere azioni diverse; pertanto nel database si devono registrare anche gli utenti e i ruoli che essi occupano nell'utilizzo dell'applicazione. Per tenere traccia sia della topologia di rete che della collocazione dei dispositivi, si deve popolare l'archivio memorizzando i vari collegamenti punto-punto e i locali in cui sono posti i dispositivi stessi.

La sezione del database dedicata ai dispositivi deve registrare tutti i dispositivi aventi un'identità nella rete, ovvero quei dispositivi che possono essere identificati mediante un indirizzo IP e collegati ad uno switch. Analizzando i dati prodotti da una trap, si possono rilevare solamente l'indirizzo dell'agent generante la trap e la porta sulla quale è stato rilevato l'evento; quindi diventa fondamentale popolare il database in modo tale da evidenziare i collegamenti tra le interfacce di rete. Essendo uno switch dotato di più porte è possibile che alcune di esse siano collegate ed altre no; un computer o una stampante di rete solitamente hanno un'interfaccia di rete ed è plausibile pensare che essa sia connessa alla porta di uno switch, altrimenti nel caso contrario il dispositivo risulterebbe isolato e quindi invisibile per il sistema.

Considerando come ambiente ospitante la rete un complesso anche molto esteso, il database deve poter registrare i dati riguardo gli edifici, i piani e le stanze; in questo modo è possibile avere informazioni molto dettagliate sui luoghi che accolgono i dispositivi della rete. Naturalmente un dispositivo si trova in una sola stanza, mentre una stanza può accoglierne anche molti. È importante registrare nel database il locale in cui si trova un dispositivo affinché in caso di emergenza sia possibile localizzarlo; se di un dispositivo non si conosce il luogo in cui si trova, una situazione di furto unita alla vastità dell'edificio

elude il sistema di sicurezza poiché, nonostante esso sia in grado di rilevare l'evento, non può localizzarlo.

Sarebbe utile inoltre avere un registro di tutte le situazioni sospette, cosicché si possa sapere quali di queste sono state gestite o risolte e quali no. Il database deve pertanto registrare tutte le trap insieme ai dati che esse contengono: l'indirizzo dell'agent, la porta e il tipo di trap; ulteriori dati che potrebbero tornare utili, nonostante non siano contenuti direttamente nella trap, sono la data e l'ora in cui la trap è stata rilevata dal server. Grazie alla struttura del database, dalle informazioni della trap si può conoscere il luogo in cui è presente il dispositivo disconnesso.

Nelle specifiche di progetto si è posto l'accento sul fatto che le disconnessioni possono verificarsi in orari di lavoro e pertanto considerarsi lecite, poiché sono effettuate dal personale autorizzato per motivi noti. Questa possibilità conduce all'introduzione di uno stato da associare ad ogni dispositivo, in modo tale da sapere se esso è utilizzato da un operatore che sta lavorando. Considerando la regolarità dei turni di lavoro, ad ogni dispositivo può essere associato un orario di lavoro nel database, in modo tale che il sistema, interrogando il database ad intervalli regolari, modifichi lo stato dei dispositivi sulla base dei loro orari di lavoro. In termini pratici, quando un dispositivo si trova nello stato attivo o di lavoro, l'allarme non è inserito, mentre se è inattivo l'allarme viene innescato. Quando l'allarme è inserito per un dispositivo, tutte le segnalazioni dovute ad eventi relativi a quel dispositivo vengono trattate dal sistema come potenziali pericoli; se invece l'allarme è spento il sistema ignora le segnalazioni qualora dovessero giungere al manager. Per analogia si pensi al sistema anti-intrusione di un'abitazione: quando l'allarme è inserito l'apertura della porta di ingresso lo fa scattare, mentre se non lo è la porta può essere aperta e chiusa senza farlo scattare.

Infine nel database devono essere presenti gli elenchi degli utenti autorizzati all'uso del sistema con le relative password per consentirne l'autenticazione e dei ruoli che possono essere ricoperti.

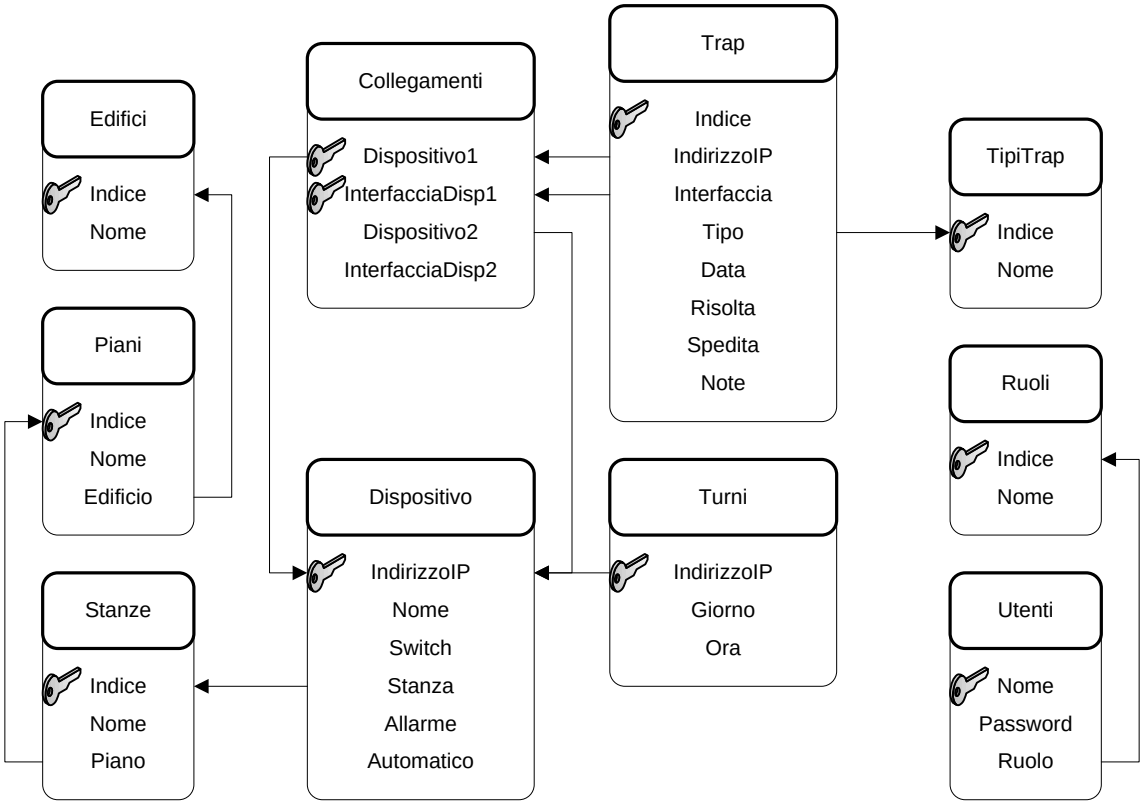

Figura 4.2: Schema del database

4.2 Struttura del sistema

Come mostrato nella figura 4.3, quando un host viene disconnesso dalla rete, lo switch ad esso collegato rileva l'evento e genera una trap da spedire attraverso la rete al server SNMP.

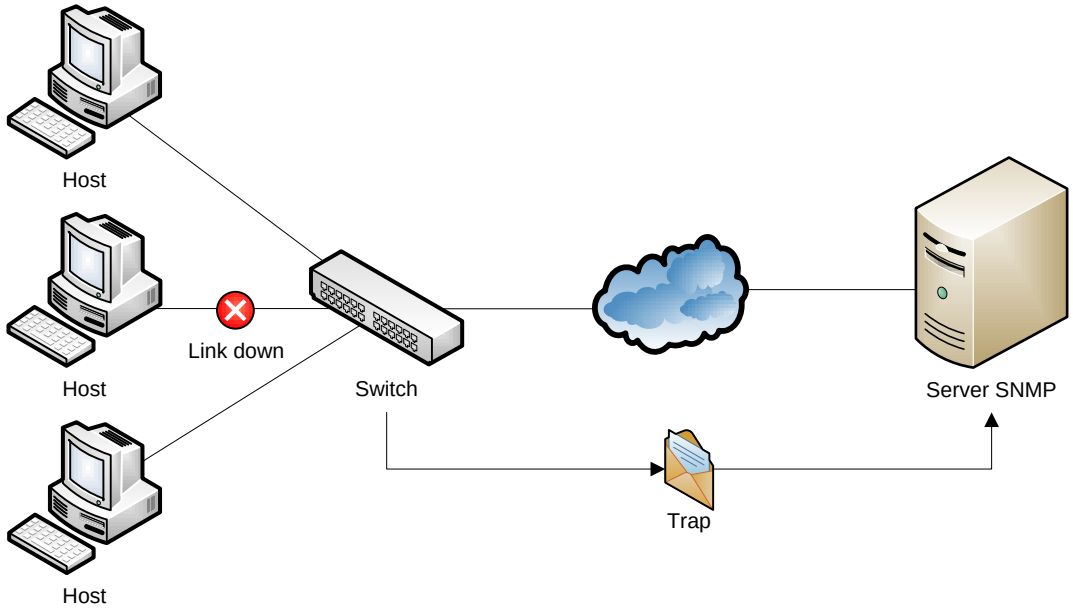

Figura 4.3: Generazione di una trap

Sul server SNMP deve essere eseguito l'applicativo snmptrapd che lo pone in ascolto, situazione nella quale è possibile rilevare le trap. Per il manager essere in ascolto si traduce nell'avere attivo un programma chiamato listener che è in grado di percepire l'arrivo di una nuova trap. Il file di configurazione snmptrapd.conf indica al listener come comportarsi quando giunge una nuova trap: nella fattispecie tale file comunica quale applicazione o quale insieme di comandi eseguire in funzione del tipo di trap giunta al listener. L'applicazione che il file di configurazione suggerisce e che viene eseguita dal listener subito dopo aver identificato il tipo di trap si chiama handler. Questo riceve dal listener le informazioni sulla trap, comunica con il database e produce l'avviso recante tutte le informazioni (evento, dispositivo coinvolto, luogo) da spedire all'amministratore, al quale deve arrivare sia attraverso il servizio e-mail che attraverso il servizio sms, in modo tale che possa ricevere la segnalazione in maniera tempestiva su due piattaforme distinte. Ad intervalli predefiniti crontab lancia un file eseguibile di aggiornamento (updater) che interviene sul database, attivando e disattivando gli allarmi in base ai turni di lavoro. La struttura ed il funzionamento del servizio nel server SNMP sono mostrati nella figura 4.4.

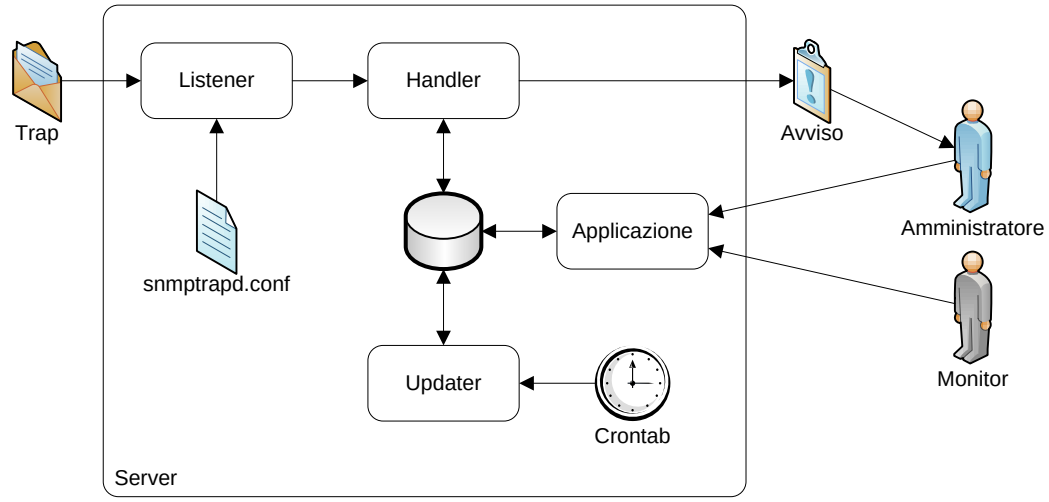

Figura 4.4: Server SNMP

Listener

Nello specifico il listener ha solamente il compito di rilevare le trap e inoltrarne la gestione ad un handler. Per il sistema il tipo di trap di interesse è quello denominato link down, il quale è associato a qualunque evento di disconnessione. Per avere una visione più globale della rete rientra nell'interesse del sistema anche il tipo di trap link up, dovuto alle azioni di connessione dei dispositivi. Nel file di configurazione snmptrapd.conf sono elencati i diversi tipi di trap e le azioni da intraprendere nel caso se ne rilevi una. Come si può notare dalla figura 4.5, la maggior parte dei tipi di trap, una volta identificati, non determinano alcuna azione; invece i tipi link down e link up pongono il listener nelle condizioni di lanciare un handler, il quale è lo stesso per entrambi i tipi. Nel file di configurazione quindi saranno presenti solamente due righe composte dal comando traphandle, dal tipo di trap (link down per la prima riga e link up per la seconda) e dal nome dell'handler (completo di percorso). Il listener, dopo aver dato risposta al quesito riguardante il riconoscimento del tipo di trap, esegue la riga del file di configurazione relativa al tipo rilevato, provocando così il lancio dell'handler.

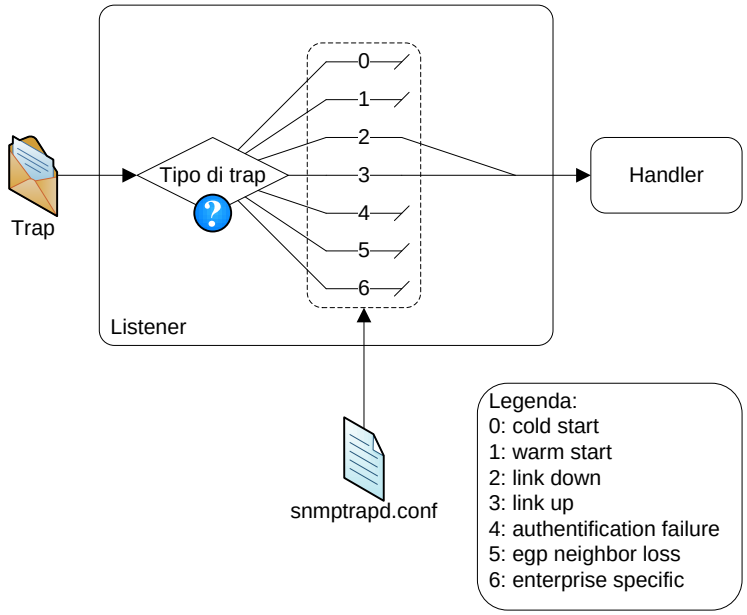

Figura 4.5: Listener

Handler

L'handler, come anticipato precedentemente, è un eseguibile lanciato dal listener che ha come compito principale la traduzione di un messaggio fortemente strutturato e sintetico come la trap in uno più discorsivo e dettagliato come l'avviso per l'amministratore, utilizzando il database come dizionario per convertire tutti i dati; in più lo stesso database viene aggiornato al fine di contenere una cronologia completa di tutti gli eventi rilevati. Una volta lanciato, l'handler come prima cosa legge i dati contenuti nella trap: il tipo di trap, l'indirizzo IP e la porta dello switch che l'ha generata. Questi tre dati insieme vengono passati al database in modo tale da aggiungere la nuova trap alla cronologia con tutti i dettagli essenziali. Quindi mediante la conoscenza delll'indirizzo IP e della porta dello switch, si interroga il database così da ottenere l'indirizzo IP del dispositivo (ed eventualmente il suo nome) connesso o disconnesso dalla rete e soprattutto il locale in cui esso si trova (dettagliato in edificio, piano e stanza). Queste due informazioni unite al tipo di trap permettono di compilare l'avviso preciso e completo da spedire all'amministratore. Ovviamente tale segnalazione viene generata e spedita solo se l'allarme relativo al dispositivo è attivo, altrimenti, pur mantenendo la trap nel database, non si prepara alcun avviso per l'amministratore. Nella figura 4.6 sono evidenziate in maniera semplificata le operazioni svolte dall'handler.

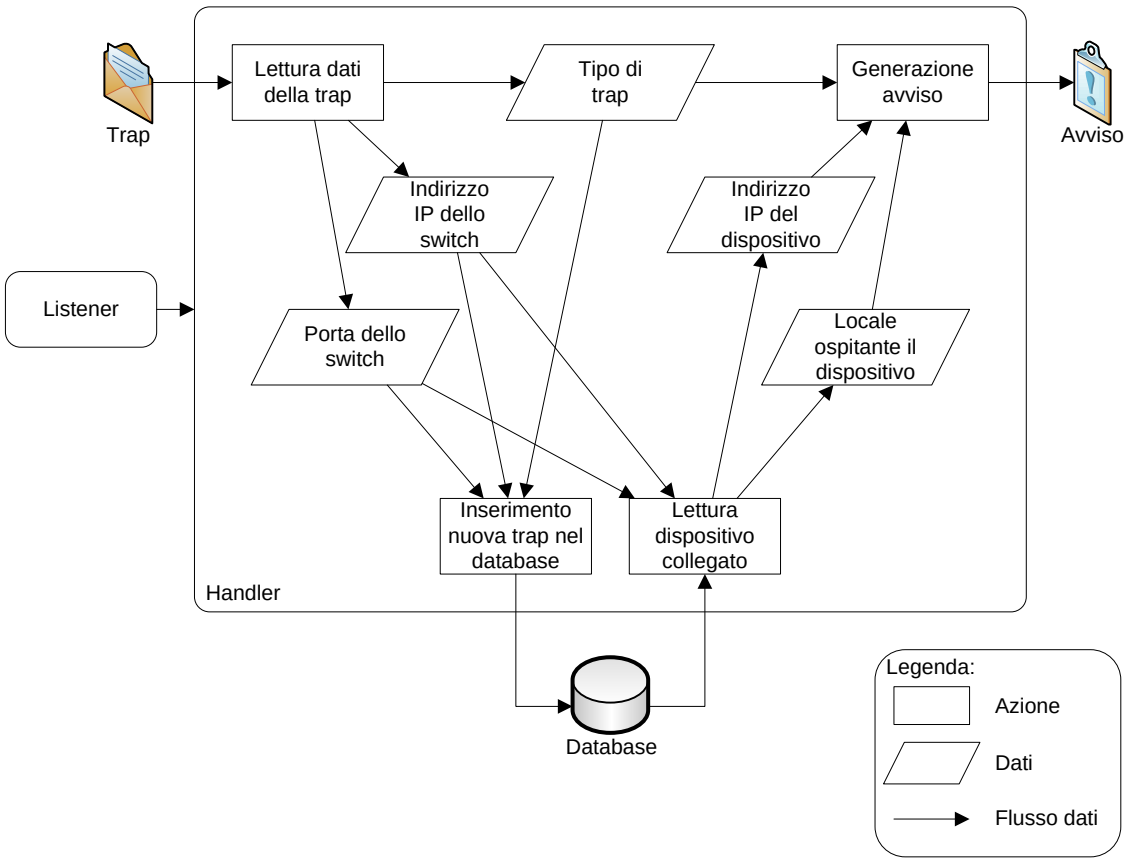

Figura 4.6: Handler

Updater

Per come è impostato crontab, questo ad ogni ora lancia l'updater, il quale come prima cosa rileva il giorno della settimana e l'ora attuali. Queste due informazioni servono per attivare l'allarme dei dispositivi che a quell'ora di quel giorno non sono utilizzati. Infatti nel database c'è una tabella che tiene traccia dei turni di lavoro di tutte le apparecchiature della rete; analizzando più in dettaglio tale tabella, ad ogni indirizzo IP sono associate più coppie giorno della settimana-ora, le quali indicano quando il dispositivo corrispondente a quell'indirizzo IP è in orario di lavoro, cioé quando l'allarme non deve essere inserito. Analogamente le postazioni che a quell'ora sono utilizzate subiscono la disattivazione dell'allarme.

L'updater aggiorna lo stato degli allarmi solamente per i dispositivi la cui gestione è automatica; l'amministratore può anche decidere di impostare manualmente gli allarmi, so-

prattutto per situazioni in cui non è prevista regolarità nei turni di lavoro. Per quanto riguarda gli switch il sistema predispone l'attivazione degli allarmi continuamente, a tutte le ore di tutti i giorni; tuttavia l'amministratore può comunque trattare in maniera alternativa il loro allarme passando alla gestione manuale.

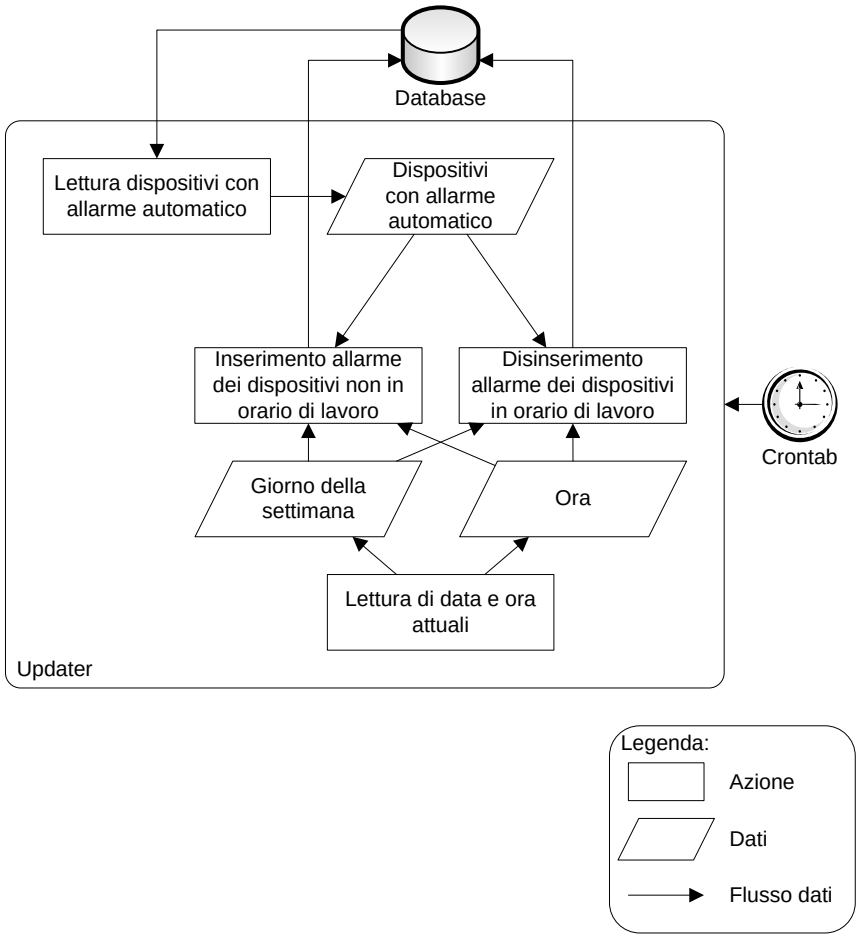

Figura 4.7: Updater

4.3 Progettazione dell'applicazione

L'applicazione ad interfaccia grafica consente all'operatore di interagire con il database affinché egli possa:

- osservare la cronologia delle trap,
- risolvere le trap,

- osservare la cronologia delle trap risolte,
- eliminare le trap risolte,
- gestire e osservare i locali (edifici, piani e stanze) in cui risiede la rete,
- gestire e osservare i dispositivi che compongono la rete e i collegamenti tra di loro,
- gestire e osservare i turni di lavoro e gli allarmi per i dispositivi,
- gestire gli utenti che possono utilizzare l'applicazione.

L'applicazione è stata realizzata come una web application, per cui può essere eseguita anche da una postazione remota, purché sia collegata alla rete locale. Innanzitutto l'applicazione richiede il riconoscimento e l'autenticazione dell'utente che avviene tramite l'inserimento di user name e password. L'applicazione riconosce due figure diverse di utente:

- amministratore,
- monitor.

Al primo sono permesse tutte le operazioni di gestione e controllo del sistema, mentre il secondo può solamente svolgere le operazioni di monitoraggio della rete. Nella figura 4.8 sono mostrate le funzionalità e le sottofunzionalità fornite dall'applicazione accessibili alle due diverse figure di utenti.

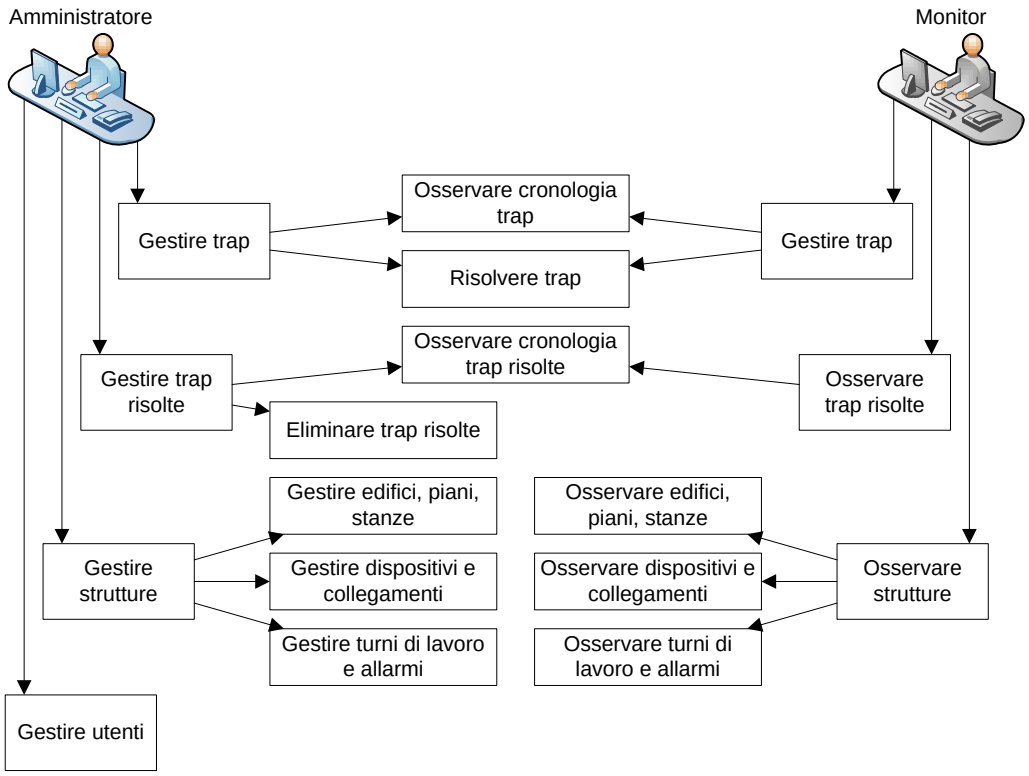

Figura 4.8: Struttura dell'applicazione

La sezione più importante dell'applicazione è rappresentata dalla cronologia delle trap, in cui si elencano tutti i collegamenti che hanno provocato un link down o un link up, ponendo in evidenza i due dispositivi che facevano capo a quel collegamento e il luogo in cui si trovano (Figura 4.9).

Trap	Edificio	Piano	Stanza	Agent / Dispositivo	Data	
	Torre nord	Piano terra	Sala server	Switch Gavia Systems, porta 1	2009-10-26 02:30:00	Risolvi
	Torre nord	Piano terra	Segreteria	PC Segreteria 1, porta 1		

Figura 4.9: Esempio di trap link down nella cronologia

L'utente che per primo vede una nuova segnalazione nella cronologia ha il compito di capire tempestivamente chi o che cosa ha provocato tale evento; la segnalazione giunge anche all'amministratore mediante le modalità precedentemente descritte (e-mail ed sms). Una volta che si è scoperta la causa che ha provocato la trap, la segnalazione può essere

marcata come risolta e quindi spostata tra le trap risolte (una sorta di cestino), aggiungendo delle note opportune, relative per esempio alla causa stessa. In un secondo momento l'amministratore può eliminare completamente dal database le trap risolte.

L'applicazione prevede diverse funzioni di contorno che servono a mantenere il sistema aggiornato alla situazione attuale affinché le segnalazioni delle trap siano corrette e affidabili; tali funzioni devono essere svolte dall'amministratore. La prima di queste è la gestione delle strutture in cui sono posizionate le apparecchiature informatiche. Infatti l'amministratore deve registrare nel database gli edifici che compongono il complesso, i piani presenti e a loro volta le stanza relative ad ogni piano. Questa operazione viene effettuata nei primi utilizzi dell'applicazione, poiché solitamente la struttura ambientale non cambia o al limite cambia poco. Come conseguenza di questa fase è stato ricostruito l'insieme dei luoghi in cui sono collocati i dispositivi. A questo punto l'amministratore, attraverso un'altra funzione dell'applicazione, deve creare nel database la struttura della rete, cioé deve registrare tutti i dispositivi associandoli ai locali in cui si trovano e tutti i collegamenti tra di essi. Anche questa operazione viene svolta nei primi utilizzi, immaginando rari cambiamenti nella topologia della rete (spostamento di un computer da una stanza ad un'altra oppure il cambio di collegamento da una porta ad un'altra di uno switch).

Dopo aver ricreato le situazioni ambientale e topologica della rete, l'amministratore deve fissare gli orari di lavoro per ogni postazione; questa operazione è molto importante perché corrisponde alla pianificazione degli allarmi che si attiveranno automaticamente ogni volta che un dispositivo deve essere controllato, ovvero quando non è più utilizzato da alcun operatore e si ritrova ad essere "incustodito". Solitamente i turni di lavoro assumono una certa regolarità e periodicità, tuttavia l'amministratore deve comunque agire nel modificarli quando sono previsti turni straordinari, manutenzioni sulle apparecchiature o altri casi particolari, passando alla gestione manuale che gli permette di ignorare gli orari prestabiliti. Nella figura 4.10 è mostrato l'esempio della pianificazione del turno di lavoro per un dispositivo.

Un discorso diverso meritano i dispositivi che in una rete sono in funzione per la maggior parte del tempo anche in assenza del personale (switch, router, server); questi dispositivi non dovrebbero contemplare gli eventi di link down e link up, nemmeno durante il loro funzionamento, anche perché nel qual caso potrebbero verificarsi l'isolamento di una parte della rete oppure l'assenza di un servizio importante. Quindi su tali dispositivi si predispone l'attivazione dell'allarme per tutti i giorni e a tutte le ore.

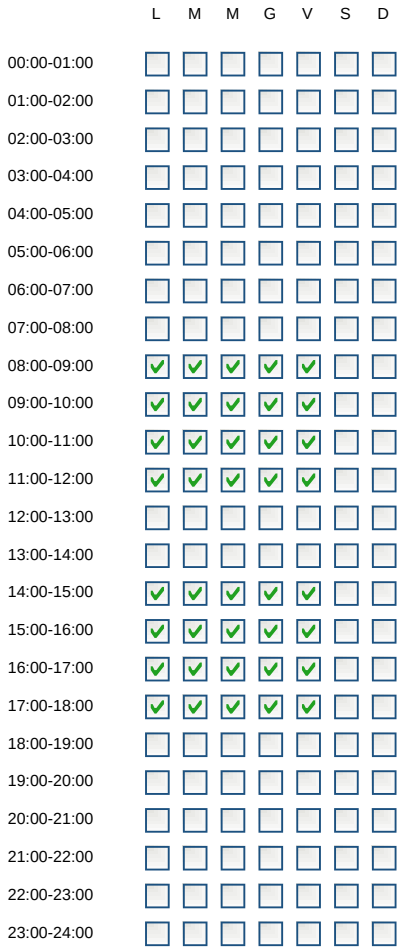

Figura 4.10: Esempio di pianificazione dell'orario di lavoro

Un'ulteriore funzione accessibile solamente all'amministratore riguarda la gestione degli utenti che possono utilizzare l'applicazione. Si possono registrare nuovi utenti e associare ad ognuno di essi un ruolo (amministratore o monitor). Quando uno di questi accederà all'applicazione, mediante i propri user name e password, sarà abilitato a svolgere le funzioni consentite al ruolo che occupa.

Capitolo 5

Conclusione

Il prototipo è stato sviluppato seguendo il piano di progetto previsto. Ricordando le tre iterazioni principali eseguite in successione, di volta in volta si è ampliato il dominio di rete a cui applicare il sistema. Nel primo passo è stato testato il sistema considerando un solo computer, il quale svolgeva il ruolo del manager e allo stesso tempo simulava i vari agent, grazie alla funzionalità della suite Net-SNMP (in particolare il comando snmptrap). Dopo aver popolato il database in modo tale da ricostruire l'ambiente e la topologia di una rete immaginaria, sono state simulate alcune trap. Il sistema rispondeva in tempi rapidissimi e soprattutto in maniera corretta: la cronologia veniva aggiornata immediatamente e all'amministratore giungevano con tempestività gli avvisi relativi alle trap, sia attraverso il servizio sms che attraverso quello e-mail. Anche l'aggiornamento degli allarmi si è mostrato affidabile; sebbene i tempi di aggiornamente risultino ottimi con una rete di piccole dimensioni, diventa interessante e cruciale capire se le prestazioni rimangano accettabili anche all'aumentare del numero di dispositivi componenti la rete. Ricordando che c'è un allarme per ogni dispositivo, più sono i dispositivi e più sono gli allarmi da aggiornare con conseguente aumento del tempo necessario per le modifiche. Pertanto è stata ampliata la rete immaginaria inserendo molto dispositivi nel database al fine di poter verificare l'efficienza del processo di aggiornamento degli allarmi in condizioni di carico simili, se non addirittura superiori, a quelle di una rete molto estesa. I risultati hanno mostrato che nonostante i molti dati da aggiornare, il tempo necessario per tale processo è più che accettabile, garantendo così l'aggiornamento regolare del sistema.

Nella seconda iterazione si è passati da una rete simulata ad una rete reale, ovvero quella dell'azienda in cui è stato sviluppato il sistema, la quale ha delle dimensioni contenute contando pochi dispositivi. Il database ovviamente è stato popolato secondo la

struttura dello stabile e la topologia della rete; gli orari di lavoro e quindi gli allarmi sono stati impostati in modo tale da poter effettuare tutti i casi di prova possibili. Innanzitutto la prima verifica da effettuare riguardava la reazione di uno switch ad un evento di disconnessione, reazione che è confluita nella generazione di una trap e nella sua spedizione al server. In questo caso i tempi di risposta sono leggermente più alti ma accettabili (pochissimi millisecondi), tanto che all'occhio umano è impercettibile il ritardo dovuto al tempo di rilevamento dell'evento e al tempo di trasmissione della trap dallo switch al ser-ver. A seconda della casa costruttrice ogni switch aggiunge nelle trap delle informazioni non standard che non sono utilizzate dal sistema; sebbene le trap risultino più cariche di dati, la dimensione in termini di byte aumenta di poco, al punto tale che non ci sono ritardi nella propagazione attraverso la rete. Purtroppo nel caso di due disconnessioni avvenute contemporaneamente, gli switch hanno generato la prima trap all'istante, mentre la seconda solo dopo un breve intervallo di ritardo. Questa caratteristica non provoca problemi in quanto il furto contemporaneo di due dispositivi posti in due stanze molto distanti può essere gestito comunque in modo efficace dato il ritardo contenuto della seconda segnalazione. Poiché la rilevazione di questo problema è avvenuta nella fase di test su una rete reale e non su quella simulata, si deduce che tale caratteristica è legata agli switch oppure agli agent che risiedono in essi. I vari casi di prova hanno evidenziato che l'accensione e lo spegnimento di un computer provocano una breve interruzione del segna-le tra la scheda di rete del computer e la porta dello switch, che viene interpretata dallo switch stesso come un link down seguito immediamtamente da un link up; ovviamente questi due eventi causano la generazione di altrettante trap da parte dello switch. Poiché l'accensione e lo spegnimento di un computer sono effettuate dall'operatore addetto a tale postazione, è naturale che il tutto avvenga durante il turno di lavoro fissato per il dispositivo stesso e quindi le segnalazioni, nonostante giungano al server, vengono ignorate. È risultato che anche un reset ha come conseguenza la creazione di due trap, una di tipo link down e una di tipo link up; infatti il reset è interpretato come una accensione e pertanto provoca anch'esso una breve interruzione di segnale tra il computer e lo switch. Quindi durante un turno di lavoro viene confermata la necessità di disattivare l'allarme al fine di ignorare tutte le segnalazioni dovute a operazioni di accensione, spegnimento e reset di un computer. Così si alleggerisce il compito nell'interpretazione delle trap da parte dell'utente che sta monitorando la rete, poiché alcune segnalazioni sono direttamente ignorate dal sistema.

Alcuni dispositivi come ad esempio gli hub non sono dotati di alcun agent SNMP; a con-

ferma di questa caratteristica vi è la situazione in cui è stato disconnesso un computer collegato ad un hub a sua volta facente capo ad uno switch; quest'ultimo non ha rilevato l'evento e quindi non ha generato alcuna trap; effettivamente lo switch in questa situazione non può generare trap, poiché alla sua porta continua ad essere collegato l'hub. Tutti i casi presentati evidenziano l'importanza delle apparecchiature utilizzate (gli switch sono più adatti degli hub) e della pianificazione degli orari (turni di lavoro regolari consentono una gestione più automatizzata e quindi meno onerosa per l'utente).

Nella terza ed ultima iterazione si è passati da una rete reale piccola ad una molto più grande. Sono stati riproposti gli stessi casi di prova per verificare il funzionamento del sistema, il quale risponde in modo analogo al caso relativo alla rete con un numero inferiore di apparecchiature. Pertanto la dimensione della rete comporta solamente un maggior dispendio di risorse in termini di tempo per predisporre il database al funzionamento del sistema, cioé per inserire tutti i locali, i dispositivi, i collegamenti e i turni di lavoro.

In conclusione il prototipo del sistema si è dimostrato affidabile, tempestivo, semplice da impostare e da utilizzare.

Elenco delle figure

Bibliografia

[1] “Reti di calcolatori” di Andrew S. Tanenbaum, Prentice Hall, IV edizione, 2003

[2] “Reti di calcolatori” di Larry R. Peterson e Bruce S. Davie, Apogeo, III edizione, 2003

[3] ”Essential SNMP” di Douglas Mauro e Kevin Schmidt, O’Reilly, 2001

Siti consultati

[4] http://www.net-snmp.org/