



Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Progettazione e Realizzazione di un Intrusion Management System

Rapporto Tecnico N.:
RT-ICAR-PA-10-02

luglio 2010



Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR)
– Sede di Cosenza, Via P. Bucci 41C, 87036 Rende, Italy, URL: www.icar.cnr.it
– Sede di Napoli, Via P. Castellino 111, 80131 Napoli, URL: www.na.icar.cnr.it
– Sede di Palermo, Viale delle Scienze, 90128 Palermo, URL: www.pa.icar.cnr.it



Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Progettazione e Realizzazione di un Intrusion Management System

P. Storniolo¹, A. Messina²

Rapporto Tecnico N.:
RT-ICAR-PA-10-02

Data:
luglio 2010

¹ Istituto di Calcolo e Reti ad Alte Prestazioni, ICAR-CNR, Sede di Palermo
Viale delle Scienze edificio 11 90128 Palermo

² Università degli Studi di Palermo, Dipartimento di Ingegneria Informatica
Viale delle Scienze 90128 Palermo

I rapporti tecnici dell'ICAR-CNR sono pubblicati dall'Istituto di Calcolo e Reti ad Alte Prestazioni del Consiglio Nazionale delle Ricerche. Tali rapporti, approntati sotto l'esclusiva responsabilità scientifica degli autori, descrivono attività di ricerca del personale e dei collaboratori dell'ICAR, in alcuni casi in un formato preliminare prima della pubblicazione definitiva in altra sede.

Sommario

1	INTRODUZIONE	5
2	SYSTEM MANAGEMENT	8
2.1	Introduzione	8
2.2	Tool di System Management	9
2.3	Scelta dei Tool di System Management	10
2.4	Soluzioni di Management Open Source	10
2.5	Criteria per la scelta di un tool di Management Open Source	11
2.5.1	Requisiti obbligatori	12
2.5.2	Requisiti desiderabili	12
2.6	Definizione di Network and Systems Management	12
2.6.1	Network Management	12
2.6.2	System Management	13
2.7	Confronto tra Nagios, OpenNMS, Zenoss e Zabbix	14
2.7.1	Confronto delle funzionalità	15
2.7.2	Pro e Contro	17
2.8	Scelta del Sistema di Management	18
2.9	Zabbix	19
2.9.1	Discovery	19
2.9.2	Availability Monitoring	21
2.9.3	Problem Management	22
2.9.4	Flusso delle informazioni	23
3	INTRUSION MANAGEMENT SYSTEM CON ZABBIX, SNORT ED OSSEC	24
3.1	Introduzione	24
3.2	Intrusion Detection Systems	24
3.2.1	HIDS vs NIDS	26
3.3	Architettura	28
3.4	Componenti del sistema	30
3.4.1	Net-SNMP	30
3.4.2	Snort	30
3.4.3	Il plugin Snort-SNMP	33
3.4.4	OSSEC	34
3.4.5	SNMP Trap Translator	37

3.4.6	Gli script zsw e zow	40
3.5	Estensioni al Frontend di Zabbix	40
3.5.1	Monitoring -> Dashboard	42
3.5.2	Monitoring -> IDS Events e Reports -> Most Busy IDS triggers top 100	43
3.5.3	Configuration -> Import	44
3.5.4	Configuration -> SNMP Builder	46
4	CONCLUSIONI	48
4.1	Introduzione	48
4.2	Ulteriori possibili sviluppi della piattaforma IDM	48
4.3	Applicabilità della soluzione ad altri ambiti operativi	50
5	BIBLIOGRAFIA	51

1 INTRODUZIONE

Seguire i continui sviluppi del mercato ICT porta le aziende ad avere la necessità di dotarsi di sistemi hardware e software sempre più all'avanguardia, per supportare e gestire i propri sistemi informativi computerizzati. Di conseguenza, è in continuo aumento anche la complessità del sistema informativo delle aziende, indipendentemente dalle dimensioni che essa assume nel proprio mercato di riferimento.

Assistiamo, anno dopo anno, alla progressiva introduzione di nuovi apparati o soluzioni software per soddisfare esigenze crescenti e per agevolare comunicazione e rapporti verso clienti e fornitori.

Apparati di rete quali switch, router, firewall, server di dominio, posta elettronica e software applicativi sono diventati elementi chiave per ottimizzare la riuscita di ogni business.

Purtroppo però, spesso accade che l'amministratore dei sistemi informativi rilevi anomalie, guasti o disservizi solo al momento della segnalazione da parte degli utenti (dipendenti, collaboratori interni ed esterni, business partner, ecc): tutto questo può provocare disagio e ingenti rallentamenti nel lavoro, costituendo un costo "imprevisto" per l'azienda.

Poter disporre, quindi, in tempo reale di una visione d'insieme delle risorse aziendali e del loro stato, permette di intervenire in modo mirato in caso di anomalie o indisponibilità delle stesse, prevenendo in certi casi anche la stessa indisponibilità.

Le moderne aziende (grandi e piccole, con una o più sedi) hanno quindi la necessità di dotarsi di un sistema di monitoraggio per verificare "lo stato di salute" dei propri sistemi informativi.

In alcuni casi, invece, può verificarsi che le aziende dispongano di più sistemi di monitoraggio per ogni tipologia di sistema informatico integrato nella rete aziendale, e ciò comporta una frammentazione dei report dello stato di salute di tutti gli apparati e servizi, con la conseguente impossibilità di monitorare in maniera globale il funzionamento dell'infrastruttura complessiva dell'azienda.

E' quindi auspicabile poter disporre di una soluzione che migliori l'accessibilità di monitoraggio, concentrando i rilevamenti dello status dei diversi componenti del sistema informativo nell'ambito di un'unica centrale di controllo, che possa, secondo i casi, anche fungere da sistema di reazione, oltre che di informazione.

Se adesso, per la misurazione e il controllo dei dispositivi si utilizza un protocollo standard, che abbia la *interoperabilità* come sua caratteristica principale e che sia, inoltre, anche *espandibile*, si può facilmente intuire come sia possibile effettuare il monitoraggio non solo di apparati/sistemi (router, switch, firewall, server, ecc.) e servizi applicativi (web server, mail server, database server, ecc.) tipicamente afferenti al mondo ICT, ma anche di qualunque altro dispositivo o servizio che sia "accessibile" da un'infrastruttura di rete informatica.

Un esempio di protocollo dotato di tali caratteristiche è il *Simple Network Management Protocol* (SNMP).

L'applicabilità di un tale tipo di soluzione potrebbe quindi estendersi anche ad ambiti applicativi diversi da quello ICT in senso stretto: è sufficiente, infatti, che in questi altri ambiti vi sia una qualche possibilità d'interfacciamento con il "mondo" ICT.

Nel campo della Domotica e Building Automation, ad esempio, sono disponibili dei convertitori studiati espressamente per interconnettere tra loro i più diffusi bus utilizzati in quell'ambito¹, consentendo quindi l'integrazione di mondi eterogenei quali climatizzazione, automazione, illuminazione, sistemi di misura, sistemi antincendio e sistemi di antintrusione. Essendo dotati di porte ethernet, tali convertitori possono anche essere connessi a sistemi di controllo e monitoraggio, con i quali colloquieranno tramite protocollo SNMP.

O ancora, ad esempio, esistono sul mercato dei dispositivi per il Monitoring Ambientale, a basso costo e basati su SNMP, che possono essere configurati per prevenire specifici casi di esposizione all'umidità, allagamenti, gas, vento, alte/basse temperature, sbalzi di tensione, movimento, suono.

A prescindere, quindi, dal particolare ambito, la piattaforma di monitoraggio sarà sempre e comunque costituita da un server centrale (eventualmente ridondato) cui sono collegate delle *sonde*, le quali raccolgono le informazioni dai vari apparati che compongono l'infrastruttura in oggetto.

Alla luce di tale generalizzazione e intendendo per *servizio* ciascun dispositivo o componente dell'infrastruttura, si potrà parlare di *Service Management*.

Scopo del presente lavoro sarà di verificare la fattibilità dell'integrazione di sistemi di acquisizione delle informazioni con sistemi di elaborazione e presentazione, attraverso l'uso di un protocollo comune.

In particolare, si estenderanno le funzionalità di base della piattaforma di management Open Source denominata Zabbix, che nasce come "semplice" Network Management System (NMS) di classe enterprise, integrandovi nuove funzionalità di *Intrusion Detection Management* (si potrebbe in definitiva anche parlare, generalizzando gli ambiti applicativi, di *Event Detection & Management*).

Si utilizzeranno come "sonde" specializzate i due più diffusi e potenti strumenti Open Source per l'intrusion detection oggi disponibili, Snort ed OSSEC, appartenenti rispettivamente alle tipologie di Network-based Intrusion Detection System (NIDS) e Host-based Intrusion Detection System (HIDS).

¹ EIB-Konnex, LON, BACnet, ModBus, ed altri ancora

Si dimostrerà quindi come sia possibile trasferire, grazie al protocollo SNMP, gli alert di sicurezza generati da tali software eterogenei ed integrarli nell'ambito di un'unica piattaforma di management e controllo, potendo anche associare, infine, delle azioni/reazioni personalizzate al verificarsi di un particolare evento o tipologia di eventi.

2 SYSTEM MANAGEMENT

2.1 INTRODUZIONE

Ognuno di noi, così come qualunque organizzazione, ha il proprio punto di vista sui requisiti che debba soddisfare un sistema di System Management; quindi, il primo essenziale passo, quando si ha intenzione di adottarne uno, è quello di definire quali siano tali requisiti. Ciò dà l'idea di come poter misurare il successo di un progetto.

Esistono molte diverse metodologie e discipline per il System Management, a partire dall'acronimo "FCAPS" (Fault, Configuration, Accounting, Performance, Security) coniato dall'International Standards Organization, fino ad arrivare all'Information Technology Infrastructure Library (ITIL) che propone il framework ITIL V2 suddiviso in due categorie:

- Service Support, che include:
 - funzione di Service Desk;
 - processo di Incident management;
 - processo di Problem management;
 - processo di Configuration management;
 - processo di Change management;
 - processo di Release management.

- Service Delivery, che include:
 - processo di Service Level management;
 - processo di Capacity management;
 - processo di IT Service Continuity management;
 - processo di Availability management;
 - Financial management per i servizi IT.

Elemento chiave dell'intero framework ITIL è il concetto di Configuration Management Database (CMDB), dentro il quale memorizzare e mantenere i Configuration Items (CI) e le loro inter-relazioni.

L'arte del System Management consiste nel definire cosa è importante, cioè "in ambito", e, aspetto forse ancora più significativo, cosa è effettivamente "fuori ambito".

Scopo di un ambiente di System Management è quello di fornire dati in modo efficace, accurato ed affidabile al fine di soddisfare i requisiti di base.

Un tool di System Management che possa fornire migliaia di metriche già pronte ma che sia anche poco affidabile e/o non facilmente configurabile, è semplicemente sinonimo di un progetto implementato male e magari anche con costi eccessivi.

Per progetti più piccoli ed organizzazioni di tipo PMI (SMB: Small/Medium Business), è spesso utile un approccio più pragmatico, consistente nell'applicazione combinata dell'applicazione dei requisiti in modalità top-down, unito all'approccio di tipo bottom-up effettuato a partire dalle migliori "10 metriche" che possono essere facilmente erogate da un tool; tale approccio si traduce in un processo iterativo che conduce abbastanza velocemente ad un prototipo di soluzione.

2.2 TOOL DI SYSTEM MANAGEMENT

Non esistono soluzioni di System Management perfette. La riuscita dell'implementazione dei requisiti di un System Management è data, infatti, dalla combinazione di più fattori:

- Definizione appropriata dei requisiti;
- Tools adatti;
- Competenze nella traduzione dei requisiti in personalizzazione dei tools;
- Project management;
- User training;
- Documentazione.

In teoria, la scelta del tool dovrebbe essere guidata dai requisiti. In pratica, invece, questo spesso non accade e una soluzione valida per uno solo degli aspetti del System Management in un'area di business può diventare lo standard de facto per l'intera organizzazione.

Esistono diverse buone ragioni perché questo capiti.

Non è pratico, ad esempio, utilizzare un servizio di Service Desk centralizzato con una miriade di tool diversi. Un tool basato su framework con un database centralizzato, un adeguato look-and-feel per entrambe la Graphical User Interface (GUI) e la Command Line Interface (CLI), dotato di moduli che coprano i diversi ambiti del System Management, è una soluzione molto più conveniente rispetto all'utilizzo sconsiderato di tool distinti a coprire ambiti distinti, specialmente quando devono essere presi in particolare considerazione i costi di implementazione, di mantenimento delle competenze e di addestramento degli utenti.

L'integrazione è il fattore fondamentale dell'adozione di un ambiente di System Management e l'elemento chiave di esso è dato dal concetto di singolo Configuration Management Database (CMDB) che tutti i tool vanno ad alimentare ed utilizzare.

Un buon tool fornisce facilmente strumenti immediatamente utilizzabili ed è dotato di modalità standard per le successive fasi di personalizzazione.

Lo spettro per la scelta dei tool ha quindi come base di partenza la presenza di un compilatore o interprete (C, bash, ...) e la personalizzazione consiste nello scrivere programmi ex-novo.

Nella parte finale, complessa, di tale spettro il tool sarà invece costituito da varie suite di moduli forniti da uno dei quattro maggiori produttori commerciali, IBM, HP, CA e BMC.

2.3 SCELTA DEI TOOL DI SYSTEM MANAGEMENT

Ogni organizzazione ha priorità diverse riguardanti i criteri di selezione di un tool.

Tralasciando al momento le metriche di carattere prettamente tecnico, alcuni degli altri fattori di decisione sono:

- Facilità d'uso;
- Competenze necessarie per implementare i requisiti, rispetto alle competenze effettivamente disponibili;
- Necessità e disponibilità per il training degli utenti;
- Costo complessivo, dato da acquisto delle licenze, tempo di valutazione, mantenimento, training, ecc...;
- Assistenza, dal fornitore e/o comunità di utenti;
- Scalabilità;
- Semplicità di esercizio, in termini di facilità d'installazione del server di management e di distribuzione ed installazione degli agent;
- Affidabilità;
- Responsabilità, cioè la possibilità di rivalersi sul vendor in caso di danni/guasti.

Se quella della “Responsabilità” è tra le priorità più elevate e invece conta relativamente molto meno il costo del software, allora è preferibile scegliere una tra le alternative commerciali.

Se si ha invece a disposizione uno staff di lavoro ben competente, o comunque ben preparato e pronto ad imparare velocemente, e il costo complessivo è un fattore limitante, allora le soluzioni Open Source sono sicuramente meritevoli di attenzione.

L'aspetto in generale interessante è che è possibile trovare offerte che soddisfino tutti i fattori su elencati sia in ambito commerciale sia in ambito Open Source.

2.4 SOLUZIONI DI MANAGEMENT OPEN SOURCE

Vi sono una miriade di diverse soluzioni di management Open Source e, ovviamente, molte di esse si basano sul protocollo SNMP. Tra queste, alcune sono sicuramente delle ottime soluzioni per specifici requisiti di nicchia.

MRTG (Multi Router Traffic Grapher), scritto da Toby Oetiker, è un esempio eccellente di applicazione compatta che utilizza il protocollo SNMP per collezionare e memorizzare informazioni di performance e per visualizzarle graficamente. Se già questo soddisfa le proprie necessità, non è necessario considerare altre eventuali alternative, ma si tenga presente che lo strumento non sarà molto di aiuto nella considerazione di problemi provenienti da dispositivi diversi e nella gestione di tali problemi fino alla loro risoluzione.

Un'estensione di MRTG è RRDTool (Round Robin Database Tool), sempre di Tobi Oetiker. E' sempre fondamentalmente uno strumento orientato alla raccolta periodica ed alla visualizzazione dei dati numerici di performance, ma si basa su database. La dimensione del database è comunque predeterminata al momento della creazione e i nuovi dati vanno a sovrascrivere i vecchi dopo un intervallo di tempo prefissato.

Un'ulteriore estensione di RRDTool è Cacti, che fornisce un frontend completo alle funzionalità di RRDTool. Con Cacti è possibile usare un database relazionale MySQL come back-end a monte dei database Round Robin.

Le sorgenti dati, inoltre, possono essere costituite anche da script, oltre che ovviamente SNMP, e sono incluse funzionalità di gestione utenti. Cacti rimane comunque uno strumento per il collezionamento e la visualizzazione di dati di performance, e non una soluzione completa, basata su frame work, per il system management.

Salendo la scala delle funzionalità e complessità, troviamo alcune soluzioni più orientate al network management (netdisco, The Dude), ed altre invece più orientate al system management (Nagios, Zabbix).

Le soluzioni più complete e di più alto livello presentano un'architettura basata su database centralizzato (Nagios, Zenoss, OpenNMS, Zabbix).

Infine, alcuni sono progetti totalmente Open Source, tipicamente rilasciati sotto la licenza Gnu GPL (MRTG, RRDTool, Cacti) o la licenza BSD (netdisco), mentre alcuni sono disponibili gratuitamente (tipicamente in GPL) ma presentano estensioni con licenza commerciale (Zenoss).

In aggiunta alle licenze free, per diversi prodotti sono anche disponibili contratti di supporto a pagamento (Zenoss, Nagios, OpenNMS, Zabbix).

2.5 CRITERI PER LA SCELTA DI UN TOOL DI MANAGEMENT OPEN SOURCE

Per un progetto di system management è essenziale definire cosa è “ambito” e cosa è “fuori ambito”.

E' quindi utile considerare una lista prioritizzata di requisiti obbligatori e desiderabili.

2.5.1 *REQUISITI OBBLIGATORI*

- Software Open Source;
- Alta attività nei forum/ mailing list collegate;
- Rilasci regolari di bug-fix e nuove release;
- Management integrato di reti e sistemi, comprese le gestioni di:
 - Configurazioni;
 - Disponibilità;
 - Problemi;
 - Performance.
- Database open centralizzato;
- Disponibilità di ambiente GUI e CLI;
- Installazione semplificata degli agent;
- Scalabilità a diverse centinaia di dispositivi;
- Documentazione adeguata.

2.5.2 *REQUISITI DESIDERABILI*

- Supporto a SNMP V3;
- Gestione degli utenti e dei loro privilegi di accesso;
- Rappresentazione grafica degli schemi di rete;
- Accesso remoto ai dispositivi rilevati;
- Nessuna richiesta di web browser proprietari;
- Scalabilità a diverse migliaia di dispositivi;
- Buona documentazione;
- Disponibilità di supporto a pagamento.

2.6 DEFINIZIONE DI NETWORK AND SYSTEMS MANAGEMENT

La locuzione “Management integrato di reti e sistemi” necessita di ulteriore approfondimento.

2.6.1 *NETWORK MANAGEMENT*

- Configurazione
 - Discovery automatico, controllabile, di dispositivi di rete layer 3;
 - Visualizzazione della topologia dei dispositivi;
 - Supporto di SNMP V1, V2 e preferibilmente V3;
 - Discovery dei dispositivi che non supportano il ping;

- Discovery dei dispositivi che non supportano SNMP;
- Database centralizzato per la memorizzazione delle informazioni dei dispositivi.
- Monitoraggio della Disponibilità
 - Ping test personalizzabile per tutti i dispositivi rilevati;
 - Test di disponibilità per i dispositivi che non rispondono al ping (ad es. confronto tra gli stati SNMP administrative e operational Interface);
 - Visualizzazione semplificata dello stato di disponibilità dei dispositivi, preferibilmente sia in forma tabellare che grafica;
 - Generazione di eventi quando un dispositivo fallisce il test di disponibilità;
 - Possibilità di monitorare lo stato interno dei dispositivi di rete (CPU, memoria, ventole);
 - Differenziazione tra i problemi di dispositivo/interfaccia down e rete non raggiungibile.
- Gestione dei Problemi
 - Eventi configurabili per ogni dispositivo rilevato;
 - Console centralizzata degli eventi con possibilità di assegnazione di priorità;
 - Possibilità di categorizzare gli eventi per la loro visualizzazione solo ad utenti specifici;
 - Possibilità di ricezione di trap con i protocolli SNMP V1, V2 e preferibilmente V3;
 - Personalizzazione delle azioni di risposta agli eventi.
- Performance
 - Monitoraggio regolare e personalizzabile delle variabili MIB SNMP, sia standard che enterprise, con memorizzazione dei valori ed impostazione di soglie per la generazione degli eventi;
 - Possibilità di importare qualunque MIB;
 - Possibilità di eseguire il browse delle MIB su qualunque dispositivo;
 - Grafici personalizzati dei dati di performance.

2.6.2 *SYSTEM MANAGEMENT*

Molti dei criteri per il system management sono simili a quelli relativi al network management, ma sono comunque riproposti di seguito per comodità.

- Configurazione
 - Discovery automatico, controllabile, di sistemi Windows e Linux;
 - Visualizzazione della topologia dei sistemi;
 - Supporto di SNMP V1, V2 e preferibilmente V3;
 - Discovery dei sistemi che non supportano il ping;
 - Discovery dei sistemi che non supportano SNMP;
 - Database centralizzato per la memorizzazione delle informazioni dei sistemi.

- Monitoraggio della Disponibilità
 - Ping test personalizzabile per tutti i sistemi rilevati;
 - Test di disponibilità per i sistemi che non rispondono al ping (ad es. confronto tra gli stati SNMP administrative e operational Interface, supporto per test ssh);
 - Possibilità di monitorare servizi su porte a piacere (ad es. tcp/80 per server http);
 - Possibilità di monitorare “applicazioni” (ad es. accesso ssh/snmp per il monitoraggio dei processi, wget per il prelievo di pagine web);
 - Visualizzazione semplificata dello stato di disponibilità dei sistemi, preferibilmente sia in forma tabellare che grafica;
 - Generazione di eventi quando un sistema fallisce il test di disponibilità;
 - Possibilità di monitorare le metriche di base di un sistema, come CPU, memoria, ventole, spazio su disco, processi, servizi (ad es. il MIB SNMP Host Resource);
- Gestione dei Problemi
 - Eventi configurabili per ogni sistema rilevato;
 - Console centralizzata degli eventi con possibilità di assegnazione di priorità;
 - Possibilità di categorizzare gli eventi per la loro visualizzazione solo ad utenti specifici;
 - Possibilità di ricezione di trap con i protocolli SNMP V1, V2 e preferibilmente V3;
 - Possibilità di monitorare i syslog Unix e l’Event Log di Windows e di generare eventi personalizzati;
 - Possibilità di monitorare qualunque file di log e di generare eventi personalizzati;
 - Personalizzazione delle azioni di risposta agli eventi.
- Performance
 - Monitoraggio regolare e personalizzabile delle variabili MIB SNMP, sia standard che enterprise, con memorizzazione dei valori ed impostazione di soglie per la generazione degli eventi;
 - Possibilità di importare qualunque MIB;
 - Possibilità di eseguire il browse delle MIB su qualunque sistema;
 - Grafici personalizzati dei dati di performance.

2.7 CONFRONTO TRA NAGIOS, OPENNMS, ZENOSS E ZABBIX

Prima di effettuare un qualsiasi tipo di confronto è necessario avere una visione chiara su quali funzionalità vanno considerate importanti per il particolare ambito di applicazione, altrimenti il confronto stesso potrebbe perdere di validità.

Nagios è considerato il prodotto più vecchio e maturo, nato nel 2002 come evoluzione del progetto NetSaint. Anche OpenNMS è nato nello stesso periodo, ma sembra che il suo primo sviluppatore, Tarus Balog, abbia tenuto particolarmente d'occhio Nagios. Zenoss è un prodotto più recente, rilasciato dallo sviluppatore Erik Dahl intorno al 2006.

Zabbix, infine, è nato originariamente nel 1998 come progetto interno in una banca curato dallo sviluppatore Alexei Vladishev. Nel 2001 è stato rilasciato in GPL e solo nel 2004 ne è stata rilasciata la prima versione stabile.

Tutti i prodotti supportano SNMP, e addirittura OpenNMS e Zenoss lo utilizzano come protocollo predefinito.

Parimenti, tutti consentono altre alternative: Zenoss supporta ssh e telnet oltre che ZenPacks personalizzati; Nagios ha gli agenti NRPE e NCSA; OpenNMS supporta JMX e http e può inoltre utilizzare i plug-in di Nagios; anche Zabbix ha il proprio agent ma supporta inoltre telnet, ssh, database monitor e IPMI.

Tutti i prodotti hanno funzionalità di gestione degli utenti per la definizione di utenti, password e ruoli con la possibilità di personalizzare, in termini di restrizioni, ciò che un utente può vedere.

OpenNMS e Zenoss utilizzano RRDTool per la memorizzazione e visualizzazione dei dati di performance; Nagios non ha in realtà nemmeno tale possibilità, ma è possibile accoppiare Cacti come prodotto esterno.

Zabbix, invece, memorizza tutti i tipi di dati, compresi quelli di performance, su database, e per ogni tipo di dato numerico è in grado di generare direttamente dei grafici predefiniti, oltre che permettere la costruzione di grafici personalizzati.

Sorprendentemente, nonostante tutti i prodotti si basino, in maniera più o meno “forte”, su SNMP, nessuno di essi è dotato di serie di un SNMP MIB Browser che possa assistere nella selezione delle MIB per il collezionamento dei dati di stato e di performance. Tale funzionalità è comunque implementabile su Zabbix in maniera facile e immediata grazie ad un apposito plug-in GPL.

2.7.1 CONFRONTO DELLE FUNZIONALITÀ

Nelle tabelle seguenti si confronteranno le funzionalità dei quattro prodotti sulla base dei requisiti indicati nel Paragrafo 3.8

Discovery

	Nagios	OpenNMS	Zenoss	Zabbix
Node discovery	File di configurazione per ogni nodo	File di configurazione con intervalli di inclusione/esclusione	GUI, CLI e importazione batch da file di testo o XML	GUI e importazione batch da file XML
Discovery automatico	No	Sì, per i nodi all'interno di intervalli definiti	Sì, reti e nodi	Sì
Discovery delle interfacce	Possibile tramite file di configurazione	Sì, incluse le porte degli switch	Sì, incluse le porte degli switch	Sì
Discovery dei nodi che non supportano il ping	Sì, utilizzando il plugin check_ifstatus	Sì, send_event.pl	Sì, usando SNMP, ssh o telnet	Sì, usando SNMP, ssh, telnet o http
Database SQL	No	PostgreSQL	MySQL e ZOPE ZEO	MySQL, Oracle, PostgreSQL
Discovery dei servizi	Sì, via plugin	Sì	Sì	Sì
Discovery di applicazioni	Sì, definendo un servizio	No senza agenti extra (NRPE)	Sì, con ssh, ZenPacks o plugin	Sì
Supporto NRPE/NSClient	Sì	Sì	Possibile	No, utilizzo di proprio agent
Supporto SNMP	V 1, 2 e 3	V 1, 2 e 3	V 1, 2 e 3	V 1, 2 e 3
Topologie di rete Layer 3	Sì	No	Sì, fino a 4 hop	Sì
Topologie di rete Layer 2	No	No	No	No

Availability

	Nagios	OpenNMS	Zenoss	Zabbix
Ping status	Sì	Sì	Sì	Sì
Alternative al ping	Sì, via plugin	Nagios plugin	Sì, tramite ssh, telnet, ZenPacks e plugin Nagios	Sì, tramite ssh, telnet, SNMP
Port sniffing	Sì	Sì	Sì	Sì
Monitoraggio processi	Sì, via plugin	Nagios plugin	Sì, tramite la MIB Host Resource	Sì, tramite Zabbix_Agent e SNMP
Tecnologia ad Agent	Generalmente dipende dai plugin Nagios installati	SNMP predefinito; è possibile l'utilizzo di plugin personalizzati	SNMP, ssh, telnet, WMI per Windows, ZenPacks	SNMP, ssh, telnet, WMI per Windows, Zabbix_Agent
Report di disponibilità	Sì	Sì	Sì	Sì

Problem Management

	Nagios	OpenNMS	Zenoss	Zabbix
Console degli eventi configurabile	No	Sì	Sì	Sì
Personalizzazione delle severità	Sì	Sì	Sì	Sì
Configurazione degli eventi	No	Sì	Sì	Sì
Gestione Trap SNMP	No	Sì	Sì	Sì
Notifiche email/sms	Sì	Sì, con escalation configurabile	Sì	Sì, con escalation configurabile
Automazioni	No	Sì	Sì	Sì
De-duplicazione	No	Sì	Sì	Sì
Dipendenze host/servizi	Sì		No	Sì
Analisi cause	Limitata	Interruzioni di percorso	No	Sì

Performance Management

	Nagios	OpenNMS	Zenoss	Zabbix
Collezionamento dei dati di performance con SNMP	No	Sì	Sì	Sì
Collezionamento con altri metodi	No	NSClient, JMX, HTTP	Ssh, telnet, ZenPacks	Zabbix_agent, ssh, telnet, database monitoring, IPMI
Soglie dati performance	No	Sì	Sì	Sì
Grafici dati di performance	No	Sì	Sì	Sì
MIB Compiler	No	No	Sì	No
MIB Browser	No	No	No	Sì, con l'addon SNMP Builder

2.7.2 PRO E CONTRO

Nagios

Pro	Contro
Codice per il system management buono e stabile	No auto-discovery
Buona correlazione tra eventi di servizi ed eventi di host	Console degli eventi scarsa
Controllo della validità dei file di configurazione	Nessuna funzionalità predefinita per il collezionamento e l'impostazione di soglie su dati di performance
Rilettura dei file di configurazione senza interrompere il funzionamento	Nessun modo semplice per ricevere ed interpretare Trap SNMP
Buona documentazione	Nessun MIB compiler o browser

OpenNMS

Pro	Contro
Buone funzionalità predefinite	Scritto in Java, con file di log complessi. Difficoltà di controllare lo stato dei singoli demoni
Codice solido	Assenza di mappe
Configurazione pulita e standardizzata con file XML	GUI pesante. Difficoltà di attenzionare le cose veramente rilevanti
Database singolo (PostgreSQL)	Necessità di riavviare tutto alla modifica dei file di configurazione
Diverse possibilità di personalizzazione delle Trap	Architettura eventi/allarmi/notifiche confusa
Importazione semplificata di MIB Trap	Nessun MIB compiler o browser
Supporto a pagamento da parte di The OpenNMS Group	
Supporto dei plugin di Nagios	Nessuna documentazione in pdf. Difficoltà di trovare informazioni dettagliate nel wiki
Buona disponibilità di documenti Howto di base	Diversi aspetti non documentati

Zenoss

Pro	Contro
Buone funzionalità predefinite	Nessuna correlazione tra eventi riguardanti i servizi e gli eventi relativi agli host
Buona architettura object-oriented basata sul concetto di database CMDB	Implementazione non esente da bug
Mappe di topologia, fino a 4 hop	
Disponibilità di numerosi plugin e ZenPacks	Nessun MIB browser
Email di notifica contenenti link diretti a Zenoss	Nessun modo per cambiare i colori degli eventi
Disponibilità di versione commerciale	Disponibilità di versione commerciale
Disponibilità di libri e manuali Quick Start e Amministrazione di buon livello	Diversi aspetti non documentati
Supporto di Cacti e di plugin Nagios	

Zabbix

Pro	Contro
Buone funzionalità predefinite	Mappe di topologia manuali
Reportistica e andamenti grafici per ogni tipo di dato	Reportistica migliorabile
MySQL, PostgreSQL ed Oracle come database supportati	
MIB Browser come plugin	Nessun MIB Compiler
SLA per servizi IT	Curva di apprendimento ripida
Supporto commerciale a pagamento da parte di Zabbix SIA	

2.8 SCELTA DEL SISTEMA DI MANAGEMENT

In definitiva, la scelta del sistema di management più appropriato deve essere unicamente guidata dai requisiti desiderati.

Per piccoli ambienti di management, potrebbe essere preso in considerazione Nagios, ben testato ed affidabile e con alle spalle il supporto di una grossa comunità. Si tenga presente però che, per qualunque altro controllo che non sia dei semplici ping check o SNMP check, potrebbe esservi la necessità di installare plugin anche sugli host da monitorare. Le notifiche sono abbastanza semplici da impostare, ma se si ha bisogno anche di effettuare un minimo di analisi sul log degli eventi, allora Nagios non è la scelta migliore.

OpenNMS e Zenoss sono entrambi dei prodotti estremamente competitivi, dotati di funzionalità di auto-discovery, monitoraggio della disponibilità, problem management, performance management e reporting.

Zenoss ha una qualche forma di mapping di topologie di rete ed una migliore documentazione, ma il codice sembra essere meno affidabile.

OpenNMS soffre d'altra parte di un approccio un po' confusionario riguardo alla gestione di eventi, allarmi e notifiche. Inoltre, la modifica di un file di configurazione richiede il riavvio dell'intero ambiente.

Zabbix, infine, nonostante abbia una curva di apprendimento ripida, è dotato di quasi tutte le funzionalità normalmente richieste, supporta diversi tipologie di database di backend, ed offre delle peculiarità non riscontrabili in altri sistemi, quali il Browser SNMP e la gestione dei Service Level Agreement (SLA) per i servizi.

Data la sua architettura ben definita, la possibilità documentata di estenderne le funzionalità utilizzando apposite API, il supporto quasi completo a SNMP (manca solo un MIB compiler), la scelta più appropriata nell'ambito del presente lavoro è quindi sembrata essere quella di Zabbix.

2.9 ZABBIX

Zabbix è nato nel 1998 come progetto interno in una banca, a cura dello sviluppatore Alexei Vladishev. Nel 2001 è stato rilasciato in GPL e solo nel 2004 è venuta alla luce la prima versione stabile. L'ultima versione disponibile è la 1.8.2 ed è attualmente sviluppato da Zabbix SIA.

Zabbix è composto sostanzialmente di tre moduli distinti: server, agent, front-end. Server e agent sono scritti in linguaggio C, mentre il front-end è implementato in PHP e Javascript.

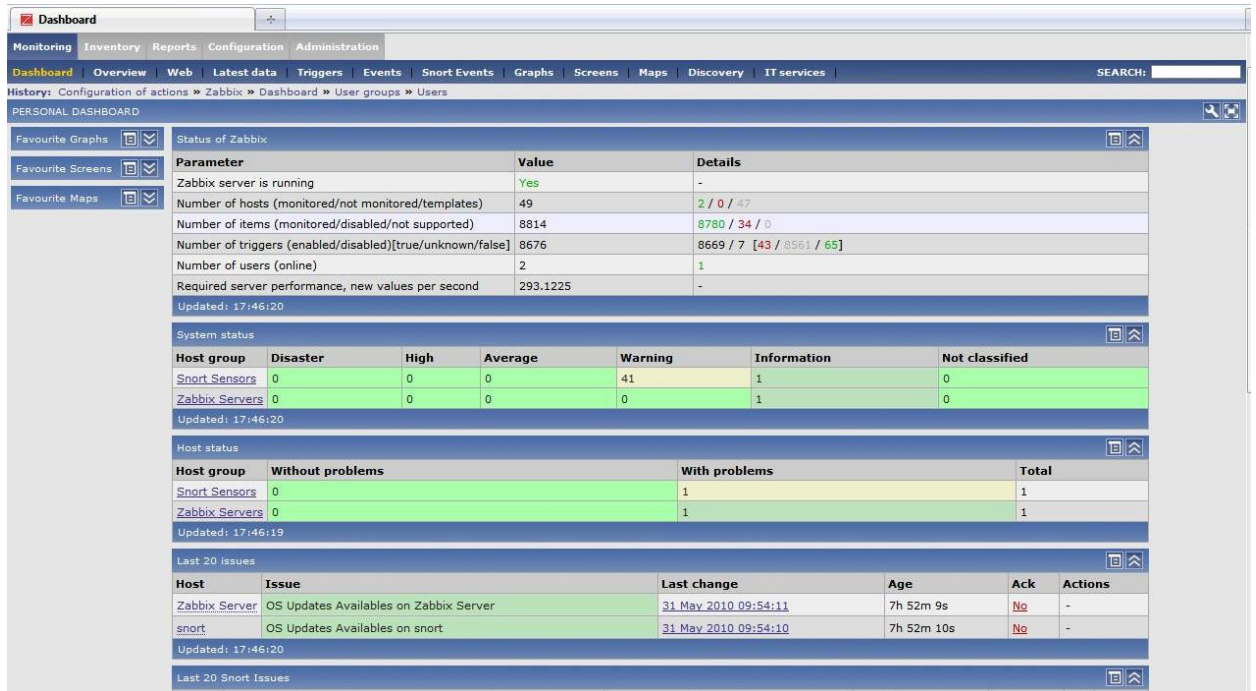


FIGURA 2.1: DASHBOARD DI ZABBIX

2.9.1 DISCOVERY

Zabbix è un sistema molto potente e complesso ma la sua documentazione, pur essendo chiara e dettagliata riguardo a tutte le caratteristiche e funzionalità, lascia un po' a desiderare in termini di setup iniziale del monitoraggio. Bisogna, infatti, tenere presente che esiste un ordine di precedenza tra i passi configurazione, che devono essere effettuati per attivare il monitoring di un host o di un servizio:



La creazione di un host avviene premendo il bottone *Create Host* nella pagina "Configuration > Hosts".

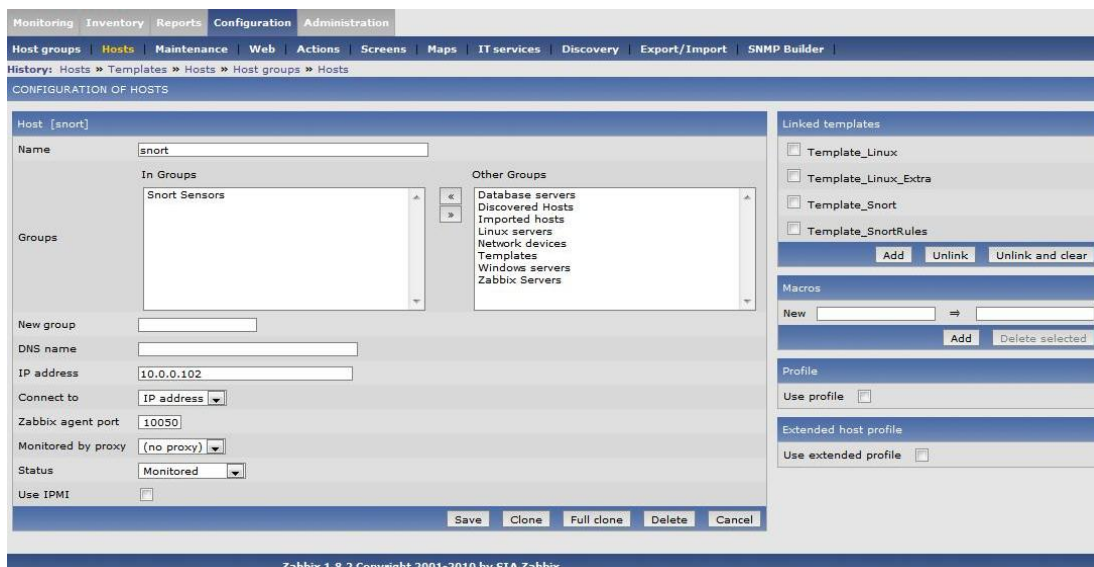


FIGURA 2.2: CREAZIONE DI UN HOST IN ZABBIX

In questa fase si dovranno indicare il nome dell'host, gli eventuali gruppi cui dovrà appartenere, l'indirizzo IP, il numero di porta da interrogare ed infine l'associazione con uno o più template, dai quali l'host dovrà ereditare le caratteristiche (Items, Applications, Triggers, Graphs) ivi definite.

Riguardo al numero di porta, in caso di semplice controllo ICMPcheck si indicherà il valore "0"; nel caso di agent Zabbix si indicherà il valore "10050" (o eventualmente altro numero di porta, allineato con quello indicato in fase di configurazione dell'agent); nel caso di SNMP, infine, si utilizzerà il valore "161".

Oltre alla procedura di creazione manuale, è disponibile una potente funzionalità di discovery automatica, basata sulle seguenti informazioni:

- Intervalli di indirizzi IP;
- Disponibilità di servizi (FTP, SSH, HTTP, POP3, IMAP, TCP, etc.);
- Informazioni ricevute da agent Zabbix;
- Informazioni ricevute da agent SNMP.

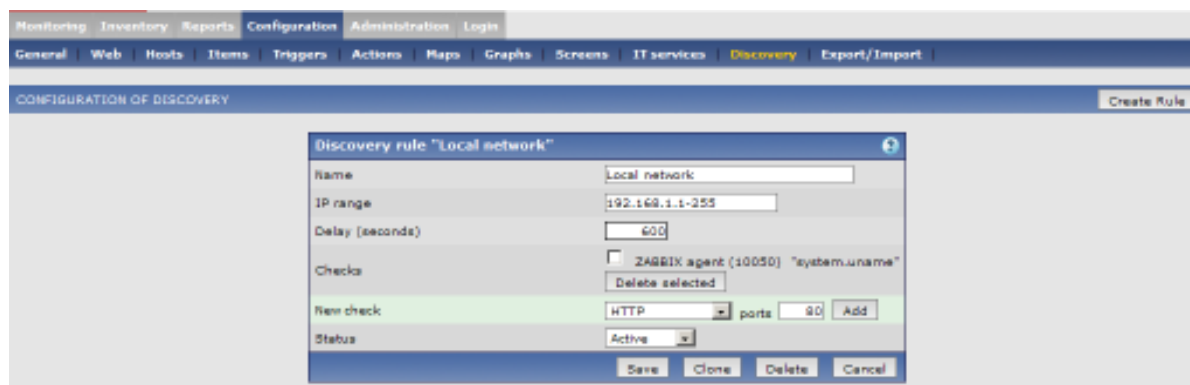


FIGURA 2.3: DISCOVERY AUTOMATICO IN ZABBIX

Ogni servizio e host (IP) verificato dal modulo di discovery genera degli eventi che possono essere utilizzati per creare regole per le seguenti azioni:

- Generazione di notifiche;
- Aggiunta e rimozione di host;
- Attivazione e disattivazione di host;
- Aggiunta e rimozione di un host ad/da un gruppo;
- Associazione host-template;
- Esecuzione di script remoti.

2.9.2 AVAILABILITY MONITORING

Vi sono diverse modalità per monitorare un servizio in esecuzione da qualche parte all'interno della propria rete, o anche all'esterno ed il metodo da utilizzare dipenderà dal livello di accesso che si ha sull'host e dal servizio.

Nel caso del controllo della risposta di un web server, ad esempio, per assicurarsi che il server stia rispondendo si potrà utilizzare una semplice richiesta sulla porta 80, senza alcuna necessità di installare agent sull'host. Nel caso di un router, si utilizzeranno query SNMP. Ma se di un server web che prende le pagine da un database si volesse controllare anche lo stato del database, allora sarà necessario installare sull'host l'agent.

Avendo definito un host, il modo più semplice per associarvi un test di disponibilità è di creare un item di tipo *simplecheck*, con associata la key *icmping*, corrispondente a un semplice ping test all'indirizzo IP dell'host.

Le tipologie di item disponibili sono:

- Zabbix agent (anche con check attivi);
- Simple check;
- SNMP agent V1, V2 e V3;
- Zabbix trapper;
- Zabbix internal;
- Zabbix aggregate;
- External check;
- Database monitor;
- IPMI agent;
- SSH agent;
- Telnet agent;
- Calculated.

2.9.3 PROBLEM MANAGEMENT

Definiti gli Item per il collezionamento dei dati, a questi possono essere associati uno o più Trigger, cioè delle espressioni logiche che rappresentano lo stato del sistema in funzione dei valori dei dati stessi.

Lo stato di un Trigger (expression) viene ricalcolato ogni qualvolta Zabbix riceve dei nuovi valori per un Item, se tale Item fa parte dell'espressione. L'espressione può assumere i seguenti valori:

- **PROBLEM:** Indica normalmente che è successo qualcosa. Ad es. carico della cpu troppo elevato.
- **OK:** E' lo stato normale di un trigger.
- **UNKNOWN:** Indica che Zabbix non è stato in grado di valutare l'espressione. Ciò può accadere quando:
 - il server non è raggiungibile;
 - l'espressione del trigger non può essere valutata per mancanza di dati sufficienti;
 - l'espressione del trigger è stata modificata di recente.

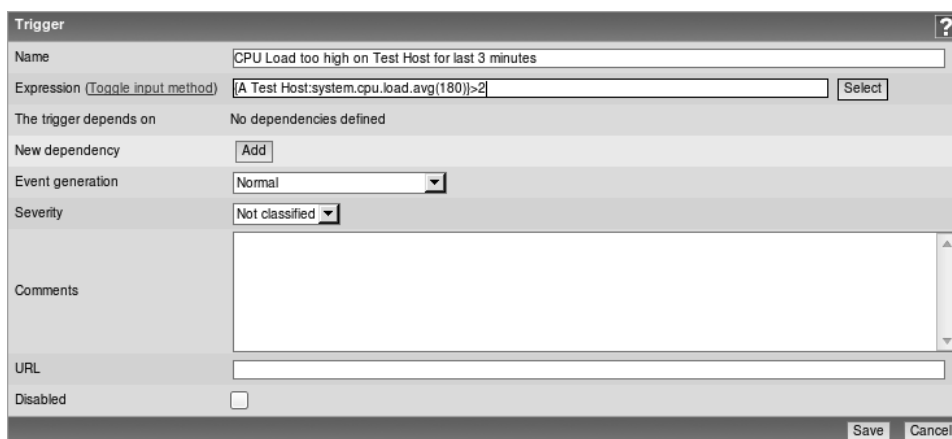


FIGURA 2.4: CREAZIONE DI UN TRIGGER IN ZABBIX

In fase di creazione, per ciascun trigger possono anche essere definite delle relazioni di dipendenza e può essere impostato il livello di severità che dovrà caratterizzare l'allarme.

Il passo successivo potrà essere a questo punto quello di definire un'azione, per inviare ad esempio una notifica via email o sms.

Un'azione è definita da tre componenti principali:

- **Configurazione generale:** consente l'impostazione delle opzioni generali, come il subject della mail da inviare e il messaggio;
- **Operazioni:** specificano cosa deve essere esattamente fatto, compreso chi invia il messaggio, a chi deve essere inviato, e quale messaggio inviare;

- **Condizioni:** permettono di specificare quando deve essere attivata l'azione e quando eseguire le operazioni. E' possibile indicare, anche insieme, diversi tipi di condizioni, come host, host group, tempo, problemi specifici (trigger) e loro severità, e tanti altri.

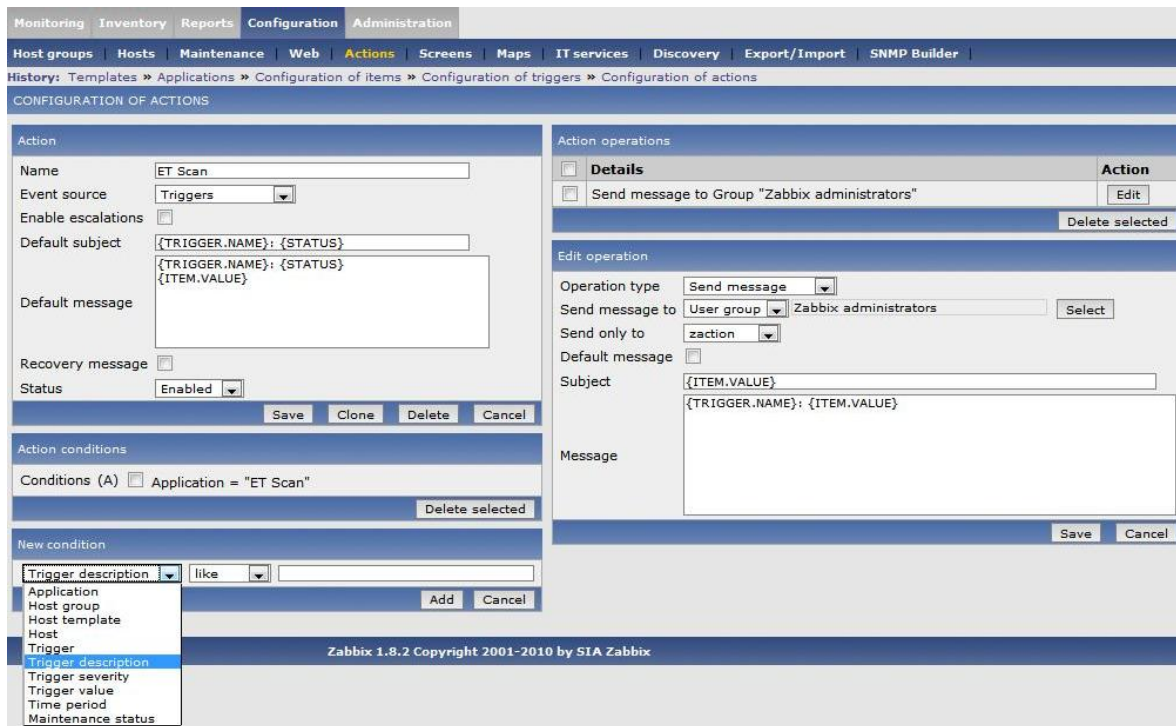


FIGURA 2.5: DEFINIZIONE DI UN'AZIONE IN ZABBIX

2.9.4 FLUSSO DELLE INFORMAZIONI

Dati un host, un suo item e un trigger ad esso riferito, vediamo un attimo quale sia il flusso delle informazioni scambiate tra le varie entità in Zabbix.

- Quando, in base al valore dell'item, l'espressione di un trigger diventa vera, lo stato del trigger viene impostato a `PROBLEM`.
- Quando l'espressione ritorna ad essere falsa, lo stato torna ad essere `OK`.
- Ogni volta che avviene il cambio di stato di un trigger, viene generato un evento, che contiene i dettagli del cambiamento di stato del trigger, quando è successo e quale è lo stato attuale.
- Quando si configura un'azione, si possono impostare diverse condizioni, in modo da considerare solo alcuni degli eventi. Ad ogni azione possono inoltre essere associate delle operazioni, che indicano cosa esattamente debba essere fatto.

3 INTRUSION MANAGEMENT SYSTEM CON ZABBIX, SNORT ED OSSEC

3.1 INTRODUZIONE

Avendo scelto Zabbix come sistema di management di riferimento, si vedrà adesso in che modo poterlo estendere ed utilizzare per renderlo una soluzione di Service Management, verificandone l'integrazione con sistemi di acquisizione delle informazioni diversi dai canonici dispositivi di rete o server, componenti tipici di un'infrastruttura ICT.

Si dimostrerà come sia effettivamente possibile integrare un qualunque oggetto/servizio in una piattaforma di management tradizionale, grazie all'utilizzo del protocollo SNMP.

Quindi, pur continuando a considerare in questa sede un ambito applicativo ICT, un semplice ed immediato processo di generalizzazione e astrazione ci consente di applicare potenzialmente gli stessi principi ad un qualunque altro ambito applicativo, con i soli vincoli che quest'ultimo sia in qualche modo interfacciabile con un'infrastruttura di rete informatica e che i suoi dispositivi/oggetti/servizi possano comunicare su tale rete mediante protocollo SNMP.

Qui, in particolare, ci si preoccuperà di realizzare una piattaforma per il management del servizio di Intrusion Detection.

Nei paragrafi successivi si descriveranno brevemente le caratteristiche generali dei sistemi di Intrusion Detection (Network-based e Host-based), si descriverà l'architettura implementata ed, infine, verranno descritte in dettaglio tutte le componenti del sistema, rivolgendo particolare attenzione a quelle funzionalità che sono state utilizzate nello sviluppo della parte sperimentale di questo lavoro.

3.2 INTRUSION DETECTION SYSTEMS

Collegare un PC in rete significa esporlo alle possibili minacce di attacchi e tentativi di intrusione da parte di altri utenti.

Su una rete di piccole dimensioni questo rischio, per quanto sia molto basso, è comunque reale, ma su una rete di maggiori dimensioni, con accessi e servizi pubblici, l'entità del rischio diventa potenzialmente grave.

Occorre quindi avere a disposizione dei sistemi in grado di rilevare e di bloccare i tentativi di accesso non autorizzati e di notificarli all'amministratore di rete o di sistema, in modo che questi possa prendere le dovute precauzioni anche da remoto.

Un Intrusion Detection System (IDS) è uno strumento hardware o software in grado di rilevare attività anomale o sospette di tentativi di accesso non autorizzato ad una rete o ad un singolo host.

E' possibile identificare due famiglie ben distinte di IDS, sulla base di cosa controllano:

- **NIDS** (*Network based Intrusion Detection Systems*), che assicurano la sicurezza a livello di rete. Questi IDS risiedono su macchine dedicate ed operano analizzando in real time il traffico del segmento di rete nel quale si trovano. Sono in grado di monitorare simultaneamente il traffico generato verso e da più host e solitamente non hanno alcun impatto negativo sulle performance degli host stessi.
- **HIDS** (*Host based Intrusion Detection Systems*), che assicurano la sicurezza a livello di host. Sono installati sull'host da proteggere ed analizzano i dati di log alla ricerca di possibili anomalie. Hanno lo svantaggio che, consumando risorse di sistema, possono influire negativamente sulle performance dell'host in cui risiedono.

A queste due categorie se ne aggiunge a rigore una terza, rappresentata dai sistemi ibridi, che cerca di combinare le caratteristiche degli NIDS e degli HIDS al fine di sfruttare i vantaggi di entrambi, al fine di rilevare un maggior numero di intrusioni in modo più preciso.

Gli attacchi alle reti informatiche o ai singoli host possono sfruttare diverse tecniche di aggancio, quali ad esempio:

- sfruttamento di una vulnerabilità nota di un servizio;
- invio di dati malformati;
- tentativi di accesso agli host tramite l'innalzamento illecito dei privilegi degli utenti;
- accessi non autorizzati ai file ed i classici programmi malevoli come virus, cavalli di troia (trojan) e worm.

Un IDS deve essere quindi in grado di rilevare tutti i tentativi di attacco dalle diverse fonti, tools automatici, virus, crackers, o semplici utenti che cercano di misurare le proprie capacità.

S'intuisce, allora, come uno strumento di questo tipo sia fondamentale per una qualsiasi architettura di network security, fornendo un livello di difesa tale da monitorare in real time la rete stessa.

Gli IDS, sulla base dell'approccio adottato per l'attività di rilevamento delle intrusioni, possono essere classificati in due macrocategorie:

1. Sistemi basati su regole;
2. Sistemi adattativi.

Gli IDS basati su regole utilizzano database, librerie e firme di attacco (signature) per rilevare le intrusioni. Quando il sistema rileva che il traffico o un'attività di rete corrisponde a una regola nota all'IDS, questo viene interpretato come tentativo di intrusione e segnalato. L'affidabilità di questo tipo di IDS dipende interamente dalla tempestività con cui il database degli attacchi viene aggiornato, altrimenti lo strumento perderebbe immediatamente efficacia, non essendo in grado di rilevare le nuove minacce.

Inoltre, non sempre le regole sono efficienti, cioè, nel caso in cui una regola venga scritta in maniera troppo specifica, gli attacchi che sono simili, ma non identici, alla regola, passeranno inosservati.

Questa tecnica di rilevamento basata sulle firme è molto simile a quella usata per il rilevamento dei virus, ma così strutturato tale tipo di approccio evidenzia la debolezza di non riuscire a rilevare nuovi attacchi, perché non precedentemente catalogati nel database.

Gli IDS adattativi usano tecniche più avanzate, come il pattern recognition e l'intelligenza artificiale, per riconoscere i diversi attacchi, ma anche per "imparare" a riconoscere quelli nuovi.

Un IDS di questo tipo identifica i segnali di un'intrusione per il semplice fatto che questa viene identificata come una anomalia di comportamento rispetto al normale traffico di rete. La caratteristica di questi sistemi è quindi la loro attitudine a scoprire i nuovi attacchi, e per fare questo richiedono una fase di messa in opera più lunga e impegnativa rispetto ad un IDS basato su firme, in quanto deve preventivamente analizzare e studiare il comportamento "tipico e normale" della rete in modo da riuscire a isolare le aggressioni da tutte quelle altre attività che invece generano solo "rumori" e possono quindi condurre a falsi rilevamenti.

3.2.1 HIDS vs NIDS

Un *Host-based Intrusion Detection System* (HIDS) è un software in grado di monitorare il funzionamento di un sistema "dal suo interno", anziché ricorrere all'uso delle interfacce di rete².

I software HIDS si occupano di sorvegliare costantemente il comportamento del sistema oggetto d'esame e lo stato in cui sta operando.

Mentre i NIDS ispezionano il contenuto di ogni singolo pacchetto dati in transito, un HIDS può stabilire a quali risorse tentano di accedere i programmi installati, bloccando tempestivamente azioni sospette.

² Come fanno, invece, i *Network-based Intrusion Detection Systems* (NIDS)

Ad esempio, un word-processor non deve, ovviamente, modificare le password di sistema o la configurazione di aree vitali: se lo fa, significa che, con buona probabilità, sta eseguendo del codice maligno, collegato all'azione di un malware.

Un HIDS controlla lo stato del sistema verificando le informazioni memorizzate, sia in RAM sia sul file system, i file di log, e così via, con l'obiettivo di rilevare tempestivamente le anomalie. Questa attività viene eseguita analizzando continuamente i log di sistema, verificando l'integrità dei file memorizzati, controllando la presenza di root-kit e tenendo traccia delle performance.

Un NIDS, invece, fornisce una visione granulare di tutto ciò che sta accadendo nella rete locale, monitorando il traffico tramite una scheda di rete connessa direttamente alla LAN mediante un hub, una porta "SPAN" di uno switch oppure tramite un apparato detto "network tap", che agevola il monitoraggio non invasivo del flusso di dati in transito.

Un NIDS è un sistema di monitoraggio molto potente, che consta però di qualche limite.

Che cosa succede, ad esempio, quando vengano impiegate tecniche conosciute per "dribblare" il NIDS? Che cosa accade quando le comunicazioni tra due sistemi sono crittografate³? Che cosa succede se l'attacco sferrato da un aggressore è effettuato impiegando tecniche crittografiche?

In questi frangenti un NIDS non può essere efficace.

Un HIDS, invece, può rilevare le attività in corso su macchine server e singole workstation, generando messaggi di allerta simili a quelli di un NIDS, con la differenza che è in grado di ispezionare l'intero flusso di dati scambiati nelle varie comunicazioni.

Le tecniche precedentemente citate non hanno efficacia nei confronti di un HIDS perché, tra l'altro, nel caso di comunicazioni crittografiche, queste possono essere comunque monitorate, perché i dati cifrati sono analizzati prima che diventino tali.

Un HIDS, inoltre, può svolgere compiti aggiuntivi quali i controlli sull'integrità dei file, sul contenuto del registro di sistema, operazioni di analisi dei log, rilevamento di root-kit e così via:

- **Controllo dell'integrità.** Per ciascun file presente sul sistema è sempre possibile generare una firma digitale attraverso l'utilizzo di una "funzione hash", al fine di produrre un'impronta che caratterizza in modo univoco il singolo file preso in esame. Se un file con lo stesso nome, ad esempio, ha una firma differente significa che il suo contenuto è variato. Un HIDS è in grado di monitorare file importanti con lo scopo di rilevarne eventuali cambiamenti dovuti, spesso, all'applicazione di aggiornamenti ma anche, purtroppo, all'azione di malware.

³ Un NIDS non permette, ovviamente, di esaminare il contenuto di pacchetti dati cifrati

- **Monitoraggio del registro di sistema.** Il registro di Windows è una delle componenti più importanti e allo stesso tempo più delicate del sistema operativo Microsoft. Un HIDS può monitorare gli interventi applicati al contenuto del registro di sistema. Grazie a questo tipo di attività, è possibile assicurarsi che un utente oppure un'applicazione non introduca delle modifiche con scopi maligni.
- **Rootkit.** Un HIDS si occupa anche di preservare i sistemi tenuti sotto controllo dall'azione di rootkit. Qualora un malware del genere riuscisse ad insediarsi sul sistema, i danni sarebbero, infatti, enormi. Un rootkit sa nascondere servizi, processi, file, cartelle, chiavi del registro di Windows e porte sia agli occhi dell'utente sia allo stesso sistema operativo.
- **Avvio di operazioni in modo automatico.** Spesso gli HIDS permettono anche di eseguire particolari operazioni, in modo del tutto automatico, in risposta al verificarsi di uno specifico evento o di un insieme di attività potenzialmente pericolose.

3.3 ARCHITETTURA

In termini generali, l'architettura della piattaforma implementata non si discosta da quelle in cui opera un tipico sistema di network e system management.

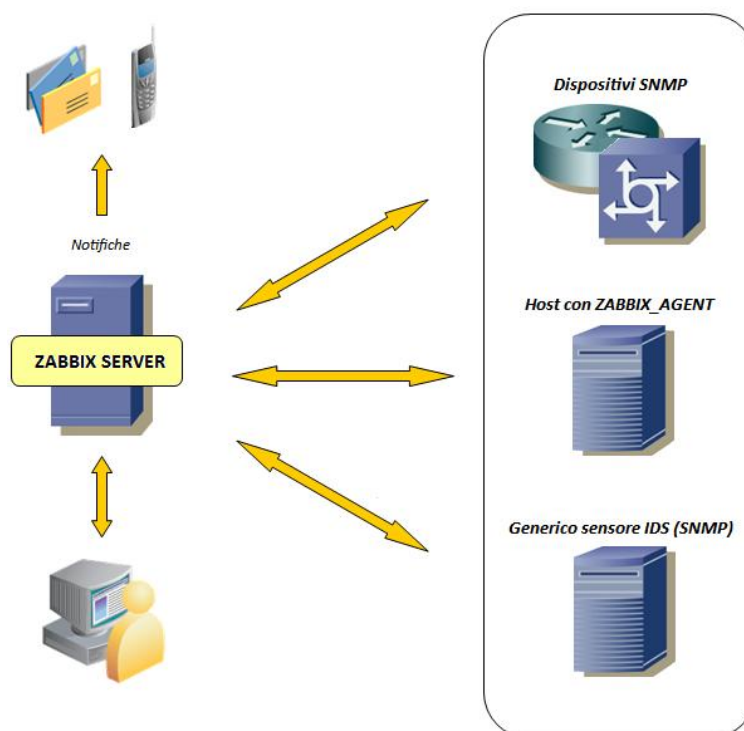


FIGURA 3.1: ARCHITETTURA DELLA PIATTAFORMA

La sola particolarità risiede nella presenza dei sensori IDS (Snort ed Ossec) che, al verificarsi di un evento da segnalare, inviano alla stazione di management una trap SNMP, contenente un messaggio di testo in cui il carattere “|” fa da separatore per le informazioni riguardanti il nome del sensore, l’identificativo della regola che ha fatto scaturire l’allarme ed il testo dell’alert vero e proprio.

In questo contesto si evidenzia, piuttosto, l’esigenza di rimappare le informazioni contenute nelle trap ricevute in dati item (appartenenti ad un host) di Zabbix.

A tale scopo, le informazioni contenute nelle trap sono in realtà preventivamente elaborate da una componente intermedia costituita dalla coppia di processi *snmptrapd* e *snmptt*, che si occuperanno di:

- ricevere le trap SNMP;
- effettuare la conversione delle informazioni ricevute dal formato binario dell’OID ad un formato testuale;
- estrarre le informazioni relative al nome del sensore, all’identificativo della regola e al messaggio di testo dell’allarme;
- invocare, tramite gli script *zow* e *zws*, il programma *zabbix_sender* parametrizzato con le informazioni dell’host, dell’item e del valore di quest’ultimo.

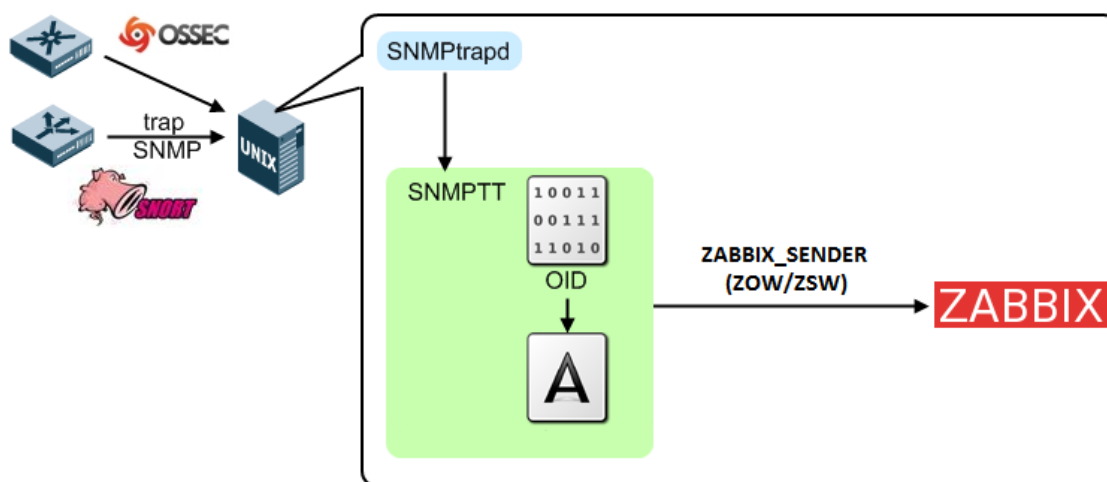


FIGURA 3.2: ELABORAZIONE DELLE TRAP SNMP GENERATE DA SNORT ED OSSEC

3.4 COMPONENTI DEL SISTEMA

3.4.1 NET-SNMP

Net-SNMP è una suite di applicazioni utilizzate per implementare i protocolli SNMP v1, SNMP v2c e SNMP v3, sia su IPv4 sia su IPv6.

La suite include:

- Applicazioni a linea di comando per:
 - leggere informazioni da un qualsivoglia dispositivo SNMP, utilizzando richieste singole (*snmpget*, *snmpgetnext*) o multiple (*snmpwalk*, *snmpstable*, *snmpdelta*);
 - modificare informazioni di configurazione su dispositivi SNMP (*snmpset*);
 - estrarre una collezione prefissata di informazioni da dispositivi SNMP (*snmpdf*, *snmpnetstat*, *snmpstatus*);
 - convertire gli OID delle MIB tra le forme numerica e testuale e visualizzare il contenuto e la struttura delle MIB (*snmptranslate*).
- Un browser grafico per le MIB (*tkmib*) scritto in Tk/perl;
- Un'applicazione di tipo daemon per la ricezione delle notifiche SNMP (*snmptrapd*). Le notifiche possono essere loggate (syslog, Windows Event log, file di testo), inoltrate ad altro sistema di management SNMP o passate ad un'applicazione esterna;
- Un agente programmabile per rispondere a richieste SNMP (*snmpd*), capace di riconoscere un gran numero di moduli MIB, estendibile mediante l'utilizzo di moduli caricati dinamicamente, script e comandi esterni, e mediante l'utilizzo dei protocolli SNMP multiplexing (SMUX) e Agent Extensibility (AgentX);
- Una libreria per lo sviluppo di nuove applicazioni SNMP, utilizzando i linguaggi C e Perl.

In questo lavoro, la suite è stata utilizzata in particolare:

1. come libreria di ausilio allo sviluppo, per abilitare Snort all'invio delle notifiche SNMP, tramite il plug-in di output appositamente realizzato;
2. per la ricezione tramite il programma *snmptrapd* delle notifiche stesse ed il successivo inoltro al tool *snmptt* per il preprocessing e l'invio al sistema NMS Zabbix.

3.4.2 SNORT

Tra i diversi software commerciali ed open source in grado di svolgere i compiti di un IDS, troviamo il pacchetto Snort, l'IDS open source sicuramente più noto e diffuso caratterizzato da elevate performance, potenza e versatilità.

Snort può essere configurato per operare in quattro modalità:

- Sniffer, che semplicemente legge i pacchetti dalla rete e li visualizza a schermo come stream continuo;
- Packet Logger, che memorizza i pacchetti su disco;
- Network Intrusion Detection System, quella più complessa e configurabile, che consente a Snort di analizzare il traffico di rete, di effettuare il match con un set di regole definite dall'utente e di eseguire delle azioni a seconda di ciò che viene riscontrato;
- Inline, nella quale i pacchetti da analizzare vengono prelevati dal firewall di Linux *IPTables*, piuttosto che tramite le librerie *libcap*, e con la quale, sempre grazie ad *IPTables*, è possibile bloccare o far passare i pacchetti a seconda di quali regole usano la direttiva *inline*.

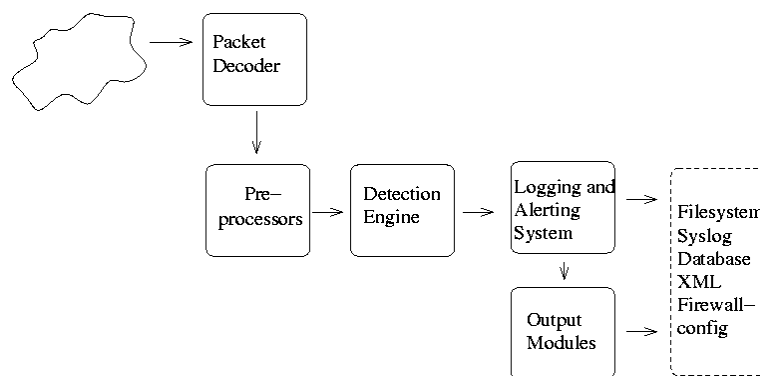


FIGURA 3.3: INTERAZIONE TRA LE COMPONENTI INTERNE DI SNORT

La configurazione di Snort prevede la modifica del file *snort.conf* e la creazione di files con estensione *.rules*, che comporranno il database interno delle firme, ovvero le regole che verranno applicate ai pacchetti in transito, da applicare per rilevare le intrusioni.

Dal sito principale è possibile scaricare un file di configurazione e i file delle regole relativamente aggiornati rispetto a quanto installato di default con il pacchetto standard. Successivamente a questa fase Snort consente di essere adattato alle specifiche esigenze di controllo del traffico di rete modificando il file di configurazione, in base a quattro punti fondamentali:

1. impostazione delle variabili globali di ambiente (*HOME_NET*, *EXTERNAL_NET*, *DNS_SERVERS*);
2. configurazione dei preprocessori, cioè quei chunk di codice integrati nel software che assolvono uno specifico compito, come per esempio la corretta gestione dei pacchetti frammentati o l'impostazione di una soglia di sensibilità per il riconoscimento delle attività di scansione delle porte TCP/UDP;
3. configurazione dei plug-in di output, utilizzati da Snort per definire come i log devono essere registrati;

4. personalizzazione del database interno delle regole, che vengono definite mediante l'utilizzo di un semplice linguaggio di descrizione e sono sostanzialmente suddivise in due parti:
 - header, che contiene le informazioni sull'azione, il protocollo, gli indirizzi IP, le porte e le netmask sorgente e destinazione;
 - options, dove sono definiti i messaggi di alert e le informazioni sulle parti del pacchetto da analizzare per determinare se deve essere eseguita l'azione.

Snort può utilizzare differenti meccanismi di log, ma fondamentalmente ne possiamo distinguere due:

- il primo sistema consiste nel produrre dei file che salvano una traccia di tutto il traffico di rete catturato;
- il secondo coinvolge un file `audit.ids`, che conterrà l'indicazione dei pacchetti per cui è stata trovata una corrispondenza esatta nel database interno di Snort.

E' importante che questi file, a seguito di un tentativo di attacco e/o intrusione, non siano alterati o distrutti. Come misura precauzionale al fine di ottenere questo risultato è possibile indirizzare l'output di Snort verso un database fisicamente ospitato su un altro host, più protetto ed utilizzato solo per questo scopo.

E' inoltre necessario definire le corrette politiche di accesso ai dati da parte degli utenti della rete, magari facendo in modo che a queste informazioni di log si acceda solo se autorizzati.

Infine è corretta pratica effettuare periodicamente un backup dei log su supporti protetti e rimovibili.

I file di log, secondo le caratteristiche dei server e delle "attenzioni" esterne che essi ricevono, possono crescere nel tempo in modo considerevole.

Questo può comportare un'attività di revisione impegnativa e che come tale può essere soggetta a errori, apparentemente banali, ma di cui si deve tenere conto, poiché per la complessità del problema possono nascere delle sviste, oppure possono passare inosservati i segni di un'attività che preannuncia un attacco imminente.

È evidente come sia fondamentale interpretare nel modo più corretto possibile questi log, definendo ad esempio delle politiche di consultazione, oppure abbinare il software più opportuno che semplifichi l'accesso e la presentazione dei tracciati log, come ad esempio il tool *SnortSnarf*⁴.

⁴ <http://sourceforge.net/projects/snortsnarf/>

3.4.3 IL PLUGIN SNORT-SNMP

Snort SNMP consiste in un addon per Snort inizialmente sviluppato e rilasciato in licenza open source dall'azienda giapponese Cyber Solutions, che l'ha mantenuto e aggiornato fino alla versione 2.2.0 di Snort, per consentire a quest'ultimo l'invio di notifiche SNMP (trap o inform) a un qualunque sistema di network e system management.

Poiché a Snort.org è stato assegnato l'OID *iso.org.dod.internet.private.enterprises.snort* (1.3.6.4.1.10234), come suo sotto albero sono state definite tutte le coppie nome-valore che possono essere di interesse nell'ambito di una notifica SNMP riguardante gli alert generabili da Snort, arrivando alla definizione di due tabelle contenenti tutti i Managed Objects (MO) implementati e relativi OID:

- *sidaSensorTable*: le cui righe, indicizzate dal campo *sidaSensorID*, descrivono ogni singolo sensore Snort;
- *sidaAlertTable*: le cui righe, indicizzate dai campi *sidaSensorID* ed *alertID*, descrivono gli alert generati da ogni singolo sensore.

L'attivazione del plug-in richiede l'aggiunta di una apposita riga all'interno del file di configurazione principale di Snort *snort.cfg*, utilizzando la sintassi seguente:

```
output trap_snmp: alert, <sensorID>, [NotificationOptions] {trap|inform} -v
<SnmVersion> <SnmParameters>
```

dove:

- *alert* indica che sarà usata la facility alert di Snort;
- *<sensorID>* è l'identificativo unico associato al sensore Snort;
- *[NotificationOptions]* può assumere i valori *[c],[p]*, con i seguenti significati:
 - *c*: genera notifiche compatte, non inviando quei MO i cui valori sono sconosciuti, non disponibili o non applicabili;
 - *p*: produce un estratto del pacchetto
- *{trap|inform}* indica se debba essere generata una trap od una notifica di tipo *inform*
- *<SnmVersion>* indica la versione di SNMP da usare per la comunicazione
- *<SnmParameters>* dipendono dalla versione di SNMP utilizzata

Per trap SNMPv2c destinate alla destinazione *myTrapListener*, sulla porta 162 ed utilizzando *myCommunity* come community name, avremo quindi:

```
output trap_snmp: alert, 7, trap -v 2c -p 162 myTrapListener myCommunity
```

Per generare invece delle *inform* verso la stessa destinazione, la riga da inserire in *snort.cfg* sarà:

```
output trap_snmp: alert, 7, inform -v 2c -p 162 myTrapListener myCommunity
```

Le trap o le inform generate verranno quindi ricevute da un processo *snmptrapd*, sotto forma di eventi binari SNMP standard del tipo:

```
2001-08-01 21:21:00 megahira.priv.cysol.co.jp [192.168.0.11]:
sysUpTime.0 = Timeticks: (52918825) 6 days, 2:59:48.25
snmpTrapOID.0 = OID: sidaAlertGeneric
sidaSensorVersion.7 = Snort! <*- Version 1.8-RELEASE (Build 43)
sidaSensorAddressType.7.7 = unknown(0)
sidaSensorAddress.7.7 = "==" NA =="
sidaAlertTimeStamp.7.7 = 996668460. 98668
sidaAlertMsg.7.7 = Scan
sidaAlertMoreInfo.7.7 = Protocol: ICMP
sidaAlertSrcAddressType.7.7 = ipv4(1)
sidaAlertSrcAddress.7.7 = "nnn.mmm.qqq.rr"
sidaAlertDstAddressType.7.7 = ipv4(1)
sidaAlertDstAddress.7.7 = "192.168.0.11"
```

Per gli scopi di questo lavoro, e soprattutto poiché il progetto Snort SNMP è stato abbandonato da tempo e quindi non è stato più allineato alle successive release di Snort, si è reso necessario un importante intervento di riscrittura del codice sorgente originario.

Il nuovo codice è stato realizzato prendendo come modello di riferimento uno dei più semplici plug-in di output standard di Snort, in particolare il modulo *spo_alert_syslog.c* che utilizza il demone *syslogd* di sistema per il log degli alert, nonché il codice sorgente del tool *snmptrap* contenuto nella suite *net-snmp-5.3.2.2*.

Le modifiche al codice sorgente si sono concretizzate in particolare nei seguenti interventi:

- riscrittura del codice relativo alle operazioni di inizializzazione e chiusura di una sessione SNMP e per la costruzione del datagramma da inviare, utilizzando le nuove chiamate alle API della libreria *net-snmp*;
- reingegnerizzazione e semplificazione della funzione *sendInform*, all'interno della quale si effettua la vera e propria costruzione del datagramma SNMP;
- utilizzo delle nuove strutture dati e funzioni disponibili in Snort 2.8.x e non presenti nelle versioni precedenti.

3.4.4 OSSEC

OSSEC è un host-based intrusion detection system (HIDS) Open Source realizzato da Daniel B. Cid dal 2004. Nel giugno 2008 il progetto OSSEC e tutti i diritti detenuti dal suo project leader sono stati acquisiti dalla Third Brigade Inc., che a sua volta è stata incorporata nel maggio 2009 in Trend Micro, la quale ha continuato a mantenere il progetto Open Source.

Il software fornisce funzionalità di intrusion detection per diversi sistemi operativi, inclusi Linux, OpenBSD, FreeBSD, Mac OS X, Solaris e Windows e si presenta come un'architettura centralizzata e cross-platform per le attività di analisi dei log, controllo

dell'integrità dei file, monitoraggio del registro di sistema di Windows, rilevamento di rootkit, alerting e risposta attiva.

OSSEC è composto da un'applicazione principale, un agent Windows ed una interfaccia web:

- **Applicazione principale.** Il core di OSSEC è necessario sia per le installazioni di tipo distribuito che stand-alone, ed è supportato dagli ambienti Linux, Solaris, xBSD e Mac.



- **Agent Windows.** E' necessario per gli ambienti Microsoft Windows, e richiede un core di OSSEC installato altrove in modalità server.



- **Interfaccia Web.** Consiste in una GUI, realizzata in Php, da accoppiare al core di OSSEC per effettuare il monitoraggio degli alert generati nei sistemi.



OSSEC è dotato di un motore di analisi dei log alquanto potente, in grado di analizzare e correlare le informazioni provenienti in diversi formati e da sistemi diversi, quali:

- **Unix:** Unix PAM, sshd (OpenSSH), Solaris telnetd, Samba, Su, Sudo
- **Server FTP:** ProFTPD, Pure-FTPd, vsftpd, Microsoft FTP Server, Solaris ftpd
- **Server Mail:** Imapd e pop3d, Postfix, Sendmail, vpopmail, Microsoft Exchange Server
- **Database:** PostgreSQL, MySQL
- **Server Web:** Apache HTTP Server (access log ed error log), Microsoft IIS (log NCSA e W3C), Zeus
- **Applicazioni Web:** Horde IMP, Modsecurity
- **Firewall:** Iptables, Solaris IPFilter, AIX ipsec/firewall, Netscreen, Windows Firewall, Cisco PIX, Cisco FWSM, Cisco ASA
- **NIDS:** Cisco IOS IDS/IPS module, Snort (log di tipo full, fast e syslog)
- **Tool di sicurezza:** Symantec AntiVirus, Nmap, Arpwatch, Cisco VPN Concentrator
- **Altri:** Named (BIND), Squid, Zeus eXtensible Traffic Manager
- **Event log di Windows** (login, logout, informazioni di audit, ecc.)
- **Log di Routing e Accesso Remoto di Windows**
- **Autenticazione generica Unix** (adduser, login, ecc.)

Nell'ambito degli obiettivi di questo lavoro, al fine di uniformare le modalità di alerting utilizzando esclusivamente il protocollo SNMP per l'invio dei messaggi, OSSEC è stato installato e configurato in modalità stand-alone.

Grazie alla funzionalità *Active Response*, che consente l'esecuzione di programmi/script esterni al verificarsi di un qualunque evento, è stata attivata una risposta attiva predefinita che intercetta gli alert di ogni grado di severità, mediante la specifica nel file di configurazione */var/ossec/etc/ossec.conf* delle seguenti direttive:

```
<command>
  <name>snmptrap</name>
  <executable>snmptrap.sh</executable>
  <expect></expect>
</command>

<active-response>
  <command>snmptrap</command>
  <location>local</location>
  <level>1</level>
</active-response>
```

Lo script *snmptrap.sh* sviluppato ad-hoc effettua il parsing delle informazioni inviate dall'engine di OSSEC al fine di estrarne l'identificativo della regola corrispondente all'alert che si è verificato ed il corrispondente messaggio.

Queste informazioni, unitamente al nome dell'host sul quale è in esecuzione OSSEC, vengono quindi inviate sotto forma di trap SNMP mediante invocazione diretta del programma *snmptrap*:

```
snmptrap -v 2c -c public $SERVER "" OSSEC-ALERT-MIB::ossecTrap alertMsg s "$ALERTMSG"
```

3.4.5 SNMP TRAP TRANSLATOR

Il tool SNMP Trap Translator (SNMPTT) è un gestore di trap SNMP scritto in Perl da utilizzare in abbinamento al programma *snmptrapd* fornito dalle suite Net-SNMP/UCD-SNMP.

La quasi totalità dei dispositivi di rete, e non solo, tra i quali switch, router, server di accesso remoto, ups, stampanti nonché sistemi operativi quali Unix e Windows, sono in grado di inviare delle notifiche, sotto forma di messaggi SNMP Trap o Inform, ad un manager SNMP in esecuzione su una stazione di network management.

Le notifiche possono contenere informazioni di tutti i generi e sono definite all'interno delle Management Information Base (MIB) che ciascun vendor predispone per ogni dispositivo.

Il file MIB contiene quindi le codifiche dei messaggi di tipo TRAP (SMIv1) o INFORM (SMIv2), che definiscono le variabili da passare alla stazione di management quando si verifica un particolare evento.

Il programma *snmptrapd* è un'applicazione che riceve e tiene traccia dei messaggi *Trap* ed *Inform* ricevuti via TCP/IP.

Ad esempio, per la trap *cpqDa3LogDrvStatusChange* di Compaq che notifica dell'operazione di rebuild di un array dischi, si avrà:

```
Feb 12 13:37:10 server11 snmptrapd[25409]: 192.168.110.192: Enterprise
Specific Trap (3008) Uptime: 306 days, 23:13:24.29, .1.3.6.1.2.1.1.5.0 =
SERVER08, .1.3.6.1.4.1.232.11.2.11.1.0 = 0,
.1.3.6.1.4.1.232.3.2.3.1.1.4.8.1 = rebuilding(7)
```

oppure

```
Feb 12 13:37:10 server11 snmptrapd[25409]: 192.168.110.192: Enterprise
Specific Trap (3008) Uptime: 306 days, 23:13:24.29, sysName.0 = SERVER08,
cpqHoTrapFlags.0 = 0, cpqDaLogDrvStatus.8.1 = rebuilding(7)
```

L'output fornito da *snmptrapd* può essere modificato tramite l'opzione *-O* per visualizzare gli OID in forma numerica o simbolica, ma generalmente segue sempre il formato "nomeVariabile=valore", "nomeVariabile=valore", etc.

L'utilizzo di SNMPTT consente di ottenere un messaggio di trap molto più leggibile e descrittivo, grazie all'utilizzo della sostituzione delle variabili. La stessa trap sopra riportata, gestita da SNMPTT, può quindi diventare:

```
Feb 12 13:37:13 server11 TRAPD: .1.3.6.1.4.1.232.0.3008 Normal "XLOGONLY"
server08 - Logical Drive Status Change: Status is now rebuilding
```

semplicemente definendo nel file di configurazione di SNMPTT per la trap *cpqDa3LogDrvStatusChange* la seguente informazione di "formattazione":

```
FORMAT Logical Drive Status Change: Status is now $3.
```

dove \$3 rappresenta la terza variabile come da definizione nel file MIB, che nel caso specifico sarebbe la variabile *cpqDaLogDrvStatus*.

Un altro esempio di riga di formattazione potrebbe essere anche:

```
FORMAT Compaq Drive Array Spare Drive on controller $4, bus $5, bay $6
status is $3.
```

che produrrebbe come risultato:

```
Compaq Drive Array Spare Drive on controller 3, bus 0, bay 3 status is
Failed.
```

SNMPTT può effettuare le operazioni di log utilizzando le seguenti destinazioni: log su file di testo, syslog, Windows Event log, database SQL. Inoltre, la trap rielaborata può anche essere passata a un programma esterno da eseguire in tempo reale, come un client email o, come nel caso in esame, il programma *zabbix_sender* per l'inoltro al sistema NMS Zabbix⁵.

Oltre alla possibile sostituzione di variabili, SNMPTT consente anche delle configurazioni più complesse, quali ad esempio:

- la possibilità di accettare o rifiutare una trap sulla base del nome dell'host, dell'indirizzo IP, di un intervallo di rete, o del valore di una o più variabili contenute nella stessa trap ricevuta;
- l'esecuzione di programmi esterni per l'invio di messaggi o l'inoltro a software terzi;
- l'esecuzione di operazioni di search & replace basate su espressioni regolari.

Nella figura seguente viene illustrato come avviene l'interazione tra un generico dispositivo di rete, NET-SNMP e SNMPTT al momento della elaborazione di una trap.

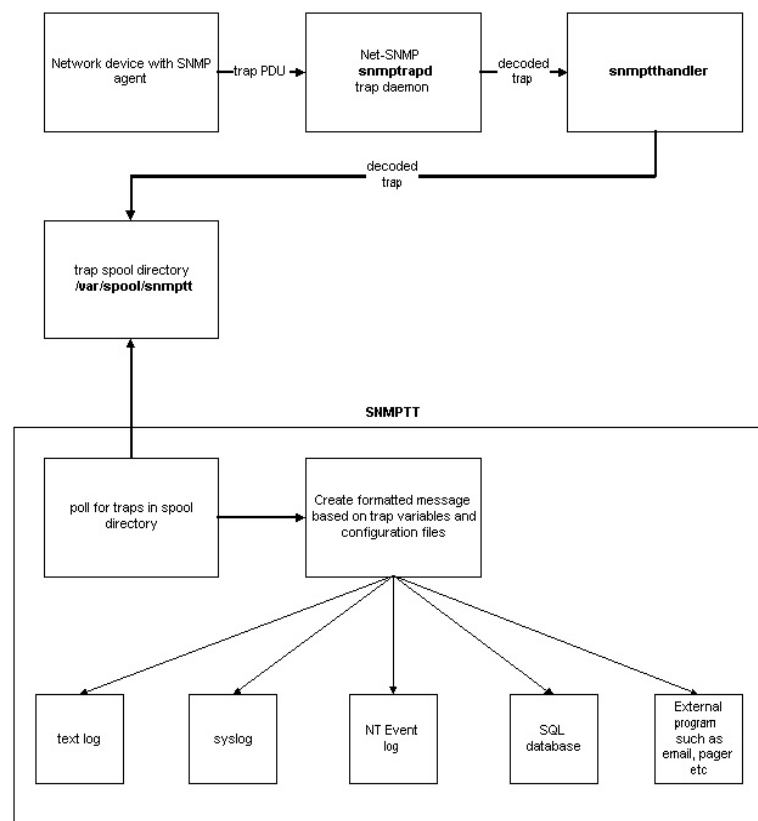


FIGURA 3.4: FLUSSO DEI DATI IN SNMPTT

⁵ Come si vedrà in seguito, il programma *zabbix_sender* verrà in realtà invocato tramite due opportuni script in bash che fungono da wrapper.

Nel caso in specie, il file di configurazione di default di SNMPTT (*/etc/snmp/snmpd.conf*) è stato ampliato in modo da includere due ulteriore file di configurazione, *snmpd.conf.snort* e *snmpd.conf.ossec*, nei quali:

- si esplicitano quali siano i file MIB di riferimento per l'interpretazione delle trap in arrivo;
- si definiscono le trap denominate *sidaAlertGeneric-2* ed *ossecTrap*, caratterizzate rispettivamente dagli OID *.1.3.6.1.4.1.10234.2.1.3.3* e *.1.3.6.1.4.1.6101.5000.2.0.1*;
- si descrive come debba essere formattato il messaggio in uscita;
- si eseguono gli script */usr/local/bin/zsw* e */usr/local/bin/zow* passando come parametro nella linea di comando la stringa risultato della formattazione.

3.4.6 GLI SCRIPT ZSW E ZOW

I programmi *zsw* e *zow* richiamati da SNMPTT all'arrivo di una trap non sono altri che dei piccoli script di shell bash che fanno da wrapper per il programma *zabbix_sender*, che viene invocato avendo prima provveduto al pre-processing ed all'estrazione delle informazioni indispensabili per identificare univocamente il nome del sensore Snort o dell'host monitorato da Ossec che ha originato la trap, l'identificativo della regola che ha provocato l>alert di sicurezza, il messaggio vero e proprio contenuto nella trap ricevuta.

Al fine di garantire una corrispondenza biunivoca tra gli identificativi degli alert generati da Snort o da Ossec e le denominazioni degli item definiti all'interno di Zabbix, vengono aggiunti agli identificativi anzidetti i prefissi "*sid*" e "*oid*", rispettivamente per Snort e Ossec

In tal modo si riesce a parametrizzare totalmente l'invocazione del programma *zabbix_sender*.

3.5 ESTENSIONI AL FRONTEND DI ZABBIX

Zabbix consiste di un demone lato server, che raccoglie le informazioni dagli host/dispositivi configurati tramite l'apposito agent o via SNMP e li memorizza in un database MySQL, PostgreSQL, Oracle o SQLite, e di un'interfaccia GUI di amministrazione e configurazione web-based.

Le nuove funzionalità implementate hanno riguardato sia gli ambiti di presentazione (*Dashboard*, *IDS Events*, *Most Busy IDS triggers top 100*), che di configurazione (*Import*, *SNMP Builder*).

Data l'architettura implementativa di Zabbix, basata su dbms di backend, demone `zabbix_server` e frontend PHP, gli interventi di codifica hanno riguardato esclusivamente la GUI web-based.

Particolare accento è stato posto nell'evidenziare le informazioni relative agli eventi rilevati dai sensori HIDS e NIDS, separandole dalle altre già gestite normalmente da Zabbix.

La distinzione tra eventi di natura IDS ed eventi "normali" è stata implementata inserendo nella definizione delle classi `CEvent` e `CTrigger` e nella `make_latest_secissues` richiamata dalla Dashboard l'attivazione della condizione

```
AND isIdsTrigger(triggerID)>0
```

nella clausola `WHERE` di una generica query.

La funzione `isIdsTrigger` è una stored procedure che ritorna valore positivo ogniqualvolta l'identificativo del trigger passato come parametro sia relativo ad un trigger di natura IDS, cioè riconducibile ai template *Template_OssecRules* o *Template_SnortRules*.

Per rendere accessibili le funzionalità introdotte ex-novo (*IDS Events*, *Most Busy IDS triggers top 100*, *SNMP Builder*) si è reso altresì necessario intervenire sul file `include /var/www/html/include/menu.inc.php`.

Il menu è implementato nel frontend come array composto da ulteriori array annidati; di conseguenza sono state aggiunte le seguenti istruzioni per le nuove voci:

```
.
.
.
    array(
        'url'=>'idsevents.php',
        'label'=>'IDS Events',
        'sub_pages'=>array('tr_events.php')
    ),
.
.
.
    array(
        'url'=>'idsreport5.php',
        'label'=>'Most busy IDS triggers top 100'
    ),
.
.
.
    array(
        'url' => 'snmp builder.php',
        'label' => 'SNMP Builder'
    )
.
.
.
```

3.5.1 MONITORING -> DASHBOARD

La dashboard è stata estesa con la visualizzazione della nuova tabella “*Last 20 IDS Issues*”, che mostra le ultime venti tipologie di eventi di sicurezza rilevati, ordinate su base temporale in senso decrescente.

La particolarità in questo caso risiede nell’esplicitare anche il valore relativo all’evento verificatosi, in modo tale da avere immediatamente evidenza di quanto effettivamente rilevato dai sensori IDS.

Host/Sensor	Issue	Value	Last change	Age	Ack	Actions
n201.pa.icar.cnr.it	HIDS Login session closed.	Jul 15 16:08:01 n201 sshd[27185]: pam_un ...	15 Jul 2010 16:08:03	19m 40s	No	-
n202.pa.icar.cnr.it	HIDS Login session closed.	Jul 15 16:07:56 n202 sshd[16061]: pam_un ...	15 Jul 2010 16:07:59	19m 44s	No	-
n201.pa.icar.cnr.it	NIDS SNMP trap udp	147.163.3.202:53325 -> 147.163.3.201:162	15 Jul 2010 16:07:59	19m 44s	No	-
n201.pa.icar.cnr.it	NIDS ICMP Timestamp Request	147.163.1.41 -> 147.163.3.73	15 Jul 2010 16:01:07	26m 36s	No	-
n201.pa.icar.cnr.it	NIDS ET SCAN NMAP -sA (2)	147.163.1.41:56242 -> 147.163.3.73:80	15 Jul 2010 16:01:01	26m 42s	No	-
n201.pa.icar.cnr.it	NIDS ET SCAN NMAP -sS	147.163.1.41:56243 -> 147.163.3.73:443	15 Jul 2010 16:00:59	26m 44s	No	-
n201.pa.icar.cnr.it	NIDS ET SCAN NMAP -f -sS	147.163.1.41:56243 -> 147.163.3.73:443	15 Jul 2010 16:00:57	26m 46s	No	-
n201.pa.icar.cnr.it	NIDS ICMP PING NMAP	147.163.1.41 -> 147.163.3.73	15 Jul 2010 16:00:49	26m 55s	No	-
n202.pa.icar.cnr.it	HIDS Login session opened.	Jul 15 15:38:08 n202 sshd[16061]: pam_un ...	15 Jul 2010 15:38:11	49m 33s	No	-
n202.pa.icar.cnr.it	HIDS SSHD authentication success.	Jul 15 15:38:08 n202 sshd[16061]: pam_un ...	15 Jul 2010 15:38:10	49m 34s	No	-
n201.pa.icar.cnr.it	HIDS SSHD authentication success.	Jul 15 15:33:53 n201 sshd[27284]: pam_un ...	15 Jul 2010 15:33:56	53m 48s	No	-
n201.pa.icar.cnr.it	HIDS Login session opened.	Jul 15 15:33:53 n201 sshd[27284]: pam_un ...	15 Jul 2010 15:33:55	53m 49s	No	-
n201.pa.icar.cnr.it	HIDS Interface entered in promiscuous(sniffing) mode.	Jul 15 15:24:06 n201 kernel: device eth1 ...	15 Jul 2010 15:24:10	1h 3m 34s	No	-
n202.pa.icar.cnr.it	HIDS Ossec server started.	ossec: Ossec started.	15 Jul 2010 13:34:07	2h 53m 37s	No	-
n201.pa.icar.cnr.it	HIDS Yum package updated.	Jul 15 10:17:12 n201 yum: Updated: 1:net ...	15 Jul 2010 10:17:14	6h 10m 30s	No	-
n201.pa.icar.cnr.it	HIDS New Yum package installed.	Jul 15 10:13:59 n201 yum: Installed: pcr ...	15 Jul 2010 10:14:00	6h 13m 44s	No	-
n201.pa.icar.cnr.it	HIDS Attempt to login using a non-existent user	Jul 15 01:28:48 n201 sshd[32581]: Failed ...	15 Jul 2010 01:28:55	14h 58m 49s	No	-
n201.pa.icar.cnr.it	HIDS User login failed.	Jul 15 01:28:46 n201 sshd[32581]: pam_su ...	15 Jul 2010 01:28:48	14h 58m 56s	No	-
n201.pa.icar.cnr.it	HIDS Attempt to login with an invalid user.	Jul 15 01:28:46 n201 sshd[32581]: pam_su ...	15 Jul 2010 01:28:48	14h 58m 56s	No	-
n201.pa.icar.cnr.it	HIDS Ossec server started.	ossec: Ossec started.	14 Jul 2010 21:59:54	18h 27m 50s	No	-

FIGURA 3.5: ZABBIX - DASHBOARD: LAST 20 IDS ISSUES

Il codice necessario per la sua presentazione è simile a quello relativo alla tabella “*Last 20 Issues*”, con la differenza che in questo caso viene indirettamente richiamata la nuova funzione `make_latest_secissues`⁶, che effettua la query vera e propria attivando nella clausola WHERE la condizione `isIdsTrigger`.

⁶ Implementata in `/var/www/html/include/block.inc.php` e derivata dalla funzione `make_latest_issues` esistente.

3.5.2 MONITORING -> IDS EVENTS E REPORTS -> MOST BUSY IDS TRIGGERS TOP 100

Come accennato precedentemente, *IDS Events* e *Most Busy IDS Triggers top 100* sono due delle nuove pagine implementate per le quali è stato necessario definire delle nuove voci di menu.

Anche in questi due casi, in modo simile a quanto fatto per l'introduzione del blocco *Last 20 IDS issues* nella Dashboard, si è esteso il codice pre-esistente contenuto nei due file *events.php* e *reports5.php*, ottenendo da questi, rispettivamente, i nuovi file *idsevents.php* e *idsreports5.php*.

In particolare, riguardo a *IDS Events*, oltre all'utilizzo della stored procedure *isSnortTrigger*, per ogni evento viene anche visualizzato il suo valore, come si può evincere dalla seguente figura:

Time	Description	Value	Status	Severity
2010.Jul.15 16:08:03	HIDS Login session closed.	Jul 15 16:08:01 n201 sshd[27185]: pam_un ...	PROBLEM	Warning
2010.Jul.15 16:07:59	NIDS SNMP trap udp	147.163.3.202:53325 -> 147.163.3.201:162	PROBLEM	Warning
2010.Jul.15 16:07:56	HIDS Login session closed.	Jul 15 16:07:54 n201 sshd[27284]: pam_un ...	PROBLEM	Warning
2010.Jul.15 16:01:07	NIDS ICMP Timestamp Request	147.163.1.41 -> 147.163.3.73	PROBLEM	Warning
2010.Jul.15 16:01:07	NIDS ICMP Timestamp Request	147.163.1.41 -> 147.163.3.73	PROBLEM	Warning
2010.Jul.15 16:00:57	NIDS ET SCAN NMAP -f -sS	147.163.1.41:56243 -> 147.163.3.73:443	PROBLEM	Warning
2010.Jul.15 16:01:01	NIDS ET SCAN NMAP -sA (2)	147.163.1.41:56242 -> 147.163.3.73:80	PROBLEM	Warning
2010.Jul.15 16:00:59	NIDS ET SCAN NMAP -sS	147.163.1.41:56243 -> 147.163.3.73:443	PROBLEM	Warning
2010.Jul.15 16:00:49	NIDS ICMP PING NMAP	147.163.1.41 -> 147.163.3.73	PROBLEM	Warning
2010.Jul.15 16:00:47	NIDS ICMP PING NMAP	147.163.1.41 -> 147.163.3.73	PROBLEM	Warning
2010.Jul.15 16:00:16	NIDS ICMP PING NMAP	147.163.1.41 -> 147.163.3.242	PROBLEM	Warning
2010.Jul.15 16:00:16	NIDS ICMP PING NMAP	147.163.1.41 -> 147.163.3.242	PROBLEM	Warning
2010.Jul.15 16:00:16	NIDS ICMP PING NMAP	147.163.1.41 -> 147.163.3.242	PROBLEM	Warning
2010.Jul.15 16:00:16	NIDS ICMP PING NMAP	147.163.1.41 -> 147.163.3.242	PROBLEM	Warning
2010.Jul.15 16:00:16	NIDS ICMP PING NMAP	147.163.1.41 -> 147.163.3.242	PROBLEM	Warning
2010.Jul.15 16:00:15	NIDS ICMP PING NMAP	147.163.1.41 -> 147.163.3.52	PROBLEM	Warning
2010.Jul.15 16:00:14	NIDS ICMP PING NMAP	147.163.1.41 -> 147.163.3.37	PROBLEM	Warning
2010.Jul.15 16:00:14	NIDS ICMP PING NMAP	147.163.1.41 -> 147.163.3.37	PROBLEM	Warning
2010.Jul.15 16:00:10	NIDS ICMP PING NMAP	147.163.1.41 -> 147.163.3.23	PROBLEM	Warning

FIGURA 3.6: ZABBIX - IDS EVENTS

Host	Trigger	Severity	Number of status changes
n201.pa.icar.cnr.it	NIDS ICMP PING NMAP	Warning	12
n201.pa.icar.cnr.it	HIDS Interface entered in promiscuous(sniffing) mode.	Warning	8
n201.pa.icar.cnr.it	HIDS Login session closed.	Warning	7
n201.pa.icar.cnr.it	HIDS Login session opened.	Warning	6
n201.pa.icar.cnr.it	HIDS SSHD authentication success.	Warning	6
n201.pa.icar.cnr.it	HIDS Yum package updated.	Warning	6
n201.pa.icar.cnr.it	NIDS ET SCAN NMAP -f -sS	Warning	4
n201.pa.icar.cnr.it	NIDS ET SCAN NMAP -sS	Warning	4
n201.pa.icar.cnr.it	NIDS SNMP trap udp	Warning	3
n202.pa.icar.cnr.it	HIDS Attempt to login using a non-existent user	Warning	3
n202.pa.icar.cnr.it	HIDS Ossec server started.	Warning	3
n201.pa.icar.cnr.it	HIDS Attempt to login using a non-existent user	Warning	2
n201.pa.icar.cnr.it	HIDS New Yum package installed.	Warning	2
n201.pa.icar.cnr.it	NIDS ICMP Timestamp Request	Warning	2
n202.pa.icar.cnr.it	HIDS Attempt to login with an invalid user.	Warning	2
n202.pa.icar.cnr.it	HIDS Login session closed.	Warning	2
n202.pa.icar.cnr.it	HIDS User login failed.	Warning	2
n201.pa.icar.cnr.it	HIDS Attempt to login with an invalid user.	Warning	1
n201.pa.icar.cnr.it	HIDS Ossec server started.	Warning	1
n201.pa.icar.cnr.it	HIDS User login failed.	Warning	1
n201.pa.icar.cnr.it	NIDS ET SCAN NMAP -sA (2)	Warning	1
n201.pa.icar.cnr.it	HIDS Tenable shadowcat checked	Warning	1

FIGURA 3.7: ZABBIX - MOST BUSY IDS TRIGGERS TOP 100

3.5.3 CONFIGURATION -> IMPORT

La programmabilità della piattaforma, che in questo contesto equivale alla facoltà di saper riconoscere un evento di tipo IDS, è stata implementata come funzionalità nell'ambito delle procedure di Import.

The screenshot shows the Zabbix web interface for the 'Import' configuration page. The page is divided into three main sections, each with a title bar and a help icon:

- Import:** Contains an 'Import file' field with a 'Sfogliala...' button. Below it is a table with columns 'Element', 'Update Existing', and 'Add Missing'. The rows are: Host (Update Existing: checked, Add Missing: checked), Template linkage (Update Existing: checked, Add Missing: checked), Item (Update Existing: checked, Add Missing: checked), Trigger (Update Existing: checked, Add Missing: checked), and Graph (Update Existing: checked, Add Missing: checked). An 'Import' button is at the bottom right.
- NIDS Rules Import:** Contains a 'Snort Rules File' field with a 'Sfogliala...' button and a 'Snort Rules Group' text input field. An 'Import' button is at the bottom right.
- HIDS Rules Import:** Contains an 'Ossec Rules File' field with a 'Sfogliala...' button and an 'Ossec Rules Group' text input field. An 'Import' button is at the bottom right.

At the bottom of the page, there is a footer: 'Zabbix 1.8.2 Copyright 2001-2010 by SIA Zabbix'.

FIGURA 3.8: ZABBIX - IMPORT

Le funzionalità native di Import di Zabbix consentono la creazione/aggiornamento delle seguenti tipologie di oggetti:

- Host;
- Template linkage;
- Item;
- Trigger;
- Graph.

le cui definizioni possono essere indicate all'interno di un file XML⁷, che nel caso di un host sarà del tipo:

```
<?xml version="1.0"?>
<zabbix_export version="1.0" date="11.05.07" time="11.11">
  <hosts>
    <host name="ZABBIX Server">
      <useip>1</useip>
      <ip>127.0.0.1</ip>
      <port>10050</port>
      <status>1</status>
      <groups>
      </groups>
      <items>
        <item type="0" key="agent.ping" value type="3">
          <description>Ping to the server (TCP)</description>
          <delay>30</delay>
          <history>7</history>
          <trends>365</trends>
          <snmp port>161</snmp port>
          <valuemap>Service state</valuemap>
          <applications>
            <application>General</application>
          </applications>
        </item>
        ....
      </items>
      <triggers>
        <trigger>
          <description>Trigger description on {HOSTNAME}</description>
          <expression>{{HOSTNAME}:agent.version.diff(0)}&gt;0</expression>
          <priority>3</priority>
        </trigger>
        ....
      </triggers>
    </host>
    ....
  </hosts>
</zabbix_export>
```

Poiché gli eventi IDS che potenzialmente possono verificarsi sono complessivamente, tra NIDS e HIDS, circa 10.000, e poiché a ciascuno di esso, etichettato da un identificativo unico, è associato un singolo item, si capisce bene quanto possa essere impraticabile la strada della loro definizione manuale all'interno di Zabbix.

⁷ Per ulteriori approfondimenti si faccia riferimento alla sezione “XML Import and Export” della documentazione online di Zabbix (http://www.zabbix.com/documentation/1.8/manual/xml_export_import).

Sono state quindi sviluppate delle procedure d'importazione che vanno a operare direttamente sui file delle regole di Snort e di OSSEC e che automaticamente creano, all'interno dei template `Template_SnortRules` e `Template_OssecRules`, per ciascuna regola incontrata un item e relativo trigger.

Le procedure d'importazione eseguono, di fatto, un pre-processing dei file d'ingresso e ottengono come risultato un file temporaneo in formato XML Zabbix, che poi viene processato in modo trasparente all'utente.

Il pre-processing è fondamentale perché, benché le definizioni delle regole sia in Snort che in Ossec siano contenute in file di testo, queste o non sono date in formato XML, o se lo sono (come nel caso di Ossec) il formato non è standard.

Infatti, il contenuto di un tipico file di regole di Snort è del tipo:

```
...
alert tcp $EXTERNAL_NET any -> $HOME_NET 79 (msg:"FINGER cmd_rootsh backdoor attempt";
flow:to_server,established; content:"cmd_rootsh"; metadata:service finger;
classtype:attempted-admin; sid:320; rev:11;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 79 (msg:"FINGER account enumeration attempt";
flow:to_server,established; content:"a b c d e f"; nocase; metadata:service finger;
classtype:attempted-recon; sid:321; rev:6;)
...
```

mentre per Ossec si ha:

```
...
<group name="local,syslog,">
  <rule id="100001" level="0">
    <if sid>5711</if sid>
    <srcip>1.1.1.1</srcip>
    <description>Example of rule that will ignore sshd </description>
    <description>failed logins from IP 1.1.1.1.</description>
  </rule>
</group>
...
<group name="another,syslog,">
...
</group>
...
```

3.5.4 CONFIGURATION -> SNMP BUILDER

Normalmente in Zabbix la creazione di nuovi template per dispositivi SNMP richiede l'uso preventivo di un MIB browser per tenere traccia degli OID corretti da utilizzare in fase di creazione degli item.

Quando il numero di item da creare è considerevole, questa modalità diventa sicuramente poco pratica e molto tediosa.

Un primo approccio per automatizzare il processo di creazione di template SNMP custom potrebbe essere quello di utilizzare per l'importazione degli OID in item di Zabbix lo script

zload_snmpwalk (http://zabbix.com/wiki/howto/monitor/snmp/zload_snmpwalk). Si tratta di uno script in perl, del tutto esterno alla piattaforma Zabbix e invocabile esclusivamente da linea di comando.

Nell'ottica però di un utilizzo più generalistico di Zabbix, sarebbe invece sicuramente alquanto più utile poter disporre di uno strumento integrato nella GUI web-based.

Per tale ragione si è scelto di integrare nel frontend di Zabbix il tool *SNMP Builder* (<http://github.com/giapnguyen/snmpbuilder>), che consente il browse delle MIB alla ricerca di valori ed informazioni, convertendoli automaticamente in item Zabbix ed inserendoli in un template prefissato.

L'add-on è dotato delle seguenti funzionalità:

- **MIB Browser:** per ottenere un OID tree, si può selezionare uno dei file di MIB predefiniti oppure uno di quelli relativi al particolare dispositivo che s'intende definire. Cliccando su un nodo dell'albero si ottengono i valori e le informazioni riguardanti uno specifico OID, dopodiché la sua scelta comporta la definizione di un corrispondente item;
- **Selezione di colonne:** in una tabella OID, il cliccare su di un'intestazione di una colonna, ne provoca la sua intera selezione come item di Zabbix. Ciò può tornare utile, ad esempio, quando si devono creare template SNMP per switch a 48 porte.
- **Conversione automatica:** semplice conversione dai tipi di dati SNMP ai tipi di item Zabbix.

PhysAddress	ifAdminStatus	ifOperStatus	ifLastChange	ifInOctets	ifInUcastPkts	ifInUcastPkts	ifInDiscards	ifInErr
:22:55:80:57:81	up	down	0:0:00:25.32	17439	98	0	0	0
:22:55:80:57:82	up	down	0:0:00:28.32	0	0	0	0	0
:22:55:80:57:83	up	up	0:0:00:28.31	2724012339	4846870	338558	0	0
:22:55:80:57:84	up	up	0:0:00:28.31	385295599	4470295	32184	0	137
	up	up	0:0:08:49.24	2568240591	4173579	0	0	0
	up	up	0:0:00:00.00	0	0	0	0	0
	up	up	0:0:08:46.24	?	?	?	?	?
	up	up	0:0:08:46.24	?	?	?	?	?
	up	up	0:0:08:46.24	2584993349	4174484	?	0	0
	up	up	0:0:08:46.24	2584993349	4174484	?	0	0

Key	Type	Data	Unit	Interval	Multiple	Delta
IF-MIB::ifInOctets.1	Numeric	Decimal		60		Yes
IF-MIB::ifInOctets.3	Numeric	Decimal		60		Yes
IF-MIB::ifInOctets.4	Numeric	Decimal		60		Yes

FIGURA 3.9: ZABBIX - SNMP BUILDER

4 CONCLUSIONI

4.1 INTRODUZIONE

I risultati ottenuti hanno dimostrato la riuscita e piena integrazione tra Zabbix e gli strumenti d'intrusion detection Snort e OSSEC, utilizzando SNMP come protocollo di comunicazione tra le parti.

Questo risultato coincide esattamente con l'obiettivo che ci si era inizialmente prefissato: “Verificare la fattibilità dell'integrazione di sistemi di acquisizione delle informazioni con sistemi di elaborazione e presentazione, attraverso l'uso di un protocollo comune, al fine di pervenire a una soluzione completa per il Service Management”.

Nell'architettura implementata in questa sede, quindi, le componenti oggetto dell'obiettivo possono essere definite nel seguente modo:

- **Service Management:** Intrusion Detection Management (IDM)
- **Sistemi di acquisizione delle informazioni:** Snort e OSSEC, rispettivamente un NIDS e un HIDS;
- **Sistema di elaborazione e presentazione:** Zabbix, piattaforma Open Source per il network e system management di livello enterprise;
- **Protocollo comune:** SNMP.

4.2 ULTERIORI POSSIBILI SVILUPPI DELLA PIATTAFORMA IDM

Considerato quanto realizzato, è chiaro che la piattaforma implementata avrebbe bisogno di ulteriori interventi ed estensioni prima di poter essere utilizzata come “vero” Intrusion Detection Management System in ambiente di produzione.

La componente “reattiva”, ad esempio, consiste oggi in un semplice script dimostrativo che scrive semplicemente su file di log le informazioni relative all'identificativo della regola che ha scatenato l>alert sulla sonda, l'identificativo del trigger di Zabbix, gli indirizzi IP sorgente e destinazione.

Nonostante ciò, si può facilmente intuire che il suddetto script possa essere utilizzato come base per la scrittura di un wrapper nei confronti di tool di azione/reazione più sofisticati e realmente utilizzabili.

Un esempio di tali strumenti di reazione è dato da SnortSam che, benché nasca sostanzialmente come plug-in per Snort, può anche essere utilizzato in configurazione

client/server, potendo quindi intervenire direttamente eseguendo il blocco di indirizzi IP sui seguenti (e non solo) firewall:

- Checkpoint Firewall-1;
- Cisco PIX;
- Router Cisco (mediante ACL o Null-Routes);
- Juniper Firewall;
- IP Filter (ipf);
- FreeBSD ipfw2;
- OpenBSD Packet Filter (pf);
- Linux IPchains/IPtables/EBtables;
- WatchGuard Firewall;
- Microsoft ISA Server.

La componente agent di SnortSam, da eseguire sul firewall o su un host ad esso collegato, consente di sfruttare svariate ulteriori funzionalità che vanno ben oltre il meccanismo di blocco automatico, quali ad esempio:

- Supporto di white list di indirizzi IP;
- Finestre temporali di blocco, anche per quei firewall che non supportano i ban a tempo;
- Liste di SID consentiti/negati;
- Multi-threading per il blocco simultaneo su dispositivi diversi;

Come si è visto, le funzionalità di importazione operano direttamente sui file di regole standard di Snort ed OSSEC.

Si potrebbe, quindi, pensare di utilizzare la piattaforma anche per il deploy di nuove regole direttamente sui vari sensori, ottenendo in tal modo una completa gestione centralizzata degli stessi.

Oppure, ad esempio, si potrebbero integrare ulteriori funzionalità di reportistica avanzata, e volendo anche di business intelligence, utilizzando strumenti specializzati che possono essere facilmente integrati, come JasperReports.

*JasperReports*⁸ è un motore di reporting Open Source realizzato in Java. E' in grado di utilizzare dati prelevati da qualunque tipo di data source e di produrre documenti di report che possono essere visualizzati, stampati ed esportati in diversi formati di documento come HTML, PDF, Excel, OpenOffice e Word.

⁸ <http://jasperforge.org/projects/jasperreports>

4.3 APPLICABILITÀ DELLA SOLUZIONE AD ALTRI AMBITI OPERATIVI

La possibilità di realizzazione e utilizzo di nuove MIB SNMP ha consentito l'interfacciamento al sistema di management di due nuove categorie di "sensori/dispositivi": gli strumenti di intrusion detection host-based e network-based.

Quindi, il protocollo SNMP consente potenzialmente, grazie alle sue caratteristiche d'interoperabilità e di espandibilità, la misurazione e il controllo di qualunque tipo di dispositivo o servizio accessibile da un'infrastruttura di rete informatica, permettendo l'utilizzo di un tradizionale sistema di network e system management, eventualmente opportunamente esteso nelle sue funzionalità, anche in ambiti diversi dall'Information and Communication Technology.

Si è già fatto cenno in Introduzione all'esistenza di dispositivi interrogabili via SNMP da impiegare nei campi della Domotica e Building Automation o nel Monitoring Ambientale.

Ulteriori campi di applicabilità potrebbero, ad esempio, anche essere:

- **Hospitality**, grazie al controllo e monitoraggio di switch VDSL (Very High Bit Rate Subscriber Line). VDSL è una tecnologia che consente di offrire agli utenti accessi a banda larga sfruttando il normale cablaggio telefonico in rame e viene utilizzata tipicamente all'interno delle strutture alberghiere, ma è applicabile anche a strutture quali ospedali, università, musei, caserme militari, supermercati, stabilimenti produttivi e logistici;
- **Telemedicina**, mediante, ad esempio, sistemi integrati di videocomunicazione e teleconsulto, che consentono di gestire i dati anagrafici e sanitari di pazienti provenienti da apparati elettromedicali. Questi sistemi, monitorabili anche via SNMP, sono progettati e sviluppati per supportare la consultazione medica remota e la formazione a distanza, gestiscono referti medici elettronici, elaborano e possono acquisire immagini mediche anche radiografiche, forniscono, infine, la semplice connessione a un'ampia gamma di periferiche mediche e apparecchiature, inclusi: monitor per i segnali vitali, stetoscopi digitali, scanner radiografici, ECG e telecamere digitali;
- **Medicale**, come ad esempio per il controllo della qualità delle postazioni di lavoro PACS, in termini di verifica dello stato dei display (temperatura, durata della retroilluminazione, luminosità attuale), controllo dell'uniformità della luminanza, e tutte quelle altre informazioni utili per la garanzia della qualità.
- **(Decision Support Systems DSS)**, ai quali piattaforme del tipo in esame potrebbero fornire dati "filtrati" mediante i meccanismi di azione/reazione. In tal modo, a partire da un qualunque tipo d'informazione misurabile mediante protocollo SNMP, è possibile ottenere un'integrazione tra un sistema deterministico ed uno non deterministico, quale appunto un DSS.

5 BIBLIOGRAFIA

- Baker, Max. "*NETDISCO - Network Management Tool*". 2010. <http://netdisco.org/>.
- Balsamo, S. "*Il protocollo SNMP*" - *Approfondimento per il corso di Reti di Calcolatori*. Università degli Studi Ca' Foscari di Venezia, 2004.
- Burger, Alex. "*SNMP Trap Translator*". 2010. <http://www.snmpptt.org/>.
- Cacti. "*Cacti: The Complete RRDTool-based Graphing Solution*". 2010. <http://www.cacti.net>.
- Curry, Jane. "Open Source Management Options." *Skills 1st consultancy: directory services, network and systems management*. 2008. <http://www.skills-1st.co.uk>.
- Cyber Solutions Inc. "*SnortSNMP*". 2010. <http://www.cysol.co.jp/contrib/snortsnmp/index.html>.
- Knobbe, Frank. "*Snortsam - A Firewall Blocking Agent for Snort*". 2010. <http://www.snortsam.net/>.
- Nagios Enterprises. "*Nagios - The Industry Standard in IT Infrastructure Monitoring*". 2010. <http://www.nagios.org/>.
- Net-SNMP Community. "*Net-SNMP*". 2010. <http://www.net-snmp.org/>.
- Oetiker, T. "*RRDtool high performance data logging and graphing system for time series*". 2010. <http://oss.oetiker.ch/rrdtool/>.
- Oetiker, Tobi. "*MRTG - The Multi Router Traffic Grapher*". 2010. <http://oss.oetiker.ch/mrtg/>.
- RFC-1155. "*Structure and Identification of Management Information for TCP/IP-based Internets*". 1990.
- RFC-1157. "*A Simple Network Management Protocol (SNMP)*". 1990.
- RFC-1212. "*Concise MIB Definitions*". 1991.
- RFC-1213. "*Management Information Base for Network Management of TCP/IP-based internets: MIB-II*". 1991.
- RFC-1215. "*Convention for defining traps for use with the SNMP*". 1991.
- RFC-1514. "*Host Resources MIB*". 1993.
- RFC-2263. "*SNMPv3 Applications*". 1998.
- RFC-2578. "*Structure of Management Information Version 2 (SMIV2)*". 1999.

- RFC-2579. *"Textual Conventions for SMIV2"*. 1999.
- RFC-2580. *"Conformance Statements for SMIV2"*. 1999.
- RFC-2790. *"Host Resources MIB"*. 2000.
- RFC-3411. *"An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks"*. 2002.
- RFC-3412. *"Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)"*. 2002.
- RFC-3413. *"Simple Network Management Protocol (SNMP) Applications"*. 2002.
- RFC-3414. *"User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)"*. 2002.
- RFC-3415. *"View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)"*. 2002.
- RFC-3416. *"Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)"*. 2002.
- RFC-3417. *"Transport Mappings for the Simple Network Management Protocol (SNMP)"*. 2002.
- RFC-3418. *"Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)"*. 2002.
- Sourcefire Inc. *"Snort"*. 2010. <http://www.snort.org/>.
- The OpenNMS Group. *"OpenNMS"*. 2010. http://www.opennms.org/wiki/Main_Page.
- Trend Micro Inc. *"OSSEC"*. 2010. <http://www.ossec.net/>.
- Zabbix SIA. *"Zabbix :: An Enterprise-Class Open Source Distributed Monitoring Solution"*. 2010. <http://www.zabbix.com/>.
- Zenoss Inc. *"Zenoss Open Source Server and Network Monitoring - Core and Enterprise"*. 2010. <http://www.zenoss.com/>.