Single Sign-On con Kerberos e LDAP

Giuseppe Paternò

Single Sign-On con Kerberos e LDAP
Una soluzione per ambienti eterogenei

Autore ed Editore: Giuseppe Paternò

ISBN 88-901141-1-8

Edizioni:

Luglio 2004: Prima edizione

SOMMARIO

INDICE DELLE FIGURE

PREFAZIONE

Ero stanco di inserire tutte le volte utenza e password sulle macchine che amministravo, sia di casa che nella Intranet per l'azienda per cui lavoro. Ho visto che queste esigenze sono simili a quelle di molti dei miei clienti che devono quotidianamente amministrare centinaia di macchine e mi sono chiesto: possibile che non ci sia un metodo sicuro per poter effettuare un "Single Sign-On" vero e proprio? Tutti parlano di avere un "Web Single Sign-On" o qualcosa di simile, scoprendo poi che non è nient'altro che la centralizzazione di utenze su di un repository LDAP, o peggio su un database. Voglio autenticarmi una sola volta con il sistema, senza ripetere l'operazione di immettere utenza e password all'infinito.

Durante le mie elucubrazioni mentali, il mio pensiero è andato ad una famosa azienda di Redmond, sì avete indovinato: Microsoft. Ebbene, loro sono riusciti ad effettuare un Single Sign-On vero e proprio: una volta autenticati ad un dominio, riescono a collegarsi a risorse di rete senza più immettere utente e password, inclusi i siti Web che girano sotto Internet Information Server (IIS). Come ci sono riusciti ?

Coloro che mi conoscono sanno del mio "attaccamento" alla riga comandi (Unix ovviamente), ma sanno che non disdegno di "vedermi intorno": è inutile fare delle guerre di religione, non esiste mai il "tutto giusto" o "il tutto sbagliato", è solo una questione di gusti. Molto spesso rifiutiamo le idee degli altri nella nostra convizione di essere "dalla parte della ragione", ma ignoriamo il fatto che altri possono avere buone idee da prendere in considerazione, o anche (se vogliamo essere un po' "superiori") da migliorare.

Quello che volevo realizzare era un modello simile a quello adottato da Microsoft per la realizzazione di Active Directory, che però si applicasse non solo alle workstation Windows, ma soprattutto ai server e workstation Unix, incluso i servizi erogati (principalmente web, ma anche accesso di terminale, posta elettronica, ecc..)

Lo scopo di questo libro è quello di documentare la realizzazione in laboratorio di una soluzione di Single Sign-On. Non si tratta di una guida all'installazione, nè un manuale di Kerberos o LDAP, piuttosto si tratta di una descrizione di un modello architetturale. Oltre alla praticità di questo modello, ne vanno anche considrati gli aspetti legali nell'ottica della redazione del Documento Programmatico sulla

Sicurezza (D.P.S.), reso obbligatorio dal Decreto Legislativo 196/03. In particolare, sia l'Articolo 34 sia l'Allegato B si riferiscono ad alcune misure minime previste, quali ad esempio l'autenticazione informatica, l'adozione di procedure di gestione delle credenziali di autenticazione e l'utilizzazione di un sistema di autorizzazione. In questo senso, l'impiego di un sistema di single sign-on con Kerberos e LDAP può essere una metodologia per rispondere alle misure minime richieste per legge, centralizzando le utenze in un unico repository valido per tutte le piattaforme.

Buona lettura.

Giuseppe Paternò

Ringraziamenti

Un grazie enorme va a mia moglie Maria per avermi sopportato pazientemente mentre dedicavo il mio tempo ai computer e soprattutto per incoraggiarmi nel continuare le ricerche nel campo informatico. Grazie anche a Gabriella Cattaneo, che mi ha letto e corretto pazientemente questo libro. Un saluto va al mio amico Luca Sciortino, decisamente più "fulminato" di me, ed ai miei genitori. Grazie a tutti coloro che hanno contribuito alla realizzazione, in particolare Silvio Danesi.

L'autore

Giuseppe Paternò ha conseguito la certificazione CCNP ed è membro di IEEE e della Italian Linux Society. La sua passione ha spinto Giuseppe ad esplorare fin da giovanissimo tutti i settori dell'informatica, con particolare riguardo al settore della sicurezza e delle reti, senza tralasciare le nuove sfide tecnologiche. Attualmente lavora come consulente senior per Sun Microsystems occupandosi di architetture di rete e di sicurezza, ma nel suo passato spiccano esperienze di lavoro stimolanti tra cui IBM e Infostrada.

Convenzioni

Le seguenti convenzioni sono state adottate durante la stesura del presente libro:

Corsivo
: È usato per nomi di files, nomi di directory, comandi da eseguire e riferimenti a comandi/zone di una determinata finestra di Windows.

Sottolineato
: È usato per indicare URLs

`Testo a lunghezza costante`
: Per indicare esempi di codice o di configurazioni

Commenti e suggerimenti

Qualsiasi commento o suggerimento é benvenuto e può essere inviato al seguente indirizzo:

Giuseppe Paternò
Casella Postale 27
20090 Trezzano S/N (MI)
Italia

Oppure via e-mail a *info@gpaterno.com*. Successive edizioni ed eventuali correzioni di questa pubblicazione saranno disponibili sul sito Internet *http://www.gpaterno.com/*.

1. UNIFIED USER MANAGEMENT E SINGLE SIGN-ON

Durante i miei "pellegrinaggi" tra i clienti e anche tra i miei stessi colleghi ho scoperto che c'è un po' di confusione con il significato del termine Single Sign-On. Molto spesso pensano a questo termine come il fatto di poter immettere sempre la stessa username e password, ma in realtà il termine esatto per questo concetto è Unified User Management.

L'Unified User Management è in realtà un data-store unico contenete la base utenti, in pratica un singolo punto dove gli utenti vengono caricati (*user provisioning*) e manutenuti, incluso le loro caratteristiche (es: indirizzo e-mail, numero di telefono, home directory, ecc...). Per utenti si intende sia gli utenti "umani" che quelli digitali, ad esempio un'applicazione che deve autenticarsi verso un'altra applicazione. Tipicamente il repository viene collocato su un database capace di interfacciarsi attraverso il Lightweight Directory Access Protocol (LDAP), si tratta fondamentalmente di un database fortemente specializzato per dati i cui cambiamenti sono piuttosto rari e ottimizzato per la loro lettura e ricerca: vedremo in seguito cosa è e come funziona in dettaglio un LDAP server. Teoricamente un Unified User Management può essere realizzando mettendo la base utenti e le loro caratteristiche di un database più tradizionale come Oracle, DB2 o MySQL, ma si è dimostrato che l'uso di databases specializzati aumentano la scalabilità in un ambiente distribuito e con un alto numero di utenti (tipicamente oltre i 2/3 milioni). Inoltre LDAP è un protocollo standard definito nel RFC 3673 e universalmente riconosciuto per l'interrogazione della base dati di utenti: molte applicazioni supportano LDAP, mentre supportare differenti database potrebbe essere più oneroso (compatibilità SQL, API differenti, ecc..). Tutte le applicazioni, sia esse Web che le più tradizionali applicazioni di logon possono appoggiarsi quindi al repository LDAP per verificare le credenziali dell'utente, ma quest'ultimo solitamente è "costretto" ad inserire usename e password ad ogni accesso (login) dell'applicazione.

Il *Single Sign-On* è differente, o meglio, è qualcosa *in più* dello Unified User Management, in quanto ne presuppone l'esistenza. Il Single Sign-On è la possibilità per un utente di autenticarsi (più banalmente di inserire username e password) **una sola volta** sul sistema e di essere successivamente autenticato automaticamente ogni volta che tenta di accedere ad una risorsa di rete. Per risorsa di rete si intende un qualsiasi servizio erogato su qualsiasi sistema operativo, ad esempio l'accesso interattivo telnet/ssh, la condivisione di file e stampanti o l'accesso ad un'applicazione Web. Come si vede, in realtà si tratta di un processo molto più complesso dello Unified User Management nel quale il concetto di *single* assume un'importanza fondamentale. I benefici sono molteplici, ad esempio si **esemplifica l'amministrazione**, in quanto l'utente viene inserito una volta sola e modificato in un singolo punto con una singola interfaccia di ammininistrazione, e conseguentemente una **consolidazione di reti e sistemi eterogenei** in quanto operante cross-platform. Avendo un singolo punto di accesso e un repository unico si ha conseguentemente un **maggior controllo**, evitando di lasciare utenze non più usate nelle applicazioni (es: per un dipendente che ha lasciato l'azienda). Si ha anche una **maggior produttività dell'utente** che non è più costretto a collegarsi a diverse applicazioni o ricordarsi per ogni sito Web o risorsa di rete una username e password diversa, con minor chiamate all'helpdesk interno per problemi di login o reset di passwords, e grazie a questo potrebbe derivarne una **sicurezza migliore,** in quanto l'utente sarà meno invogliato a scriversi le password in un foglietto (ne deve ricordare solo una).

2. IL MODELLO DI MICROSOFT ACTIVE DIRECTORY

Nella mia esperienza di consulente presso fornitori di servizi, ho trovato differenti sistemi ottenuti per il Single Sign-On, quasi esclusivamente relativi ai soli servizi Web. Molti di essi si basano su un agente che si installa su un web server e un "policy manager" che è in grado di autenticare/autorizzare l'utente ad una specifica risorsa. La tecnica impiegata è efficiente e raggiunge lo scopo per il Single Sign-On su sistemi web, ma cosa succede se vogliamo autenticarci ad una risorsa non-web, ad esempio collegarci via SSH ad un sistema Unix ? Dovremmo ricordare un'ulteriore utenza e password. Questo non sarebbe stato sufficiente al fine di dimostrare la fattibilità del modello previsto, in quanto dovrebbe coprire l'autenticazione utente a 360 gradi.

Secondo il mio modesto avviso, il produttore che è riuscito a effettuare un vero e proprio "Single Sign-On" è Microsoft: attraverso Active Directory, ovvero il nuovo modello di dominio introdotto da Windows 2000, è in grado di effettuare un'autenticazione su tutte le risorse di rete (condivisione file/stampanti, siti web, ecc...) semplicemente facendo una "join" al dominio.

Microsoft ha riutilizzato differenti protocolli standard già usati e definiti in ambienti Unix attraverso RFC, anche se a mio avviso paradossalmente poco usati. Devo ammettere che alla casa di Redmond va tutta la mia ammirazione per aver saputo amalgamare in maniera opportuna i vari protocolli, anche se una nota di demerito va al fatto che ne hanno modificato la loro struttura, come in molti casi avviene quando la famosa software company adotta altri protocolli standard. Per soddisfare la curiosità, vedremo in seguito brevemente quali sono le differenze fra lo standard e l'implementazione Microsoft.

Il modello Microsoft prevede un repository centrale realizzato attraverso il protocollo LDAP che contiene le utenze e la configurazione delle macchine e dei servizi correlati. Per quanto riguarda invece l'autenticazione degli utenti (siano

essi umani o servizi), ha creato un layer di autenticazione che prevede la negoziazione della tipologia di "authentication service" supportato. In particolare si tratta di tre tipologie di autenticazione:

- **Windows NT LAN Manager (NTLM).** Non si tratta di una vera e propria tecnologia di Single Sign-On: anche se l'utente non si accorge della richiesta di autenticazione, ogni volta che si tenta di accedere ad una risorsa, il client invia le credenziali di autenticazione (username e password) che mantiene in una cache in memoria. Questa funzionalità viene mantenuta per compatibilità "verso il basso", ovvero verso i domini di Windows NT.
- **Kerberos.** Nuova metodologia di autenticazione introdotta con Windows 2000 e rappresenta un vero e proprio sistema di Single Sign-On. Si basa sempre sul concetto di username e password, ma durante la negoziazione del servizio viene passato un ticket (una sorta di pass) che garantisce la nostra identità, senza pero' rivelarla (come avviene per NTLM). Kerberos è la tecnologia di default usata in Windows 2000 o superiori ed è quella scelta per la realizzazione del prototipo di Single Sign-On: vedremo in seguito come funziona il sistema Kerberos in dettaglio.
- **SSL.** Un'altro metodo introdotto per l'autenticazione è l'adozione della crittografia a chiave pubblica/privata attraverso SSL. In realtà l'uso vero e proprio di SSL per l'autenticazione si ha solamente attraverso le applicazioni Web. Per effettuare l'accesso su altre risorse di rete, ad esempio condivisione file e stampanti, Microsoft ha esteso il protocollo Kerberos per usare chiavi pubbliche/private SSL anzichè una chiave privata (la password): questa modifica è descritta in un Internet Draft dal titolo "Public Key Cryptography for Initial Authentication in Kerberos" e permette di utilizzare una smart card per il logon interattivo ai client.

3. LA TECNOLOGIA

Attraverso l'analisi del modello fornito da Microsoft si è giunti all'estrapolazione delle tecnologie usate dalla casa di Redmond, ovvero Kerberos e LDAP, con alcune varianti. Questo capitolo vuole descrivere le due tecnologie e come esse vengono impiegate in ambiente Windows e come possono essere usate per stabilire un modello tecnologico di Single Sign-on.

Kerberos

Kerberos è un protocollo di autenticazione sviluppato presso il MIT, il cui nome evoca il famoso cane a tre teste della mitologica greca, e fornisce un meccanismo per una autenticazione reciproca (mutual authentication) tra client e server o tra due server. Il protocollo assume che le transazioni iniziali tra client e server avverrà in una rete insicura, possibilmente sotto monitor di qualche utente smaliziato, e dove anche i computer non sono messi in un luogo sicuro: praticamente Internet. In un ambiente insicuro, sia esso Internet o una intranet, non è escluso poter trovare un attaccante che potrebbe intercettare le comunicazioni tra client e server e puo' catturare dati o peggio modificarli.

Il sistema di autenticazione Kerberos, così come la sua figura mitologica, è composto da tre elementi: il Key Distribution Center (KDC), l'utente e il server con il servizio a cui l'utente vuole accedere. Nei prossimi paragrafi si riassumeranno le caratteristiche salienti di Kerberos, ma per maggiori informazioni si suggerisce una lettura approfondita della bibliografia, in particolare del whitepaper Kerberos: An Authentication Service for Open Network Systems, presentato a USENIX nel 1988, e del WhitePaper "Windows 2000 Kerberos Authentication".

Il Key Distribution Center

In Kerberos esiste il concetto di *Realm*, simile a quello dei domini di Windows. Si tratta di un dominio (o Realm appunto) di autenticazione su cui gli utenti verificano le proprie credenziali, ovvero username e password. Tipicamente ad un nome di dominio DNS viene associato un Realm, ad esempio ATHENA.MIT.EDU oppure AZIENDA.IT. Convenzionalmente i *Realm* di Kerberos sono scritti in maiuscolo, contrariamente ad i nomi DNS che convenzionalmente vengono scritti in minuscolo (anche se il DNS non è case sensitive). Il Key Distribution Center è il cuore di un realm e svolge due funzioni principali: l'*Authentication Service (AS)* e il *Ticket Granting Service (TGS)*. Come si evince dalla figura sottostante, lo scambio di chiavi dell'utente per accedere ad una risorsa avviene in tre fasi: *AS Exchange*, *TGS Exchange* e *Client/Server (CS) Exchange* (Fig. 1).

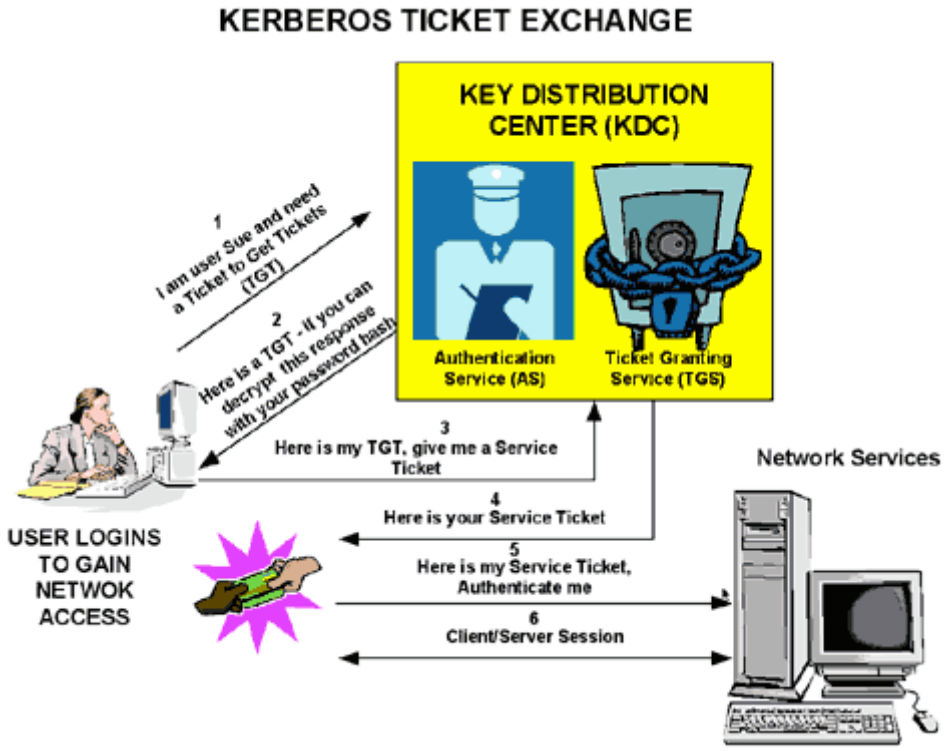


Figura 1 - Kerberos Ticket Exchange

AS Exchange

Quando l'utente effettua un'autenticazione alla rete, esso deve immettere utenza e password in modo da essere verificato dall'Authentication Service (AS) che fa parte del KDC. Il KDC interrogherà il suo database di utenza e, una volta verificato le credenziali dell'utente, rilascia all'utente un Ticket Granting Ticket (TGT) che è valido per il Realm di appartenenza. Il TGT ha una "vita limitata" e può essere rinnovata automaticamente qualora la sessione utente sia ancora attiva e senza che quest'ultimo reinserisca username/password. Il TGT è mantenuto in memoria nella macchina client e viene usato per richiedere accesso ad un determinato servizio.

Vediamo in dettaglio lo scambio per l'ottenimento del TGT. La richiesta dell'AS identifica il client al KDC in chiaro. Se la preautenticazione è abilitata sull'utente, un time stamp viene criptato con l'hash della password dell'utente come chiave di crittografia. Se il KDC, una volta decriptato il timestamp con l'hash della password, identifica come valido il timestamp allora il KDC saprà che non si tratta di un playback attack di una richiesta precedente. La preautenticazione dell'utente può essere disabilitata, ma è consigliabile usarla quando possibile. Se il KDC autorizza la richiesta del client all'ottenimento del TGT, la riposta dell'AS nei confronti del client includerà due sezioni: un TGT criptato con una chiave che solamente il KDC (TGS) può decriptare e una chiave di sessione criptata con l'hash della password dell'utente (serve per le future comunicazioni con il KDC). Siccome il client non può leggere il contenuto del TGT, esso dovrà presentarlo così com'è al TGS per avere un *service ticket*. Il TGT include parametri di time-to-live, autorizzazione, una chiave di sessione per comunicare con il client e il nome del client/utente.

TGS Exchange

L'utente presenta il TGT al TGS (una delle due parti del KDC) quando vuole accedere ad un determinato servizio. Il TGS verifica il TGT dell'utente e crea un ticket e una chiave di sessione sia per il client che per il server remoto. Queste informazioni, conosciuti come *service ticket*, è poi memorizzato localmente sulla macchina client.

Il TGS riceve il TGT del client e lo legge attraverso la propria chiave. Se il TGS verifica correttamente la richiesta del client, un Service Ticket viene generato sia per il client che per il server su cui l'utente sta tentando di accedere. Il client legge la sua porzione usando la chiave di sessione del TGS che ha ricevuto precedentemente durante la risposta dell'AS. Il client presenta la porzione del Service Ticket relativa al server alla risorsa che sta accedendo durante il *Client/Server Exchange*.

Client/Server Exchange

Una volta che l'utente ha ottenuto il Service Ticket, può stabilire la sessione con il servizio erogato dal server. Il server può decriptare le informazioni che arrivano indirettamente dal TGS usando la propria chiave stabilita con il KDC. Il Service Ticket è usato quindi per autenticare l'utente e stabilire una sessione di erogazione servizio tra il server e il client. Dopo che il time-to-live (lifetime) del ticket è terminato, il Service Ticket deve essere rinnovato per usare il servizio.

Nello specifico, il client passa la porzione relativa al server del Service Ticket, al server verso cui sta richiedendo il servizio per ottenere la sessione client/server. Se la mutual authentication è abilitata, il server ritorna un timestamp criptato con la chiave di sessione del Service Ticket. Se il timestamp viene decriptato correttamente, non solo il client si è autenticato al server, ma il server si è anche autenticato con il client. Il target server non ha mai comunicato direttamente con il KDC, il che riduce il carico sul KDC.

Kerberos e l'orologio di sistema (clockskew)

Come accennato in precedenza, il meccanismo di autenicazione di Kerberos si basa sui TimeStamp, ovvero sulla data e l'ora. È necessario quindi che tutti i sistemi coinvolti abbiamo l'orologio di sistema allineato: seppur una cosa banale, molte volte durante il set-up del laboratorio si sono verificati problemi di autenticazione a causa del disallineamento dell'ora (nel mio caso era il fuso orario). Si consiglia, qualora non si abbia, di installare in azienda un time server di riferimento e sincronizzare sia i server che le workstations con questo server.

GSSAPI, SSPI e SASL

Anche se non fa parte direttamente del protocollo Kerberos, descrivere le GSSAPI è propedeutico alla comprensione del resto del WhitePaper. Le Generic Security Service Application Programming Interface (GSSAPI) sono un framework che fornisce servizi di sicurezza alle applicazioni. Lo scopo della nascita delle GSSAPI era creare un "abstraction layer" attraverso delle API standard per l'autenticazione, in modo che ogni programma potesse implementare l'autenticazione astraendosi dal sistema di autenticazione sottostante. L'implementazione più usata delle GSSAPI è quella relativa a Kerberos 5, tanto che quando una particolare applicazione supporta le GSSAPI, significa in realtà il supporto di Kerberos 5. Inoltre ogni implementazione Kerberos 5 include

un'implementazione GSSAPI.

Le GSSAPI sono state declinate nell'ambiente Windows come Microsoft Security Support Provider Interface (SSPI): tali interfaccie sono del tutto simili a quelle fornite da GSSAPI. Molti applicativi Windows si riferiscono però al supporto Kerberos come GSSAPI, in realtà però sfruttano le SSPI intrinseche del sistema operativo Windows 2000 o superiori.

Per quanto riguarda il mondo Unix, ad oggi il modello di security "abstraction layer", che avrebbe dovuto implementare GSSAPI, è stato implementato attraverso le *Simple Authentication and Security Layer (SASL)*. Queste API sono state create dalla Carnegie Mellon University e documentate nell'RFC 2222. Le SASL saranno usate durante la creazione del modello principalmente durante l'implementazione di LDAP, che userà le SASL come meccanismo di autenticazione dei clients.

SPNEGO e l'autenticazione Web

Come abbiamo accennato precedentemente, le GSSAPI foriniscono un set di API generiche che si astraggono dal meccanismo di autenticazione sottostante. Quando due applicazioni parlano tra di loro attraverso le GSSAPI e sfruttando lo stesso meccanismo di autenticazione, si dice che hanno stabilito un contesto di sicurezza (security context). Il problema di questo meccanismo è che le due entità devono sapere a priori quale meccanismo di autenticazione hanno a disposizione e quale quindi possono usare: GSSAPI non prevede un meccanismo di "handshake" tra i due peer per sapere quale meccanismo di sicurezza hanno in comune e stabilire quindi un security context.

Il Simple and Protected GSS-API Negotiation Mechanism (SPNEGO) è stato creato appositamente per determinare durante una connessione tra due peer quale sono i meccanismi di autenticazione disponibili e selezionare il miglior meccanismo comune. SPNEGO viene usato in Windows 2000 per negoziare quali sono i meccanismi di autenticazione supportati (tipicamente Kerberos e NTLM) e sceglierne il migliore: ad esempio, se un sistema Windows 98 deve colloquiare con un server Windows 2003, SPNEGO sceglierà il meccanismo NTLM perchè Windows 98 non supporta Kerberos. Al contrario, se un Windows XP colloquierà con un server Windows 2003 selezionerà Kerberos in quanto supportato da entrambi e reputato migliore di NTLM.

Vediamo un po' più da vicino il comportamento di SPNEGO. L'iniziatore della connessione propone un meccanismo di autenticazione o una lista ordinata di sistemi di autenticazione, il destinatario sceglie accettando il meccanismo di sicurezza proposta, oppure ne sceglie uno fra la lista proposta o rigetta i valori proposti. Successivamente il desitinatario della trasmissione informa l'iniziatore della connessione sulla sua scelta. Questo meccanismo potrebbe essere lento su connessioni non particolarmente veloci (tipicamente le WAN o intefaccie radio) per cui SPNEGO tenta subito di proporre un meccanismo di autenticazione al setup della connessione, cosi' da minimizzare i tempi se il meccanismo proposto va bene al destinatario della connessione.

Il meccanismo della connessione è ben più complesso e include anche un modo per proteggere la negoziazione: qualora si fosse interessati ad approfondire tale meccanismo, è consigliabile leggersi l'RFC relativo (rif. Bibliografia).

SPNEGO è alla base di tutti i meccanismi di autenticazione di Windows, incluso la condivisione file e stampanti, LDAP, IPSec. Microsoft ha anche applicato SPNEGO alle connessioni Web (HTTP), estendendo il meccanismo dell'autenticazione: una descrizione di come questo sia stato realizzato è disponibile nell'Internet Draft di Brezak "HTTP Authentication: SPNEGO Access Authentication As implemented in Microsoft Windows 2000". Sostanzialmente SPNEGO viene incapsulato nell'header di HTTP con la keyword *WWW-Authenticate: Negotiate* e un base64 encoding del risultato delle API. Tutte i successivi dialoghi tra il client e il server verranno veicolati attraverso questo meccanismo fino al completamento della sessione SPNEGO.

Il meccanismo di SPEGO su HTTP verrà utilizzato per realizzare il modello di Single Sign-On per quanto riguarda l'autenticazione Web.

LDAP

Il Lightweight Directory Access Protocol (LDAP) è, come suggerisce il nome stesso, un protocollo standard documentato nel RFC 2251 per accedere a servizi di directory. Una directory è un database specializzato ottimizzato per la lettura e per la ricerca dei dati attraverso filtri sofisticati, in cui il dato è formato da un attributo e relativa descrizione (ad esempio "email=rossi@azienda.it"). Le directories, inteso come database, non supporta transazioni complicate o azioni di rollback che si trovano nei classici RDBMS, di solito ottimizzati per le transazioni e aggiornamenti complessi. Gli aggiornamenti riguardanti i repository LDAP sono tipicamente rari, in quanto i dati sono semi-statici, semplici e non richiedono transazioni; invece il database è ottimizzato per dare risposte veloci a molte ricerche contemporanee. Molti dei directory server in commercio sono in grado di replicare il loro database per aumentarne l'affidabilità, la disponibilità e sono in grado con questo meccanismo di ridurre il tempo di risposta scalando orizzontalmente.

Il tipo di informazioni inserite in LDAP è basato sul modello di entries. Una entry è l'insieme di una serie di attributi relativi ad una chiave univoca detta Distinguish Name (DN). Ogni entry ha un tipo ed uno o più valori. I tipi sono generalmente stringhe mnemoniche come "cn" per Common name o "mail" per indirizzi e-mail. La sintassi del valore dipende dal tipo di attributo dichiarato, per esempio cn può contenere il valore Mario Rossi, mentre il tipo jpegPhoto conterrà una fotografia in formato JPEG (binario).

Le informazioni sono organizzati secondo un modello gerarchico, in una maniera simile al DNS. Tipicamente hanno sotto di loro delle alberature di tipo *organizational units,* ovvero organizzazioni aziendali, *People* che contengono le persone oppure *Groups* che contengono i gruppi. L'organizzazione di un alberatura LDAP è molto flessibile e può essere gestita secondo esigenze specifiche, ad esempio applicative, ma in ogni caso deve essere ben pianificata. Tutti gli oggetti però devono avere uno o più attributi di tipologia *objectClass*. I valori di *objectClass* sono relativi a degli schemi (*schema* appunto) a cui la entry deve obbedire: ad esempio una entry con un *objectClass* di valore *person* ha come obbligatorio il tipo *sn* (Surname), pertanto deve avere un attributo di tipo *sn* all'interno della entry.

Abbiamo accennato precedentemente al Distinguish Name (DN): esso rappresenta il modo per referenziare univocamente una entry nel database LDAP (anche chiamata RDN). Tipicamente il DN è formato dal Common Name (CN), Organizational Unit (OU) e Base DN (la base dell'alberatura LDAP); non è detto che per semplicità o per evitare univocità si possa scegliere come DN altri parametri, l'importante che questi parametri appaiano sempre all'interno della entry stessa e che contenga anche informazioni relativi alle entry da cui discendono (nel nostro esempio DEVE esistere una entry di tipo Base DN e una entry OU che derivi dalla Base DN).

LDAP definisce operazioni per interrogare e aggiornare il suo database. Le operazioni definite sono aggiunta, cancellazione, modifica e rinomina di una entry. Molte delle volte, come abbiamo detto, LDAP viene usato per ricercare: la ricerca viene effettuata su parte dell'alberatura del database con dei criteri chiamati filtri. Ogni entry che corrisponde ai filtri determinati verrà restituita. Ad esempio,

nell'alberatura *dc=azienda,dc=it* si possono cercare le entries relative a tutte le persone con il cognome Rossi, da cui vogliamo estrarre solamente i loro indirizzi di e-mail (Fig. 2).

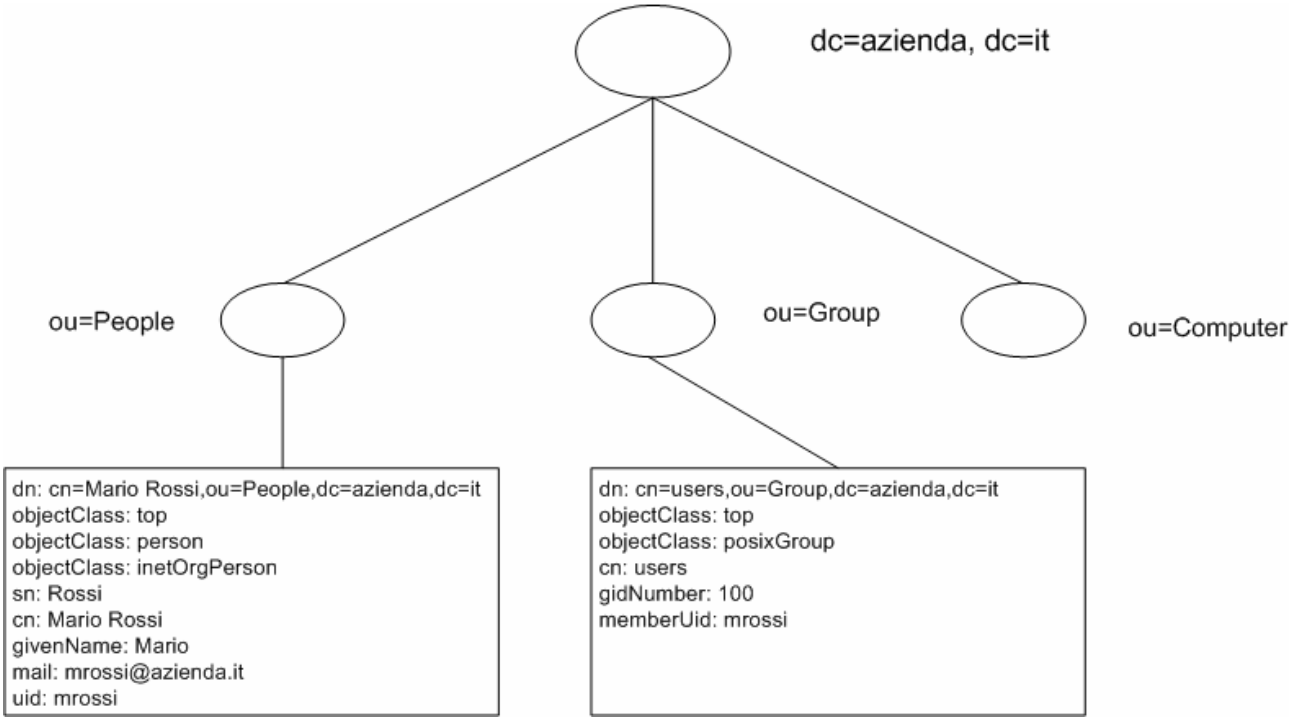


Figura 2 - Esempio di alberatura LDAP

LDAP è un protocollo di tipo client/server. Uno o più LDAP server contengono i dati che formano il Directory Information Tree (DIT). I client si connettono al server e sottopongono una query. Il server risponde con le informazioni e/o con un puntatore a dove il client può richiedere informazioni ulteriori, tipicamente ad un altro LDAP. Quest'ultimo processo è detto di referral. In un modello simile, non importa a quale LDAP server il client sia collegato, in quanto tutti i server vengono collegati tra di loro similmente a quanto avviene nel sistema DNS. Questo è una feature per il concatenamento di più server LDAP, ad esempio tra organizzazioni dello stessa azienda oppure tra aziende diverse che cooperano tra di loro.

Per quanto riguarda la sicurezza delle informazioni contenute in un directory, LDAP contempla un meccanismo di autenticazione tra un client e un server LDAP, inoltre permette anche la crittografia attraverso SSL. L'autenticazione di un client non è però l'autorizzazione dello stesso a quali dati può accedere: è giusto che il consulente riesca a vedere il numero di telefono di casa dell'amministratore delegato? LDAP non specifica come questo deve avvenire: molti degli LDAP server si sono dotati al loro interno di un sistema di Access Control Lists (ACL) per autorizzare classi di utenti a determinati attributi delle entried del databases (nel nostro esempio, il consulente potrà leggere le e-mail addresses dell'azienda, mentre il commerciale ha accesso anche ai numeri dei cellulari aziendali). Tuttavia queste ACL non sono standard ed ogni implementatore ha adottato una sua metodologia.

LDAP e Kerberos

Alcuni server LDAP, spesso attraverso plugin, sono in grado di usare un password back-end differente, ovvero la possibilità di accedere alla password associata ad un utente LDAP tramite altri meccanismi, ad esempio facendo il check sul file delle password di sistema (/etc/passwd), su di un database RDBMS, oppure tremite Kerberos.

Durante l'implementazione della soluzione, andremo a sfruttare questa caratteristica di un LDAP server per poter effettuare il controllo della password sul repository di Kerberos. Vedremo in seguito che useremo LDAP solamente come repository delle caratteristiche dell'utente (es: home directory, gid/uid, ecc..), ma non ci autenticheremo mai direttamente con LDAP. Anche se l'uso di un back-end Kerberos per la verifica delle password dell'utente sembra superfluo, non lo è per quelle legacy applications che non riescono ad interfacciarsi direttamente con Kerberos, ma comunque riescono ad accedere ad un repository LDAP.

Ammettiamo ad esempio di aver acquistato un gestionale client/server che provvede all'autenticazione/autorizzazione dell'utente attraverso LDAP: usando come back-end Kerberos, l'utente potrà sempre mantenere allineata la sua password. Un caso più concreto è il file server OpenSource Samba, che alla release 3 riesce solo a sfruttare Kerberos come Active Directory client: nel nostro caso si potrebbe usare Samba con un back-end LDAP.

4. LO SCENARIO

Quando ho cominciato questa "avventura", mi sono prefisso alcuni punti. Il primo in assoluto è che prima di procedere alla rinfusa con l'implementazione del software, ho preferito usare un approccio più strutturato al problema, definendomi degli obiettivi o requisiti. Vediamoli insieme:

- Il KDC doveva risiedere principalmente su un ambiente Unix, ma si doveva comunque dimostrare teoricamente la possibilità di usare un KDC su ambiente Windows (Active Directory)
- Il server LDAP doveva risiedere su un ambiente Unix, con gli stessi requisiti del KDC per quanto riguarda l'ambiente Windows
- Il web server doveva risiedere su piattaforma Unix
- Il mail server doveva risiedere su piattaforma Unix
- La workstation Unix doveva permettere ad un nuovo utente non presente sul sistema locale di:
    - autenticarsi con il KDC
    - ottenere un TGT al login, che dura fino alla chiusura della sessione
    - scaricare i parametri relativi all'utente dal LDAP, in particolare gid/uid e home directory
    - accedere ad un sistema unix remoto tramite ssh senza fornire utente e password
    - accedere ad un web server protetto senza fornire utente e password
    - accedere alla propria casella di posta senza fornire utente e password
- La workstation Windows doveva permettere ad un nuovo utente non presente sul sistema locale di:
    - autenticarsi con il KDC
    - ottenere un TGT al login, che dura fino alla chiusura della sessione
    - accedere ad un sistema unix remoto tramite ssh senza fornire utente e password
    - accedere ad un web server protetto senza fornire utente e password
    - accedere alla propria casella di posta senza fornire utente e password

Uno degli altri obbiettivi, che non rientra però nei requisiti, è che l'ambiente di test doveva riflettere un ambiente di produzione: è abbastanza semplice produrre qualcosa in laboratorio, ma era necessario capire quali sono le implicazioni quando questo si traduce in un ambiente di produzione o di pre-produzione.

A questo scopo è stato creato un laboratorio come segue:

- Un server che ospiterà il Key Distribution Centre e LDAP server: si noti che verranno installati in modo da essere indipendenti l'uno dall'altro.
- Un server che ospiterà il Web Server e il mail server
- Un firewall che separi i server e i client
- Un client Windows 2000
- Un client Linux
- Un client MacOS X

Durante il laboratorio sono stati usati server Linux, con distribuzione Debian: il fatto di aver usato Linux è puramente legato ad un problema di disponibilità di risorse. Da un punto di vista teorico, questo set-up vale per qualsiasi ambiente Unix che supporti il sistema di autenticazione basato sul Pluggable Authentication Module (PAM), incluso quindi anche il sistema operativo della Apple MacOS X. Visto l'incremento della diffusione del sistema operativo della Apple, sono stati condotti dei test anche sul sistema MacOS X 10.3 "Panther", anche se non previsti inzialmente. Sono stati fatti dei test anche con Solaris, ma non approfonditamente.

Definiamo per chiarezza l'ambiente e le macchine convolte (Fig. 3).

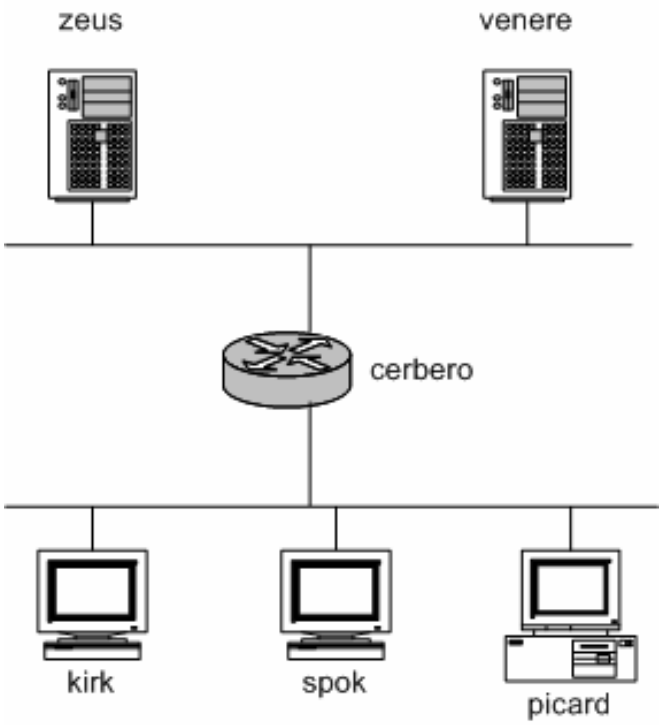


Figura 3 - L'ambiente del laboratorio

Nome macchina	**Alias**	**Ruolo**
zeus.azienda.it	kdc.azienda.it ldap.azienda.it ns.azienda.it time.azienda.it	NTP, DNS, KDC e LDAP Server
venere.azienda.it	www.azienda.it mail.azienda.it	Server Web e Mail
cerbero.azienda.it		Firewall
kirk.azienda.it		Client Linux
picard.azienda.it		Client Windows 2000
spok.azienda.it		Client MacOS X

5. L'INFRASTRUTTURA DI BASE

Una volta definito l'ambiente in cui verrà sviluppato il modello, si è proceduto alla scelta dei programmi da integrare: la scelta è ricaduta su programmi OpenSource, in quanto più facilmente reperibili ed eventualmente modificabili. È possibile realizzare la stessa infrastruttura con programmi supportati, ma bisogna verificare con il produttore se le funzionalità siano disponibili, ad esempio se l'LDAP server supporta Kerberos come password back-end.

Preparazione del DNS

Kerberos richiede che vi siano alcune entries nel DNS che puntino ai servizi erogati, anche se pochi programmi sembrano cercare queste entries. È bene verificare che il reverse look-up di ciascun server sia garantito: alcuni problemi legati al kerberos sono relativi al fatto di non riuscire a risolvere in modo corretto i nomi. Per quanto riguarda il laboratorio, questo è il DNS relativo alla parte kerberos:

```
$TTL    604800
$ORIGIN azienda.it.
@       IN      SOA     azienda.it. postmaster.azienda.it. (
                      26010400         ; Serial
                        604800         ; Refresh
                         86400         ; Retry
                       2419200         ; Expire
                        604800 )       ; Negative Cache TTL
;
@                      IN  NS ns.azienda.it.
```

```
; Kerberos principals (DA AGGIUNGERE!!!!)
_kerberos               IN  TXT "AZIENDA.IT"
_kerberos-master._udp   IN  SRV 0 0 88 kdc
_kerberos-adm._tcp      IN  SRV 0 0 749 kdc
_kpasswd._udp           IN  SRV 0 0 565 kdc
_kerberos._udp          IN  SRV 0 0 88 kdc
_ldap._tcp              IN  SRV 0 0 389 ldap

; Aliases
kdc                     IN  CNAME zeus.azienda.it.
ldap                    IN  CNAME zeus.azienda.it.
ns                      IN  CNAME zeus.azienda.it.
www                     IN  CNAME venere.azienda.it.
mail                    IN  CNAME venere.azienda.it.
time                    IN  CNAME zeus.azienda.it.

; Real IP Addresses
zeus                    IN  A   192.168.0.10
venere                  IN  A   192.168.0.20
cerbero-server          IN  A   192.168.0.1

cerbero-client          IN  A   192.168.1.1
kirk                    IN  A   192.168.1.20
picard                  IN  A   192.168.1.40
spok                    IN  A   192.168.1.60
```

Come si può notare nell'esempio precedente, le entries importati sono quelle che si riferiscono ai Service Locator (SRV) e al *_kerberos* che specifica, attraverso il record TXT, il realm della rete locale (nel nostro caso AZIENDA.IT). I numeri successivi alla definizione dei record sono nell'ordine: la priorità (simile alla priorità del record MX), il peso da usare in caso di load-balancing e la porta TCP o UDP su cui è in attesa il servizio. Come accennato precedentemente, non tutti i programmi sfruttano la risoluzione del realm e/o dei servizi tramite DNS, ma è bene provvedere alla loro definizione per una corretta configurazione.

Il server NTP

Un altro prerequisito al funzionamento di Kerberos è la sincronizzazione degli orologi di sistema. A tal scopo è stato creato un NTP server sul sistema *zeus.azienda.it* e referenziato come *time.azienda.it*. Vediamo un esempio di configurazione di un NTP server (file */etc/ntp.conf*):

```
driftfile /var/lib/ntp/ntp.drift
statsdir /var/log/ntpstats/

statistics loopstats peerstats clockstats
filegen loopstats file loopstats type day enable
filegen peerstats file peerstats type day enable
filegen clockstats file clockstats type day enable
```

```
authenticate no

## Mettere il proprio server di riferimento, esempio quello del
## vostro provider. Il server time.ien.it si riferisce
## all'orologio ufficiale italiano, ovvero l'Istituto Galileo
## Ferraris di Torino. Prima di usare un NTP server è bene
## avvisare il relativo l'amministratore di sistema.
server time.ien.it prefer
```

Per maggiori informazioni su come configurare un server NTP e i relativi client si faccia riferimento alla manualistica del sistema operativo. Nei capitoli successivi si darà per assunto che la sincronizzazione degli orologi sia stata effettuata correttamente.

Il server Kerberos

Il server Kerberos installato è il *MIT Kerberos V*: molti sistemi operativi Unix includono ormai un server Kerberos, quindi non mi soffermerò sui dettagli della compilazione del server: si faccia riferimento alla documentazione relativa al proprio vendor, oppure si verifichi per Linux l'elenco dei pacchetti installabili, solitamente *libkrb5-dev, krb5-admin-server* e *krb5-kdc.* È comunque possibile scaricarne i sorgenti nella home page del progetto del MIT, ovvero http://web.mit.edu/kerberos/www/.

Durante l'installazione del server è bene installare anche i pacchetti relativi allo sviluppo del kerberos, solitamente le librerie e gli *include files*: tali pacchetti ci serviranno successivamente per compilare le applicazioni aggiungendo il supporto Kerberos.

Dopo aver installato i pacchetti, è necessario personalizzare il file di configuazione */etc/krb5.conf* presente sul KDC: tale file verrà usato sia dal KDC che dal client kerberos della macchina. Nell'esempio seguente si noti il tipo di crittografia specificata: di defaut il MIT Kerberos 5 usa *des3-hmac-sha1*, ma l'implementazione Kerberos di Windows è in grado solamente di comprendere la crittografia *des-cbc-crc* o *des-cbc-md5*. Per assicurare la compatibilità con Windows, si è scelto pertanto di adottare questa crittografia: essendo DES una crittografia debole, è consigliabile usare quando possibile comunicazioni protette.

```
[libdefaults]
        default_realm = AZIENDA.IT
## For Windows 2000 compatibility
        default_tgs_enctypes = des-cbc-crc des-cbc-md5
        default_tkt_enctypes = des-cbc-crc des-cbc-md5
        permitted_enctypes = des-cbc-crc des-cbc-md5
        krb4_config = /etc/krb.conf
        krb4_realms = /etc/krb.realms
        kdc_timesync = 1
```

```
        ccache_type = 4
        forwardable = true
        proxiable = true

# The following libdefaults parameters are only for Heimdal Kerberos.
        v4_instance_resolve = false
        v4_name_convert = {
                host = {
                        rcmd = host
                        ftp = ftp
                }
                plain = {
                        something = something-else
                }
        }

[realms]
        AZIENDA.IT = {
                kdc = kdc.azienda.it
                admin_server = kdc.azienda.it
        }

[domain_realm]
        .azienda.it = AZIENDA.IT

[login]
        krb4_convert = true
        krb4_get_tickets = true
```

Una volta configurato */etc/krb5.conf* si è pronti a creare il Realm Kerberos. Da utenza di *root* è necessario eseguire i seguenti comandi:

```
# kdb5_util create -s
# kadmin.local -q "ktadd -l /etc/krb5kdc/kadm5.keytab kadmin/admin"
# kadmin.local -q "ktadd -l /etc/krb5kdc/kadm5.keytab
kadmin/changepw"
# kadmin.local -q "addprinc -pw miapassword1 krbadm@AZIENDA.IT"
# kadmin.local -q "addprinc -pw miapassword2 ldapadm@AZIENDA.IT"
```

Le ultime due righe si riferiscono all'aggiunta di due utenze: la prima ci servirà come amministratore di Kerberos, la seconda invece sarà associata all'amministratore di LDAP. A questo punto siamo pronti per avviare il server:

```
# krb5kdc &
# kadmind &
```

Il metodo migliore per avviare i due servizi kerberos (KDC il primo e Admin server il secondo) è attraverso gli script di avvio: riferirsi al manuale del proprio vendor per ulteriori dettagli.

L'ultimo passo per l'installazione di Kerberos è quello relativo alle access lists, in quanto dobbiamo abilitare l'utenza (o il principal) *krbadm@AZIENDA.IT* ad

amministrare il server Kerberos. Il file in oggetto è */etc/krb5kdc/kadm5.acl*, vediamone un esempio:

```
kadmin/admin@AZIENDA.IT       *
krbadm@AZIENDA.IT             *
*/*@AZIENDA.IT                i
```

L'asterisco dopo l'account significa che può eseguire qualsiasi operazione, mentre la lettera "*i*" significa che può leggere le informazioni nel database.

La libreria SASL

Precedentemente è stato accennato come la libreria SASL è in grado di offrire un livello di astrazione per l'autenticazione di protocolli di rete, come definito nell'RFC 2222. Molti sistemi Unix hanno le librerie SASL, ma non tutti supportano GSSAPI come meccanismo: ad esempio Debian è in grado di supportare le GSSAPI attraverso il pacchetto *libsasl-gssapi-mit* e RedHat attraverso *cyrus-sasl-gssapi,* ma è necessario contattare il proprio vendor per sapere se implementa/distribuisce SASL e se GSSAPI è un meccanismo valido.

Qualora non si disponesse delle GSSAPI come meccanismo di autenticazione, è possibile scaricare il sorgente di Cyrus SASL dal sito http://asg.web.cmu.edu/sasl/sasl-library.html. Per compilarlo, occorre specificare i seguenti parametri:

```
# ./configure --prefix=/usr/local --enable-static --enable-login \
   --enable-gssapi=/usr/kerberos/ --disable-krb4
# make
# make install
```

È necessario sostituire */usr/kerberos* con la directory contenente gli include files del MIT Kerberos V. Anche per la libreria SASL è importante installare i files di include, pertanto quando si installa attraverso i pacchetti della propria distribuzione è necessario installare anche i files di development.

Il server LDAP

Come server LDAP è stato scelto OpenLDAP, disponibile su http://www.openldap.org. Questo server, oltre ad essere gratuito, ha la possibilità di sfruttare Kerberos come back-end delle password utente, pertanto fa al caso nostro.

Molte distribuzioni dispongono di openldap server tra i pacchetti disponibili, raramente però viene compilato con le opzioni necessarie, pertanto serve ricompilarlo. Come prerequisito dobbiamo avere gli header files e le librerie dei seguenti pacchetti: Kerberos, SASL, OpenSSL e Berkley DB (o equivalente). Le opzioni che ci serviranno in particolare sono:

```
--disable-cleartext
--disable-rlookups
--with-tls
--enable-kpasswd
```

Queste sono le opzioni che sono state usate per compilare il server OpenLDAP:

```
$ ./configure --prefix=/usr/local/ldap --enable-debug --enable-syslog
\
--enable-proctitle --enable-cache --enable-referrals --enable-ipv6 \
--enable-local --with-readline --with-threads --disable-cleartext \
--enable-multimaster --enable-phonetic --disable-rlookups \
--enable-wrappers --enable-dynamic --enable-dnssrv --enable-ldap \
--enable-ldbm --enable-passwd --enable-shell --enable-sql --enable-
slurpd \
--enable-shared
$ make depend
$ make
# make install (da root!)
```

Prima di configurare il server LDAP è necessario prima preparare un principal Kerberos e un certificato digitale per usare SSL. Per creare un principal per LDAP, sul nostro sistema *zeus.azienda.it*, eseguire il seguente comando:

```
# kadmin.local -q "addprinc -randkey ldap/ldap.azienda.it@AZIENDA.IT"
# kadmin.local -q "ktadd ldap/ldap.azienda.it@AZIENDA.IT"
```

Il primo comando crea il principal, mentre il secondo inserisce la chiave nel file di default */etc/krb5.keytab* che verrà usato dal server. Successivamente è necessario ottenere un certificato SSL (X.509) per abilitare SSL con LDAP. Qualora non si disponga già di una Certification Authority (CA), con OpenSSL viene distribuito uno script perl (CA.pl) capace di generare e mantenere una piccola CA. Grazie a questo script è possibile generare una CA ed un certificato di test con i seguenti comandi:

```
CA.pl -newca
CA.pl -newreq
CA.pl -signreq
```

Come creare e mantenere una CA è al di fuori dello scopo di questo documento, per maggiori informazioni sull'uso di CA.pl come script, si faccia riferimento al sito http://www.openssl.org/docs/apps/CA.pl.html. L'output di questo sono due files, *newreq.pem* e *newcert.pem*, che andranno rinominati rispettivamente in *ldap-priv.pem* (la chiave privata) e *ldap-pub.pem*(la chiave pubblica). Vediamo la configurazione adottata nei nostro esempio in */etc/ldap/slapd.conf*:

```
# LDAP SEVER
# Configuration for: ldap.azienda.it
include         /etc/ldap/schema/core.schema
include         /etc/ldap/schema/cosine.schema
include         /etc/ldap/schema/nis.schema
include         /etc/ldap/schema/inetorgperson.schema
include         /etc/ldap/schema/krb5-kdc.schema
include         /etc/ldap/schema/samba.schema

# Schema check allows for forcing entries to
# match schemas for their objectClasses's
schemacheck     on
pidfile         /var/run/slapd.pid

# Where to store the replica logs
replogfile /var/lib/ldap/replog

# Read slapd.conf(5) for possible values
loglevel        0

## Kerberos support
sasl-realm AZIENDA.IT
sasl-host  ldap.azienda.it

## TLS support
TLSCACertificateFile /etc/ldap/cacert.pem
TLSCertificateFile /etc/ldap/ldap-pub.pem
TLSCertificateKeyFile /etc/ldap/ldap-priv.pem
TLSRandFile /dev/random

# The backend type, ldbm, is the default standard
database        ldbm

# The base of your directory
suffix          "dc=azienda,dc=it"

# Where the database file are physically stored
directory       "/var/lib/ldap"

## RootDN
rootdn          "cn=Directory Manager,ou=People,dc=azienda,dc=it"
rootpw          {KERBEROS}ldapadm@AZIENDA.IT

# Indexing options
index   objectClass                                     eq
```

```
index   cn,mail,surname,givenname                       eq,subinitial

# Save the time that the entry gets modified
lastmod on

# Include the access lists
include                 /etc/ldap/slapd.access
```

Rivediamo brevemente le parti salienti di questa configurazione. La prima parte del file riguarda la definizione degli *schema files* da includere, ovvero la sintassi degli attributi di LDAP. In particolare sottolineiamo la inclusione degli schemi relativi al *nis (nis.schema),* che serviranno ai sistemi Unix per la profilazione degli utenti (contengono gid/uid, home directory, shell, ecc ...), e *kerberos (krb5-kdc.schema),* che permetteranno di definire gli attributi per l'interfacciamento con il KDC. Successivamente i parametri *sasl realm* e *sasl host*, come si intuisce dal nome, si riferiscono al metodo di interfacciamento con le librerie SASL e definiscono il Kerberos realm il primo (AZIENDA.IT) e il principal relativo al server LDAP il secondo (che coincide con l'hostname, ovvero *ldap.azienda.it*). A seguire i parametri relativi al TLS per permettere di usare SSL con LDAP (protocollo LDAPS): *TLSCACertificateFile* specifica il file contenente la chiave pubblica della Certification Autority, *TLSCertificateFile* contiene la chiave pubblica del server LDAP in formato PEM, *TLSCertificateKeyFile* punta alla chiave privata del server ed infine *TLSRandFile* contiene il device */dev/random* per la generazione casuale dei files.

Proseguendo con la configurazione si specifica il *base dn* del nostro LDAP server, ovvero la radice del nostro albero LDAP, con il parametro *suffix* nel nostro caso impostato a *dc=azienda,dc=it*. Si identifica poi l'amministratore del directory, solitamente chiamato anche *Directory Manager*, che viene specificato con il parametro *rootdn*: da notare che è stato collocato sotto la *Organizational Unit* (OU) di *People*, anche se ancora non abbiamo ancora popolato i dati relativi alla struttura. A *Directory Manager* è stata associata, tramite il parametro *rootpw* la relativa password: in questo caso è molto particolare in quanto si riferisce a *{KERBEROS}ldapadm@AZIENDA.IT*. Proprio come si può intuire la keyword *KERBEROS* espressa tra parentesi graffe permette, in congiunzione all'opzione *-kpasswd* espressa durante la compilazione del server LDAP, di usare Kerberos come password server. Cosi' facendo, quando ci si collegherà al LDAP server come *Directory Manager*, andrà specificata la password di *ldapadm@AZIENDA.IT* (ovvero miapassword2, come specificato precedentemente). Durante l'inserimento dei dati vedremo che questa opzione vale anche per le password utente espresse nel parametro *userPassword*, ma è da notare che in successive queries al database LDAP il contenuto verrà offuscato tramite *base64*.

L'ultima opzione, ma non ultima come importanza, è l'inclusione di un file esterno per la definizione delle Access Lists: in LDAP è possibile definire delle ACL con lo scopo di limitare gli utenti a quali attributi possono accedere e/o modificare. Vediamo un access lists di esempio caricata nel file */etc/ldap/slapd.access:*

```
# Attributes visibile to everyone, but can be changed only by the
owner
access to attr=cn,givenName,sn,krbName,krb5PrincipalName,gecos
        by dn="cn=Directory Manager,ou=People,dc=azienda,dc=it" write
        by dn="uid=ldapadm.+\+realm=AZIENDA.IT" write
        by self write
        by * read

access to attr=loginShell,gecos
        by dn="cn=Directory Manager,ou=People,dc=azienda,dc=it" write
        by dn="uid=ldapadm.+\+realm=AZIENDA.IT" write
        by self write
        by * read

# Since wère using {KERBEROS}<PRINCIPAL>, we can't allow the user
# to change the password. They have to use the Kerberos 'kpasswd' to
# do this. But the admin can change (if need be).
# Please see krb5 userPassword attribute
access to attr=userPassword
        by dn="cn=Directory Manager,ou=People,dc=azienda,dc=it" write
        by dn="uid=ldapadm.+\+realm=AZIENDA.IT" write
        by anonymous auth
        by * none

# Directory Manager can do anything
access to *
        by dn="cn=Directory Manager,ou=People,dc=azienda,dc=it" write
        by dn="uid=ldapadm.+\+realm=AZIENDA.IT" write
        by * read
```

A questo punto è possibile avviare il server con:

```
# /usr/local/ldap/sbin/slapd -u ldap -g ldap -h "ldaps://0.0.0.0/"
```

Da notare che con l'opzione *-h* si è specificato quali protocolli e quali IP address il server LDAP deve ascoltare, in questo caso solo ed esclusivamente LDAPS. Inoltre è suggeribile avviare il server LDAP con un utente e gruppo diverso da root, in questo caso è stato avviato con l'utente *ldap* e gruppo *ldap*, creando appositamente un utente *ldap* con home directory */var/lib/ldap* (la directory del database). Proviamo quindi se LDAP funziona e se supporta i meccanismi di autenticazione preposti:

```
# ldapsearch -H "ldaps://localhost/" -x -b "" -s base -LLL
supportedSASLMechanisms
```

Il risultato dovrebbe essere come segue, facendo attenzione che ci sia GSSAPI:

```
supportedSASLMechanisms: PLAIN
supportedSASLMechanisms: LOGIN
supportedSASLMechanisms: ANONYMOUS
supportedSASLMechanisms: GSSAPI
```

A questo punto siamo pronti per popolare il database LDAP con i dati fittizi, a questo scopo useremo l'utility *slapadd* e un file temporaneo in cui scriviamo i dati, ad esempio */tmp/ldapentries.diff* come segue:

```
dn: cn=Mario Rossi, ou=People, dc=azienda,dc=it
objectClass: top
objectClass: person
objectClass: inetOrgPerson
objectClass: krb5Principal
objectClass: posixAccount
sn: Rossi
cn: Mario Rossi
givenName: Mario
mail: mrossi@azienda.it
krb5PrincipalName: mrossi@AZIENDA.IT
userPassword:: {KERBEROS}mrossi@AZIENDA.IT
uid: mrossi
uidNumber: 500
homeDirectory: /home/mrossi
gidNumber: 100

dn: ou=People,dc=azienda,dc=it
objectClass: top
objectClass: organizationalUnit
ou: People

dn: cn=Directory Manager,ou=People,dc=azienda,dc=it
objectClass: organizationalRole
cn: Directory Manager
description: Directory Manager

dn: dc=azienda,dc=it
objectClass: dcObject
objectClass: organization
dc: azienda
o: azienda.it
description: Root LDAP

dn: ou=Group,dc=azienda,dc=it
objectClass: top
objectClass: organizationalUnit
ou: Group

dn: cn=users,ou=Group,dc=azienda,dc=it
objectClass: posixGroup
objectClass: top
cn: users
gidNumber: 100
memberUid: mrossi

dn: ou=Computers,dc=azienda,dc=it
objectClass: top
objectClass: organizationalUnit
ou: Computers
```

Importiamo il file eseguendo il comando:

```
# slapadd -l /tmp/ldapentries.diff
```

Possiamo a questo punto provare ad effettuare la prima chiamata a LDAP con l'utente Directory Manager, ad esempio con il seguente comando

```
# ldapsearch -H "ldaps://localhost/" -b "dc=azienda,dc=it" -x -W -D
"cn=Directory Manager,ou=People,dc=azienda,dc=it" "objectclass=*"
```

Il comando dovrebbe restituire l'intero database LDAP. È possibile aggiungere anche una replica a LDAP per avere una maggiore alta affidabilità e disponibilità, si consiglia la lettura dell'ottimo How-To di Turbo Fredriksson LDAPv3 per un approfondimento.

Il Firewall

Come accennato in precedenza, uno degli obiettivi prioritari di questo laboratorio è di riflettere quanto possibile una realtà di produzione. A questo scopo è stato inserito nell'architettura un firewall, in modo da capire le problematiche relative all'aggiunta di un filtro IP. Nella tabella sottostante si riassumono le necessità di cominicazione dai client verso il KDC e i servizi Kerberos all'interno del firewall.

Tipo di Macchina	**Porta di destinazione**	**Descrizione**
KDC	88/udp 88/tcp	Kerberos 5 erogazione ticket
KDC	749/tcp	Kerberos 5, servizio kpasswd per cambio password e servizio di amministrazione remota
LDAP	636/tcp	LDAP con SSL per l'accesso ai dati utente

Nel nostro caso, queste necessità si declineranno in policy da applicare al firewall cerbero. Vediamo nella tabella sottostante queste policy:

Nome della macchina	**Porta di destinazione**	**Descrizione**
zeus.azienda.it	88/udp	Accesso alle funzionalità di Kerberos (autenticazione e cambio password)
	88/tcp	
	749/tcp	
zeus.azienda.it	636/tcp	LDAPS per lookup utenti
venere.azienda.it	80/tcp	Accesso al web server
venere.azienda.it	25/tcp	Accesso alla posta elettronica (SMTP ed IMAP)
	143/tcp	

6. SETUP DEI CLIENT

Con l'installazione di LDAP si è conclusa la parte relativa all'infrastruttura di base. Proseguendo con il nostro laboratorio dovremmo configurare i client, un Linux, un MacOS X ed un sistema Windows 2000, in modo tale che autentichino gli utenti su Kerberos. Il setup del client Linux è molto più complesso di quello Windows in quanto il primo è in grado di profilare gli utenti attraverso LDAP, caricandosi cioè le caratteristiche utente, mentre il secondo non è in grado di farlo attraverso un LDAP standard, ma solo attraverso Active Directory.

Linux

Per configurare il client Linux al fine di autenticare e profilare l'utente si è agito sul *Pluggable Authentication Module* (PAM) e sul *Name Service Switch* (NSS). Come accennato precedentemente, il laboratorio è stato effettuato con un client Linux, ma è possibile effettuare la stessa configurazione anche su tutti quei sistemi Unix che dispongono di entrambi i sottosistemi. Il sottosistema PAM si occupa di autenticare/autorizzare l'utente, ad esempio con la password, limitandolo ad esempio determinati orari o terminale, mentre NSS permette di recuperare le informazioni necessarie all'utente ed al sistema, come ad esempio gli attributi utenti e gli hosts. Sia Linux che altri ambienti Unix necessitano però delle estensioni necessarie al fine di usare PAM e NSS con Kerberos e LDAP: nel nostro caso useremo il modulo Kerberos per PAM e il modulo LDAP per NSS. Si poteva scegliere di usare un modulo LDAP per PAM per autenticare l'utente: il risultato di autenticazione si sarebbe ottenuto in entrambi i casi (forse con meno sforzo), ma non rientrerebbe nello scopo del laboratorio. Il modulo PAM Kerberos, al contrario di LDAP, permette di ottenere un TGT che consentirà successivamente di non immettere ulteriori utenze e password.

Tutte le distribuzioni Linux e molti vendor Unix includono tra i loro pacchetti i moduli Kerberos e LDAP per PAM e NSS, in Debian rispettivamente *libpam-krb5* e *libnss-ldap*. Nel caso si desiderasse, è possibile scaricare i sorgenti dal sito di

http://www.padl.com/ (modulo PAM e NSS per LDAP) e http://pam-krb5.sourceforge.net/ per il modulo PAM.

I file di configurazione del PAM si trovano generalmente in */etc/pam.d*, ognuno dei quali si riferisce ad un determinato servizio, come ad esempio il file *ssh* si riferisce all'omonimo servizio e contiene la metodologia di autenticazione. Alcune distribuzioni, come ad esempio RedHat, usano il modulo *pam_stack* per accentrare la configurazione dei vari servizi in un singolo file. Per praticità si descrive un file di configurazione completo, che contiene sia l'autenticazione che l'accounting e il cambio della password, che può essere usato come template per tutti i servizi (ad esempio creando dei symlink):

```
auth        sufficient    /lib/security/pam_krb5.so forwardable
auth        sufficient    /lib/security/pam_unix.so use_first_pass
likeauth nullok md5 shadow
auth        required      /lib/security/pam_deny.so
account     sufficient    /lib/security/pam_unix.so
account     required      /lib/security/pam_deny.so
password    required      /lib/security/pam_cracklib.so retry=3
password    sufficient    /lib/security/pam_krb5.so use_authok
password    sufficient    /lib/security/pam_unix.so use_first_pass
nullok use_authtok md5 shadow
password    required      /lib/security/pam_deny.so
session     required      /lib/security/pam_limits.so
session     sufficient    /lib/security/pam_krb5.so forwardable
session     required      /lib/security/pam_unix.so
```

Si copi questo files (o si creino i relativi symlink) almeno sui files *login* e *passwd* per permettere il login al prompt e il cambio della password. Prima di procedere con la configurazione del Name Service Switch, è necessario modificare i parametri di default relativi ad LDAP. Generalmente il file di configurazione è */etc/ldap.conf*, ma su Debian il file dei parametri per il plugin LDAP di NSS è */etc/libnss-ldap.conf.* Vediamo come andrebbe configurato per il nostro esempio:

```
# See ldap.conf(5) for details
# This file should be world readable but not world writable.

base dc=azienda,dc=it
uri ldaps://ldap.azienda.it
timelimit 10
bind_timelimit 2
scope sub
nss_base_passwd         ou=People,dc=azienda,dc=it?one
nss_base_shadow         ou=People,dc=azienda,dc=it?one
nss_base_group          ou=Group,dc=azienda,dc=it?one
```

A questo punto è possibile modificare la configurazione di NSS, ovvero */etc/nsswitch.conf*, inserendo i riferimenti a LDAP.

```
# Example configuration of GNU Name Service Switch functionality.
# If you have the `glibc-doc' and `info' packages installed, try:
# `info libc "Name Service Switch"' for information about this file.

passwd:         files ldap
group:          files ldap
shadow:         files

hosts:          files dns
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files

netgroup:       nis
```

Come test, da root, potremmo ad esempio provare a cercare l'utente *mrossi* definito in LDAP:

```
# id mrossi
uid=500(mrossi) gid=100(users) groups=100(users)
```

Il risultato viene estrapolato dal server LDAP, compreso i gruppi di appartenenza. Creiamo, a titolo di esempio, una home directory per l'utente. In un ambiente complesso si consiglia di usare NFS con l'automount o anche il mouting della home directory Windows/Samba attraverso un apposito modulo PAM chiamato *pam_mount*.

```
# mkdir /home/mrossi
# chown mrossi:users /home/mrossi
```

Prima di provare il logon, sul server *zeus,* creiamo il principal corrispondente all'utente *mrossi*, in modo da verificare username e password, ed un principal relativo alla workstation *kirk*, che servirà alla workstation per validare il ticket dell'utente con il KDC:

```
# kadmin.local -q "addprinc -pw miapassword3 mrossi@AZIENDA.IT"
# kadmin.local -q "addprinc -pw miapassword4
host/kirk.azienda.it@AZIENDA.IT"
# kadmin.local -q "ktadd -k /tmp/kirk.key
host/kirk.azienda.it@AZIENDA.IT"
```

L'ultimo comando crea un file */tmp/kirk.key* contenente la chiave Kerberos corrispondente alla workstation Linux: è necessario trasferire in modo sicuro (ad esempio tramite ssh) la chiave da *zeus* a *kirk* e copiare la stessa sul file */etc/krb5.keytab*.

Siamo a questo punto pronti per provare il log-in con l'utente *mrossi*. In un nuovo virtual terminal del client Linux, digitiamo la username e password corrispondente. Se tutto funziona correttamente, ci troveremo loggati sul sistema con una shell ed inoltre avremmo a disposizione un TGT con cui poterci collegare ad altre macchine, o meglio ad altri servizi. Vediamo il nostro TGT con l'utility

klist:

```
$ klist
Ticket cache: FILE:/tmp/krb5cc_500
Default principal: mrossi@AZIENDA.IT

Valid starting     Expires            Service principal
03/10/04 18:13:40  03/11/04 04:13:07  krbtgt/AZIENDA.IT@AZIENDA.IT

Kerberos 4 ticket cache: /tmp/tkt500
klist: You have no tickets cached
```

Windows 2000/XP

Contrariamente a Linux, dove il modulo per Kerberos è da installare, su Windows 2000 e superiori invece è già parte del sistema operativo, in quanto è alla base dell'autenticazione Microsoft. Per fare il setup del Kerberos di Windows come workstation invece è necessario avere il *Resource Kit*, in quanto contiene delle utilities a riga di comando che ci serviranno per il set-up, in particolare i programmi *ksetup.exe*, *klist.exe* e opzionalmente *ktpass.exe*. Per quanto riguarda LDAP, abbiamo precedentemente accennato che Windows non è in grado di rivolgersi direttamente ad un LDAP server per scaricare il profilo utente, in quanto il servizio di autenticazione Microsoft, seppur usando un backend LDAP (Active Directory), viene effettuato attraverso NetBIOS over TCP/IP.

Non potendo cercare le caratteristiche dell'utente su LDAP, Windows ne cerca le caratteristiche sul database utente locale, in quanto è necessario mappare in qualche modo il nome utente ad un determinato *Security ID* (detto SID), questa mappatura viene effettuata al set-up dell'ambiente Kerberos. Esistono due diversi approcci: la prima è quella di mappare ad ogni utente kerberos un analogo utente locale; la seconda è quella di creare un utente "contenitore" e di associare tutti gli utenti kerberos su un singolo account locale. Il primo approccio è utile soprattutto per le macchine personali, ad esempio un portatile o una scrivania su cui sicuramente siederà una determinata persona. Il secondo è più adatto per gli utenti *roaming*, ovvero per quelli che hanno le scrivanie o i computer in condivisione, ad esempio per le persone che non sono spesso in ufficio.

L'approccio proposto sarà di sfruttare il primo scenario, ovvero di associare un utente Kerberos ad un utente locale, ad esempio il nostro Mario Rossi che usa principalmente la workstation windows (nel nostro laboratorio *picard.azienda.it*). A questo punto è necessario creare con *Local Users and Groups* gli utenti locali, nel nostro caso *mrossi*; si assume che si sappia creare gli utenti attraverso i tools di sistema. Prima di provare il logon, sul server *zeus,* creiamo il principal corrispondente alla workstation Windows:

```
# kadmin.local -q "addprinc -pw miapassword5
host/picard.azienda.it@AZIENDA.IT"
```

Attenzione: non bisogna riusare un principal esistente, ad esempio uno già

creato per ambiente Unix, ma è necessario crearne uno nuovo con un hostname differente. Vedremo in seguito nel paragrafo "Gestire il dual boot Linux/Windows" quale sia il problema e come risolverlo. Da riga di comando, e con i tools del Resource Kit nella directory corrente, procedere come segue:

```
C:\> net time /sntp:time.azienda.it
C:\> ksetup /SetDomain AZIENDA.IT
C:\> ksetup /SetMachPassword miapassword5
C:\> ksetup /AddKdc AZIENDA.IT kdc.azienda.it
C:\> ksetup /AddKpasswd AZIENDA.IT kdc.azienda.it
C:\> ksetup /mapuser * *
```

Vediamo adesso il significato dei comandi. Il primo comando permette di aggiungere un server NTP per la sincronizzazione dell'orologio di sistema, per evitare il fenomeno di disallineamento denominato *clockskew* (vedi Kerberos). Il secondo definisce il realm di default, mentre il terzo imposta la password della workstation Windows associato al principal creato in precedenza: si ricorda che il nome host del principal si ottiene mediante il nome NetBIOS della workstation aggiunto al dominio di default del realm specificato. Il quarto e il quinto comando definiscono esplicitamente il KDC e il server di riferimento per il cambio delle password relativo al realm AZIENDA.IT. Infine, il sesto si riferisce alla mappatura degli utenti come avevamo determinato precedentemente.

A questo punto è necessario un reboot per applicare le impostazioni del sistema Kerberos. Una volta riavviato, nella schermata di Log-On grafico (*GINA*), tramite la drop-down combo, sarà possibile scegliere *AZIENDA.IT (Kerberos Realm)* tra le opzioni ed immettere quindi la username e password Kerberos (esempio in fig. 4).

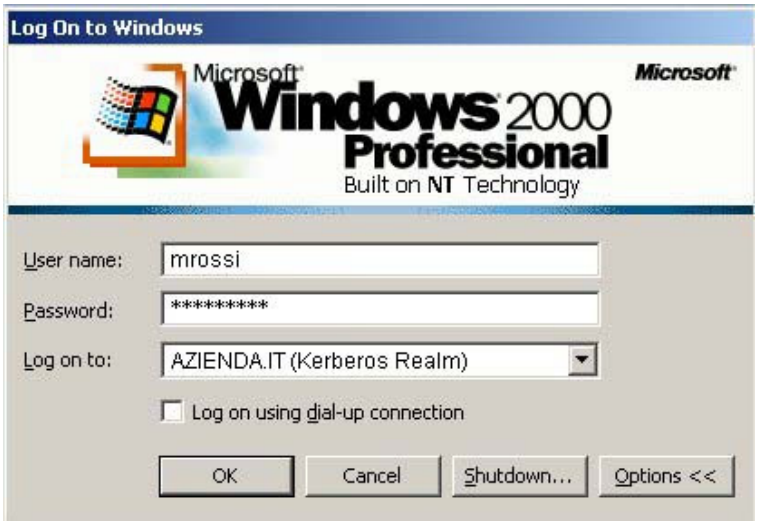


Figura 4 - Login di Windows 2000

Già attraverso il logon abbiamo verificato lo scambio di informazioni tra la workstation Windows e il KDC. Come ulteriore prova, attraverso il tool *klist.exe* potremmo elencare i nostri tickets:

```
C:\>klist tickets

Cached Tickets: (2)

   Server: krbtgt/AZIENDA.IT@AZIENDA.IT
      KerbTicket Encryption Type: Kerberos DES-CBC-CRC
      End Time: 3/19/2004 5:30:57
      Renew Time: 3/25/2004 19:30:57


   Server: host/voyager.azienda.it@AZIENDA.IT
      KerbTicket Encryption Type: Kerberos DES-CBC-CRC
      End Time: 3/19/2004 5:30:57
      Renew Time: 3/25/2004 19:30:57
```

Oppure visualizzare il TGT rilasciato dal nostro KDC:

```
C:\>klist tgt

Cached TGT:

ServiceName: krbtgt
TargetName: krbtgt
FullServiceName: mrossi
DomainName: AZIENDA.IT
TargetDomainName: AZIENDA.IT
AltTargetDomainName: AZIENDA.IT
TicketFlags: 0x40e00000
KeyExpirationTime: 1/1/1601 1:00:00
StartTime: 3/18/2004 19:30:57
EndTime: 3/19/2004 5:30:57
RenewUntil: 3/25/2004 19:30:57
TimeSkew: 1/1/1601 1:00:00
```

Mac OS X

Durante la stesura dei requisiti si è considerato la maggioranza dei sistemi attualmente installati presso le aziende italiane. Visto la crescente diffusione dei sistemi Apple dovuta principalmente a MacOS X, uno Unix "user friendly" con interfaccia grafica Mac, si è deciso di prendere in considerazione anche questi sistemi, anche se non si erano identificati inizialmente come piattaforma primaria.

In laboratorio si è usato MacOS X 10.3, cui nome in codice è Panther, che differisce lievemente dalla release precedente 10.2. Per coloro che dispongono della versione 10.2, si consiglia la lettura dei documenti AppleCare numero 107154 intitolato "Mac OS X 10.2: How to Enable Kerberos Authentication for Login Window" che potrete trovare al link http://docs.info.apple.com/article.html?artnum=107154 e del documento "Mac

OS X 10.2: About Using Kerberos" (AppleCare 107157) strettamente correlato al precedente e disponibile all'indirizzo http://docs.info.apple.com/article.html?artnum=107153.

Essendo fondamentalmente un sistema Unix, le stesse regole applicate per Linux valgono nellamaggior parte dei casi anche per MacOS X. Su Panther esiste una utility, il *kerberosautoconfig*, che configura buona parte della configurazione Kerberos. Nel nostro caso procederemo con:

```
# sudo kerberosautoconfig –r AZIENDA.IT –m kdc.azienda.it
```

Questa utility configurerà automaticamente il file di sistema */Library/Preferences/edu.mit.Kerberos*, corrispondente al file */etc/krb5.conf* su altri sistemi Unix. Bisogna quindi creare un principal relativo al sistema Apple sul KDC come segue:

```
# kadmin.local -q "addprinc -pw miapassword4a
host/spok.azienda.it@AZIENDA.IT"
# kadmin.local -q "ktadd -k /tmp/spok.key
host/spok.azienda.it@AZIENDA.IT"
```

L'ultimo comando crea un file */tmp/spok.key* contenente la chiave Kerberos corrispondente alla workstation MacIntosh: è necessario trasferire in modo sicuro (ad esempio tramite ssh) la chiave da *zeus* a *spok* e copiare la stessa sul file */etc/krb5.keytab*.

L'ultima configurazione relativa a Kerberos è quella del login: il Mac OS X Security and Authorization Services usa il file */etc/authorization*, e questo è il file che useremo per usare Kerberos nella finestra di login (*loginwindow*). MacOS X può operare con e senza host keytab, ma si consiglia di sfruttare il Kerberos keytab appena creato per difendersi da un attacco di tipo man-in-the-middle contro il sistema Kerberos. Per permettere una login con un utente Kerberos con un host principal è necessario ricercare la proprietà *system.login.console* nel file */etc/authorization* e modificarla come segue:

```
<key>system.login.console</key>
<dict>
          <key>class</key>
          <string>evaluate-mechanisms</string>
          <key>comment</key>
          <string>Login mechanism based rule. Not for general use,
yet. builtin:krb5authenticate can be used to hinge local
authentication on a successful kerberos authentication and kdc
verification. builtin:krb5authnoverify skips the kdc verification.
Both fall back on local authentication.</string>
          <key>mechanisms</key>
          <array>
                    <string>loginwindow_builtin:login</string>
                    <string>authinternal</string>
                    <string>loginwindow_builtin:success</string>
                    <string>builtin:getuserinfo</string>
                    <string>builtin:sso</string>
                    <string>builtin:krb5authenticate</string>
</array>
</dict>
```

Si noti l'ultima espressione *builtin:krb5authenticate* che permette sia l'autenticazione delle utenze attraverso Kerberos, che il rilascio al logon di un TGT. L'ultimo passo della configurazione è relativa al collegamento con l'LDAP server, che permette di ricavare le informazioni relative all'utente. Sul Finder, aprire la cartella *Applicazioni*, quindi *Utilities* ed eseguire l'applicazione *Directory Access*. Si verrà posizionati nella sezione *Services*: selezionare solo l'opzione *LDAPv3* come segue in fig. 5.

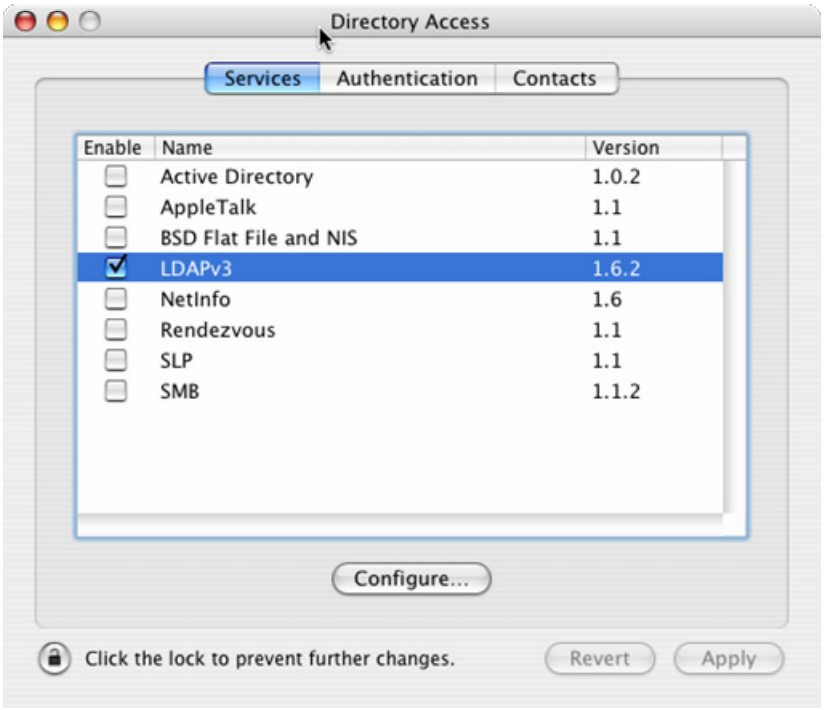


Figura 5 - MacOS X Directory Access, Selezione LDAP

.

Selezionare *LDAPv3* e premere *Configure*. Deselzionare l'opzione "Use DHCP-supplied LDAP Server" come in fig. 6, quindi premere il pulsante accanto a *Show Options* per selezionare l'intera gamma di opzioni.

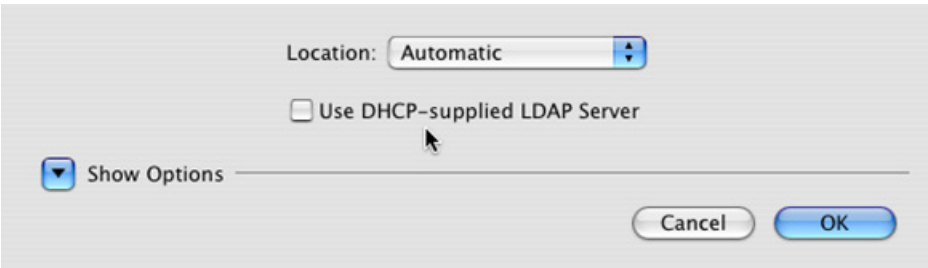


Figura 6 - MacOS X Directory Access, deselezione DHCP Supplied LDAP

Premere il pulsante *New...* (Fig. 7) e inserire AZIENDA.IT come parametro nella

sezione *Configuration Name.*

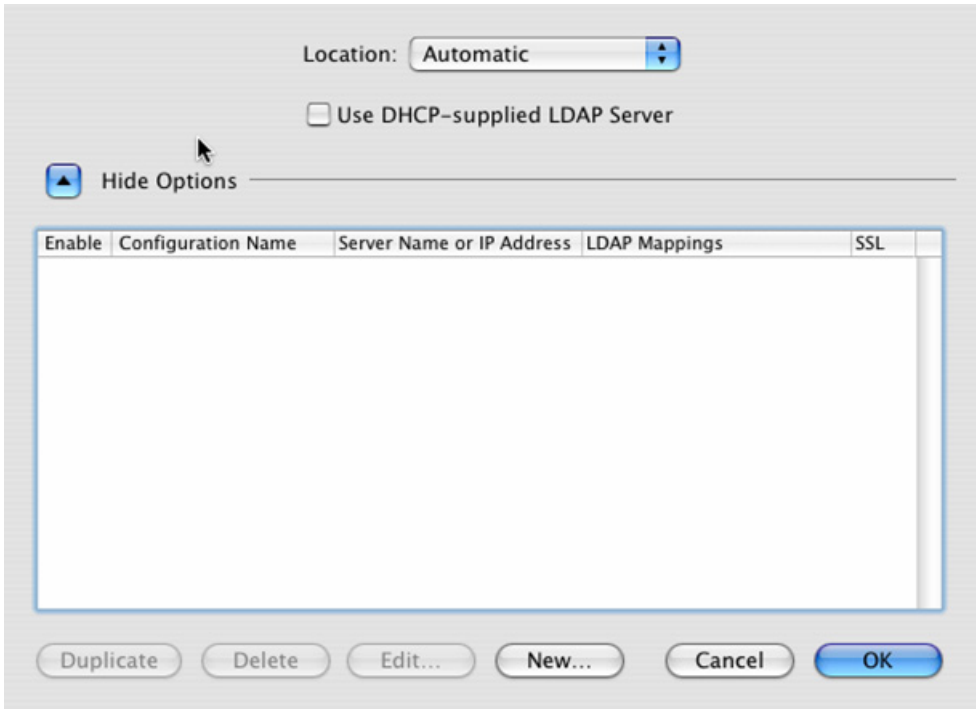


Figura 7 - MacOS X Directory Access, visualizzazione LDAP server

Selezionare la riga corrispondente come in fig. 8 e premere quindi il pulsante *Edit...*

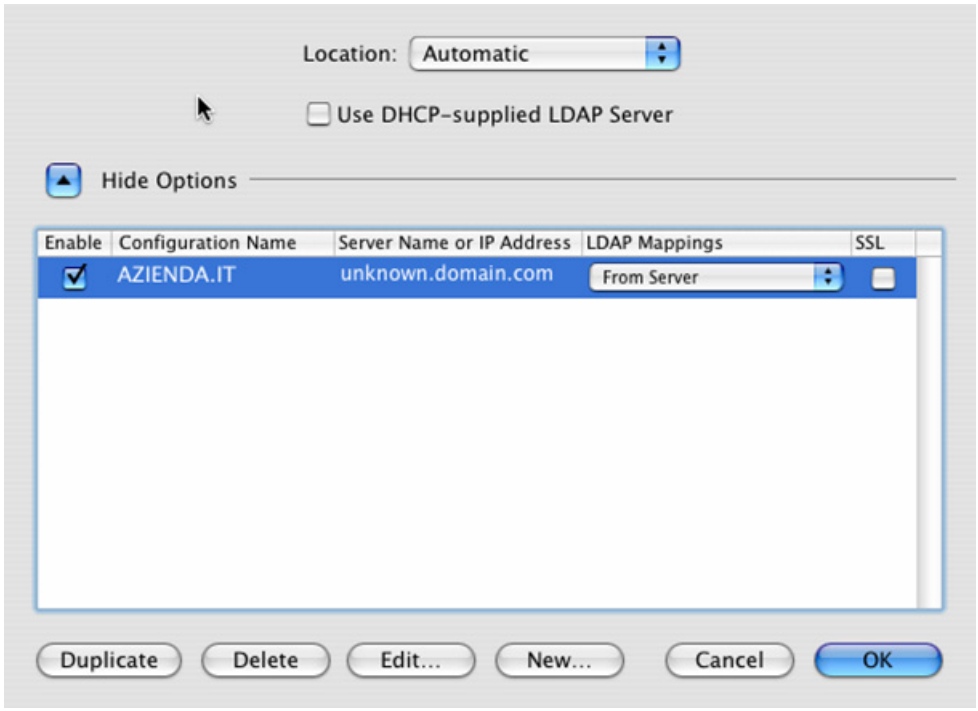


Figura 8 - MacOS X Directory Access, creazione voce LDAP

Verrà automaticamente selezionata la sezione *Connection*. Inserire in *Server Name or IP Address* il nome del nostro LDAP server, ovvero *ldap.azienda.it*. Per mantenere la privacy, attivare l'opzione *Encrypt using SSL* (Fig. 9).

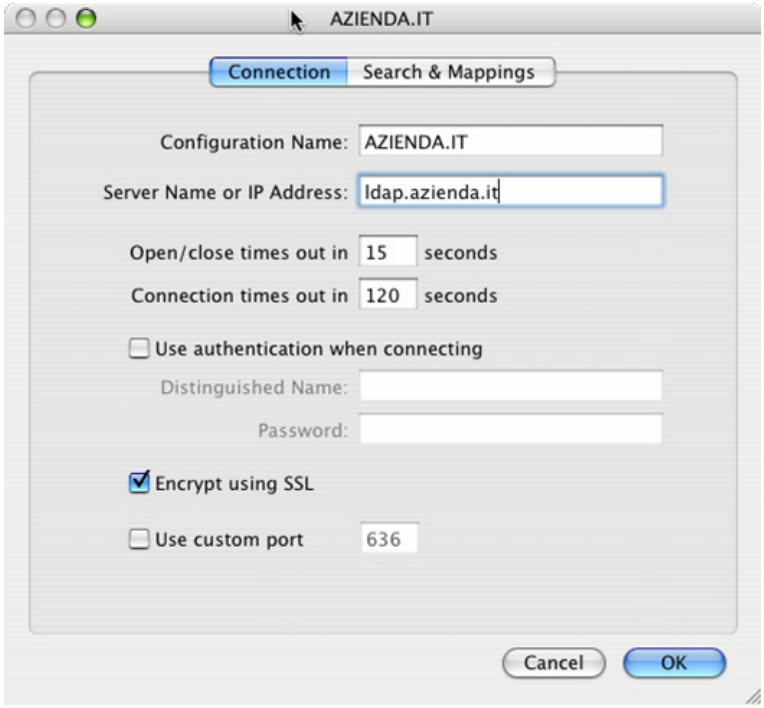


Figura 9 - MacOS X Directory Access, configurazione LDAP di dettaglio

Selezionare quindi la sezione *Search & Mappings*. Successivamente selezionare da *Access this LDAPv3 server using* il valore *RFC 2307 (Unix)*. Verrà quindi presentata una finestra di dialogo intitolata *Search Base Suffix*, inserire il BaseDN del nostro LDAP server, ovvero *dc=azienda,dc=it* (Fig. 10).

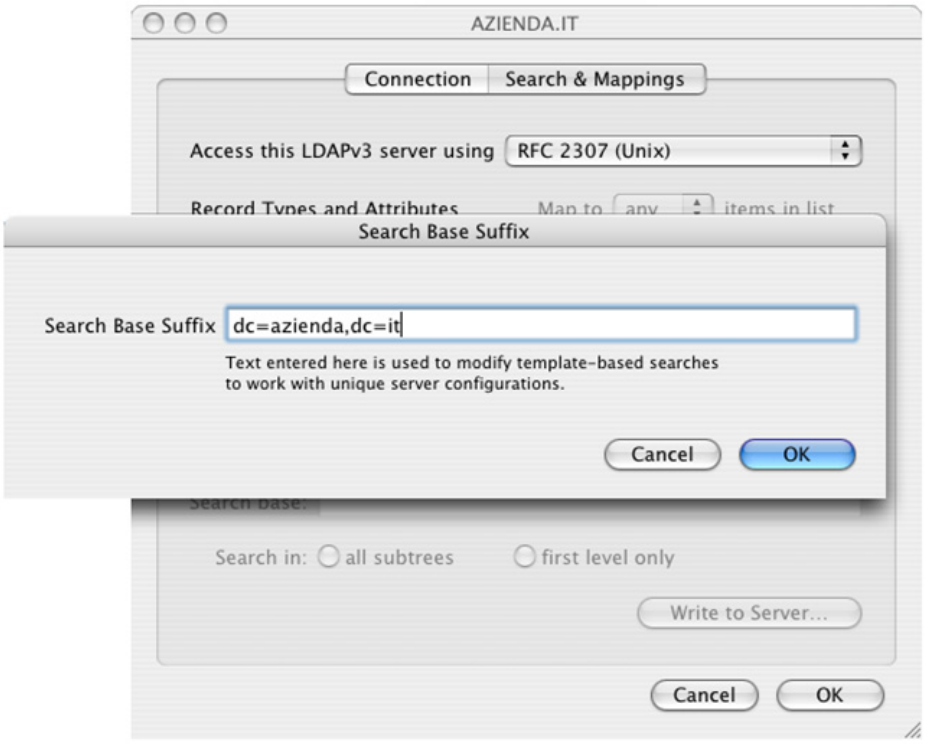


Figura 10 - MacOS X Directory Access, specifica Base DN

Premere quindi *Ok*. Ritornati alla finestra con la configurazione, premere *Ok*. Selezionare quindi la sezione *Authentication* del Directory Access (Fig. 11) e quindi il pulsante *Add...*

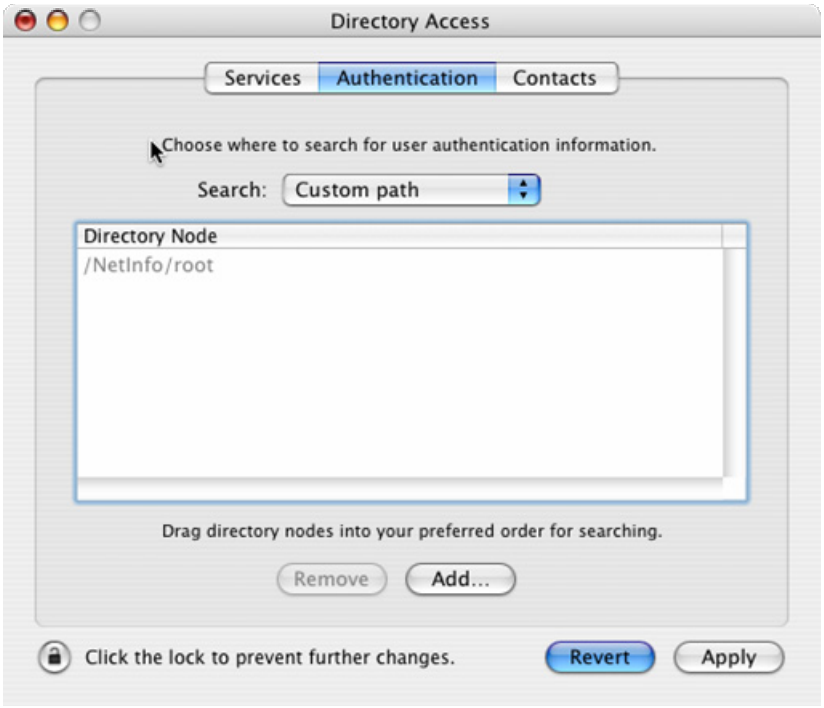


Figura 11 - MacOS X Directory Access, configurazione autenticazione

Verrà presentata una finestra con un elenco di directory disponibili. Selezionare dall'elenco */LDAPv3/ldap.azienda.it* e quindi premere *Add* come in fig. 12.

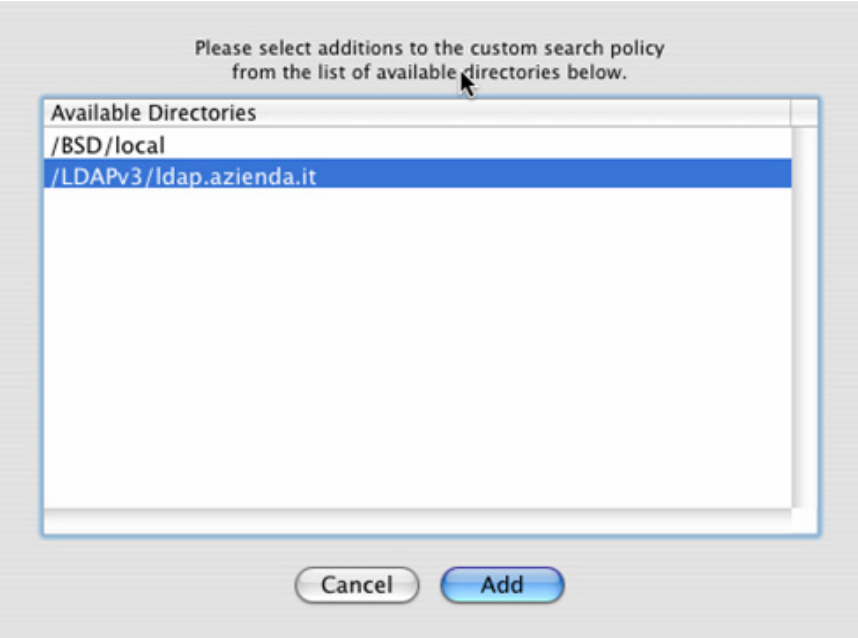


Figura 12 - MacOS X Directory Access, aggiunta LDAP all'autenticazione

Premere quindi Apply. A questo punto siamo pronti per poter entrare con l'utente *mrossi*. Fare quindi logout, selezionare *Other...* e immettere come utente *mrossi* e la password associata. Et voilà, siamo perfettamente collegati al sistema MacOS X con il nostro utente. Se apriamo la finestra di terminale e digitiamo *klist* vedremo il TGT ottenuto dal KDC.

Anche se MacOS X 10.2 (Jaguar) e MacOS X 10.3 (Panther) hanno molte porzioni del software Kerberos del MIT, esso però non supporta applicazioni di terze parti (ad esempio Eudora) per l'autenticazione attraverso Kerberos. A questo scopo è necessario scaricare i *MacOS X Kerberos Extras* direttamente dal MIT che provvedono a creare appositi link in */Applications/Utilities* (Kerberos di solito è installato in */System/Library/CoreServices*) e ad installare un file di configurazione di esempio. Questo tool può essere scaricato al seguente indirizzo: http://web.mit.edu/macdev/Download/Mac_OS_X_Kerberos_Extras.dmg

Per maggiori informazioni su questi tools, consultare il sito: http://web.mit.edu/macdev/Development/MITKerberos/Common/Documentation/osx-kerberos-extras.html

Altri Unix

Per dovere di completezza, si elencano i collegamenti ai siti di altri produttori Unix che supportano PAM ed in cui è possibile effettuare un'autenticazione attraverso Kerberos e LDAP. Si ricorda che altri sistemi Unix al di fuori di quelli elencati nelle precedenti pagine non sono stati testati.

HP-UX
http://www.hp.com/products1/unix/operating/security

Solaris
http://wwws.sun.com/software/solaris/pam

Linux
http://www.kernel.org/pub/linux/libs/pam

FreeBSD
http://www.freebsd.org/doc/en_US.ISO8859-1/articles/pam

AIX
http://publib16.boulder.ibm.com/pseries/en_US/aixbman/security/pam_overview.htm

Per maggiori informazioni sulla lista di moduli PAM esistenti, si consiglia di consultare il seguente sito Internet: http://www.kernel.org/pub/linux/libs/pam/modules.html.

Gestire il dual boot Unix/Windows

Nel paragrafo "Windows 2000/XP" di questo capitolo si è accennato al fatto che non è possibile usare lo stesso principal Kerberos sia per ambienti Unix che ambienti Windows. Il problema derivato da non poter riusare lo stesso principal, quindi lo stesso hostname, è particolarmente sentito in un ambiente dual boot, in cui l'hostname della macchina è identico sia per l'ambiente Unix che per l'ambiente Windows.

Il problema è dovuto ad un parametro della chiave di Kerberos, ovvero il *Key Version Number*, conosciuto anche come *KVNO*, che viene incrementato ogni qual volta viene apportata una modifica alla chiave. Questo parametro viene controllato dal KDC all'atto dell'ottenimento del TGT, insieme ovviamente alla chiave simmetrica ed al clockskew. Esportare la chiave per Unix con l'opzione *ktadd* della utility *kadmin* incrementa il KVNO, mentre Windows presuppone che il KVNO sia impostato al valore 1: questa è l'incongruenza che fa sì che non si

possa usare la stessa chiave sia per Windows che per Unix in un ambiente dual boot.

Esiste comunque un modo per aggirare il problema: creare un keytab sul sistema Unix senza esportarla dal KDC, partendo quindi dalla password usata durante la creazione del principal sul KDC, in maniera simile a quello che avviene con Windows.

Ipotizziamo di aver installato una doppia partizione Linux/Windows sul sistema *picard* e che il sistema sia acceso e funzionante in Linux. Una volta collegatosi con l'utente root, invocare la utility *ktutli*, e dal prompt della stessa procedere come segue:

```
# ktutil
addent -password -p host/picard.azienda.it@AZIENDA.IT -k 1 -e des-
cbc-crc
(verrà chiesta la password, inserire quella relative al principal di
picard, ovvero miapassword5)
wkt /etc/krb5.keytab
```

Vediamo in dettaglio le operazioni. Il comando *addent* crea in memoria una chiave Kerberos attraverso una password (opzione *–password*) relativa al principal *host/picard.azienda.it@AZIENDA.IT* con KVNO di 1 (opzione *–k*) e con la encryption della chiave compatibile con il KDC (opzione *–e*). Il secondo comando, *wkt*, scrive sul keytab della macchina la chiave appena creata in memoria. È bene ricordarsi che vanno assegnati i permessi di scrittura/lettura al solo utente di root sul keytab, mentre nessun permesso per gli altri utenti.

7. L'ACCESSO INTERATTIVO

Ora che il setup delle workstation è stato completato, possiamo passare a dimostrare la vera forza di Kerberos, ovvero l'uso di servizi di rete senza mostrare altre credenziali. L'obiettivo di questo capitolo è infatti dimostrare l'efficacia di Kerberos accedendo in maniera interattiva con Windows, Linux e MacOS sia ad un server Unix attraverso SSH che ad un Cisco attraverso telnet.

L'accesso ad un server Unix con SSH

Ormai molti sistemi Unix hanno SSH, molti però non dispongono delle funzionalità Kerberos perchè non compilati con questa opzione. SSH con Solaris 9, ad esempio, dispone di un server OpenSSH già incluso nella distribuzione compilato con le estensioni GSSAPI, mentre in Debian Woody esistono due pacchetti ssh, di cui il ssh-krb5 ha il supporto GSSAPI incluso. È necessario pertanto verificare che la versione del server e del client SSH incluso con la propria distribuzione Linux o Unix disponga del supporto GSSAPI. Qualora questo non fosse disponibile, è possibile scaricare i sorgenti del programma OpenSSH dal sito http://www.openssh.org, e compilandolo come segue:

```
# ./configure --prefix=/usr/local --with-tcp-wrappers --with-pam \
--with-privsep-user=sshd --with-dns --with-kerberos5=/usr/kerberos \
--with-privsep-path=/var/empty --with-4in6
```

L'abilitazione a kerberos è data dall'opzione --with-kerberos5 ed è necessario modificare il path /usr/kerberos con la directory base dell'installazione Kerberos. Non spiegheremo come configurare SSH al di fuori dell'integrazione con Kerberos, in quanto fuori dallo scopo di questo documento, consigliamo la lettura delle man pages corrispondenti a sshd_config(5). SSH cosi' modificato va installato sia sul

server, nel nostro caso venere.azienda.it, che sul client. In particolare, sul server è necessario aggiungere le seguenti righe nel file di configurazione /usr/local/etc/sshd_config:

```
KerberosAuthentication yes
KerberosOrLocalPasswd yes
GssapiAuthentication yes
```

Prima di avviare il daemon sshd, è necessario creare ed importare la chiave di Kerberos nel keytab della macchina, analogamente a quanto fatto su *kirk* per l'accesso di login. Sul server *zeus* creare il principal ed esportare il keytab come segue:

```
# kadmin.local -q "addprinc -pw miapassword7
host/venere.azienda.it@AZIENDA.IT"
# kadmin.local -q "ktadd -k /tmp/venere.key
host/venere.azienda.it@AZIENDA.IT"
```

A questo punto è necessario trasferire la chiave da zeus a venere e copiare la stessa sul file */etc/krb5.keytab*. Avviare quindi il daemon sshd sul server venere e controllare che sia attivo e senza errori.

Collegamento da Linux

Il collegamento da Linux è piuttosto semplice. Prima di tutto è necessario aver copiato i files relativi ad SSH (i files binari installati in /usr/local), appena compilati nel server, anche nel client per permettere l'uso delle estensioni GSSAPI. Una volta copiato i files, ci si collega al sistema client (kirk) con l'utente mrossi, e verificato di aver ricevuto il TGT dal KDC attraverso il comando klist, si procede ad usare il client SSH come segue:

```
$ ssh venere.azienda.it
```

Et voilà: se il set-up dell'ambiente è stato effettuato correttamente, saremo collegati al server venere.azienda.it senza inserire nè username e password. La "magia" di Kerberos è avvenuta!

Collegamento da Windows

Analogamente a Linux, ci collegheremo a Windows con l'utente mrossi, avendo cura di scegliere il logon al realm AZIENDA.IT. Windows non dispone di un client ssh nativo, pertanto è necessario scaricare un client SSH che dispone dell'autenticazione GSSAPI/SSPI: durante le prove è stato provato il programma shareware SecureCRT 4.1

(http://www.vandyke.com/products/securecrt/index.html), che dispone di tale autenticazione. Esiste una versione modificata del famoso client freeware Putty (http://www.certifiedsecuritysolutions.com/downloads.html) che consente l'autenticazione atttraverso GSSAPI, ma si è preferito il primo per le maggiori funzionalità.

Una volta installato, per configurare il client scegliere dal menù File, la voce Quick Connect. Questa opzione avvierà una dialog-box, scegliere nel pannello a sinistra come Protocol il valore SSH, come hostname il server di destinazione (nel nostro caso *venere.azienda.it*) e lasciare 22 come porta. Inserire l'utente di logon, nel nostro caso *mrossi*, e come authentication nella voce primary selezionare GSSAPI e come secondary il valore password, in modo tale che ci verrà chiesta la password qualora l'autenticazione Kerberos dovesse fallire. Premere quindi il bottone *Connect* (Fig. 13).

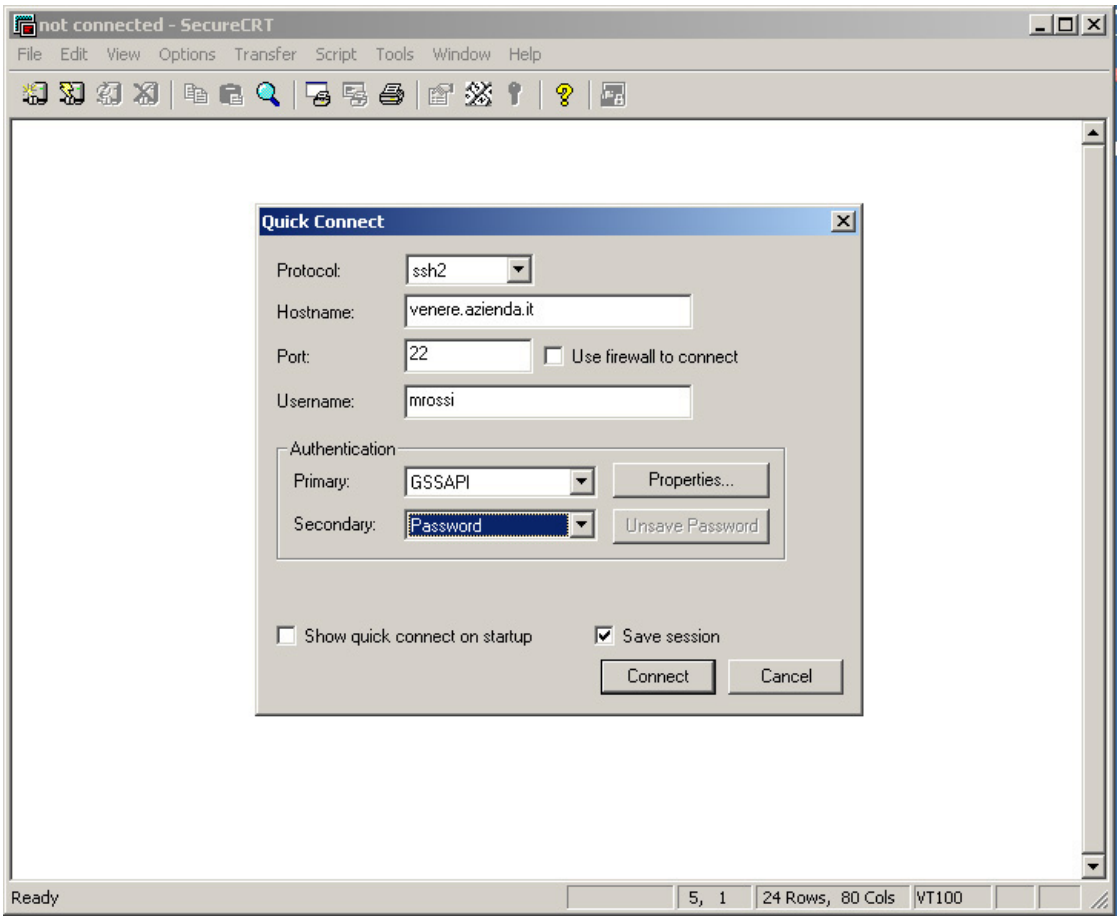


Figura 13 – Configurazione di SecureCRT per l'uso di GSSAPI

Anche in questo caso, Kerberos ci ha permesso il login al server venere.azienda.it senza richiedere di inserire le nostre credenziali.

Collegamento con MacOS X

Ancor più semplice è il collegamento con MacOS X: il sistema già dispone di una versione di SSH compilata con kerberos. Una volta verificato a riga di comando l'ottenimento del TGT con *klist*, sarà sufficiente collegarsi al sistema remoto con:

```
$ ssh venere.azienda.it
```

Limitazione degli account

In uno scenario quale quello presentato appare immediatamente evidente che ogni account presente sul Realm e sul repository LDAP possa accedere in maniera interattiva indistintamente ad ogni server della nostra azienda. In molti casi è necessario limitare l'accesso interattivo a determinate categorie di utenti, ad esempio su di un web server vogliamo che solo gli amministratori di sistema riescano ad accedere al server. A questo scopo ci aiutano i parametri di SSH *AllowUsers* e *AllowGroups* che limitano l'accesso rispettivamente ad una determinata lista di utenti il primo e uno o più gruppi il secondo. Per maggiori informazioni, si invita a fare riferimento alle man pages sshd_config(5).

L'accesso ad un Cisco con telnet

Anche Cisco supporta nel suo IOS la possibilità di autenticarsi con Kerberos 5. Questa modalità serve da un lato il fatto di acquisire un TGT quando l'utente fa log-on al sistema IOS (ad esempio attraverso una linea seriale), ma soprattutto per permettere una sessione telnet utente senza digitare username e password. Di seguito un esempio di configurazione Kerberos con IOS:

```
aaa new-model
aaa authentication login default krb5-telnet krb5 enable
aaa authorization exec krb5-instance
kerberos local-realm AZIENDA.IT
kerberos srvtab entry host/router.azienda.it 0 891725446 4 1 8
012345678901234567
kerberos server AZIENDA.IT 192.168.0.10
kerberos instance map admin 15
```

Questa configurazione permette ad un utente con un client telnet kerberizzato di autenticarsi attraverso il TGT ed, in caso fallisse, di verificare attraverso la password di enable. Il parametro instance map congiuntamente con la linea authorization exec permette alle persone con l'instanza di admin di accedere all'IOS direttamente nella shell di enable.

Guardando la configurazione, è importante capire i numeri che seguono il principal host/router.azienda.it.: il primo è il principal type, il timestamp (in secondi da 1970), il key version number (4), il keytype (1 = des), key length (sempre 8 per des), e quindi la chiave.

A questo punto, è possibile collegarsi da Linux con il telnet, assicurandosi di eseguire il client telnet corretto: alcune distribuzioni differenziano il telnet standard da quello kerberizzato, ad esempio RedHat posiziona il telnet in */usr/kerberos/bin/telnet*. Windows 2000 e MacOS X, invece, dispongono già delle estensioni Kerberos nei loro client telnet nativi.

8. L'AUTENTICAZIONE WEB

Una delle cose sorprendenti dell'integrazione realizzata da Microsoft è relativa all'autenticazione degli utenti che hanno fatto log-on al dominio Active Directory sugli applicativi Web. Attraverso Internet Information Server (IIS) e il web browser Microsoft Internet Explorer, l'utente viene riconosciuto trasparentemente, realizzando cosi' un vero e proprio Single Sign-On.

La tecnologia usata per effettuare un login dell'utente attraverso il Web è SPNEGO, già descritto nel paragrafo *SPNEGO e l'autenticazione Web*. Grazie allo sforzo della comunità OpenSource, è possibile utilizzare SPENGO anche su Unix, sia come client che come server web. In questo capitolo spiegheremo come è possibile realizzare con Apache l'autenticazione attraverso SPNEGO e Mozilla come client di autenticazione, oltre a vedere più da vicino l'utilizzo di Microsoft Internet Explorer

Apache sotto Unix

La scelta del Web server da utilizzare per l'autenticazione è ricaduta sul popolarissmo Apache (http://httpd.apache.org/), in quanto è il web server più usato su Internet. Sebbene non si tratti di un requisito fondamentale, si è scelto di installare anche PHP, uno scripting language molto diffuso, che ci permetterà in seguito di poter presentare al lettore un esempio il programma. Si assume che il web server Apache sia stato correttamente installato e configurato, e che risponda correttamente ad un browser. Si assume inoltre che il development kit (gli include files e il tool *apxs*) per compilare i moduli aggiuntivi sia installato (per Debian è il pacchetto *apache-dev*)

Per effettuare l'autenticazione Kerberos/SPNEGO si è utilizzato il modulo

mod_auth_kerb scaricabile dal sito Internet http://modauthkerb.sourceforge.net/. Una volta scompattato, si è proceduto alla compilazione del sorgente con il seguente comando:

```
# ./configure --with-krb5=/usr --with-apache=/usr
# make
# make install
```

Dove *--with-krb5* rappresenta la directory base contenente gli include files di Kerberos, mentre *--with-apache* la directory base con gli include files di Apache e il tool *apxs*. Successivamente il *make* compilerà il modulo e *make install* lo installerà nella directory dei moduli.

A questo punto era necessario creare una nuova directory che dovrà essere protetta dall'accesso non autorizzato ed inserirvi una pagina di esempio. A partire dalla directory dei documenti di Apache (*DocumentRoot*), nel nostro caso */var/www/*, si è creata un'ulteriore directory chiamata *protected.* In questa directory è stato creato un file *index.php* contenente la seguente linea:

```
<? phpinfo() ?>
```

Questa istruzione fa vedere tutte le informazioni relative a PHP e al web server su cui sta girando: ci servirà come pagina di test. Successivamente, è necessario configurare il web server per caricare il modulo compilato e configurare la protezione sulla directory */protected*, aggiungendo al file di configurazione *httpd.conf* le seguenti linee:

```
# Load Kerberos module
LoadModule auth_kerb_module /usr/lib/apache/1.3/mod_auth_kerb.so

# Protected directory
<Directory /var/www/protected>
    Options Indexes SymLinksIfOwnerMatch

    AuthType Kerberos
    AuthName "AZIENDA.IT Login"
    Krb5Keytab /etc/apache/krb5.keytab
    KrbAuthRealms AZIENDA.IT
    require valid-user
</Directory>
```

La prima riga fa si che il modulo di autenticazione venga caricato all'interno di Apache. Successivamente si definisce la directory protetta e si specifica attraverso *AuthType* il tipo di autenticazione, ovvero *Kerberos.* È necessario soffermarsi su altre due keyword di questa configurazione: *KrbAuthRealms* che specifica il o i realms verso cui si effettuerà l'autenticazione e *Krb5Keytab* che specifica il file keytab di Kerberos 5 contenente il principal associato al servizio web. In un web server il principal deve essere nella forma di *HTTP/hostname.dominio@REALM* e nel nostro caso sarà quindi *HTTP/venere.azienda.it@AZIENDA.IT*. Prima di fare rileggere la configurazione al server Apache, è necessario pertanto creare tale realm sul server *zeus* con i seguenti comandi:

```
# kadmin.local -q "addprinc -randkey
HTTP/venere.azienda.it@AZIENDA.IT"
# kadmin.local -q "ktadd -k /tmp/http-venere.key
```

```
HTTP/venere.azienda.it@AZIENDA.IT"
```

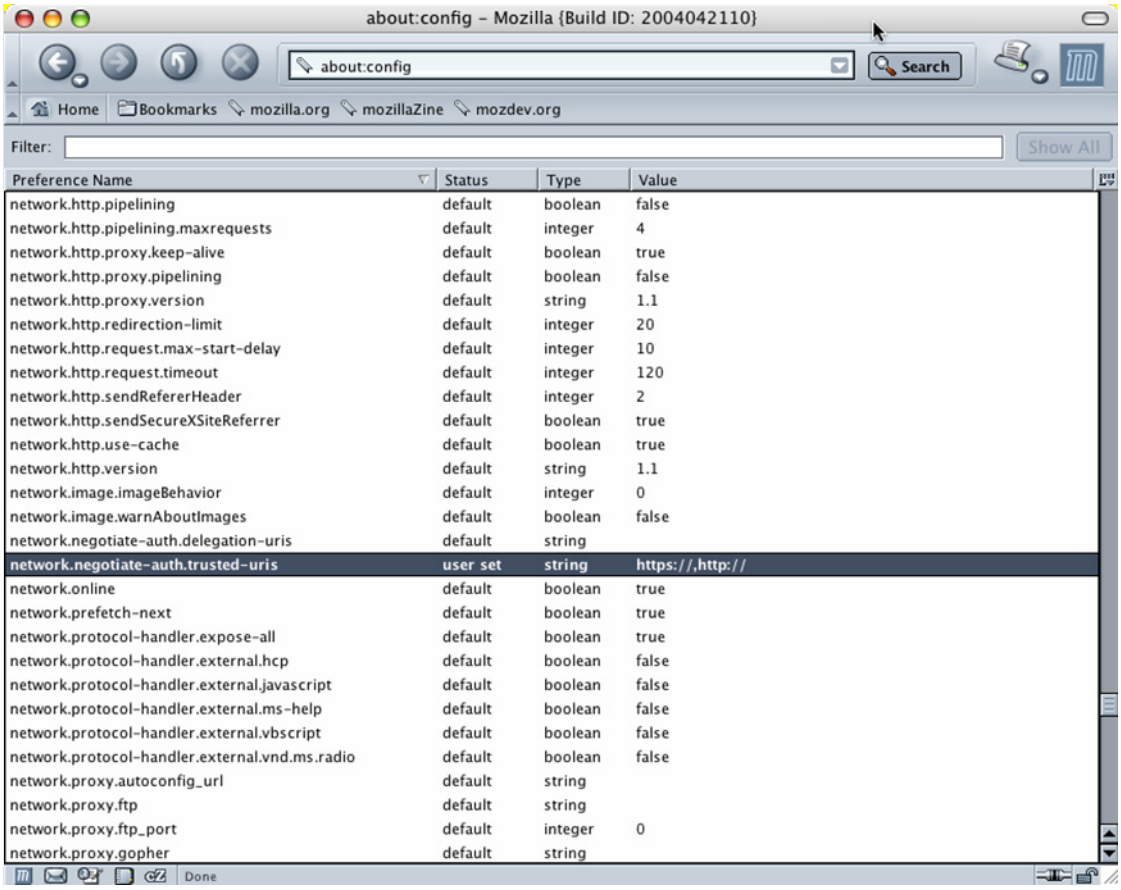
L'ultimo comando crea un file */tmp/http-venere.key* contenente la chiave Kerberos corrispondente al nostro servizio web: è necessario trasferire in modo sicuro (ad esempio tramite ssh) la chiave da *zeus* a *venere* e copiare la stessa sul file */etc/apache/krb5.keytab* specificato nel parametro *KrbAuthRealms*. A questo punto è possibile riavviare il server Apache con il comando *apachectl restart*.

A titolo di esempio, apriremo un browser qualsiasi (non ancora modificato cioè) e inseriremo l'URL http://venere.azienda.it/protected/. Il server ci risponderà con un codice di errore *401 Authorization Required*: abbiamo verificato che il server richiede un'autenticazione. Ma come fare adesso ad accedere lecitamente ?

Mozilla per Linux e MacOS X

Mozilla è uno dei web browser più diffusi su piattaforma Unix, ma anche su piattaforma Windows, che permette di estendere le proprie funzionalità attraverso dei *plugin*. Precedentemente alla versione 1.7 di Mozilla, era necessario scaricare un Plugin chiamato *Negotiateauth,* presente sull'indirizzo http://negotiateauth.mozdev.org/index.html, cui scopo è la creazione di un plugin che supporti il metodo di autenticazione *HTTP Negotiate* (ovvero SPNEGO), in particolare implementando il meccanismo Kerberos attraverso le GSSAPI. A partire da Mozilla 1.7 il supporto SPNEGO sarà incorporato per tutte le versioni Unix e MacOS X, mentre è ancora in fase di adattamento alle API SSPI per Windows.

Descriveremo la procedura per MacOS X, tenendo presente che la stessa è applicabile anche per Linux: la differenza è sostanzialmente grafica. Mozilla 1.7 è disponibile dal sito Internet http://www.mozilla.org; una volta scaricato ed installato, si proceda ad avviarlo normalmente. Quando l'applicazione aprirà una finestra del browser, inserire come URL *about:config* e premere invio: questa schermata permette la configurazione di tutti i parametri del browser. È necessario modificare il parametro *network.negotiate-auth.trusted-uris* in modo da fornire informazioni relative al nostro kerberos principal: di default le informazioni vengono fornite a tutti i siti HTTPS (con SSL). Per il nostro scopo abbiamo modificato il parametro inserendo i valor *https://,http://* come nella figura 14.



Preference Name	Status	Type	Value
network.http.pipelining	default	boolean	false
network.http.pipelining.maxrequests	default	integer	4
network.http.proxy.keep-alive	default	boolean	true
network.http.proxy.pipelining	default	boolean	false
network.http.proxy.version	default	string	1.1
network.http.redirection-limit	default	integer	20
network.http.request.max-start-delay	default	integer	10
network.http.request.timeout	default	integer	120
network.http.sendRefererHeader	default	integer	2
network.http.sendSecureXSiteReferrer	default	boolean	true
network.http.use-cache	default	boolean	true
network.http.version	default	string	1.1
network.image.imageBehavior	default	integer	0
network.image.warnAboutImages	default	boolean	false
network.negotiate-auth.delegation-uris	default	string	
network.negotiate-auth.trusted-uris	user set	string	https://,http://
network.online	default	boolean	true
network.prefetch-next	default	boolean	true
network.protocol-handler.expose-all	default	boolean	true
network.protocol-handler.external.hcp	default	boolean	false
network.protocol-handler.external.javascript	default	boolean	false
network.protocol-handler.external.ms-help	default	boolean	false
network.protocol-handler.external.vbscript	default	boolean	false
network.protocol-handler.external.vnd.ms.radio	default	boolean	false
network.proxy.autoconfig_url	default	string	
network.proxy.ftp	default	string	
network.proxy.ftp_port	default	integer	0
network.proxy.gopher	default	string	

Figura 14 - Mozilla, configurazione negotiate auth

Una volta effettuata la modifica, collegarsi al web server all'URL http://venere.azienda.it/protected/: automaticamente verremo riconosciuti nel sistema, non ricevendo più l'errore *401 Authorization Required* come avvenuto precedentemente, ma invece avremmo la nostra pagina di test con il logo di PHP (Fig. 15).

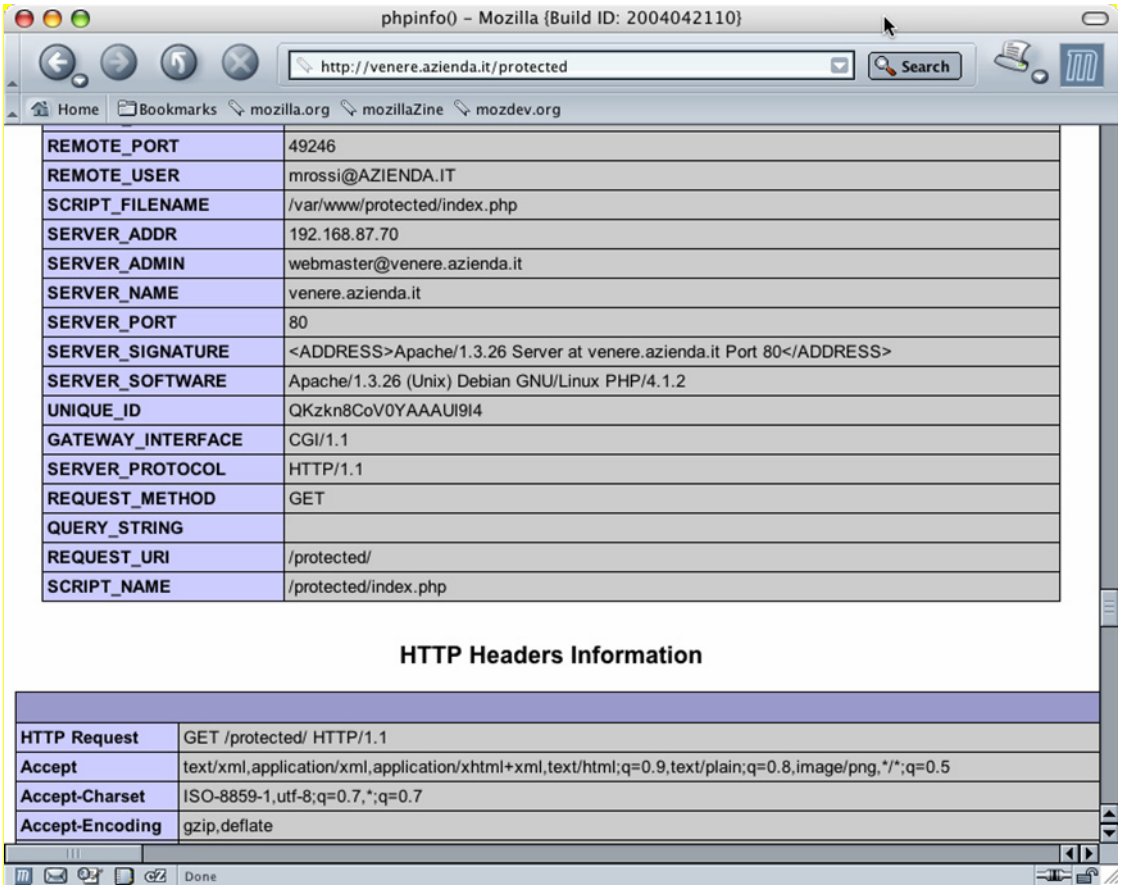


Figura 15 - Mozilla, accesso al sito protetto Kerberos

Da notare che nella sezione *Apache Environment* della pagina di test, la variabile *REMOTE_USER* è impostata a *mrossi@AZIENDA.IT*: il modulo di autenticazione ha anche riportato ad apache qual'è il nostro Kerberos Principal. In seguito vedremo in che modo possiamo sfruttare questa variabile.

Microsoft Internet Explorer

Sebbene Microsoft Internet Explorer ha di default il supporto Kerberos, questo va comunque configurato in modo da mandare informazioni relative all'autenticazione: di default, infatti, MSIE non tenta di negoziare l'autenticazione in quanto non riconosce i web sites come "local Intranet".

Aprendo Internet Explorer, fare click su Tools e poi Internet Options. Fare click sulla sezione *Security*, selezionare *Local Intranet* e successivamente *Sites* (Fig. 16).

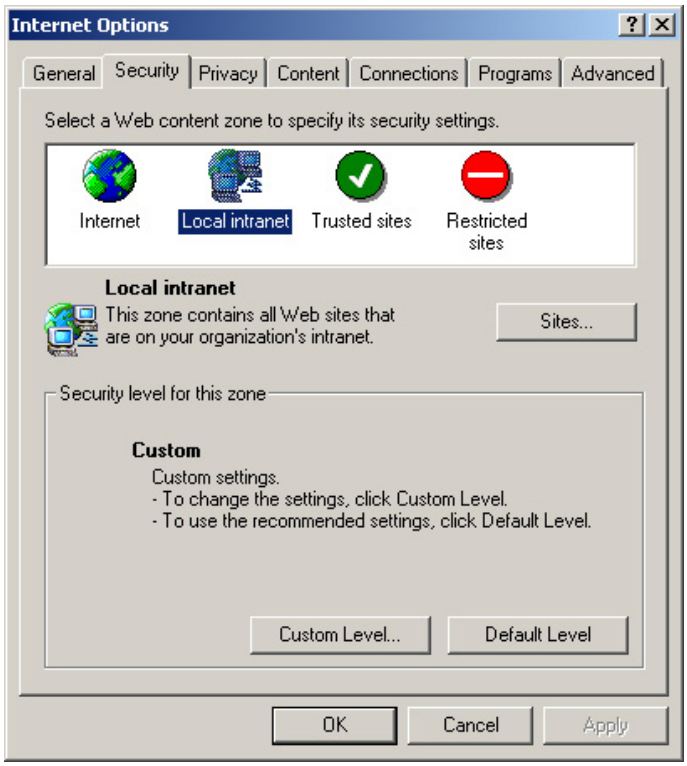


Figura 16 - Internet Explorer, propietà di sicurezza

Assicurarsi che *Include all sites that bypass proxy server* sia selezionato e premere *Advanced* (Fig. 17).

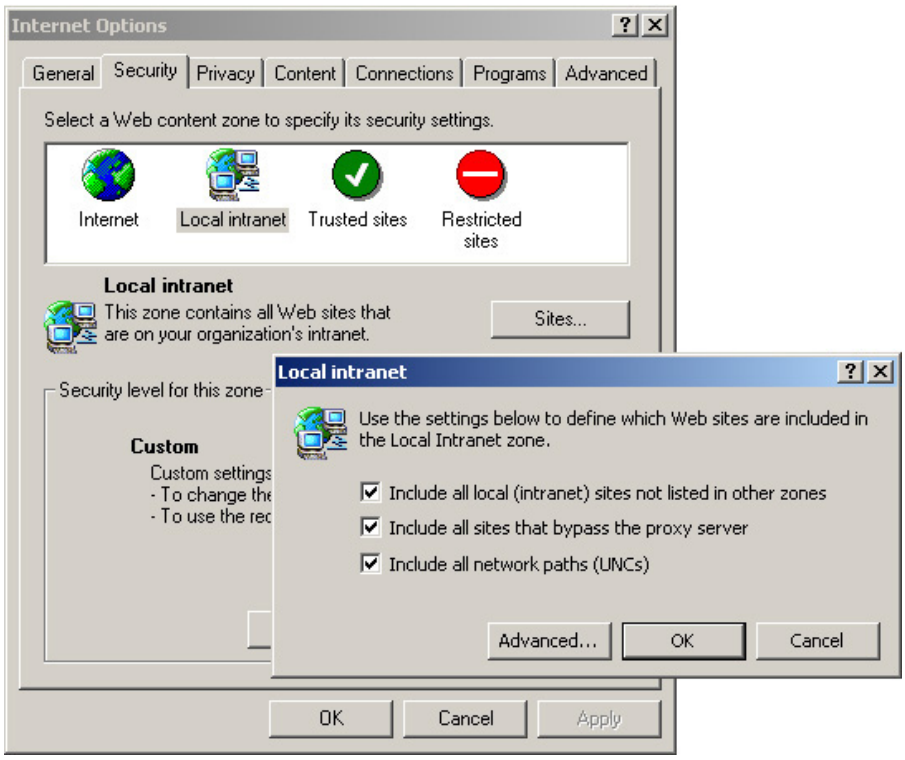


Figura 17 - Internet Explorer, definizione siti nella Local Intranet

Nella finestra di dialogo Local Intranet (Advanced), inserire tutti i siti verso le quali intendiamo effettuare l'autenticazione Kerberos, nel nostro caso **.azienda.it*. Selezionare poi Ok (Fig. 18).

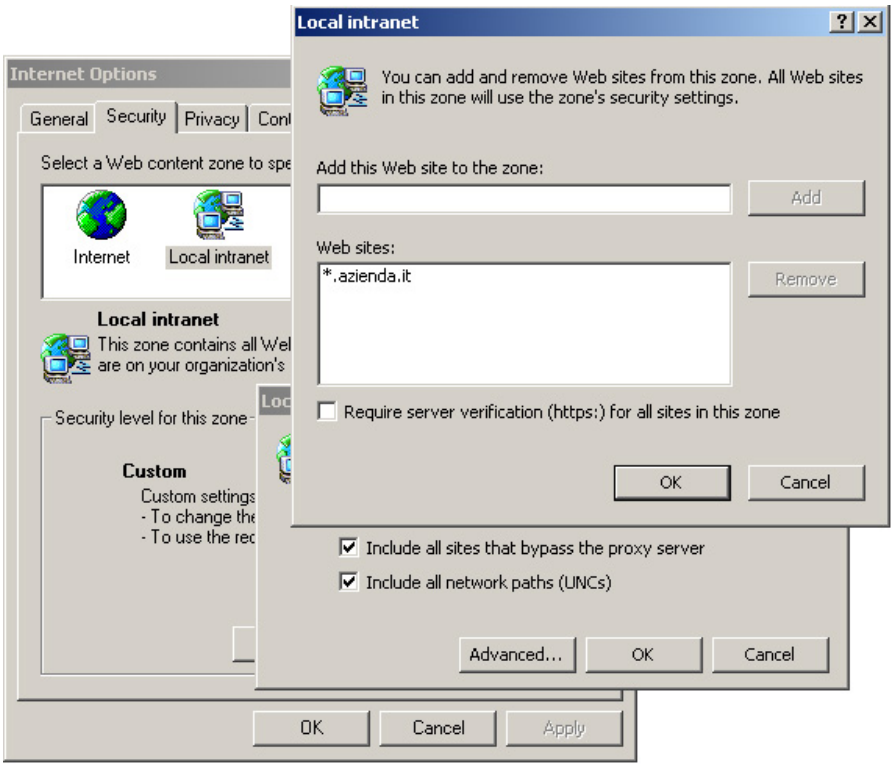


Figura 18 - Internet Explorer, aggiunta di web sites nella Intranet Zone

Successivamente, nella sezione *Security*, selezionare *Local Intranet* e poi il bottone Custom Level. Nella finestra di dialogo *Security Settings*, cercare la sezione *User Authentication* e selezionare *Automatic logon only in Intranet zone* e premere Ok (Fig. 19).

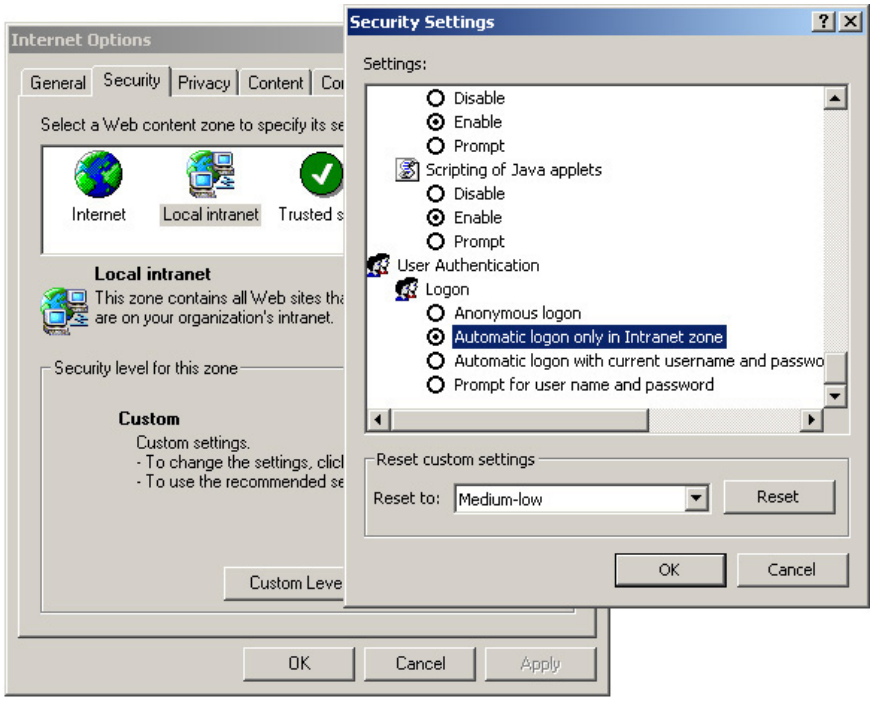


Figura 19 - Internet Explorer, definizione autologon nei siti Intranet

Per Internet Explorer 6.0, selezionare Advanced e nella sezione security selezionare la voce *Enable integrated Windows Authentication (requires restat)* (Fig. 20). Premere Ok ed eseguire il reboot del computer.

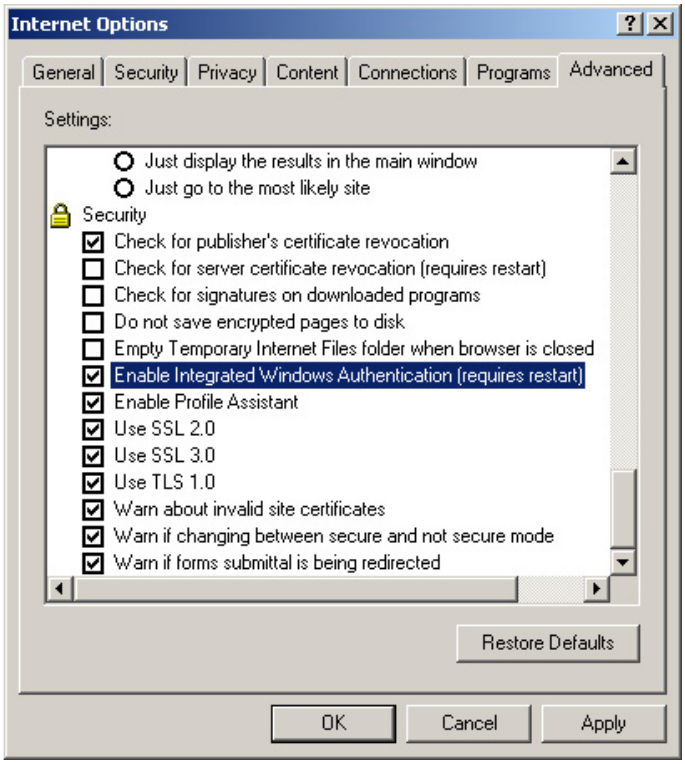


Figura 20 - Internet Explorer, definizione autenticazione Windows integrata

Proviamo anche con Internet Explorer a collegarci al web server http://venere.azienda.it/protected, a verificare di essere riconosciuti sul sistema e che la variabile *REMOTE_USER* sia impostata correttamente.

Un esempio di applicazione

Questo esempio vuole dimostrare la fattibilità di poter identificare e successivamente profilare l'utente attraverso Kerberos e LDAP. Cosi' come per una sessione interattiva al sistema, l'autenticazione Kerberos infatti è solo il primo step: è necessario dopo ricavare le informazioni relative all'utente, ad esempio a quale gruppo appartiene ed assegnargli i privilegi corrispondenti. L'esempio che segue, scritto in PHP, è in grado di ricavare il Kerberos Principal dalla variabile di ambiente *REMOTE_USER*, quindi effettua due queries al LDAP server per ricavarne nome (*givenName*) e User ID (*uid*), per poi successivamente ricavarne i gruppi di appartenenza.

```
<?php
// Sample authentication module for GSSAPI
// By Giuseppe Paternò <gpaterno@gpaterno.com>

$krbuser=getenv("REMOTE_USER");
echo "<h3>Autenticazione utente kerberos<br>$krbuser</h3><br>";
$ds=ldap_connect("ldaps://ldap.azienda.it/");

if ($ds) {
   // Bind and search for Given Name and uid
   $r=ldap_bind($ds);
   $sr_user=ldap_search($ds, "ou=People, dc=azienda,dc=it",
"krb5PrincipalName=$krbuser");

   if (ldap_count_entries($ds, $sr_user)>1) {
       echo "Error, too many entries with the same user
$krbuser<br>";
   }

   $info = ldap_get_entries($ds, $sr_user);
   $givenname = $info[0]["givenname"][0];
   $uid = $info[0]["uid"][0];

   // Print out welcome message
   echo "Ciao $givenname!<br><br>";

   // Search for groups in which we are members
   $sr_group=ldap_search($ds, "ou=Group,dc=azienda,dc=it",
"memberUid=$uid");

   if (ldap_count_entries($ds, $sr_group) == 0) {
       echo "L'utente $uid non fa parte di nessun gruppo.<br>";
   } else{
      $group_info = ldap_get_entries($ds, $sr_group);
      echo "Fai parte dei seguenti gruppi:<br>";
      for ($i=0; $i<$group_info["count"]; $i++) {
         echo "Gruppo <b>" . $group_info[$i]["cn"][0] . "</b> (gid="
.  $group_info[$i]["gidnumber"][0] . ")<br>";
      }
      echo "<br><br>";
      echo "Qui verrà creata la sessione PHP e ti verranno concesse
le autorizzazioni appropriate.<br>";
      // Create session here for further access: profile user and
      // give permissions to data.
   }

   ldap_close($ds);

} else {
   echo "<h4>Unable to connect to LDAP server</h4>";
}
?>
```

In un’applicazione reale, bisognerebbe aggiungere più controlli di errore nello script e creare successivamente una sessione in cui l’utente viene profilato in base ad i gruppi di appartenenza.

Applicazioni Web, Kerberos e l'accesso sicuro ai database

Potremmo spingerci oltre con l’accesso web: alcuni database, come ad esempio PostgresSQL (http://www.postgresql.org/) supportano l'autenticazione Kerberos. Attivando l'opzione *KrbSaveCredentials* del modulo *mod_auth_kerb* è possibile salvare il file del ticket dopo che l'utente viene autenticato, in modo da poterlo sfruttare attraverso gli script CGI. Combinando entrambe le funzionalità è possibile collegarsi al database con l'utente che effettua la richiesta web, applicandone così le policy di accesso ai dati implementati direttamente nel database server. Il vantaggio è che non è più necessario inserire nella configurazione del CGI l'utente e la password con cui accedere al database, con una conseguente maggiore sicurezza.

9. POSTA ELETTRONICA

L'ultimo passo della nostra dimostrazione è relativo alla posta elettronica: anche in questo caso dobbiamo essere in grado di collegarci con il nostro server di posta elettronica senza dover immettere le credenziali utente. Nel modello Microsoft il mail server Exchange usa un protocollo proprietario incapsulato su NetBIOS ed utilizza un sistema di autenticazione differente, che si basa sempre sul framework di SPENGO. Il nostro scopo è quello di usare tecnologia Open, basata su standard quali SMTP e IMAP, ma in particolare si è deciso di focalizzarsi sull'autenticazione relativa ad IMAP. È possibile abilitare l'autenticazione GSSAPI anche per alcuni server SMTP, però a causa della scarsa disponibilità di client che supportano questo tipo di autenticazione, si è deciso di implementare la policy relativa all'invio di posta tramite il controllo degli IP address sorgenti. La mancanza di client, soprattutto in ambiente Windows, ha creato numerosi problemi durante la creazione del laboratorio: esistono numerosi client Unix e Mac OSX, ma pochi sono quelli per Windows, probabilmente perchè gli sviluppatori Windows considerano Kerberos un qualcosa di "nuovo" introdotto da Microsoft, per cui ancora poco conosciuto, o peggio un qualcosa di "troppo vecchio", ignorando che l'intera architettura di Active Directory è basata proprio sul "vecchio" Kerberos.

L'installazione del server

Per implementare il server di posta elettronica si sono scelti due popolari software OpenSource, ovvero Postfix (http://www.postfix.org) e University of Washington UW-IMAPD (http://www.washington.edu/imap/). Per quanto riguarda la distribuzione Debian, il primo software è disponibile attraverso il pacchetto *postfix-tls* e non si è proceduto a nessuna personalizzazione particolare, a parte la modifica della variabile *mydestination* nel file main.cf e specificando *venere.azienda.it:* questa variabile permette a Postfix di capire qual'è il dominio o il nome host appartenente alla macchina locale. È consigliabile procedere alla configurazione della variabile *mynetworks* specificando la propria rete locale (es: *mynetworks = 192.168.1.0/24*): questo permette a tutti i client della rete locale

di usare Postfix come relay di posta elettronica. Postfix è già stato compilato per diverse piattaforme, ma nel caso è possibile ricompilarlo scaricandone i sorgenti dal sito specificato.

Il server IMAP UW-IMAPD è disponibile anch'esso come pacchetto Debian, ma non è stato compilato con l'autenticazione GSSAPI: è pertanto necessario ricompilare il server a partire dai sorgenti, questi ultimi disponibili sul sito dell'Università di Washington. I sorgenti del server sono contenuti nel file *imap.tar.Z*: una volta decompresso il file è sufficiente procedere alla compilazione come segue:

```
make VERSION=uw-imap-ssl-gssapi lnp SSLTYPE=unix
EXTRAAUTHENTICATORS=gss
```

Da notare l'opzione *EXTRAAUTHENTICATORS* che permette l'autenticazione attraverso le GSSAPI. Si ricorda che gli include files relativi a Kerberos e ad SSL devono essere già negli include path, altrimenti è necessario specificare nella variabile *EXTRACFLAGS* la lista degli include path (es: *EXTRACFLAGS="-I/usr/kerberos/include"*).

Una volta eseguita la compilazione, identificare il file *imapd* che contiene l'IMAP daemon, che si trova nella sottodirectory *imapd/* e copiarlo come */usr/local/sbin/imapd.orig*. Con un editor di testo creiamo un nuovo file */usr/local/sbin/imapd*, che fungerà da wrapper, e inseriamo il seguente contenuto:

```
#!/bin/sh
KRB5_KTNAME=/etc/krb5.keytab.imap
export KRB5_KTNAME

/usr/sbin/imapd.orig $*
```

Questo wrapper ci consentirà di creare un file di keytab separato per il servizio IMAP (/etc/krb5.keytab.imap). Analogamente a quanto fatto per Apache nel paragrafo precedente, è necessario creare sul KDC il principal per il servizio, ad esempio:

```
# kadmin.local -q "addprinc -randkey
imap/venere.azienda.it@AZIENDA.IT"
# kadmin.local -q "ktadd -k /tmp/imap-venere.key
imap/venere.azienda.it@AZIENDA.IT"
```

Copiare quindi il file temporaneo che è stato creato (nel nostro esempio */tmp/imap-venere.key*) sul server IMAP con il nome del file specificato nel wrapper */etc/krb5.keytab.imap*, ricordando di proteggerlo alla sola lettura di root. Successivamente modificare il file */etc/inetd.conf* e includere quindi il nostro daemon IMAP come segue:

```
imap2  stream  tcp   nowait      root    /usr/sbin/tcpd
/usr/local/sbin/imapd
imaps  stream  tcp   nowait      root    /usr/sbin/tcpd
/usr/local/sbin/imapd
```

L'ultimo step della configurazione riguarda la creazione di un file di accesso PAM relativo al servizio IMAP. Anche se non è strettamente necessario in quanto l'autenticazione avverrà tramite GSSAPI, questo ci consentirà di erogare il servizio IMAP anche a quei client che non supportano GSSAPI. Creare quindi un file */etc/pam.d/imap* con lo stesso contenuto di quello creato per il servizio SSH o login, ad esempio:

```
#%PAM-1.0
auth       required     pam_nologin.so
auth       required     pam_env.so # [1]
auth       sufficient   pam_unix.so likeauth nullok
auth       sufficient   pam_krb5.so use_first_pass
auth       required     pam_deny.so

account    sufficient   pam_unix.so
account    sufficient   pam_krb5.so
#account    sufficient   pam_ldap.so
account    required     pam_deny.so

session    required     pam_limits.so
session    required     pam_unix.so
session    optional     pam_krb5.so
session    optional     pam_lastlog.so # [1]
session    optional     pam_motd.so # [1]
session    optional     pam_mail.so standard noenv # [1]

password   sufficient   pam_unix.so nullok use_authok
password   sufficient   pam_krb5.so use_authok
password   required     pam_deny.so
```

È fortemente suggerito attivare anche la funzionalità SSL: in questo caso è necessario provvedere alla generazione di un certificato ed ad installarlo come file su */usr/lib/ssl/certs/usr/local/sbin/imapd.orig.pem*. Il server riconoscerà automaticamente la connessione SSL in ingresso ed avvierà la negoziazione SSL.

Client Linux

Come client per Linux si è scelto Ximian Evolution, un client grafico per Linux che sta avendo molto successo in quanto simile a Microsoft Outlook. Quasi tutte le nuove distribuzioni di Linux includono Evolution tra i loro client di posta elettronica, ma è comunque scaricabile gratuitamente dal sito Internet http://www.ximian.com/products/evolution/. Si assume che il client sia stato installato e che si abbia effettuato il primo set-up.

Dal menù *Tools*, selezionare *Settings* e una finestra di dialogo verrà presentata all'utente.

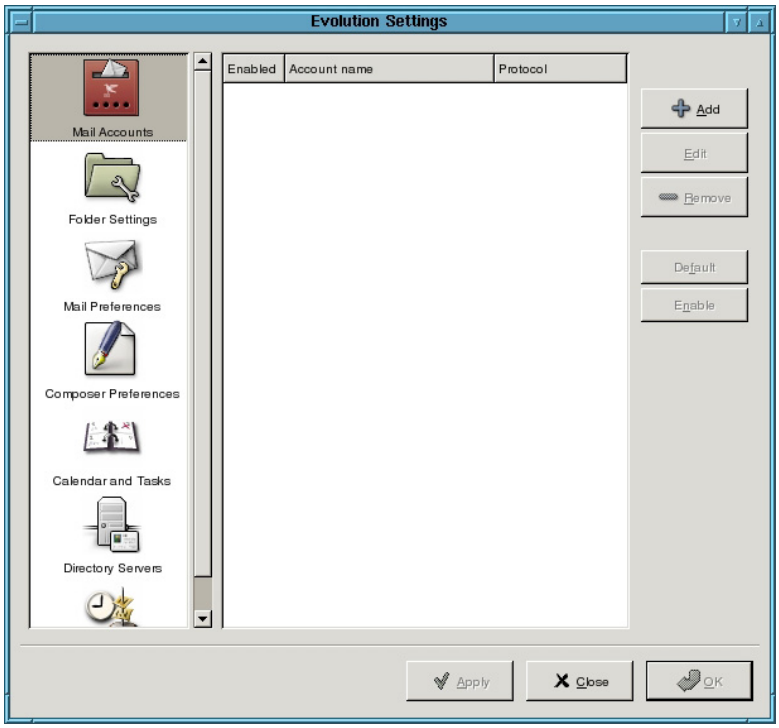


Figura 21 - Evolution, Mail Accounts

Selezionare quindi *Mail Accounts* (Fig. 21) ed il pulsante Add. Un wizard guiderà l'utente nella creazione del profilo: alla prima schermata *Mail Configuration* premere *Forward* (Fig. 22).

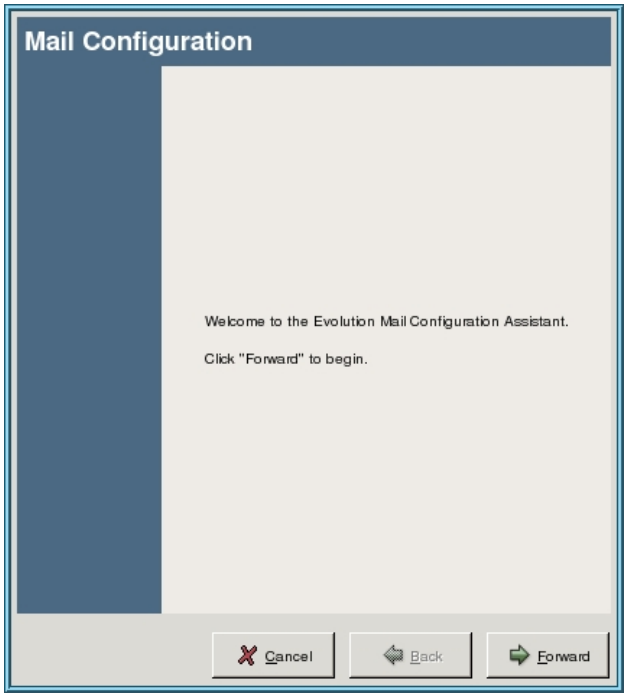


Figura 22 - Evolution, inizio wizard configurazione mail

Nella schermata *Identity* inserire i propri dati, ad esempio come mail *mrossi@venere.azienda.it*, e premere *Forward* (Fig. 23).

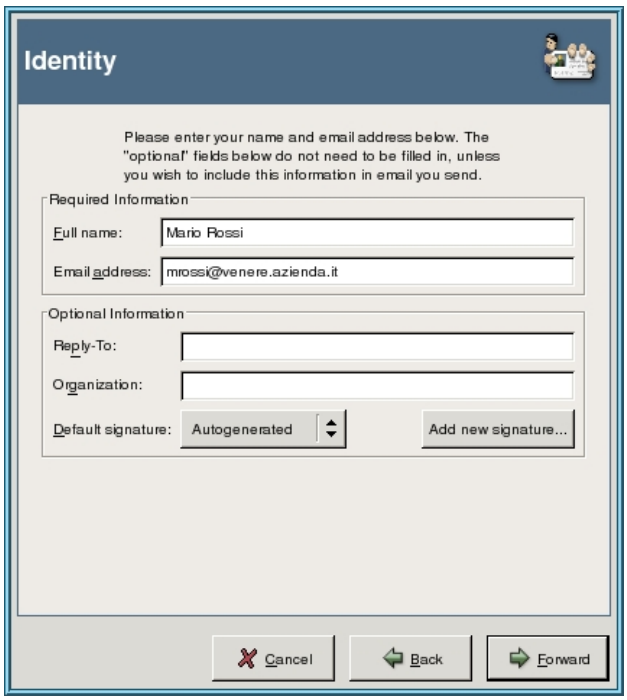


Figura 23 - Evolution, definizione nome ed email nel wizard

Nella finestra *Receiving Email,* selezionare come *Server Type* il valore IMAP. Nella sezione *Configuration* come *Host* specificare il mail server, ad esempio *venere.azienda.it*, lo *username*, es: *mrossi*, e in *Use secure connection (SSL)* selezionare *Always*. Nella sezione *authentication* specificare nell'*Authentication type* il valore *GSSAPI* (Fig. 24) e premere *Forward*.

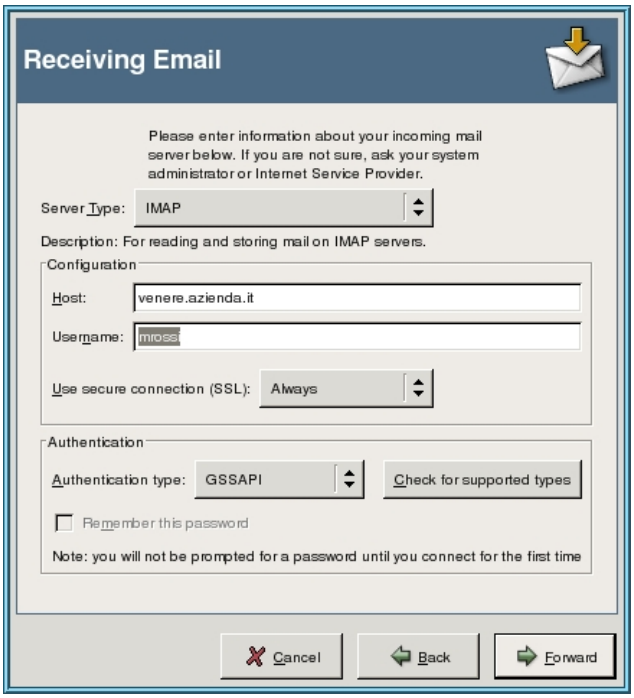


Figura 24 - Evolution, definizione mail server ed uso di GSSAPI

Nella finestra successiva *Receiving Email* lasciare il default e premere *Next.* Successivamente verrà chiesto nella finestra *Sending Email* come inviare la mail. Lasciare come *Server Type* il valore *SMTP*, e come *Host* inserire il nome del nostro server di posta, ad esempio *venere.azienda.it* (Fig. 25). Premere *Forward*.

Figura 25 - Evolution, definizione SMTP server

Nella schermata di *Account Management*, lasciare il default e premere *Next*. Se la configurazione e l'autenticazione andrà a buon fine, allora verrà creato una cartella sulla sinistra e potremmo accedere alla cartella IMAP presente sul server.

Client Windows

Come accennato in precedenza, è stato difficile trovare un client IMAP per Windows che supporti Kerberos. Ne sono stati trovati alcuni, come ad esempio Eudora, ma su Windows cercano l'implementazione MIT di Kerberos e non sfruttano le SSPI. L'unico software in grado di usare le API SSPI è PC-Pine disponibile gratuitamente sul sito Internet http://www.washington.edu/pine/getpine/pcpine.html. È necessario scaricare il "PC-Pine for Windows 2000 Kerberos setup program", che è stato compilato per sfruttare le SSPI di Windows 2000. Il file si presenta come un normale installer per Windows, basterà quindi scaricarlo ed eseguirlo. Una volta eseguito, copiare il file *pinerc.adv*, solitamente presente nella directory *C:\Programmi\PC-Pine* in una cartella personale (ad esempio chiamandolo *pinerc*) e modificare il parametro *inbox-path* come segue:

```
inbox-path={venere.azienda.it/novalidate-cert}INBOX
```

Il parametro prevede tra le parentesi graffe l'hostname del nostro mail server IMAP, seguito da parametri relativi alla connessione, e successivamente la cartella INBOX. Come si può notare, non si specifica nessuna autenticazione particolare: in automatico Pine selezionerà il protocollo migliore per l'autenticazione, nel nostro caso GSSAPI, e il protocollo TLS/SSL. Se non si vuole usare SSL per collegarsi al server IMAP sarà necessario specificare il parametro /notls dopo l'hostname (es: *{venere.azienda.it/notls}INBOX*), mentre il parametro */novalidate-cert* serve per evitare il controllo del certificato nell'elenco delle Certification Authorities di fiducia. Solitamente quest'ultimo parametro viene sempre messo quando il certificato del server non è stato acquistato da una CA ufficiale, ma è stato generato in una CA interna.

Dopo aver modificato il file, nel nostro caso *pinerc*, eseguire il programma *PC-Pine.* Dopo lo splash screen, verrà visualizzata una schermata di configurazione come in figura 26.

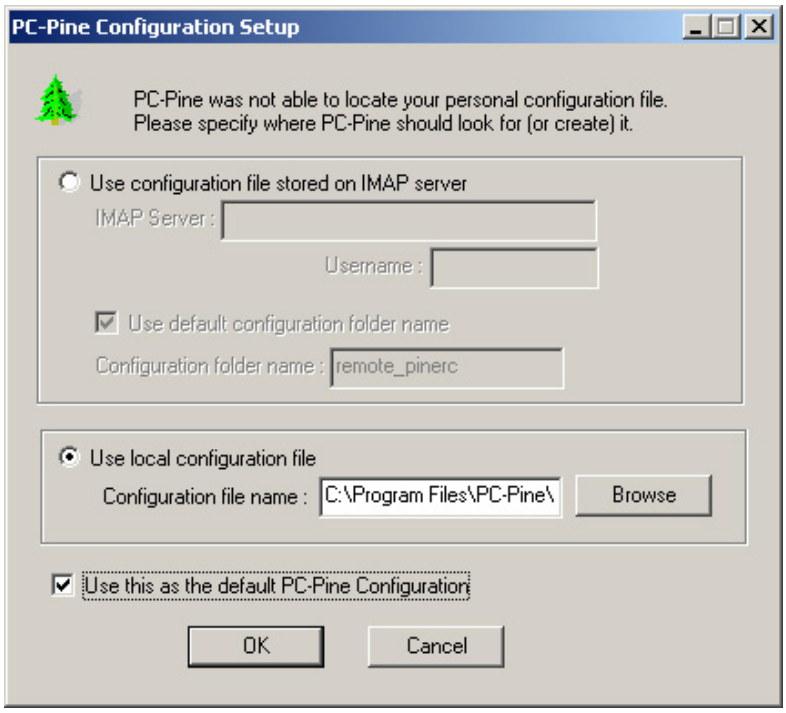


Figura 26 - Prima configurazione di PC-Pine

Selezionare *Use local configuration file* ed attraverso il tasto *Browse* selezionare il file *pinerc* modificato. Selezionare quindi il parametro *Use this as the default PC-Pine Configuration* e premere *Ok*. Successivamente si aprirà il programma esattamente alla Inbox desiderata. Se non visualizzate errori, il vostro mail client si è autenticato correttamente al server usando GSSAPI/SSPI. Qualora si desiderasse usarlo come client di posta, sarà necessario personalizzarlo ulteriormente con i propri dati e le proprie preferenze. Per maggiori informazioni si consiglia la lettura del manuale allegato.

Seppur si tratti di un ottimo client, PC-Pine però non dispone di un interfaccia grafica: il fatto di non essere "intuitivo" è un lato negativo che influenza negativamente la sua adozione. Alla data della scrittura del presente documento

nessun client Windows grafico è in grado di sfruttare le SSPI: Mozilla Thunderbird e Mozilla 1.7 (http://www.mozilla.org/) hanno entrambi la feature tra le "wish list", ma nessuno ha ancora deciso di implementarlo, come successo per SPNEGO e il client Web.

Client MacOS X

Al contrario del sistema operativo Windows, dove alla data della presente non esistono veri e propri client grafici (lo sarà Mozilla in una prossima versione), MacOS X dispone di un client Mail nativo che supporta l'autenticazione GSSAPI. L'applicazione, chiamata per l'appunto *Mail*, si trova inserita di default nel *Dock* di MacOS X. Una volta eseguita l'applicazione, selezionare il menù *Mail* e selezionare la voce *Preferences*. Selezionare la sezione *Accounts* (Fig. 27) e premere sul simbolo + in basso a sinistra.

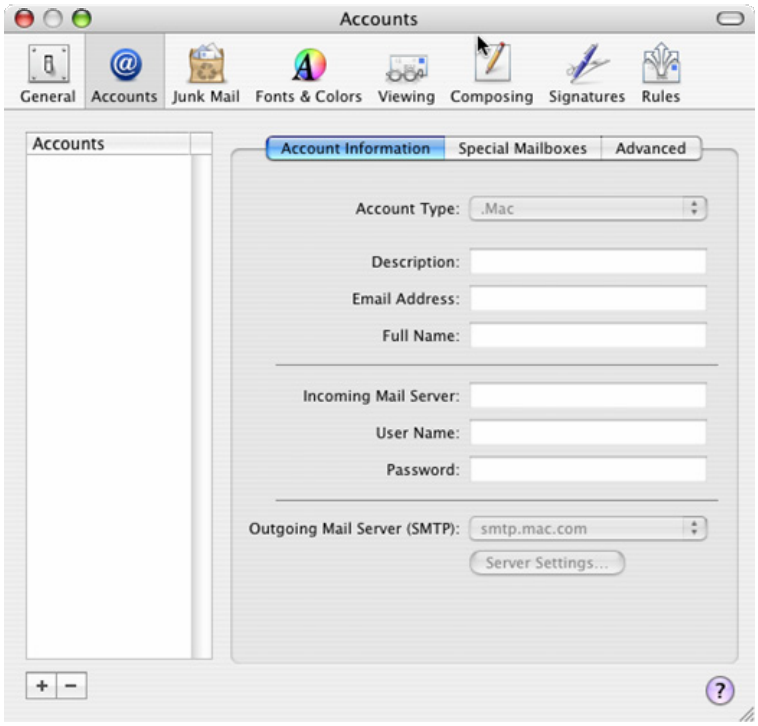


Figura 27 - Apple Mail, vista degli account

La prima schermata è relativa alla sezsione *Account Information*. Selezionare come *Account Type* il valore *IMAP*, come Description un valore mnemonico come ad esempio *Venere*, inserire in *Email Address* il valore *mrossi@venere.azienda.it* e come *Full Name* il nome completo del nostro utente, ovvero *Mario Rossi*. Successivamemnte come *Incoming Mail Server*, indicare *venere.azienda.it* e come *User Name* il valore *mrossi* (Fig. 28).

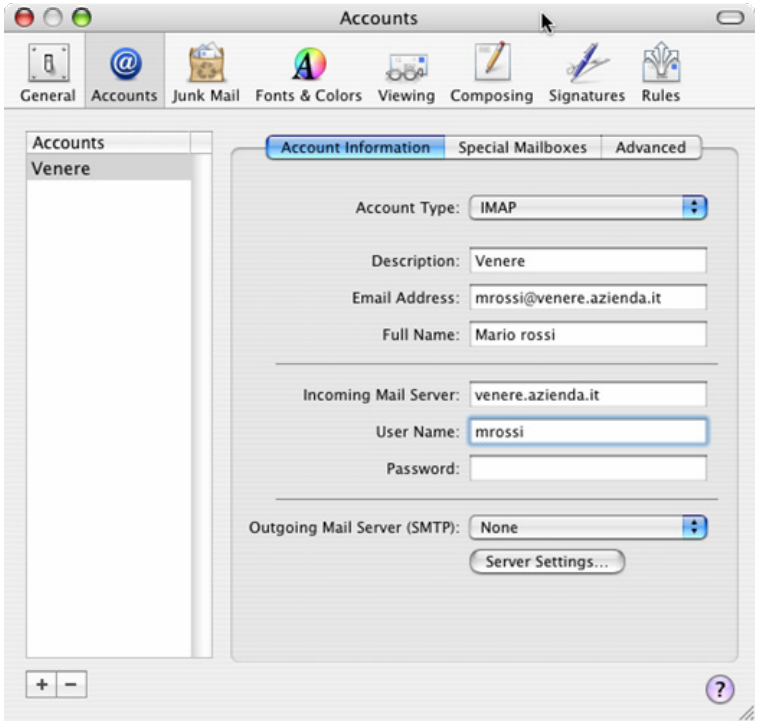


Figura 28 - Apple Mail, definizione account IMAP

Appena sotto *Outgoing Mail Server (SMTP)* selezionare Server Settings. Qui verrà indicato il server SMTP di uscita. Nella finestra indicare *venere.azienda.it* come *Outgoing Mail Server* e premere quindi *Ok* (Fig. 29).

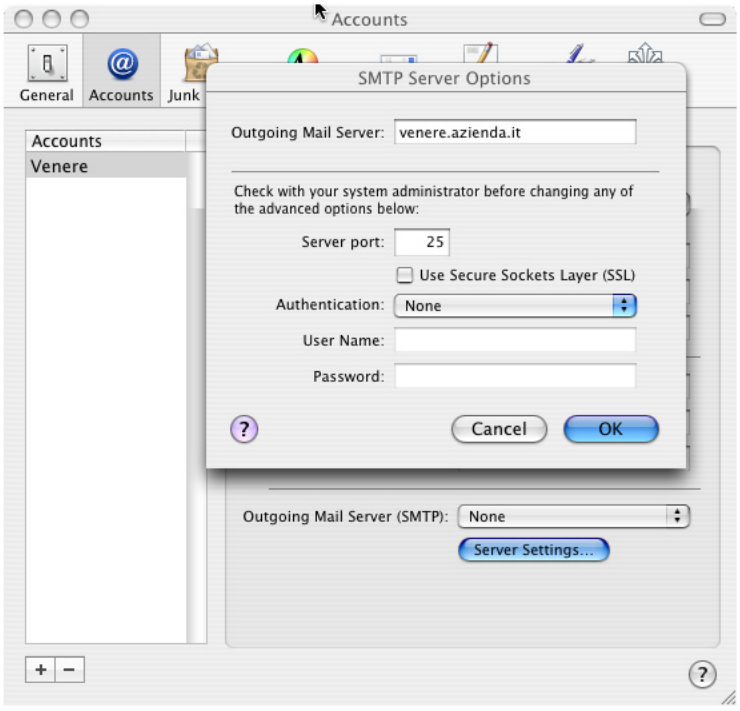


Figura 29 - Apple Mail, definizione SMTP server

Ritornati al menù Account Information, selezionare quindi la sezione Advanced. In fondo alla finestra, nella voce Authentication specificare *Kerberos Version 5 (GSSAPI)* come in figura 30.

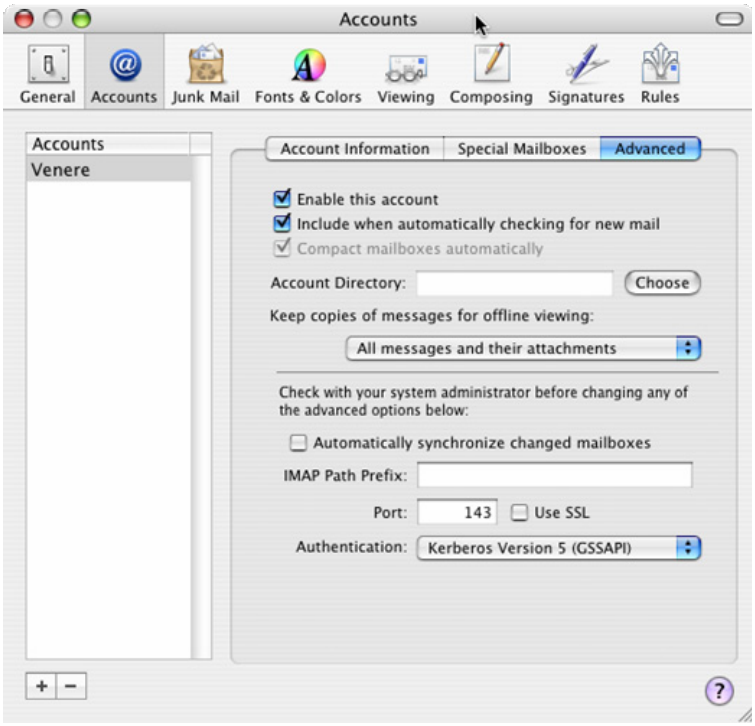


Figura 30 - Apple Mail, definizione autenticazione con GSSAPI

Chiudere la finestra e salvare. Se la nostra configurazione avrà funzionato, nella finestra *In* troverete i vostri messaggi nell'INBOX.

10. INTEGRAZIONE

Dopo aver dimostrato la fattibilità tecnica di un'architettura di Single Sign-On, è necessario calarla in una realtà aziendale ben definita, ovvero come faccio ad inserire una simile soluzione in azienda ?

Ho identificato tre diversi scenari:

- Assenza di un dominio Windows 2000 con il server Kerberos installato su un sistema Unix.
- Active Directory installato in azienda e uso del server Kerberos integrato in Windows 2000
- Active Directory installato in azienda e uso di due server kerberos: sia quello integrato in Windows 2000 che un server Kerberos su Unix, e una relazione di fiducia (trust relationship) tra i due realm Kerberos.

Vediamo in dettaglio quali sono i vantaggi e gli svantaggi di ognuna di queste soluzioni.

Uso del KDC su sistemi Unix in assenza di Active Directory

In questo scenario, l'azienda non dispone di un server Windows 2000 (o superiori) oppure non ha scelto di non adottare Microsoft Active Directory. Tipicamente è un ambiente con prevalenza di sistemi server Unix, dove ogni client è una workstation a se, in particolare i client Windows 2000 non eseguono logon su nessun dominio.

In questo caso sarebbe auspicabile installare il KDC e un LDAP server su un server Unix, creando in pratica quanto descritto in questo documento, facendo fare logon alle workstation Windows sul realm realizzato con il MIT Kerberos.

Uso del KDC integrato in Windows 2000

In questo ambiente l'azienda ha scelto di adottare Microsoft Active Directory, autenticando i client tramite questa metodologia. La prevalenza dei sistemi server si basa su Windows, con una bassa percentuale di sistemi Unix. In questo caso è possibile sfruttare il KDC integrato con Windows per "accogliere" i sistemi Unix in quanto AD, attraverso i tools forniti con il Resource Kit, è in grado di generare i Kerberos keytab files.

La prima cosa da fare è usare *ktpass* per generare l'account della macchina e creare il file */etc/krb5.keytab*, ad esempio:

```
C:> Ktpass –princ host/hostname@NT-DNS-REALM-NAME –mapuser account –
pass password –out UNIXmachine.keytab
```

dove:

- *hostname* è il nome del file DNS del server unix
- *NT-DNS-REALM-NAME* è il nome del realm (o del domino) di Active Directory
- *account* è il nome utente di AD associato alla macchina
- *password* è la password associata al sistema

Il file *UNIXmachine.keytab* contiene il keytab da inserire in */etc/krb5.keytab*: si raccomanda di trasferirlo in una maniera sicura, ad esempio attraverso SSH.

Editare quindi sul sistema Unix il file */etc/krb5.conf*

```
[libdefaults]
default_realm = AZIENDA.IT
default_tkt_enctypes = des-cbc-md5 des-cbc-crc
default_tgs_enctypes = des-cbc-md5 des-cbc-crc

[realms]
AZIENDA.IT = {
          kdc = domaincontroller.azienda.it:88
}
```

Come si può notare, l'installazione è simile a quella del KDC in ambiente Unix, pertanto sulla workstation Unix si procederà con la normale personalizzazione dell'ambiente, ad esempio modificando il PAM oppure installando il modulo di autenticazione Web qualora si volesse erogare simili servizi.

Per completare l'installazione è necessario modificare Active Directory in modo tale da poter rispondere alle esigenze di LDAP, in particolare per gli attributi utente (home directory, uid/gid, ecc...) esattamente come nel nostro esempio si è usato OpenLDAP. Durante la sperimentazione ho avuto modo di provare l'ottimo plugin AD4UNIX di Maxim Batourine che permette di modificare le impostazioni Unix direttamente dalla console di Active Directory. Il file chiamato *MKSADPlugins.msi* può essere scaricato da http://www.padl.com/download/.

Si riporta a titolo di esempio un file ldap.conf che può essere usato in un ambiente Active Directory.

```
# @(#)$Id: ldap.conf,v 1.8 2002/02/26 08:50:37 root Exp $
base dc=azienda,dc=it
ldap_version 3
uri ldaps://dc.windows.azienda.it
binddn anonymous@azienda.it
timelimit 10
bind_timelimit 2

scope sub

pam_filter objectclass=user
pam_login_attribute sAMAccountName
pam_password ad

nss_base_passwd      ou=users,dc=windows,dc=azienda,dc=it?one
nss_base_shadow      ou=users,dc=windows,dc=azienda,dc=it?one
nss_base_group       ou=group,dc=windows,dc=azienda,dc=it?one

nss_map_objectclass posixAccount User
nss_map_attribute uid sAMAccountName
nss_map_attribute uniqueMember Member
nss_map_attribute userPassword msSFUPassword
nss_map_attribute homeDirectory msSFUHomeDirectory
nss_map_objectclass posixGroup Group
nss_map_attribute cn sAMAccountName
```

Trust relationship tra un KDC Windows e uno Unix

Esiste un terzo scenario, in cui l'azienda ha sia deciso di adottare Microsoft Active Directory, ma ha una base significativa di server Unix installati. In questo caso la gestione di un numero così significativo di macchine Unix potrebbe risultare difficile, soprattutto se il dipartimento IT che gestisce i sistemi Microsoft è differente dal dipartimento che segue i sistemi Unix. In questo caso è possibile effettuare una relazione di fiducia tra due realm di Kerberos, ovvero tra il mondo AD e quello Unix. Cosi' facendo, i due mondi continueranno ad esistere in maniera indipendente, ma permetterà comunque agli amministratori Unix di poter accedere ai sistemi dalla loro workstation Windows oppure erogare servizi verso il mondo Microsoft.

Quanto descritto nei prossimi paragrafi vuole fornire un esempio dei comandi usati per effettuare il Trust Relationship tra AD e un KDC su Unix: per maggiori informazioni si faccia riferimento alla documentazione fornita da Microsoft e dal MIT Kerberos.

Set-up del KDC di Windows Active Directory

Tutte le operazioni relative a Windows devono essere fatte su di un Active Direcotry Server per il dominio. Di default la relazione di trust è non transitiva, che vuol dire che solo gli utenti direttamente provenienti dal dominio di fiducia si possono autenticare: sebbene in questo esempio non si voglia dimostrare un "trust chain" tra realm, si sappia che questo comportamento può essere cambiato usando il tool *netdom.exe*.

Come analogamente fatto per il client, anche sul KDC Microsoft è necessario aggiungere il riferimento al KDC appartenente al realm Unix, ad esempio con:

```
C:> ksetup /addkdc UNIX.AZIENDA.IT kdc.unix.azienda.it
```

Questa operazione va effettuata anche su ogni workstation Windows che vuole accedere al dominio UNIX.AZIENDA.IT. Come accennato precedentemente con netdom.exe è possibile cambiare il comportamento di default del "trust chain", impostandolo a transitivo, ad esempio con:

```
c:\> netdom query /d:WINDOWS trust*
c:\> netdom query /d:WINDOWS UNIX.AZIENDA.IT /trans:yes*
c:\> netdom query /d:WINDOWS UNIX.AZIENDA.IT /trans*
```

Successivamente è necessario aggiungere le chiavi per l'autenticazione cross-realm sul KDC Windows. Avviare il *Domain Tree Management* (posizionato in *Programs*, *Administrative Tools*, *Active Directory Domains and Trusts*), con il tasto destro del mouse selezionare *Properties* relativo al proprio dominio e selezionare la sezione *Trust*. Premere *Add* ed inserire il nome del dominio e password, quando verrà notificato che non si tratta di un dominio Kerberos di Windows selezionare Ok.

Successivamente, è necessario legare un utente al realm remoto. Sempre da *Domain Tree Management*, selezionare *Active Directory Users and Computer*s, poi *View* ed *Advanced Features*. Fare click con il tasto destro sul nome utente, selezionare *Name Mappings*, il tab *Kerberos Names* e il pulsante Add. Aggiungere a questo punto il principal remoto, ad esempio utente@UNIX.AZIENDA.IT

È necessario effettuare il reboot del KDC Microsoft per applicare le modifiche.

Set-up del KDC di Unix

A questo punto è necessario creare le chiavi anche sul KDC presente sul server Unix. Bisogna sempre tenere presente che il sistema di crittografia Kerberos di Windows si basa su des-cbc-crc e non su des3-hmac-sha1 (il default del MIT). Una volta configurato correttamente il KDC, come già configurato durante la dimostrazione descritta in questo documento, per inserire le chiavi è sufficiente effettuare i seguenti comandi sul KDC attraverso il tool kadmin:

```
addprinc krbtgt/WINDOWS.AZIENDA.IT@UNIX.AZIENDA.IT
addprinc krbtgt/UNIX.AZIENDA.IT@WINDOWS.AZIENDA.IT
```

È necessario usare la stessa password per entrambe le chiavi. Oltre alla creazione dei principals, è necessario configurare ogni client per avere i puntatori al realm remoto, ad esempio vediamo un file krb5.conf, si noti in particolare la sezione capaths:

```
[libdefaults]
          default_realm = UNIX.AZIENDA.IT
          default_tkt_enctypes = des-cbc-md5
          default_tgs_enctypes = des-cbc-md5

[realms]
          WINDOWS.AZIENDA.IT = {
                    kdc = kdc.windows.azienda.it:88
          }

          UNIX.AZIENDA.IT = {
                    kdc = kdc.unix.azienda.it
                    admin_server = kdc.unix.azienda.it
          }

[domain_realm]
          .unix.azienda.it = UNIX.AZIENDA.IT
          .windows.azienda.it = WINDOWS.AZIENDA.IT
```

```
[capaths]
          UNIX.AZIENDA.IT = {
                    WINDOWS.AZIENDA.IT = .
          }
          WINDOWS.AZIENDA.IT = {
                    UNIX.AZIENDA.IT = .
          }
```

Da notare nella sezione capath il punto dopo l'uguale. Per ogni utente unix, creare un .k5login file nella home directory utente con gli altri principals che sono abilitati all'accesso, ad esempio:

```
# cat /home/utente/.k5login
utente@WINDOWS.AZIENDA.IT
utente@UNIX.AZIENDA.IT
```

11. NOTE SU ALTRI APPLICATIVI

In questo breve capitolo ho voluto inserire alcuni riferimenti ad applicativi che supportano o supporteranno a breve l'autenticazione Kerberos, ma che non ho avuto occasione di sperimentare durante il laboratorio.

IPSec

L'IP Security Protocol, conosciuto come IPSec, è oggi la tecnologia più diffusa per lo scambio sicuro di dati tra aziende, o più comunemente fra due computer. IPSec è stato definito dall'Internet Engineering Task Force già dall'Agosto 1995 attraverso l'RFC 1825: sono passati otto anni da allora e IPSec è cresciuto notevolmente, affermandosi nel mercato come lo standard per la VPN, e fornendo al TCP/IP le funzionalità di autenticazione, integrità e crittografia di cui era sprovvisto. Uno dei fattori che ne hanno determinato la sua affermazione è che IPSec può essere inoltrato attraverso qualsiasi rete che supporta il protocollo IP, senza dover cambiare nessun nodo di rete, senza cambiare le applicazioni e senza cambiare in maniera sostanziale il sistema operativo dei nodi. Per maggiori informazioni su IPSec, invito il lettore a leggere il paragrafo relativo sul libro "Sicurezza Nelle Wireless LAN" (ISBN 88-901141-0-X).

Anche IPSec può avvantaggiarsi di Kerberos durante la fase di negoziazione della chiave attraverso ISAKMP: la chiave Kerberos viene utilizzata nello stesso modo di una Pre-Shared Key (o chiave condivisa) o di un certificato digitale. Attualmente l'implementazione più diffusa è quella di Microsoft, ma anche il progetto Kame (http://www.kame.net/) ha implementato nel suo daemon ISAKMP Racoon uno scambio attraverso GSSAPI. Racoon è attualmente usato dai sistemi BSD (NetBSD, FreeBSD, OpenBSD, ecc...) da MacOS X e da Linux a partire dal kernel 2.6.

NFS

Il Network File System (NFS) è un protocollo sviluppato da Sun Microsystems e definito in RFC 1094, che consente a un computer di montare un disco remoto. NFS è nato prima della condivisione di Windows ed è usato tipicamente in ambienti Unix. A partire da NFSv4, definito nell'RFC 3530, è possibile usare GSSAPI come metodo di autenticazione sicura di RPC e di NFS. Le reference implementation per Linux e FreeBSD possono essere trovate sul sito Internet http://www.citi.umich.edu/projects/nfsv4/, mentre il client HummingBird Maestro (http://www.hummingbird.com/products/nc/nfs/index.html) fornisce analoghe funzionalità per Windows attraverso le SSPI.

SAMBA

Samba è un file e printer server per i client Window, in grado di emulare in piena regola un server Windows NT, incluso le funzionalità di Primary e Backup Domain Controller. A partire dalla versione 3, Samba è in grado di far parte di un domino Active Directory e nella prossima versione sarà in grado da fungere da Active Directory Server. In questo modo, sarà possibile effettuare un Single SignOn completo in ambiente Windows con server unicamente basati su piattaforma Unix.

12. POSSIBILI ATTACCHI A KERBEROS E CONTROMISURE

Anche Kerberos non è immune agli attacchi, tuttavia possiamo rendere più difficile l'accesso ad un potenziale agressore in quanto esistono delle contromisure che possiamo adottare. Vediamo in particolare quali sono i possibili attacchi.

Compromissione del KDC. Una compromissione a livello di root di uno qualsiasi dei KDC (sia master che uno degli slave) da la possibilità all'eventuale attaccante di prendere il controllo totale del sistema di autenticazione Kerebros. Anche se il database è criptato su disco con la master key, quest'ultima è anche tenuta sul disco del KDC in quanto è possibile avviare il server senza intervento manuale quando il servizio viene avviato. In più l'accesso al database Kerberos viene sempre garantito all'amministratore di sistema (root o Administrator), per cui un eventuale break-in sul KDC potrebbe potenzialmente portare ad una compromissione dell'interno database di autenticazione Kerberos. Il suggerimento in questo caso è quello di proteggere il KDC attraverso l'hardening della macchina, rimuovendo tutti i servizi non necessari, applicando un personal firewall e applicando una password policy.

Compromissione dell'amministratore Kerberos. Se un potenziale intruso ottiene la password del principal collegato all'amministratore Kerberos, allora l'attaccante ha completo accesso al database Kerberos. Molte implementazioni del KDC permettono ad un amministratore di fare un dump remoto del contenuto del database per effettuare il backup dei dati: un attaccante può usare questa funzionalità per scaricare il database sulla sua macchina e tentare un brute-force attack in off-line. In più, avendo pieno accesso alle funzionalità del KDC, l'intruso potrà creare e modificare utenti Kerberos. Il suggerimento è quello di dare l'accesso di amministratore ad un numero molto ristretto di persone e su queste applicare una policy di password molto stretta, ad esempio costringendo gli amministratori a cambiare password ogni mese.

Compromissione di una macchina server. Per effettuare una mutual authentication in Kerberos tra server e client, il servizio deve accedere al suo principal associato. Il service principal risiedono nel filesystem del server, sia esso in un keytab Unix o negli LSA Secret in Microsoft (rif. KnowledgeBase Microsoft Q184017). Se un attaccante ottiene accesso di root alla macchina, tutti i servizi Kerberos erogati dalla stessa sono compromessi. In più alcuni servizi come l'Andrew Filesystem (AFS) hanno lo stesso principal in tutti i server, per cui tutti i file nella cella AFS sarebbero compromessi. Una volta che l'intruso venisse in possesso del principal, potrebbe impersonare quel servizio e decriptare il traffico tra client e server. Ovviamente la sicurezza di un servizio Kerberos dipende dalla macchina su cui questo è eseguito, per cui si consiglia di fare l'hardening anche delle macchine in cui si stanno erogando i servizi Kerberos.

Compromissione di una macchina client. Una compromissione della macchina client potrebbe permettere ad un intruso di prendere i Kerberos tickets in cache sulla macchina. Siccome i ticket sono limitati nel tempo, non è una gravissima compromissione; avendo l'accesso alla macchina client, l'attaccante potrebbe installare un key logger e rilevare tutte le password inserite dall'utente. Il suggerimento è in caso di compromissione di cambiare immediatamente le password e di impostare tra le password policies un cambio password più frequente.

Compromissione delle credenziali di un utente. Ci sono due possibilità in questo caso: catturare le credenziali utente (ticket) presenti nella cache o la password utente. Come espresso precedentemente, il ticket in cache ha un tempo limitato e per cui alla scadenza del ticket l'attaccante non avrebbe più modo di accedere ai servizi. Il secondo è più importante ed è necessario cambiare immediatamente la password.

Brute force attack e attacchi tramite dizionari. Un classico degli attacchi: l'intruso cercherà di ottenere le password di amministratore o di un utente, tentando di indovinarla tramite un dizionario ben definito o di ricavarla tramite un attacco di forza bruta. In questo caso il consiglio è di rendere complicata la password, impostando le policy in modo da introdurre caratteri non facili da indovinare.

Replay attacks. Uno degli attacchi che potrebbero perpetrarsi ai danni di Kerberos è un replay attack. Il replay attack avviene quando un utente legittimo si sta autenticando al KDC per accedere ad un determinato servizio e un intruso si impossessa del traffico di autenticazione. In questo modo, l'intruso userà il ticket ottenuto per riproporsi al servizio e autenticarsi. È da notare che in questo scenario non viene cercata la password dell'utente in nessun modo. Vediamo come avviene in pratica. Esistono tre protezioni già implementate in Kerberos, ovvero *Address field in tickets, Time-based authenticators* e le *Replay caches.* La prima protezione inserisce l'IP address della workstation per cui il ticket è valido; la seconda inserisce un timestamp (o authenticator) nel ticket di accesso al servizio, in modo che la sessione non sia valida dopo il tempo di emissione del ticket (più il clockskew di tolleranza). L'ultimo livello di protezione implementato in Kerberos 5 è la replay cache, che impedisce ad un attaccante di riusare il ticket di accesso durante la tolleranza temporale del clockskew. Ogni servizio mantiene una cache di authenticators che ha ricevuto recentemente: se il servizio trova una copia dell'authenticator già in cache, rigetta la richiesta, altrimenti la accetta e aggiunge l'authenticator nella cache per validare le richieste successive.

Attacchi di tipo Man-in-the-Middle. In questo attacco l'intruso cerca di impersonare il KDC o il server di accesso, in modo che l'utente pensi che sia

collegato al server legittimo, mentre sta parlando con un server fittizio introdotto illecitamente che redirige l'utente al server reale. In questo modo l'attaccante ha il controllo della sessione, per cui impersonare l'utente durante la conversazione e cambiare i messaggi che transitano. Esiste un programma che dimostra l'efficacia dell'attacco man-in-the-middle, ovvero il KDCspoof disponibile sulla URL http://www.monkey.org/~dugsong/kdcspoof.tar.gz. Il suggerimento in questo caso è di abilitare la *mutual authentication*, così che anche il server debba dimostrare la sua identità all'utente: il server sfrutterà la sua propria chiave per criptare una risposta e mandarla indietro al client. Se il server non fosse quello lecito, e quindi non in possesso della service key, allora il server non riuscirebbe ad inviare una risposta valida e il client si disconnette automaticamente.

Tutti questi scenari inducono ad un concetto fondamentale: **installare Kerberos nella propria rete non diminuisce l'importanza di rendere sicure tutte le macchine installate**, compresi i desktop. Una compromissione di una macchina potrebbe comunque avere ripercussione sulla sicurezza dell'ambiente Kerberos. Riassumiamo brevemente in una lista di sei semplici passi che possiamo implementare per proteggerci da questi tipi di attacchi.

1. Fare l'hardening di tutti i server che erogano i servizi Kerberos, in particolare quei server che svolgono il ruolo di KDC.
2. Proteggere i KDC attraverso IP filtering e segmentazione della rete.
3. Impostare una corretta policy delle password, ad esempio costringendo l'utente a scegliere una password di almeno 8 caratteri alfanumerici e facendola cambiare spesso senza reinserire password già usate in precedenza. Nell'implementazione MIT di Kerberos si implementa attraverso il comando kadmin: *add_policy [-maxlife time] [-minlife time] [-history num] policy_name*; per maggiori informazioni, si faccia riferimento alle man pages. Su Windows invece si trova in *Domain Security Policy*, aprire *Security Settings*, Account *Policies* e infine *Password Policy*; i parametri sono *Minimum password age*, *Maximum password age* ed *Enforce password history*.
4. Proteggere i client con un personal firewall, un antivirus, un anti-spyware e assegnando all'utente i minori privilegi possibili sulla macchina.
5. Usare la pre-autenticazione: sul KDC di Microsoft è abilitata di default, invece sul MIT Kerberos è necessario effettuare il seguente comando kadmin: *modify_principal +requires_preauth principal*.
6. Uso di sessioni criptate attraverso SSL o SSH.

CONCLUSIONI

Nel mio lavoro ho avuto occasione di sentire ancora commerciali che continuano a vendere firewalls come "la soluzione" di sicurezza: "hai un problema di sicurezza? Allora inseriamo un firewall". Scommetto che anche voi lettori avete sentito almeno una volta questa frase. Ma siamo veramente sicuri che il firewall ci possa servire? Ormai i firewall come li conoscevamo prima non esistono più, sono "morti" per "risorgere" negli apparati di rete (switch e router): per fare un paragone automobilistico, tempo fa avere i freni a disco era un fattore di differenza, ora diamo per scontato che l'automobile li abbia. Quando andiamo a comprare una nuova macchina cerchiamo altre dotazioni di sicurezza, come ABS e antipattinamento; allo stesso modo in campo informatico dobbiamo guardare oltre all'IP filtering, focalizzandoci sulla sicurezza applicativa.

La sicurezza non è e non deve essere sinonimo di complessità, bisogna che sia il più possibile trasparente all'utente: è possibile far convivere la sicurezza con la semplicità d'uso, ad esempio abilitando tramite policy di Active Directory l'uso di IPSec verso i sistemi principali. Con un semplice gesto, e senza che l'utente si accorga minimamente del cambiamento, abbiamo reso più sicura la nostra rete ed i nostri dati. Ricordiamoci che come amministratori di rete o di sistemi, o come IT managers, l'utente finale è il nostro vero valore e non bisogna farlo sentire a disagio: se applichiamo complesse procedure rischiamo che l'utente tenti di "aggirare l'ostacolo", con gravi ripercussioni sulla sicurezza della nostra rete.

Kerberos, se usato congiuntamente ad altri protocolli sicuri (es: SSL ed SSH), può dare alla propria infrastruttura una buona sicurezza applicativa, non aggravando l'utente finale di ulteriori password da ricordare o complesse procedure.

BIBLIOGRAFIA

Microsoft Corp.
Single Sign-On in Windows 2000 Networks
Whitepaper, 1998

Mark Walla
Kerberos Explained
Articolo di "Windows 2000 Advantage", Maggio 2000
http://www.microsoft.com/technet/prodtechnol/windows2000serv/maintain/security/kerberos.mspx

Turbo Fredriksson
LDAPv3
How-To, Novembre 2003

Jennifer G. Steiner, Clifford Neuman, Jeffrey I. Shiller
Kerberos: An Authentication Service for Open Network Systems
WhitePaper, Marzo 1988

J. Linn
Generic Security Services Application Program Interface Version 2, Update 1
RFC 2743, Gennaio 2000

J. Wray
Generic Security Service API: C-bindings
RFC 1509, Settembre 1993

J. Linn
The Kerberos Version 5 GSS-API Mechanism
RFC 1964, Giugno 1996

J. Myers
Simple Authentication and Security Layer (SASL)
RFC 2222, Ottobre 1997

OpenLDAP project
OpenLDAP 2.2 Administrator's Guide
Manuale, Dicembre 2003

E. Baize,D. Pinkas
The Simple and Protected GSS-API Negotiation Mechanism
RFC 2478, Dicembre 1998

J. Brezak
HTTP Authentication: SPNEGO Access Authentication As implemented in Microsoft Windows 2000
Internet Draft, Ottobre 2002

Microsoft Corp.
Step-by-Step Guide to Kerberos 5 (krb5 1.0) Interoperability
Whitepaper, Gennaio 2000
http://www.microsoft.com/technet/prodtechnol/windows2000serv/howto/kerbstep.mspx

Streicher Bremer
Linux-AD Integration
How-To, Febbraio 2002
http://jaxen.ratisle.net/~jj/nss_ldap-AD_Integration_how-to.html

David "Del" Elson
Active Directory and Linux
Articolo, Aprile 2002
http://www.securityfocus.com/infocus/1563

Andrew G. Morgan
The Linux-PAM System Administrators' Guide
Manuale, Giugno 2002

Jason Garman
Kerberos: The Definitive Guide
ISBN 0-596-00403-6, Agosto 2003

Giuseppe Paternò
Sicurezza Nelle Wireless LAN
ISBN 88-901141-0-X, Agosto 2003

INDICE