UNIVERSITÀ DEGLI STUDI DI PADOVA

Facoltà di Ingegneria

Corso di Laurea in
Ingegneria Informatica

Ingegnerizzazione di un sistema multipiattaforma per l'archiviazione e consultazione di file di LOG

Relatore: **Ch.mo Prof. Federico FILIRA**

Tutor aziendale: **Ing. Giuseppe DONATO**

Tesi di Laurea di:

Francesco NATALE

Anno Accademico 2010 - 2011

Indice Generale

Sommario

Sommario

Questa tesi descrive la mia esperienza di tirocinio svoltosi presso “Ne-t by Telerete Nordest”, azienda di ICT con sede a Padova.
Il compito affidatomi prevede l’elaborazione di un software, basato su un programma sviluppato alternativamente da 4 tirocinanti a partire dal 2009, per l'archiviazione, l'ordinamento e la consultazione di file di log.
Il prodotto in esercizio al mio inserimento era progettato per integrarsi nella specifica realtà aziendale al fine di adempiere a determinate norme di legge atte al monitoraggio della connettività internet degli utenti fruitori del servizio.
L'obiettivo del lavoro è la modifica dell’applicativo in uso per poter ottenere un prodotto espandibile, scalabile e multipiattaforma a partire dal software esistente, così da rendere possibile l’utilizzo dello stesso in altri contesti aziendali con le medesime necessità di Telerete.
Il seguente testo documenta il lavoro svolto durante le 250 ore di tirocinio, descrive l’evoluzione del prodotto e illustra e motiva le scelte implementate.
L'elaborato viene così strutturato: il primo capitolo descrive il contesto lavorativo in cui il tirocinio viene svolto, proseguendo con la presentazione dell’azienda ospitante nel successivo.
Nel terzo capitolo viene presentato il lavoro da volgere per poi riassumere la situazione presente al mio arrivo nel capitolo 4.
Successivamente si passa a descrivere le scelte progettuali adottate e l'implementazione del sistema nel capitolo 5.
Nel capitolo 6, infine, vengono proposte delle possibili vie di sviluppo del software realizzato.

Capitolo 1

Introduzione

1.1 Scopo del tirocinio

La scelta di fare un'esperienza in azienda è nata dal voler approfondire, in un contesto lavorativo, il settore della connettività; in particolare con questa esperienza sì è analizzata la problematica della gestione di grandi moli di dati in ambiti di rete.
Il tirocinio si è svolto presso Telerete Nordest, azienda con sede a Padova, di cui viene fornita una descrizione più accurata nel capitolo 2.
Telerete gestisce le reti WiFi PadovaWiFi, Monselice WiFi ed ESUnet, la rete wireless dell'Università di Padova; queste contano singolarmente svariate centinaia di utenti unici giornalieri, producendo una grande quantità di file log (qualche centinaio di righe al secondo).
Per legge, le informazioni necessarie a rintracciare la navigazione di un individuo devono essere archiviate e rese facilmente consultabili agli organi di controllo per un certo periodo di tempo dalla loro creazione.
Al mio arrivo era presente ed in funzione una piattaforma, sviluppata in precedenza da altri tirocinanti, le cui funzioni, non sempre operative come da progetto, erano state studiate e calibrate per funzionare nella realtà aziendale di Telerete.
L'obiettivo del lavoro è stato ingegnerizzare tale software al fine di renderlo generico, utilizzabile quindi al di fuori dall'ambiente di sviluppo in cui è nato.
Inoltre, per ambire a diventare un servizio potenzialmente vendibile, era desiderabile aumentare le sue funzionalità, con l'obiettivo ultimo di rendere i log raccolti una risorsa e non più un onere.
Date le recenti modifiche delle normative, che rilassano gli impegni da parte del gestore di una rete riguardo il mantenere traccia delle azioni svolte dai suoi utenti, la strada che l'applicativo prenderà nelle sue future implementazioni sarà la raccolta di file di log per sviluppare previsioni affidabili sull'andamento dei servizi in monitoraggio e sullo stato delle macchine della struttura.
Tutto questo oltretutto deve piegarsi a un vincolo economico, per far sì che tale sistema brilli non solo per espandibilità e configurabilità ma risulti il più possibile contenuto come costo.
Per ridurre i tempi di sviluppo, tale obiettivo è stato raggiunto utilizzando come base di partenza il materiale esistente e appoggiandosi solo ad applicativi gratuiti e open source, riducendo così le spese al solo acquisto dell'hardware ospitante l'applicativo.
Per quanto riguarda lo svolgimento dell'esperienza, questa non si è sviluppata durante un periodo temporale continuativo: in parte per il trasloco dell'azienda in un diverso locale, a qualche chilometro dalla precedente sede, in altra dovuta a impegni personali non prorogabili.
Nella prima fase del tirocinio si è proceduto con un periodo di orientamento

all'interno dell'azienda, in particolare nel settore tecnico; questo tempo è stato necessario per l'adattamento alla realtà aziendale,per apprenderne la struttura e conoscere il personale.
Contemporaneamente è stato preso il tempo necessario per la documentazione su linguaggi e software non noti in precedenza, colmando le lacune più evidenti in quanto vincolanti per svolgere l'attività. Oltre a questo, è stato studiato a fondo il problema che ci si accingeva ad affrontare, cercando di chiarirsi le idee su cosa significasse "pacchettizzare".

1.2 Il Progetto LogLoader

Dal 15 dicembre 2009 ogni azienda che fornisca, tra i propri servizi, la connessione ad Internet ha l'obbligo, dettato dalle norme di legge vigenti (155/2005, anche noto come 'pacchetto Pisanu'), di raccogliere i dati che permettano di identificare chi accede ai servizi telefonici e telematici offerti, acquisendo i dati anagrafici riportati su un documento di identità.
Oltre a questo, l'azienda che funge da Internet Service Provider (ISP) deve memorizzare e mantenere i dati relativi alla data/ora della comunicazione e alla tipologia del servizio utilizzato. Escludendo comunque i contenuti delle comunicazioni, si deve essere in grado di estrapolare informazioni sulle attività svoltesi in un certo lasso temporale, ad esempio chi ha visitato un certo sito o quale server è stato visitato da un certo utente.
Al fine di rispettare tale normativa è richiesto un sistema capace di analizzare una grande quantità di file log, anche diversi tra loro, e ottenere le informazioni desiderate.
Per far fronte a questa esigenza l'azienda aveva già messo al lavoro altri tirocinanti negli anni passati, al fine di soddisfare questa necessità nel migliore dei modi. Al tempo furono scartate le ipotesi d'utilizzo di software commerciale, in quanto ritenuto non del tutto in grado di adattarsi all'ambito aziendale.
Il prodotto così sviluppato, utilizzante solo software open source, si occupava di leggere i log prodotti dall'azienda e archiviarli in un NAS secondo determinati criteri.
Questi, anche se funzionale in relazione allo scopo per cui era stato realizzato, risentiva però del lavoro svolto "a più mani", dilazionato nel tempo; il suo codice infatti risultava spesso disordinato, contenente ambiguità e disallineamenti derivati da una mancata revisione globale dell'applicativo. Tali mancanze hanno reso l'implementazione complessa e dispendiosa in termini di tempo, oltre ad a richiedere un maggiore sforzo per la correzione dei bug.
Il sistema finale varrà realizzato con architettura client-server, con accesso all'applicativo tramite browser web.
Come specificato nei successivi capitoli, si è optato di offrire la piattaforma in un unico pacchetto includente hardware e software; anche se questa scelta può apparire in contraddizione alle specifiche di economicità del prodotto, analizzando le fonti di spesa accessorie è risultato meno oneroso offrire un

server con l'applicativo preinstallato piuttosto che impiegare maggior tempo e risorse per le problematiche derivanti da assistenza e sviluppo di versioni parallele del prodotto.
Le tempistiche non hanno permesso di completare il programma in tutte le sue funzioni; lo sviluppo si è arrestato comunque ad un buon punto, tale da poter dire che il nucleo del sistema risulta efficiente e pronto.
Il software realizzato al termine del tirocinio permette la consultazione dei log aziendali, correggendo alcuni bachi trovati nella versione precedente; inoltre è stata realizzata e testata una procedura di prima installazione, che permette di configurare il software e d'integrarlo alla rete di destinazione. Oltre a questo, è stata scritta un'interfaccia per l'amministrazione basilare dell'hardware del server, così da poter essere gestito completamente da remoto con una comoda interfaccia web.
Il prodotto finale permetterà una consultazione completa ed efficiente dei dati archiviati, l'inserimento automatizzato degli stessi e una gestione differenziata delle reti da monitorare con diversi profili d'utenza.
Riassumendo, il progetto da sviluppare durante l'esperienza di tirocinio richiede una profonda conversione: da un software studiato per funzionare con determinati servizi in una precisa architettura di rete a un prodotto che possa integrarsi in un generico parco macchine aziendale, in grado di poter ricevere, catalogare ed interrogare log provenienti da server con servizi anche differenti da quelli originali, il tutto al minor prezzo possibile.

1.3 Il Progetto "Pro.Sy.Go."
(PROcesses and SYstems GOvernance)

Il lavoro svolto rientra in un più ampio quadro di attività di cui si compone il progetto ProSyGo.
Il progetto PROSYGO ha l'obiettivo di modificare l'approccio delle imprese, soprattutto di piccole e medie imprese (PMI), verso l'evoluzione tecnologica.
Attualmente le aziende tendono a mantenere tutta la conoscenza al loro interno, e questo provoca un lento adattamento alle possibilità tecnologiche attuali.
Con PROSYGO si cerca invece di introdurre le aziende ad una realtà di collaborazione con il mondo accademico, in modo da permettere un più rapido assorbimento delle nuove tecnologie ed in modo da mitigare la tendenza delle aziende a creare solo delle soluzioni ad hoc per il cliente, che sono difficili da riusare ed adattare in caso di richieste future di quest'ultimo.
L'approccio tecnologico attuale delle aziende porta inevitabilmente a ritardi, che possono essere un grave handicap per il business dell'azienda e per il progresso tecnologico che questa voglia eventualmente effettuare.
Il progetto vede l'adesione di complessivamente 11 aziende aderenti al Distretto e 3 aziende operanti all'interno del settore Terziario Avanzato, non aderenti al Distretto.
Valutazioni nel settore hanno portato alla individuazione dei seguenti ambiti

strategici di sviluppo ad alto valore innovativo:
1. Business Process Management (BPM);
2. Business Performance Analysis (BPA);
3. Process Mining (PM);
4. Service Oriented Architecture (SOA);
5. Applicazioni che utilizzino e valorizzino la tecnologia RFID;
6. Virtualizzazione e Grid Computing.

Lo stage in oggetto di questa tesi appartiene, come argomento, al punto 3, concentrandosi sull'analisi di grossi moli di dati e sull'utilizzo del process mining per inferirne informazioni.

Capitolo 2

Presentazione Azienda ospitante

2.1 Ne-t by Telerete

L'esperienza di tirocinio è stata svolta presso Telerete NordEst. La sede centrale è situata a Padova, nella nuova struttura condivisa con APS Holding.
Telerete NordEst mette la propria competenza e le proprie risorse a disposizione, tra gli altri, del Comune di Padova, della Provincia di Padova, di APS Holding, dell'Azienda Ospedaliera di Padova e dell'Università degli Studi di Padova.
L'azienda dispone di numerose risorse, sia a livello di competenze umane e professionali, sia a livello tecnologico con un'ingente quantità di dispositivi hardware e componenti software.

Figura 1.1. Logo dell'azienda ospitante.

Ne-t fornisce inoltre servizi di consulenza per ottimizzare i processi organizzativi e servizi a supporto dell'intero ciclo di vita della soluzione. Con una struttura di oltre 60 collaboratori, è in grado di assicurare assistenza post-vendita e servizi di manutenzione ai clienti in modalità 24h x 365.
Una struttura di questo tipo permette di:

- Fornire una gamma di servizi completa che, attraverso l'utilizzo delle tecnologie più avanzate, rispondano alle necessità del tessuto urbano e delle aziende, sia pubbliche che private che in esso operano.
- Porsi come interlocutore privilegiato per chiunque abbia la necessità di ottenere in tempi rapidi risposte concrete a esigenze di natura tecnologica, fornendo inoltre servizi accessori con il valore aggiunto della professionalità espressa dai singoli collaboratori.
- Affiancare gli Enti con il proprio know-how nello sforzo di offrire la maggiore accessibilità possibile ai servizi, sia attraverso l'uso di media diversi sia attraverso la connettività diffusa, con un'attenzione particolare alla tutela della sicurezza dei cittadini.

2.2 Soci

L'azienda nasce come costola del gruppo APS Holding di Padova per operare nelle telecomunicazioni e nell'ICT. Ancora ad oggi si mantiene sotto il controllo

di APS Holding, che ne detiene la proprietà affiancata da altre società, come descritto di seguito.

APS HOLDING	50,123%	
PRONET	38,120%	
INFRACOM	8,044%	
CAMERA DI COMMERCIO	3,713%	

2.3 Servizi

L'azienda offre svariati servizi ai propri clienti, tali da non permettere una approfondita trattazione in questo contesto; ci si limiterà quindi a fornire una succinta panoramica delle principali attività svolte

- Progetti di integrazione tecnologica in ambito urbano: progettazione ed implementazione di tecnologie urbane e progetti integrati, tra cui la rete del metrobus di Padova (nell'ambito del progetto SAE - Sistema di Ausilio all'Esercizio)
  - Videosorveglianza, per conto di Carabinieri, Polizia Municipale e Questura:
  - Progetto “Padova città sicura”
  - Consorzio Padova Ovest
  - Pensiline alle fermate del metrobus padovano
  - Stadio Euganeo
  - ZTL (Zone a Traffico Limitato)
- Installazione infrastrutture e gestione di rete per servizi di connettività wireless:

  - Padova WiFi
  - Unipd WiFi
  - Monselice WiFi
- Fornitura di connettività: MAN cittadina (protocollo HiperLAN), apparecchiature tra cui hotspot e antenne WiFi, gestione di postazioni co-siting per antenne di telefonia mobile, gestione della rete in fibra ottica e disaster recovery
- Fornitura di servizi ISP: connessioni ISDN, DSL, wireless, fibra ottica, Hosting, Housing, servizi di sicurezza, registrazione domini, servizi DNS, server farm e disaster recovery
- Servizi informatici:
  - Sviluppo ed installazione software
  - Sviluppo website, servizi ed applicazioni web
  - Servizi avanzati di gestione anagrafica animale
- Call center: servizi informativi per clienti, servizi di CRM, assistenza e Help Desk, booking di eventi culturali
- Campagne pubblicitarie di eventi nel padovano
- Anagrafe Canina (e di altri animali) di varie città:
  - Padova
  - Trento
  - Roma
- Portali e applicazioni informatiche di supporto all'E-Government e per le aziende
- Fornitura hardware di supporto

2.4 System Integration

L'integrazione ha lo scopo di creare collegamenti e mettere in comunicazione tra loro vari sistemi, servizi, applicazioni che risiedono su sistemi operativi differenti, si appoggiano a database e DBMS differenti, utilizzano linguaggi differenti.
L'obiettivo finale dell'integrazione è la creazione di una piattaforma in cui i vari sotto-sistemi funzionino sia presi singolarmente sia presi globalmente assieme all'intero contesto: all'utente, il tutto dovrebbe apparire in maniera trasparente come un unico sistema.
I metodi di integrazione sono essenzialmente 3:

1. **Integrazione verticale**: è il processo di integrazione tra i vari sottosistemi

in accordo con le loro funzionalità, creando entità funzionali spesso chiamate con il termine silos. Il beneficio di questo metodo consiste nel fatto che l'integrazione viene raggiunta in maniera rapida e coinvolge solo un certo numero di sottosistemi necessari; il costo nel breve periodo è quindi più

contenuto. Dall'altra parte però il costo di mantenimento è più elevato, dal momento che in caso vengano richieste funzionalità aggiuntive, l'unico modo possibile consiste nell'implementare un altro silo apposito.

2. **Integrazione a stella**: è anche nota come Star Integration o Spaghetti Integration. E' il processo di integrazione in cui ciascun sottosistema è interconnesso con ognuno dei rimanenti sottosistemi. Il costo è variabile e dipende fortemente dalle tipologie di interfacce che mettono a disposizione le varie componenti di sistema. I tempi ed i costi necessari per integrare il sistema con nuove funzionalità e nuove componenti crescono in maniera esponenziale nel numero di sottosistemi. Nel caso siano necessari un numero limitato di componenti o un numero limitato di interconnessioni tra essi, il metodo è efficace e si rivela estremamente flessibile e riutilizzabile nell'insieme di funzionalità.
3. **Integrazione orizzontale**: un nome alternativo è Enterprise Service Bus (ESB). E' un metodo di integrazione in cui un sottosistema specializzato viene dedicato esclusivamente a realizzare la comunicazione tra gli altri sottosistemi. Ciò permette un notevole taglio nel numero di connessioni (interfacce) da realizzare: è infatti sufficiente prevederne una per ciascun sottosistema, la quale lo connette direttamente all'ESB. Quest'ultimo è in grado di tradurre un'Interfaccia in un'altra. Il costo di integrazione viene sensibilmente ridotto ed il grado di flessibilità del sistema è elevato. Anche i costi di mantenimento e aggiornamento restano limitati: per esempio la sostituzione di una componente con un'altra affine richiede al più di definire una nuova interfaccia (tra il nuovo modulo e l'ESB), in maniera del tutto trasparente al resto del sistema.

Attualmente in azienda le varie piattaforme di integrazione presenti sono utili ad ottimizzare compiti relativi a:

- Networking, Connettività
- Servizi di Web-call, call-center, ticketing
- Info-mobilità
- E-service, E-Government, E-Service
- Integrazione di applicativi e scambio di dati fra di essi

.

2.5 Piattaforma di integrazione ESB

Mule è un sistema ESB (Enterprise Service Bus) di integrazione orizzontale, soluzione utilizzata come dorsale per servizi software e componenti applicativi. Si occupa di interconnessione tra servizi, brokering (intermediazione), orchestration tra i servizi stessi, routing, messaging, data transformation, sicurezza, ...
Attualmente, la piattaforma Mule presente in Telerete integra:

- un sistema di pagamento, compreso un meccanismo di gestione degli

account (correntemente utilizzato da 2 siti web)
- uno strumento di reportistica, utilizzato per la generazione automatica di report periodici relativi alla gestione della rete del Metrobus padovano, contenente anche dei programmi (oggetti) per la generazione di file pdf, grafici (formato jpg), allegati (come fogli Excel)
- due database contenenti informazioni sulla rete del Metrobus
- un sistema per la generazione e l'invio automatici di fax, per conto dell'Ufficio Diritti Animali di Roma, servizio fornito tramite web service (con stile architetturale di tipo Rest: REpresentational State Transfer)

Mule permette di integrare potenzialmente tutte le applicazioni si desideri; il limite deriva solo ed esclusivamente dalla disponibilità di connettori, oggetti che hanno il ruolo di mettere in comunicazione l'applicazione con la piattaforma di integrazione. Come detto, alla base dell'integrazione orizzontale risiede il fatto per cui i vari applicativi non sono mai in comunicazione diretta tra di loro; al contrario, ciascun applicativo sa come comunicare con l'Enterprise Service Bus (ESB), il quale agisce poi da intermediario (broker) tra i vari moduli integrati.
Ogni applicazione fa uso di uno specifico linguaggio per parlare con l'esterno; Mule si occupa di imparare tutti i linguaggi utilizzati dalle varie applicazioni che integra, e per farlo si serve di appositi connettori implementati al suo interno. Per i protocolli di comunicazione standard, quelli più utilizzati, esiste una moltitudine di connettori liberamente scaricabili da Internet.

2.6 Gestionale ERP

Altra piattaforma presente è l'ERP (Enterprise Resource Planning), sistema informativo aziendale utilizzato per la gestione delle risorse, degli acquisti e delle vendite, della qualità, del rapporto con il cliente e di altri aspetti di business di natura prevalentemente commerciale e amministrativa. Per conoscere ed approfondire gli aspetti di interesse sull'ERP, si sono svolti alcuni incontri con i quadri del settore commerciale e con il referente della compagnia che ha realizzato e venduto il software gestionale a Telerete.
Il sistema ERP Freeway Skyline by Eurosystem, attualmente utilizzato in azienda, mette numerose funzionalità gestionali a disposizione del settore commerciale ed amministrativo. Al suo interno gestisce un database di magazzino, il cui DBMS è Oracle. Tutti i dati gestiti dall'ERP sono contenuti in un unico database; noi siamo interessati solamente alla porzione di database che contiene le informazioni relative al magazzino, in particolar modo ai prodotti tecnologici con cui ha a che fare l'area tecnica. Le funzionalità dell'ERP sono svariate e coprono molteplici ambiti del business aziendale: la gestione del magazzino è soltanto uno di questi aspetti.

CESPITI
Se in ambito tecnico si parla di prodotto tecnologico, nel settore commerciale si utilizza il termine cespite (in inglese asset). I cespiti sono tutti quei valori strumentali, materiali e immateriali, che sono di proprietà dell'azienda; il termine ingloba anche tutte le attività aziendali che sono fonte di profitto per la compagnia. I cespiti aziendali di nostro interesse possono essere:

- merci già in servizio/utilizzo (per esempio apparati di rete attivi e funzionanti);
- merci stoccate in magazzino, già allocate e in attesa di essere messe in opera;
- merci stoccate in magazzino, con funzioni di scorta, per eventualità future.

.

2.7 OTRS

La gestione di manutenzione ed interventi porta il nostro sistema a doversi interfacciare con il sistema OTRS (Open-source Ticket Request System), che si occupa di gestire il flusso relativo alle richieste di assistenza da parte degli utenti sui prodotti e sui servizi acquistati. OTRS è installato in azienda in un apposito server di produzione, ed è accessibile tramite interfaccia grafica a pagine web.

TICKETING
Un ticket è un numero identificativo che viene associato ad ogni richiesta di supporto derivante dall'apertura di ticket. Attualmente i Ticket OTRS vengono creati ed aperti manualmente, da parte delle dipendenti del call center (help-desk):

- in seguito a telefonate di richiesta assistenza da parte di clienti
- in seguito alla ricezione di email o di fax provenienti da clienti (es: utente del personale APS).

Successivamente l'OTRS invia tramite mail la segnalazione al corretto gruppo di addetti dell'area tecnica e mantiene aperto il ticket fino a risposta confermata; l'OTRS tiene costantemente traccia di ogni ticket e del suo stato : 'new', 'open', 'closed unsuccessful', 'closed successful'.

PROFILI GESTITI
OTRS implementa le nozioni di 'utente', 'gruppo' e 'ruolo', relativi ai dipendenti aziendali (membri dell'area tecnica nel nostro caso). Per ciascun gruppo o utente è definito un insieme di 'code' cui quel gruppo o utente è addetto. Per quanto riguarda i clienti dell'azienda, sono definiti 'utente cliente' e 'gruppo cliente'. Oltre alle informazioni di contatto (persona di riferimento, indirizzo email, ...)

ad ogni cliente, privato o ente che sia, è associata una coda, a sua volta legata ad un utente o gruppo di assistenza tecnica.

OTRS::ITSM

OTRS è uno strumento molto importante e molto utile per il supporto agli utenti e per la gestione degli interventi; permette inoltre una collaborazione stretta e veloce tra help desk (call center), che riceve le richieste di assistenza, e l'area tecnica, che fornisce effettivamente il supporto al cliente.
Per venire incontro anche alle esigenze del settore commerciale/amministrativo, in azienda si utilizza OTRS in combinazione con l'estensione ITSM (Information Technology Service Management), che aggiunge a OTRS i concetti di 'servizio' e di 'service level agreement' (SLA).
.

Capitolo 3

La “Pacchettizzazione”

3.1 Definizione del problema

Per “pacchettizzazione” si vuole intendere quella fase che interviene subito prima della distribuzione di un software, con lo scopo di rendere utilizzabile a terzi il codice scritto.
Gli approcci normalmente utilizzati possono essere diversi, da caso a caso; in generale le strade sono due:

1. rilascio dei soli sorgenti
2. rilascio dei binari

Il primo è la soluzione più semplice per lo sviluppatore, poiché lascia all'utente finale la gestione di tutte le problematiche collegate all'innesto del nuovo codice nel proprio sistema.
Il secondo prevede invece la preparazione di un binario installabile. Questa via è da preferire in tutti casi si voglia dare maggiore diffusone e visibilità al proprio software, ma è fonte di non poche complicazioni a monte: bisogna decidere verso quale architettura precompilare il prodotto ed è necessario preparare un installatore diverso per ogni diverso sistema operativo. Senza addentrarsi nelle problematiche che si incontrano seguendo questa strada,, è indicativo pensare che, ad oggi, ancora non esistono standard universalmente riconosciuti per il Packaging system .

3.2 Il caso d'interesse

Il target del prodotto sviluppato è una generica impresa che necessiti di archiviare e consultare file di log. Premesso questo, il software deve essere fornito pronto all'uso, in modo tale da permettere una rapida integrazione con l'infrastruttura esistente.
Per raggiungere questo obiettivo nessuna delle due alternative sopra esposte si presentava interessante; il software deve essere potenzialmente vendibile e questo scartava a priori la prima strada. La seconda invece richiedeva un eccessivo dispendio di energie e non garantiva affidabilità; le infrastrutture dell'utente finale in cui sarebbe stato necessario calarsi risultavano troppo disomogenee per rendere conveniente la realizzazione di un pacchetto auto installante (sistemi operativi diversi, altro software installato, ecc..), senza considerare i problemi a cui si sarebbe andato incontro con l'andar del tempo: aggiornamenti, cattiva amministrazione della macchina ospitante, interferenze con software terzi, ecc..
Per questi motivi si è deciso di adottare una terza soluzione, ovvero fornire il software già preinstallato su una macchina ad hoc. Questa scelta permette di

calare il prodotto così fornito in una miriade di realtà differenti senza doversi preoccupare dei problemi sopra descritti. Inoltre vi sono molteplici vantaggi accessori:

- **garanzia di funzionamento**: la macchina può essere testata in casa e si è sicuri che l'hardware sia ben dimensionato per il lavoro da svolgere
- **maggiore sicurezza**: il cliente è solo utilizzatore e non amministratore del sistema
- **semplificata gestione degli aggiornamenti**: sia per quanto riguarda la loro installazione, sia per lo sviluppo degli stessi, avendo sempre la medesima situazione clonata in varie realtà
- **assistenza più efficace**: lo sviluppo è mirato e le problematiche, in linea di massima, note.

Di contro, il prezzo da pagare è un costo maggiore per l'eventuale l'acquisto del pacchetto, svantaggio in realtà limitato in quanto le realtà che necessitano di un servizio simile, non potrebbero fare a meno di una macchina dedicata a questa funzione.
Ovviamente, per potersi adattare all'architettura di rete del cliente, quest'ultimo dovrà farsi carico di un minimo di configurazione, che verrà richiesta al primo avvio della macchina.

Se, in un futuro, sarà necessario distribuire tale prodotto solo come software, lo si potrà sempre fornire come immagine di un server, ad esempio Virtual Box (disponibile sia open source che proprietaria) o VMWare (solo proprietaria).
Un'altra alternativa accettabile sarebbe quella di fornire un servizio da remoto, ma al momento questa ipotesi è da scartare per via del collo di bottiglia dato dalla banda disponibile e dal carico di rete aggiuntivo che andrebbe a pesare sull'attuale infrastruttura.

3.3 L'indipendenza dalla piattaforma

Un altro vantaggio dato dalla soluzione adottata è aver aggirato il problema dell'indipendenza della piattaforma, ovvero far si che un programma possa girare in qualunque computer lo si voglia utilizzare, indipendentemente dall'architettura o sistema operativo in uso.
Questo punto è stato deciso a priori ed è una delle specifiche irrinunciabili del progetto.
I benefici che conseguono l'adozione di tale politica sono evidenti ma la sua implementazione in fase di realizzazione è tutt' altro che banale.
Fortunatamente già nelle prime fasi di progettazione si era tenuto conto di questo aspetto e si è operato per creare una struttura client-server, rendendo più semplice la realizzazione dell'indipendenza.
Nel caso specifico, il software è stato pensato dotato d’un motore php con database mysql, le cui interrogazioni e relative risposte vengono fornite

attraverso browser web.
Per il lato client, la maggior parte dei browser in circolazione seguono degli standard ben definiti. Dando questo punto per assunto, l'unica azione da intraprendere è il rendere la piattaforma usabile nella sua totalità da pagine php. La scelta progettuale quindi si limita a selezionare quali applicativi e quale sistema operativo utilizzare dal lato server.
.

3.4 Il sistema operativo

Il sistema operativo è, senza dubbio, un aspetto cardine tra le scelte da effettuare.
Questi infatti si occupa di far eseguire correttamente il software di analisi log e gli applicativi con esso utilizzati. Per far sì che questo avvenga con un certo criterio, si andrà ad analizzare quali aspetti e caratteristiche il sistema dovrà avere.

Stabilità
Essendo questa una macchina di produzione, la stabilità è un aspetto imprescindibile nella scelta del sistema.
Per stabilità si intende la capacità del sistema di restare performante, in operatività ed efficienza anche dopo un prolungato periodo d'uso, senza necessità di riavvii o manutenzioni constanti.
Puntare sulla stabilità significa mettere in servizio una macchina e potersi dimenticare della sua presenza, salvo rari interventi di manutenzione una tantum.

Sicurezza
Analogamente a quanto detto sopra, anche la sicurezza è una caratteristica non opzionale.
Premesso che il prodotto è pensato per lavorare in una rete interna e che quindi operi in un ambiente di per se relativamente sicuro, il sistema dovrà comunque non cadere per errori o danneggiamenti involontari provenienti dalla stessa intranet.
In questo contesto si considera sicuro un sistema non amministrabile da chi non ne abbia le credenziali, personalizzabile senza servizi non richiesti e porte aperte non necessarie. Questo gli dovrebbe permettere di essere resistente a virus ed intrusioni non autorizzate.

Aggiornamenti
Una politica di aggiornamenti costante e continuativa permette di poter mantenere un certo standard di sicurezza e modernità degli applicativi. D'altro canto, aggiornamenti rilasciati con eccessiva frequenza non sono una scelta ottima in questa situazione, in quanto la macchina non può permettersi troppi

interventi o riavvii di manutenzione.
La politica di upgrade ricercata quindi vuole una serie di aggiornamenti costanti e a lungo termine, ma a cadenze pluri-mensili e non giornaliere o settimanali

Supporto
La presenza di un supporto al sistema da adottare è senza dubbio auspicabile in quanto, da un lato permette di gestire eventuali errori valendosi dell'esperienza di altri maggiormente competenti, dall'altra può permettere di addentrarsi nel sistema per personalizzarlo e, se necessario, modificarlo, per renderlo maggiormente performante e meglio sagomato alla realtà di destinazione.

Oltre a quanto appena enunciato, nel determinare il sistema operativo da adottare, è necessario considerare anche alcuni vincoli, dati da scelte prese nelle prime fasi del progetto e quindi ereditate con l'avanzare dello sviluppo dello stesso.

Open source
Uno dei vincoli iniziali richiedeva che tutto il codice possibile fosse open source, per cui, non di meno, anche il sistema operativo incluso nel prodotto dovrà rispettare questa scelta.

Utilizzo per fini commerciali
Spesso c'è confusione sull'argomento, ma non tutto il mondo open source è sfruttabile per fini commerciali. Essendo questo un prodotto che andrà potenzialmente venduto, è necessario che le licenze che seguono ai software di terzi, sistema operativo compreso, siano impiegabili con l'intento di trarne un utile.

Disponibilità di software terzi
Il sistema ricercato deve poter installare servizi dati anche da software di terzi, in particolare un webserver, database mysql ed un interprete per i linguaggi di scripting lato server perl, bash e php.

Free
L'intero progetto deve costare il meno possibile per la sua realizzazione, per cui è preferibile adottare prodotti gratuiti quando disponibili.

3.5 La scelta operata

Buona parte del codice è stato scritto in php su server Gentoo.
Php è un linguaggio di scripting molto diffuso e attualmente supportato da un gran numero di piattaforme. Però, ciò nonostante, vi sono dei problemi per la sua portabilità, in particolare quando si voglia migrare da ambiente Linux a

Microsoft Windows. Le differenze sono sufficienti da non rendere eseguibile il codice scritto su entrambe le piattaforme. Volendo sarebbe stato possibile superare questo inconveniente ma avrebbe richiesto un lungo lavoro di adattamento del materiale esistente. Questo ha portato a scartare fin da subito i sistemi Microsoft Windows Server.
Inoltre, dai vincoli esposti precedentemente, sono stati scartati anche i sistemi Red Hat e Suse Enterprise, in quanto a pagamento.

Rimaneva comunque un nutrito ventaglio di scelte nel mondo dei sistemi Unix-like, open source e gratuiti, molti dei quali risultavano ugualmente idonei a svolgere il ruolo richiesto. Tra le distribuzioni Ubuntu server, Debian , Gentoo, CentOS, Fedora e FreeBSD si è optato per Debian, che è risultato convincente su tutti i fronti. Gode di una grande diffusione, un enorme parco software, ha una installazione rapida e, inoltre, si aveva già una certa familiarità con distribuzioni simili.

3.6 Software terzi richiesti

Data la struttura dell'applicativo, questi richiede l'utilizzo di software di terze parti per poter essere utilizzato, sia dal lato server sia da quello client.
Per il primo si richiede un server http, un database mySQL e supporto a php v.5. Inoltre è richiesto che il sistema interpreti i linguaggi di scripting Perl e il Bash. Per il secondo invece è sufficiente un browser web.
Ad essere precisi la libertà di scelta con questi strumenti è stata alquanto limitata.
Infatti, essendo lo scopo del lavoro a me assegnato il completamento e l'ingegnerizzazione della piattaforma esistente, molte scelte erano già state precedentemente adottate.
Questo non ha impedito di operare modifiche al codice o, marginalmente, di struttura, ma per contenere il lavoro nei tempi previsti non si è potuta rivedere l'intera struttura.
A tal proposito, ora si procederà ad analizzare cosa necessita il codice scritto per poter funzionare.

HTTP SERVER
Come http server è stato scelto Apache. I motivi che hanno portato alla sua adozione sono dovuti quasi esclusivamente alla sua notevole diffusione, popolarità ed affidabilità.
Apache ha tra i vari punti di forza la modularità, con un numero di estensioni veramente molto vasto, utilizzabili per qualsiasi tipo di esigenza. Viene utilizzato da più di dieci anni sui web server di tutto il mondo e si è costruito una reputazione di affidabilità e solidità ed è facilmente configurabile, con una sintassi chiara e lineare. Inoltre accompagnato da un'ottima e ben fornita documentazione.

Tutte queste caratteristiche però si pagano in termini di gran consumo di memoria. Esistono alternative altrettanto affidabili ma meno esose in termini di risorse, tra cui il noto Lighttpd o Nginx ma si è comunque preferito lasciare Apache in bundle Lamp (Linux + Apache + Php + Mysql), già preconfigurato all'installazione di Debian. La scelta non è comunque vincolante e, a piacere, è sempre possibile sostituire questo software con un altro più leggero.

DATABASE
Riguardo all'adozione di mySQL, questi era già stato implementato precedentemente alla riorganizzazione del codice. Premesso che un cambio di database avrebbe comportato un certo lavoro aggiuntivo, con la modifica di notevoli parti del programma, questi è sembrato comunque una buona scelta.
Principalmente, i motivi per cui è vantaggioso abbracciare MySQL sono le sue prestazioni, la sua diffusione e l'ottima integrazione che ha con php.

Quanto allo scripting php, bash e perl ci si è adeguati al materiale già presente, cercando quando possibile di unificare il codice, traducendo in php il codice scritto in altri linguaggi.
La scelta di appoggiare php e perl è stata spinta dalla notevole capacità di questi linguaggi di maneggiamento delle stringhe di testo e, di conseguenza, estrazioni delle informazioni utili dai file di log.

Oltre a quanto già scritto, dal lato server sarà presente anche un syslog server, per poter gestire, catalogare ed eventualmente archiviare i log del parco macchine che l'applicativo dovrà monitorare.

3.7 Scelta e dimensionamento dell'hardware

Al termine del lavoro svolto non è stato acquistato nessun hardware specifico per la messa in funzione del software, in compenso questi è stato testato in due server fisici. Il primo è operante in azienda con una ricca dotazione di calcolo e memoria, mentre il secondo, di gran lunga più modesto, è stato impiegato per lo sviluppo, il testing e la demo di presentazione.

Visto il continuo evolvere di nuovo hardware e il continuo aggiornamento dei prezzi dei vari componenti, data la necessità di suggerire un compromesso tra costo e prestazioni, verranno qui esposte soltanto delle linee guida per procedere, quando necessario, all'acquisto di un server per ospitare l'applicativo.
Tutto è fortemente in funzione delle necessità del cliente e dell'importanza che viene data a questo tipo si servizio, dal carico di macchine da monitorare e dalla quantità di log da esse generate. Si è visto che per la gestione e la consultazione di log nell'ordine di 6GB/giorno è sufficiente una macchina con 4Gb di Ram e un affidabile processore dual core, avendo presente che, in fase di query, sarà necessario attendere un tempo variabile (da qualche decina a qualche centinaio

di secondi) prima di avere una risposta.
Con la macchina di test invece, con una modestissima CPU Atom e 1 Gb Ram ddr2, le risposte alle medesime query si misuravano in tempi fino a decine di minuti, senza che la macchina facesse null'altro.

Si procede ora con una breve visione di quali siano i processi del software che vanno ad insistere sulla macchina ospitante:

- le risorse di calcolo servono in buona parte per il processo di compressione/decompressione dei file di log. Servono quindi siano adeguate ma non sono da ricercarsi soluzioni particolarmente complesse o costose.
- la memoria di sistema viene principalmente occupata dal database nel porting dai file di testo alle tabelle e, in minor parte, dai servizi dell'http server. Vivamente consigliabile non lesinare su questo tipo di risorsa.
- la memoria di massa non deve essere particolarmente capiente ma è desiderabile sia molto veloce. Vi verranno copiati temporaneamente i file da consultare durante una query e può venire utilizzata per stoccare i log di giornata prima dell'archiviazione, se si demanda a questa macchina il compito. Per tale motivo si consiglia un disco allo stato solido oppure un'architettura raid veloce.
- la scheda rete è sufficiente sia gigabit ma, visto il loro basso costo, è proponibile un raddoppio per garantire quantomeno una minima ridondanza.

Capitolo 4

La piattaforma esistente

4.1 Origini

Il Progetto è nato con lo scopo di realizzare un sistema in grado di raccogliere ed organizzare grosse moli di file di log e, successivamente, elaborare tali dati con un algoritmo di Data Mining, al fine di ottenere risultati precisi e semplicemente leggibili, tramite grafici o tabelle.
ll software è stato concepito per la raccolta e l'analisi dei dati riguardanti i servizi di connettività forniti da Net by Telerete, in particolar modo per le reti wireless “Padova WiFi”, “Monselice WiFi” e “Unipd WiFi”, come richiesto dalla normativa in vigore (legge anti-terrorismo alias legge Pisanu, descritta di seguito).

Legge Pisanu
Decreto-legge 27 luglio 2005, n. 144 : Legge 31 luglio 2005, n. 155
Ogni azienda che fornisca, tra i propri servizi, la connessione ad Internet ha l'obbligo, dettato dalle norme di legge vigenti (155/2005, anche noto come `pacchetto Pisanu'), di raccogliere i dati che permettano di identificare chi accede ai servizi telefonici e telematici offerti, acquisendo i dati anagrafici riportati su un documento di identità. Inoltre, l'azienda che funge da Internet Service Provider (ISP) deve memorizzare e mantenere i dati relativi alla data ed ora della comunicazione ed alla tipologia del servizio utilizzato, esclusi comunque i contenuti delle comunicazioni, e deve essere in grado di estrapolare informazioni (in un range temporale) quali: chi è entrato in determinato server e quale server è stato visitato da un certo utente o IP (fornito alla connessione).
Per quanto riguarda i tempi di mantenimento delle informazioni contenute nei log, i requisiti di data retention dettati dalle norme di legge sono di 6 mesi per il traffico telematico, periodo eventualmente estendibile a 12 mesi totali su richiesta delle autorità competenti (relativamente ai casi di reati particolarmente gravi).

Oltre a raccogliere questo tipo di dati, si deve anche garantire un metodo di consultazione ed analisi delle informazioni raccolte. Ciò serve in caso di crimini informatici, su richiesta delle autorità competenti in materia.

Vista quindi l'obbligatorietà della raccolta dei dati, il software offre anche la possibilità di sfruttarli per ottenere informazioni utili alla salute del servizio, tenendo attivo un monitoraggio delle connessioni giornaliere e annuali mostrando queste informazioni sotto forma di grafico a barre.

4.2 L'infrastruttura aziendale

La struttura della rete aziendale è proposta nell'immagine seguente

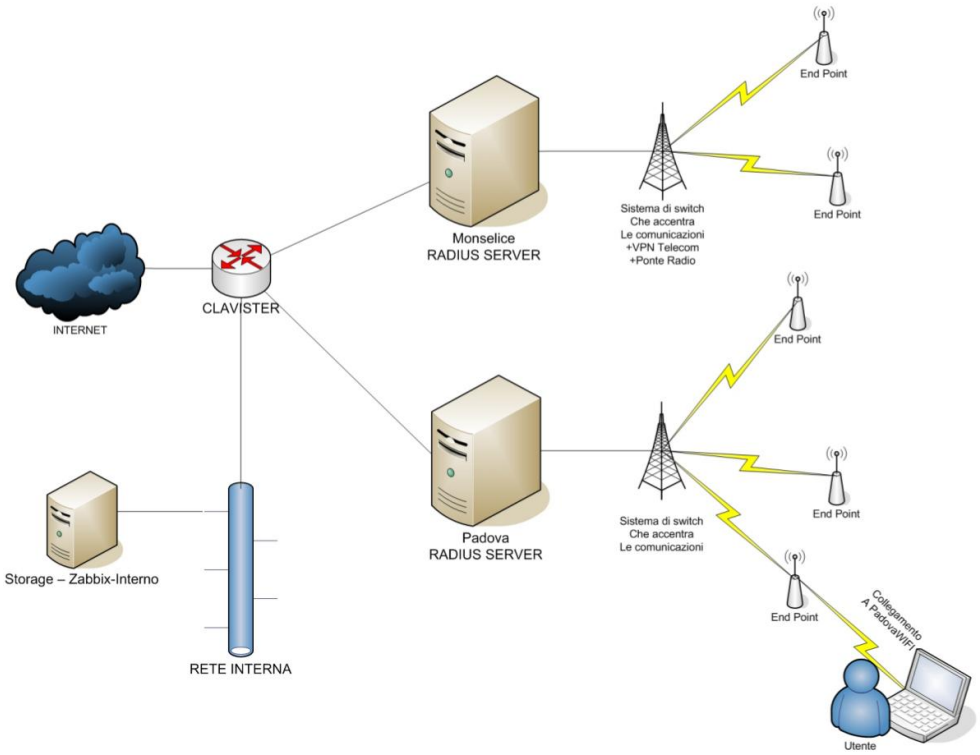

Figura 4.1. Struttura servizio wi-fi

Tutti i clienti del servizio wifi accedono alla rete identificandosi ad un server radius che gestisce gli accessi per una determinata rete wireless.
I vari server radius sono collegati a loro volta al firewall aziendale, il quale fornisce connettività agli utenti e reindirizza i log inviati dai server verso la rete interna, nel server denominato Zabbix, che si occupa del loro trattamento, per poi archiviarli presso un NAS aziendale.

Tutti gli utenti WiFi sono visibili all'esterno con il medesimo indirizzo IP, associato a Telerete, mentre all'interno della sottorete aziendale sono identificati univocamente tramite un IP privato. Di conseguenza, se viene commesso qualche illecito in Rete da parte di un cliente dei servizi WiFi forniti dall'azienda, l'indirizzo figurante sarà di Telerete, la quale deve poter provare l'identità dell'utente incriminato.

Si procederà ora a illustrare i componenti principali del sistema in uso.

Autentificazione
L'identificazione dell'utenza avviene tramite un server Radius per ogni rete WiFi.
Questi si frappone tra il database dei clienti, contenente le credenziali e i dati personali degli utenti iscritti ai servizi Wifi, ed il Captive Portal, che ha il compito di validare i dati inseriti in fase di login.
Il servizio si occupa di verificare i dati forniti dagli utenti, tramite Coovachilli, e di segnalare se le credenziali inserite risultino valide o meno.

Captive_Portal
È la prima interfaccia che l'utente si trova davanti al momento dell'accesso alla rete WiFi ed è implementato dal software Coovachilli; in questa fase si richiede l'inserimento di credenziali per poter usufruire del servizio di navigazione. Per far questo viene inizialmente fornito, dal server DHCP, un IP per permettere all'utente di inviare le proprie credenziali. Ovviamente, fino a quando non va a buon fine l'autenticazione, all'IP rilasciato viene bloccato tutto il traffico uscente. Una volta che il database consultato da Radius conferma a Coovachilli l'effettiva autenticità delle credenziali, quest'ultimo sblocca l'IP dell'utente e ne permette la navigazione libera.
Coovachilli mantiene nel proprio database un'associazione tra IP e username corrispondente ad ogni sessione.

Proxy
Squid viene utilizzato come web-proxy per poter usufruire dei servizi di web-cache, comodi per ottimizzare la banda in uscita e ridurre il collo di bottiglia della rete.

In figura 4.2 viene riassunto il processo logico di autentificazione ad una rete.

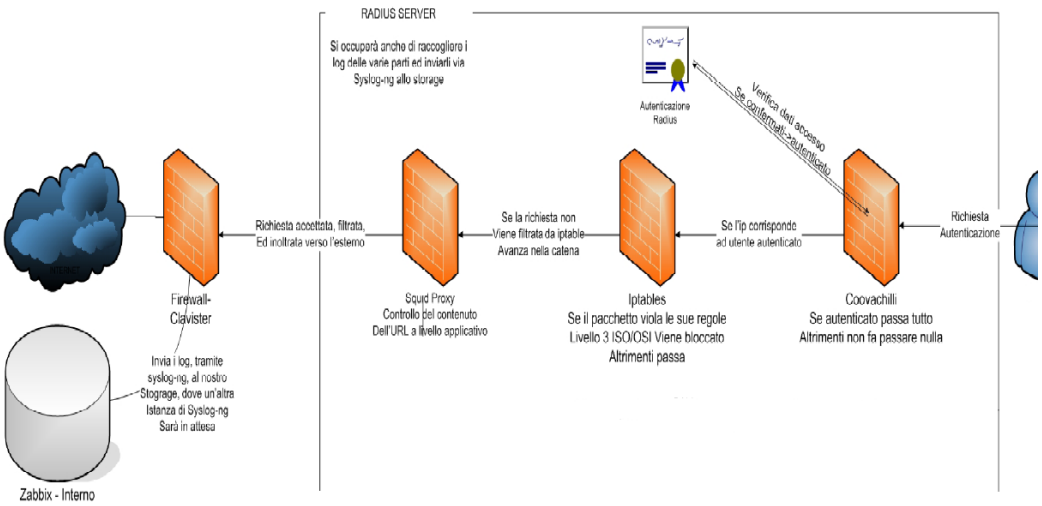

Figura 4.2. Schema logico del processo di autentificazione

4.3 I log monitorati

In una struttura di questo genere vi è la necessità di raccogliere log dai vari servizi per poter fornire delle risposte esaustive alle interrogazioni poste dagli organi competenti.
Di seguito sono analizzati i servizi di cui vengono raccolti i file di log, spiegando quali sono le informazioni utili da essi forniti.

4.3.1 FreeRadius

FreeRadius è un'implementazione open-source del protocollo Radius. Come già detto, nell'infrastruttura di rete FreeRadius svolge il ruolo di server di autenticazione, è dotato di un database PostgreSQL e mantiene traccia:

- dei dati anagrafici (forniti durante la fase di registrazione) degli utenti, che permettono di identificare la persona fisica che accede al servizio,
- delle sessioni di utilizzo da parte di ogni singolo utente, ma non delle disconnessioni,
- dell'accounting relativo all'utilizzo del sistema da parte di ogni singolo utente,
- altre informazioni, tra cui quelle relative ai NAS (Network Access Server), ai profili di accesso al servizio, ecc...

Il log di FreeRadius (di cui un campione è visibile in figura 4.3) contiene informazioni relative ai tentativi di autenticazione da parte degli utenti, sia nel caso di Login OK che in quello di Login incorrect: tra queste informazioni compaiono lo username ed il MAC address della postazione da cui l'utente effettua la connessione al servizio, mentre la password (CHAP) non viene riportata nei log per ovvi motivi di privacy.
Vengono poi registrati eventuali messaggi d'errore, per esempio la mancata connessione con il database PostgreSQL o il fallimento dell'esecuzione di alcuni script di supporto.
Questa implementazione di Radius non fornisce dati riguardanti disconnessioni e utenti attivi al di fuori dell'arco temporale selezionato.

```
Tue Jul 13 08:45:39 2010 : Auth: Login OK: [ctema8/*****] (from client
localhost port 64 cli 00-25-D3-F4-E6-A0)
Tue Jul 13 08:48:06 2010 : Auth: Login OK: [xanus6/*****] (from client
localhost port 151 cli 00-25-D3-72-7F-08)
Tue Jul 13 12:26:36 2010 : Auth: Login incorrect: [marsst/*****] (from client
localhost port 163 cli 00-23-6C-91-F3-5C)
```

Figura 4.3. Estratto di un file di log di FreeRadius

4.3.2 Coova-Chilli

Coova-Chilli, alias il captive portal, fornisce all'utente l'interfaccia di autenticazione in cui inserire login e password e dialoga con il server Radius per effettuare il controllo degli accessi al servizio.
Funge inoltre da DHCP server, assegnando un'indirizzo IP agli utenti che si autenticano con successo e mantenendo l'associazione IP address - MAC address - Data e ora.
Il log di Coova-Chilli quindi registra:

- i login, salvando username e IP address;
- i logoff, solo se l'utente da esplicitamente il comando di disconnessione, e nuovamente salva username e IP address;
- le DHCP request, memorizzando il MAC richiedente;
- le assegnazioni di IP address ad un MAC, da parte del DHCP server;
- i rilasci, da parte di un MAC address, del corrispondente indirizzo IP, che torna quindi a disposizione del DHCP server per nuove allocazioni;
- le notifiche di eventuali anomalie (es. nella comunicazione con il server Radius) o dell'esecuzione di operazioni di routine (es. reload dei file di configurazione).

Nella figura sottostante è possibile vedere un estratto dei log prodotto da Coovachilli

```
May  5 08:07:03 192.168.1.9 coova-chilli[1176]: chilli.c: 2694: New DHCP
request from MAC=xx-xx-xx-xx-xx-xx
May  5 08:07:03 192.168.1.9 coova-chilli[1176]: chilli.c: 2661: Client MAC=xx-
xx-xx-xx-xx-xx assigned IP xxx.xxx.xxx.xxx
May  5 08:08:03 192.168.1.9 coova-chilli[1176]: chilli.c: 2785: DHCP addr
released by MAC=xx-xx-xx-xx-xx-xx IP=xxx.xxx.xxx.xxx
May  5 08:13:06 192.168.1.9 coova-chilli[1176]: chilli.c: 3050: Successful
UAM login from username=XXYYZZ IP=xxx.xxx.xxx.xxx
```

Figura 4.4. Estratto di un file di log di Coovachilli

4.3.3 IPTable

IPTable è un noto firewall utilizzato in ambiente Linux ed esegue un packet filtering a livello 3 dello stack ISO-OSI.; il suo ruolo è quello di gestore del traffico di rete interno a ogni macchina ed è possibile specificare regole molto precise a tal riguardo.
Ogni pacchetto in entrata o in uscita dalla macchina viene analizzato da Iptable una o più volte, a seconda delle regole impostate.
Esistono quattro tabelle sulle quali impostare regole: filter, nat, mangle, raw; nel caso in analisi viene utilizzata quasi unicamente la tabella FILTER.
Il log di IPTable (estratto visualizzato in figura 4.5) memorizza numerose informazioni relative ai pacchetti IP in transito. Possono essere impostate varie regole di logging, a seconda della specifica tabella e della chain utilizzata: INPUT (regole di ricezione pacchetti), FORWARD (regole di inoltro pacchetti)

e OUTPUT (regole di invio pacchetti).
In particolare, vengono salvati i dati di:

- interfacce di rete impiegate (in entrata e in uscita);
- IP address sorgente;
- IP address destinazione;
- porta usata presso la sorgente;
- porta usata presso la destinazione;
- lunghezza del pacchetto;
- identificativo (ID) del pacchetto;
- protocollo di trasporto utilizzato;
- valore della window size di TCP;
- altri parametri specifici, tra cui il TTL (Time-To-Live), il ToS (Type-of-Service), o i flag di TCP.

```
May  8 00:00:05 192.168.1.9 tun0: IN=tun0 OUT= MAC= SRC=xxx.xxx.xxx.86
DST=10.192.0.1 LEN=48 TOS=0x00 PREC=0x00 TTL=128 ID=55775 DF
PROTO=TCP SPT=1292 DPT=3990 WINDOW=16384 RES=0x00 SYN URGP=0
May  8 00:00:05 192.168.1.9 tun0: IN=tun0 OUT= MAC= SRC=xxx.xxx.xxx.86
DST=10.192.0.1 LEN=40 TOS=0x00 PREC=0x00 TTL=128 ID=55776 DF
PROTO=TCP SPT=1292 DPT=3990 WINDOW=17520 RES=0x00 ACK URGP=0
```

Figura 4.5. Estratto di un file di log di IPTable

4.3.4 Squid Log

Squid é un Proxy con capacitá di web-cache, il suo ruolo in questa struttura, fino ad oggi, era ristretto a web-cache, utile per evitare di effettuare richieste ridondanti verso l'esterno. Infatti, spesso capita che piú utenti richiedano le stesse informazioni in brevi lassi temporali; grazie a Squid é possibile mantenere nella cache i dati di ogni richiesta per un breve periodo, rendendo cosí il server capace di restituire la pagina o il contenuto già in cache, senza dover inoltrare richieste uguali verso Internet. Con questo sistema si riesce ad alleggerire il “collo di bottiglia” che normalmente si crea verso l'esterno.
Squid analizza ogni pacchetto che inoltra ed è in grado di estrarre molte informazioni; di queste solo alcune sono di interesse per i nostro obiettivi:

- l'IP address sorgente;
- l'IP address destinazione;
- la porta usata presso la sorgente;
- la porta usata presso la destinazione;
- la lunghezza del pacchetto;
- l'URL visitato;
- l'oggetto MIME della comunicazione;
- il metodo utilizzato (POST/GET/CONNECT/...)
- l'esito della richiesta HTTP (codice);
- altri parametri specifici.

I log di Squid (visibili in figura 4.6) contengono informazioni utili riguardanti gli oggetti in transito, ottenibili unicamente da servizi che agiscono a livello applicativo, come fa, per l'appunto, Squid.
La conoscenza dei contenuti dei server che vengono visitati fornisce la possibilità di effettuare filtraggio ad alto livello e limitare, o se non altro tracciare, la visione di contenuti illegali (es: pedo-ponografia, terrorismo ecc..)

```
May  8 00:00:00 192.168.1.9 (squid): xxx.xxx.xxx.xxx 2546 195.210.96.4
GET application/x-javascript 200 http://codice.shinystat.com/cgi-bin/getd.cgi?
May  8 00:00:00 192.168.1.9 (squid): xxx.xxx.xxx.xxx 2547 81.31.152.204
GET text/html 302 http://top.joomla.it/button.php?
May  8 00:00:00 192.168.1.9 (squid): zzz.zzz.zzz.zzz 50364 193.45.15.113
POST text/json 200 http://live.chess.com/cometd/connect
May  8 00:00:00 192.168.1.9 (squid): xxx.xxx.xxx.xxx 2548 91.209.163.184
GET text/html 302 http://adserving.favorit-network.com/eas?
```

Figura 4.6. Estratto di un file di log di Squid

I servizi finora illustrati sono quelli che producono il maggior numero di righe di log e le cui informazioni sono essenziali per poter soddisfare le ottemperanze di legge.
Per completezza, si rende noto che questi non sono gli unici servizi di cui vengono archiviati log: nello specifico, viene eseguita una raccolta di ulteriori applicazioni per avere accesso a dati riguardanti agli accessi ssh, ai cron job, il dhcp server, alla posta elettronica, ecc.. Questi però non risultano attualmente utilizzati nell'analisi con il software in produzione ma vengono mantenuti per un suo futuro ulteriore sviluppo.

4.4 Syslog

Si è fino a ora parlato di quali unità di informazione si ha necessità di immagazzinare per poter essere archiviate e consultate in caso di necessità. Adesso invece si presenterà il come questi dati vadano ad archiviarsi in una specificata unità di memorizzazione.

4.4.1 Un accenno sul protocollo

Gli apparati spesso sono in grado di generare un log della propria attività ma non hanno memorie di massa su cui conservarli, quindi in caso di incidenti si perdono le informazioni che potrebbero aiutare a determinare le origini del problema. Da qui la necessità, oltre alla convenienza, di avere tali log raccolti in un unico sicuro archivio .
SYSLOG (System Log) è un protocollo appartenente alla Suite di protocolli Internet utilizzato per trasmettere attraverso una rete semplici informazioni di log.
Il client invia al server un certo messaggio testuale di massimo 1024 caratteri, comunemente definito come "syslog daemon" o "syslog server". La semplicità

del protocollo fa si che il server possa gestire messaggi provenienti da una variegata tipologia di macchine, da computer, stampanti, dispositivi di rete, macchinari, ecc. Il server può limitarsi a registrare l'evento, per avere un archivio centrale degli avvenimenti, oppure reagire a particolari livelli di severità chiamando programmi, inviando e-mail, ecc...
Il protocollo SYSLOG nacque nel 1980 come componente di Sendmail, ma presto la sua semplicità e praticità lo portarono alla ribalta per un impiego generalizzato. Questi non rappresenta uno standard rigidamente definito e solo in tempi recenti è stato standardizzato dall' IETF.

I messaggi Syslog possono essere inviati sia via UDP sia via TCP e vengono generalmente spediti in chiaro: sebbene non faccia parte delle specifiche del protocollo originario, è possibile utilizzare un wrapper in grado di fornire cifratura alla connessione tramite SSL/TLS. Per fare un esempio, un'applicazione Syslog viene spesso impiegata in simbiosi con un tunnel.
Syslog può quindi essere sfruttato per integrare informazioni di log provenienti da differenti sistemi, convogliandole in un'unica repository centralizzata.
Da notare che il termine `syslog' viene utilizzato per indicare sia il protocollo Syslog, sia per l'applicazione o la libreria che si occupa della spedizione e della ricezione dei messaggi di log.

4.4.2 L'implementazione aziendale

In azienda viene utilizzato Syslog-NG, una implementazione open source del protocollo Syslog, che estende le funzionalità dell'originale con alcune aggiunte che lo hanno reso più versatile ed adattabile, tra cui:

- compatibilità con apparati Linux, Unix-based e molti dispositivi di rete (Syslog è presente nativamente nei devices prodotti da Cisco); certi applicativi Windows comunicano con questo protocollo,
- il software è open source e offre una buona versatilità a livello di trasporto, permettendo di stabilire se utilizzare TCP oppure UDP (unica opzione nel caso di syslog),
- il numero della porta è personalizzabile (514 di default assegnata),
- effettua un remote logging di tipo incrementale. Questa è una caratteristica molto importante: infatti, considerate le dimensioni che possono raggiungere alcuni dei log aziendali (fino ad alcuni GB/giorno), costruire il file di log remoto mano a mano che giungono le log entries (alcuni KB/s) è decisamente meno oneroso per la rete rispetto al trasferimento dell'intero file, operazione che rischia di sovraccaricare la rete,
- fornisce la possibilità di effettuare secure logging, adottando meccanismi di crittografia basati sul tunneling SSH oppure su SSL / TLS. Questa proprietà può essere ritenuta trascurabile per log che vengono trasferiti all'interno della intranet aziendale, considerata sicura, ma rappresenta invece una caratteristica essenziale qualora le informazioni loggate debbano transitare attraverso la rete internet o altre reti non date,
- fornisce la possibilità di riscrivere, o di aggiungere informazioni, allo

stesso header del pacchetto syslog.

4.5 Il server in uso

Dopo aver descritto quali log risultano utili e come questi vengano inviati, merita un po' d'attenzione anche il come e dove queste informazioni vengano memorizzate.
Questo è necessario se si tiene conto della reale dimensione che i log prodotti possono raggiungere. Per tale motivo viene impiegata una forma di compressione del testo: da un lato per ridurre lo spazio di archiviazione necessario, dall'altra con l'intento di rendere meno oneroso sulla rete il richiamo dei dati archiviati per una loro interrogazione.

Dall'altro lato della medaglia, la compressione e decompressione richiede una certa quantità di risorse di calcolo e, soprattutto, tempo d'attesa aggiuntivo prima di avere a disposizione le informazioni ricercate.
Un buon livello di velocità di estrazione dei dati, o meglio della lettura dei dati contenuti all'interno dei logfile compressi, è data dall'utilizzo di Gzip, che offre il miglior compromesso in termini di tempo e spazio occupato.
I log così compressi vengono quindi immagazzinati all'interno di un'unità NAS di rete, sul quale è presente una cartella, dedicata alla piattaforma di Log Mining.
L'unità NAS è dotata di discrete capacità di memorizzazione e garantisce una certa ridondanza, data da una architettura RAID (Redundant Array of Inexpensive Disks), per poter mantenere l'integrità dei dati.

4.5.1 Il server aziendale

Il server denominato Zabbix è la macchina che si occupa di raccogliere, ordinare ed inviare allo storage di rete i dati. Inoltre ospita il software per la consultazione dei log e, in generale, si occupa di fare da interlocutore tra i vari nodi del sistema in studio.
Da evidenziare subito che il NAS di rete viene reso accessibile a questa macchina tramite Samba: in tal modo la directory remota di archiviazione log viene montata all'interno del server come se fosse fisicamente presente all'interno di Zabbix. Il rendere locale, a livello logico, la directory del server di stoccaggio semplifica la gestione del movimento dei file da parte degli script tra poco descritti.

4.5.2 Configurazione del Syslog

Il Syslog Server è configurato in modo tale da eseguire automaticamente la log rotation giornaliera. In particolare, i file di log vengono organizzati secondo precise regole di formattazione, determinate dai seguenti parametri: host di provenienza, anno, mese, giorno, logging facility (che rappresenta il servizio che genera quella log entry).
Di seguito si riportano le righe di codice che istruiscono il Syslog server a creare i diversi file di log, a seconda dei valori assunti da $*HOST, $YEAR, $MONTH, $FACILITY* e *$DAY*, tutti parametri definiti dal protocollo Syslog. Sono presenti anche i comandi relativi ai permessi che attribuiscono a Syslog-NG la facoltà di creare i file e le tabelle di cui necessita per rispettare lo schema descritto.

```
source logsudp { udp(ip(0.0.0.0)); };
source logstcp { tcp(ip(0.0.0.0)); };
destination logfile {
     file("/root/local/NAS/$HOST/$YEAR/$MONTH/
            $FACILITY/$FACILITY.$YEAR$MONTH$DAY"
     owner(apache) group(apache) perm(0600)
       create_dirs(yes) dir_owner(apache)
       dir_group(apache) dir_perm(0700));
};
```

Blocco di codice 4.1. Estratto della configurazione di Syslog

Tale configurazione impone una struttura ben definita per l'insieme delle directory e dei file utilizzati, insieme che va a comporre un albero gerarchico che rende agevole reperire i dati ricercati.
Sreen albero cartelle
Il Server è impostato per eseguire un **cron job** notturno che ha il compito di comprimere i log file chiusi, ovverro i file relativi al giorno solare precedente al giorno corrente. Il cron job viene eseguito alle 2 di notte, orario in cui, statisticamente, le risorse di calcolo della macchina Linux vengono impegnate in minima parte.
Il cron job viene impostato aggiungendo in coda al file /etc/crontab la seguente riga:

```
0   2  *   *   *    root      /opt/log_find.sh
                                                root,
```

Blocco di codice 4.2. Esempio di cron job

Sono inoltre presenti ulteriori script per poter far svolgere correttamente il lavoro: lo script log_find.sh, scritto in Bash Shell, è composto da due comandi. Il primo ricerca e seleziona tutti i file contenuti nella directory dei log che non siano già stati compressi e che non siano relativi al giorno corrente (in quanto è possibile che Syslog ci stia ancora scrivendo all'interno). Il fullpath di questi file viene fornito come parametro in input allo script log_zip.sh.

```
/usr/bin/find /root/local/NAS/
       ! -name "*.gz" -type f
       ! -path "*`/bin/date +%Y%m%d`"
       -exec /opt/log zip.sh '{}' \;
```

Blocco di codice 4.3. Sezione di uno script

Anche /opt/log_zip.sh è scritto in Bash Shell. Le operazioni eseguite da questo script sono principalmente le seguenti:

1. prende in input il fullpath di un file (passatogli da /opt/log_find.sh),
2. copia il file in una directory temporanea,
3. esegue la compressione del file (e aggiunta dell'estensione '.gz'),
4. ricopia il file compresso nella directory originaria,
5. copia il file compresso in una directory nel disco fisso dello Storage Server. Qualora non fosse presente la directory opportuna, lo script la crea (seguendoi parametri $HOST, $YEAR, $MONTH, $FACILITY),
6. rimuove il file compresso dalla directory temporanea e rimuove il file originario dalla directory di provenienza.

Avendo a che fare con file di log di grandi dimensioni (fino ad alcuni GB/giorno se non compressi), si rende necessario definire delle regole di cancellazione dei log file non più utili; senza un'operazione ciclica di questo tipo porterebbe nel lungo periodo ad un esaurimento delle capacità di stoccaggio. Considerati i requisiti temporali da soddisfare per ottemperare alle norme di legge sul mantenimento dei dati, è stato predisposto uno script Perl, richiamato alla fine del cron job descritto in precedenza, con il compito di controllare il timestamp di creazione di tutti i log salvati, rimuovendo i file più vecchi di un anno. Quest'ultimo valore si può variare, a seconda degli interessi di analisi dei dati o per stare al passo con eventuali nuove leggi.
La seguemte riga, tratta dallo script /opt/log_find.sh, procede a chiamare lo script Perl che opera la rimozione dei vecchi file di log .

```
perl /opt/clean.pl;
```

Blocco di codice 4.4. Sezione di /opt/log_find.sh

Lo script analizza i timestamp relativi a tutti i file presenti nella directory impostata come parametro all'inizio del codice: attualmente il parametro $base_dir punta a /root/local/NAS/.
Anche in questo caso la ricerca avviene ricorsivamente su tutte le sottocartelle della base directory: il comando impiegato è /usr/bin/find.
Il timestamp che viene controllato è il decimo parametro ($stat[9]) dell'array restituito dal comando /usr/bin/stat($file); . Il valore, che rappresenta l'epoch time trascorso dall'ultima modifica, viene confrontato con 31536000 secondi (ovvero 365 giorni) per determinare se il file è da rimuovere o meno. Lo script a sua volta produce un file di log, /opt/clean_file.log, che registra tutte le invocazioni dello script e le rimozioni da esso operate.

Per concludere, il Server Zabbix è dotato che di un sistema di alerting, che avvisa gli amministratori in caso di problemi. Questo sistema sfrutta ssntp, utility di Linux che permette d'inviare mail direttamente all'interno di uno script.
Gli avvisi inviati monitorano il buon esito della compressione ad opera dello script log_zip.sh precedentemente descritto; inoltre viene compiuto un controllo delle dimensioni del file compresso, per verificare se il log sia troppo più piccolo rispetto alla media; in tal caso spetta agli amministratori verificare le motivazioni di questo sotto-dimensionamento.
Di seguito un estratto dello script che svolge questo compito:

```
info=`du -lah | grep kern | grep ${ieri} |
  grep -e "^[[1-9][0-9]\|^[3-9]][0-9]M.*$"`
if [ "${info[0]}" ]
then
echo "Ok,dimensioni file sono nella norma"
else
cat /opt/mailheader /var/log/temp.log |
  ssmtp ++@telerete.net
fi
```

Blocco di codice 4.5. Sezione dello script di controllo

4.6 Il codice del software implementato

Adesso si procederà col presentare il codice del software oggetto del lavoro di questa tesi. Buona parte del tirocinio è stata infatti impiegata per analizzare il prodotto esistente, dalle interazioni che questo ha con applicativi terzi ai commenti riportati tra le righe di codice. Seguirà quindi una rapida descrizione del codice originale, affrontando le funzioni principali di ogni modulo.

4.6.1 Documentazione del codice esistente

Struttura di /var/www/log_mining/

adodb5/
    |- *.* (libreria ADO DB)
include/
    |- connect.php
    |- constants.php
    |- database.php
    |- form.php
    |- mailer.php
    |- session.php
page_files/
    |- style.css
    |- *.GIF/*.JPG

dbquery.php
get_interval.php
index.php
ini_write.php
input_validator.php
interval.sh
iptables.php
iptables_logins.png
iptables_parse.pl
iptrace.php
log2db.php
log_find.sh

```
phpgraphlib_v2.02/
        |- phpgraphlib.php
        |- phpgraphlib_pie.php
anomalous_lines.pl
clean.pl
configuration.php
coovachilli.php
coovachilli_logins.php
database_functions.php
date_convert.php
log_zip.sh
logged_users.pl
login.php
logloader.php
mask1.php
menu.php
playtime.php
process.php
settings.ini
squid.php
```

Gli script .pl e .sh si occupano di eseguire determinate operazioni ottimizzando i tempi di esecuzione mentre il file settings.ini contiene parte delle configurazioni riguardanti i singoli moduli che compongono la piattaforma in analisi.
I file .php sono le pagine, o parti di esse, richiamate dall'utente e forniscono l'interfaccia di dialogo.
Viene utilizzata una libreria grafica per il modulo coovachilli.php, al fine d'avere delle immagini dinamiche indicanti l'andamento del numero di login nelle varie fasce orarie.

4.6.2 Login

La pagina relativa al login è un modulo che permette l'inserimento delle credenziali per utilizzare la piattaforma. Utilizzando le sessioni integrate in PHP, parte di questo modulo (/include/session.php) è richiamata da tutti gli altri per verificare l'effettiva validità della sessione corrente.
Inoltre è stato implementato, anche se non risulta funzionante, il salvataggio di ogni accesso eseguito alla piattaforma.

4.6.3 Logloader

Questo modulo effettua l'operazione di caricamento di generici file di log sul database mysql.
La struttura della pagina consente di inserire il percorso del log che si desidera analizzare, il numero di colonne, i loro nomi e quali di esse siano da utilizzare come indici; tutti questi dati poi vengono utilizzati per creare la tabella ad-hoc all'interno del database Log_Mining.
Infine è presente il campo per inserire la regular expression con la quale analizzare il log ed estrarne i dati da inserire nella tabella appena creata.
Il modulo 'portante' della piattaforma risulta essere il file log2db.php, al quale è affidato l'effettivo compito di estrarre dati utili dai logs e di inserirli nella tabella opportuna.
Gestisce sia file in formato testo non compressi, sia file compressi con gzip, con prestazioni molto simili in entrambi i casi.
Il file log2db.php ha due funzioni in parte ridondanti: entrambe infatti ricevono il pattern che le righe devono rispettare e le colonne da riempire ma la prima è pensata per ricevere un unico file in ingresso, mentre la seconda un array di path

dei file di interesse. Non appena i log verranno caricati, sarà possibile eseguire Query sui dati ed effettuare l'analisi voluta.
Da sottolineare che il numero di campi inseriti nella regular expression deve coincidere con il numero di campi utilizzati nella tabella, altrimenti verrà restituito un errore.
Inoltre il campo di input file accetta solamente un percorso locale al server, senza dare la possibilità di caricarne uno dall'esterno.

4.6.4 Iptrace

Questo modulo è necessario per evadere rapidamente richieste provenienti dalla Polizia Postale in merito alla legge anti-terrorismo.
Le richieste forniscono un elenco di logfile pescati automaticamente dallo storage di rete, i quali vengono passati al sopra citato 'log2db.php'. La scelta dei file da selezionare, a partire dal range temporale desiderato, viene fatta da una funzione Perl che con dati in ingresso "data/ora" iniziale e finale, preleva dai logfile unicamente le righe comprese in questo intervallo e le scrive in un file temporaneo creato appositamente, restituendo un array contenente i percorsi dei file prodotti.
La visualizzazione avviene analizzando i risultati delle select effettuate; una volta caricati i dati sul DB è possibile analizzarli rapidamente se non viene variato l'intervallo temporale dei dati, essendo il caricamento dei file sul database l'operazione più dispendiosa.
Una volta che l' elenco viene visualizzato compare un bottone per ogni linea della tabella di risposta, che permette di inoltrare una richiesta la modulo Squid per estrarre i dettagli della richiesta corrispondenti alla linea presente su Iptables.

4.6.5 Squid

La pagina relativa ai log di Squid segue la struttura di Iptrace.
Il campo iniziale permette di specificare il range temporale, grazie al quale vengono poi selezionati i log corrispondenti e lo script perl provvede ad estrarne le linee utili.
Come già accennato, i dati possono provenire anche dal modulo Iptrace, indicando l'istante e l'IP di provenienza del pacchetto del quale si desidera conoscere ulteriori informazioni. Il modulo Squid in questo caso preleva non solo la richiesta specificata ma tutte le righe corrispondenti a quell'utente nei 5 minuti successivi.
Una volta caricati i dati sulla tabella relativa a Squid è possibile specificare parti di URL da ricercare, Indirizzi IP di Host e di server e costruire automaticamente l'interrogazione che verrà inviata al DBMS. Questo è possibile unicamente ove la struttura della tabella e dei log sia nota a priori.

4.6.6 Coovachilli

Questa pagina permette di visualizzare in maniera grafica l'utilizzo effettivo del servizio, individuando i picchi di utilizzo e gli errori di autenticazione avvenuti nell'ultima ora.

Il grafico si aggiorna dinamicamente grazie ad un cronjob che rileva i login del server Radius di PadovaWifi e li aggiorna una volta all'ora sul Database utilizzato dalla piattaforma.
Segue lo script orario e parte dello script che, ogni mezzanotte, controlla se nella tabella sono presenti entry che hanno superato l'anno di presenza, diventando quindi superflue e, in tal caso, rimosse.

```
set -- $(perl /var/www/log_mining/logged_users.pl
***.**.**.** | awk -F"," '{print $1,$2}')
mysql --user=root --password=****** <<!!
update log_mining.logins set logs`date +%H` = '$1',
all_day = '$2' WHERE date = '`date +%Y-%m-%d`';
quit
!!
```

Blocco di codice 4.6. Sezione dello script "orario"

```
mysql --user=root --password=****** <<!!
DELETE FROM log_mining.logins WHERE date <
'`date --date='365 days ago' +%Y-%m-%d`';
quit
!!
```

Blocco di codice 4.7. Sezione dello script di "pulizia"

Questo script ricerca ed elimina ogni entry che supera l'anno di anzianità, evitando un sovraccarico di dati visualizzati.
All'interno della pagina sono utilizzate librerie Php Graph Lib e Adobe per la creazione dinamica di grafici in PHP.
Queste librerie si appoggiano ai dati contenuti nella tabella "logins" del database mysql, che è strutturata utilizzando un campo per ogni ora del giorno.
Le righe anomale vengono caricate grazie allo script perl "anomalous_lines.pl" e indicano eventuali comportamenti non corretti da parte del server, ad esempio richieste troppo grandi, tentativi di accesso non autorizzati ecc..

4.6.7 Configuration

La pagina di configurazione vuole permettere di modificare le impostazioni del programma dal web, semplificando il cambio dei parametri.
In questo caso è presente un file unico 'settings.ini' contenente varie sezioni del software con relativi parametri e impostazioni.
Ogni sezione corrisponde ad un modulo con all'interno parte delle informazioni utili, come path, script, nome della tabella e colonne , ecc..
Questa struttura facilita la gestione dei file e riduce la possibilità di errori in fase di backup.
Le sezioni presenti sono:

- Database
- Iptables
- Coovachilli
- Squid

- Hosts
- Iptrace
- LogLoader

Sfortunatamente alcune informazioni erano ridondanti, nascoste tra le righe di altri file, facendo venir meno l'efficacia del modulo.

4.7 Il Database MySQL

Il Database Server è una macchina Linux con un database installato: il DBMS (DataBase Management System) usato è, come già accennato, MySQL.
Il database crea una nuova tabella on-demand, su richiesta dell'operatore tecnico: quindi, quando l'utente inserisce nel form gli input richiesti, l'applicativo PHP inserisce automaticamente i dati relativi alla richiesta effettuata.
La creazione della tabella e l'inserimento dei dati in essa sono operazioni che richiedono full access sul database MySQL. Per questo motivo, l'accesso allo script PHP avviene con privilegi di root. Dall'altra parte invece, l'accesso sul database già creato dovrebbe essere in sola lettura ed è presente un secondo account chiamato readonly, con il quale sono ammessi soltanto i SELECT SQL.
Le tabelle principali sono quelle relative ai vari moduli :

- coovachilli
- iptables
- squid

Le tabelle relative alle sole sessioni degli accessi alla piattaforma stessa invece sono :

- logged_users
- user
- active_users

4.7.1 Coovachilli

La tabella squid serve a memorizzare i dati relativi alle connessioni giornaliere e viene aggiornata ogni ora mediante degli script che Cron richiama. Per non affollare la tabella di dati inutili vengono eliminate tutte le entry più vecchie di 365 giorni, in modo tale da fornire sia un trend giornaliero che annuale.
La struttura della tabella è

- Data
- Data e Ora
- 24 colonne, una per ogni ora della giornata
- Log totali della giornata

4.7.2 Iptables

Fornisce il supporto per il caricamento dei dati livello 3 dello stack TCP/IP, ovvero quelli provenienti da Netfilter:

- Data e ora,
- IP dell'host,
- IP del Server oggetto della comunicazione,
- le porte (sorgente e destinazione),
- il protocollo,
- indica se il pacchetto è un tentativo di login oppure una normale comunicazione.

4.7.3 Squid

Questa tabella, come per Iptrace, fornisce il supporto per il caricamento dei dati, mediante l'utilizzo di queste due tabelle è possibile ottenere informazioni che, utilizzando un solo strumento alla volta, sarebbe impossibile ottenere.
Le colonne utilizzate da Squid sono :

- Data e ora,
- ip server e host,
- URL di destinazione,
- oggetto e tipo di comunicazione
- il codice HTTP risultante dalla richiesta.

Capitolo 5

Il lavoro svolto

5.1 Analisi del prodotto

Il tirocinio svolto aveva come obiettivo rendere il pacchetto software scritto da precedenti colleghi esportabile anche in altre realtà aziendali.
Anche se, ufficialmente, il prodotto esistente doveva già permettere una semplice migrazione da una realtà ad un'altra, nei fatti ciò non è stato possibile.
In parte questo è stato causato dall'essere un prodotto studiato e scritto in periodi diversi da persone diverse e la sua natura è andata mutando con il programmatore di turno.
In prodotto in questione nasce come software modulare, cresciuto però sulle specifiche aziendali e diversamente dalle previsioni iniziali e dalla documentazione allegata, non era sufficiente il cambiamento di qualche parametro per renderlo esportabile.
La pianificazione del tirocinio prevedeva un paio di settimane per lo studio del codice presente; questo tempo nella realtà si è protratto ben più a lungo, in parte per una mia carenza di nozioni, dall'altra per una documentazione parziale e incompleta, con codice non uniforme, assemblando assieme segmenti provenienti da molteplici mani.
Di seguito viene indicata l'attività svolta durante il periodo di tirocinio, anche se le tempistiche sono state ampiamente sforate rispetto ai piani originali.

5.1.1 Analisi della documentazione presente

La prima operazione che si è proceduto col fare è stata cercare e studiare la documentazione inerente al software già sviluppato. Sfortunatamente le uniche informazioni disponibili sono state fornite dalla tesi del precedente tirocinante,descrivente un'ottima veduta generale del programma, ma che per ovvie ragioni sorvolava sui dettagli implementativi, in particolar modo riguardo alla stesura del codice. Riguardo a quest'ultimo, i commenti erano scarsi, spesso confusi e in diverse lingue.
Inoltre si è sentita fortemente la mancanza di una diagramma delle dipendenze per i vari file, così da poterne più agevolmente comprendere relazioni e funzionamento.
Data l'assenza di una documentazione precisa, si è dovuto procedere con un'opera di reverse engineering per poter comprendere a fondo il programma esistente.

5.1.2 Analisi del software nel suo insieme

In primo luogo si è provveduto a analizzare il prodotto nel suo insieme, seguendo quanto scritto nella tesi del mio predecessore e verificando la presenza dei componenti da lui scritti. Fatto questo è iniziata una fare di test, per chiarire il funzionamento dell'applicativo, capire a quali problematiche cercava di venire

incontro e se svolgesse correttamente quanto si prefiggeva di eseguire. Si sono così cominciati a scoprire i primi bachi, le migliorie da apportare necessarie e a prendere confidenza con le interfacce create.
Le prove fatte rilevavano dei problemi con la selezione di range temporali includenti due giorni, l'impossibilità di poter caricare un file di log esterno per poterlo analizzare, la mancanza di controlli nei campi di ingresso, necessari per poter procedere con una corretta interrogazione senza fare lavorare il server a vuoto, scomodità nell'inserimento delle interrogazioni e così via.

5.1.3 Analisi del codice dell'applicativo

Terminata questa fase di studio, si è provveduto ad aprire i sorgenti per poter leggerne il codice. È saltato subito all'occhio come il lavoro sia stato eseguito a più mani; complice questo, la scarsità di commenti e la mia mancanza di esperienza, il risultato è stato che la comprensione di quanto scritto è stata lenta e parziale e si è dovuti giungere al termine dell'esperienza di tirocinio per avere una visone globale dello scritto.

In particolare, la confusione proveniva dai numerosi "include" presenti, riferiti a pagine scritte con sintassi notevolmente differenti e che a loro volta dipendevano da altre. Questo, di per sé, non costituirebbe un grosso problema, a meno che variabili con dati di configurazione vengano ripetute all'interno di più file, generando un'enorme confusione al momento di un loro cambio.
Discorso analogo per le funzioni, a volte presenti in locale e altre importate; per queste inoltre vi era una difficoltà aggiunta: spesso la stessa funzione veniva ripetuta più volte per funzionare, di volta in volta, per un caso specifico. Ciò portava ad allungare inutilmente il codice e a rendere le modifiche complesse e macchinose, salvo trovare, compattate e riscrivere una unica funzione loro sostitutiva.
In ultimo, ma non ultimo come difficoltà accessorie, la ridondanza di costanti ripetute da pagina a pagina: anche se il precedente tirocinante aveva eseguito un bel lavoro di compattamento delle stesse indirizzandole verso un unico file di config, sfortunatamente alcune di essere sono rimaste sparse e la loro mappatura e traslazione ha richiesto parecchio tempo aggiuntivo.
Ulteriore nota merita lo stile grafico, probabilmente importato da un modello preesistente, a mio parere inutilmente ricco di parametri, non necessari in un prodotto di questo genere.
Un trafiletto a parte è necessario per parlare degli script, richiamati dalle pagine php: questi, per buona parte, sono di semplice comprensione ma, anche in questo caso, mancano le indicazioni su come usarli, di quali parametri in ingresso necessitino e cosa restituiscano in uscita, oltre che il formato degli stessi. Infine, da segnalare la presenza un file di script perl di oltre 2000 righe, scritto da terzi, che per buona parte è risultato essere non necessario.

5.1.4 Collocazione dell'applicativo

Il nucleo del software si colloca interamente nella cartella \var\www di Apache.
Vi sono inoltre altri script dentro /var/opt, in particolare in uso al modulo

coovachilli e per maneggiamenti sui log.
Vi è una cartella per i file temporanei, creati quando vengono richieste interrogazioni sui file, remota al server e montata tramite samba, in modo da poter essere gestita come se fosse locale.
Lo stesso vale per la directory contenenti i log, fisicamente memorizzati un'unità NAS, ma montata anch'essa sulla macchina per semplificare la gestione delle operazioni sui log.

5.1.5 Software terzi richiesti

È stato ovviamente necessario controllare i vari file di configurazione degli applicativi terzi usati dal programma. Per primo si è analizzato il file php.ini, responsabile delle impostazioni dell'interprete php. Su questo non si sono notati particolati accorgimenti, tanto che è possibile lasciarlo invariato in fase di installazione. È stato comunque di ausilio, in quanto la modifica di determinati suoi parametri hanno permesso di segnalare o ignorare determinati errori sul codice, a seconda delle esigenze.
Lo stesso discorso è valido per l'http server Apache; anche in questo caso non si sono notate variazioni di nota rispetto ad una installazione standard.
L'http server Apache è noto per il suo largo consumo di risorse: visto la modesta utenza a cui dovrà far fronte, è possibile modificare alcuni parametri, in particolari quelli riguardanti il massimo numero richieste simultanee, e disabilitare alcuni servizi, se non utilizzati.
Syslog invece deve essere configurato per ricevere le righe di log dalle varie macchine in servizio e per salvarli e dirottarli nei file adeguati. Si occupa di questo il file "syslog-ng.conf", situato sotto /etc/syslog-ng/, di cui di seguito è riportato un estratto.

```
options {
                            chain_hostnames(off);
                            sync(0);
                            stats(43200);
};
source logsudp { udp(ip(0.0.0.0)); };
source logstcp { tcp(ip(0.0.0.0)); };
source src {
             internal();
           pipe("/proc/kmsg");
           unix-stream("/dev/log");
             file("/proc/kmsg" log_prefix("kernel: ") );
```

Blocco di codice 5.1. Sezione del file "syslog-ng.conf"

Infine è necessario che le righe raccolte vadano archiviate nel NAS con un certo ordinamento; per far questo era presente un cronjob per svolgere l'azione. Si è quindi andati nella directory \etc per visualizzare le impostazioni date al file "crontab", di seguito riportate in parte

```
0 2 * * *       root       /opt/log_find.sh
0 3 * * *       root    sh /opt/mailtest.sh
59 * * * *      root       sh /opt/coova.sh
20 0 * * *       root      sh /opt/coovaday.sh
```

Blocco di codice 5.2. Sezione del file "cronotab"

5.1.6 Analisi dei permessi

I file del software, quelli contenuti all'interno della cartella di Apache, devono poter essere letti, e in taluni casi modificati, direttamente dall'utente interrogante. Per questo motivo l'intera cartella è gestibile con il solo utente "apache", che sarà quello usato sul lato server alla visita delle pagine php da parte dell'utente. Lo stesso vale per la cartella temporanea montata in locale dal NAS di rete; anche in questo caso sono presenti, e necessari, i premessi in lettura e scrittura su tutti i file, mentre per l'accesso ai log archiviati, i permessi concessi sono limitati alla lettura.

5.1.7 Analisi dei database d'appoggio

Una volta esplorati tutti i file interessati dalla piattaforma, si è andati ad analizzare il database di supporto.
Avendo scelto, giustamente, di non avervi accesso tramite interfaccia grafica remota, si è proceduto a fare un'esportazione dello stesso per potervi lavorare senza rischio di fare danni.

Figura 5.1.. Struttura del database

La struttura del database mysql è già stata descritta nel capitolo precedente, tuttavia l'analisi dello stesso ha rilevato come determinate funzionalità previste non erano in funzione. Non era presente nessuna interfaccia per poter decidere quale utenza fare accedere alle interrogazioni, né per crearne. Inoltre erano previste ma non in funzione procedure di protezione da attacchi al sistema, funzionalità interessante ma prematura per la fase di sviluppo del software e

poco utile per l'uso in una rete interna protetta, che quindi andava a complicare inutilmente lo stesso

5.1.8 Studio del contesto di rete

La collocazione del server nella rete era un fattore di importanza relativa, in quanto il software che ci si accingeva a modificare doveva, a prodotto finito, potersi integrare in una architettura di rete differente ed ignota. A tal proposito ci si è limitati a comprendere un'architettura essenziale nella quale un prodotto del genere possa venire inserito. Tralasciando quindi i dettagli implementativi, il server in funzione riceve dei log in tempo reale e ne consulta altri, archiviati, in differita. Inoltre deve avere un'ulteriore porta aperta per poter effettuare le interrogazioni e le consultazioni sulle pagine php.
La macchina non ha particolari accorgimenti di sicurezza, come l'invio e la ricezione di dati su canali criptati. Questa è una modifica facilmente implementabile ma non prioritaria e si è preferito tralasciarla, dando per scontato che la macchina resterà entro una rete protetta.

5.1.9 Studio dei file di log

Questa attività è stata necessaria per poter ristrutturare adeguatamente alcuni moduli del software presente. Inoltre un loro studio si rendeva necessario per comprendere le basi delle “Regular Expression” con cui estrarre le informazioni desiderate.
Essendo però l'argomento incredibilmente vasto, ma il tempo a disposizione limitato, non è stato possibile approfondire la tematica quanto si sarebbe voluto; un'idea sorta durante lo svolgimento del lavoro è stata di realizzare uno strumento che riconoscesse automaticamente, o per lo meno aiutasse ad estrarre, l'espressione regolare di un generico file di log. Il progetto però è velocemente stato accantonato dopo una stima dei tempi necessari per la sua implementazione.

5.2 Esportazione dell’applicativo

Dopo aver passato il primo periodo ad analizzare, testare e prendere confidenza con il programma esistente, si è iniziata l'opera di esportazione dello stesso su un nuovo hardware per verificarne il funzionamento. Come prevedibile non è stato sufficiente effettuare qualche cambio di variabili e l'operazione ha rubato una quantità di tempo ben superiore alle previsioni.
L'azione svolta doveva effettuare l'installazione del prodotto in un'altra macchina locata nella stessa rete, per verificarne lo stato di sviluppo riguardo la sua portabilità.
Dopo aver preparato la macchina, ospitante equipaggiata con Linux e tutto il software accessorio richiesto (webserver, php e mysql), e averla messa in rete, si è provveduto a riversare il codice sorgente estratto dal server Zabbix, limitandosi a modificare le credenziali di accesso del database. Superata questa

prima fase dopo aver trovato la posizione delle costanti cercate, si è staccata la nuova macchina dalla rete aziendale con l'intenzione di forzarla a prendere i log da un'altra sorgente. A tal proposito si era preparata una cartella locale con dei log fittizi, già ordinati e compressi con la stella logica degli originali.
Questo passaggio non è andato brevemente a buon fine per due motivi in particolare:

- la presenza di variabili, contenenti percorsi dei file e parole chiave, ancora sparpagliate tra le varie pagine dell'applicativo, rendendo onerosa la loro ricerca e il loro aggiornamento
- la necessità di collegarsi al database PostgreSQL di FreeRadius per poter soddisfare alle interrogazioni. Questo, si è scoperto, era stato fatto per semplificare l'accesso alle informazioni del cliente, solo lì memorizzate. Di contro però questa implementazione strideva con l'intera logica del progetto, ovvero estrarre informazioni dai log.

Data la necessità di dover modificare piuttosto radicalmente i moduli che si appoggiavano a questo database, si è cominciato a maneggiare il codice proprio da questo punto.

5.3 Ristrutturazione del codice

Dato il tipo lavoro svolto risulta di scarsa utilità riportare per intero l'intera opera di modifica e riscrittura eseguita. Per questo motivo si darà una visione d'insieme dell'esperienza fatta, soffermandosi di volta in volta solo sulle modifiche che si ritengono maggiormente significative.

5.3.1 Modulo IPTrace

5.3.1.1 Il Database FreeRadius

Si è iniziata l'opera di ristrutturazione partendo dal “problema FreeRadius” del modulo “iptables.php”, il più complesso e lungo, come numero di righe di codice. Il database richiama l'anagrafica degli utenti iscritti al servizio wifi e tali informazioni non sono riportate nei file di log. La scelta fatta ha portato dunque a una perdita inevitabile di informazioni e la consultazione di quel database resta comunque inalienabile per poter identificare una persona fisica dal suo nick.
La modifica operata infatti estrae dai log di Coovachilli l'id dell'utente associato all'ip assegnatoli, con il quale si possono ricercare i dati di navigazione sui log di iptable e viceversa. Fortunatamente non è stato necessario includere nella lista dei file da consultare i log di radius, contenenti Id, password e MAC Adress, poiché Coovachilli ad avvenuta autentificazione mantiene nella propria righe informazioni sufficienti per una univoca identificazione.
Una complicazione aggiuntiva è stata data dal dover verificare se un dato utente sia connesso o meno ad una tale orario, poiché i log di Coovachillli riportano una disconnessione solo se l'utente ne invia l'ordine esplicitamente; in caso

contrario l'ip assegnato all'utente rimane assegnato fino a quando non verrà riassegnato automaticamente ad altri tramite Dhcp. A salvare la situazione è intervenuta un'impostazione assegnata al captive portal, che per liberare gli indirizzi non più in uso, ogni mezzanotte sconnette tutta l'utenza richiedendone nuovamente l'autentificazione.
Sfruttando questo fatto si è creata una finzione che estraesse l'elenco degli utenti connessi per fasce orarie, associando per ognuno di essi uno o più ip (in caso di connessioni multiple di un utente).
Tramite questa tabella diviene semplice identificare uno o più utenti: è stata quindi realizzata un'altra comoda funzione, "chie", che svolge il lavoro di associare uno o più Id ad un ip visitato. Una volta ottenuto l'id univoco dell'utente basterà fare una interrogazione da un'altra macchina per ottenere i dati cercati.

5.3.1.2 Le costanti non dichiarate

All'interno di questo, come di altri moduli, sono state trovate ripetute più volte delle stringhe chiave non dichiarate in una costante. Questo da un lato fa diventare il codice molto statico, rendendo difficoltosa l'applicazione di modifiche; dall'altra ne rallenta la comprensibilità, dovendo sempre sapere a monte il significato di determinate stringhe.

```
$gz_file_name = "kern." . $year . $month . $day . ".gz";
                $gz_file_path =
"$stored_kern_dir/$ip_host_radius/$year/$month/kern/$gz_file_name";
```

Blocco di codice 5.3. Sezione del modulo "iptrace.php"

Discorso analogo per i percorsi dei file locali, per i quali sarebbe corretto dar la possibilità di modifica da un file di configurazione o, se il caso lo rende possibile, farli estrapolare direttamente dall'interprete php.

```
$rangeoutput2 = '/root/local/' . $gzfilename2 . '_00:00:00_' . $hms2 . '.gz';
```

Blocco di codice 5.3. Sezione del modulo "iptrace.php"

La presenza di quest'ultimo tipo di stringhe inoltre rende particolarmente problematica la portabilità di tutto il software e rendendo di conseguenza onerosa la loro correzione in modo proporzionale alla frequenza con cui si presentano nel codice.

5.3.1.3 La gestione di reti aggiuntive

Per un software di monitoraggio di questo tipo è essenziale avere la possibilità di inglobare log provenienti da reti differenti, quindi essere scalabile a seconda delle necessità dell'azienda. Quest'aspetto era stato pensato in precedenza ma non ben implementato nella pratica. Era infatti presente una voce, nella pagina di configurazione, per poter aggiungere un'ulteriore rete da cui attingere i log, ma veniva dato per scontato che ogni espansione dovesse essere identica alle reti

precedenti, salvo nome e ip del server Radius.
Inoltre si era fortemente legati alle sintassi precedenti, tanto che sarebbe stato sufficiente cambiare nome o aggiornare un servizio operante nella rete per non renderlo più monitorabile.
Per svincolarsi da questa scelta e dare la possibilità di impostare sintassi, indirizzi e storage differenti per ogni rete si è deciso di modificare radicalmente il file di configurazione. Si è quindi passati da un unico (in realtà due, considerando quello dedicato ai dati del database) file di config a una struttura a stella: un file "di core" che mantenga dati utili alla sola struttura comune e gli altri dedicati ognuno alla propria rete. Così facendo i dati di sistema, le credenziali d'accesso al database, e l'elenco degli host saranno nel centro stella, mentre i dettagli della singola rete verranno scritti su un file dedicato.
Se da un lato può apparire scomodo mantenere un certo numero di configurazioni distinte, dall'altro risulta più semplice individuare un determinato file di interesse per, ad esempio, farne un backup o sostituirlo.
Inoltre questa scelta porta con se una serie di benefici, tra cui:

- **robustezza**
  se una singola rete viene configurata male o se il file di config risultasse danneggiato, questo bloccherebbe l'accesso solo alla singola rete e non a tutta la piattaforma.
- **scalabilità**
  si possono configurare o eliminare più reti senza vincoli. Questo permette anche di sperimentare il sistema con delle reti virtuali oppure di azzardare configurazioni su servizi secondari senza intaccare quelli vitali.
- **sicurezza**
  disporre di diversi file di configurazione permette una semplice gestione di utenti con permessi distinti, lasciando all'amministrazione il controllo dell'intero sistema e delegando a un responsabile la supervisione di una singola rete. In questo modo, in caso di poca oculatezza o errori del singolo, l'intero sistema resterà operativo, salvo la rete di chi ha fatto il danno. Oltre a questo è interessante notare che risulta semplice anche ripristinare la singola rete (avendo cura di aver fatto un backup della configurazione), senza dover intervenire, con il rischio di complicare il problema, sul nucleo del sistema.

Tutto questo si paga aggiungendo qualche riga di codice alle singole pagine e rendendo più lunga la configurazione iniziale; prezzo modesto, considerando che la configurazione di una rete avviene una tantum, al contrario del suo monitoraggio.

5.3.1.4 Gestione dei log del giorno corrente

Similmente a quanto accennato prima, non era prevista la possibilità di andare a prendere i log giornalieri in percorsi che non fossero già montati nel sistema. Premettendo che era già stato reso possibile esaminare i log prodotti nella giornata corrente (e quindi non ancora archiviati), non era però possibile

montare destinazioni differenti da quelle già impostate.

5.3.1.5 Accorpamento delle funzioni evocate

Questo modulo in particolare, anche se il problema si è presentato in quasi tutte le pagine, si trova a dover richiamare un notevole numero di funzioni per poter operare. Inizialmente queste erano sparse tra numerosi file importati dalla pagina, oltre che dichiarate localmente nella pagina dello stesso modulo.
Per facilitare la lettura del codice e garantirne un miglior riutilizzo, si è provveduto ad accorpare il maggior numero di funzioni in un nuovo file ad esso espressamente dedicato.
Dopo questa operazione, resa lenta dal fatto di dover ricercare e cambiare i percorsi su ogni modulo, è iniziata un'ulteriore opera di accorpamento, cercando di rendere maggiormente elastiche le funzioni esistenti e integrandone di nuove, documentando il loro scopo ed il loro utilizzo.

5.3.1.6 Debug e grafica

Si è proceduto infine con la correzione di alcuni bug, trovati nella prima fase di test, il più grave dei quali agente procedendo con una interrogazione in un intervallo temporale comprendente il giorno corrente e il precedente. In questo caso non venivano visualizzati i risultati del periodo temporale dopo la mezzanotte, perdendo tutti le eventuali corrispondenze dell'ultimo giorno. Il problema è stato risolto con una modifica parametrica in un punto condizionale.
Per quanto riguarda la grafica della pagina, rigore vorrebbe che fosse fatto un accorpamento delle pagine di stile, cercando di eliminare elementi non utilizzati. Non essendo questo un obiettivo di primaria importanza, ci si è limitati a eliminare delle text-box non utilizzate e uniformare su tutta la piattaforma i crediti di fine pagina, tramite un import.

5.3.2 Le Funzioni

Le funzioni, in un software, sono come le macchine nell'edilizia: non ci si può rinunciare, salvo perderci un sacco di tempo. Premettendo inoltre che un software deve essere efficiente perché sia utilizzato, si capisce bene l'importanza d'avere delle funzioni efficienti, ordinate e ben documentate.
Nel codice dell'applicativo le funzioni erano per lo più sparse, non documentate e spesso ripetute con piccole variazioni.
Non essendo un'operazione entusiasmante, all'inizio del lavoro si era cercato di evitare quest'onere, ma è bastato “riscoprire l'acqua calda” un paio di volte per convincersi che l'ordinamento e la catalogazione delle funzioni presenti non poteva attendere oltre.
Oltre, a qualche funzione sparsa tra le pagine, buona parte di queste erano già state raggruppate in alcuni file, che si andranno rapidamente a vedere:

- ***log2db.php***
  Al suo interno sono importantissime due funzioni che svolgono il parsing dei log verso le tabelle del database mysql. Le due (generic_log_parsing e multi_log_parsing) sono molto simili tra loro; la differenza sostanziale è data dal tipo di input che gli viene fornito.

- ***date_converter.php***
  Contiene delle funzioni che operano le conversioni di formato per le date dei log. Vi una specifica funzione per ogni servizio dell'azienda che il software si propone di analizzare e un'ulteriore funzione che le ingloba , avendo come parametro di ingresso un valore numerico, a seconda del tipo di log che gli viene passato.
- ***playtime.php***
  Sono presenti due semplici funzioni per sommare ore e date tra loro, ottenendone il risultato: poco utili in quando esistono funzioni php già incluse con le stesse funzionalità.
- ***input_validator.php***
  Mescola funzioni per la correttezza degli input con altre per il controllo di date.
- ***database_functions.php***
  Come suggerito dal nome, mette a disposizione strumenti per creare e maneggiare tabelle in mysql.
- ***t.php e test.php***
  File non utili, in quanto usati, pare, come test per script javascript.

Anche se la suddivisione per categoria poteva andar bene, mantenere vari file senza documentazione sugli stessi rende meno evidente la loro presenza. Inoltre, dato il non eccessivo numero di righe e l'impiego spesso promiscuo delle stese, si è provveduto a creare un nuovo file "funzioni.php" che unisce e amplia le funzionalità date dai precedenti file.
Il nuovo file è stato quindi diviso in varie sezioni interne, con commenti standardizzati per ogni funzione indicante funzioni e utilizzo della stessa.
La sezione "Data-Ora" mantiene un paio di funzioni precedentemente create e unisce le restanti in un'unica nuova, utilizzabile con qualsiasi tipo di log, a patto di fornirne la sintassi (presa, questa, dallo standard delle espressioni regolari).
Segue di seguito il settore per la manipolazione dei file di log, ove sono presenti le nuove funzioni "connectionlist", "userconnectiontable" e "chie", responsabili dell'estrazione dei dati prima forniti dal database di Freeradius.
Continuando, sono state riportare le funzioni provenienti da log2db.php, fatte le dovute correzioni, e altre di controllo dati e di generica utilità.
Infine è stata creata una nuova sezione con funzioni pensate per amministrare il sistema e l'hardware in uso. Quest'ultime meritano un occhio di riguardo, in quanto, dovendo operare con privilegi root, sono state formulate per chiamare o esser chiamate da script esterni, dai quali ereditano i permessi limitatamente all'operazione da svolgere.

5.3.3 Modulo Logloader

Il modulo LogLoader nasce con l'intensione di analizzare file di log esterni a quello prodotti dal parco macchine locale. Questo può essere utile in caso si debbano effettuare particolari ricerche o rivendere questo tipo di servizio a terzi da remoto.

Si è cominciato a lavorare su questo modulo dopo aver terminato buona parte della ristrutturazione del modulo IPTrace, con l'intenzione di renderlo l'equivalente generico di quest'ultimo. La prima cosa su cui ci si è focalizzati è stato il modulo di input del file; infatti questi permetteva l'analisi dei soli log raggiungibili localmente dal server Zabbix. Non si è capito se questa scelta fosse voluta per motivi si sicurezza o, al contrario, fosse una dimenticanza, ma di sicuro rendeva il modulo praticamente inutilizzabile. Per tal motivo si è provveduto ad effettuare le modifiche necessarie affinchè fosse possibile caricare file dall'esterno, file che vengono memorizzati nella directory temporanea "di lavoro" dell'applicazione. Si è fatto comunque in modo che, automaticamente, il sistema riconosca se si stia passano un percorso come input, caricando in quel caso il file a cui va a puntare la pach inviata.
Il secondo step programmato prevedeva l'introduzione di un sistema semplificativo per l'inserimento della regular expression necessaria all'analisi del log esterno. Sfortunatamente, si era già abbondantemente fuori dai tempi prefissati e si è dovuto abbandonare il progetto a favore della procedura di prima configurazione del software e amministrazione del server con cui verrà ceduto in bundle.

5.3.4 Menù dinamico

La necessità, in un futuro prossimo, di dover gestire utenti con privilegi differenti ha fatto cadere l'attenzione sul bisogno di fornire ad ognuno di essi una visuale con accesso ai soli moduli a cui siano autorizzati. Il modo più semplice per ottenere questo risultato è stato lavorare sul menù del sistema.
Questi non è altro che un file di testo con del codice html, l'elenco delle label e relativi indirizzamenti locali. Nell'ottica di fornire questa funzionalità ma di impiegarci la minor quantità ti tempo possibile, per recuperare parte del ritardo, si è scelto di mantenere la precedente struttura, contenente all'interno le sole voci essenziali e disponibili a chiunque. Quindi, all'interno dello stesso, tramite un "include" si procede a inserire la visione personalizzata sotto forma di un ulteriore file di testo o di una variabile passata tramite post, che permetterà la costruzione della pagina con la visuale e i collegamenti adatti.
Unendo quest'aspetto con quando detto in precedenza sui file di configurazione strutturati a stella, si può dire di aver predisposto tutto il necessario per la futura costruzione e implementazione di un modulo per la gestione delle utenze.

5.3.4 La scelta implementativa

Avendo volendo puntare sulla stabilità dell'applicativo e su una sua semplice integrazione in una struttura di rete incognita a priori, pur mantenendo la facilità d'utilizzo, si è deciso, in corso d'opera, di offrire un prodotto basato su soluzione ibrida, preinstallando il software in un server fisico.
Questa scelta d'implementazione ibrida ha reso necessario dover gestire diverse situazioni vincolate da questa decisione, in particolare la necessità di integrare due componenti essenziali nell'applicativo:

- un modulo di prima configurazione
- un modulo per l'amministrazione e gestione dell'hardware.

Di seguito si andrà ad analizzare la loro progettazione, implementazione ed un test finale per verificarne il funzionamento.

5.4 Procedura di configurazione iniziale

Pur rinnovando e modificando profondamente il codice esistente, i cambiamenti fin'ora eseguiti avevano sempre implementato moduli o materiale già esistente.
Al contrario, un modulo di prima configurazione risultava inedito, non essendoci mai stata la necessità di trasferire il prodotto in altre realtà.
Si è quindi proceduto con la progettazione di una pagina che, data la natura ibrida del progetto, aiutasse il cliente finale in ogni fase della prima installazione; inoltre per poter venire incontro ai successivi sviluppatori e alle future nuove funzionalità, si è deciso di strutturare il modulo stesso in vari settori, rendendolo così dinamico e semplicemente implementabile a seconda delle necessità.

5.4.1 Parametri di rete

Prima di procedere con la personalizzazione del software, vi è un altro parametro fondamentale per l'utilizzo del prodotto. Infatti, la scelta di una architettura ibrida comportava la realizzazione di una piattaforma di gestione della macchina fisica.
Partendo con questo presupposto, il primo step della configurazione si è occupato dell'integrazione della macchina nella rete di destinazione, con la richiesta dei dati relativi: ip, subnet mask, gateway, dns, ecc...
Come valore di default si è optato per un ip statico rispetto a un dhcp:

- in primo luogo perché non si crea confusione per trovare il server al primo avvio,
- in secondo luogo perché logicamente più coerente in quanto, salvo eccezioni, la macchina dovrà avere un ip statico per ricevere i log dai servizi in monitoraggio.

5.4.2 Gestione del Database

Terminate le incombenze sul posizionamento della macchina in rete, si è passati ad analizzare di quali dati il software necessiti per funzionare.
Data la sua cruciale importanza, si è partiti con la gestione del database; infatti senza una sua preventiva creazione sarebbe risultato impossibile proseguire la configurazione, avendo quasi ogni modulo uno spazio di lavoro in esso. Inoltre l'accesso alle pagine viene gestita sempre con l'ausilio del database, risultando quindi fondamentale anche per la sola visualizzazione delle pagine.
I dati che si è deciso di richiedere sono solamente il nome del database, la password di root e le credenziali di accesso al sistema per l'utente amministratore.
Il nome del database viene richiesto, anche se poteva esser assegnato

automaticamente in funzione della rete da monitorare, nell'ottica di future estensioni, che permettano l'utilizzo e la migrazione in database in server esterni.
La password di root del database invece viene richiesta per garantire la privacy dei dati al cliente, sollevando così il produttore da responsabilità riguardo gli stessi.
Scelta opposta invece per la password di root del server: questo in parte per offrire, in futuro, assistenza remota, e soprattutto per evitare danneggiamenti involontari alla piattaforma.

5.4.3 Selezione e configurazione dei moduli

Finalmente, terminate le fasi preliminari, si passa alla configurazione dei moduli dell'applicativo.
Partendo dal principio che potenzialmente ogni cliente può disporre di un'architettura a se stante, con propri servizi e propri log, si è reso necessario fornire un ambiente quanto più personalizzabile possibile. D'altra parte i singoli servizi operanti in rete sono, tutto sommato, limitati in numero e per poter offrire commercialmente un prodotto del genere è necessario che buona parte dei servizi disponga di un modulo già implementato.
Per venire incontro a queste esigenze tra loro contrastanti si è strutturata una pagina che da un lato permetta di selezionare dei moduli già pronti e funzionanti con determinati servizi, dall'altra di poter inserire al volo un nuovo modulo, venendo così incontro a particolari necessità o a servizi non ancora studiati.
L'obiettivo nel tempo sarà d'incrementare la quantità di servizi monitorabili e rendere quindi il prodotto sempre più completo, pur mantenendo la possibilità di personalizzarlo con moduli nuovi o “proprietari”.
Selezionati i moduli da implementare nel sistema mancano ancora i dati richiesti per la loro configurazione.
Se per i moduli noti non presentano perplessità di sorta, gli interrogativi invece arrivano per i moduli non noti. Dopo aver tentato di approfondire l'argomento, per mancanza di tempo e per la necessità di formulare uno standard per la creazione di nuovi moduli, si è preferito rilassare momentaneamente i vincoli e mantenere la coerenza con i moduli già implementati. Le richieste quindi si limiteranno ai parametri “ip” e “percorso locale” ove siano locati i log nelle macchine di servizio, nello storage di rete per l'archiviazione degli stessi e le relative stringhe in sintassi regolare per il parsing delle informazioni.
Consci che questo rappresenterà presto un limite della piattaforma, si demanderà a successivi studi la realizzazione di un sistema più organico per la creazione di modelli sufficientemente duttili per il monitoraggio di un generico servizio.
La procedura di prima installazione termina quindi con un riavvio della macchina, al seguito del quale il sistema entrerà in funzione.

5.5 Modalità e scelte d'implementazione

La scelta di fornire l'intera configurazione da remoto con un'interfaccia web è data della volontà di offrire un prodotto quanto più trasparente al cliente, che deve poter godere del servizio senza la necessità di conoscerne il funzionamento. Per lo steso motivo la macchina è stata interamente blindata, fornita senza credenziali di amministrazione del sistema operativo, la cui manutenzione resterà al produttore.

5.5.1 Configurazione di rete

Le sole azioni richieste sono accendere della macchina e seguire una procedura guidata per l'installazione di una o più reti.
Per poter perseguire quest'obiettivo è stato necessario ricorrere alla modifica di alcuni file di sistema, possibile solo con i privilegi di amministratore. Non essendo concepibile fornire all'utente tali credenziali, per ovvie ragioni di sicurezza,è stato aggirato l'ostacolo attraverso il richiamo di alcuni script; questi sono eseguibili ma non visualizzabili o modificabili e vengono richiamati da delle funzioni in php. In questo modo i permessi rimangono isolati in file non accessibili dall'esterno, garantendo una certa sicurezza.
Sfruttando questo meccanismo è possibile amministrare il server con un notevole controllo, senza mai disporre direttamente di permessi di alto livello.

5.5.2 Configurazione del database

Riguardo alla strutturazione segmentata del modulo di primo avvio, questa scelta progettuale ha portato a molteplici benefici.
In primis, già citato come motivo principale, la possibilità/necessità di espandere la procedura con ulteriori funzionalità senza doverla riscrivere per ogni modifica o aggiornamento.
In secondo luogo si incentiva una politica di riutilizzo e riciclo del codice. Quest'aspetto infatti è già stato sfruttato, riciclando quasi interamente la struttura di primo avvio per lanciare la configurazione di ulteriori reti da monitorare. Questo permette non solo un risparmio di lavoro e una maggiore leggibilità del software, ma anche una maggior omogeneità dello stesso, presentando all'utilizzatore finale sempre le solite viste, rendendogli quindi più agevole l'utilizzo della piattaforma.
Per finire questa scelta permette di mantenere la medesima struttura per creare ulteriori procedure guidate, in ausilio a futuri moduli, come la creazione di un nuovo gruppo o di un nuovo utente o per la configurazione di applicativi terzi collegati a questo progetto.

5.5.3 Configurazione dei moduli

L'implementazione dei nuovi moduli, se non presenti di default, avviene attraverso il caricamento del file php e degli eventuali script aggiuntivi; è da precisare che i file caricati godranno dei permessi limitati dell'utente del server http. Da un lato questo non compromette la sicurezza del sistema, dall'altro impedisce di caricare funzionalità che vadano ad operare nell'amministrazione

del sistema; servizi di questo genere potranno essere implementati solamente dal produttore.
Inoltre, come descritto nei paragrafi precedenti, la possibilità di utilizzare gli stessi moduli in reti differenti è permessa dalla gestione "a stella" dei file di configurazione. Questo permette, tra i suoi vantaggi:

- maggiore sicurezza per un più dinamico uso della piattaforma
- il riutilizzo degli stessi moduli in diversi contesti, senza introdurre ridondanza
- una gestione nativa delle utenze

A tal proposito, si sottolinea come risulti naturale la realizzazione di videate differenziate per categorie di utenza e rete, grazie alla creazione di menù, anch'essi modulari, nell'atto stesso dell'inserimento dei moduli.
A questo si va a collegare la struttura del database, che mantiene uno spazio di lavoro unico per l'autenticazione degli utenti del sistema.
Ben distinti da quest'ultimo, altri database vengono a crearsi man mano che si implementano ulteriori reti da monitorare; ognuno di questi manterrà al suo interno le strutture necessarie a soddisfare le esigenze dei vari moduli.
Questo, se da un lato aumenta lo spazio occupato, dall'altra permette di operare contemporaneamente su reti differenti, mantenendo distinti gli spazi di lavoro, aspetto che è stato ritenuto preferibile.

5.5.4 Gestione della macchina

L'implementazione ibrida per prodotto ha richiesto, come lavoro aggiuntivo, la gestione fisica del server. Per far fronte a questa necessità si è provveduto a realizzare un modulo ad-hoc, che permettesse, più che altro per scarsità di tempo, le operazioni essenziali per il controllo dell'hardware.
Volendo mantenere il prodotto quanto più blindato possibile, i comandi impartiti dovevano limitarsi a poche e ben limitate operazioni. Tale modulo attualmente si occupa:

- della gestione dell'alimentazione
- della gestione dell'interfaccia di rete
- del montaggio delle risorse di rete (per avere accesso ai file di log remoti)

La gestione dell'alimentazione si occupa dello spegnimento e riavvio della macchina; consente inoltre lo spegnimento programmato, molto utile in caso di interventi di manutenzione programmati.
La gestione di rete si occupa della modifica dei parametri di rete della macchina. Questa era stata creata inizialmente per poter operare al primo avvio dell'applicativo, ma diversamente non sarebbe potuta comunque mancare. Le ristrutturazioni di rete, per quando centellinate, sono eventi prevedibili in una infrastruttura di produzione: a tal proposito è impensabile non implementare una funzionalità del genere.
Discorso analogo vale per il "mount" di percorsi di rete, che in questo contesto permette, oltre all'analisi "al volo" di log esterni (vedesi paragrafo inerente al modulo "logloader"), il passaggio "a caldo" tra cartelle remote di archiviazione differenti.

Come già descritto nel precedente paragrafo, per tutte queste funzionalità si è preferito non dare i permessi necessari ai file di sistema interessati; si è preferito ricorrere ad una catena di chiamate i cui script terminali abbiano le credenziali limitate alla sola operazione da svolgere.

Capitolo 6

Sviluppi futuri

Per motivi legati alla durata dell'esperienza svolta, non è stato possibile giungere, come inizialmente sperato, a una versione completa del prodotto.
In queste righe si vuole stilare una linea di lavoro per portare a termine il progetto iniziato, con l'intenzione di renderlo operativo.

6.1 Rivisitazione dei moduli esistenti

Sono stati lasciati in sospeso i moduli di “squid.php”, “coovachilli.php” e “configuration.php”. Per il primo, essendo speculare a “iptrace.php”, sarebbe sufficiente effettuare le medesime modifiche ma, avendo a disposizione un po' più di tempo, sarebbe interessante procedere con una rivisitazione del modulo. Infatti buona parte del suo codice è copia di “iptrace.php”; a tal proposito sarebbe utile assimilare le righe in comune tra le due pagine al fine di evitare inefficienti ripetizioni.
Riguardo a “coovachilli.php”, è necessario rivederne la struttura; questi utilizza le righe provenienti direttamente dal servizio, andando contro gli obiettivi del progetto di ottenere informazioni esclusivamente dai log. La modifica da effettuare riguarda comunque le soli sorgenti d'informazione: sia per la visualizzazione dei moduli d'errore, sia per l'inserimento degli accessi nel database mysql; una volta ripristinate correttamente le sorgenti d'informazione, non si prevedono ulteriori ostacoli alla rimessa in funzione del servizio.
La pagina “configuration.php” invece necessita di notevoli rimanipolazioni. Questa infatti è stata pensata per modificare un unico file e garantire accesso completo a tutto il programma. La nuova pagina di cui l'applicativo necessita invece richiede una modulatizzazione tale da permettere l'accesso differenziato ai vari file di amministrazione periferici, senza la modifica del file di “core”, salvo a opera dell'amministratore. Di positivo c'è da sottolineare che le funzioni incaricate della modifica del file sono generiche, per cui tranquillamente riutilizzabili per il modulo ristrutturato. In corso d'opera sarebbe inoltre di grande utilità realizzare un modulo per il backup delle impostazioni correnti ed il relativo ripristino.
In secondo piano, vi sono delle migliorie che non hanno alta priorità d'intervento, ma che sono da tenere in considerazione. Tra queste un generatore di espressioni regolari che venga in aiuto a chi voglia usare il modulo “logloader.php”, di per sé perfettamente funzionante ma non di immediata comprensione. Il lavoro in questo caso si presenta ostico e lungo, per cui, data la sua non essenzialità, rimandabile senza timori in fasi d'intervento successive.
Più stringente invece la necessità di formulare uno standard per l'implementazione di moduli esterni o futuri e, di conseguenza, anche adattare la sezione ad essi relativa nel file di prima configurazione “start.php”.

6.2 Implementazione di nuovi moduli

Sono molteplici gli sviluppi che il prodotto potrà prendere, a seconda dei moduli che vi verranno aggiunti, ma in ogni caso risulterà essenziale assemblare quanto prima una pagina di **gestione delle utenze**. Questa probabilmente è la mancanza più grave al momento presente nel software, che dovrà presto andar colmata. L'opera non dovrebbe richiedere un grande dispendio di energie, in quanto buona parte dell'applicativo risulta già essere orientato a diversi utenti. Concisamente, il modulo dovrà permette di aggiungere le utenze ed i relativi permessi e interfacciare questi dati ai menù personalizzati, che sono già stati implementati. A semplificare il lavoro, anche il database è già strutturato per la pluri-utenza.
Altrettanto importante, anche se in buona parte pronta, è la pagina di **amministrazione del sistema**, che si occupa della manutenzione del server fisico in cui è ospitato il software. Al momento la pagina presenta la gestione dell'alimentazione e sono pronte e funzionanti le funzioni per le impostazioni di rete e il montaggio dei percorsi remoti. Risulta urgente aggiungere le poche righe di testo per rendere grafiche i richiami a queste funzionalità; per il futuro invece si raccomanda l'implementazione di un monitor di sistema, con il quale monitorare le risorse di calcolo e memoria.
Ulteriori migliorie consisterebbero nel dare accesso alla lista dei cronjob, per programmare e modificare le operazioni cicliche da far eseguere al server, e una accurata scheda di gestione delle interfacce di rete, così da rendere possibile la gestione multipla delle stesse con tutti i benefici conseguenti (ridondanza, trunk e load balancing, ecc..).
Infine è prevedibile che a breve si sentirà la necessità di un maggiore **controllo sul database** del sistema; a tal proposito risulterebbe utile un modulo per la sua gestione, in grado di effettuare l'export e l'import dei suoi salvataggi. Meno banale invece risulterebbe il dare la possibilità d'appoggio a database esterni; anche se concettualmente la modifica è relativamente semplice, è necessario studiare a fondo il problema, anche nell'ottica d'integrazione con una struttura centralizzata di controllo degli accessi.

6.3 Uno sguardo alla sicurezza

Il prodotto così assemblato è pensato per operare in un ambiente di rete protetto, senza particolari attenzioni per la sicurezza. Nell'ottica di offrire un sistema professionale però, sarà necessario evolvere l'applicativo. L'adozione di pagine in https e l'imposizione di politiche più stringenti per il server http garantirebbero una buona sicurezza senza richiedere un lavoro significativo. Inoltre l'utilizzo di un sistema operativo linux garantisce poche difficoltà nell'integrare la ricezione delle rige di log all'interno di un tunnel ssh. Da questo

si evince che il sistema è sulla buona strada per poter divenire un'alternativa sicura ed economica ad altri prodotti di produttori di certo più famosi ma, nondimeno, molto più cari.

6.4 Prospettive future

La modularità con cui si sta sviluppando l'applicativo permette di nutrire un certo ottimismo sulle sue future applicazioni. Anche quando verranno terminate le funzioni lasciate in sospeso, le possibilità che il prodotto potenzialmente può offrire sono molte.
In particolare, una delle aspirazioni a cui si ambisce è la realizzazione di un modulo, equipaggiato da un adeguato modello euristico, che riesca a segnalare anticipatamente eventuali anomalie decretate statisticamente "a priori"; in altre parole ci si aspetta che non venga semplicemente segnalato un guasto o un avvertimento, ma si vada a conformare, quando e se possibile, una sorta di macromodello comportamentale all'interno dei log generati, rendendo possibile il riconoscimento preventivo di anomalie o crash imminenti al di fuori dagli standard attuali.

Conclusioni

In questo documento si è voluto descrivere il lavoro svolto durante il periodo di tirocinio presso Ne-t by Telerete Nordest.
Sono state esposte le attività svolte, le difficoltà incontrate e il prodotto realizzato in questa esperienza.
Tale sistema è nato dal codice in uso presso la struttura ospitante, elaborato ed implementato fino a giungere a un differente applicativo che si pone come nocciolo di sviluppo per future espansioni.
L'obiettivo iniziale, ovvero l'ingegnerizzazione di una piattaforma di monitoraggio di file di log, si può dire raggiunto ma è necessario investire ancora del tempo sulla piattaforma per poter completare l'opera.
Nonostante il buon esito dell'attività, col senno di poi dato dall'esperienza maturata, alcune scelte implementative sarebbero state differenti. Queste avrebbero portato a un'ottimizzazione dei tempi di sviluppo ma, considerato il divario di nozioni tra l'inizio del tirocinio e la stesura di queste righe, ci si può ritener soddisfatti del lavoro svolto.
Date le ricche prospettive che il prodotto offre, ci si augura che altri possano continuarne lo sviluppo al fine di offrire al mercato una soluzione alternativa ed economica a quella offerta da altri noti, affidabili ma costosi marchi.

Appendice

In questo Appendice sono descritti i protocolli (e relative implementazioni) e gli strumenti utilizzati durante la stesura di questo documento.

A.1 Syslog

Syslog (abbreviazione di System Log) è uno standard per l'invio di messaggi di log in una rete IP. Il termine "syslog" viene utilizzato per indicare sia l'effettivo protocollo Syslog, sia per l'applicazione o la libreria che si occupa della spedizione e della ricezione dei messaggi di log.

A.1.1 Il protocollo

Syslog è un protocollo di tipo client/server. Il syslog sender invia un piccolo (al massimo di taglia 1 KB o 1024 caratteri) messaggio testuale al syslog receiver.
Quest'ultimo viene comunemente chiamato "syslogd" , "syslog daemon" o "syslog server" . I messaggi Syslog possono essere inviati sia via UDP sia via TCP. I dati vengono spediti in chiaro (cleartext); sebbene non faccia parte delle specifiche del protocollo stesso, è possibile utilizzare un wrapper in grado di fornire cifratura alla connessione tramite SSL/TLS. Per fare un esempio, un'applicazione Syslog viene spesso impiegata in simbiosi con tunnel.
Syslog viene tipicamente adottato per la gestione di sistemi di rete e per motivi di sicurezza ed affidabilità del sistema (security auditing).
Il protocollo è supportato da un'ampia varietà di dispositivi di rete su numerosi tipi di piattaforme; per questo motivo, Syslog può essere sfruttato per integrare informazioni di log provenienti da differenti sistemi, convogliandole in un'unica repository centralizzata.
Syslog nacque nel 1980 come parte del progetto Sendmail, ma la sua flessibilità gli permise ben presto di applicarsi efficientemente anche all'interno di altri progetti software . Il software Syslog (o meglio il demone syslogd) è stato per molti anni lo standard de facto per effettuare logging, sia in locale che in remoto, su macchine Linux e in generale con sistema operativo Unix-based oltre che su diversi dispositivi di altro genere. Recentemente Syslog è diventato un protocollo ed è stato standardizzato dalla IETF (Internet Engineering Task Force), il cui working group omonimo ha prodotto nel 2001 il documento RFC 3164. Un secondo documento, RFC 3195, rilasciato nello stesso anno riguarda la consegna affidabile (reliable delivery) nel protocollo Syslog.
La porta assegnata dalla IANA (Internet Assigned Numbers Authority) al protocollo Syslog è la 514. Bisogna prestare attenzione in quanto la porta registrata è relativa al solo protocollo UDP, mentre la 514/TCP è allocata al protocollo shell (cmd). Ad ogni modo, assicurandosi che la porta in questione non venga già impiegata da shell, nulla vieta all'istanza Syslog di utilizzare la 514 in TCP. La porta 601, riferita a syslog-conn (descritto nel RFC 3195), prevede l'utilizzo di entrambi i protocolli di trasporto.

Infine, la porta 6514 di TCP è associata all'estensione Syslog over TLS (standardizzato in RFC 5425).
Ulteriori informazioni ai siti http://www.syslog.cc/-ietf/-protocol.html e http://www.monitorware.com/common/en/articles/syslog-described.php .

A.1.2 Implementazioni

L'implementazione originaria, risalente ai tempi in cui Syslog non era ancora un protocollo, era nota come syslogd (http://linux.die.net/man/8/syslogd) ed era disponibile solamente per sistemi Unix-like (BSD e Linux). Versioni più recenti, anche queste open source, sono Rsyslog (http://www.rsyslog.com/) e Syslog-NG (http://www.balabit.com/network-security/syslog-ng/) : entrambe possono fare le veci del client così come del server.
In quanto a Syslog agents (client), sono numerosi gli apparati di rete che rispettano il protocollo: tra i maggiori produttori citiamo Cisco, Extreme Networks, Fujitsu, Huawei, IBM, NetGear, Symantec.

A.2 Radius

RADIUS (Remote Authentication Dial-In User Service) è un protocollo AAA (Authentication, Authorization, Accounting) utilizzato in applicazioni di accesso alle reti o di mobilità IP. Fu sviluppato presso la Livingston Enterprises Inc. nel 1991 come protocollo per i server d'accesso ai servizi di connettività (NAS - Network Access Server), e venne successivamente a far parte della suite di standard della IETF. RADIUS è attualmente lo standard de-facto per l'autenticazione remota, prevalendo sia nei sistemi nuovi che in quelli già esistenti.

A.2.1 Il protocollo

Il protocollo RADIUS nella sua interezza e nelle sue varianti è definito in numerosi RFC; tra questi i principali sono RFC 2865 (Remote Authentication Dial In User Service) e RFC 2866 (RADIUS Accounting), rilasciati entrambi nel 2000.
RADIUS è un protocollo che utilizza pacchetti UDP per trasportare informazioni di autenticazione e configurazione tra l'autenticatore e il server RADIUS.
L'autenticazione è basata su username, password e, opzionalmente, risposta a una richiesta di riconoscimento (una sorta di "parola d'ordine"). Se l'autenticazione ha successo, il server RADIUS invia le informazioni di configurazione al client, inclusi i valori necessari a soddisfare il servizio richiesto, come un indirizzo IP e una maschera di sottorete per PPP o un numero di porta TCP per telnet.
La figura A.1 mostra 4 tra le principali modalità di accesso alle risorse di rete con autenticazione basata su protocollo RADIUS. La quarta modalità, raffigurata in basso, è quella tipica nei casi di ISP che forniscano connettività wireless ai propri clienti registrati: questa soluzione fa uso di access point collegati direttamente ai NAS, i quali fungono da RADIUS client nella

comunicazione con il server di autenticazione.

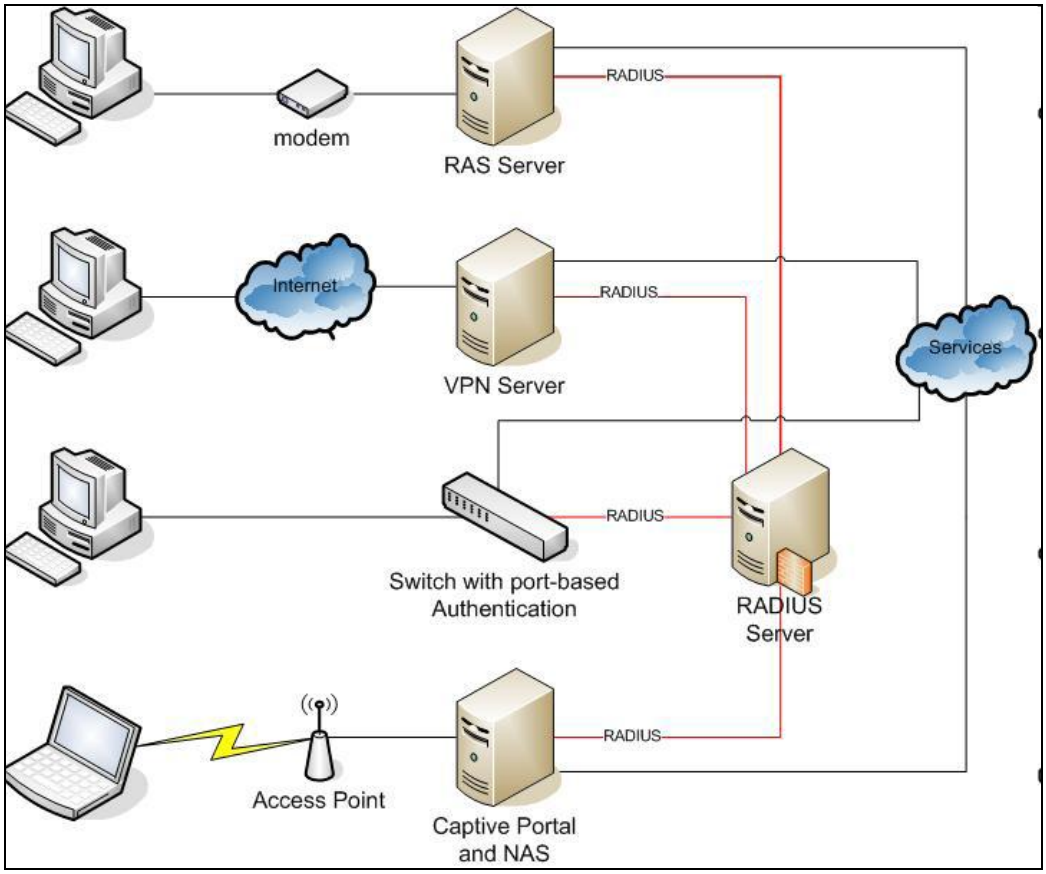

Figura A.1.. Autentificazione in Radius

Uno dei limiti del protocollo RADIUS è l'autenticazione basata esclusivamente su password: la password è trasmessa o in forma hash (utilizzando l'algoritmo di hashing MD5), oppure sottoforma di risposta a una richiesta di identificazione (CHAP-password).
Gli schemi di autenticazione supportati sono PAP, CHAP e EAP. L'Extensible Authentication Protocol (EAP) rende RADIUS capace di lavorare con una varietà di schemi di autenticazione, inclusi chiave pubblica, Kerberos e smart card. L'access point agisce da traduttore EAP-RADIUS tra il client wireless e il RADIUS server. Esso utilizza il protocollo EAP per comunicare con il client e il protocollo RADIUS per comunicare con il server RADIUS. L'access point incapsula le informazioni (come lo username o la chiave pubblica) in un pacchetto RADIUS che inoltra al server RADIUS.
Quando il server rimanda una delle possibili risposte (Access-Accept/Reject/Challenge), l'access point spacchetta il pacchetto RADIUS ed inoltra la risposta al client in un pacchetto EAP.
La RFC 2869 (RADIUS Extensions) specifica gli attributi opzionali da settare sui pacchetti RADIUS per indicare al server RADIUS che si sta utilizzando il protocollo EAP. Poiché il pacchetto EAP include un campo per specificare quale

metodo di autenticazione è in uso, il server RADIUS implementa l'autenticazione richiamando un'apposita procedura.

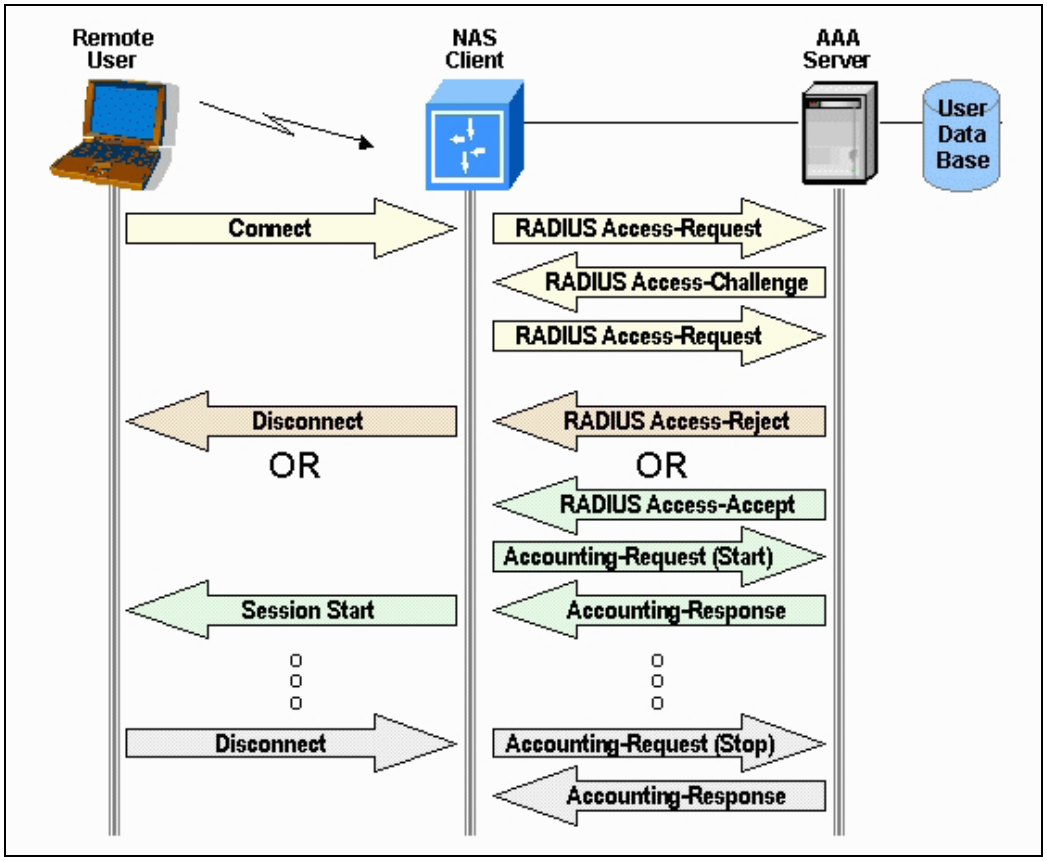

Figura A.2..Processo di autentificazione in Radius

La figura B.2 raffigura il flusso del processo con cui un utente si autentica nel protocollo RADIUS. Sono chiaramente distinguibili le 3 entità: utente, client NAS e server di autenticazione. Sottolineiamo che, tipicamente, una rete contiene un solo server di autenticazione (che gestisce il database RADIUS) e diversi client NAS, terminali a cui gli utenti si collegano direttamente.
La IANA ha assegnato le porte UDP 1812 a RADIUS Authentication e 1813 a RADIUS Accounting. Preme sottolineare che, precedentemente all'allocazione ufficiale da parte della IANA, venivano usate in maniera non ufficiale le porte 1645 and 1646 (per Authentication e Accounting, rispettivamente), che divennero così le porte impiegate di default da molte implementazioni Radius (sia client che server): per ragioni di backwards compatibility, in alcuni casi queste porte continuano tuttora ad essere adoperate.
Protocolli di autenticazione concorrenti sono Kerberos ed il recente Diameter.
Ulteriori informazioni sul protocollo sono reperibili presso http://en.wikipedia.org/wiki/RADIUS .

A.2.2 FreeRadius

Le implementazioni del protocollo, relativamente al server RADIUS, sono numerose: per una lista completa si rimanda a http://en.wikipedia.org/wiki/List_of_RADIUS_servers .
La soluzione più popolare ed utilizzata nel mondo è certamente FreeRadius (http://freeradius.org/).
L'implementazione open source (licenza GPL) del server RADIUS viene distribuita con l'inclusione di varie features:

- supporto completo per gli attributi definiti nelle RFC 2865 e RFC 2866;
- implementazione del protocollo EAP, compresi i sotto-tipi EAP-MD5, EAP-SIM, EAP-TLS, EAP-TTLS, EAP-PEAP, e Cisco LEAP EAP;
- supporto per attributi vendor-specific per circa un centinaio di produttori;
- supporto per diversi RADIUS clients, tra cui ChilliSpot/CoovaChilli, JRadius, SecureW2 EAP, Xsupplicant, e altri;
- una development library (con licenza BSD) per l'implementazione di client RADIUS;
- il modulo PAM per l'autenticazione degli utenti e l'accounting;
- un modulo per l'integrazione con Apache;
- uno strumento di amministrazione, chiamato dialupadmin e programmato in PHP, è presente nella distribuzione ed è accessibile all'utente come applicazione web.

FreeRadius è modulare, altamente flessibile nella configurazione, ad elevato livello di performance e scalabilità. Il software è stato testato e si è dimostrato scalabile in sistemi dell'ordine dei milioni di utenti. Le piattaforme supportate sono tutte le Unix-based e Windows. La documentazione è consultabile all'indirizzo http://wiki.freeradius.org/Main_Page.

A.3 CoovaChilli

CoovaChilli è un software open-source per il controllo degli accessi ad una WLAN.
Si base sul popolare ChilliSpot, progetto (abbandonato) di cui è erede: viene mantenuto attivamente da una web community, comprendente anche membri del progetto originario ChilliSpot.
L'applicativo è dotato di diverse funzionalità che lo rendono uno degli access controller e captive portal (anche noti come walled-garden environment) più utilizzati. I walled garden sono ambienti che controllano l'accesso degli utenti ai contenuti e ai servizi di rete: essi dirigono la navigazione all'interno di particolari aree, per permettere l'accesso ad un determinato insieme di risorse negando l'accesso ad altre risorse. Gli ISP possono stabilire che i propri utenti siano in grado di visitare alcune pagine web (within the garden) ma non altre (outside the walls).
Il captive portal offre il vantaggio di essere uno UAM (Universal Access Method), garantisce quindi la fruibilità da qualunque piattaforma, dispositivo o sistema operativo, e nel contempo non richiede nessun intervento lato utente. Esso prevede infatti l'utilizzo di un comune web browser piuttosto che un client

specifico per effettuare l'accesso alla rete.
Si appoggia a RADIUS per fornire l'accesso alla rete wireless, l'autenticazione al servizio e l'accounting delle risorse. Nel pacchetto software di CoovaChilli è presente, come parte integrante, il firmware CoovaAP (basato su OpenWRT).
CoovaAP è un'implementazione specifica e specializzata per hotspots, e svolge l'effettiva funzione di access controller per CoovaChilli.
La figura A.3 esemplifica il modello di rete adottato per sistemi basati su controllo degli accessi ChilliSpot/CoovaChilli e server di autenticazione RADIUS.

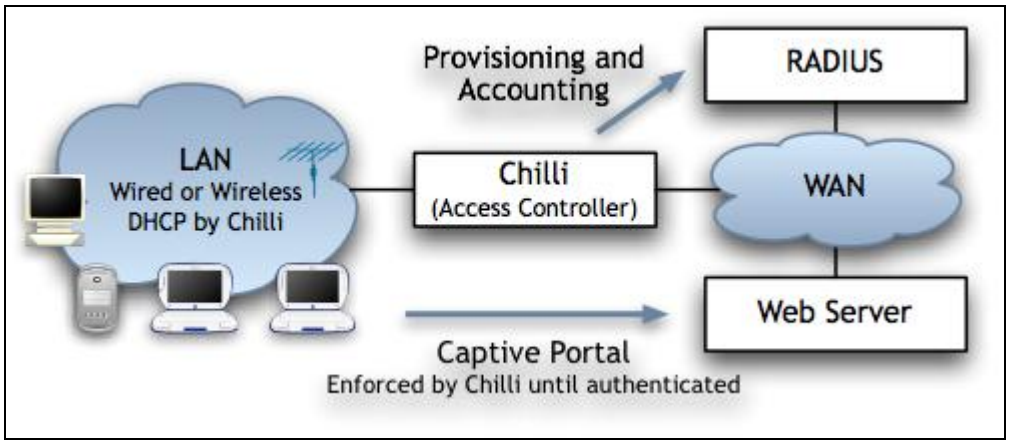

Figura A.3..Interazione tra CoovaChilli e Radius

Cenni sul funzionamento e sul processo
CoovaChilli assume il controllo dell'interfaccia Ethernet interna (eth1), per mezzo di un modulo del kernel chiamato vtun (abbreviazione di Virtual Tunnel, sito ufficiale: http://vtun.sourceforge.net): questo modulo implementa e gestisce un'interfaccia virtuale, cui associa il nome tun0. Il modulo vtun è impiegato per redirigere i pacchetti IP dal kernel a user-mode, in maniera tale che CoovaChilli è in grado di funzionare senza moduli kernel oltre a quelli standard.

Sull'interfaccia virtuale tun0, CoovaChilli imposta un server DHCP. Un client che si connetta a tale interfaccia si vedrà respingere tutti i pacchetti, almeno fino a quando esso non si sia autenticato ed abbia ricevuto l'autorizzazione: questa operazione si realizza per mezzo della login page di CoovaChilli, la quale agisce da supplicant per l'autenticazione al servizio. Quando un client non autenticato prova a connettersi ad una pagina web (sulle porte 80-http o 443-https), la richiesta viene intercettata da CoovaChilli e subisce un redirect ad uno script Perl chiamato hotspotlogin.cgi (servito da apache sulla 443-https).
hotspotlogin.cgi visualizza all'utente finale una pagina per l'inserimento dei parametri di autenticazione (tipicamente username e password).
Questi valori vengono poi inoltrati al server FreeRadius, che ne effettua il matching con le informazioni presenti nel suo backend database , secondo i protocolli di autenticazione PAP o CHAP, eventualmente proteggendo la comunicazione con cifratura SSL/TLS. A questo punto, l'utente viene rifiutato oppure autenticato dal server Radius, che istruisce hotspotlogin.cgi a presentare

al cliente, a seconda del caso, un messaggio di rifiuto oppure una pagina contenente tra le altre cose una notifica di successo ed un link per il logout dalla rete.
Recentemente, oltre alla versione basata sullo script CGI, l'implementazione del redirecting del captive portal può essere ottenuta anche per mezzo dell'interfaccia JSON (JavaScript Object Notation) che definisce un formato standard per lo scambio dei dati di autenticazione.
Ulteriori dettagli consultabili nella documentazione ufficiale del progetto: http://coova.org/wiki/index.php/CoovaChilli/Documentation.

A.4 Squid

Squid è un popolare software libero con funzionalità di proxy e web cache, rilasciato sotto la GNU General Public License. Ha una vasta varietà di usi, da quello di rendere più veloce un server web usando una cache per richieste ripetute, fornisce sia un servizio di cache per il web che per DNS e altri tipi di ricerche all'interno di reti con risorse condivise, e filtri sul traffico permesso. È stato primariamente sviluppato per piattaforme Unix-like.
Squid è in sviluppo da diversi anni ed è ormai considerato un'applicazione sicura e robusta. Supporta molti protocolli, ma è comunque primariamente un proxy HTTP e FTP. È inoltre disponibile supporto per TLS, SSL, Gopher e HTTPS.

Proxy Web La funzione di caching è un modo di salvare oggetti Internet richiesti (pagine web), è disponibile via HTTP, FTP e Gopher in un sistema più vicino al sito richiedente. Il browser può usare la cache di Squid locale come un proxy HTTP server, riducendo l'accesso ai server nonché il consumo di banda.
Questo è funzionale ai service provider. L'introduzione di server proxy introduce comunque anche questioni relative alla privacy dal momento che tutte le richieste che vi transitano possono essere salvate, si possono includere informazioni relative al tempo esatto, il nome e la versione ed il sistema operativo del browser che richiede la pagina.
Sul programma client (nella maggior parte dei casi un browser) può avere specificato esplicitamente il server proxy che si vuole usare o può usare un proxy senza altre specifiche configurazioni, in questo caso si parla di transparent proxy, nel qual caso tutte le richieste HTTP sono interpretate da Squid e tutte le risposte sono salvate. L'ultima menzionata è tipicamente una configurazione aziendale (tutti i client sono sulla stessa LAN) questo spesso introduce i problemi di privacy menzionati precedentemente.
Squid possiede alcune funzioni che possono aiutare a rendere anonime le connessioni, per esempio disabilitando o cambiando dei campi specifici nell'intestazione delle richieste HTTP. Che questi campi siano impostati o meno dipende dalla configurazione del server Squid che funziona da proxy. Le persone che richiedono pagine attraverso una rete che usa Squid in modo trasparente generalmente non sono informate sul fatto che le informazioni sono memorizzate in un registro. Per maggiori dettagli, si rimanda alla documentazione ufficiale: www. squid-cache.org/.

A 5 Espressioni regolari

Una espressione regolare (in inglese regular expression) è una sequenza di simboli che identifica un insieme di stringhe.
Gli insiemi che possono essere definiti tramite espressioni regolari sono anche detti linguaggi regolari, e coincidono con quelli generabili dalle grammatiche regolari e sono riconoscibili da automi a stati finiti.

Sebbene fossero state formalizzate già fin dagli anni quaranta, le espressioni regolari entrarono nel mondo informatico per la prima volta alla fine degli anni sessanta, in ambiente Unix: il primo editor di testo che implementava delle funzioni che ne permettessero l'uso fu una versione di QED scritta da Ken Thompson.
L'editor, dalla sua interfaccia a riga di comando, metteva a disposizione un comando chiamato global regular expression print, che successivamente fu reso un applicativo indipendente, grep.

Le espressioni regolari non ebbero grande diffusione ed utilizzo fino agli anni ottanta, quando fu inventato il linguaggio di programmazione Perl che permetteva nativamente l'uso di espressioni regolari. La versatilità del linguaggio, dovuta anche al fatto d'essere un linguaggio interpretato, ne permise l'utilizzo in svariate situazioni e favorì lo sviluppo del formalismo di Perl per le espressioni regolari, che diventò uno standard de facto.
La grandissima diffusione di questi strumenti spinse alcuni sviluppatori a implementare le espressioni regolari anche in altri linguaggi, a mezzo di librerie come PCRE o persino a svilupparne implementazioni native per alcuni linguaggi, come Java e tcl.
Ulteriori dettagli sono consultabili presso la pagina: http://en.wikipedia.org/wiki/Regular_expression

Ringraziamenti

Un profondo grazie ai miei genitori, per l'inesauribile pazienza e il sostegno che mi hanno offerto in questi anni.

Grazie al Prof. Filira, a Giuseppe e a tutto il personale di Ne-t per l'opportunità offertami e l'esperienza accumulata.

Grazie a Nena, Fra, agli amici e ai parenti che mi han dato una mano quando serviva.

P.S.
È fatta! ☺

Bibliografia/Sitografia

AA.VV. – PHP 5. Guida per lo sviluppatore (Linguaggi & programmazione)
AA.VV. – JavaScript: The Definitive Guide
L.Peterson, S.Dave – Reti di calcolatori
A.S. Tanenbaum - Computer Networks
R.Elmasri, B.Navathe - Sistemi di basi di dati
Szabo Karoly Albert - RACCOLTA ED ANALISI DI LOGS

http://en.wikipedia.org/wiki/System_integration
http://en.wikipedia.org/wiki/Regular_expression
http://en.wikipedia.org/wiki/RADIUS
http://en.wikipedia.org/wiki/FreeRADIUS
http://en.wikipedia.org/wiki/Captive_portal
http://en.wikipedia.org/wiki/Syslog
http://www.coova.org/
http://wiki.squid-cache.org/
http://www.monitorware.com/common/en/articles/syslog-described.php
http://www.dia.unisa.it/~ads/corso-security/www/CORSO-0304/Syslog/index.html
http://publib.boulder.ibm.com/tividd/td/ITWSA/ITWSA_info45/en_US/HTML/guide/c-logs.html
http://www.php.net/manual/en/index.php
http://php.html.it
httpd.apache.org/docs/
http://www.debian.org/doc/manuals/
http://www.linuxquestions.org/
http://tldp.org/LDP/abs/html/
http://web.mit.edu/rhel-doc/3/rhel-rg-it-3/s1-iptables-options.html
http://blog.chalda.it/guida-alla-sintassi-delle-espressioni-regolari-217.html
http://www.w3.org/TR/html401/
http://dev.mysql.com/doc/
http://www.w3schools.com
http://www.sudo.ws/sudo/man/1.8.2/sudoers.man.html
http://www.samba.org/

…