ALLEGATO 1

IL SISTEMA INFORMATIZZATO

IL SITO INTERNET

All'interno del portale del SEFAP (**figura 36**) è stato collocato un sottosito ad hoc per il progetto CHECK.

Questo sito è stato registrato nei principali motori di ricerca ed è possibile raggiungerlo oltre che da questi ultimi anche direttamente da un link nella home page del sito del SEFAP.
Cliccando sul logo CHECK è possibile entrare nella pagina descrittiva del progetto (**figura 37**).

Il sito del progetto, tranne una prima pagina iniziale che descrive lo studio, è completamente protetto da password. All'interno del sito CHECK ci sono infatti informazioni ed aree inerenti al progetto che riguardano unicamente lo studio e sono considerate riservate ai medici partecipanti.

La prima volta che il ricercatore accede al sito con login e password assegnate compila con i suoi dati la scheda di registrazione (**figura 38**) ed inserisce il numero di assistiti iscritti nella propria lista ASL (necessario per la randomizzazione).

Una volta che si è avuto accesso al portale, è possibile consultarne le due aree principali: "Progetto" e "Servizi" (**figura 39**).

FIGURA 36

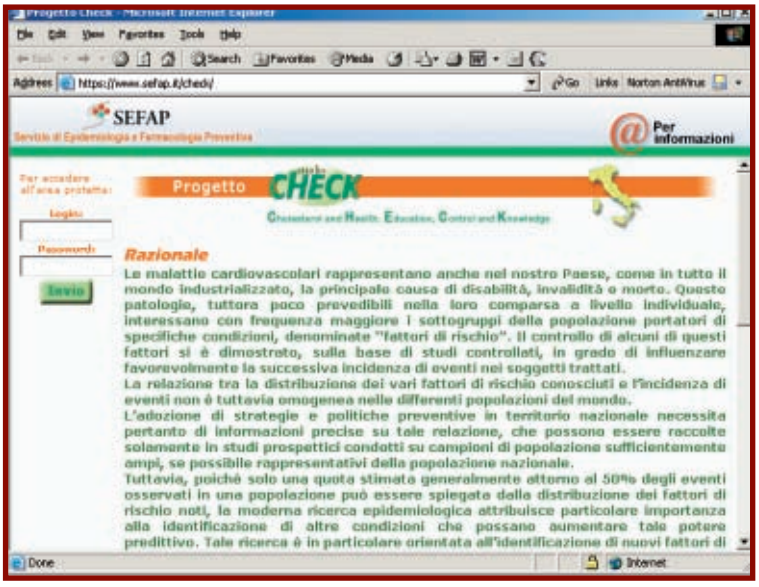

FIGURA 37

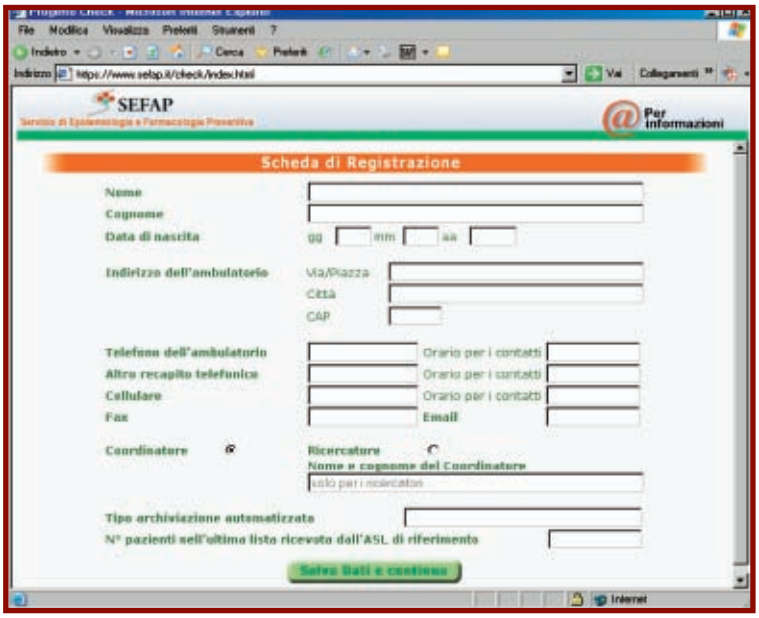

FIGURA 38

La prima contiene tutti i dati relativi al progetto, mentre la seconda riguarda i contatti e le utilità messe a disposizione degli aderenti al progetto. In particolare il menu "Progetto" contiene:

- informazioni sullo studio CHECK, cioè il protocollo dello studio, il foglio informativo ed il consenso informato e il questionario della salute;
- la scheda di rilevamento dati: consente il download del programma che permette di inserire i dati relativi al paziente;
- la scheda per la segnalazione di eventi: consente il download del programma che permette di comunicare gli eventi avvenuti a pazienti facenti parte dello studio;
- la cartella operatività.

Quest'ultima ha diverse funzionalità: la prima è quella di generare in modo randomizzato i pazienti che il medico dovrà arruolare, la seconda è quella di fornite una serie di informazioni dettagliate al medico su:

- definizione dei criteri di esclusione dei pazienti;
- descrizione dettagliata della visita;
- istruzioni per il prelievo e per la movimentazione dei campioni;

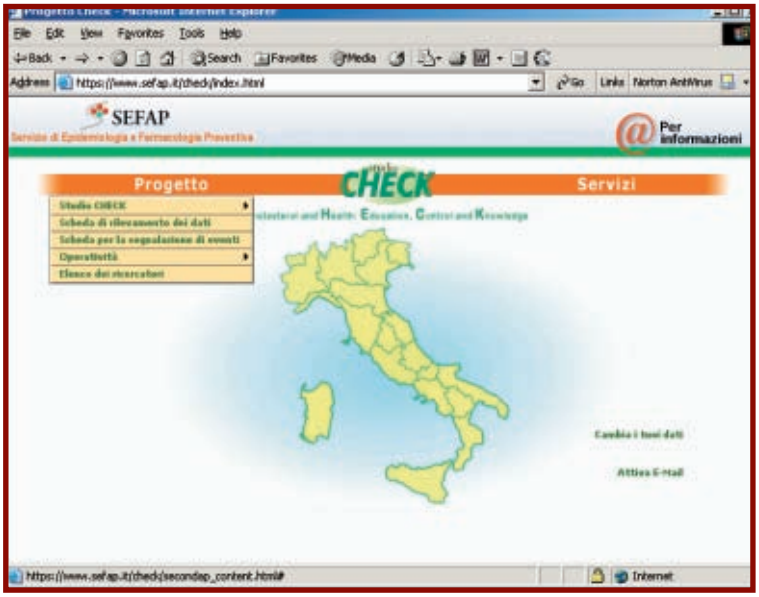

FIGURA 39

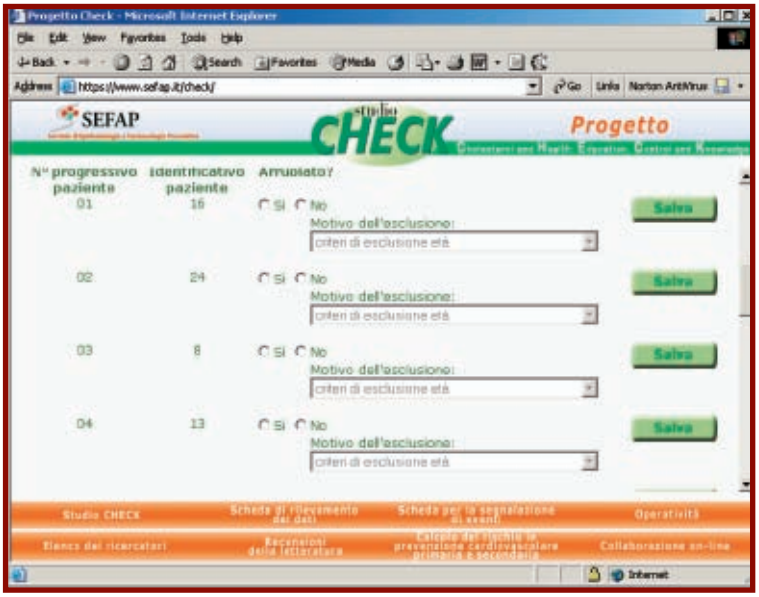

FIGURA 40

- istruzioni per la compilazione della scheda di rilevamento dati;
- istruzioni per la compilazione della scheda di segnalazione eventi;
- elenco dei ricercatori aderenti al progetto.

Come anticipato sopra, per ciascun medico, è disponibile nella pagina OPERATIVITÀ/MY FOLDER una lista di 16/32 numeri generati casualmente dal computer in base al numero di assistiti. Questa scheda è interattiva e consente di effettuare l'arruolamento dei soggetti in automatico, secondo una procedura codificata. Se il soggetto non risponde ai criteri di inclusione o non accetta di partecipare allo studio il computer genera un numero alternativo: per numeri pari avanza di una cifra nella lista, per numeri dispari retrocede di una cifra (**figura 40**).

Il software della scheda di rilevamento dei dati (CHECK-DATI), presente nel sottosito alla pagina "PROGETTO/scheda di rilevamento dei dati" può essere scaricato sul computer del ricercatore e compilato in locale. La prima pagina, relativa all'anagrafica, resta in locale (e non è quindi visibile da alcun operatore, a totale tutela della privacy del paziente), mentre viene trasmesso solo il codice del soggetto. Tutti i campi della scheda sono obbligatori, per cui non si può salvare la scheda se non sono stati compilati. Viene fornita anche un cartella cartacea, corrispondente alla scheda informatizzata, per quei ricercatori che non hanno il computer accessibile nel corso della visita. I dati raccolti vengono poi immessi nel formato elettronico e spediti via internet.

In **figura 41** è mostrata la prima schermata proposta dal software che è la pagina di login.

Al primo utilizzo del software, sarà necessario che il computer sul quale è installato il programma sia connesso ad Internet. CHECK-DATI avrà infatti la possibilità di controllare l'identità dell'utente (tramite login e password, le stesse con le quali si accede al sottosito) con il server on-line; tale operazione non verrà più ripetuta in futuro per quel login. È consigliabile utilizzare il programma disponendo di una connessione Internet attiva. In questo modo CHECK-DATI sarà in grado di consultare il database on-line, salvare le informazioni online ed, eventualmente, recuperare le informazioni su una visita inserita e salvata precedentemente anche da un altro terminale.

L'interfaccia del programma si divide in tre sezioni principali meglio illustrati nei successivi sottoparagrafi: il menu di sinistra; il navigatore; la finestra di compilazione dei dati (**figura 42**).

Il menu di sinistra prevede 8 pulsanti con le relative descrizioni; ad ogni pulsante corrisponde una diversa schermata. Le prime sei schermate (da "Identificazione del soggetto" a "Dati per la valutazione farmaco-economica") sono finestre di inserimento dati, mentre le ultime 2 ("Dati biochimici" e "Rischio coronarico assoluto") sono puramente informative, in quanto si auto-compilano con i dati provenienti dal Laboratorio Fleming e dalla cartella stessa. Il navigatore consiste di 4 pulsanti. I pulsanti freccia servono per spostarsi di pagina in pagina. L'ordine di spostamento seguito dalle frecce è lo stesso ordine riportato nel menu di sinistra.

Una volta compilate tutte e 6 le pagine, premendo il pulsante Invia si procederà al salvatag-

FIGURA 41

gio delle informazioni, sia sul computer locale (dell'user), che sul server del SEFAP (ammesso che sia attiva in quel momento una connessione Internet). Altrimenti i dati verranno salvati solo in locale e dovranno essere inviati al server successivamente. Il pulsante esci permette di ritornare alla pagina di Login.

La Finestra di Compilazione dei Dati propone una serie di campi obbligatori da compilare, differente di pagina in pagina.

In alcune pagine sono presenti dei database consultabili per fornire con facilità una risposta predeterminata:

- il database dei farmaci (aggiornato ogni 6 mesi) permette di inserire nella scheda i farmaci che il paziente sta assumendo o che si vogliono prescrivere, semplicemente digitando le prime quattro lettere e cliccando sul prodotto voluto; è quindi indispensabile specificare di seguito quante unità sono riferite al prodotto; gli inserimenti sono teoricamente infiniti;
- il database dei DRG;

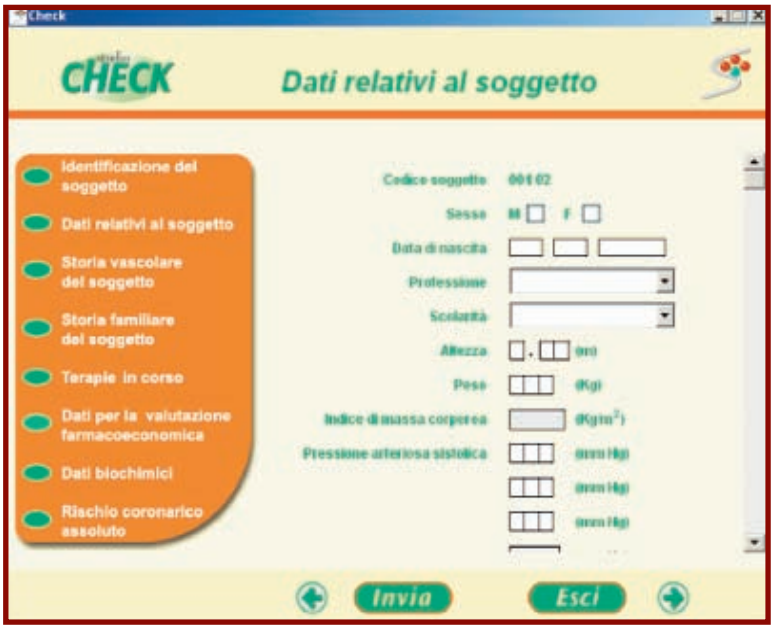

FIGURA 42

FIGURA 43

- il database dei più comuni parametri biochimici dosati;
- il database degli esami diagnostici relativi al cardiovascolare.

Quando si vogliono visionare i risultati degli esami biochimici ed il valore del rischio coronarico assoluto, oppure rivedere i dati della visita di un determinato soggetto, è sufficiente entrare nel programma, indicare durante la fase di login il relativo codice e portarsi sulle pagine desiderate. Il programma automaticamente recupera i dati dall'archivio e procede alla loro visualizzazione.

Prima di mostrare i dati della visita, il programma controlla se i valori della schermata "Dati biochimici" sono disponibili nel database; in questo caso, il programma provvede automaticamente a recuperarli. Nel momento in cui i dati biochimici sono stati scaricati, anche il valore percentuale del rischio, riportato nella pagina finale "Rischio coronarico assoluto" viene calcolato, utilizzando l'algoritmo di Framingham, e quindi visualizzato. Se il soggetto ha una storia personale di eventi car-

FIGURA 44

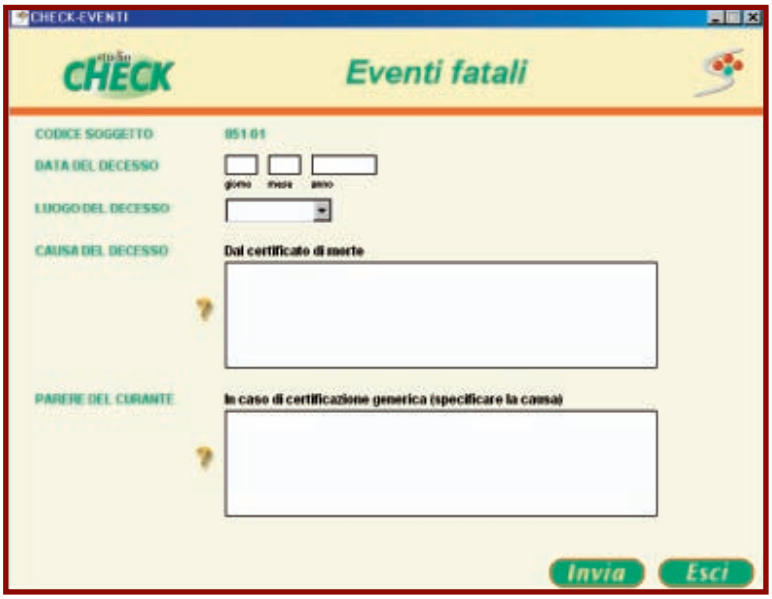

FIGURA 45

diovascolari, il rischio non viene calcolato, in quanto l'algoritmo è applicabile solo in prevenzione primaria.

Nelle ultime due schermate è disponibile anche la funzione "Stampa" che permette di stampare i referti dei dati biochimici di laboratorio e del rischio coronarico assoluto che possono essere consegnati al paziente per un'informazione più completa.

Il software della scheda di segnalazione degli eventi occorsi dopo il basale (CHECK-EVENTI), presente nel sottosito alla pagina "PROGETTO/scheda per la segnalazione di eventi", può essere scaricato sul proprio computer e compilato in locale. Il software è strutturato in modo del tutto analogo a quanto descritto per la scheda CHECK-DATI (**figura 43**). Come per i dati, nella prima finestra del programma si devono inserire cognome e nome del paziente (**figura 44**).

Si clicca su eventi fatali (**figura 45**) o eventi non fatali (**figura 46**) a seconda dell'evento da segnalare. Le due finestre contengono campi in parte obbligatori che il medico compila in relazione alle informazioni disponibili. Gli help aiutano il medico nella compilazione.

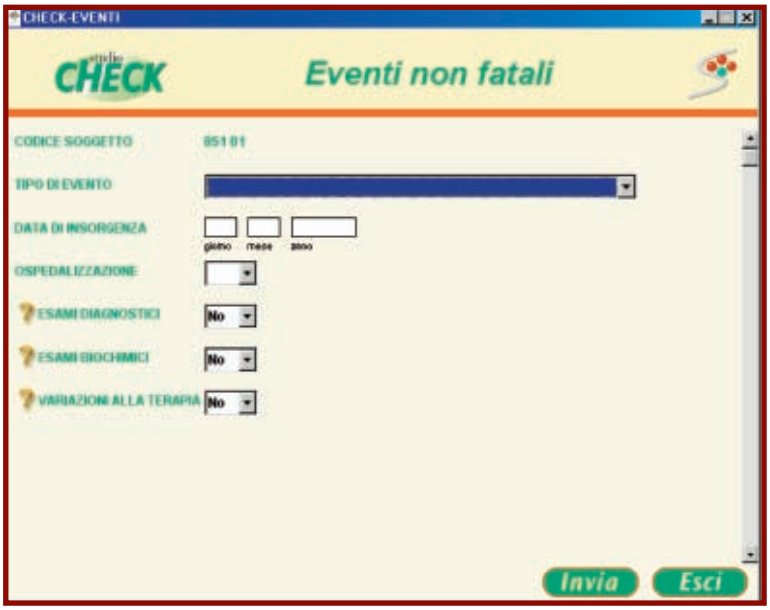

FIGURA 46

IL PORTALE

Il portale è stato realizzato con il linguaggio PHP (chiamato originariamente Personal Home Page Tools) poiché è risultato un linguaggio completo per le nostre esigenze, veloce sia in esecuzione sia da utilizzare. Sviluppato nel 1994 come semplice strumento incapsulato nell'HTML (server-parsed) per creare contenuti dinamici in pagine Web, PHP consente di sviluppare con facilità e flessibilità pagine Web, inoltre è un linguaggio molto diffuso soprattutto su Apache.

Con la versione 4.0, PHP ha acquisito un decisivo incremento di prestazioni e di una serie di funzioni normalmente richieste da diverse classi di applicazioni Web. Le caratteristiche aggiunte a questa nuova versione di PHP ne hanno aumentato l'usabilità e le performance.

Forse la caratteristica più importante e significativa di PHP è la possibilità di supportare una completa gamma di databases. Ad esempio sono supportati i seguenti database: Adabas D, Ingres, Oracle, dBase, InterBase, Ovrimos, Empress, FrontBase, PostgreSQL, FilePro, mSQL, Solid, Hyperwave, Direct MS-SQL, Sybase, IBM DB2, MySQL, Velocis, Informix e Unix dbm. Esiste anche un'estensione DBX database abstraction extension, che permette di usare in modo trasparente qualsiasi database da essa supportato. Inoltre PHP supporta *ODBC* (Open DataBase Connectivity), uno degli standard utilizzato per interfacciarsi con database, pertanto è possibile collegarsi con qualsiasi database che supporti questo standard.

PHP fornisce supporto per dialogare con altri servizi utilizzando i protocolli come LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (in Windows). È possibile anche aprire network sockets ed interagire usando qualsiasi altro protocollo. Inoltre supporta l'interscambio di dati complessi WDDX tra, virtualmente, tutti i linguaggi di programmazione web.

IL SERVER

Per quanto riguarda il server, è stato scelto il sistema operativo Linux, il web server Apache e il database relazione Mysql. Le motivazioni che ci hanno indotto a optare per tali scelte e le caratteristiche utili ai fini del progetto verranno analizzate nel dettaglio.

Secondo questo sistema, le informazioni giungono al server dai programmi client tramite il protocollo SSL e vengono inseriti dalla nostra applicazione nel database MySQL. Tali informazioni, per noi preziosissime in quanto motivo della realizzazione del progetto, risiedono nel server web ospitato presso la webfarm di Telecom Italia e giungono all'interno del sistema universitario grazie alle funzionalità di replica di MySQL. MySQL effettua la replica tramite una connessione non sicura e, quindi, per ovviare a questo problema, è stato creato un tunnel cifrato che unisce il sistema universitario al server.

Si è deciso di installare un tool visuale (Navicat della PremiumSoft) per la visualizzazione delle informazioni memorizzate nel database replica presente all'interno del terminale locale connesso al server e collocato presso il centro coordinatore (**figura 47**).

Così facendo si ha la possibilità di verificare in tempo reale l'andamento dell'inserimento dei dati di ciascun medico e di poter fornire assistenza a questi ultimi.

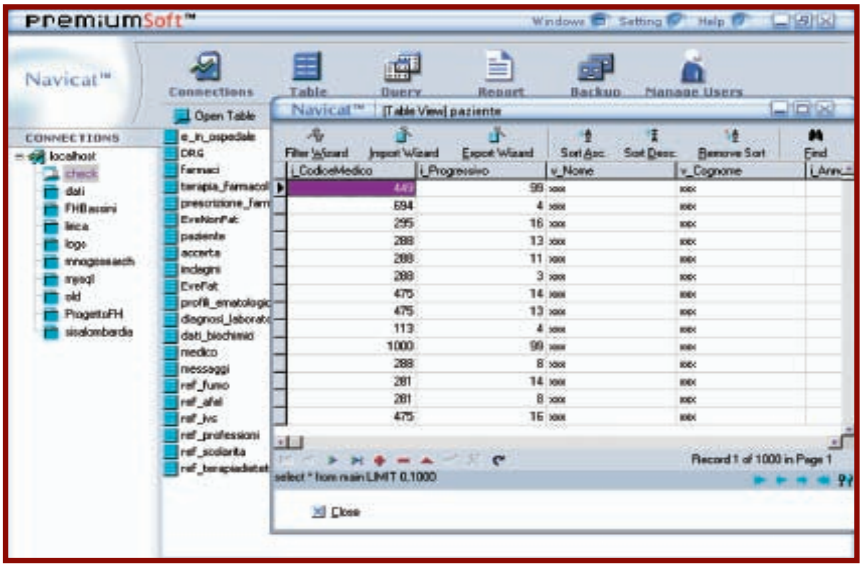

FIGURA 47

IL SISTEMA OPERATIVO

Per il sistema operativo la scelta è ricaduta sul sistema operativo Linux per molteplici ragioni.

Sicuramente una delle caratteristiche più allettanti è che Linux è gratis e lo sono quasi tutte le sue applicazioni, compreso Mysql e Apache. Inoltre è stabile, facilmente aggiornabile, meno soggetto di Microsoft Windows ai virus informatici in quanto è stato dotato, già in partenza, di difese più efficaci.

A differenza di Windows e di molti altri sistemi operativi commerciali, il codice sorgente di Linux non è segreto ed è liberamente modificabile. Questo consente all'utente non solo di avere il controllo assoluto e totale su cosa Linux contiene, ma anche di evitare orpelli inutili, sorprese sgradite e soprattutto occhi indiscreti. Quando il codice sorgente è aperto alla verifica da chiunque si parla di software open source, mentre i programmi che non pubblicano il proprio codice sorgente si chiamano closed source. Di un programma open source è quindi possibile valutarne la qualità e tale controllo prende varie forme. Al livello più semplice, consente di capire e modificare il funzionamento di un programma al fine di adattarlo meglio alle proprie necessità, per tradurlo in italiano oppure per eliminarne le parti inutili. A livelli più complessi permette all'utente esperto di verificare che nel programma non siano state racchiuse istruzioni pericolose che consentano intrusioni ostili o violino la sua privacy.

Il timore di funzioni-spia occultate nei programmi non è da sottovalutare in quanto nel passato si sono verificati casi simili con programmi di società di software molto importanti, che raccoglievano più o meno segretamente informazioni personali e commerciali inerenti ai loro utenti. Se il codice sorgente è pubblico, invece, il programmatore più difficilmente non tenta di usare routine dannose, in quanto non potrebbe rimanere nascosto, ma immediatamente scoperto.

Una delle accuse più frequenti a Linux è che non è dotato di assistenza tecnica cui rivolgersi e di documentazione. In realtà non è così: assistenza tecnica e documentazione esistono, sono solo offerte in forma diversa da quella cui ci hanno abituati i sistemi operativi commerciali. Nei programmi commerciali l'assistenza tecnica è spesso compresa nel prezzo del prodotto, sia che venga o meno utilizzata; in Linux si paga a parte e soltanto a fronte di necessità.

La migliore assistenza tecnica per Linux, comunque, la fanno gli utenti stessi. Nell'ambiente Linux vige il principio del libero scambio di conoscenze, per cui quando un utente risolve un problema, ne pubblica la soluzione e in questo modo chiunque altro avrà in futuro lo stesso problema saprà come affrontarlo.

Questa prassi ha portato all'accumulo di una massa enorme di documentazione gratuita ed estremamente specifica. Per attingervi è sufficiente mettere le parole chiave corrette in un motore di ricerca di Internet. Tale massa di documentazione è a volte addirittura sovrabbondante ed è facile perdervisi, ma più in genere raggiunge il suo scopo. Linux stesso, comunque, contiene centinaia di megabyte di file di documentazione, in buona parte tradotta anche in italiano. Nel caso in cui non fosse sufficiente, e se anche Internet non è d'aiuto, è possibile ricorrere ai Linux User Group, cioè libere associazioni di utenti Linux presenti in ogni città d'Italia le quali sono praticamente gratis.

IL WEB SERVER ED IL PROBLEMA DELLA SICUREZZA

La sicurezza è un argomento essenziale per gli amministratori di server, sia che siano connessi a Internet che ad intranet. I server Web sono bersagli molto attraenti per chi mira a compromettere la sicurezza di un sistema per rubare informazioni in esso contenute. L'argomento della sicurezza è molto ampio, la letteratura è molto estesa; in questa sede ci limitiamo a degli accenni.

La maggior parte delle organizzazioni che mantengono dei server Web è solita dividerli in due gruppi separandoli da un firewall. I server Web interni (rispetto al firewall) sono quelli destinati ad essere acceduti da parte degli host interni all'organizzazione; mentre quelli esterni possono essere acceduti da Internet. Il modo più sicuro di proteggere un server Web da un accesso indesiderato è quello di controllare che non vi si possa accedere tramite Internet. Questo procedimento è perfetto per i server Web intranet, ma va contro la ragione stessa d'esistere della maggior parte dei server Web, in quanto sono sistemi pubblici d'informazione o server di commercio Internet.

La cosa migliore da fare per proteggere un server Web da intrusioni è difendere le risorse private con alcuni meccanismi che forzino gli utenti a presentare delle credenziali di identificazione. Le misure di sicurezza che si possono attuare per proteggere un server Web sono molteplici, ma la più importante è assicurarsi di conoscere chi si sta connettendo identificando l'utente (autenticazione) e stabilendo delle regole che governino il suo agire.

Secure Socket Layers (SSL) supera il livello di base dell'autenticazione e crea una connessione cifrata con le applicazioni per permettere lo scambio sicuro di informazioni importanti. Dando agli utenti un senso di sicurezza e fiducia nella privacy dei server Web, SSL è stato una componente vitale nel fare del commercio elettronico una realtà.

I moduli che sono responsabili del controllo dell'accesso alle risorse Apache sono implementati

molto semplicemente. Nonostante la complessità di un modulo per il controllo dell'accesso, il suo fine è di considerare una singola richiesta HTTP e restituire ad Apache uno di due valori: la risposta sarà pari a "OK" se la richiesta deve essere soddisfatta; "FORBIDDEN" se deve essere negata. I moduli responsabili per la salvaguardia delle risorse protette in Apache si applicano nel corso di tre fasi del ciclo di elaborazione della richiesta:

- *Controllo dell'accesso*. Durante questa fase standard del ciclo di elaborazione della richiesta un modulo può negare l'accesso sulla base di informazioni fornite dalla richiesta HTTP o dal pacchetto IP che la contiene. Il solo modulo esaminato in questo capitolo e operante in questa fase è mod_access responsabile dei permessi e dei rifiuti di accesso ai client sulla base di un indirizzo di rete da cui ha origine la richiesta.
- *Autenticazione*. Ogni modulo chiamato durante la fase di verifica deve controllare l'identità del richiedente sulla base di credenziali presentate dall'utente. Queste credenziali possono essere elaborate come un certificato digitale e magari contrassegnate da una riconosciuta autorità di certificazione. La credenziale più comune è la semplice coppia di "nome/password" fornita dall'utente tramite una finestra di dialogo standard che appare in un browser Web, oppure riempita da dati in cache.
- *Autorizzazione*. Quando viene richiamato un modulo nella fase di autorizzazione, l'identità dell'utente si presume già nota, e il modulo controlla l'informazione di controllo d'accesso (quali le liste di controllo d'accesso), per determinare se l'utente abbia o meno il permesso per accedere alle risorse che richiede. Anche in questo caso, la risposta del modulo deve essere "OK" o "FORBIDDEN".

La maggior parte dei moduli che restringono l'accesso sulla base dell'identificazione dell'utente forniscono gestori per entrambe le due ultime fasi, prima identificando l'utente, poi determinando se quell'utente, una volta identificato, abbia il permesso di accedere alla risorsa richiesta.

Il modulo mod_access funziona durante la fase di controllo dell'accesso del ciclo di richiesta, per restringere l'accesso ai client sulla base dell'indirizzo IP da cui la richiesta ha origine.

Restrizioni sulla base dell'origine dei clienti

Il mezzo più semplice di restringere l'accesso è fornito dal modulo standard mod_access, che funziona nello stadio di controllo dell'accesso del ciclo di richiesta Apache. In questa fase, Apache non è a conoscenza della richiesta dell'utente e non ha ancora esaminato nessuna delle intestazioni HTTP. Durante la fase di controllo dell'accesso, mod_access viene usato per determinare se la richiesta potrà essere soddisfatta sulla sola base del nome dell'host o dell'indirizzo IP del client richiedente. La funzione di questo modulo è utile e molto semplice. Le direttive fornite dal modulo funzionano in un contesto a directory, (un container <Directory> o un file .htaccess). Le restrizioni di accesso imposte da queste direttive si applicano a tutte le risorse nella directory non possono essere usate per restringere l'accesso a file specifici. Nella maggior parte dei casi, mod_access è usato per negare accesso a client che si connettono da posizioni, (indirizzi IP), specificatamente proibite o assenti da una lista di indirizzi dichiaratamente approvati.

Restrizioni basate sull'identificazione dell'utente

È facile configurare restrizioni d'accesso sulla base dell'origine della richiesta, cioè dell'indirizzo IP, tecnica utile quando si vuole garantire solo a qualche host, o sottorete, l'accesso a determinate risorse, (quali le pagine di status del server). Questo approccio, tuttavia, non è molto flessibile. In primo luogo l'accesso si basa sull'indirizzo di rete del client, cosa che non è sempre sotto controllo ed è soggetto a mutamenti. Nella maggior parte dei casi ciò è indicato se si vuole restringere l'accesso solo a host sotto il proprio controllo.

Il protocollo HTTP descrive due tipi di autenticazione che dovrebbero essere supportati da tutti server e client Web. Il primo di questi è di gran lunga il più diffuso, è l'autenticazione HTTP di base che coinvolge lo scambio non cifrato di informazioni di autenticazione utente consistenti in una combinazione di "nome utente/password". Lo standard HTTP descrive anche una seconda forma di autenticazione dell'utente, chiamata autenticazione digest, che funziona in modo analogo, ma utilizzando un meccanismo per cifrare le credenziali dell'utente prima della trasmissione ed essenzialmente eliminando la minaccia che le password in transito vengano intercettate.

Il metodo standard di Apache per l'autenticazione HTTP conserva le credenziali dell'utente in file di testo. L'uso di file di testo per conservare le informazioni sull'utente pone degli handicap prestazionali che diventano sensibili quando i dati sono molti. La maggior parte dei siti che usa l'autenticazione HTTP impiega database di qualche genere per conservare le informazioni sull'utente.

Secure Sockets Layer

I dati trasmessi fra server Web e browser non sono al riparo da sguardi indiscreti, ed esiste la possibilità che qualche malintenzionato dotato di competenze tecniche adeguate possa spiare i

pacchetti di rete che i due si scambiano, leggendo informazioni importanti e private. Un secondo problema si evince dal fatto che il server Web ha un grado accettabile di sicurezza sull'identità dell'utente client (tramite autenticazione) mentre il client non ha modo di determinare se il server Web sia quello giusto.

È noto il caso di hacker che hanno simulato alcuni server su Internet, nel tentativo di apparire sotto ogni aspetto host legittimi e di catturare informazioni che l'ignaro utente era convinto di inviare a tutt'altro server.

Alla metà degli anni novanta, Netscape Communication ha sviluppato uno schema per eliminare entrambi i punti deboli nel modello di sicurezza del Web. Il protocollo Secure Sockets Layer (SSL) fornisce un canale riservato di comunicazione fra server Web e browser e assicura ai client che il server Web cui sono connessi è quello giusto tramite l'uso di certificati di autenticità a firma digitale. SSL si basa sulla crittografia, in modo particolare sulla crittografia a chiave asimmetrica, in cui la chiave usata per cifrare i dati è diversa da quella usata per decifrarli, ma matematicamente collegata.

I certificati usati da SSL sono utilizzati per comunicare la chiave pubblica. I certificati sono emessi da organizzazioni note come autorità di certificazione. Queste ultime hanno il compito di certificare che il titolare del certificato è proprio colui che dichiara di essere. Le autorità certificanti sono come gli "escrow server" in un senso: ognuno deve fidarsi di loro perché l'intero schema possa funzionare.

Non c'è ragione per cui una grande azienda non possa agire come certificatore di se stessa, ed è questo che molte fanno, allestendo complicate gerarchie di server di certificato che assicurano la privacy delle loro comunicazioni e dei loro trasferimenti di dati interni e anche, in alcuni casi, con i partner.

IL DATABASE

MySQL è un sistema di gestione di database relazionali dotato di tutte le funzionalità tipiche delle basi di dati, è molto stabile e ha saputo dimostrare le proprie qualità nel tempo.

MySQL è un server basato sul multithreading: quando viene stabilita una connessione con il server, il programma al suo interno crea un thread per gestire le richieste di ogni client, garantendone un'elevata velocità. A tutti gli effetti, ad ogni client connesso ad un server MySQL, viene assegnato un thread dedicato.

MySQL è inoltre completamente conforme allo standard ANSI SQL92, nonché a tutti gli standard emanati dall'*ANSI* (American National Standard Institute).

La portabilità è un'altra caratteristica di rilievo di MySQL ed è stata utilizzata in quasi tutte le piattaforme. Questo significa che per utilizzare MySQL non è necessario cambiare la piattaforma impiegata; qualora si intendesse passare ad un'altra piattaforma, è molto probabile riuscire a reperirne una nuova, adeguata ed adattata alle nuove esigenze.

MySQL dispone, inoltre, di numerose *API* (Application Programming Interface), tra cui quelle relative a Perl, TCL, Python, C, C++ e ODBC, quindi è possibile utilizzarlo indipendentemente dall'ambiente scelto da un'azienda.

MySQL offre una velocità e una flessibilità che nessun altro database della stessa classe eguaglia. È, inoltre, perfettamente in grado di operare con architetture differenti, può essere impiegato sia all'interno di un'architettura rigorosamente client/server, sia come database autonomo. Con la crescente necessità di memorizzare e accedere a grandi quantità di dati è aumentato anche il bisogno per i motori di database di recuperare ed elaborare le richieste in modo rapido ed efficiente. Si è dato così il via alla corsa al motore di database più veloce. Contemporaneamente sono cresciute le esigenze di maggiore funzionalità: gli amministratori di database avevano infatti bisogno di altri strumenti e di modi più efficienti per gestire la grande quantità di dati che dovevano memorizzare. Il problema, però, era che l'aggiunta di caratteristiche e funzioni ad un database portava come diretta conseguenza un degrado delle prestazioni: ci si trovava di fronte ad un sempre maggiore rallentamento della base di dati. MySQL ha cercato di ovviare a questo inconveniente per offrire il numero massimo di funzioni senza sacrificare la velocità.

Le alte performance di MySQL potrebbero far pensare ad una riduzione delle funzionalità, è stato tuttavia mantenuto un alto rapporto tra funzionalità e velocità. L'approccio utilizzato è diverso rispetto agli altri motori di database della stessa classe: sono state privilegiate le funzioni essenziali, lasciando a margine quelle opzionali. Nel prossimo futuro MySQL includerà molte delle funzioni già offerte dai database di fascia alta, continuando a mantenere gran parte della sua velocità. Gli sviluppatori lasceranno che siano gli amministratori di database a decidere se desiderano utilizzare il set di funzioni più avanzate, sacrificando la velocità.

Alcuni database usano una forma di SQL chiamata Transact-SQL o T-SQL (Microsoft SQL Server and Sybase Adaptive Server). Questa forma di SQL estende il normale linguaggio SQL aggiungendo la capacità di utilizzare strutture di controllo. Tali funzionalità possono esser incluse in MySQL con la compilazione e l'inclusione di opportuni moduli.

PHP ha diverse funzioni sofisticate per l'interfacciarsi con MySQL. Si osservi che cosa accade quando un client invia a un server web predisposto per PHP una richiesta in cui si verificherà un'interazione con un database MySQL:

- Il server riceve e legge la richiesta proveniente dal browser del client.
- Il server individua la pagina richiesta sul server.
- Il server esegue le istruzioni contenute nel codice PHP incorporato.
- PHP interroga il server di database MySQL database tramite un'API e compila il risultato.
- Il server web invia la pagina risultante al client.

PHP prevede una nutrita dotazione di funzioni per l'interfaccia con i server di database MySQL. È possibile creare potenti applicazioni ricorrendo a un semplice gruppo di esse.

La sicurezza dei database è un componente essenziale: è il sistema di sicurezza che protegge i dati, salvaguardandoli da eventuali tentativi di violazione da parte di malintenzionati. La sicurezza protegge anche i dati dagli utenti inesperti che a volte potrebbero involontariamente eliminare dei record. Come misure di protezione contro questi tipi di incidenti è possibile imporre un livello di sicurezza che impedisca agli utenti manipolare informazioni contenute all'interno del database.

La sicurezza gioca un ruolo fondamentale in qualunque applicazione cui possono accedere contemporaneamente più utenti. MySQL ha degli ottimi strumenti di gestione della sicurezza: è un sistema molto flessibile, permette di assegnare ai potenziali utenti diversi livelli di accesso che vanno dalla possibilità di connettersi da una macchina specifica come utente specifico, fino all'accesso completo come amministratore da un computer qualunque. È responsabilità di chi gestisce il database decidere quanto rigida debba essere la sicurezza.

MySQL memorizza tutte le autorizzazioni e i privilegi nel database *mysql*, è possibile eseguire query sulle tabelle di sistema proprio come con qualunque altra tabella. A queste tabelle si fa collettivamente riferimento con il termine di tabelle *grant*.

Esiste una gerarchia di sicurezza nel sistema di database di MySQL. Quando un utente si connette a un database MySQL il programma prima cerca nella tabella *user* per vedere se trova una corrispondenza con il nome di host, il nome utente e la password, se questa c'è, l'utente può accedere al sistema.

Quando l'utente emette una query relativa al database, MySQL controlla innanzi tutto la tabella user alla ricerca dei privilegi dell'utente. Se questa tabella non ne contiene, la ricerca si sposta alla tabella db, dove ancora una volta MySQL cerca una corrispondenza con il nome di post, il nome utente e il database. Se tale corrispondenza esiste, verranno analizzati i privilegi della persona in questione. Se quest'ultima non possiede tutti i privilegi necessari per emettere la query, MySQL cercherà nella tabella tables_priv e poi in columns_priv le autorizzazioni necessarie per eseguire il comando. Se non trova alcuna autorizzazione, viene generato un errore. Tutto ciò avviene ogni volta che si esegue una query in MySQL.

Come si può notare, esistono due punti di controllo: il primo è la verifica della connessione, mentre il secondo riguarda la verifica della richiesta. Tali punti offrono un ambiente più sicuro per il proprio database, perché un utente che può connettersi al database potrebbe non essere in grado di non fare nulla una volta entrato. Ciò offre un muro di protezione molto solido contro i possibili tentativi di violazione da parte di intrusi e protegge il database dagli utenti che potrebbero involontariamente creare problemi.

La verifica della connessione ha luogo nel momento in cui si cerca di connettersi a MySQL. Qualunque connessione ha bisogno di un nome utente, una password e un nome di host. È indispensabile il nome della persona che cerca di connettersi e la password è uno strumento di verifica aggiuntiva che garantisce che la persona che si sta connettendo è veramente chi afferma di essere. Il nome di host è, invece, il nome del computer da cui l'utente si sta connettendo. MySQL può limitare non solo la connessione di una persona, ma anche quella di un particolare host. È persino possibile consentire o negare l'accesso da interi domini, se necessario. Tutto ciò contribuisce a creare un luogo molto sicuro per i dati.

Un altro vantaggio è la garanzia dell'integrità dei dati: se una colonna alfanumerica deve contenere solo caratteri alfabetici, si potrebbe utilizzare un vincolo di verifica per assicurarsi che in essa non vengano mai inseriti numeri. Ciò permette di rimuovere il codice dall'applicazione client e collocarlo sul server.

Il maggior inconveniente dell'impiego dei vincoli è l'aggiunta di un carico notevole, in particolare nell'inserimento e aggiornamento dei record: il sistema deve rallentare, controllare i vincoli e poi eseguire le operazioni.

MySQL non supporta vincoli di integrità, perché sovraccaricano il sistema e lo rallentano. Il compito è lasciato nelle mani dello sviluppatore e dell'amministratore che hanno la responsabilità di assicurare l'integrità dei dati e che le relazioni tra tabelle siano imposte. Se si scrivono buone applicazioni per l'interazione con il database e lo schema è chiaro e semplice da capire, la necessità di avere vincoli è minima.