UNIVERSITA' DELLA CALABRIA

DIPARTIMENTO DI MATEMATICA ED INFORMATICA

Corso di Laurea Triennale in Informatica

TESI DI LAUREA

Syslog Server per una rete MAN comunitaria

Relatore
Prof. Giovambattista Ianni
Dott. Vincenzo Bruno

Laureando
Mazzitelli Raffaele
matr. 106698

Indice generale

Introduzione

Una rete è un sistema di dispositivi, interconnessi, in grado di comunicare tra di loro condividendo informazioni e risorse in un'area che può variare di dimensioni e locazioni. Questa architettura si è evoluta grazie al concetto di meccanismo client/server: la presenza di un server permette ad un certo numero di client di condividerne le risorse, lasciando che sia il server a gestire gli accessi ad esse per evitare conflitti di utilizzo tipici dei primi sistemi informatici.

Con sistema “client/server” si intende un'architettura di rete nella quale genericamente un computer client o terminale si connette ad un server per la fruizione di un certo servizio ed è su questo che si basa Internet.

Internet è una rete di telecomunicazioni capace di mettere in comunicazione utenti situati in diverse parti del mondo e dialogare in tempo reale a costi ridotti.

Una rete internet è composta da terminali chiamati “host” utilizzabili dagli utenti per accedere ad una vasta gamma di informazioni (il classico esempio è il PC di casa collegato ad Internet tramite un modem), e nodi intermedi a cui è deputato l'instradamento del traffico, chiamati comunemente “router". Host e router sono collegati mediante reti eterogenee del tutto indipendenti: LAN, MAN, Point-to-Point in fibra ottica o cavo coassiale, reti ISDN, reti Frame Relay, reti ATM, reti Wireless.

Internet arriva nelle nostre case sfruttando la rete telefonica già esistente, tramite fibra ottica e cavi in rame o via cavo con la sottoscrizione di un abbonamento dal nostro Internet Service Provider (ISP), che è appunto quell'ente in grado di fornire il servizio.

La rete sulla quale il presente lavoro si basa è la rete Ninux: Ninux è una Wireless Community Network e, in quanto tale, ha tra i suoi obiettivi la costruzione di una rete libera di proprietà dei cittadini. Il modello è semplice: ogni partecipante si connette ai suoi vicini che si connettono a loro volta ai loro vicini e così via, creando una rete di computer.

Questa rete è di proprietà dei cittadini perché ogni partecipante è il proprietario e il responsabile del proprio nodo della rete. Non c’è nessun Internet Service Provider, nessun

abbonamento mensile, nessun contratto, nessuna registrazione.

A causa della grande diffusione dei PC e, di conseguenza, del numero di utenti che la utilizzano, dai semplici utilizzatori domestici che posseggono più di due dispositivi agli amministratori di reti più complesse, si è posto il problema di dover monitorare il traffico che passa su internet.

Le soluzioni di networking attualmente disponibili, che siano open source o commerciali, includono una serie di prodotti messi a disposizione all'amministratore della rete per gestire, controllare e mantenere la propria infrastruttura sicura e sotto determinati standard. Usando questi software è possibile controllare i tratti di rete che si vogliono monitorare nonché le prestazioni e la qualità della rete stessa.

L'importanza di monitorare il traffico di rete è conseguenza dell'esistenza di molti software in grado di aprire connessioni di rete rendendo le porte disponibili al mondo esterno, questi software hanno il bisogno di utilizzare le porte, ad esempio, per consentire la comunicazione con alcuni servizi come telnet, ssh, ftp, mentre altri software aprono connessioni indispensabili per il proprio funzionamento come il browser o i client di posta elettronica, ma ci sono anche alcuni programmi che aprono in maniera arbitraria le porte del pc per favorire l'accesso, dall'esterno, di utenti non autorizzati. Poiché gli scopi di questi utenti possono essere discutibili se non criminali, è necessario proteggere i sistemi ed imparare a rilevare gli accessi non autorizzati, l'origine e le finalità.

Attraverso, quindi, alcuni strumenti hardware e software, è possibile tenere traccia di tutti gli eventi che accadono all'interno della rete attraverso l'invio dei messaggi, inviati tramite degli "agenti", che l'amministratore di rete deve interpretare con cura. Lo studio di questa tesi si incentra sul monitoring della rete analizzando il traffico dati attraverso messaggi di log grazie al protocollo Syslog e l'uso del software open-source LogAnalyzer.

Per questa tesi sperimentale, che andrà in produzione, l'Hacklab, associazione dedita a numerose attività come corsi supportati da donazione, organizzazione di eventi e attività pubbliche di Cosenza, ha fornito supporto tecnico attraverso computer, raspberry, router cisco per la realizzazione degli esperimenti in laboratorio e nella realtà in quanto fa parte del progetto Ninux e dispone dei server utili allo scopo. Lo scopo è quello di "mettere in piedi"

un sistema di monitoraggio centralizzato della rete e configurare adeguatamente le macchine per poi analizzarne gli eventi.

Inoltre è stato possibile fare delle sperimentazioni e test sul cluster della startup cosentina ASCloud, dedicato alla realizzazione di un private cloud interamente basato su Software Libero.

Il lavoro svolto è stato quello di dare una spiegazione dello strumento, appunto Syslog, fornendo nel capitolo 1 una approfondita introduzione su cos'è, come funziona, perché e in quale ambito viene utilizzato. Nel capitolo successivo è stato analizzato l'RFC5424: il documento ufficiale che tratta del funzionamento e architettura di Syslog. Tale documento è stato, poi, tradotto interamente nell'Appendice. Sempre nel capitolo 2 è stata fatta un'analisi e confronto di alcuni dei software open-source che potrebbero essere utilizzati allo scopo e la scelta effettuata personalmente: Adiscon LogAnalyzer. L'installazione, configurazione ed utilizzo di LogAnalyzer ed rsyslog vengono illustrati nel capitolo 3 insieme alla configurazione ed impiego di tutti gli strumenti hardware forniti dall'Hacklab. Tali apparecchi sono stati essenziali per gli esperimenti in laboratorio descritti nel capitolo 4. Infine, nell'ultimo capitolo, tutti quello fatto nel laboratorio è stato applicato nella vita reale su una rete Ninux, analizzando i risultati attraverso vari tipi di grafici.

Capitolo 1 – Introduzione al Syslog

1.0.1 - La storia

Syslog, diminutivo di System Log, è un protocollo creato nel 1980 da Eric Allman come parte di DeliverMail, poi ribattezzato in Sendmail. Syslog è un formato di logging usato dal Message Transfer Agent (MTA): un software in grado di implementare le porzioni client e server nel trasferimento di email, usato dapprima solamente da sendmail ma, nel tempo, è stato standardizzato ed adattato ad altri protocolli. Per questo motivo l'implementazione di Syslog ha visto una serie di cambiamenti, partendo dall'essere uno standard non ufficiale usato da altri programmi di logging, diventato poi ufficiale grazie al RFC 3164 nel 2001 e arrivando alla standardizzazione dal protocollo IETF e descritto nell'RFC 5424, rendendo obsoleto quello precedente.

1.0.2 - Come opera ed i vantaggi

Syslog è un modo, per i dispositivi di rete, di inviare messaggi di eventi ad un sistema centrale o ad un server chiamato Syslog Server in cui vengono memorizzati. Il protocollo è supportato da una vasta gamma di dispositivi che possono sfruttare questa tecnologia per mandare messaggi, ad esempio, alla console del router. I messaggi, detti messaggi di log, sono delle stringhe di testo formate da vari campi che ne descrivono le caratteristiche: data, ora, contesto, macchina, processo, indirizzo ip e livelli gravità. Quest'ultimi variano da 0 a 7 e sono legati al messaggio indicandone l'urgenza con la quale intervenire.

I vantaggi di implementare un sistema centralizzato di logging sono innumerevoli:

- Migliora drasticamente la capacità di effettuare troubleshooting.
- Mantiene statiche le partizioni delle applicazioni e minimizza l'I/O del disco sul proprio application server. Questo permette di ridurre i costi in quanto è possibile limitare lo spazio del disco necessario.
- Mantiene statici i requisiti di memoria, questo minimizza anche i costi.

- Consente di effettuare ricerche di log fuori dall'ambiente di produzione. Ricercando file di log usando un tool, infatti, richiede l'uso di molte risorse.
- Senza il logging centralizzato, diventa difficile effettuare una ricerca dettagliata dal momento che il personale dovrà accedere ad ogni server e avviare la ricerca.

Quando avviene un determinato evento, Syslog invia tramite pacchetti UDP, un Syslog trap al Server Syslog. Bisogna precisare, però, che quando si parla di trap ci si riferisce a SNMP. La caratteristiche che li accomuna è che entrambi sono usati per mandare alert generati da un evento, ma le informazioni collezionate da sylog vengono inviate ad un syslog server per essere mantenute a lungo termine. Quindi, in sintesi, SNMP è usato per ottenere informazioni in tempo reale, mentre Syslog per informazioni storiche.

In un sistema operativo UNIX, il kernel e altri componenti interni generano messaggi e avvisi. I messaggi sono in genere memorizzati in un file system o trasmessi ad un altro dispositivo, sotto forma di messaggi syslog. Il demone interno, chiamato Syslogd, gestisce il processo syslog. Questo demone è parte integrante di molte distribuzioni UNIX / Linux e non ha, quindi, bisogno di essere scaricato o installato. Syslog offre un punto centrale per la raccolta e l'elaborazione dei log di sistema. I registri di sistema sono poi utili per la risoluzione dei problemi e il controllo. Ad esempio, quando un hacker irrompe in un sistema, il sentiero lasciato dall'attività degli hacker viene registrato nei messaggi syslog. I messaggi possono poi essere utilizzati per comprendere l'attacco, valutare i danni, e correggere il sistema.

Syslog usa il protocollo UDP e la porta 514 per la comunicazione, è un protocollo connectionless e non prevede una “ricevuta di ritorno”. Quindi, a livello applicazione, i server syslog non mandano nessuna notifica, al mandante, di avvenuta ricezione del dato. Di conseguenza il device mandante genera messaggi syslog senza sapere se il server abbia

ricevuto il messaggio.

I messaggi sono categorizzati in base alle sorgenti da cui sono generati. Le sorgenti possono essere sistemi operativi, processi o un'applicazione. Queste categorie, chiamate facility, sono rappresentate da interi alcuni dei quali sono pre-assegnati, mentre altri possono essere impostati per quei processi che sono per uso locale come mostrato nella tabella 1.

0: Kernel Messages	**6:** Line printer subsystem	**12:** NTP subsystem	**18:** Local use 2
1: User-level messages	**7:** Network news subsystem	**13:** Log audit	**19:** Local use 3
2: Mail system	**8:** UUCP subsystem	**14:** Log alert	**20:** Local use 4
3: System daemons	**9:** Clock daemon	**15:** Clock daemon	**21:** Local use 5
4: Security/authorization messages	**10:** Security/authorization messages	**16:** Local use 0	**22:** Local use 6
5: Messages by Syslogd	**11:** FTP daemon	**17:** Local use 1	**23:** Local use 7

Tabella 1: Valori e significato del campo facility

La facility che genera i messaggi syslog specifica anche la gravità (severity) dei messaggi usando degli specifici interi come mostrato nella tabella 2.

0	Emergency: system is inusable	**4**	Warning: warning conditions
1	Alert: action must be taken immediately	**5**	Notification: normal but significant condition
2	Critical: critical conditions	**6**	Informational: informational messages
3	Error: error conditions	**7**	Debug: debug-level messages

Tabella 2: Valori e significato del campo severity

Vi sono numerose facility su cui syslog può filtrare ed includono *auth, auth-priv, cron, daemon, kern, lpr, mail, user, syslog, uucp, da local 0 a local 7 e news.* Anche le severity possono essere filtrate: *emerg, alert, crit, err, warning, notice, info, debug.* Mentre nei sistemi UNIX basta consultare un semplice file di testo tramite shell, su Windows, invece, l'operazione risulta più complessa e per questo ci vengono in aiuto alcuni tool gratuiti,

alcuni disponibili anche per Linux come: Kiwi, SysLog Sender, Syslog Forwarder, Graylog2, Syslog-ng, log2timeline, LogHound, LogReport e molti altri.

1.1 - I contesti dell'uso di Syslog

1.1.0 - Sistemi Operativi

Syslog viene usato su molteplici piattaforme, la sua versatilità offre la possibilità di essere impiegato per diagnosticare problemi di vario genere. Sui sistemi Unix potrebbe essere utilizzato per capire le motivazioni di un malfunzionamento di un'applicazione o di alcune parti del sistema. La prima fonte da consultare per le operazioni di troubleshooting sono i log che consistono in file di testo capaci di tenere traccia di errori e operazioni eseguite dal sistema: login di un utente, cambiamento di password o un messaggio di errore di un programma.

I log di sistema vengono scritti in directory come /var/log e var/adm oppure nel percorso configurato nei singoli programmi.

In quasi tutti i sistemi Unix, il demone *syslogd,* si occupa della gestione dei log tramite il file di configurazione */etc/rsyslog.conf*. Attualmente, le distribuzioni Linux utilizzano sysklogd: una versione evoluta di syslogd in grado di gestire anche il logging del kernel.

In Unix è comune usare un sistema di logging centralizzato, mentre molte distribuzioni usano Syslog su sistemi embedded e potrebbero essercene altre che lo hanno rimpiazzato. Le vecchie release di OpenWrt usavano BusyBox's sysloggd, mentre le più moderne utilizzano ubox's logd e logread. OpenWrt usa il proprio sistema di log implementato come parte di ubox.

Su Windows, l'equivalente dei log, prende il nome di “Eventi”. Di default i log si sfogliano all'interno del Visualizzatore Eventi (Pannello di Controllo – Strumenti di Amministrazione – Visualizzatore Eventi).

In una grande o piccola rete può risultare molto utile riunire i log di tutte le macchine in un unico posto. La potenzialità di ciò sta nel fatto di poter applicare, in un unico intervento,

funzioni di analisi, ricerca e statistica mostrando lo stato di tutte le macchine della rete. In questo modo, quando lo stato di una qualsiasi macchina monitorata ha subìto qualche pericolosa alterazione, sarà possibile configurare in unico strumento capace di inviare messaggi di alert relativi a quella macchina.

Su Linux esistono ottimi strumenti per creare un server di log centralizzato, su Windows, nativamente, non esiste, ma sono stati creati dei tool capaci di tradurre i log in formato standard ed inviarli ad un server centralizzato.

1.1.1 Database

Avere messaggi syslog in un database e spesso alla portata di mano è molto utile soprattutto quando si intende istituire un front-end per la loro visualizzazione.

In molti caso, i dati di syslog vengono scritti i semplici file di testo e ciò approccio presenta alcuni vantaggi come, ad esempio, la velocità ed efficienza. Tuttavia, i dati memorizzati nei file di testo non sono accessibili alla lettura per l'analisi in tempo reale, quindi i messaggi devono essere collocati in un database.

1.1.2 Networking

In una rete l'uso di Syslog si rileva essenziale quando avvengono determinate situazioni.

I device che ne fanno parte, come router, firewall, switch ecc, e che supportano Syslog, inviano dei messaggi ad un server syslog. Un amministratore di rete ha una varietà di opzioni per memorizzare e visualizzare questi messaggi, oltre ad essere avvisato quando avviene un problema.

Sui device di rete Cisco, il protocollo Syslog manda, inizialmente, messaggi di sistema e di debug riguardo un processo di logging interno. Quest'ultimo gestisce i messaggi in base alla propria configurazione ma, i messaggi di debug sono gestibili solo tramite linea di comando.

1.1.3 Posta Elettronica

Syslog nacque nel 1980 come componente di sandmail, agente di trasferimento per la posta elettronica su Unix.

I messaggi riferiti alla gestione della posta elettronica sono memorizzati nel file */var/log/maillog* nei sistemi Linux. Il programma login, dopo la visualizzazione del messaggio contenuto in */etc/motd*, se trovasse la presenza di posta per l'utente visualizzerà un messaggio di avvertimento.

La gestione dei file che rappresentano la posta elettronica dell'utente dipende dalla configurazione di sistema di gestione della posta. In genere si fa affidamento sull'utilizzo di SendMail che utilizza la directory */var/mail/*.

1.1.4 Logging as a Service

Il logging as a Service è un modello architetturale per la raccolta di ogni tipo di file di log da qualsiasi fonte o sorgente come: server, applicazioni o device. I file vengono filtrati per essere riformattati e spediti verso altri sistemi per essere processati come dati "puri" e con il quale vengono gestiti, mostrati ed elaborati in base ad un determinato numero di criteri.

1.1.5 SNMP

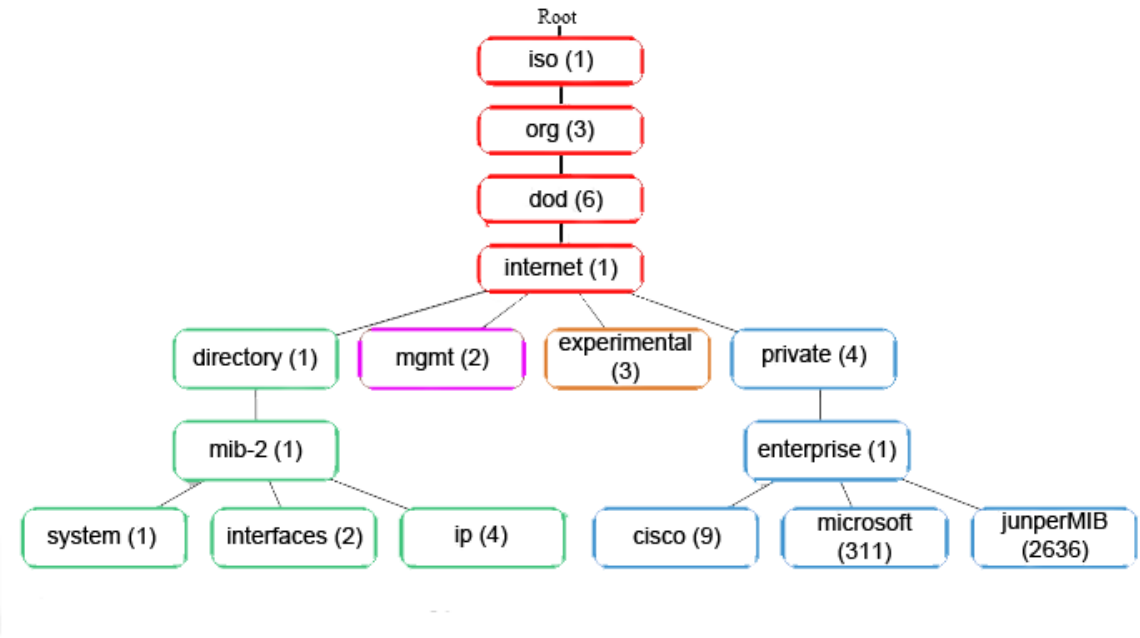
Simple Network Management Protocol (SNMP) nasce dalla precedente versione SGMP che definisce alcuni elementi come il tipo di interfaccia ed il suo stato, successivamente divenuto più complesso nel 1990 e definito da IETF.

SNMP è un protocollo di rete che permette agli amministratori di network di supervisionare, gestire e configurare gli apparati di rete per tutto quello che concerne il monitoring ed il management dei dispositivi. Tale sistema di gestione incorpora un supervisore e degli agenti: il supervisore, o sistema di management della rete (NMS), è la console che permette all'amministratore di rete di eseguire richieste di management, gli agenti sono delle entità capaci di dialogare con il manager ed attuare le decisioni, come ad

esempio recuperare delle informazioni da diversi oggetti in formato SNMP.

Nell'architettura SNMP, per ogni sottosistema, è definita una base di dati detta MIB ed ogni modifica, di essa, causa una modifica del sottosistema rappresentato, e viceversa. SNMP consente, al supervisore e agli agenti, di dialogare raccogliendo oggetti voluti nel MIB. Un MIB è strutturato in albero gerarchico che contiene variabili gestibili e identificate come Object Identifier (OID), in cui l'equivalente di una stringa è dettata da unità di numeri separati da un punto. Gli OID identificano oggetti gestiti nella gerarchia del MIB e sono organizzati basandosi su RFC standard. Questo albero include alcune variabili comuni a tutti i device, altre invece sono specifiche per altri device o alcuni venditori che possono definire le proprie variabili private.

Esempio di Albero OID:



Albero OID

L'interrogazione al MIB avviene attraverso un processo di creazione ed invio di messaggi:

1. **Il Manager SNMP crea la Get-Request:** basato sulle informazioni richieste, viene

creato un messaggio contenente il valore dell'oggetto MIB che l'applicazione vuole recuperare.

2. **Il Manager SNMP manda la Get-Request:** dopo che la prima fase è completata, il Manager SNMP invia la richiesta al dispositivo interrogato.
3. **L'agente SNMP riceve e processa la Get-Request:** l'agente riceve la richiesta e la processa. Esso guarda nella lista dei nomi gli oggetti MIB contenuti nel messaggio e controllo che siano validi e visiona che il valore di ogni variabile sia correttamente specificato.
4. **L'agente SNMP crea la risposta:** l'agente crea la risposta da inviare al Manager che contiene il valore dell'oggetto MIB richiesto e/o il codice dell'errore relativo a qualsiasi problema avvenuto, come il nome non valido dell'oggetto.
5. **L'agente SNMP invia il messaggio:** viene inviato, a questo punto, il messaggio di risposta al Manager.
6. **Il Manager SNMP processa:** il manager, una volta ricevuto il messaggio di risposta da parte dell'agente, elabora le informazioni contenute in esso.

Il protocollo SNMP è in grado di inviare messaggi di tipo:

- **`Get-Request`** – recupera un valore da una specifica variabile.
- **`Get-Response`** – risposta ad un get-request, get-nextRequest e set-request
- **`Get-NextRequest`** – recupera il valore di una variabile all'interno di una tabella, il manager SNMP non ha bisogno di conoscere il nome della variabile ed è in grado di avere informazioni accedendo in sequenza al MIB.
- **`Get-Bulk`** – usato per leggere il MIB con una sola richiesta senza mandare più richieste di Get-NextRequest. Grazie a questa operazione è possibile recuperare blocchi di dati ed è possibile eseguirla solo su SNMP V2 o superiori.
- **`Set-Request`** – può modificare le informazioni contenute nel MIB.

- **`Trap`** – definiti veri e propri messaggi inviati all'accadere di determinate eccezioni, se pur limitate, e che hanno la peculiarità di non usare grandi quantità di risorse.

Di versioni di SNMP ne esistono tre:

1. SNMPv1 – E' la prima implementazione del protocollo ed è di tipo interrogazione/risposta in cui il sistema di gestione di rete formula una query alla quale l'agent fornisce la risposta attraverso i comandi suddetti. Allo scopo di estendere queste semplici funzionalità, nel 1993 è stata introdotta la versione successiva del protocollo SNMP. La standardizzazione della seconda versione del protocollo SNMP fu terminata nel 1996.
2. SNMPv2 - La nuova implementazione presenta alcune importanti novità. La v2 del protocollo semplifica il trasferimento di grandi quantità di valori ed estende le funzioni già presenti. Le principali novità sono: l'introduzione di un nuovo tipo di data type chiamato UnInteger32 e il nuovo parametro *OSIaddress*, la collocazione dei valori e la gestione delle righe in tabella. In SNMPv2 i messaggi trap sono in un formato differente per soppiantare la versione precedente e sono stati definiti nuovi comandi tra cui *GetBulk* ed *Inform*. Quest'ultimo permette di scambiare informazioni trap tra NMS e quindi ricevere risposta, se l'agent non dovesse fornire valori per tutte le query, verranno comunque resi risultati parziali.
3. SNMPv3 - La terza versione del protocollo, nata nel 1998, è retro-compatibile ma introduce caratteristiche nell'ambito della sicurezza attraverso il login, privacy e controllo d'accesso e dell'amministrazione, nella gestione dei dati di login e tele-configurazioni. Tale versione si avvale della crittografia con MD5 e criptaggio DES.

Capitolo 2 – RFC 5424 ed i tools per l'analisi dei Log

2.0.1 – Cos'è l' RFC5424

L'RFC 5424 è un documento che specifica un protocollo di categoria Standard Track, per la Comunità Internet, soggetto a richieste di discussione e suggerimenti per ulteriori miglioramenti. Tale documentazione descrive il protocollo syslog nel dettaglio, ne spiega il funzionamento e, soprattutto, senza di esso ogni altro standard avrebbe bisogno di definire il proprio formato per i pacchetti syslog e i meccanismi di trasporto: questo porterà problemi di compatibilità. Syslog è strutturato a livelli e viene utilizzato per trasmettere messaggi di notifica all'accadere di un determinato evento. Questi messaggi devono essere interpretati correttamente dall'amministratore di rete per capire come agire. Ad ogni modo, un messaggio syslog ha delle precise caratteristiche e campi ognuno dei quali ha significati ben definiti.

Syslog si compone di alcune funzioni che vengono eseguite ad ogni livello: `originator` che ha il compito di generare contenuti syslog per essere trasportati in un messaggio, `collector` che raccoglie contenuti syslog per essere analizzati, `relay` che inoltra i messaggi accettandoli dagli originator o altri relay per mandarli ai collector. Infine `transport sender` e `transport receiver` con cui vengono spediti e prelevati i messaggi verso e da uno specifico protocollo di trasporto.

Il protocollo di trasporto utilizzato è il protocollo UDP quindi non è presente nessun meccanismo che garantisca la consegna dei messaggi e, essendo inaffidabile, alcuni di essi si potrebbero perdere. Questo può avvenire a causa della congestione della rete e le conseguenze non possono essere determinate in quanto, in base alla gravità del messaggio syslog, un amministratore potrebbe non venire a conoscenza di un grave problema.

I messaggi syslog hanno una dimensione minima di 480 ottetti ed una massima di 2048 che ogni transport receiver deve supportare. Quest'ultimo potrebbe anche troncare i messaggi ma è consigliabile avvenga alla fine del messaggio per una questione di sicurezza.

Un messaggio syslog ha il seguente formato:

```
HEADER (spazio) DATO-STRUTTURATO [(spazio) MSG]
```

L'Header rappresenta un set di caratteri ASCII a 7 bit in un campo di 8 bit, il formato dell'header è progettato per funzionare con syslog basato sul vecchio BSD (Berkeley Software Distribution) ed è formato da una serie di campi che descrivono alcune caratteristiche del messaggio syslog tra cui `PRI`, `VERSION`, `TIMESTAMP`, `HOSTNAME`, `APP-NAME`, `MSGID`.

Il PRI è composto da tre, quattro o cinque caratteri ed è formato da parentesi angolari che racchiudono un numero noto come Priority value (PRIVAL) e rappresenta la Facility e la Severity.

Il campo VERSION indica la versione delle specifiche del protocollo syslog, tale numero viene incrementato quando sono apportate modifiche all'header.

Il campo TIMESTAMP indica la data in cui viene generato il messaggio syslog, ed è composto, obbligatoriamente, dalla lettera T e Z. Se un'applicazione non fosse in grado di ottenere le informazioni dal sistema, è necessario usare il NILVALUE, ossia un valore nullo. Ecco alcuni esempi di TIMESTAMP:

```
Esempio 1: 1985-04-12T23:20:50.52Z
```

Questo esempio indica 20 minuti e 50.52 secondi dopo la 23 esima ora del 12 aprile del 1985.

```
Esempio 2: 2003-08-24T05:14:15.000003-07:00
```

L'esempio 2 rappresenta il 24 agosto 2003 alle 05:14:15am, 3 microsecondi. La presenza dei microsecondi è indicata dalle cifre aggiuntive nel TIME-SECFRAC, quest'ultimo viene incluso se fosse possibile aumentare la precisione dell'ora.

L'HOSTNAME identifica la macchina che ha mandato il messaggio syslog e contiene il nome host e del dominio del mittente nel formato chiamato Fully Qualified Domain Name (FQDN). Non tutte le applicazioni syslog, però, sono in grado di fornire tale formato quindi

potrebbero essere presenti altri valori nel campo HOSTNAME.

L'APP-NAME è una stringa che identifica il dispositivo o l'applicazione che ha generato il messaggio ed è utile per filtrare i messaggi. Anche qui è possibile utilizzare il NILVALUE quando l'applicazione non conosce il nome o non è in grado di fornirlo.

Il MSGID identifica il tipo di messaggio come ad esempio “TCPIN” o “TCPOUT” usato dal firewall per il traffico TCP. Il MSGID è una stringa e viene usato per filtrare i messaggi. Quando l'applicazione non fornisce nessun MSGID è necessario usare il NILVALUE.

Il dato-strutturato fornisce un meccanismo per esprimere le informazioni in formato dato facilmente interpretabile, può contenere zero, uno o multipli elementi di dato strutturato che sono definiti come “SD-ELEMENT”. Quest'ultimi consistono in una coppia nome e nome-valore: il nome viene indicato come SD-ID mentre le coppie nome-valore come “SD-PARAM”. Entrambi, SD-ID e SD-PARAM, sono case sensitive: il primo identifica il tipo e lo scopo di SD-ELEMENT, mentre il secondo consiste in un nome detto PARAM-NAME ed un valore detto PARAM-VALUE.

Esempio 1:

```
[exampleSDID@32473iut="3"eventSource="Application"eventID="10
11"]
```

Questo esempio è un elemento di dato strutturato con un SD-ID non controllato da IANA di tipo “exampleSDID@32473”, che ha tre parametri.

Esempio 2

```
[ exampleSDID@32473iut="3"eventSource="Application"eventID="1
011"][examplePriority@32473 class="high"]
```

L'esempio contiene un errore: il carattere SP (spaazio) è presente dopo la parentesi quadra iniziale. Un elemento di dato strutturato SD-ID deve immediatamente seguire la parentesi iniziale quindi lo spazio rende valido il dato strutturato e l'applicazione scarterà questo messaggio.

Gli SD-ID sono riservati ad IETF Review. I nomi sono validi solamente se venissero prima registrati da IANA e non devono contenere il simbolo '@', '=', ']', '" ', uno spazio o un carattere di controllo. Il nome deve essere una stringa stampabile US-ASCII. Gli attuali SD-ID registrati da IANA sono `timeQuality` , `origin` e `meta`, ognuno dei quali ha una serie di PARAM-NAME opzionali.

Il campo MSG contiene un messaggio riguardo le informazioni di un evento, il set di caratteri usato è l'UNICODE codificato in UTF-8. Grazie a questa codifica, la stringa deve iniziare con l'Unicode byte order mask (BOM) ossia ABNF %xEF.BB.BF.

Esempio1:

```
<34>1 2003-10-11T22:14:15.003Z mymachine.example.com su -
ID47- BOM'su root'failed for lonvick on /dev/pts/8
```

In questo esempio, il campo VERSION è settato a 1, la Facility è di 4 e la Severity è 2. Il messaggio è stato creato il giorno 11 ottobre 2003 alle 10:14:15pm UTC, 3 millisecondi. Il messaggio è stato originato da un host che si identifica come “mymachine.example.com”, l'APP-NAME è “su” e il MSGID è “ID47”. Il MSG è “su root failed for lonvick”, codificato in UTF-8. La codifica è definita da BOM.

Non sono presenti dati strutturati e questo lo si può vedere dal “-” nel campo STRUCTURED-DATA.

Esempio 2:

```
<165>1 2003-10-11T22:14:15.003Z mymachine.example.com
evntslog - ID47 [exampleSDID@32473 iut="3"
eventSource="Application" eventID="1011"] BOMAn application
event log entry…
```

Questo esempio è modellato sulla base del primo, questa volta contiene il dato strutturato: un singolo elemento con il valore “[exampleSDID@32473 iut=”3” eventSource=”Application eventID=”1011”]”. Il campo MSG è “An application event log entry...” Il BOM indifica la codifica in UTF-8.

Syslog può generare una grande quantità di dati, il loro trasferimento potrebbe creare problematiche considerando che il protocollo di trasporto UDP non possiede un meccanismo di controllo della congestione. Un possibile meccanismo risponde alla congestione riducendo l'ammontare di traffico ed è vitale per il funzionamento di Internet, per questo motivo è necessario il protocollo di trasporto TLS laddove potrebbe generarsi una congestione. Nei casi in cui le reti siano gestite, dove l'ammontare di traffico sia stato limitato con meccanismi di ingegneria, sarà possibile utilizzare UDP. In tutti gli altri ambienti dovrebbe essere usato il protocollo TLS.

I messaggi syslog possono essere danneggiati o durante il trasferimento o a causa della modifica da parte di un utente malintenzionato, in questo caso tali messaggi non saranno consegnati. Inoltre, se tra il mittente ed il destinatario ci fosse un utente non autorizzato, i messaggi potrebbero essere intercettati e modificati nascondendo tale attività non autorizzata.

Il protocollo syslog non possiede un meccanismo che garantisce la riservatezza dei messaggi in transito infatti, tali messaggi, sono in chiaro. Questa caratteristica, però, non è del tutto svantaggiosa in quanto il personale operativo sarà in grado di leggere i messaggi ed associarli ad altri eventi visti in altri pacchetti per tenere traccia del problema e risolverlo. Lo svantaggio, invece, consiste nel fatto che un utente malintenzionato sarà in grado di osservare il contenuto dei messaggi e fare danni. Per evitare questo problema gli operatori devono utilizzare una mappatura sicura dei trasporti.

Durante la configurazione delle macchine è necessario fare attenzione alle impostazioni per l'inoltro dei messaggi. In un caso particolare, un amministratore potrebbe erroneamente configurare due relay per l'inoltrare messaggi, con una certa gravità, fra di loro creando quindi un meccanismo di loop. Questo causerà un problema sia sulla disponibilità dei dispositivi sia una degradazione della rete.

Un amministratore di rete deve avere il tempo per stimare la giusta capacità del collector syslog. Un utente malintenzionato potrebbe eseguire un attacco Denial of Service riempiendo la memoria del collector con falsi messaggi e, per evitare ciò, potrebbe essere utile inserire i record in un file circolare ma non assicura che un amministratore, in futuro,

possa essere in grado di rivedere tali registri. Per scongiurare al meglio un attacco DOS, i costruttori del sistema syslog devono fornire le caratteristiche che riducono al minimo tale minaccia come, ad esempio, accettare messaggi syslog provenienti solo da indirizzi IP conosciuti.

2.1 – Alcuni software per l'analisi del log

Per monitorare la rete e tutto quello che ne concerne, vengono in nostro aiuto numerosi tool capaci di guidare l'utente nell'analizzare, verificare e correggere tutti gli errori che avvengono nel sistema. Non tutti i software sono open-source tra cui Netmon, NetMonitor, CommTraffic, ecc ma quelli analizzati in questo lavoro lo sono e ne vengono evidenziate caratteristiche e differenze:

2.1.1 - Ntopng

NTOPNG (ntop next generation) è un software open-source per il monitoraggio del traffico di rete, è la nuova generazione del vecchio ntop e si basa su libpcap rendendolo così portatile in modo che sia eseguibile, virtualmente, su ogni piattaforma Unix, Linux, BSD, MacOSX e Windows. Il motore di questo software è scritto in C++ e l'interfaccia web è scritta in Lua. Gli utenti possono utilizzare un browser per navigare attraverso ntop, che agisce come web server per ottenere informazioni sul traffico ed un dump sullo stato della rete. Un dump è un elemento di un database contenente un riepilogo dei dati del database.

Ntopng si basa sul server Redis: una struttura dati key-value store open source residente in memoria con persistenza facoltativa, piuttosto che su un database tradizionale. Ntopng utilizza NDPI per il rilevamento del protocollo, supporta la geo-localizzazione degli host ed è in grado di mostrare in tempo reale l'analisi del flusso di un host connesso.

NDPI, Deep Packet Inspection, è un superset della popolare libreria OpenDPI. Il suo obiettivo è quello di estendere tale libreria aggiungendo nuovi protocolli che sono disponibili solo nella versione a pagamento di OpenDPI. NDPI viene utilizzato da ntop per il rilevamento a livello di applicazione dei protocolli, indipendentemente dalla porta utilizzata. Ciò significa che è possibile rilevare sia i protocolli su porte standard (es. 80 per HTTP) che non standard. NDPI supporta molti protocolli tra cui Syslog ed SNMP.

Ntopng permette di eseguire numerose altre operazioni tra cui classificare il traffico di rete in base a diversi criteri tra cui indirizzo ip, porta, protocollo, throughout ecc. Mostra il traffico di rete e IPv4/v6 degli host attivi, produrre relazioni a lungo termine su vari parametri di rete come il throughput. E' possibile memorizzare su disco le statistiche di traffico in formato RRD (round-robin database), caratterizzare il traffico HTTP, facendo leva sui servizi di caratterizzazione di Google e HTTP Blacklist. Ntopng permette di analizzare il traffico IP, ordinare i dati in base alla sorgente/destinazione, produrre statistiche HTML5/AJAX sul traffico di rete e molto altro.

Dalla release 2, ntopng è disponibile in due versioni: la community edition il cui codice può essere trovato su Github, un servizio web di hosting per lo sviluppo di progetti software, e la versione professional che è a pagamento ma offre funzionalità aggiuntive tra cui la capacità di generare report HTML salvabili in pdf, supportare SNMP per l'interrogazione degli agent e molto altro.

2.1.2 - Ganglia

Ganglia è un sistema scalabile di monitoraggio distribuito per sistemi di calcolo ad alte prestazioni come cluster e grids. Consente di vedere in remoto statistiche live o storiche dell'utilizzo della rete o carico della CPU per tutte le macchine che vengono monitorate.

Questo tool si basa su un design gerarchico destinato a federazioni di cluster e sfrutta XML per la rappresentazione dei dati, RRDtool per l'archiviazione e la visualizzazione dei dati e la portabilità dei dati, e XDR (external data representation). L'implementazione è robusta, è stato portato ad un estensivo set di sistemi operativi e processori ed è correntemente usato su migliaia di cluster in tutto il mondo, in grado di scalare per gestire cluster con 2000 nodi.

Un nodo è una singola macchina che invia i dati di monitoraggi differenti al demone di ganglia. I cluster sono tutti i nodi usati per alcuni particolari presupposti.

Il Grid, in termini di ganglia, significa che è una collezione di cluster o è possibile raggruppare un numero di cluster e chiamarli grid.

Ganglia è un progetto open-source formato da varie parti tra cui ganglia monitor

daemon (gmond) che gira su qualsiasi nodo si voglia monitorare ed ha quattro compiti principali:

1. Monitorare i cambiamenti di stato di un host.
2. Annunciare cambiamenti rilevanti.
3. Ascoltare lo stato di tutti gli altri nodi in unicast o multicast.
4. Rispondere alle richieste per una descrizione XML dello stato di un cluster.

Ogni gmond trasmette informazioni in due modi differenti: Unicast o Multicast usando UDP e mandando XML tramite la connessione TCP.

Gmetad (Ganglia Meta Daemon) che colleziona dati da altri gmetad e da tutti i gmond. Il gmetad memorizza i dati nel formato RRD.

2.1.3 - Munin

Munin è uno strumento open-source per il monitoraggio della rete e delle risorse di sistema, utile ad analizzare i problemi che possono influire sulle performance andando a elaborare i dati del "cosa è cambiato oggi", per questo motivo è importante notare che Munin non è un sistema capace di dare prontamente i risultati, ma è molto valido nella valutazione dei dati in un tempo definito come ad esempio 24 ore. Grazie alla sua architettura plug and play, l'installazione è molto semplice e fornisce in output una serie di grafici e pagine HTML tramite l'interfaccia web. La produzione di tali grafici si basa sugli RRDTools che lo accomuna con gli altri software sopracitati ed è scritto in Perl: un linguaggio di programmazione dinamico ad alto livello.

Munin possiede circa 500 plug-in, per il monitoraggio, scritti in linguaggi diversi capaci di monitorare servizi come Apache, Mysql, SAN (storage area network), le prestazioni del computer e delle applicazioni. La potenza di tale strumento sta nella possibilità di gestire gli allarmi, a due livelli, per ogni aspetto monitorato: il livello warning che si aziona al superamento della prima soglia e critical che viene segnalato al superamento della seconda soglia. Inoltre, per ogni allarme, è possibile impostare l'invio di email in cui viene riepilogala la situazione che ha fatto scattare l'allarme.

Munin è disponibile per tutte le piattaforme: Munin-node per Windows, iMunin per iPhone e iPad e, semplicemente, Munin per Linux.

2.1.4 - Monit

Anche Monit è un software open-source, per Unix e Linux, ed è un processo di supervisione ossia una forma di servizio di gestione del sistema operativo. E' utile a monitorare lo stato di file, directory, processi e periferiche, più in generale, è possibile visionare lo stato del sistema direttamente dalla linea di comando oppure attraverso HTTP nativo, interpretando particolari misure di manutenzione una volta riscontrate anomalie. Tale utility consente la manutenzione e riparazione automatica ed azioni significative in caso di situazioni di errore.

Monit è divenuto popolare grazie a Ruby on Rails, un framework per applicazioni web scritte in Ruby: un linguaggio di programmazione dinamico, orientato agli oggetti sviluppato in Giappone. Monit deve la sua fama anche al web server Mongrel, un software open-source ed un server web che presenta un'interfaccia HTTP standard, per il bisogno di gestire molti processi Mongrel identici necessari per supportare Ruby on Rails, e Monit era abbastanza adatto per le esigenze della comunità Ruby on Rails. Molti siti popolari, come Twitter, utilizzano Monit.

E' disponibile per Linux e iOS.

2.1.5 - Nagios

Nagios è una nota applicazione open-source per il monitoraggio di computer e risorse di rete con la funzione base di controllare nodi, reti e servizi e avvisare quando questi hanno un funzionamento anomalo o quando ritornano attivi. Nagios offre molteplici funzionalità come il monitoraggio di reti SNMP, ICMP, SMTP, POP3 ecc, il monitoraggio dell'uso del processore, dell'hard disk, dei log di sistema. Supporta il monitoraggio remoto e l'installazione di plug-in, per ampliare le funzionalità, scritti in numerosi linguaggi. Il potente software messo a disposizione permette di controllare in modo parallelo i servizi, notificare quando un'applicazione riscontra problemi, effettuare il monitoraggio ridondante e molto altro ancora.

Di Nagios esistono alcune varianti tra cui Nagios Log Server che consente agli amministratori di visualizzare, ordinare e configurare i log da qualsiasi fonte. Tale versione analizza, raccoglie, memorizza dati di log in base a delle specifiche preimpostate ed è in grado di analizzare ed effettuare ricerche di tutti i tipi di dato di registro.

Nagios Network Analyzer è una soluzione grazie alla quale è possibile essere pronti alla risoluzione dei guasti e minacce di sicurezza che possono compromettere il funzionamento dei processi.

Il software è disponibile sia per i sistemi Unix che Windows.

2.1.6 - Cacti

Cacti è una soluzione grafica open-source per il monitoraggio di rete progettata per sfruttare la potenza di RRDTool. Cacti permette ad un utente di effettuare operazioni di campionamento ad intervalli prestabiliti e rappresentare graficamente i risultati.

Questo tool è utilizzato per rappresentare graficamente i dati temporali di metriche come il carico della CPU e l'utilizzo della banda di rete. E' possibile anche monitorare il traffico di rete tramite operazioni di campionamento sulle interfacce di uno switch o router attraverso SNMP.

Il front-end del tool è in grado di gestire più utenti, ognuno con il loro settaggio in modo da poterlo usare come server dedicato, server virtuale privato ecc, e mostrare le statistiche. Cacti può essere usato per configurare la raccolta dei dati stessi tramite alcuni setup senza una configurazione manuale di RRDtool e monitorare qualsiasi sorgente tramite uno script di shell.

2.1.7 - Zabbix

Anche Zabbix è un software per il monitoring sulla disponibilità e performance di una struttura. E' un tool open-source, gratuito per qualsiasi utilizzo e per raccogliere dati da qualsiasi dispositivo come server, apparato di rete e virtual machine. Grazie a questo software è possibile ottenere uno storico di tutte le informazioni necessarie con l'utilizzo di mappe, grafici e analizzarli per generare alert o azioni automatiche.

Zabbix è capace di sfruttare gli agent dei principali sistemi operativi o dei vari protocolli come SNMP, SSH, WMI, IPMI. La funzione base del tool è quella di monitorare applicazioni web e ambienti virtuali.

2.1.8 - LogAnalyzer

Il software dal nome LogAnalyzer è un open-source profect front-end. Dall'interfaccia gradevole, questo tool, permette di visionare i messaggi via web di vario tipo tra cui: Messaggi Syslog, Eventi Windows, Status Reports e Statistiche.

Sviluppato da Adiscon, LogAnalyzer è un interfaccia web per syslog e per altri eventi della rete fornendo una serie di strumenti per l'analisi degli eventi in tempo reale e servizi di reporting. Grazie ai reports è possibile tenere d'occhio le attività di rete. Consolidando i dati sugli eventi syslog, verranno fuori fogli e grafici che un amministratore di rete troverà di facile lettura. Attualmente, l'ultima versione del software è la 4.1.3beta che verrà utilizzata per questo lavoro.

2.2 - Differenze tra i software analizzati

Sono stati messi a confronto tutti i software sopra descritti sulla base di determinati parametri scelti. E' importante conoscere la differenza di piattaforma su cui si basano i vari software, le varie licenze e ovviamente la compatibilità piena con il protocollo Syslog. Inoltre, è essenziale notare alcune differenze di funzionalità che ci sono tra i vari tool per capire quale sia perfetto per le proprie esigenze. Si può notare, anche, la differenza tra gli indici di attività della community che coglie la caratteristica di quanto sia apprezzato un determinato software rispetto ad un altro e come questo venga sviluppato, supportato e migliorato nel tempo. Ad ogni modo, per la comparazione sono stati utilizzati i seguenti parametri:

Snmp: In grado di recuperare e riportare su statistiche SNMP.

Syslog: In grado di recuperare e riportare su statistiche Syslog.

Plug-in: Architettura del software basata du un numero di plugins che provvedono funzionalità aggiuntive.

Platform: Linguaggio di programmazione con cui è scritto il software.

IPv6: Supporto per IPV6.

Alert: Capacità di inviare alert all'amministratore.

Licenza: Licenza su cui è realizzato il software.

Auto-Discovery: La capacità automatica di individuare gli host o i device connessi nella rete.

Controllo Accesso: Consente all'amministratore di controllare l'accesso a certe parti.

Indice Attività Community: Livello di attività delle community per il supporto tecnico/informativo e scambio di opinioni.

2.2.1 Tabella comparativa:

Funzione/ Software	Ntopng	Ganglia	Munin	Monit	Nagios	Cacti	Zabbix	LogAnalyzer
SNMP	✓	Via Plug-in	✓	X	Via Plug-in	✓	✓	✓
Syslog	✓	X	X	✓	Via Plug-in	✓	✓	✓
Plug-ins	X	✓	✓	X	✓	✓	✓	✓
Platform	C++, Lua	C, PHP	Perl	C	C, PHP	PHP	Python, Java	PHP
IPv6	X	✓	✓	✓	✓	✓	✓	✓
Alert	✓	X	Parziale	✓	✓	✓	✓	✓
Licenza	GPLv3	BSD	GPL	AGPL	GPL	GPL	Free Core, GPL, Commercial, Enterprise	GPLv3, Commercial
Auto Discovery	Parziale	Via gmond check in	X	X	X	Via Plug-in	✓	✓
Controllo Accesso	✓	X	✓	✓	✓	✓	✓	✓
Indice Attività Community	4	5	3	5	4	5	4	4

Tabella comparativa software

La scelta effettuata per lo studio è quella di Adiscon LogAnalyzer in quanto soddisfa tutte le richieste necessarie per lo sviluppo di questa tesi inoltre, a parità di altri software analizzati, è di facile installazione, configurazione, utilizzo e interpretazione. Presenta un'interfaccia sobria e ben schematizzata, è in grado di visualizzare correttamente i messaggi di log, creare dei report personalizzati per una possibile analisi approfondita di un determinato host e personalizzare i grafici alla fine dello studio.

Capitolo 3 – Installazione e configurazione del syslog server

Questo capitolo spiega come e quali esperimenti sono stati fatti in laboratorio, le macchine utilizzate e tutte le configurazioni necessarie affinché tutto funzioni.

3.0.1 - Configurazione server rsyslog

Per prima cosa bisogna installare e configurare una piattaforma LAMP (linux, apache, mysql, php), con un semplice ed unico comando:

```
sudo apt-get install apache2 apache2-mpm-prefork php5-mysql
mysql-server php5 libapache2-mod-php5 php5-cgi php5-gd php5-
cli phpmyadmin
```

Nota: apache2-mpm-prefork non esiste sulle versioni più recenti di ubuntu. Per questa tesi è stato usato ubuntu 14.04.

Durante l'installazione, verrà chiesto di scegliere una password per l'account root di MySQL.

Una volta terminata è necessario modificare il file di configurazione *rsyslog.conf,* che si trova nella cartella *etc*, aprendolo con un qualsiasi editor. Le modifiche da apportare sono quelle di de-commentare le seguenti righe:

```
# provides UDP syslog reception
$ModLoad imudp
$UDPServerRun 514

# provides TCP syslog reception
$ModLoad imtcp
$InputTCPServerRun 514
```

In modo da accettare tutto il traffico TCP e UDP.

Salvare il file e riavviare il demone:

```
# /etc/init.d/rsyslog restart
# service rsyslog restart
```

3.0.2 - Installazione e configurazione LogAnalzyser

Per prima cosa è doveroso scaricare il software direttamente dal sito ufficiale http://www.loganalyzer.adiscon.com/downloads, per questo lavoro è stata scaricata la versione 4.1.3beta4.

A download completato, scompattare il pacchetto scaricato e spostare la cartella *src* nel nostro server Apache:

```
# tar -xzvf loganalyzer-4.1.3.tar.gz
# cd loganalyzer-4.1.3
# mv src /var/www/html
# mv /var/www/html/src /var/www/html/loganalyzer
```

A questo punto creiamo l'utente mysql e il relativo database. In questo caso è stato scelto l'utente loganalyzer e il database loganalyzer arbitrariamente.

```
# mysql -u root -p
# mysql> create user 'loganalyzer'@'localhost' identified by
'106698';
# mysql> create database loganalyzer;
```

Impostare l'utente creato per il controllo del database:

```
# mysql> grant all privileges on loganalyzer.* to
'loganalyzer'@'localhost';
# mysql> flush privileges;
# mysql> use loganalyzer;
# mysql> exit;
```

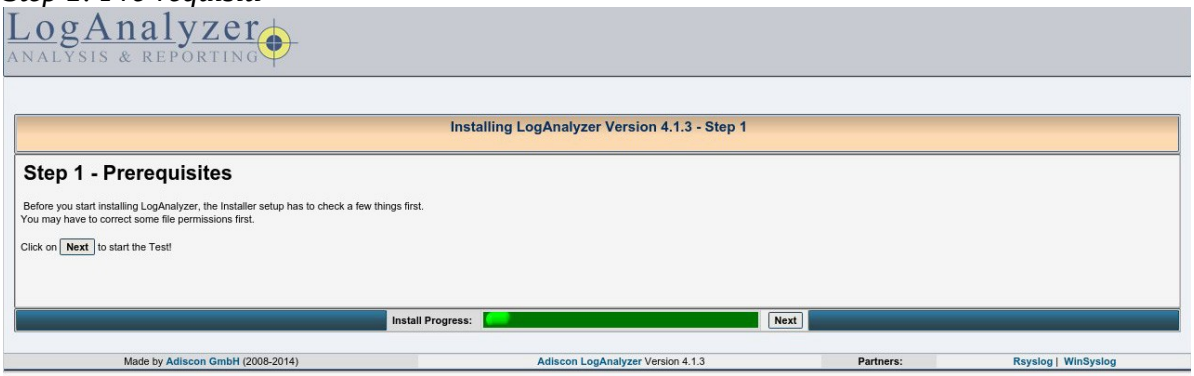
Adesso bisogna copiare i due file *configure.sh* e *secure.sh* dalla cartella *contrib* all'interno del nostro server Apache rendendoli poi eseguibili:

```
# mv loganalyzer-4.1.3/contrib/configure.sh
/var/www/html/loganalyzer
# mv loganalyzer-4.1.3/contrib/secure.sh
/var/www/html/loganalyzer
# chown www-data:www-data -R ./
# ./configure.sh
```

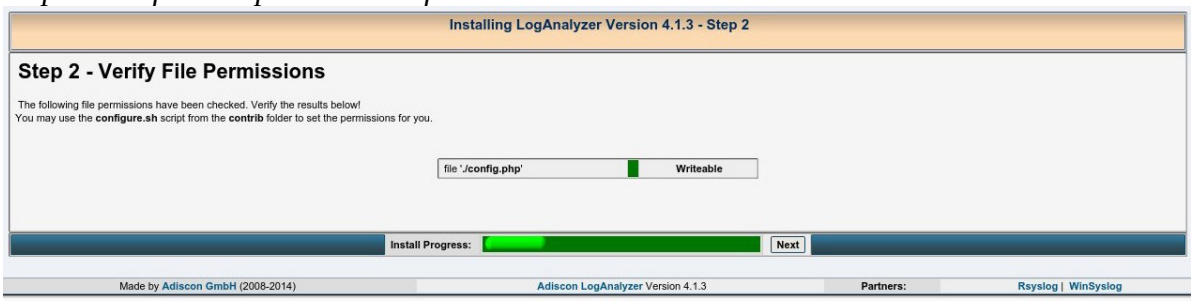
Ora che è stato lanciato lo script, sono necessari i permessi di lettura del file syslog (666) e non rimane che installare e configurare LogAnalyzer.

L'installazione si effettua attraverso un qualsiasi brower in cui, nella barra degli indirizzi, è necessario immettere l'indirizzo ip della macchina server. In questo caso è stato utilizzato l'indirizzo localhost: 127.0.0.1/loganalyzer. Sono otto i passaggi che devono essere eseguiti per terminare la configurazione:

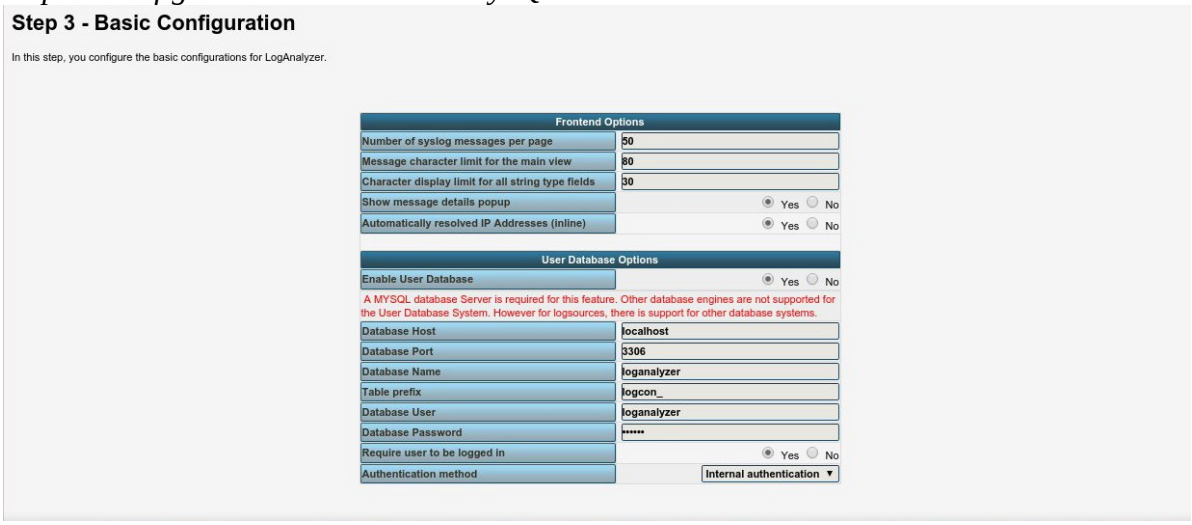
Step 1: Pre-requisiti



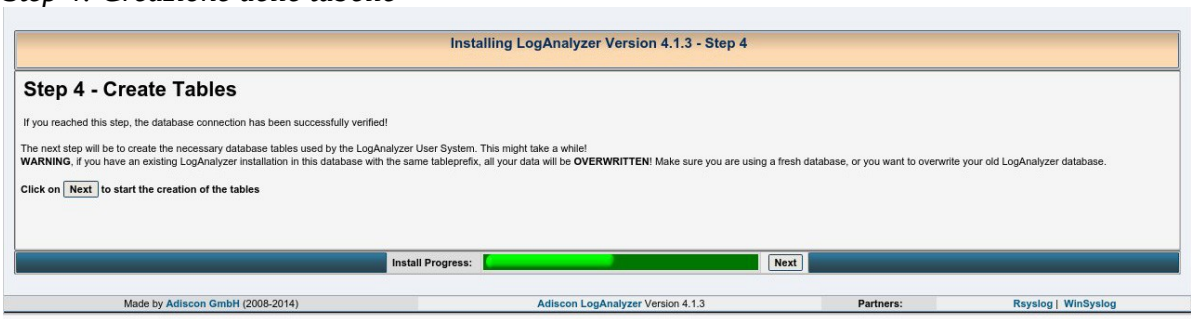
Step 2: Verifica dei permessi del file



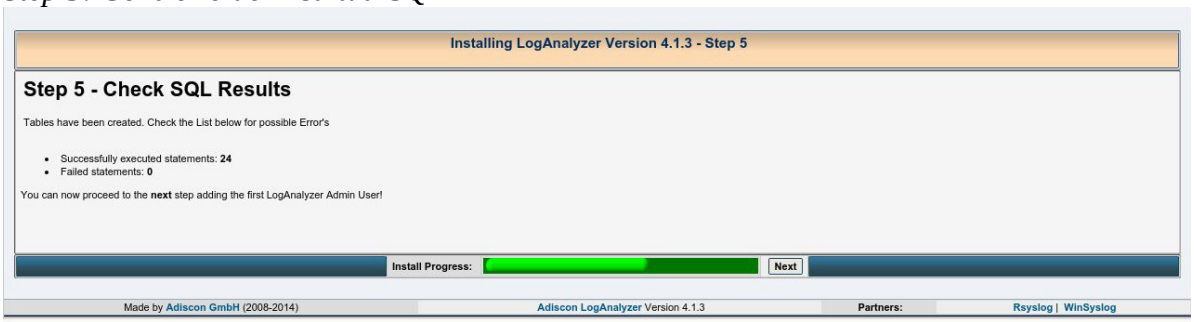
Step 3: Configurazione di base e di MySQL



Step 4: Creazione delle tabelle

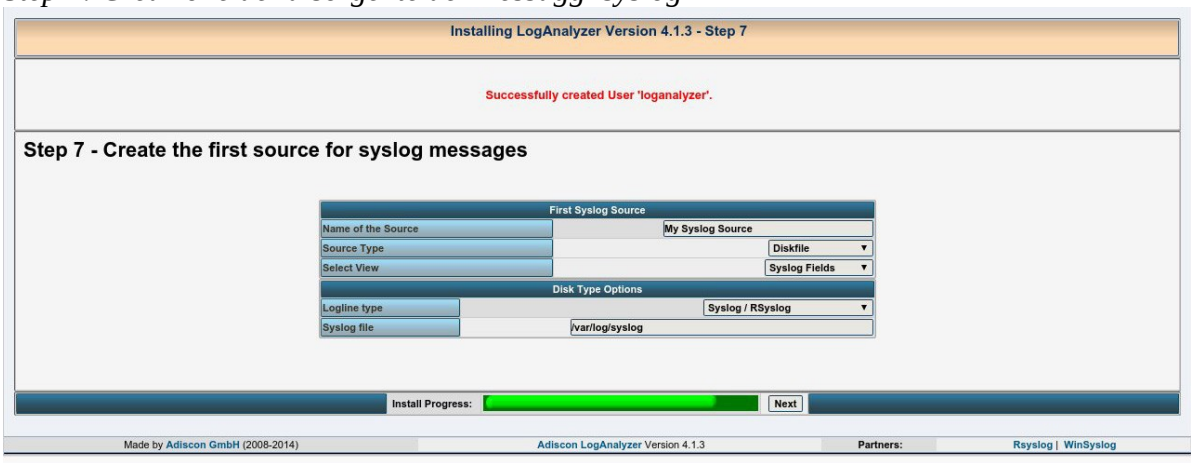


Step 5: Controllo dei risultati SQL



Step 6: Creazione dell'utente amministratore

Step 7: Creazione della sorgente dei messaggi syslog



Il formato Syslog / Rsyslog equivale al vecchio standard che fa riferimento all'ormai obsoleto RFC 3164 quindi, quello impostato è RSyslog23 che fa riferimento al RFC 5424.

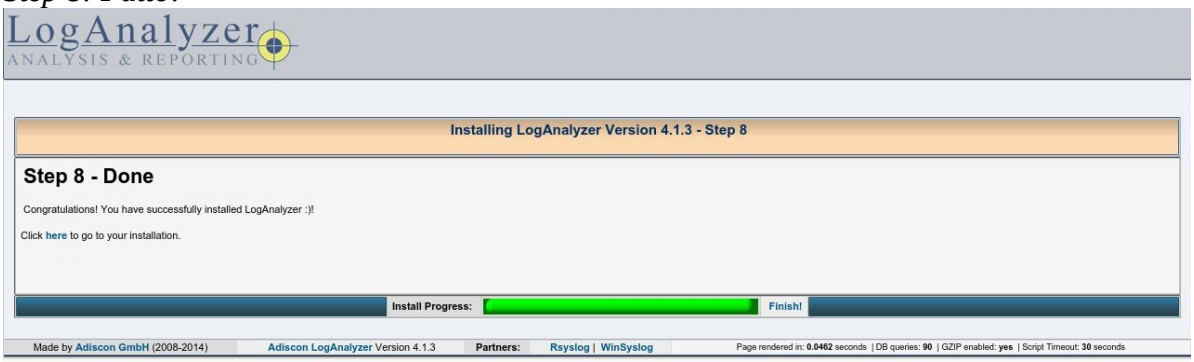
Per questo motivo è necessario apportare un'ulteriore modifica al file di configurazione di rsyslog commentando questa stringa:

```
#$ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat
```

e aggiungendo quest'altra:

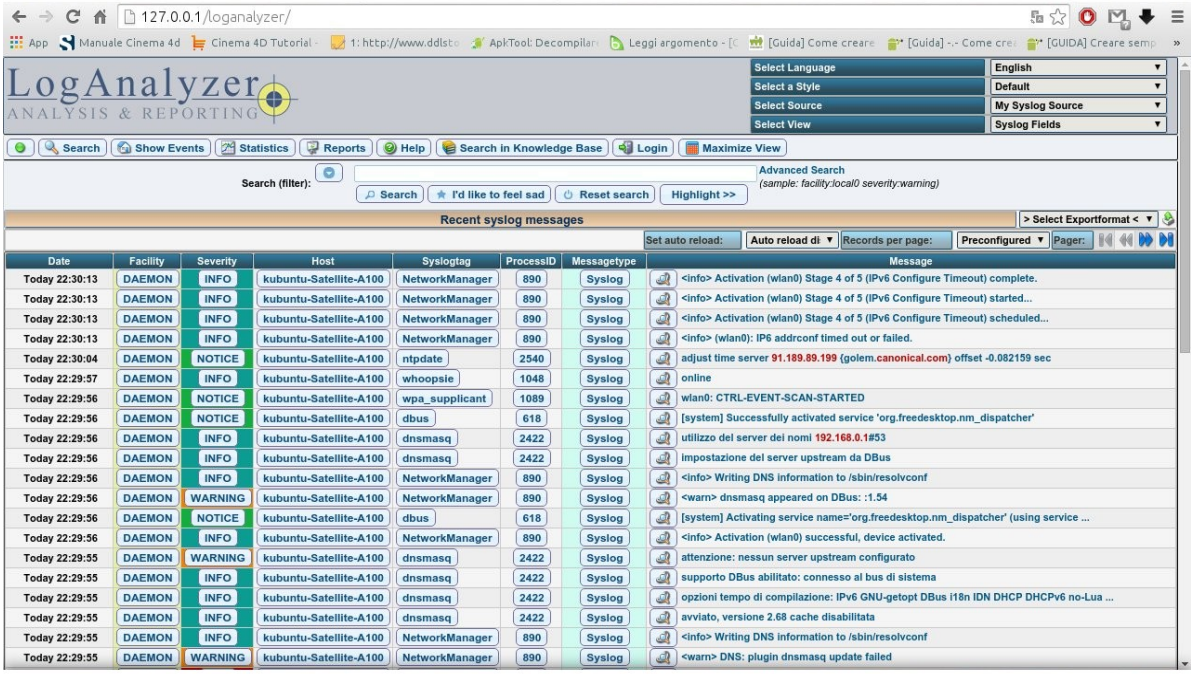
```
$ActionFileDefaultTemplate RSYSLOG_SyslogProtocol23Format
```

Step 8: Fatto!



Avendo configurato rsyslog per l'invio dei dati all'indirizzo ip della macchina, ecco che compaiono tutti i messaggi syslog generati dalla macchina nella sezione Show Events.

Immagine 1: Messaggi Syslog della macchina server



3.1 - Configurazione dell'invio di dati al server

3.1.0 - Configurazione del client rsyslog su una Raspberry

La prima cosa da fare, anche su raspberry, è installare la piattaforma LAMP esattamente come lo si è fatto in precedenza, d'altronde il sistema operativo Raspbian wheezy è basato su Linux.

Per configurare la raspberry è necessario collegarsi tramite ssh all'indirizzo ip attraverso il comando:

```
ssh pi@10.87.1.136
```

Una volta completato il procedimento, per far mandare tutti gli eventi syslog del sistema operativo al server, è sufficiente modificare il file rsyslog.com, con un editor, aggiungendo a fine file la stringa:

```
*.* @10.87.1.154
```

Tale indirizzo ip corrisponde all'indirizzo ip della macchina su cui è installato il server e quindi sulla quale noi vogliamo raccogliere tutti i dati. Una volta salvato il file, LogAnalyzer compariranno i log provenienti dalla raspberry:

3.1.1 - Configurazione PHP

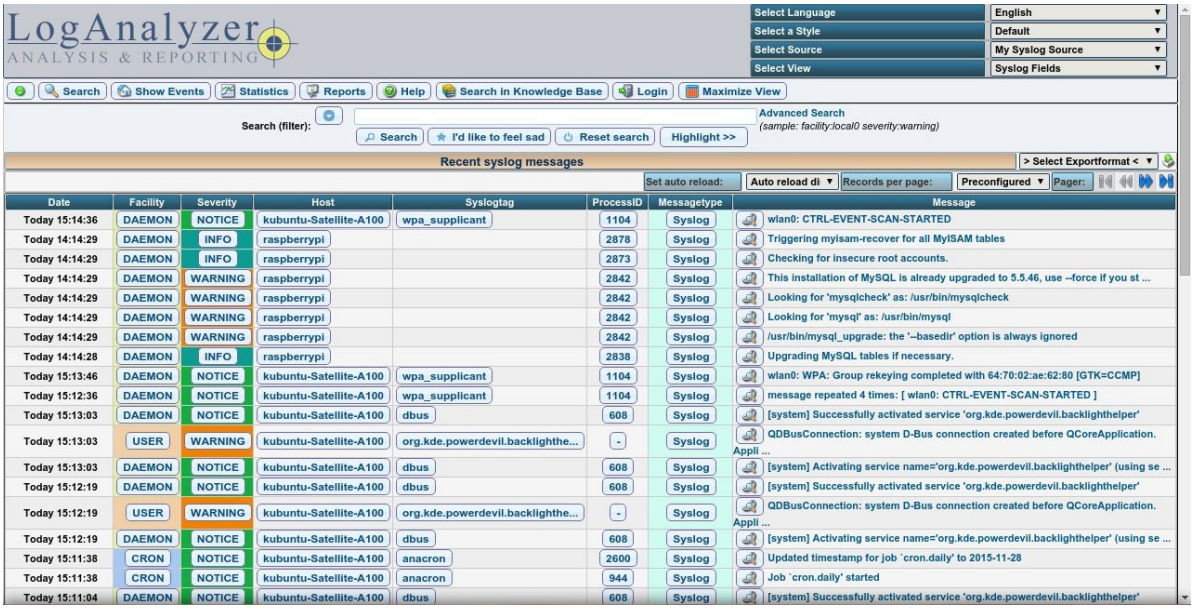
La configurazione di php è essenziale per fare in modo che l'informazione data dei log arrivi secondo la propria zona, di default è impostato su UTC ma in questo caso va cambiato in Europe/Rome.

```
sudo nano /etc/php5/apache2/php.ini
```

Cercare le seguenti stringhe e impostare i seguenti valori:

```
date.timezone = Europe/Rome
```

Immagine 2: Messaggi Syslog dalla raspberry



3.1.2 - Router Cisco

Una volta installato il programma minicom e lanciato, si configura la porta seriale e la porta usb alla quale è collegato il router.

Di default, i router e gli switch Cisco mandano messaggi di log per tutti i livelli di severity alla console e possono essere anche memorizzati in un buffer. Per abilitare queste funzioni si utilizzano i comandi `logging console` e `logging buffered` nella configurazione globale.

Ci sono tre passaggi necessari per configurare i router a mandare i messaggi syslog ad un syslog server:

1. Configurare l'hostname di destinazione o l'indirizzo ip del syslog server nella modalità di configurazione globale:

   ```
   R1(config)# logging 10.87.1.154
   ```

2. Configurare i messaggi da inviare al syslog server inserendo il livello di severity fino al quale considerarli. Per esempio se dovessimo impostare il livello 5, verranno

inviato i messaggi da 0 a 5:

```
R1(config)# logging trap 7
```

3. Configurare l'interfaccia sorgente:

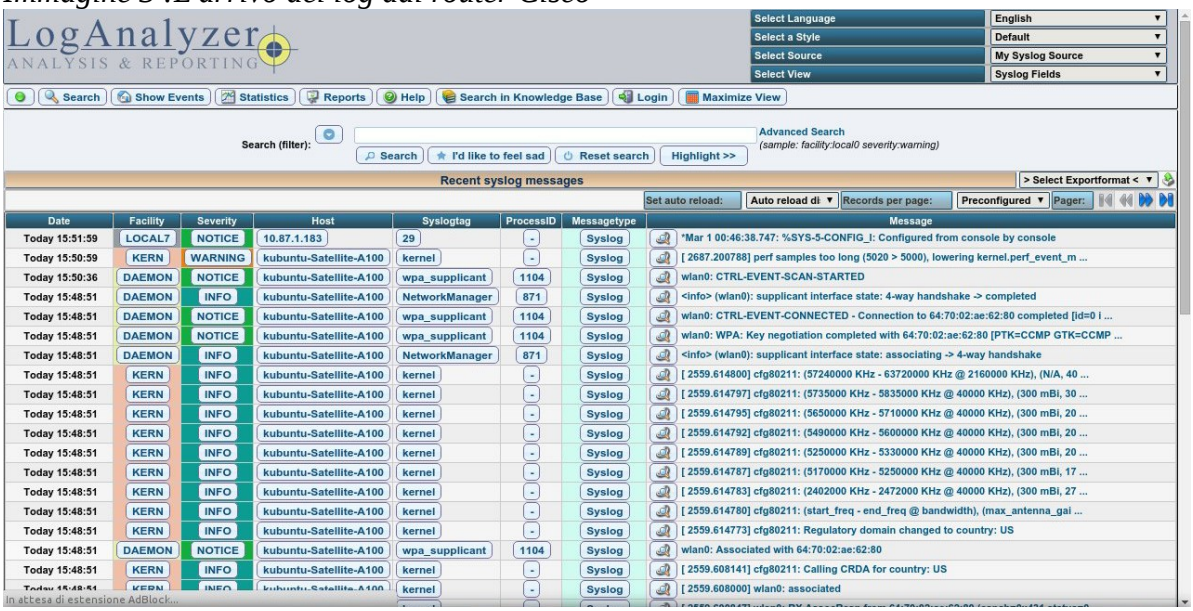
```
R1(config)# logging source-interface Ethernet1
R1(config)# no shutdown
```

E' importante a questo punto inserire il comando *no shutdown* per attivare l'interfaccia.

Al router Cisco è stato assegnato un indirizzo ip da DHCP che corrisponde a 10.87.1.80.

Ecco che anche i messaggi proveniente dal router Cisco arrivano al server:

Immagine 3 :L'arrivo dei log dal router Cisco



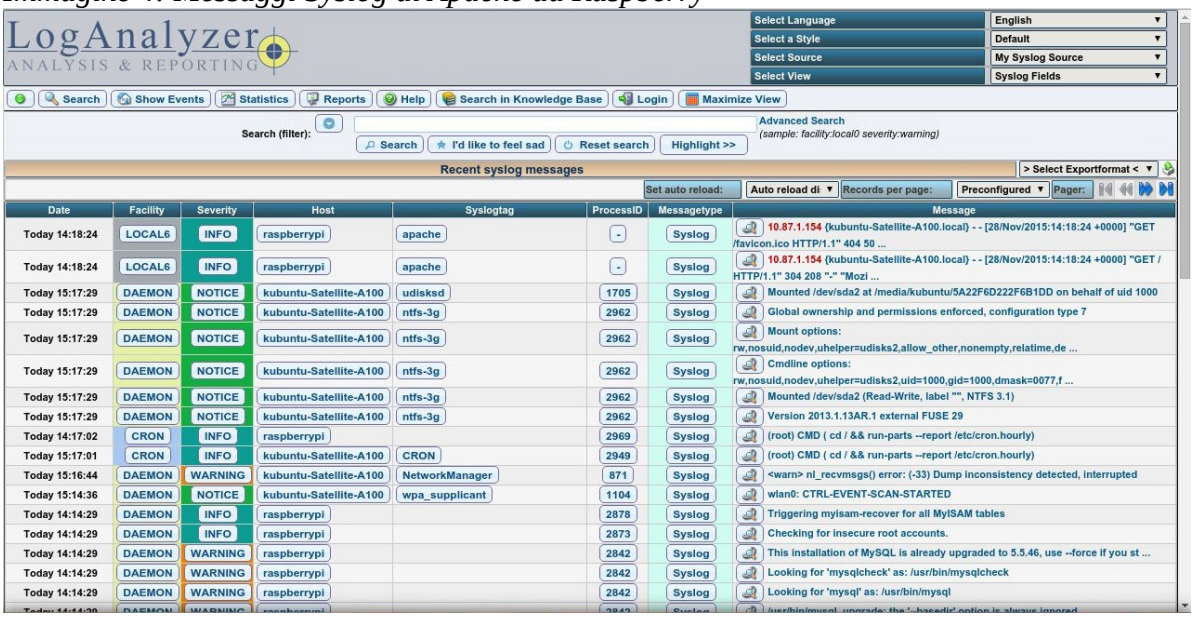
3.1.3 -Apache

Per configurare Apache aprire il file di configurazione e modificare la riga *CustomLog* nel file default della cartella /etc/apache2/sites-available/:

```
CustomLog "|/usr/bin/logger -t apache -p local6.info"
combined
```

Salvato il file e ricaricato apache, ecco che compaiono i messaggi da parte di apache sul nostro LogAnalyzer:

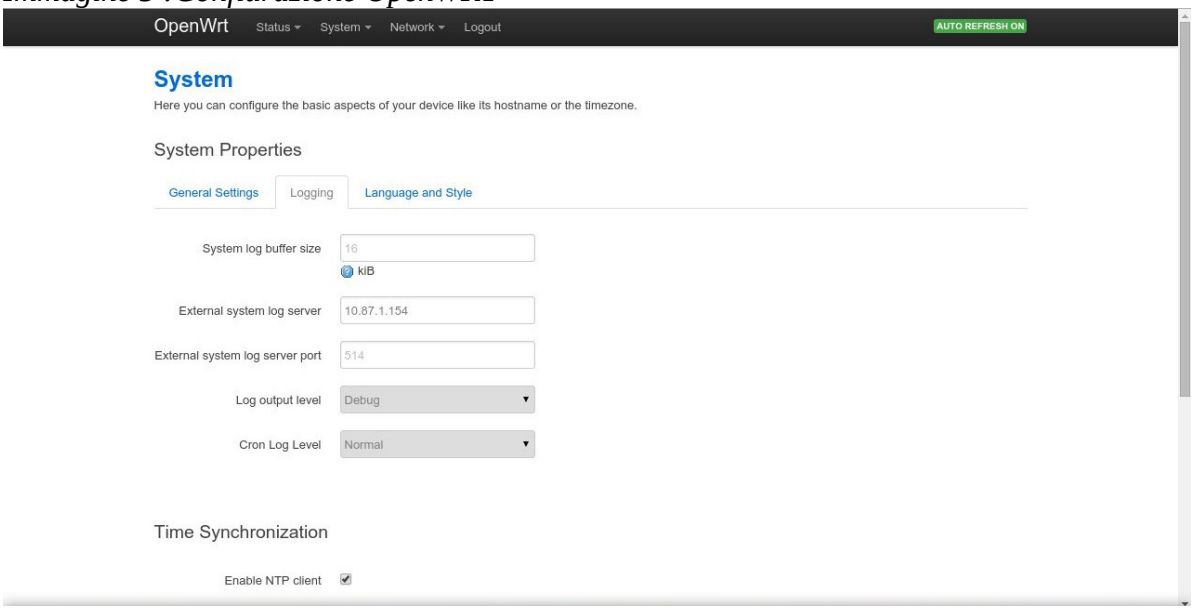
Immagine 4: Messaggi Syslog di Apache da Raspberry



3.1.4 - OpenWRT

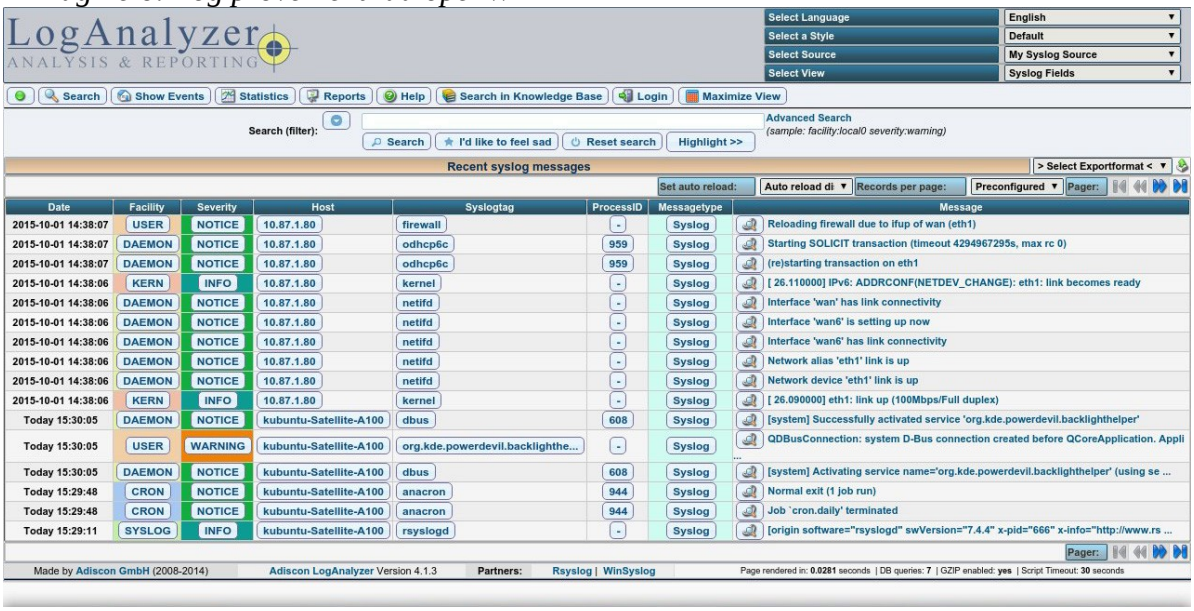
Per questo lavoro in laboratorio è stato utilizzato anche un router tp-link con sistema operativo OpenWrt e configurato in modo da inviare tutti i log sul server:

Immagine 5 :Confiurazione OpenWRT



Una volta generato qualche log, essi arrivano sulla nostra macchina configurata come server:

Immagine 6: Log provenienti da openWRT



3.1.5 - Configurazione macchina virtuale

Per l'esperimento in laboratorio è stata impiegato un computer dell'Hacklab sul quale è stata creata una macchina virtuale Linux, chiamata Mordor, con le seguenti caratteristiche hardware:

- 256 MB di RAM
- 10GB di hard disk

Una volta avviata, la configurazione è stata la medesima di quella effettuata nel precedente capitolo:

Per prima cosa è stato creato l'ambiente LAMP non prima di aver dato i comandi:

```
# sudo apt-get update
# sudo apt-get upgrade
```

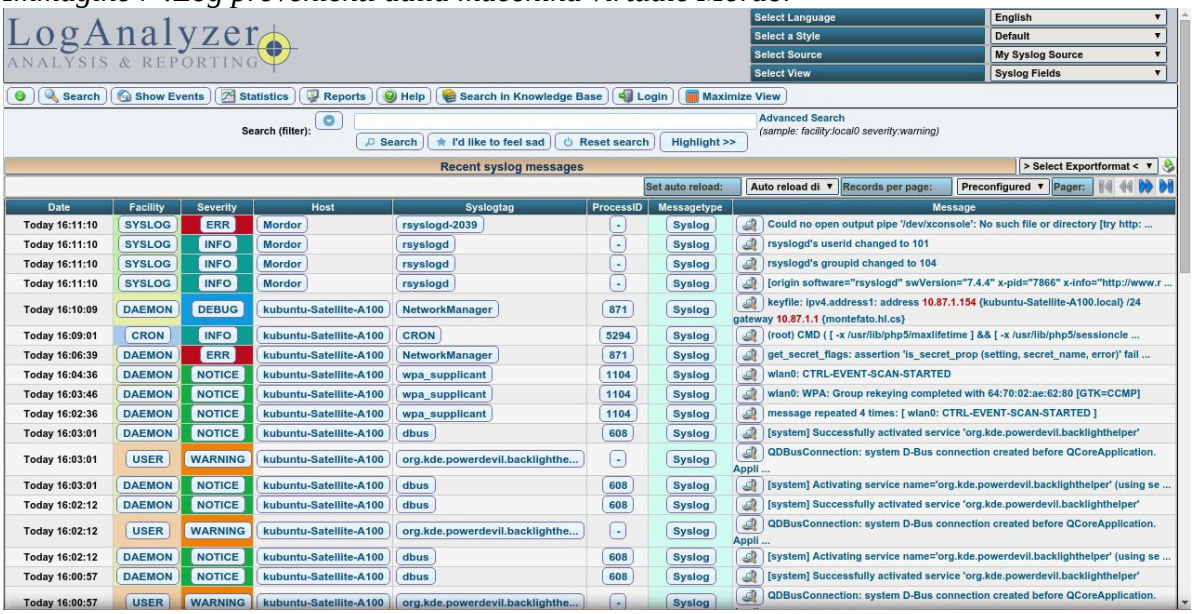
Fatto ciò, la creazione dell'ambiente è stata possibile grazie al comando:

```
# sudo apt-get install apache2 apache2-mpm-prefork php5-mysql
mysql-server php5 libapache2-mod-php5 php5-cgi php5-gd php5-
cli phpmyadmin
```

Esattamente come per la configurazione del nostro server.

Terminata la configurazione è stato modificato il file rsyslog per poter mandare tutti i log all'indirizzo ip del nostro server esattamente come è stato fatto sulla raspberry:

Immagine 7 :Log provenienti dalla macchina virtuale Mordor

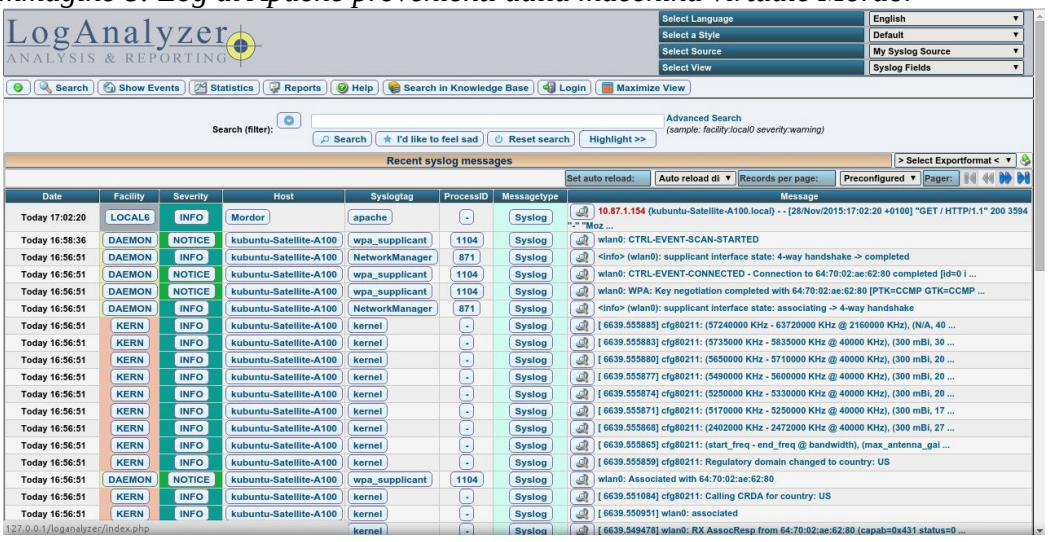


Il messaggio di ERR è stato risolto commentando alcune stringhe contenute nel file /etc/rsyslog.d/50-default.conf:

```
# daemon.*;mail.*;\
# news.err;\
# *.=debug;*.=info;\
# *.=notice;*.=warn |/dev/xconsole
```

Una volta fatto ciò, anche Apache è stato configurato per mandare i log sul server:

Immagine 8: Log di Apache provenienti dalla macchina virtuale Mordor



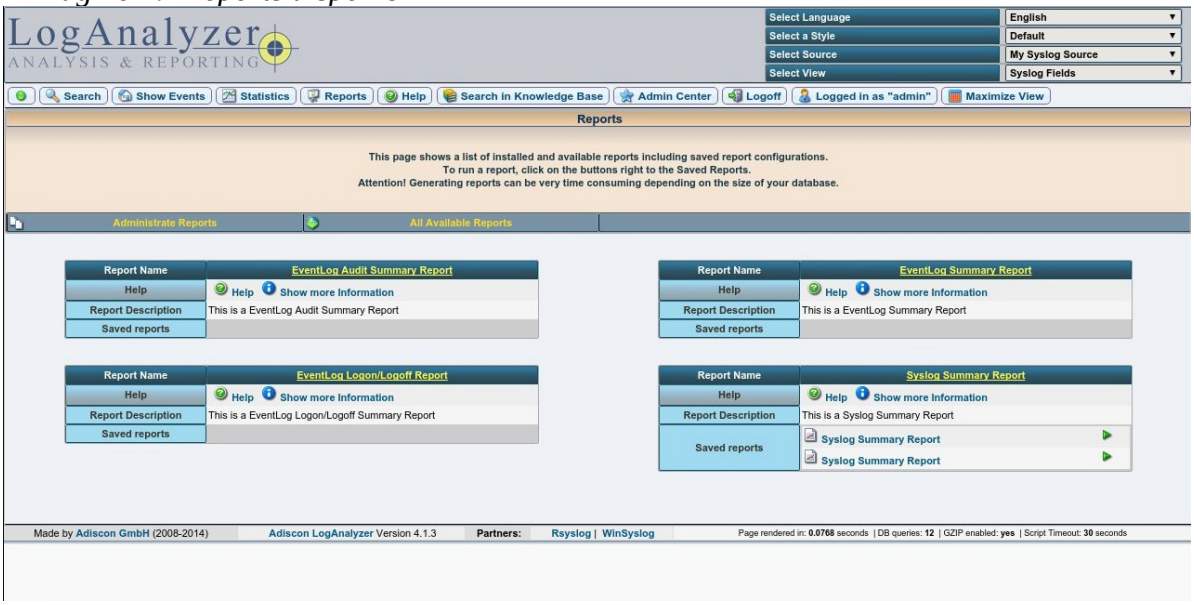
Capitolo 4 – Raccolta, analisi dati e approfondimento di LogAnalyzer

4.0.1 - Reports Server

Terminata la configurazione è arrivato il momento di utilizzare le potenzialità del programma con la raccolta e l'analisi di tutti i dati di nostro interesse. Per fare questo LogAnalyzer mette a disposizione dei reports con il quale filtrare tutti gli avvenimenti e gestirli nel miglior modo possibile infatti, è possibile configurare e visionare i reports direttamente da LogAnalyzer o, meglio ancora, inviarli tramite email o sms.

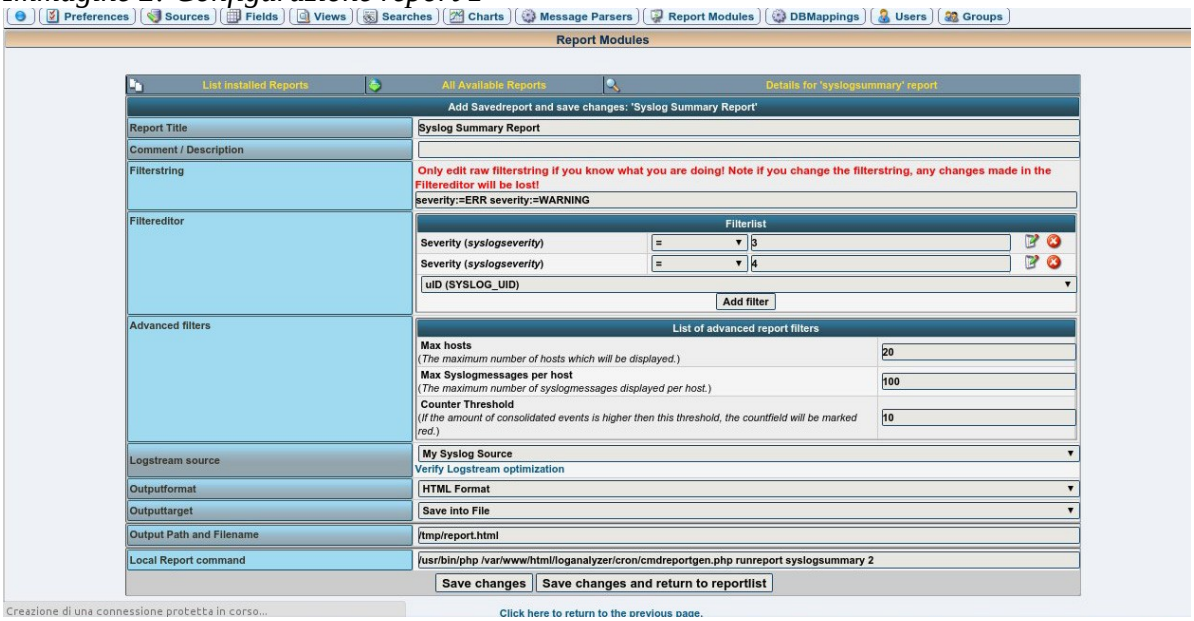
Per scoprirne il funzionamento sono stati creati alcuni reports, configurati in modo da filtrare i messaggi di Syslog che riguardano le severity 3 e 4 che corrispondono a error e warning, come è possibile vedere dall'immagine 1 . E' possibile fare in modo di visualizzare il risultato direttamente su LogAnalyzer ma, qui, è stato deciso di salvare tutto in un file html.

Immagine 1: I reports disponibili



Salvata la configurazione e premuto il tasto “play”, si ottiene il nostro file html contenente tutti i messaggi di error e warning della nostra macchina server.

Immagine 2: Configurazione report 1

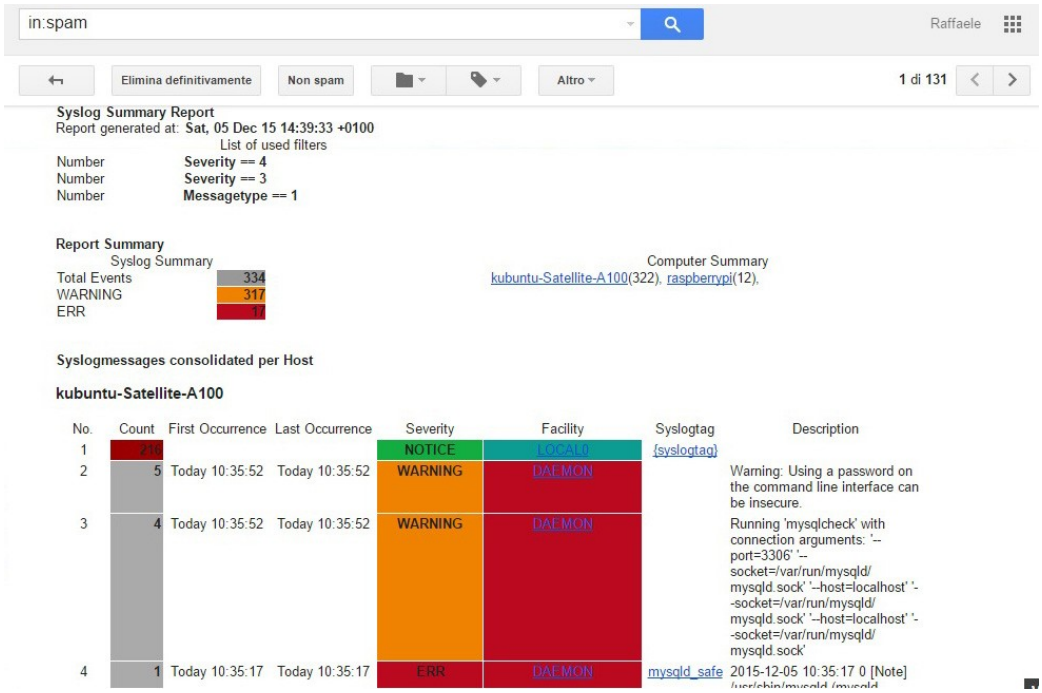


A questo punto è stato impostato, da riga di comando, l'invio di tale file sull'email utilizzando il servizio mail e l'utility CRON. Grazie a quest'ultimo è possibile, anche, temporizzare l'invio del file.

```
# /usr/bin/php/var/www/html/loganalyzer/cron/cmdreportgen.php
runreport syslogsummary 2 && mail -s "LogAnalyzer|Daily
Syslog Summary" -a "Content-type: text/html;"
raffux3@gmail.com < /tmp/report.html
```

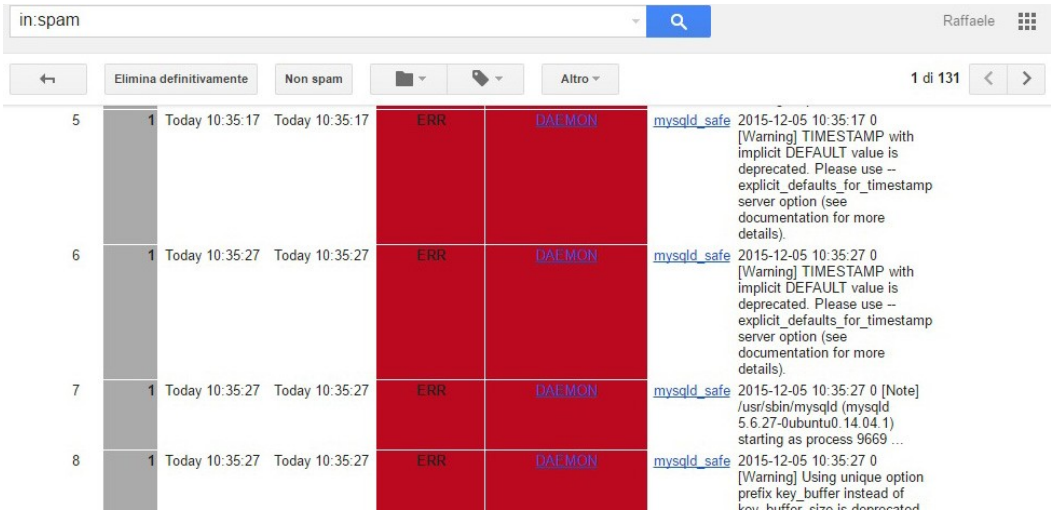
Qualche istante dopo ecco arrivato il risultato sull'email configurata:

Immagine 3: Email del report 1, 1/3



Come possiamo notare in questa schermata vengono visualizzate le prime informazioni che riepilogano il report impostato per fornire informazioni sugli eventi con Severity 3 e 4 (error e warning). Vi sono 334 eventi totali di cui 317 warnings e 17 error.

Immagine 4: Email del report 1, 2/3



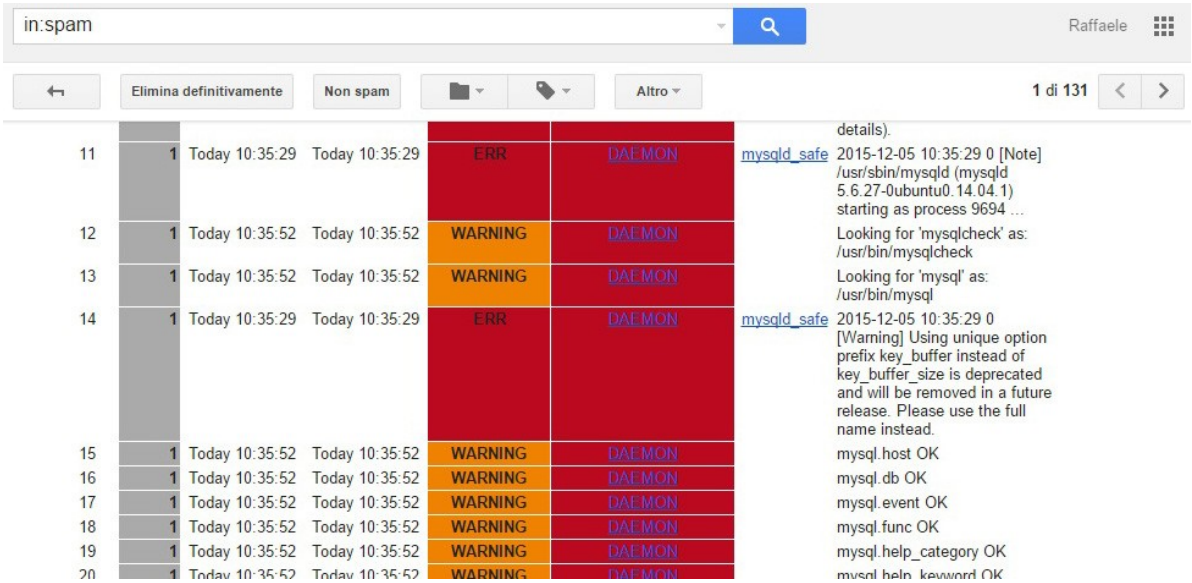
Gli error mostrati in questa immagine sono derivati dal fatto che si stava tentando di aggiornare mysql installando la versione 5.6. La macchina in questione aveva delle errate dipendenze e quindi non riusciva a scaricare i file corretti.

Mysql_safe è il modo raccomandato per far partire il server mysql ed aggiunge alcune caratteristiche di sicurezza come il riavvio del server in caso di errore e le informazioni di runtime in un file di log degli errori

I messaggi di errore sono relativi all'impossibilità di installare mysql.

E' importante notare che l'utilità di inviare questi report per email sta nel fatto di poterli visionare senza essere davanti la macchina e poter iniziare a trovare una soluzione ai vari problemi.

Immagine 5: Email del report 1, 3/3



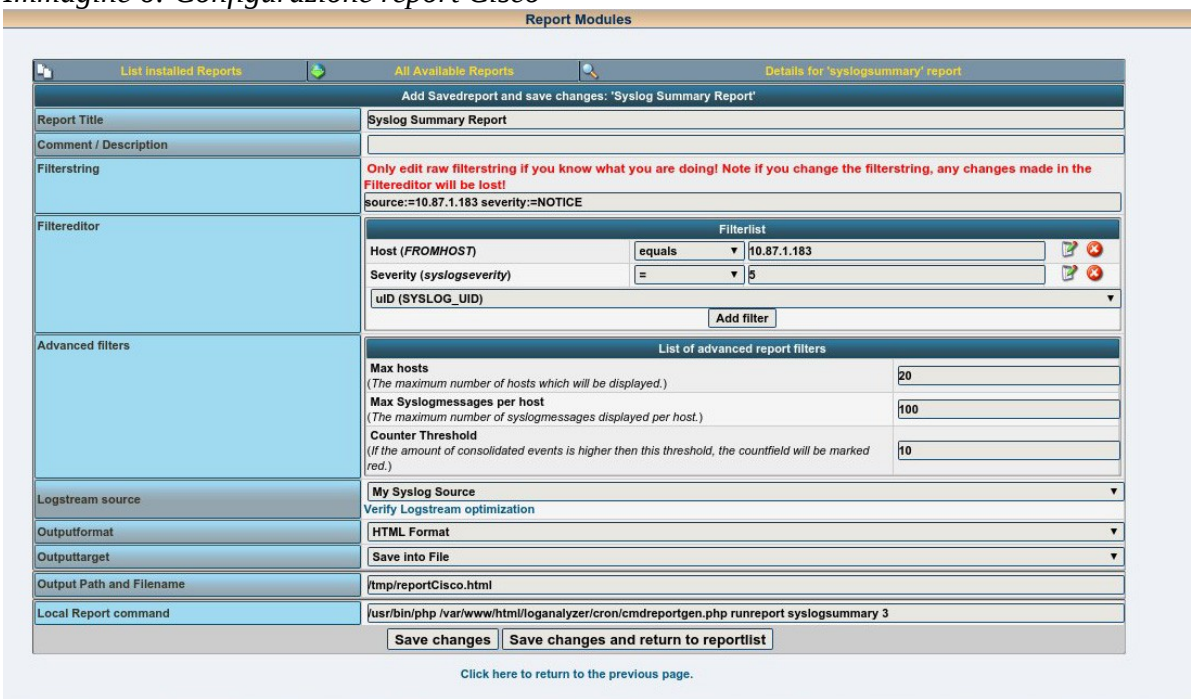
Una volta effettuate le correzioni, la versione di mysql è stata aggiornata e il servizio riavviato.

4.0.2 - Report Router Cisco

Grazie ai reports, è possibile monitorare lo stato di tutte le macchine che desideriamo controllare. Per esempio, se volessimo controllare lo stato delle interfacce del router Cisco che abbiamo precedentemente configurato, è essenziale creare un report per questa macchina.

E' importante notare che nella sezione *Filtereditor*, è stato inserito l'indirizzo ip del router Cisco, in questo modo è possibile ottenere solamente i log provenienti da quella macchina. Siccome è stato deciso di verificare quando un'interfaccia va in up o down, è stato aggiunto il filtro con Severity 5.

Immagine 6: Configurazione report Cisco

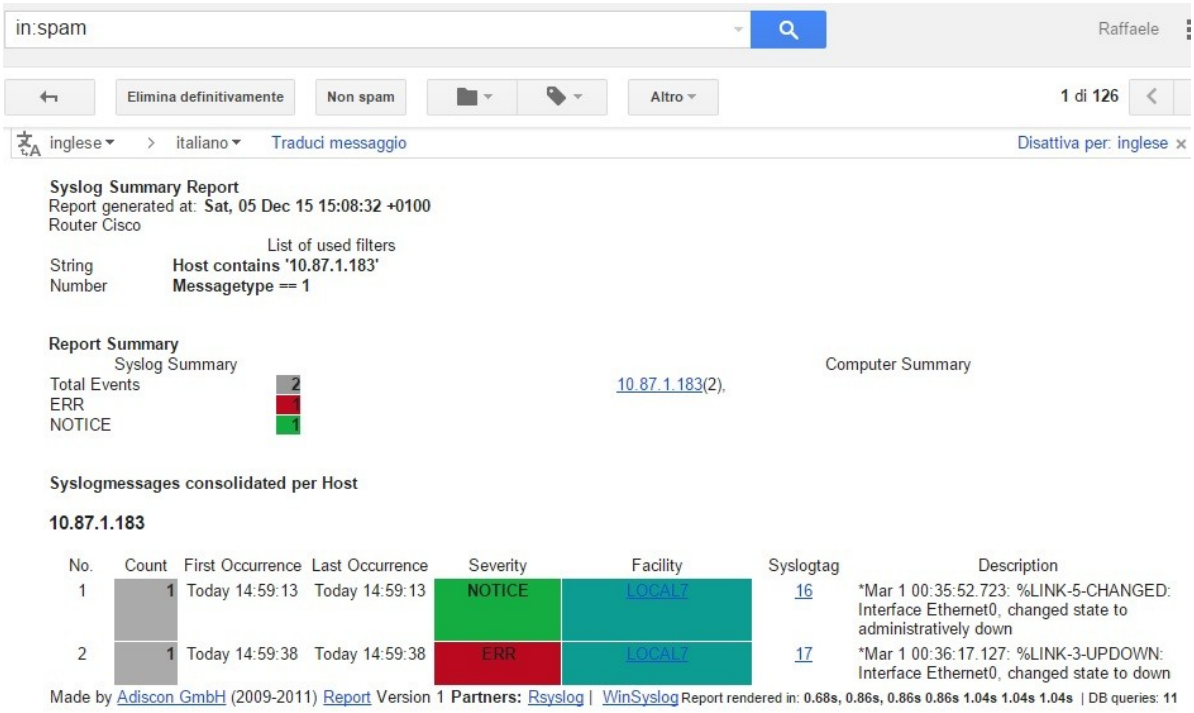


Salvata la configurazione e premuto il pulsante “play”, è stato generato il file. Anche stavolta è stato inviato sull'email inserendo da riga di comando il seguente codice, facendo attenzione ai nomi *runreport syslogsummary 3* e *reportCisco.html*, che devono differire per ogni report creato.

```
# /usr/bin/php/var/www/html/loganalyzer/cron/cmdreportgen.php
runreport syslogsummary 3 && mail -s "LogAnalyzer|DailySyslog
Summary" -a "Content-type:text/html;"raffux3@gmail.com
</tmp/reportCisco.html
```

Completata l'operazione ecco l'email arrivata:

Immagine 7: Email report Cisco

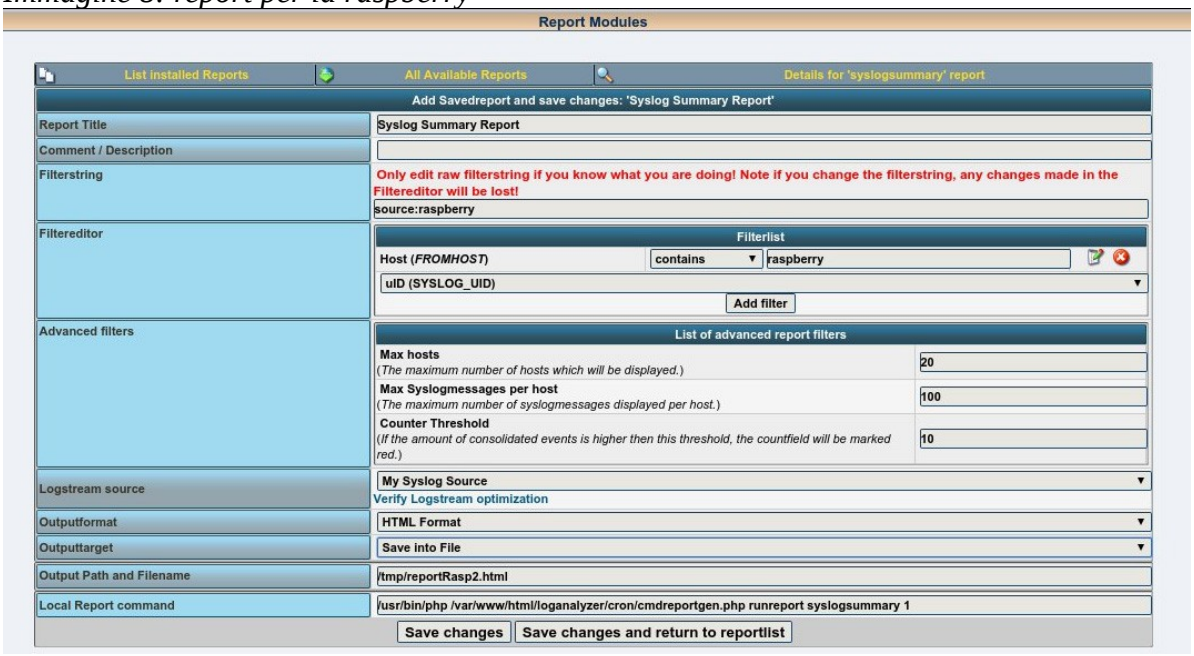


Come si nota nel report, il file generato è relativo all'host con l'indirizzo ip 10.87.1.183 che corrisponde al router Cisco. Per fare ciò è stata impostata la regola di considerare solamente i log generati dall'host 10.87.1.183. E' stato riportato l'evento Syslog relativo all'interfaccia Ethernet 0 che, volutamente, è stata messa prima in down e successivamente in up tramite i comandi di IOS quindi si è generato l'evento con Facility 5 (Notice). In questo modo siamo in grado di conoscere determinati stati di una macchina ed essere avvisati attraverso uno strumento semplice e di uso comune come l'email.

4.0.3 - Report Raspberry

Anche per la raspberry è stato configurato un report in modo da inviare, per email, ogni cosa riguardasse la macchina, infatti è stato impostato il filtro con il nome dell'host: raspberry.

Immagine 8: report per la raspberry

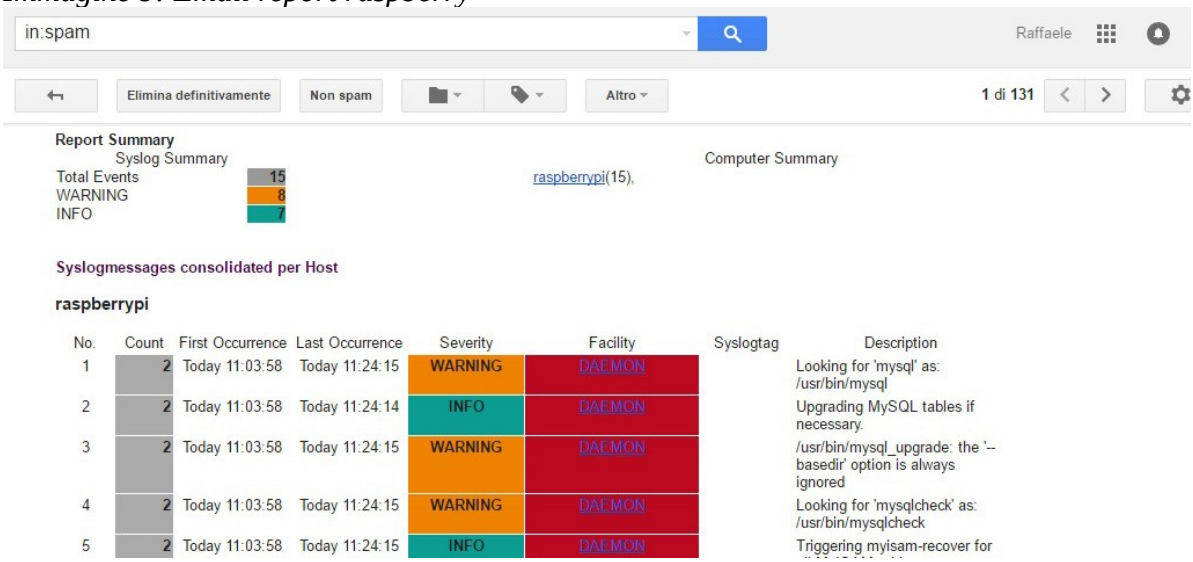


Salvata la configurazione diamo il comando:

```
# /usr/bin/php/var/www/html/loganalyzer/cron/cmdreportgen.php
runreport syslogsummary 3 && mail -s "LogAnalyzer|
DailySyslogSummary" -a "Content-
type:text/html;"raffux3@gmail.com < /tmp/reportRasp2.html
```

Pochi instanti dopo sull'email:

Immagine 9: Email report raspberry



E' semplice capire che il numero totale degli eventi sia 15 di cui 8 di tipo Warning e 7 di tipo Info.

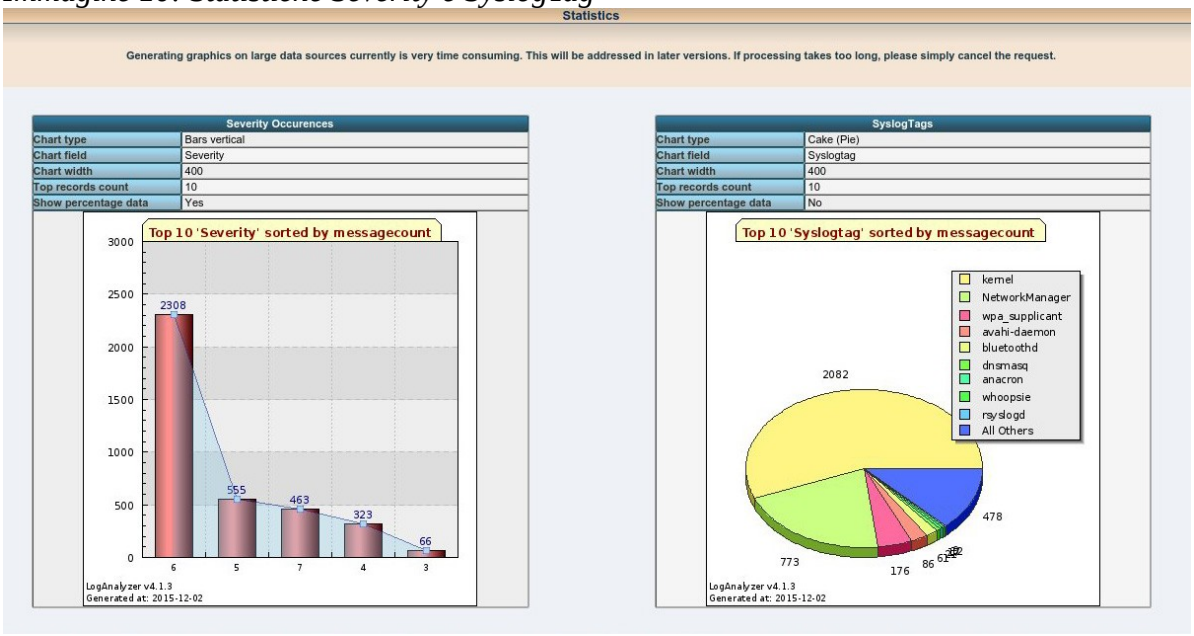
Di per se non sono problematici infatti, tutti i log, sono provenienti da mysql durante un semplice riavvio del servizio.

4.0.4 Statistiche

LogAnalyzer permette di ottenere delle statistiche personalizzate creando delle Charts, ossia dei grafici, in modo da ottenere una serie di informazioni utili allo scopo.

Dal menu *Admin Center*, sono stati selezionati 4 *charts* ognuno con uno scopo diverso e con una tipologia di grafico diversa.

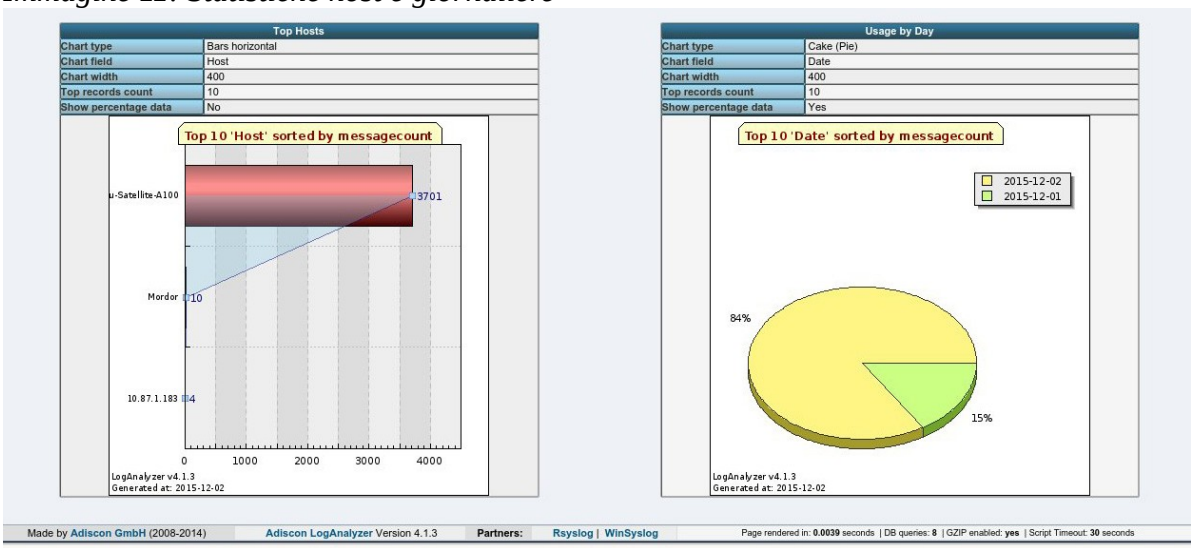
Immagine 10: Statistiche Severity e SyslogTag



Il primo chart rappresenta tutte le Severity presenti, in cui viene mostrata la percentuale di messaggi durante tutta la fase di test.

Nel secondo grafico, a torta, si evidenzia invece la percentuale dei vari tipi di SyslogTag. Come possiamo notare, la maggior parte arrivano dal kernel della macchina Server.

Immagine 11: Statistiche host e giornaliere



Gli altri due grafici si ottengono impostando Host, come campo, e rispecchia una classifica di tutti gli host collegati che hanno generato ed inviato messaggi Syslog. Anche se il server risulta il maggiore, è possibile constatare la presenza della macchina virtuale Mordor e del router Cisco (10.87.1.183).

Infine, ma non meno importante, la statistica dell'affluenza dei messaggi negli ultimi due giorni, che potrebbe dare modo di comprendere il flusso di traffico giornaliero.

Capitolo 5 – Installazione, configurazione e gestione di un syslog server la rete Ninux cosentina

Per questa parte del lavoro sono state usate le metodologie applicate in laboratorio, ma nella vita reale, in particolare sono stati monitorati alcuni nodi della rete Ninux dell'Hacklab di Cosenza. Per tali esperimenti è stata usato il pc Mordor a cui è stato assegnato un indirizzo ip privato, tale macchina è stata configurata per ricevere i log.

5.0.1 - Cos'è Ninux.org

Ninux.org è una community wireless che collabora allo sviluppo di un progetto per la creazione di una rete aperta, decentralizzata di cui il cittadino ne è proprietario. Tale progetto si traduce in un'infrastruttura di rete come parte integrante di Internet, lo scopo non è fornire accesso ad esso ma, grazie a questa rete, diventa facile diffondere, conoscere, sperimentare, creare e produrre tecnologie ed avere libertà di espressione.

Il nome Ninux si riferisce al nome del fondatore del progetto, Nino, anche se alcuni lo interpretano come l'acronimo di Neighborhood Internet, Network Under eXperiment.

Il movente della community, quindi, è la condivisione delle conoscenze e delle esperienze messe a disposizione da tutti, con la realizzazione di reti wireless libere, senza scopo di lucro e basandosi sulla filosofia del software libero, reclamando il diritto di interconnettersi e sperimentare in modo totalmente libero da strumentalizzazioni politiche e commerciali. Per questo non è un ISP infatti, Ninux.org, può essere considerata una comitiva di amici che discutono di interessi comuni, sperimentano e si divertono nel farlo.

Chiunque può entrare in una rete Ninux a patto che siano rispettate alcune regole: libero scambio di dati senza interferire, libera pubblicazione di informazioni senza vincoli di licenza così come tutti i servizi messi a disposizione.

Immagine 1: Nodi Ninux Attivi in Italia

In Calabria, è l'Hacklab di Cosenza ad aver per prima avviato l'isola cosentina e, tutt'ora, è la più grande rete dopo quella di Roma infatti conta ben 30 nodi attivi. I membri dell'associazione sono stati invitati a partecipare a questo progetto attivando i nodi per l'invio dei log al server Mordor, mettendo a disposizione le macchine, i servizi ed il loro tempo. Ricordo che Ninux è una rete comunitaria ed indica l'uso di tecnologie in rete al servizio di una comunità locale.

5.0.2 - I nodi

Un nodo della rete è un insieme di componenti hardware e software situati nel medesimo luogo e sono necessari per interconnettersi con gli altri nodi della rete. Un nodo è un luogo fisico attraverso il quale si scambiano i dati, un membro della rete si collega alle antenne del

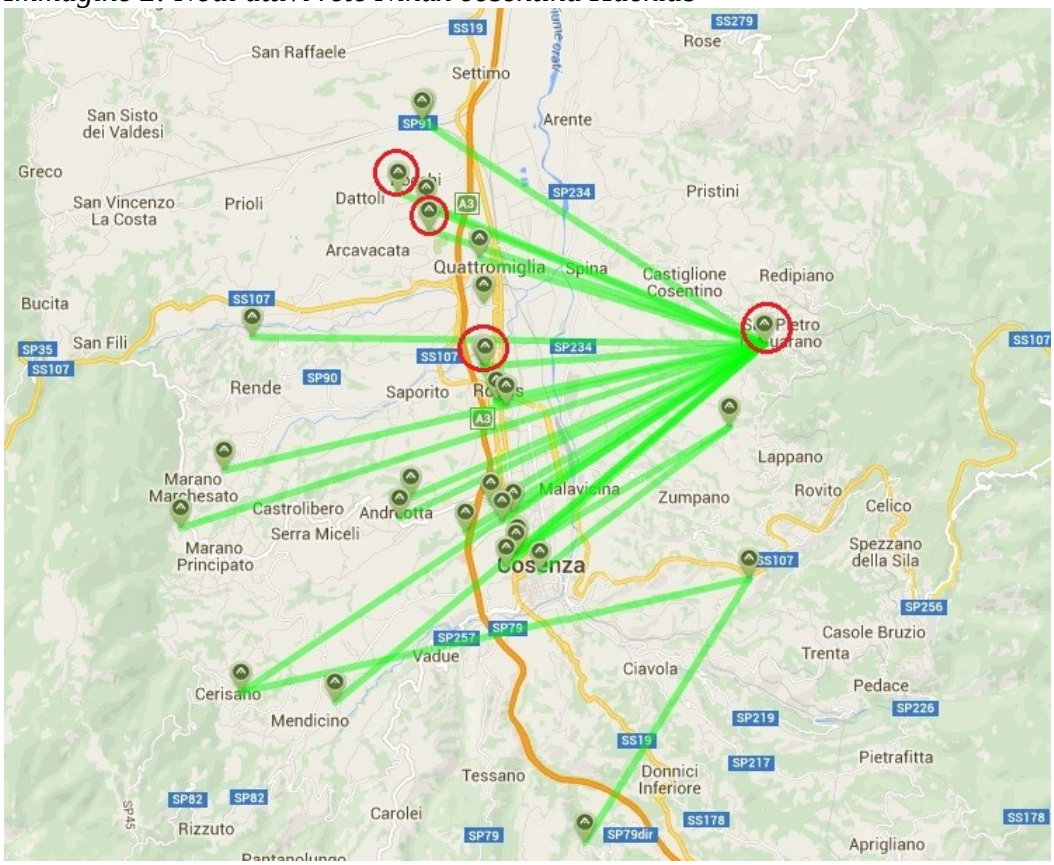
nodo, collegate a loro volta con le altre antenne via wireless, attraverso un cavo Ethernet.

I nodi configurati per inviare i log sulla server Mordor, sono rappresentati da antenne e router:

- 10.87.2.253 ubiquiti nanostation 5.
- 10.87.2.254 ubiquiti powerbim M5.
- 10.87.19.253 ubiquiti M5.
- 10.87.19.254 ubiquiti nanobim M5 16.
- 10.87.1.3 ubiquiti nanobim M5 19.
- 172.17.87.4 NewSpig (AP PtP HPCC).
- 172.17.87.38 Algemfeac.
- 172.87.3.253

Tali nodi saranno abilitati all'invio dei log ed esaminati per constatare l'attività e i possibili errori. Alcuni di essi avranno una data diversa da quella di altri e non aggiornata perché non sincronizzati.

Immagine 2: Nodi attivi rete Ninux cosentina Hacklab



5.0.3 - Configurazione del server Mordor

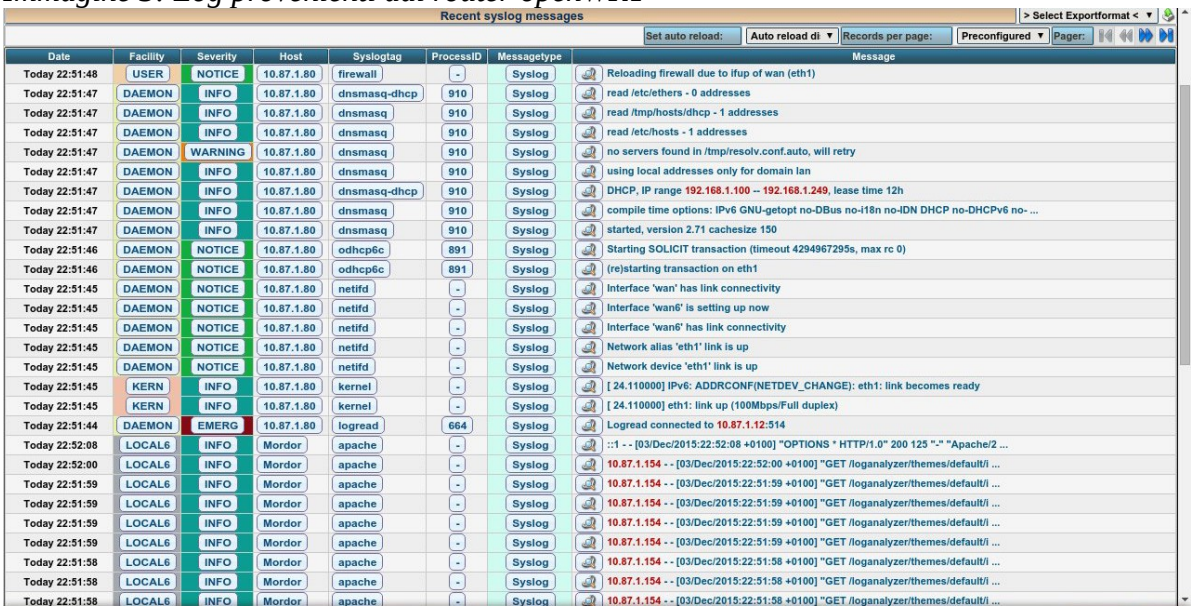
Il computer fisico con installata la macchina virtuale messa a disposizione ,precedentemente, fungeva da client ed era stata configurata per inviare i file syslog alla macchina configurata come server (10.87.1.154).

Adesso il client diventa server per cui è stato installato LogAnalyzer e configurato per ricevere e visualizzare i log secondo il formato stabilito nel RFC 5424.

Le configurazioni sono le medesime eseguite nella precedente macchina server a differenza dell'indirizzo ip inserito nel browser per la web gui di LogAnalyzer: 10.87.1.12/Loganalyzer.

Terminata l'installazione e la configurazione di LogAnalyzer è stato configurato il router OpenWRT per l'invio dei log al nuovo server:

Immagine 3: Log provenienti dal router openWRT



Recent syslog messages

Date	Facility	Severity	Host	Syslogtag	ProcessID	Messagetype	Message
Today 22:51:48	USER	NOTICE	10.87.1.80	firewall	-	Syslog	Reloading firewall due to ifup of wan (eth1)
Today 22:51:47	DAEMON	INFO	10.87.1.80	dnsmasq-dhcp	910	Syslog	read /etc/ethers - 0 addresses
Today 22:51:47	DAEMON	INFO	10.87.1.80	dnsmasq	910	Syslog	read /tmp/hosts/dhcp - 1 addresses
Today 22:51:47	DAEMON	INFO	10.87.1.80	dnsmasq	910	Syslog	read /etc/hosts - 1 addresses
Today 22:51:47	DAEMON	WARNING	10.87.1.80	dnsmasq	910	Syslog	no servers found in /tmp/resolv.conf.auto, will retry
Today 22:51:47	DAEMON	INFO	10.87.1.80	dnsmasq	910	Syslog	using local addresses only for domain lan
Today 22:51:47	DAEMON	INFO	10.87.1.80	dnsmasq-dhcp	910	Syslog	DHCP, IP range 192.168.1.100 -- 192.168.1.249, lease time 12h
Today 22:51:47	DAEMON	INFO	10.87.1.80	dnsmasq	910	Syslog	compile time options: IPv6 GNU-getopt no-DBus no-i18n no-IDN DHCP no-DHCPv6 no- ...
Today 22:51:47	DAEMON	INFO	10.87.1.80	dnsmasq	910	Syslog	started, version 2.71 cachesize 150
Today 22:51:46	DAEMON	NOTICE	10.87.1.80	odhcp6c	891	Syslog	Starting SOLICIT transaction (timeout 4294967295s, max rc 0)
Today 22:51:46	DAEMON	NOTICE	10.87.1.80	odhcp6c	891	Syslog	(re)starting transaction on eth1
Today 22:51:45	DAEMON	NOTICE	10.87.1.80	netifd	-	Syslog	Interface 'wan' has link connectivity
Today 22:51:45	DAEMON	NOTICE	10.87.1.80	netifd	-	Syslog	Interface 'wan6' is setting up now
Today 22:51:45	DAEMON	NOTICE	10.87.1.80	netifd	-	Syslog	Interface 'wan6' has link connectivity
Today 22:51:45	DAEMON	NOTICE	10.87.1.80	netifd	-	Syslog	Network alias 'eth1' link is up
Today 22:51:45	DAEMON	NOTICE	10.87.1.80	netifd	-	Syslog	Network device 'eth1' link is up
Today 22:51:45	KERN	INFO	10.87.1.80	kernel	-	Syslog	[24.110000] IPv6: ADDRCONF(NETDEV_CHANGE): eth1: link becomes ready
Today 22:51:45	KERN	INFO	10.87.1.80	kernel	-	Syslog	[24.110000] eth1: link up (100Mbps/Full duplex)
Today 22:51:44	DAEMON	EMERG	10.87.1.80	logread	664	Syslog	Logread connected to 10.87.1.12:514
Today 22:52:08	LOCAL6	INFO	Mordor	apache	-	Syslog	::1 - - [03/Dec/2015:22:52:08 +0100] "OPTIONS * HTTP/1.0" 200 125 "-" "Apache/2 ...
Today 22:52:00	LOCAL6	INFO	Mordor	apache	-	Syslog	10.87.1.154 - - [03/Dec/2015:22:52:00 +0100] "GET /loganalyzer/themes/default/i ...
Today 22:51:59	LOCAL6	INFO	Mordor	apache	-	Syslog	10.87.1.154 - - [03/Dec/2015:22:51:59 +0100] "GET /loganalyzer/themes/default/i ...
Today 22:51:59	LOCAL6	INFO	Mordor	apache	-	Syslog	10.87.1.154 - - [03/Dec/2015:22:51:59 +0100] "GET /loganalyzer/themes/default/i ...
Today 22:51:59	LOCAL6	INFO	Mordor	apache	-	Syslog	10.87.1.154 - - [03/Dec/2015:22:51:59 +0100] "GET /loganalyzer/themes/default/i ...
Today 22:51:59	LOCAL6	INFO	Mordor	apache	-	Syslog	10.87.1.154 - - [03/Dec/2015:22:51:59 +0100] "GET /loganalyzer/themes/default/i ...
Today 22:51:58	LOCAL6	INFO	Mordor	apache	-	Syslog	10.87.1.154 - - [03/Dec/2015:22:51:58 +0100] "GET /loganalyzer/themes/default/i ...
Today 22:51:58	LOCAL6	INFO	Mordor	apache	-	Syslog	10.87.1.154 - - [03/Dec/2015:22:51:58 +0100] "GET /loganalyzer/themes/default/i ...
Today 22:51:58	LOCAL6	INFO	Mordor	apache	-	Syslog	10.87.1.154 - - [03/Dec/2015:22:51:58 +0100] "GET /loganalyzer/themes/default/i ...

Il log del router OpenWRT configurato con l'indirizzo ip 10.87.1.80 sono stati inviato al server con successo.

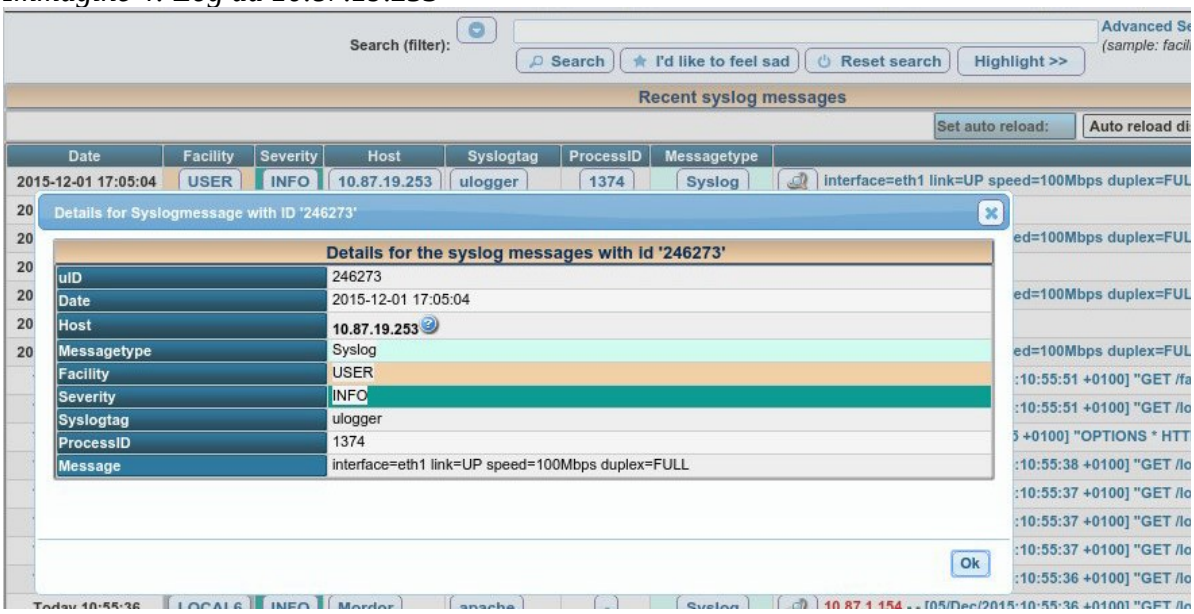
A questo punto è stato creato un report anche per Mordor in modo da conservare tutti i log di Warning ed Error, quindi con severity 3 e 4, in un file html e questo, successivamente, inviato sull'email ogni mattina alle 8.00 quindi, entrando nel file *crontab -e* come amministratore, è stato aggiungo il seguente comando:

```
# 0 8 * * *
/usr/bin/php/var/www/html/loganalyzer/cron/cmdreportgen.php
runreport syslogsummary 3 && mail -s "LogAnalyzer|Daily
Syslog Summary" -a "Content-type: text/html;"
raffux3@gmail.com < /tmp/report.html
```

5.1 Analisi dei risultati ottenuti

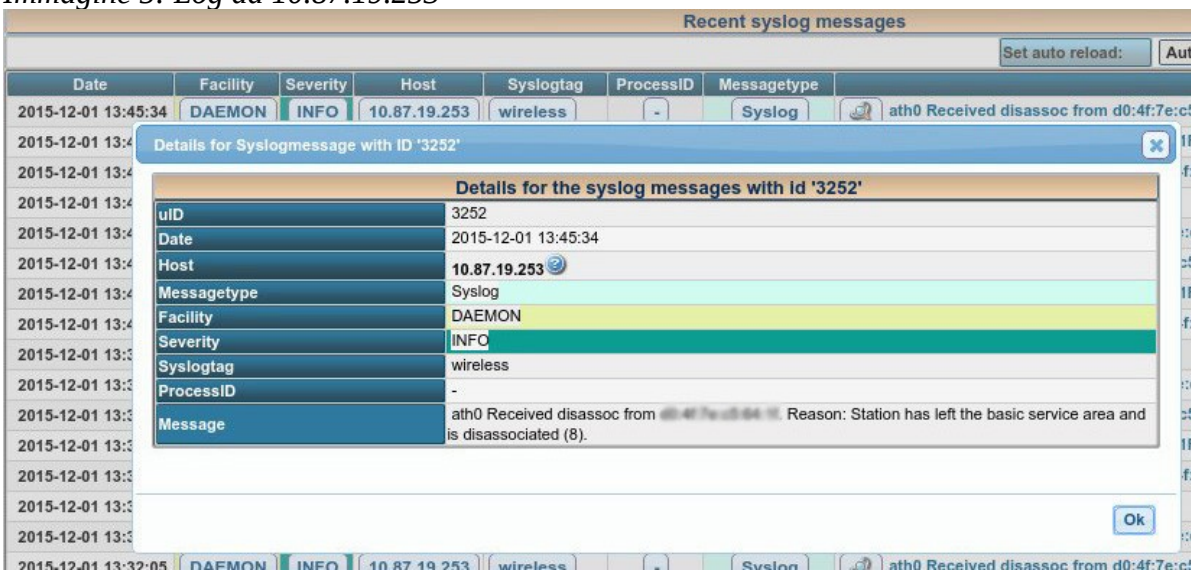
I primi log ad arrivare sono stati quelli provenienti dal nodo 10.87.19.253:

Immagine 4: Log da 10.87.19.253



Questo log, che è di Severity INFO, ci avverte che l'interfaccia *eth1* è UP, la velocità è di 100Mbps e che la modalità di trasmissione configurata è full duplex quindi sarà possibile ricevere ed inviare contemporaneamente. La Facility in questo caso è 1 (USER) e sono messaggi a livello utente.

Immagine 5: Log da 10.87.19.253



Il messaggio con id *3252*, di tipi INFO, ci avvisa che l'interfaccia *ath0* ha ricevuto una dissociazione da parte di un host (indicato con l'indirizzo mac) e la ragione è che la stazione è uscita dalla *basic service area (bsa)* e si è dissociata. La Facility di questo log è di tipo 3 ossia DAEMON che riguarda un demone di sistema.

Immagine 6: Log da 10.87.19.253

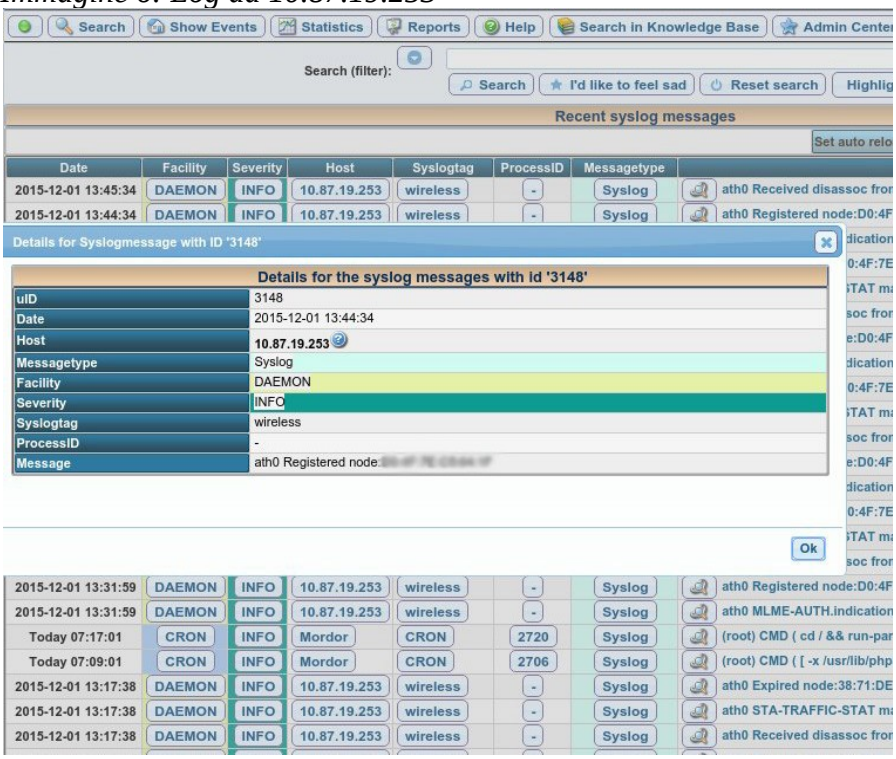


Immagine 7: Log da 10.87.19.253

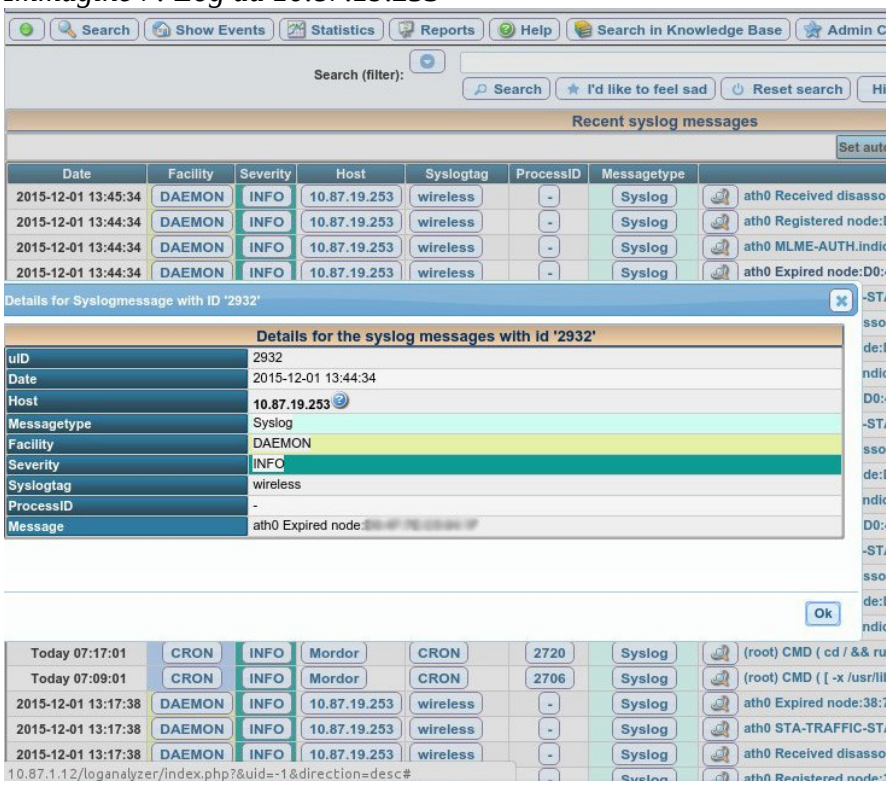
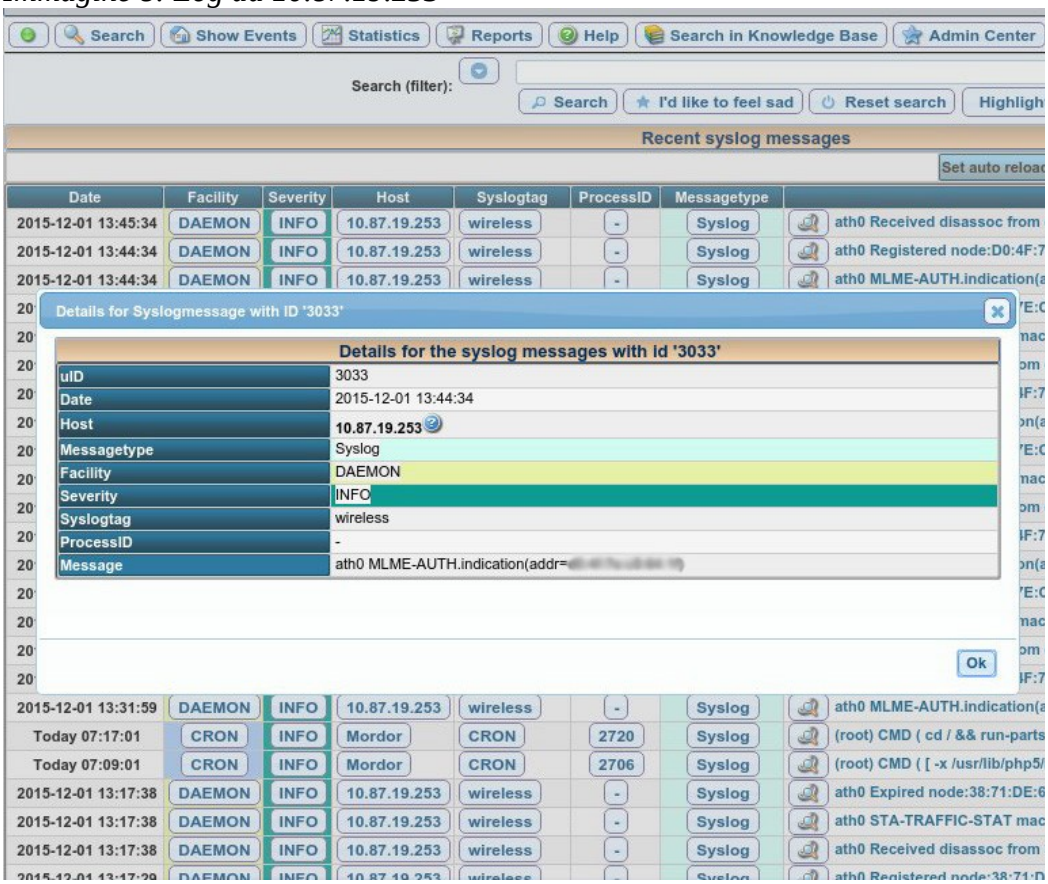
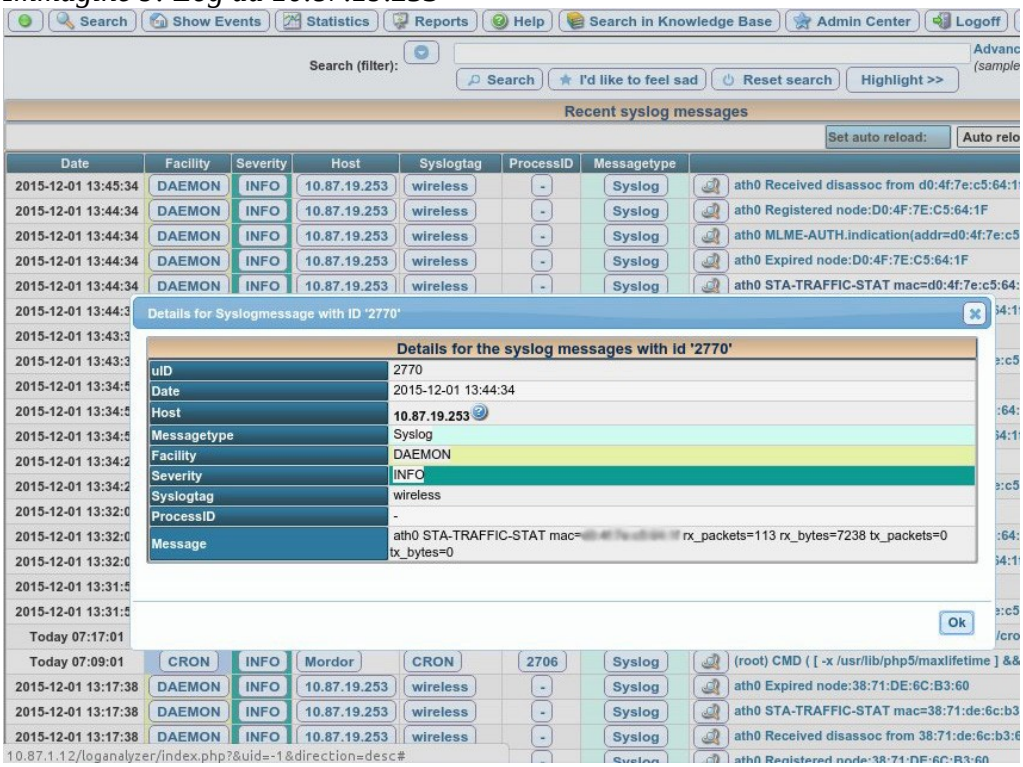


Immagine 8: Log da 10.87.19.253



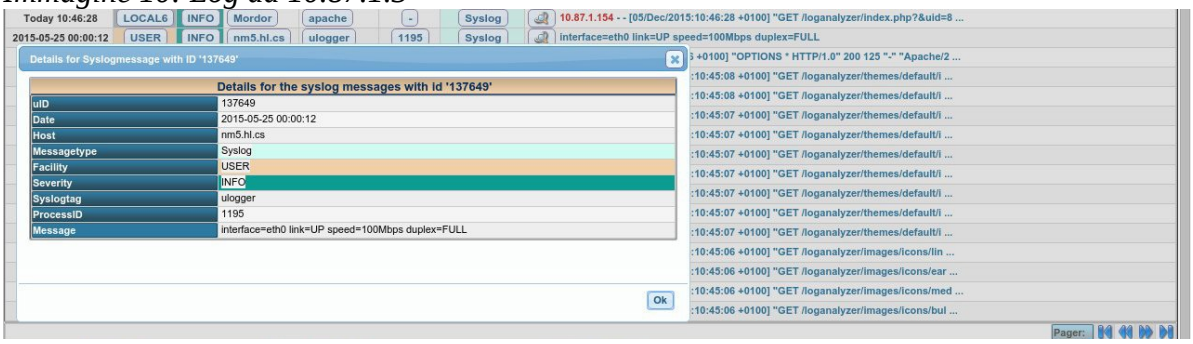
Il messaggio *ath0 mlme-auth.indication(addr=XX-XX-XX-XX-XX)* riguarda lo stesso indirizzo mac delle due precedenti illustrazioni. Questo si pone dopo che l'host si è associato ed è pronto per l'autenticazione. Il fatto che si siano susseguiti messaggi di dissociazione non implica un problema grave (come indicato dalla Severity), ma potrebbe essere legato ad una questione di mal-configurazione, oppure delle attente attività fatte da un amministratore.

Immagine 9: Log da 10.87.19.253



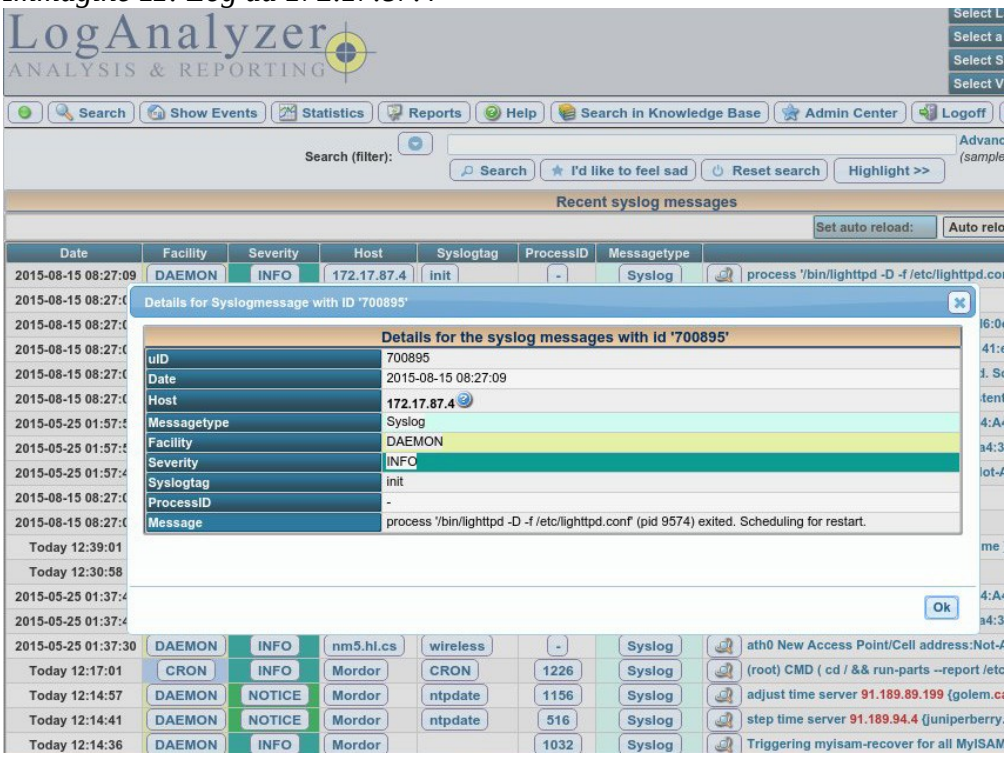
In questo log viene comunicato che c'è stato del traffico tra l'host e la stazione. In particolare sono stati ricevuti 113 pacchetti per una dimensione totale di 7238 byte e non è stato trasmesso alcun pacchetto.

Immagine 10: Log da 10.87.1.3



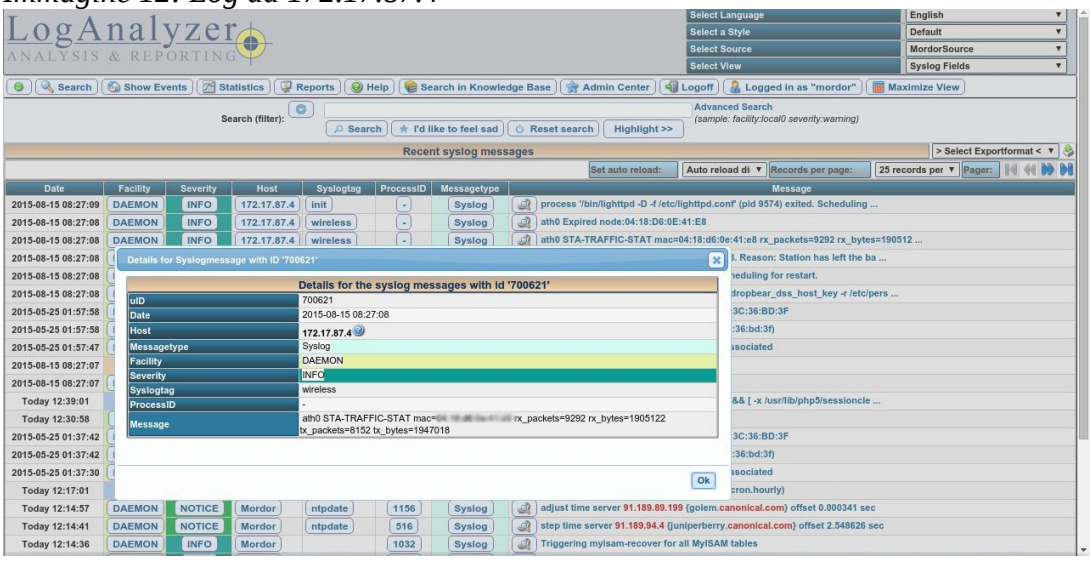
Qualche log è arrivato anche dal nodo 10.87.1.3 chiamato *nm5.hl.cs* in cui viene specificato che l'interfaccia *eth0* è attiva, il link viaggia a 100Mbps e in modalità full duplex.

Immagine 11: Log da 172.17.87.4



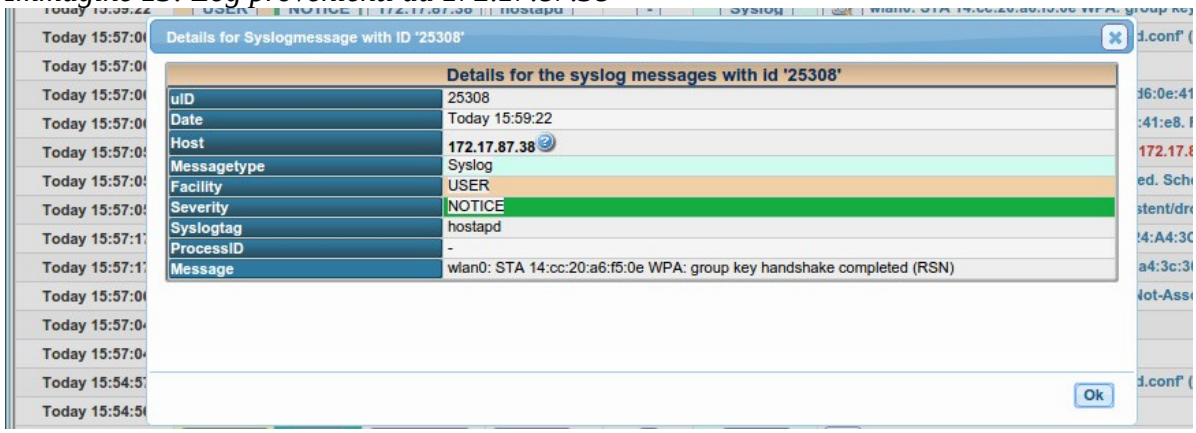
Tale log, proveniente da 172.17.87.4, indica che il processo *9574* è uscito e che è stato programmato il riavvio. Il Syslogtag *init* specifica che si tratta di uno script contenuto nel percorso */etc/init-d/*.

Immagine 12: Log da 172.17.87.4



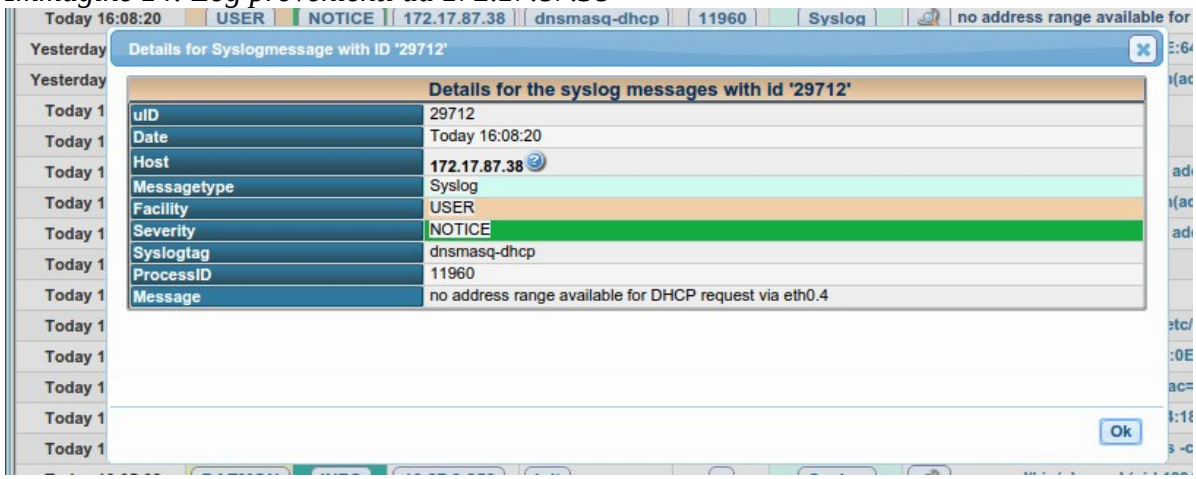
Anche qui viene specificato che c'è stato del traffico tra l'host e la stazione come nella precedente immagine a differenza della tipologia traffico che, infatti è diverso: sono stati ricevuti 9292 pacchetti, come indica la voce *rx_packets*, per un totale di 1905122 bytes mentre sono stati trasmessi 8152 pacchetti, come si nota dalla voce *tx_packets* arrivando a 1947018 bytes.

Immagine 13: Log provenienti da 172.17.87.38



Il messaggio di log dall'host 172.17.87.38 è di tipo 5 (Notice) evidenzia come l'handshake per lo scambio della WPA tra host e station sia stato completato. La Facility è User e il Syslogtag è *hostapd,* usato per l'autenticazione negli *IEEE 802.11 AP e IEEE 802.1X/WPA/WPA2/EAP/RADIUS.*

Immagine 14: Log provenienti da 172.17.87.38



Questo log indica anche il *ProcessID 11960* e contiene il messaggio che non ci sono indirizzi disponibili nel range per il *DHCP request* attraverso l'interfaccia eth0.4. *Dnsmasq-dhcp* da solo indirizzi nello stesso blocco.

Immagine 15: Log provenienti da 172.17.87.38

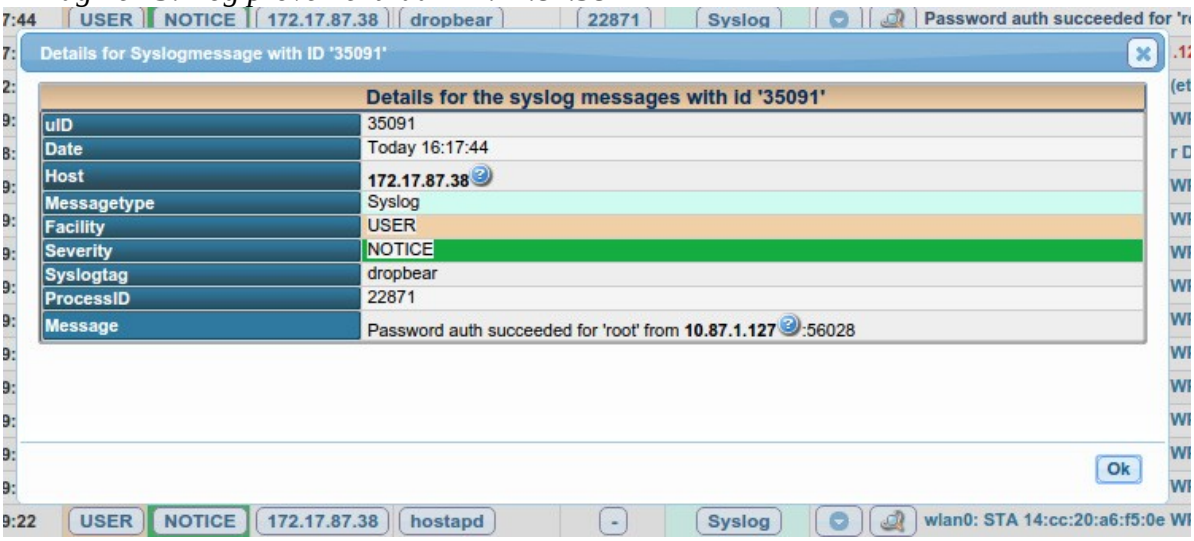
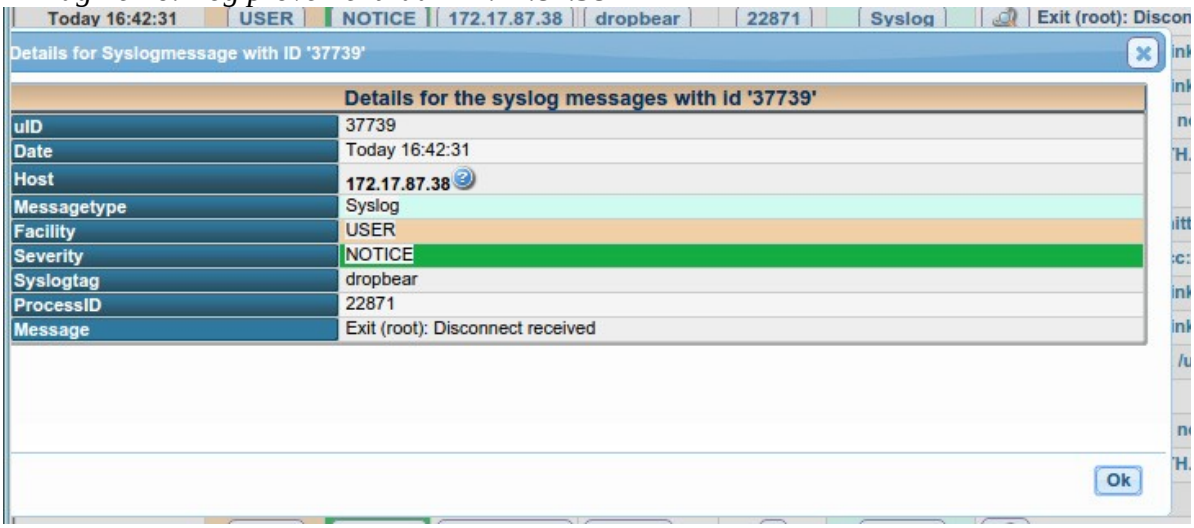
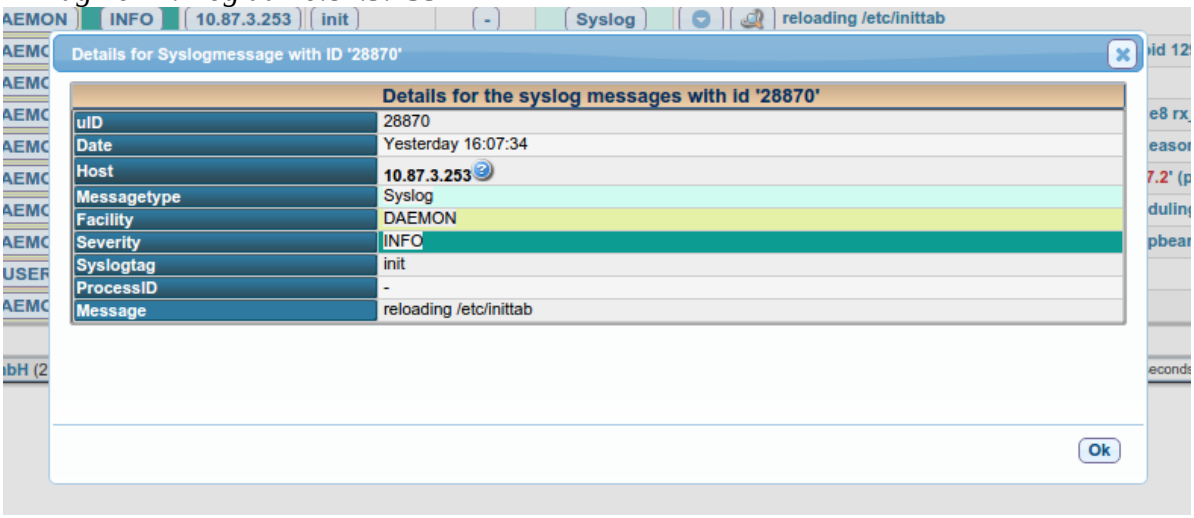


Immagine 16: Log provenienti da 172.17.87.38



Le precedenti due illustrazioni indicano, rispettivamente, che è stata inserita la password corretta per l'utente root e che ,successivamente, lo stesso utente ha mandato una richiesta di disconnessione ricevuta correttamente.

Immagine 17: Log da 10.87.3.253



Infine, quest'ultima immagine del messaggio syslog dal nodo 10.87.3.253 indica che il servizio inittab è in fase di ricaricamento.

Immagine 1: Statistiche dei log

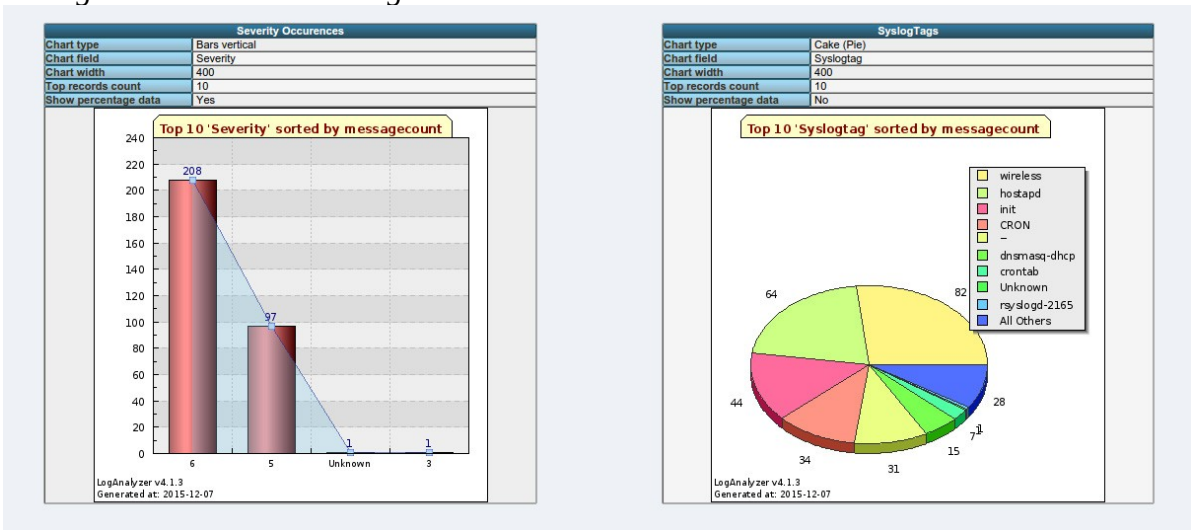
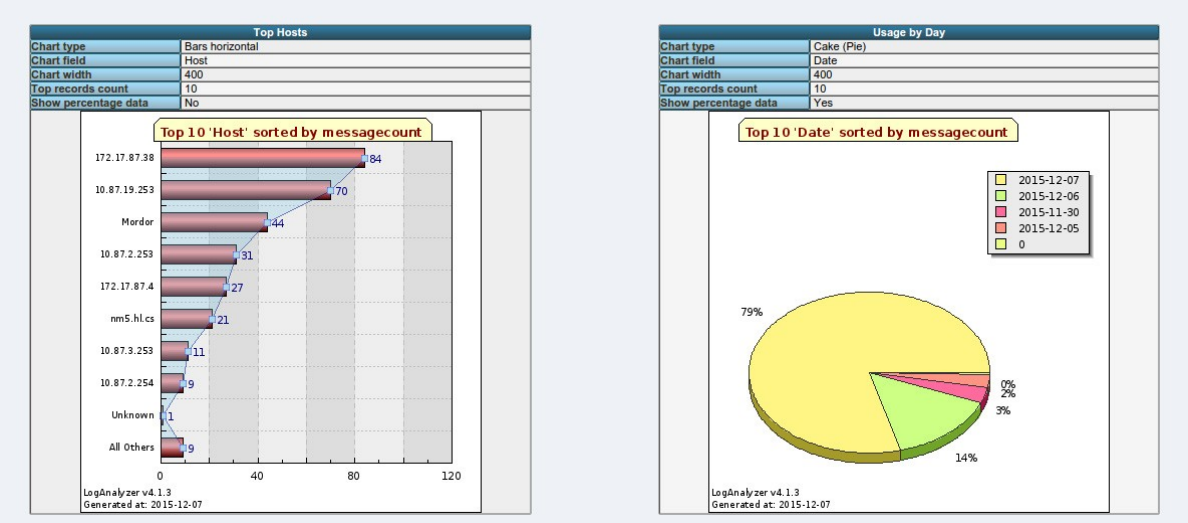


Immagine 18: Statistiche dei Log



Dopo aver effettuato 3 giorni di ricezione ed analisi del messaggi Syslog provenienti dai nodi, non è stato riscontrato alcun problema di rete né qualcosa che potesse destare preoccupazione. Come possiamo vedere dai primi due grafici, la maggior parte dei messaggi sono stati di livello 6 di Severity (INFO), il resto di livello 5 (NOTICE) e solo uno di livello 3 (ERR) che riguardava la macchina Mordor. Il primo grafico a torta stila la percentuale di syslotag ricevuti, il maggiore è di tipo wireless seguito da hostapd e init.

Nella seconda immagine, invece, possiamo vedere il grafico a colonne che evidenzia il numero di messaggi provenienti dai vari host. Il primo in classifica risulta il 172.17.87.38 che è un router, mentre il secondo è il 10.87.19.253 che riguarda un'antenna. Probabilmente l'alta intensità di traffico proveniente da quest'ultimo host si riferisce al fatto che molti utenti hanno tentato di collegarsi alla rete tramite quella antenna in quanto priva di meccanismo di autenticazione con password. Infine, il secondo grafico a torta illustra i 3 giorni, che vanno dal 5 dicembre al 7, che sono stati impiegati per la raccolta dei messaggi. La data 30-11-2015 risulta, come spiegato prima, per il fatto che alcune antenne non avevano la data sincronizzata.

Conclusioni

Il presente lavoro di tesi conclude un percorso di studi arricchendo il proprio bagaglio culturale per quanto riguarda la gestione delle reti, utile all'interno di un'azienda per amministrare il traffico ed effettuare, laddove occorre, tempestivo troubleshooting. L'utilizzo del software LogAnalyzer potrebbe essere utile accoppiato con altri software più completi per tenere traccia non solo dei vari messaggi di syslog, ma anche del carico della cpu, ottenere i risultati attraverso grafici in tempo reale ed inviare dei trigger all'amministratore non appena avviene qualsiasi tipo di evento importante.

L'analisi svolta nel corso della sperimentazione non ha riportato problemi gravi ma, fortunatamente, solo degli innocui messaggi dai vari nodi configurati che rappresentavano situazioni di riavvio dei servizi, connessioni e disconnessioni ad un nodo e un resoconto dell'ammontare dei dati inviati e ricevuti. I grafici sono stati utili nel stilare un resoconto di una serie di informazioni utili per lo scopo del lavoro svolto.

Il protocollo Syslog non era una completa novità, ma un suo approfondimento è stato utile nella comprensione di quanto ci sia dietro a qualcosa che le persone lontane da questo ambito ignorano. Non è stato semplice il primo approccio a questo tipo di ambiente. La parte più complessa è stata quella di utilizzare un software completamente nuovo e capirne, in poco tempo, tutte le sue potenzialità. Con continue ricerche, prove e rifacimento di tutto il processo, alla fine il risultato ottenuto è stato quello di avere un software front-end che leggesse in maniera adeguata il formato del log e sopratutto che generasse in maniera perfetta tutti i reports.

Tutto il lavoro in laboratorio svolto, avere a che fare con hardware nuovo come la raspberry ed un router openWRT, configurare un server avvicinandomi al sistema Linux, è stato altamente istruttivo e divertente, oltre ad arduo, e la riuscita del lavoro molto appagante. I risultati ottenuti, sperimentando le varie situazioni come la manomissione di un'interfaccia del router o il grande flusso di traffico proveniente da un'antenna, quindi in un ambiente reale, mi ha permesso di capire come un sistema di monitoraggio della rete non sia da sottovalutare ma rappresenta un potente strumento in grado di dare informazioni su una molteplicità di situazioni.

Sitografia

[1] SYSLOG

https://tools.ietf.org/html/rfc5424

https://en.wikipedia.org/wiki/Comparison_of_network_monitoring_systems

[2] NTOPG

http://www.ntop.org/products/traffic-analysis/ntop/

[3] GANGLIA

http://ganglia.sourceforge.net/

https://en.wikipedia.org/wiki/Ganglia_(software)

http://www.slashroot.in/introduction-ganglia-monitoring-and-graphing-tool

[4] MUNIN

https://en.wikipedia.org/wiki/Munin_(software)

http://munin-monitoring.org/

https://it.wikipedia.org/wiki/Munin

[5] MONIT

https://en.wikipedia.org/wiki/Monit

https://mmonit.com/monit/documentation/monit.html

[6] ZABBIX

http://www.zabbix.com/

http://zabbixitalia.it/page/cose-zabbix

[7] SNMP

http://it.ccm.net/contents/32-il-protocollo-snmp

http://openskill.info/topic.php?ID=169

http://www.whatsupgold.com/blog/2011/04/27/which-is-better-snmp-or-syslog/

https://it.wikipedia.org/wiki/Simple_Network_Management_Protocol

http://www.acmeconsulting.it/Squid-Book/HTML/sec-parole-snmp.html

http://www.logicmonitor.com/blog/2012/10/05/whats-with-the-different-snmp-versions-s1-v2c-v3/

http://www.tcpipguide.com/free/t_SNMPProtocolBasicRequestResponseInformationPol

lUsi.htm

[8] SYSLOG

http://www.excitingip.com/421/an-overview-of-syslog-and-syslog-server/

http://www.rexconsulting.net/tip-centralized-logging-benefits.html

http://lwn.net/Articles/369075/

http://www.ciscopress.com/articles/article.asp?p=426638

http://freelinuxtutorials.com/tutorials/configure-centralized-syslog-server-in-linux-setup-syslog-clients-on-different-platforms/

[9] CACTI

https://en.wikipedia.org/wiki/Cacti_(software)

[10] LOGANALYZER

http://loganalyzer.adiscon.com/

https://www.digitalocean.com/community/tutorials/how-to-create-and-manage-databases-in-mysql-and-mariadb-on-a-cloud-server

http://guide.debianizzati.org/index.php/Configurare_un_server_Syslog_su_Debian_Squeeze

[11] PIATTAFORMA LAMP E CONFIGURAZIONE APACHE

http://rafaelsteil.com/apache-remote-logging-with-rsyslog/

http://www.kreatore.it/blog/installare-lamp-su-ubuntu-facile-e-veloce

Ringraziamenti

Di persone che sono state a fianco a me in questo cammino sono state tante, alcune ancora presenti altre no. Il primo ringraziamento va alla mia famiglia, pilastro indispensabile nel completamento del mio percorso, mi ha sostenuto, incoraggiato ed aiutato nei momenti più difficili. Devo ringraziare Kindalf che mi ha aperto gli occhi insistendo sulle cose essenziali della vita. Sono stati importanti tutti i suoi consigli di cui ho fatto tesoro. Uno speciale grazie a Valentina, una splendida amica e compagna di studi, di avventure, di disavventure, che negli ultimi anni è rimasta accanto superando numerosi ostacoli e rilevatasi una persona su cui poter contare. Ce ne sono poche così.

Un grazie immenso va a tutti i membri dell'Hacklab di Cosenza che si sono offerti di farmi crescere professionalmente supportandomi e sopportandomi per tutta la stesura della tesi e non solo. Tantissime grazie vanno a Vincenzo che ha creduto in me e, con molta pazienza, mi ha seguito durante tutto il cammino. Un grazie a Stefano che mi ha aiutato in alcuni momenti difficili, a Pietro, Luca e a tutti gli altri per aver collaborato e messo a disposizione l'hardware ed il tempo per questo progetto.

Mille grazie vanno anche al professore G. Ianni che mi ha accettato come tesista nonostante i suoi mille impegni. Grazie per avermi seguito nelle sue materie di insegnamento e di avermi dato fiducia nella scelta della tesi. Un uomo serio, professionale e sempre disponibile che mi ha fatto comprendere che le seconde occasioni non ci saranno sempre. Un grazie particolare va anche al professore Calimeri che mi ha fatto comprendere le difficoltà che incontrerò ed altre già incontrate, mi ha spinto a non sottovalutarmi e a credere nelle mie capacità. Grazie anche alla professoressa Perri, un'insegnate come pochi, dal cuore grande e dalla serietà e comprensione imparagonabili.

Un grazie va anche ai miei amici che, in qualche modo, hanno colorato le mie giornate per affrontare le cose in maniera diversa.

Appendice - RFC 5424 Completo (in italiano)

1. Introduzione

Questo protocollo utilizza un'architettura stratificata per syslog. L'obiettivo di questa architettura è quello di separare il contenuto del messaggio dal trasporto del messaggio abilitando le semplici estensioni per ogni livello.

Questo documento descrive il formato standard per i messaggi syslog e delinea il concetto di transport mappings. Esso descrive, anche, elementi di dato strutturato che possono essere usati per trasmettere, facilmente, informazioni strutturate analizzabili e consentire estensioni del venditore.

Questo documento non descrive tutti i formati di memorizzazione per syslog. Ciò è aldilà dello scopo del protocollo syslog e non è necessario per l'interoperabilità del sistema.

Questo documento è stato scritto con gli originali obiettivi di design per il tradizionale syslog. La necessità di una nuova specifica a livelli è sorta perché gli sforzi di standardizzazione, per affidabili e sicure estensioni syslog, soffrono per la mancanza di uno Standards-Track e un RFC transport-indipendent. Senza questo documento, si dovrebbe definire il proprio formato dei pacchetti e il meccanismo di trasporto di syslog che, nel tempo, introdurrà problemi di compatibilità; esso cerca di fornire una base su cui costruire le estensioni di syslog.

Questo documento cerca di fornire una base su cui costruire le estensioni syslog. Questo approccio ad un'architettura a strati fornisce anche una solida base che consente di scrivere il codice una volta sola per ogni funzione syslog invece che per ogni protocollo di trasporto.

Questo documento rende obsoleto RFC 3164, che è un documento internazionale che descrive alcune implementazioni trovate nel campo.

2. Convenzioni usate in questo documento:

Le parole chiavi “DEVE”, “NON DEVE”, “RICHIESTO”, “DOVREBBE”, “NON DOVREBBE”, “RACCOMANDATO”, “POTREBBE”, e “OPZIONALE”, in questo documento devono essere interpretate come descritto nel RFC 2119.

3. Definizioni:

Syslog utilizza tre livelli:

1. “Syslog content” è la gestione delle informazioni contenute in un messaggio syslog.

2. “Syslog application” che gestisce generazione, interpretazione, routing e archiviazione dei messaggi syslog.

3. “Syslog transport” mette ed estrae i messaggi sul cavo.

Alcuni tipi di funzioni vengono eseguite ad ogni livello concettuale:

- Un “originator” genera contenuti syslog per essere trasportati in un messaggio
- Un “collector” raccoglie contenuti syslog per ulteriori analisi.
- Un “relay” inoltra i messaggi, accetta i messaggi dai mittenti o di altri relay per mandarli ai collectors o altri relay.
- Un “transport sender” passa messaggi syslog ad uno specifico protocollo di trasporto.
- Un “transport receiver” prende i messaggi syslog da uno specifico protocollo di trasporto.

Il Diagramma 1 mostra differenti entità separate per livello.

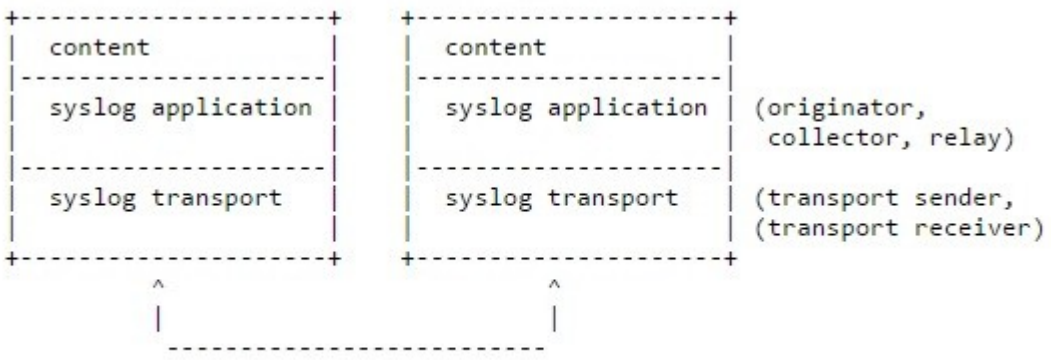


Diagram 1. Syslog Layers

4. Principi base

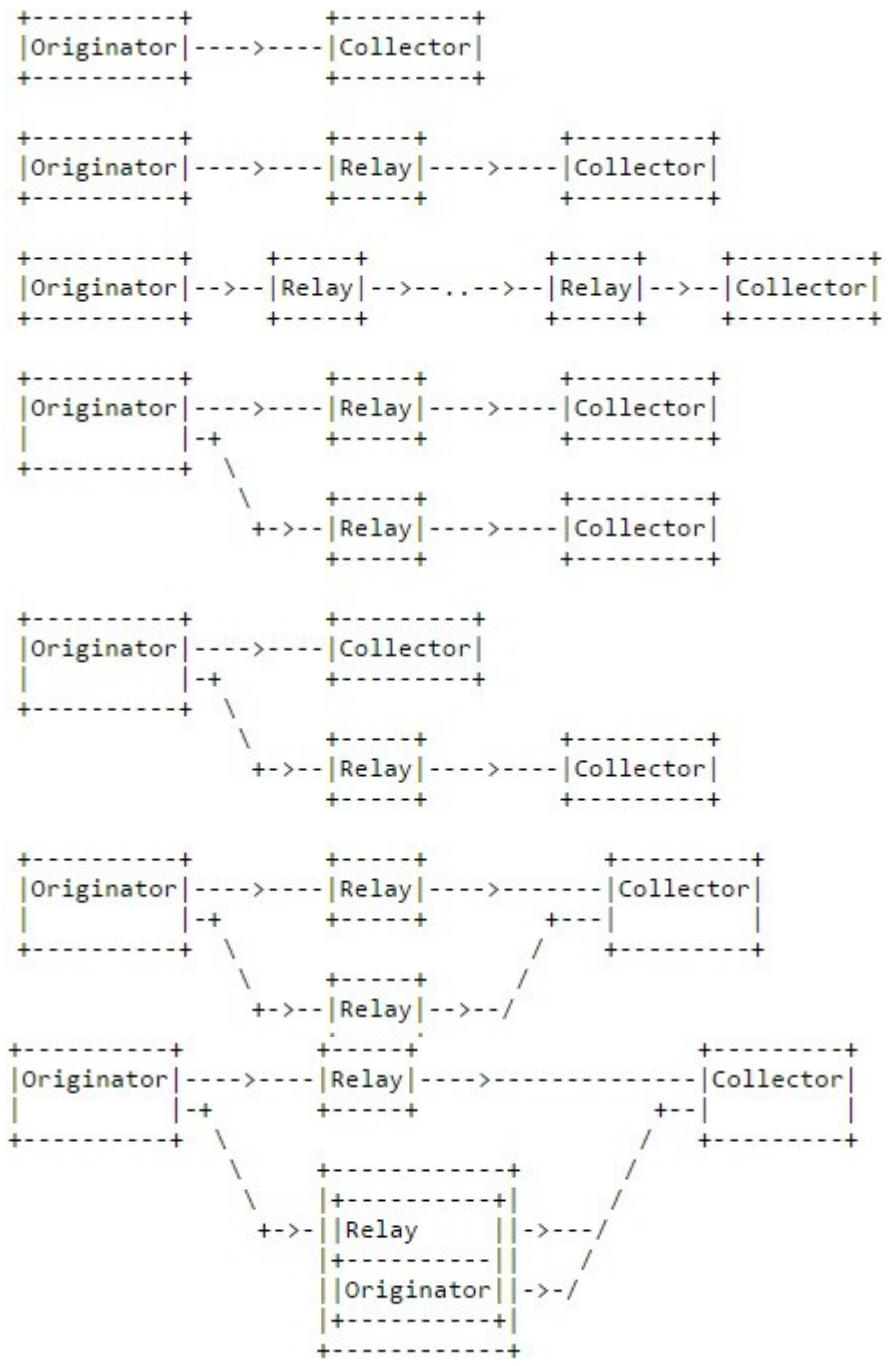
I seguenti principi si applicano nelle comunicazioni syslog:

- Il protocollo syslog non prevede la ricevuta sull'avvenuta consegna del messaggio. Sebbene alcuni protocolli di trasporto potrebbero provvedere lo stato dell'informazione, concettualmente, syslog è un semplice e puro protocollo di comunicazione.

- Gli originator e i relay potrebbero essere configurati per mandare lo stesso messaggio a multipli collectors e relay.
- La funzionalità degli originator, relay e collector potrebbero risiedere sullo stesso sistema.

4.1 Esempi di scenario di distribuzione

Esempi di scenari di distribuzione sono mostrati nel Diagramma 2. Altri arrangiamenti di questi esempio sono accettabili. Come notato, nello schema seguente, i relays potrebbero inviare tutti o alcuni messaggi che ricevono o che hanno generato internamente.
Le caselle rappresentano applicazioni abilitate per syslog.



5. Protocollo di livello di trasporto

Questo documento non specifica alcun protocollo a livello di trasporto ma descrive il formato di un messaggio syslog in maniera indipendente. I protocolli di trasporto syslog sono definiti in altri documenti, un tale protocollo, infatti, è definito nel RFC5426 ed è coerente con il tradizionale protocollo di trasporto UDP. Questo protocollo è necessario per mantenere l'interoperabilità così come UDP è stato, storicamente, usato per la trasmissione di messaggi syslog.

Qualsiasi protocollo di trasporto NON DEVE volutamente alterare il messaggio syslog, se il protocollo dovesse eseguire una temporanea trasformazione al transport sender, queste informazioni DEVONO essere riservate dal protocollo di trasporto al transport receiver in modo tale che il relay o il collector vedranno un'esatta copia del messaggio generato dall'originator o dal relay. In caso contrario i verificatori di crittografia (come la firma) saranno corrotti. Naturalmente l'alterazione del messaggio può verificarsi a causa di errori o altri problemi ma ciò è aldilà dello scopo di questo documento.

5.1 Transport Mapping minimo richiesto

Tutte le implementazioni di questa specifica DEVONO supportare un trasporto basato su TLS come descritto nel RFC5425.

Tutte le implementazioni di questa specifica DOVREBBERO anche supportare un trasporto basato su UDP come descritto nel RFC5426.

E' RACCOMANDATO che gli sviluppi di questa specifica usino un trasporto basato su TLS.

6. Formato Messaggio Syslog

Il messaggio syslog ha la seguente ABNF [RFC5234] definizione:

SYSLOG-MSG = HEADER SP STRUCTURED-DATA [SP MSG]

HEADER = PRI VERSION SP TIMESTAMP SP HOSTNAME
SP APP-NAME SP PROCID SP MSGID

PRI = "<" PRIVAL ">"

PRIVAL = 1*3DIGIT ; range 0 .. 191

```
VERSION        = NONZERO-DIGIT 0*2DIGIT

HOSTNAME       = NILVALUE / 1*255PRINTUSASCII


APP-NAME       = NILVALUE / 1*48PRINTUSASCII

PROCID         = NILVALUE / 1*128PRINTUSASCII

MSGID          = NILVALUE / 1*32PRINTUSASCII


TIMESTAMP      = NILVALUE / FULL-DATE "T" FULL-TIME

FULL-DATE      = DATE-FULLYEAR "-" DATE-MONTH "-" DATE-MDAY

DATE-FULLYEAR  = 4DIGIT

DATE-MONTH     = 2DIGIT  ; 01-12

DATE-MDAY      = 2DIGIT  ; 01-28, 01-29, 01-30, 01-31 based on month/year

FULL-TIME      = PARTIAL-TIME TIME-OFFSET

PARTIAL-TIME   = TIME-HOUR ":" TIME-MINUTE ":" TIME-SECOND [TIME-SECFRAC]

TIME-HOUR      = 2DIGIT  ; 00-23

TIME-MINUTE    = 2DIGIT  ; 00-59

TIME-SECOND    = 2DIGIT  ; 00-59

TIME-SECFRAC   = "." 1*6DIGIT

TIME-OFFSET    = "Z" / TIME-NUMOFFSET

TIME-NUMOFFSET  = ("+" / "-") TIME-HOUR ":" TIME-MINUTE



STRUCTURED-DATA = NILVALUE / 1*SD-ELEMENT

SD-ELEMENT     = "[" SD-ID *(SP SD-PARAM) "]"

SD-PARAM       = PARAM-NAME "=" %d34 PARAM-VALUE %d34
```

SD-ID = SD-NAME

PARAM-NAME = SD-NAME

PARAM-VALUE = UTF-8-STRING ; characters '"', '\'and ']'MUST be escaped.

SD-NAME = 1*32PRINTUSASCII except '=', SP, ']', %d34 (")

MSG = MSG-ANY / MSG-UTF8

MSG-ANY = *OCTET ; not starting with BOM

MSG-UTF8 = BOM UTF-8-STRING

BOM = %xEF.BB.BF

UTF-8-STRING = *OCTET ; UTF-8 string as specified in RFC 3629

OCTET = %d00-255

SP = %d32

PRINTUSASCII = %d33-126

NONZERO-DIGIT = %d49-57

DIGIT = %d48 / NONZERO-DIGIT

NILVALUE = "-"

6.1 Lunghezza del messaggio

I limiti di dimensione dei messaggi Syslog sono dettati dal syslog transport mapping in uso. Non esiste limite superiore. Ogni mappatura definisce il supporto minimo massimo richiesto per la lunghezza del messaggio e il minimo massimo deve essere almeno 480 caratteri in lunghezza.

Qualsiasi transport receiver DEVE essere in grado di accettare i messaggi superiori o minori di 480 ottetti di lunghezza. Tutte le implementazioni del transport receiver possono essere in grado di accettare messaggi superiori, e non, di 2048 ottetti in lunghezza. I transport receivers POTREBBERO

ricevere messaggi più grandi di 2048 ottetti. Se uno di essi dovesse ricevere un messaggio più grande di quello che supporta, DOVREBBE troncare il payload o, in alternativa, POTREBBE scartare il messaggio.

Il troncamento del messaggio DEVE avvenire alla fine dello stesso, dopo di che, il messaggio, potrebbe contenere o un'invalida codifica UTF-8 o un STRUCTURED-DATA non consentito. Il transport receiver POTREBBE scartare il messaggio o POTREBBE provare ad elaborarlo quanto possibile in questo caso.

6.2 HEADER

Il set di caratteri utilizzati nel Header DEVE essere ASCII a sette bit in un campo di otto bit, come descritto nel RFC5234. Questi codici ASCII sono definiti nel "USA Standard Code for Information Interchange".

Il formato del Header è progettato per fornire interoperabilità con il vecchio syslog basato su BSD.

6.2.1 PRI

La parte PRI DEVE avere tre, quattro o cinque caratteri e sarà legata con parentesi angolari come primo e ultimo carattere. La parte PRI inizia con un "<" ('minore di'carattere, %d60) e da un numero che è seguito dal simbolo ">" ('maggiore di'carattere, &d62). Il numero contenuto all'interno di queste parentesi angolari è conosciuto come valore Priorità (PRIVAL) e rappresenta sia la facility che la severity. Il valore Priority consiste in uno,due o tre interi decimali usando i valori da %d48 (per "0") a %d57 (per "9").

I valori di facility e severity non sono normative, ma spesso usati. Essi sono descritti nella tabella successiva a solo scopo puramente informativo. I valori facility DEVONO essere nel range compreso tra 0 e 23.

Numerical Code	Facility
0	kernel messages

1	user-level messages
2	mail system
3	system daemons
4	security/authorization messages
5	messages generated internally by syslogd
6	line printer subsystem
7	network news subsystem
8	UUCP subsystem
9	clock daemon
10	security/authorization messages
11	FTP daemon
12	NTP subsystem
13	log audit
14	log alert
15	clock daemon (note 2)
16	local use 0 (local0)
17	local use 1 (local1)
18	local use 2 (local2)
19	local use 3 (local3)
20	local use 4 (local4)
21	local use 5 (local5)
22	local use 6 (local6)
23	local use 7 (local7)

Ogni messaggio di Priority è anche composto da un decimale che indica il livello di Severity. Questi valori DEVONO essere nel range da 0 a 7:

Numerical Severity

Code	
0	Emergency: system is unusable
1	Alert: action must be taken immediately
2	Critical: critical conditions
3	Error: error conditions
4	Warning: warning conditions
5	Notice: normal but significant condition
6	Informational: informational messages
7	Debug: debug-level messages

Il valore Priority è calcolato moltiplicando per 8 il numero della Facility e quindi aggiungendo il valore numerico della Priority. Per esempio, un messaggio del kernel (Facility=0) con una Severity di Emergency (Severity=0) avrà un valore Priority uguale a 0. Ancora, un messaggio "local use 4" (Facility=20) con una Severity di Notice (Severity=5) avrà un valore Priority di 165. Nel PRI di un messaggio syslog, questi valori dovrebbero essere piazzati tra parentesi angolari: <0> e <165> rispettivamente. Comunque, gli "0" NON DEVONO essere usati.

6.2.2 VERSIONE

Il campo Version indica la versione del protocollo syslog. Il numero di versione DEVE essere incrementato per ogni nuova specifica del protocollo syslog, che cambia qualsiasi parte del formato del header. Le modifiche includono l'aggiunta o la rimozione di campi, o di un cambio di sintassi, o semantica dei campi esistenti. L'RFC5424 utilizza il valore "1" per indicare la versione. I valori VERSION vengono assegnati da IANA tramite il metodo Standards Action come descritto nel RFC5226.

6.2.3 TIMESTAMP

Il campo TIMESTAP è un formato timestamp derivato da RFC3339.

Mentre, però, nel RFC3339 sono consentite multiple sintassi, nel RFC5424 si impongono alcune

restrizioni. Il valore TIMESTAMP deve seguire queste restrizioni:

- I caratteri “T” e “Z” devono essere maiuscole.
- L'utilizzo del carattere “T” è richiesto.
- Non devono essere usati i secondi leap.

Il mittente POTREBBE includere TIME-SECFRAC quando il proprio orologio permette accuratezza e performance. Il “timeQuality” di SD-ID consente al mittente di specificare l'accuratezza e l'affidabilità del timestamp.

Un'applicazione syslog DEVE utilizzare il NILVALUE come TIMESTAMP se non è in grado di ottenere il tempo del sistema.

6.2.3.1 ESEMPI

Esempio 1:

1985-04-12T23: 20: 50.52Z

Questo rappresenta 20 minuti e 50.52 secondi dopo la 23esima ora del 12 Aprile 1985 in UTC.

Esempio 2:

1985-04-12T19:20:50.52-04:00

Questo rappresenta lo stesso tempo dell'esempio 1, ma espresso in US Eastern Standard Time (osservando l'ora legale).

Esempio 3:

2003-10-11T22: 14: 15.003Z

Questo rappresenta l'11 ottobre 2003 alle 10:14.15pm e 3 millisecondi. Il timestamp è in UTC e fornisce la risoluzione in millisecondi. Il creatore può, effettivamente, aver avuto una risoluzione migliore ma, fornendo solo tre cifre per la parte frazionale di un secondo, non è dato saperlo.

Esempio 4:

2003-08-24T05: 14: 15,000003-07: 00

Questo esempio rappresenta il 24 agosto 2003 alle 05:14:15, 3 millisecondi. La risoluzione in microsecondo è indicata dalle cifre aggiuntive in TIME-SECFRAC. Il timestamp indica che l'ora locale è -7 ore da UTC. Questo timestamp potrebbe essere stato creato durante l'ora legale del fuso orario del Pacifico.

Esempio 5:

2003-08-24T05: 14: 15,000000003-07: 00

Questo esempio è quasi lo stesso dell'esempio 4, ma è specificato un TIME-SECFRAC in nanosecondi. Ciò si traduce in un TIME-SECFRAC più lungo delle 6 cifre consentite rendendolo, così, non valido.

6.2.4 HOSTNAME

Il campo HOSTNAME identifica la macchina che, in origine, ha inviato il messaggio syslog.

Il capo HOSTNAME DOVREBBE contenere il nome host e il nome di dominio del mittente nel formato specificato nello STD 13 del RFC1034. Questo formato è chiamato, nel RFC5424, Fully Qualified Domain Name (FQDN).

In pratica, non tutte le applicazioni syslog sono in grado di fornire un nome di dominio. In quanto tale, altri valori POTREBBERO essere anche presenti nel HOSTNAME. Questo documento rende le disposizioni per l'utilizzo di altri valori in tali situazioni. Un'applicazione syslog dovrebbe fornire il valore più specifico prima.

L'ordine di preferenza per i contenuti del campo HOSTNAME è il seguente:

1. FQDN
2. Static IP address
3. hostname
4. Dynamic IP address
5. NILVALUE

Se è usato un indirizzo IPV4, esso DEVE essere nella notazione di ottetti decimali nel STD 13 del RFC1035. Se è usato un indirizzo IPV6, una valida rappresentazione testuale è descritta nel RFC4291 e DEVE essere usata.

Le applicazioni syslog DOVREBBERO, consistentemente, usare lo stesso valore, nel campo HOSTNAME, il più a lungo possibile.

Il NILVALUE DOVREBBE essere usato, solo, quando l'applicazione syslog non ha modo di ottenere il proprio reale hostname. Questa situazione è considerata altamente improbabile.

6.2.5 APP-NAME

Il campo APP-NAME DOVREBBE identificare il dispositivo, o l'applicazione, che ha originato il messaggio. Si tratta di una stringa senza ulteriore semantica ed è utile a filtrare i messaggi su un relay o collector.

Il NILVALUE POTREBBE essere usato quando l'applicazione syslog non ha idea del proprio APP-NAME o non è in grado di fornire tale informazione. Questo può succedere quando un dispositivo non è abilitato a fornire tale informazione oppure a causa di una politica locale o perché, semplicemente, l'informazione non è disponibile, o non applicabile, sul device.

Questo campo POTREBBE essere assegnato da un operatore.

6.2.5 PROCID

PROCID è un valore che è incluso nel messaggio, non ha significato interoperabile, tranne quando un cambiamento nel valore indica che c'è stata una discontinuità nel reporting di syslog. Il campo PROCID non ha nessuna sintassi o semantica specifica, il valore è dipendente dall'implementazione e/o assegnato da un operatore. Il NILVALUE deve essere usato quando nessun valore è fornito.

Il campo PROCID è spesso usato per procurare il nome del processo, o il suo ID, associato ad un sistema di syslog. Il NILVALUE viene usato quando l'ID di un processo non è disponibile. Su un sistema embedded, senza l'ID del processi del sistema operativo, PROCID potrebbe essere un valido strumento.

PROCID può consentire all'analizzatore di log di scovare discontinuità nel reporting di syslog individuando un cambiamento nell'ID del processo di syslog, ma non è un'identificazione affidabile di un processo riavviato in quanto, ad un processo syslog riavviato, potrebbe venir assegnato lo stesso ID del precedente processo.

PROCID può anche essere usato per identificare quali messaggi appartengono ad un gruppo di messaggi. Ad esempio, un agente SMTP Mail Transfer potrebbe inserire il proprio ID di transazione SMTP all'interno di PROCID, che consentirebbe, il collector o il relay, di raggruppare i messaggi basati sulla transazione SMTP.

6.2.7 MSGID

Il MSGID identifica il tipo di messaggio. Per esempio, un firewall potrebbe usare il MSGIG "TCPIN" per traffico TCP entrante e il MSGID "TCPOUT" per il traffico TCP uscente. I messaggi con lo stesso MSGID dovrebbero riflettere eventi la stessa semantica. Il MSGID stesso è una stringa senza ulteriore semantica ed è usato per il filtraggio di messaggi sul relay o collector.

Il NILVALUE DOVREBBE ESSERE usato quando l'applicazione syslog non riesce, o non può, fornire nessun valore.

Questo campo POTREBBE essere assegnato da un operatore.

6.3 STRUCTURED-DATA

STRUCTURED-DATA fornisce un meccanismo per esprimere informazioni in un formato ben definito, facile da interpretare ed interpretabile. Ci sono multipli scenari di utilizzo. Per esempio, potrebbe esprimere meta-informazioni riguardo un messaggio syslog o informazioni come contatori di traffico o indirizzi IP.

STRUCTURED-DATA può contare zero, uno o multipli elementi di dato strutturato, che sono riferiti come "SD-ELEMENT".

In caso di zero elementi, il campo STRUCTURED-DATA DEVE contenere il NILVALUE.

Il set di caratteri usato in STRUCTURED-DATA DEVE essere ASCII a 7 bit in un campo di otto bit come scritto nel RFC5234. Questi sono codici ASCII definiti nel "USA Standard Code for Information Interchange" (ANSI.X3-4.1968). Un'eccezione è il campo PARAM-VALUE in cui viene usata la codifica UTF-8.

Un collector POTREBBE anche ignorare elementi malformati di STRUCTURED-DATA. Un relay DEVE, invece, trasmette malformati STRUCTURED-DATA senza nessuna alterazione.

6.3.1 SD-ELEMENT

Un SD-ELEMENT consiste in un nome ed una coppia nome-valore. Il nome viene indicato come SD-ID, le coppie nome-valore, invece, come “SD-PARAM”.

6.3.2 SD-ID

Gli SD-ID sono case sensitive e, unicamente, identificano il tipo e lo scopo del SD-ELEMENT. Lo stesso SD-ID NON DEVE comparire più di una volta in un messaggio.

Ci sono due formati per i nomi SD-ID:

1. Nomi che non contengono un “at” (“@”, ABNF %d64) sono riservati per essere assegnati da IETF Review come descritto nel BCP26 del RFC5226. I nomi di questo formato sono validi solo se sono prima registrati con lo IANA. Tali nomi NON DEVONO contenere un “at” , un uguale (“=”, ABNF %d61), una parentesi quadra di chiusura (“]”, ABNF %d93), caratteri di citazione ("“ ', ABNF %d34), spazi vuoti, o caratteri di controllo (codice ASCII 127 e codici 32 o minori).

2. Chiunque può definire ulteriori SD-ID utilizzando nomi nel formato name@<private enterprise number>, ad esempio “ourSDID@32473”. Il formato della parte che precede il segno 'at' non è specificato; comunque, questi nomi DEVONO essere stringhe US-ASCII stampabili e non devono contenere il simbolo 'at', il simbolo 'uguale', una parentesi quadra di chiusura, il carattere di citazione, spazi bianchi o caratteri di controllo. La parte che segue l'“at” DEVE essere il numero privato dell'impresa. Si noti che il numero 32473 è stato riservato da IANA per essere utilizzato nel RFC5424 a titolo di esempio. Chiunque, dovrà utilizzare il proprio numero privato nel parametro “enterpriceId” e durante la creazione di nomi SD-ID localmente estensibili.

6.3.3 SD-PARAM

Ogni SD-PARAM consiste in un nome, denominato PARAM-NAME, e di un valore, chiamato PARAM-VALUE.

Param-name è case-sensitive e lo IANA li controlla tutti ad eccezione di quei SD-ID i cui nomi

contengono il simbolo “at”. L'importanza del PARAM-NAME è all'interno di uno specifico SD-ID. Quindi, due valori uguali di PARAM-NAME in due differenti SD-ID non saranno gli stessi.

Per supportare i caratteri internazionali, il campo PARAM-VALUE DEVE essere codificato utilizzando UTF-8. Un'applicazione syslog POTREBBE inserire una qualsiasi sequenza valida UTF-8 e DEVE accettare qualsiasi sequenza valida nel formato “shortest form” e NON DEVE fallire se i caratteri di controllo fossero presenti nel PARAM-VALUE. L'applicazione POTREBBE modificare i messaggi syslog contenenti caratteri di controllo (ad esempio cambiando un ottetto con il valore 0 (USACII NUL) in quattro caratteri “#000”). Per la ragione sottolineata in UNICODE TR36, sezione 3.1, un originator DEVE codificare i messaggi nella “shortest form” e un collector o rlay NON DEVONO interpretare i messaggi nella “non-shortest form”.

All'interno del PARAM-VALUE, i caratteri '“ '(ABNF %d34), '\ '(ABNF %d92), e '] '(ABNF %d93) DEVONO essere evitati. Questo è necessario per non avere errori di analisi. Non è strettamente necessario non usare ']'ma è RICHIESTO da questa specifica per sfuggire in errori di implementazione di un'applicazione syslog. Ognuno di questi tre caratteri deve essere evitato: '\ ', '\\ ', e '\] 'rispettivamente.

Un backslash ('\ ') seguito da nessuno dei tre caratteri descritti è considerato una sequenza di escape non valida. In questo caso, il backslash, DEVE essere trattato come una regolare barra rovesciata e come un regolare carattere. Quindi, la sequenza invalida NON DEVE essere alterata.

Un SD-PARAM potrebbe essere ripetuto multiple volte all'interno di un SD-ELEMENT.

6.3.4 Change Control

Una volta che gli SD-ID e i PARAM-NAME sono definiti, sintassi e semantiche di questo oggetti NON DEVONO essere alterate. In caso di variazione di un oggetto già esistente, un nuovo SD-ID o PARAM-NAME DEVE essere creato e il vecchio rimanere invariato. Dei PARAM-NAME OPZIONALI POTREBBERO essere aggiunti ad un SD-ID esistente.

6.3.5 Esempi

Tutti gli esempi in questa sezione mostrano solo la parte di dati strutturati di un messaggio. Gli esempi dovrebbero essere considerati su una sola riga. Alcuni di essi sono stati scritti su due righe per migliorarne la leggibilità.

Esempio 1 – Valido

[exampleSDID@32473 iut="3" eventSource="Application" eventID="1011"]

Questo esempio è un elemento di dato strutturato con non-IANA SD-ID controllato di tipo "exampleSDID@32473", che a tre parametri.

Esempio 2 – Valido

[exampleSDID@32473 iut="3" eventSource="Application" eventID="1011"]
[examplePriority@32473 class="high"]

Questo è lo stesso esempio del primo ma con un secondo elemento di dato strutturato. Bisogna notare che questo elemento segue, immediatamente, il primo (non ci sono spazi tra loro).

Esempio 3 – Non Valido

[exampleSDID@32473 iut="3" eventSource="Application"
eventID="1011"] [examplePriority@32473 class="high"]

Questo esempio somiglia all'esempio 2, ma ha un errore: vi è un carattere SP tra i due elementi di dato strutturato ("] SP ["). Ciò non è valido e farà si che il campo STRUCTURED-DATA termini dopo il primo elemento. Il secondo elemento sarà interpretato come parte del campo MSG.

Esempio 4 – Non Valido

[exampleSDID@32473 iut="3" eventSource="Application"

eventID="1011"][examplePriority@32473 class="high"]

L'esempio è simile al 2, anche qui c'è un errore: il carattere SP è presente dopo la parentesi iniziale. L'elemente SD-ID di un dato strutturato deve, immediatamente, seguire la parentesi iniziale, il carattere spazio invalida il STRUCTURED-DATA e un'applicazione syslog scarterà questo messaggio.

Esempio 5 – Valido

[sigSig ver="1" rsID="1234" ... signature="…"]

L'esempio 5 mostra un ipotetico SD-ID assegnato da IANA. Le ellissi indicano il contenuto mancante, che è stato lasciato fuori da questo esempio per renderlo più breve.

6.4 MSG

La parte MSG contiene un messaggio in formato libero che fornisce informazioni riguardo l'evento.

Il set di caratteri utilizzato in MSG DOVREBBE essere in UNICODE, codificato in UTF-8 come specificato nel RFC3629. Se l'applicazione syslog non potesse codificare MSG in Unicode, è possibile utilizzare qualsiasi altra codifica.

L'applicazione syslog DOVREBBE evitare valori di ottetto sotto i 32 (il tradizionale range di caratteri di controllo US-ASCII eccetto DEL). Questi valori sono legali, ma un'applicazione syslog POTREBBE modificarli alla ricezione. Ad esempio potrebbe cambiarli in una sequenza di escape (es. il valore 0 può essere cambiato in “\ 0”). L'applicazione NON DOVREBBE modificare qualsiasi altro valore.

Se un'applicazione syslog codifica MSG in UTF-8, la stringa DEVE iniziare con la Unicode byte order mask (BOM), che per UTF-8 è ABNF %xEF.BB.BF. L'applicazione deve codificare nella "shortest form" e usare qualsiasi sequenza valida UTF-8. Se stesse processando un MSG iniziando con un BOM e il MSG contenesse UTF-8 the non è in forma breve, il MSG NON DEVE essere interpretato come codificato in UTF-8 per le ragioni esposte in UNICODE-TR36, sezione 3.1, inoltre, un'applicazione syslog non deve interpretare i messaggi nel "non-shortest form". Essa NON DEVE interpretare un'invalida sequenza UTF-8.

6.5 Esempi

I seguenti sono esempi di messaggi syslog validi. Una descrizione di ogni esempio è sotto di essi. Gli esempi sono basati su esempi simili dal RFC3164. L'Unicode BOM è rappresentato come "BOM" negli esempi.

Esempio 1 – senza STRUCTURED-DATA

<34>1 2003-10-11T22:14:15.003Z mymachine.example.com su - ID47

- BOM'su root'failed for lonvick on /dev/pts/8

In questo esempio, la VERSIONE è 1, Facility ha il valore di 4 e Severity di 2. Il messaggio è stato creato il giorno 11 Ottobre 2003 alle ore 10:14:15pc UTC, 3 millisecondi. Il messaggio è originato da un host che si identifica come "mymachine.example.com". L'APP-NAME è "su" e il PROCID è unknown, il MSGID è "ID47" ed il MSG è "su root" non riuscito per lonvick…", codificato in UTF-8. La codifica è definita da BOM e non c'è STRUCTURED-DATA presente nel messaggio; questo è indicato dal "-" nel campo STRUCTURED-DATA.

Esempio 2 – Senza STRUCTURED-DATA

<165>1 2003-08-24T05:14:15.000003-07:00 192.0.2.1

myproc 8710 - - %% It's time to make the do-nuts.

In questo esempio, la versione è di nuovo 1. La Facility è 20, la Severity è 5, il messaggio è stato creato il 24 Agosto 2003 alle 5:14:15am, con un -7 offset da UTC, 3 microsecondi. Il nome host è “192.0.2.1”, per cui l'applicazione syslog non sapeva il suo FQDN (Fully Qualified Domain Name = nome di dominio) e usato uno dei suoi indirizzi IPv4. L'APP-NAME è “myproc” e la PROCID è “8710” (per esempio, questo potrebbe essere il UNIX PID). Non vi è alcun dato strutturato presente nel messaggio; questo è indicato con “-” nel campo STRUCTURED-DATA. Non c'è uno specifico MSGID e questo è indicato dal “-” nel campo MSGID.

Il messaggio è “%% It's time to make the do-nuts.”. Manca l'Unicode BOM quindi l'applicazione non conosce la codifica della parte MSG.

Esempio 3 – con il dato strutturato

<165>1 2003-10-11T22:14:15.003Z mymachine.example.com

evntslog - ID47 [exampleSDID@32473 iut="3" eventSource=

"Application" eventID="1011"] BOMAn application

event log entry…

Questo esempio è modellato dall'esempio 1. Comunque, questa volta contiene il dato strutturato, un singolo elemento con il valore “[exampleSDID@32473 iut=”3” eventSource=”Application”eventID=”1011]”. Il MSG stesso è “An application event log entry...” Il BOM all'inizio del MSG indica la codifica UTF-8.

Esempio 4 – Solo STRUCTURED-DATA

<165>1 2003-10-11T22:14:15.003Z mymachine.example.com

evntslog - ID47 [exampleSDID@32473 iut="3" eventSource=

"Application" eventID="1011"][examplePriority@32473

class="high"]

Questo esempio mostra un messaggio con solo il dato strutturato e nessuna parte MSG. Questo è un valido messaggio.

7. Structured Data ID

Questa sezione definisce gli iniziali SD-ID registrati da IANA. Tutti gli SD-ID definiti qui sono opzionali.

In seguito, una lunghezza massima è quantificata per i valori di parametro, in ognuno di questi casi, l'applicazione syslog deve essere preparata a ricevere il numero di caratteri definito in qualsiasi punto di codice valido UTF-8. Poiché ciascun carattere può essere fino a 6 ottetti, è RACCOMANDATO che ogni applicazione sia pronta a ricevere fino a 6 ottetti per carattere.

7.1 timeQuality

Il SD-ID “timeQuality” DOVREBBE essere usato dall'originator per descrivere la sua nozione di tempo del sistema. Questo SD-ID deve essere scritto se l'originator non è propriamente sincronizzato con un tempo sorgente esterno affidabile o se non sa se le proprie informazioni sul fuso orario siano corrette. L'uso principale di questo elemento di dato strutturato e fornire alcune informazioni sul livello di fiducia che ha il TIMESTAMP. Tutti i parametri sono opzionali.

7.1.1 tzKnown

Il parametro “tzKnown” indica se l'originator sa il proprio fuso orario. Se cosi non fosse, DEVE essere usato il valore “1”. Se, invece, le informazioni fossero in dubbio, il valore usato DEVE essere “0”. Se l'originator conoscesse il proprio fuso orario ma decidesse di emettere il tempo in UTC, DEVE essere usato il valore “1”(perché il fuso orario è conosciuto).

7.1.2 isSynced

Il parametro “isSynced” indica se l'originator è sincronizzato con un tempo sorgente esterno affidabile, per esempio via NTP (Network Time Protocol). Se tale tempo fosse sincronizzato, il valore usato DEVE essere “1” altrimenti “0”.

7.1.3 syncAccuracy

Il parametro “syncAccuracy” indica quanto l'originator pensa sia precisa la sua sincronizzazione dell'ora. Esso è un intero che descrive il numero massimo di microsecondi in cui l'orologio potrebbe essere spento tra intervalli di sincronizzazione.

Se fosse usato il valore “0” per “isSynced”, questo parametro NON DEVE essere specificato, se il valore fosse “1” ma fosse assente il parametro “syncAccuracy”, un collector o relay possono assumere che le informazioni del tempo fornito siano accurate abbastanza da essere considerate corrette. Il parametro “syncAccuracy” DEVE essere scritto solo se, attualmente, l'originator avesse la conoscenza dell'affidabilità del tempo della sorgente esterna. Nella maggior parte dei casi, si avrà questa conoscenza attraverso la configurazione dell'operator.

7.1.4 Esempi

Quanto segue è un esempio di un originator che non conosce il proprio fuso orario o se venisse sincronizzato:

[timeQuality tzKnown="0" isSynced="0"]

Con questa informazione, l'originator indica che le proprie informazioni sul tempo non è affidabile. Questo può essere un suggerimento per il collector o relay ad usare la propria ora locale invece del TIMESTAMP per la correlazione di più messaggi provenienti da diversi mittenti.

Quanto segue è un esempio di un originator che conosce il suo fuso orario e sa di essere correttamente sincronizzato con una fonte esterna affidabile:

[timeQuality tzKnown = "1" isSynced = "1"]

Quanto segue è un esempio di un originator che conosce sia il suo fuso orario che di essere sincronizzato esternamente. Conosce anche quanto sia accurata la sincronizzazione esterna:

[timeQuality tzKnown="1" isSynced="1" syncAccuracy="60000000"]

La differenza tra questo e l'esempio precedente è che l'originator aspetta che il suo orologio sia mantenuto entro 60 secondi dal tempo ufficiale. Quindi, se l'originator dovesse riportare 9:00:00, esso sarà non prima delle 8:59:00 e non oltre le 09:01:00.

7.2 Origin

L'SD-ID “origin” POTREBBE essere utilizzato per indicare l'origine del messaggio syslog. I seguenti parametri possono essere usati e sono tutti OPZIONALI.

Specificando uno di questi parametri è, soprattutto, un aiuto per accedere agli analizzatori di log e applicazioni simili.

7.2.1 ip

Il parametro “ip” denota gli indirizzi IP che l'originator sa di avere al momento di originare il messaggio e DEVE contenere la rappresentazione testuale di un indirizzo ip.

Questo parametro può essere usato per fornire l'informazione di identificazione in aggiunta a ciò che è presente nel campo HOSTNAME e potrebbe essere utile se l'indirizzo IP dell'host è incluso nel messaggio, mentre il campo HOSTNAME contiene ancora il nome di dominio. E'anche utile per descrivere tutti gli indirizzi IP di un host multihomed.

Se un originator ha multipli indirizzi IP, può mostrarne uno nel parametro “ip” oppure include più parametri “ip” in un unico elemento di dato strutturato “origin”.

7.2.2 enterpriseId

Il parametro “enterpriseId” DEVE essere un 'SMI Network Management Private Enterprise Code', mantenuto da IANA, il cui prefisso è iso.org.dod.internet.private.enterprise (1.3.6.1.4.1). Il numero che segue DEVE essere unico e registrato con IANA secondo l'RFC2578. Un'impresa è autorizzata solo ad assegnare valori all'interno della sottostruttura iso.org.dod.internet.private.enterprise.<private enterprise number>. In generale, è necessario solamente il numero privato enterprise assegnato da IANA (un singolo numero). Un'impresa potrebbe decidere di usare degli identificatori al di sotto del proprio numero privato. Se ciò avvenisse, questi sub-identificatori DEVONO essere separati da periodi e rappresentati come numeri decimali. Per esempio “32473.1.2”. La completa lista up-to.date di Private Enterprise Numers (PEN) è mantenuto da IANA.

Specificando un numero enterprise privato, il venditore permette più elaborazioni specifiche del messaggio.

7.2.3 software

Il parametro “software” identifica univocamente il software che ha generato il messaggio. Se venisse usato, DOVREBBE essere specificato solo “enterpriseId”, così che uno specifico software di un venditore possa essere identificato. Il parametro “software” non è lo stesso del campo APP-NAME, esso contiene sempre il nome del software generatore, considerando che APP-NAME può contenere qualsiasi altra cosa, tra cui un valore configurato dall'operatore.

Il parametro “software” è una stringa e non deve essere più lunga di 48 caratteri.

7.2.4 swVersion

Il parametro “swVersione” identifica unicamente la versione del software che ha generato il messaggio. Se usato, i parametri “software” e “enterpriseId” DOVREBBERO anche essere forniti.

Il parametro “swVersione” è una stringa e non può essere più lunga di 32 caratteri.

7.2.5 Esempi

Il seguente esempio contiene multipli indirizzi IP:

[origin ip="192.0.2.1" ip="192.0.2.129"]

In questo esempio l'originator indica che ha usato due indirizzi IP, uno è 192.0.2.1 e l'altro è 192.10.2.129.

7.3 meta

L'SD-ID “meta” POTREBBE essere usato per fornire meta-informazioni riguardo il messaggio. Possono essere usati i successivi parametri opzionali. Se “meta” venisse usato, solo un parametro DOVREBBE essere specificato.

7.3.1 sequenceId

Il parametro “sequenceId” traccia la sequenza in cui l'originator manda i messaggi al transport syslog per essere inviati. Esso è un intero che deve essere settato a 1 quando la funzione syslog è iniziata e deve essere aumentato ad ogni messaggio fino ad un massimo valore di 2147483647. Se venisse raggiunto tale valore, il prossimo messaggio dovrà essere inviato con un sequenceId di 1.

7.3.2 sysUpTime

Il parametro “sysUpTime”POTREBBE essere usato per include il parametro “sysUpTime” di SNMP nel messaggio. La sua sintassi e semantiche sono definite nel RFC3418.

Dato che syslog non supporta direttamente la sintassi “INTEGER” di SNMP, il valore deve essere rappresentato come un intero decimale usando solo i caratteri “0”, “1”, “2”, “3”, “4”, “5”, “6”, “7”, “8” e “9”.

7.3.3 language

Il parametro “language” POTREBBE essere specificato dall'originator per trasmettere informazioni riguardo il naturale linguaggio usato dentro MSG. Se tale parametro dovesse essere specificato, DEVE contenere un linguaggio identificativo come definito in BCP47 RFC4646.

8. Considerazioni di sicurezza

8.1 UNICODE

Questo documento usa la codifica UTF-8 per i campi PARAM-VALUE e MSG. Ci sono un numero di problemi di sicurezza con UNICODE. Qualsiasi implementer e operator è consigliabile veda l'UNICODE TR36 (UTR36) per imparare su questi temi. Questo documento mettere in guardia contro i problemi tecnici delineati nel UTR36 richiedendo la codifica “shortest form” per le applicazioni syslog. Comunque, lo spoofing visivo dovuto alla confusione del carattere persiste ancora. Questo documento cerca di ridurre, al minimo, gli effetti di spoofing visivo consentendo solo UNICODE laddove sia previsto e necessario uno script locale. In tutti gli altri campi, è richiesto US-ASCII. Inoltre, i campi PARAM-VALUE e MSG non dovrebbero essere la fonte primaria di informazioni di identificazione riducendo, ulteriormente, i rischi associati allo spoofing visivo.

8.2 Caratteri di controllo

Questo documento non impone alcuna limitazione sul contenuto di MSG o PARAM-VALUE quindi, essi possono contenere caratteri di controllo, tra cui il carattere NUL.

In alcuni linguaggi di programmazione (i più noti C e C++), il carattere NUL (ABNF %d00) tradizionalmente ha uno speciale significato di terminatore di stringa. La maggioranza delle implementazioni di questi linguaggi assume che una stringa non sarà estesa oltre il primo carattere NUL. Questo è, primariamente, una restrizione del supporto di librerie run-time. Quindi, il carattere NUL deve essere considerato con grande attenzione ed essere correttamente gestito. Un utente malintenzionato può deliberatamente includere il carattere NUL per nascondere informazioni successive. La cattiva gestione di NUL potrebbe anche invalidare i controlli crittografici che sono trasmetti all'interno del messaggio.

Molti popolari editor di testo sono anche scritti in linguaggi con questa restrizione. E'consigliabile la codifica che caratteri NUL durante la scrittura di un file di testo. Se fossero memorizzati senza la codifica, il file può diventare illeggibile.

Altri caratteri di controllo potrebbero, anche, essere problematici. Per esempio, un utente malintenzionato potrebbe includere caratteri di backspace per rendere parti del messaggio di log illeggibili. Esistono problemi simili per quasi tutti i caratteri di controllo.

In fine, una sequenza non valida di UTF-8 potrebbe essere usato da un utente malintenzionato per inserire caratteri di controllo ASCII.

Questa specificazione permette, ad un'applicazione syslog, di riformattare i carattere di controllo ricevuti. I rischi di sicurezza associati ai caratteri di controllo sono stati un'importante forza guida alla base di questa importante restrizione. Gli originators sono avvisati che se venisse utilizzata qualsiasi codifica diversa ASCII e UTF-8, il receiver potrebbe corrompere il messaggio nel tentare di filtrare i carattere di controllo ASCII.

8.3 Troncamento del messaggio

Il messaggio di troncamento può essere utilizzato da un utente malintenzionato per nascondere un log vitale di informazioni. I messaggi oltre la dimensione minima supportata possono essere scartati o troncati dal transport receiver per cui, alcune informazioni di log vitali potrebbe venire perse.

Al fine di prevenire la perdita di informazioni, i messaggi devono essere più lunghi rispetto le dimensioni minime massime richieste. Per una migliore performance e affidabilità, i messaggi dovrebbero essere più piccoli possibile.

Un'importante informazione potrebbe essere posta all'inizio del messaggio in quanto è meno probabile che vengano scartati dal transport receiver.

Un originator dovrebbe limitare la dimensione di tutti i dati forniti dall'utente all'interno di un messaggio syslog. In caso contrario, un utente malintenzionato potrebbe immettere dati di grandi dimensioni nella speranza di sfruttare una potenziale debolezza.

8.4 Replay

Non ci sono meccanismi nel protocollo syslog per rilevare la replay di un messaggio. Un utente malintenzionato potrebbe registrare una serie di messaggi che indicano una normale attività di una macchina. In un secondo momento, l'attaccante, può rimuovere quella macchina dalla rete e riprodurre i messaggi syslog al relay o collector. Anche con il campo TIMESTAMP nella parte HEADER, un attaccante potrebbe registrare i pacchetti e semplicemente modificarli per riflettere l'ora corrente prima di ritrasmetterli.

Gli amministratori potrebbe trovare nulla di inusuale nei messaggi ricevuti e la loro ricezione dovrebbe, falsamente, indicare una normale attività della macchina.

Segnando in maniera crittografica i messaggio, dovrebbe prevenire l'alterazione dei TIMESTAMP e quindi l'attacco di replay.

8.5 Consegna affidabile

Poiché non c'è nessun meccanismo descritto in questo documento che garantisce la consegna e il trasporto (es. UDP), alcuni messaggio potrebbero venire persi. Quest'ultimi potrebbe essere o eliminati attraverso la congestione della rete, o possono essere intercettati maliziosamente e scartati. Le conseguenze dell'eliminazione di uno o più messaggi syslog non può essere determinata. Se i messaggi fossero semplici aggiornamenti di stato, allora la mancata ricezione potrebbe non essere notata o causare un fastidio agli operatori del sistema. D'altra parte, se i messaggi fossero più critici, gli amministratori non saranno consapevoli di un grave problema sviluppatosi. I messaggi potrebbero anche essere intercettati e scartati da un attaccante nascondendo attività non autorizzate.

Può anche essere opportuno includere caratteristiche rate-limiting negli originator e relay. Questo può ridurre potenziali problemi di congestione quando si ha un grande ammontare di messaggi.

La consegna affidabile potrebbe non essere sempre desiderabile. Avere tale funzione significa che l'originator o il relay syslog deve bloccare quando il relay o il collector non è in grado di accettare ulteriori messaggi. In alcuni sistemi operativi, ovvero Unix/Linux, l'originator o il relay syslog gira all'interno di un processo di sistema ad alta priorità (syslogd). Se questo processo venisse bloccato, il sistema nel suo complesso avrà un arresto. Lo stesso si verifica se vi è una situazione di deadlock tra syslogd e, ad esempio, il server DNS.

Per evitare questi problemi, la consegna affidabile può essere implementata in modo tale da scartare, intenzionalmente, i messaggi quando l'applicazione syslog, altrimenti, lo blocca. Il vantaggio di tale funzione, in questo caso, è che l'originator o relay syslog scarta il messaggio ed è in grado di informare il relay o collector di questo fatto. Così il relay o collector riceve le informazioni che qualcosa è andato perso. Senza la consegna affidabile, il messaggio potrebbe essere perso senza qualsiasi indicazione dell'avvenuta perdita.

8.6 Controllo della congestione

Considerato che syslog può generare un illimitato ammontare di dati, trasferirli tutti su UPD è problematico in quanto in UDP manca il meccanismo di controllo della congestione. I meccanismi di controllo della congestione, che rispondono riducendo l'ammontare di traffico e stabilire un grado di equità tra i flussi che condividono lo stesso percorso, sono di vitale importanza per il funzionamento stabile di Internet [RFC2914]. Questo è il motivo per cui il trasporto syslog TLS sia RICHIESTO da implementare e RACCOMANDATO per l'uso generale.

Solo gli ambienti dove il trasporto UDP POTREBBE essere utilizzato come alternativa a TLS sono reti gestite, dove il percorso di rete è stato esplicitamente previsto per il traffico syslog UDP attraverso meccanismi di ingegneria del traffico, come limitare il tasso o la capacità di prenotazioni. In tutti gli altri ambienti, DOVREBBE essere usato TLS.

In qualsiasi implementazione, può verificarsi la situazione in cui un originator o relay avrebbe bisogno di bloccare l'invio di messaggi. Un caso comune è quando una coda interna è piena. Questo potrebbe succedere a causa del rate-limiting o basse performance dell'applicazione syslog. In qualche evento, è altamente RACCOMANDATO che nessun messaggio sia eliminato ma che vengano temporaneamente memorizzati fino alla loro trasmissione. Comunque, se essi devono essere eliminati, è RACCOMANDATO che l'originator o relay elimini i messaggi con bassa severity in favore dei messaggi con severity maggiore.

I messaggi con un più basso valore di SEVERITY hanno una maggiore gravità rispetto a quelli con un valore numero più elevato. In tale situazione, i messaggi DOVREBBERO essere scartati. L'applicazione syslog può notificare un collector o relay riguardo tale scarto.

8.7 Integrità del messaggio

Oltre ad essere scartati, i messaggi syslog possono essere danneggiati durante il trasporto, o un utente malintenzionato potrebbe modificarli. In tali casi, il contenuto originale del messaggio non sarà consegnato al collector o relay. In aggiunta, se un malintenzionato fosse posizionato tra il transport sender e il transport receiver dei messaggi syslog, sarà possibile intercettare e modificare questi messaggi mentre sono in transito per nascondere attività non autorizzate.

8.8 Osservare i messaggio

Mentre non ci sono linee guida relative al formato MSG, la maggior parte dei messaggi syslog sono generati in forma leggibile con l'ipotesi che gli amministratori capaci dovrebbero essere in grado di leggerli e capirne il significato. Il protocollo syslog non ha meccanismi per garantire la riservatezza dei messaggi in transito. Nella maggior parte dei casi, scambiare messaggi in chiaro è un vantaggio per il personale operativo se essi stanno "sniffando" pacchetti dal cavo. Gli operatori potrebbero leggere i messaggi e associarli con altri eventi visti da altri pacchetti che attraversano il cavo per tenere traccia dei problemi e correggerli. Purtroppo, un malintenzionato può anche essere in gradi di osservare i contenuti leggibili dei messaggi syslog. Egli può quindi utilizzare le conoscenze acquisite da quei messaggi per compromettere una macchina o fare altri danni.

Gli operatori sono invitati a utilizzare una mappatura dei trasporti sicura per evitare questo problema.

8.10 Forwarding Loop

Come mostrato nel Diagramma 2, le macchine potrebbero essere configurate per l'inoltro dei messaggi syslog al successivo relay prima di raggiungere un collector. In un caso particolare, un amministratore ha scoperto di avere, erroneamente, configurato due relay per inoltrare con un certo valore di SEVERITY verso l'altro. Quando una di queste macchine ha ricevuto o generato quel tipo di messaggio, lo avrà trasmesso agli altri relay. Quel relay, a sua volta, lo manderà indietro. Questo ciclo ha causato la degradazione della rete nonché alla disponibilità di elaborazione dei due dispositivi. Gli amministratori di rete devono fare attenzione a non causare tale spirale della morte.

8.11 Load Considerations

Gli amministratori di rete devono prendersi il tempo per stimare l'appropriata capacità del collector syslog. Un attaccante potrebbe effettuare un attacco Denial of Service riempiendo il disco del collector con messaggi falsi. Ponendo i registri in un file circolare potrebbe alleviare questo ma ha la conseguenza di non che garantire la ricezione di futuri registri. Lungo questa linea, un transport receiver deve avere un'interfaccia di rete capace di ricevere i messaggi inviati ad esso.

Gli amministratori e progettisti di rete devono criticamente rivedere i percorsi di rete tra gli originator, relay e collector. I messaggi syslog generati non dovrebbero sopraffare qualunque collegamento di rete.

Per ridurre l'impatto a questo problema è raccomandato usare trasporti con consegna garantita.

8.12 Denial of Service

Come con qualsiasi sistema, un utente malintenzionato può solo sopraffare un transport receiver mandando più messaggi di quanto l'infrastruttura o il dispositivo stesso possa gestire. I responsabili dell'implementazione dovrebbero cercare di fornire le caratteristiche che riducono al minimo questa minaccia, come accettare solo messaggi syslog da indirizzi IP noti.

9. Considerazioni IANA

9.1 VERSIONE

IANA ha creato un registro intitolato “syslog Version Values” dei valori di versione. I numeri di versione DEVONO essere incrementati per tutte le nuove specifiche del protocollo syslog che cambiano qualsiasi parte dell'intestazione. I cambiamenti includono l'aggiunta o la rimozione dei campi o un cambiamento di sintassi o semantica dei campi esistenti.

I numeri di versione devono essere registrato con il metodo Standards Action come descritto nel RFC5226. IANA ha registrato le versioni mostrate nella tabella sotto:

VERSION	FORMAT
1	Defined in [RFC5424]

9.2 SD-ID

Iana ha creato un registro intitolato “syslog Structured Data ID Values” devi valori dello Structured Data ID (SD-ID) insieme ai valori PARAM-NAME associati.

Nuovi valori SD-ID e PARAM-NAME devono essere registrati con il metodo IETF Review descritto nel RFC5226.

Una volta che gli SD-ID e i SD-PARAM sono definiti, sintassi e semantiche di questi oggetti NON DEVONO essere alterati. In caso di variazione di un oggetto esistente, un nuovo SD-ID o SD-PARAM DEVE essere creato e il vecchio rimanere inalterato.

Una disposizione è fatta qui per i nomi localmente estensibili. IANA non li registrerà e non controllerà il nome con il simbolo at (ABNF %d64) all'interno.

IANA ha registrato gli SD-ID e PARAM-NAME mostrati nella seguente tabella:

SD-ID	PARAM-NAME	
timeQuality		OPTIONAL
	tzKnown	OPTIONAL
	isSynced	OPTIONAL
	syncAccuracy	OPTIONAL
origin		OPTIONAL
	ip	OPTIONAL
	enterpriseId	OPTIONAL
	software	OPTIONAL
	swVersion	OPTIONAL
meta		OPTIONAL
	sequenceId	OPTIONAL
	sysUpTime	OPTIONAL
	language	OPTIONAL

10. Working Group

Il gruppo di lavoro può essere contattato via email:

```
syslog@ietf.org
```

Gli attuali presidenti del gruppo di lavoro possono essere contatti a:

Chris Lonvick

Cisco Systems

EMail: clonvick@cisco.com

David Harrington

Huawei Technologies USA

EMail: dbharrington@comcast.net

11. Acknowledgments

Gli autori desiderano ringraziare Chris Lonvick, Jon Callas, Andrew Ross, Albert Mietus, Anton Okmianski, Tina Bird, Devin Kowatch, David Harrington, Sharon Chisholm, Richard Graveman, Tom Petch, Dado Colussi, Clement Mathieu, Didier Dalmasso, e tutte le altre persone che hanno commentato su diverse versioni di questa proposta.

12. Riferimenti

12.1 Normative

[ANSI.X3-4.1968] American National Standards Institute, "USA Code for Information Interchange", ANSI X3.4, 1968.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate

Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2914] Floyd, S., "Congestion Control Principles", BCP 41, RFC 2914, September 2000.

[RFC3339] Klyne, G., Ed. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, July 2002.

[RFC3418] Presuhn, R., "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3418, December 2002.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

[RFC4646] Phillips, A. and M. Davis, "Tags for Identifying Languages", BCP 47, RFC 4646, September 2006.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[RFC5425] Fuyou, M., Yuzhi, M., and J. Salowey, "TLS Transport Mapping for Syslog", RFC 5425, March 2009.

[RFC5426] Okmianski, A., "Transmission of Syslog Messages over UDP", RFC 5426, March 2009.

[UNICODE-TR36] Davis, M. and M. Suignard, "UNICODE Security Considerations", July 2005.

12.2 Informativa

[RFC3164] Lonvick, C., "The BSD Syslog Protocol", RFC 3164, August 2001.