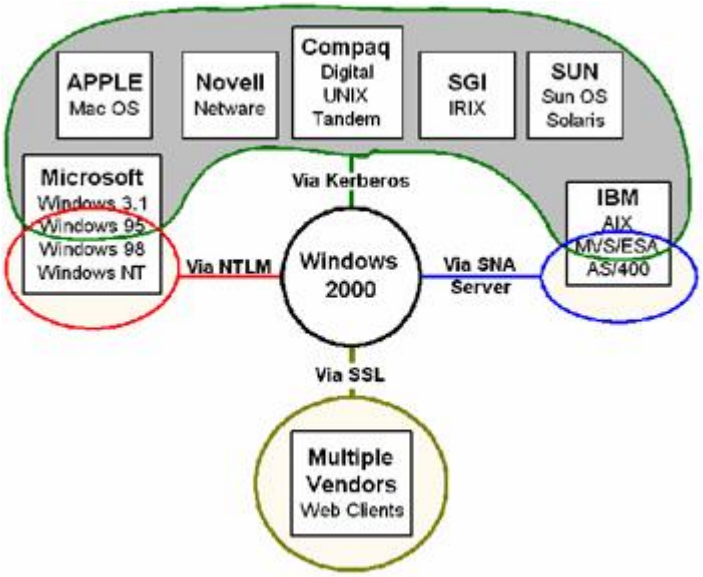
Sicurezza in Windows NT/2000
Single sign on

L'acronimo SSO (*Single Sign-On*) si riferisce a qualsiasi tecnologia che sia in grado di facilitare il *logon* e l'accesso alle risorse della rete da più piattaforme. La procedura SSO non è necessariamente composta da una sola fase, così come non sempre prevede un *sign-on*. Per esempio, a volte le soluzioni SSO richiedono ulteriori informazioni di autenticazione per accedere a un sistema o a una rete differente. I meccanismi di SSO possono inoltre autenticare un utente che non ha compiuto il *sign-on* sul sistema (una cosa forse non particolarmente utile, ma comunque possibile).



Lo schema SSO permette di risolvere molti problemi delle organizzazioni che si affidano a importanti applicazioni che richiedono più di una piattaforma. Per l'autenticazione su più sistemi, gli utenti devono tenere a mente varie *password* ed essere in grado di usare interfacce differenti. In conseguenza, spesso scelgono *password* molto semplici oppure le scrivono su un foglio di carta, provocando in questo modo dei potenziali punti di debolezza nella sicurezza. Gli amministratori devono inoltre navigare attraverso molte interfacce di gestione per aggiungere nuovi utenti, eliminarne altri, oppure cambiare le *password* su vari sistemi diversi come NT, *Unix* e *Novell NetWare*. Questo procedimento è scomodo e può ridurre la sicurezza della rete. Lo schema SSO consente invece di risolvere questi problemi.

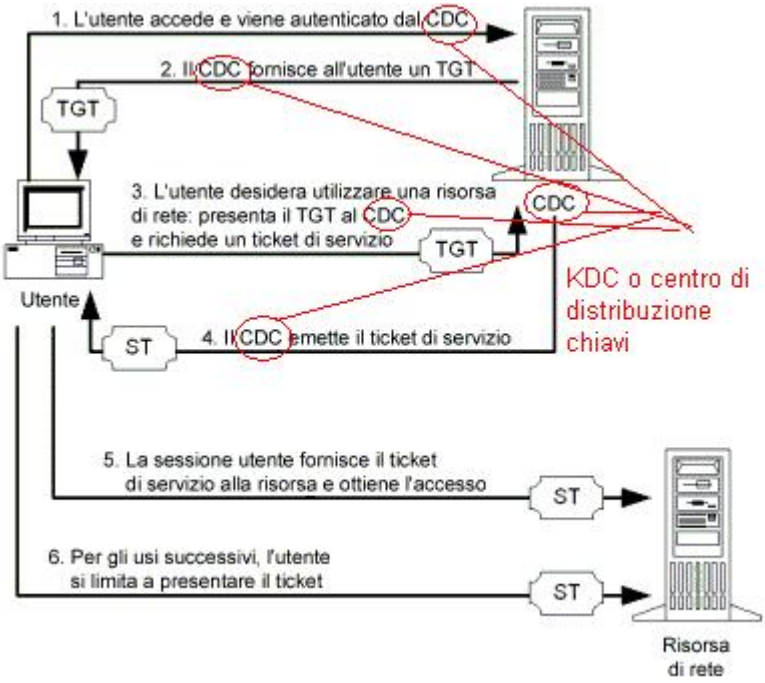
La procedura di SSO è più complessa negli ambienti misti, dal momento che nessuna soluzione universale va bene per qualsiasi ambiente e sistema operativo. In conseguenza, le organizzazioni devono identificare la portata della soluzione SSO e limitarla soltanto ad alcune applicazioni o piattaforme. Sono disponibili varie tecnologie SSO che permettono di adattare le soluzioni a problemi specifici. Queste alternative SSO comprendono la sincronizzazione delle *password*, l'automazione del *logon* e l'autenticazione basata su *token*.

SSO in Windows

La funzionalità SSO è disponibile in modo nativo nei domini *Windows* 2000 mediante il protocollo di autenticazione *Kerberos*, ovvero il protocollo di autenticazione predefinito di *Windows* 2000. L'utilizzo del protocollo *Kerberos* contribuisce in modo significativo al miglioramento delle prestazioni di rete, alla semplificazione della gestione amministrativa nonché alla protezione.

Il protocollo *Kerberos* si basa sul concetto di ticket, ovvero pacchetti di dati crittografati emessi da un'autorità ritenuta affidabile detta Centro distribuzione chiavi (KDC, *Key Distribution Center*). Un

ticket comprova l'identità di un utente e contiene anche altre informazioni. Un centro KDC fornisce i ticket per tutti gli utenti della propria area di autorità o area di autenticazione. In *Windows* 2000, ogni *domain controller* funge da centro KDC e l'area di autenticazione di un *domain controller* corrisponde al dominio. Per ulteriori informazioni sul protocollo *Kerberos*, vedere il *white paper* di *Microsoft* dedicato all'anteprima delle funzionalità di protezione della rete mediante la tecnologia dei servizi di protezione distribuita di *Windows* 2000.



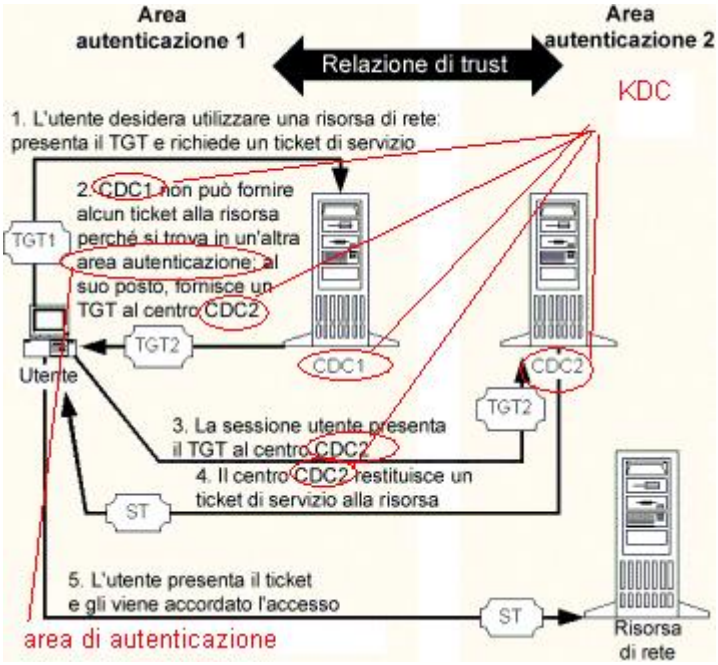
Al momento dell'accesso, l'utente viene autenticato da un centro KDC, che gli fornisce un ticket iniziale detto ticket di concessione (TGT, *Ticket Granting Ticket*). Quando l'utente deve utilizzare una risorsa di rete, la sessione utente corrispondente presenta il TGT al *domain controller* e richiede un ticket per quella determinata risorsa detto ticket di servizio (ST, *Service Ticket*), il quale viene presentato alla risorsa, che a sua volta concede l'accesso all'utente.

L'integrazione del protocollo *Kerberos* nel modello di protezione e amministrazione di *Windows* 2000 consente di gestire e controllare gli utenti e le risorse di rete in modo semplice ed efficiente. Un inconveniente ricorrente delle precedenti implementazioni del protocollo *Kerberos* è rappresentato dal fatto che esse sono aggiunte al sistema operativo anziché essere integrate. Il *software Kerberos* funziona al di sopra della normale architettura di protezione del sistema operativo piuttosto che come parte di esso, pertanto occorre che le informazioni specifiche della funzionalità SSO vengano memorizzate separatamente dalle altre informazioni di sistema e che l'amministratore apprenda l'utilizzo di ulteriori strumenti amministrativi al solo scopo di gestire l'infrastruttura SSO.

Tuttavia, il protocollo *Kerberos* è completamente integrato in *Windows* 2000 e ne rappresenta il metodo di autenticazione predefinito. Le informazioni relative alla funzionalità SSO vengono memorizzate in *Active Directory* insieme a tutte le altre informazioni concernenti gli oggetti di rete.

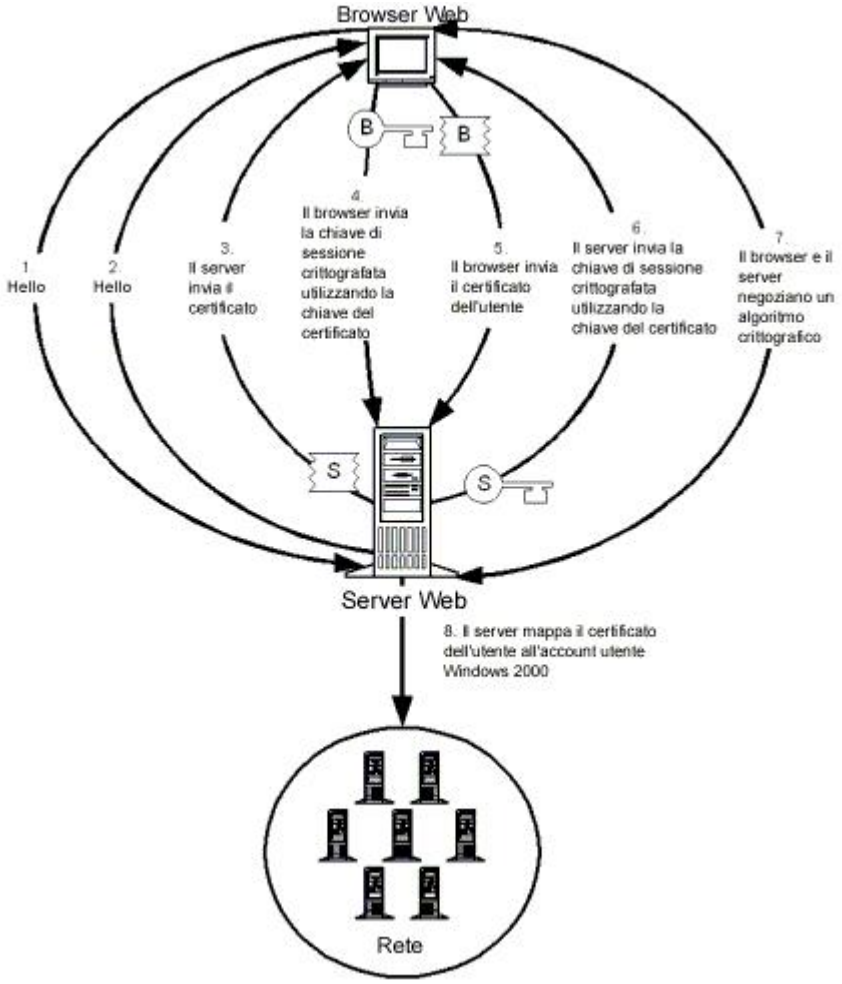
Le relazioni di *trust* tra domini in realtà presentano i *domain controller*, ossia i centri KDC *Kerberos*, ai due domini. Come illustrato di seguito, quando un utente di un dominio deve accedere a una risorsa di un dominio con il quale esiste una relazione di *trust*, il *domain controller* dell'utente provvede, mediante un riferimento tra aree di autenticazione, alla richiesta di un ticket al *domain controller* che possiede la risorsa. Tale *domain controller* considera affidabile il riferimento ed emette il ticket per l'utente. Questa integrazione è uno dei motivi fondamentali per cui le reti basate su *Windows* 2000 possono raggiungere dimensioni notevoli pur garantendo il controllo locale sulle

risorse locali. Per impostazione predefinita, esiste una relazione di *trust* reciproca tra tutti i domini di una struttura di domini *Windows* 2000, che accettano pertanto i rispettivi riferimenti. Non esistono relazioni di *trust* predefinite tra le foreste in *Windows* 2000, ma è possibile definirle in modo semplice, se necessario.



In una rete omogenea basata su *Windows* 2000 l'amministratore non è costretto ad eseguire attività amministrative specifiche della funzionalità SSO, poiché tali funzioni sono integrate nelle funzionalità amministrative. Ad esempio, l'amministratore non ha mai bisogno di stabilire le chiavi crittografiche condivise dall'utente e dal *domain controller*. Quando l'amministratore crea l'*account* utente, le chiavi condivise vengono generate in modo trasparente nell'ambito del processo di creazione dell'*account* e vengono distribuite in modo protetto quando necessario. Analogamente, quando l'amministratore stabilisce le relazioni di *trust* tra i domini, le chiavi necessarie per gestire i riferimenti tra le aree di autenticazione vengono generate in modo trasparente e scambiate in modo sicuro.

Il protocollo SSL utilizza i certificati digitali come base per l'autenticazione. Per certificato digitale si intende un pacchetto di dati a prova di manomissione emesso da un'autorità di certificazione (CA), che fornisce una chiave pubblica e un nome e comprova che la persona o il *server* indicati sono i proprietari di tale chiave pubblica. Nella forma in cui vengono utilizzati in riferimento alla funzionalità SSO, i certificati vengono scambiati tra il *client Web* e il *server*, quindi le chiavi pubbliche in essi incorporate vengono utilizzate per lo scambio di informazioni quali le chiavi di crittografia delle sessioni. In questo modo vengono autenticati entrambi gli utenti, in quanto, se uno dei due non corrisponde all'entità indicata nel certificato presentato, non gli sarà possibile decifrare quanto inviato dall'altro utente. Per una descrizione completa del protocollo SSL, consultare la bozza delle relative specifiche formulate dal comitato IETF disponibile all'indirizzo: http://search.ietf.org/rfc/rfc2246.txt.



Tutti i componenti necessari a rendere disponibile la funzionalità SSO tramite il protocollo SSL sono inclusi in *Windows* 2000 e completamente integrati nell'architettura di protezione del sistema operativo. La gestione dei certificati digitali è possibile grazie all'infrastruttura a chiave pubblica PKI (*Public Key Infrastructure*) di *Windows* 2000, che è costituita da due componenti: Servizi certificati, che consente agli amministratori di emettere e revocare i certificati; *Active Directory*, che fornisce informazioni sui criteri e sulla posizione delle autorità di certificazione e consente la pubblicazione delle informazioni relative ai certificati revocati. I servizi di *hosting Web* di *Windows* 2000 sono forniti da *Microsoft Internet Information Service* (IIS), il *server Web* incluso nel S.O., che utilizza l'architettura di protezione di *Windows* 2000 per fornire il protocollo SSL e gli altri servizi di protezione.

L'integrazione del protocollo SSL e dei certificati digitali nella piattaforma *Windows* 2000 si riflette nell'integrazione degli strumenti di gestione del protocollo SSL e dei certificati, realizzata nel gruppo di strumenti amministrativi di *Windows* 2000. Tutte le attività amministrative necessarie per rendere disponibile la funzionalità SSO tramite il protocollo SSL vengono svolte mediante *snap-in* MMC quali *Active Directory Manager*, Gestione certificati e Gestione servizio *Internet*.

Le password

NT utilizza il *manager* SAM (*Security Account Manager*) per archiviare e leggere le credenziali degli utenti come le *password*. Dal momento che questo modulo archivia le proprie informazioni nel *database* SAM, si può ritenere che NT sia sicuro soltanto quando lo sono i suoi dati SAM.

NT mantiene sul disco fisso una copia permanente e funzionante del *database* SAM. È possibile accedere a questo *database* sotto la chiave SAM nell'*hive* del Registro di Configurazione

HKEY_LOCAL_MACHINE, sia scrivendo un apposito programma che utilizzando un *editor* del Registro di Configurazione come `regedt32.exe`.

Hash **delle** ***password*** **in memoria**

Per *default*, NT compie il *caching* delle credenziali di *logon* per gli ultimi dieci utenti che si sono collegati in modo interattivo. Lo scopo di questa funzione è quello di consentire all'utente di compiere il *logon* sul sistema anche se questo è scollegato dalla rete, oppure se i *controller* di dominio non sono disponibili. Anche se NT offre una certa protezione per la *cache* delle credenziali di *logon*, se l'ambiente richiede un livello maggiore di sicurezza potrebbe essere necessario disattivare completamente il *caching*, dal momento che qualcuno potrebbe attaccarlo.

È necessario tenere presente che le credenziali della *cache* di *logon* contengono gli *hash* delle *password* di altri *hash*, che rendono i dati difficili da leggere o da usare per compiere un tentativo di *logon* non autorizzato. Attualmente, non si è ancora verificato nessun tentativo noto e coronato da successo di attacco a questa *cache*. Per disattivare il *caching* delle credenziali, è necessario modificare l'elemento *CachedLogonCount* (di tipo REG_DWORD, valore 0) nella chiave:

`HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVerion\Winlogon` `del Registro di Configurazione.`

Il ***database*** **SAM sulla rete**

NT utilizza il protocollo SMB (*Server Message Block*), che è stato sviluppato congiuntamente da *Microsoft, IBM* e *Intel* e definisce una serie di comandi a livello di programma per ottenere o fornire servizi di *file* remoti in un ambiente di rete. Una sessione SMB prevede la trasmissione in rete di pacchetti contenenti informazioni riservate. Tra le altre informazioni, quando NT trasmette i pacchetti durante la fase di autenticazione di una sessione SMB, questi pacchetti contengono tipicamente una versione crittografata del *challenge* NTLM.

Gli attaccanti che utilizzano un qualsiasi *sniffer* di pacchetti tra quelli disponibili sul mercato possono facilmente ottenere i dati inviati attraverso la rete. Anche se il lavoro necessario per ottenere i pacchetti giusti ed estrarre da questi ultimi le informazioni delle *password* non è mai stato particolarmente semplice, la situazione è tuttavia cambiata notevolmente dopo il rilascio di SMB *Packet Capture* da parte di *L0pht Heavy Industries*; si tratta di un *tool sniffer* SMB che ora è stato strettamente integrato in *L0phtCrack*.

Chi dispone di questo *software* può ottenere facilmente dalla rete gli *hash* delle *password* basati sul protocollo SMB. L'*utility sniffer* incorporata in *L0phtCrack* legge silenziosamente gli *hash* delle *password* SMB e li archivia per poterli decodificare.

Una volta decodificate, queste *password* permettono un facile accesso a qualsiasi risorsa di rete alla quale può accedere l'*account* dell'utente. Il rischio in questo caso è evidente, ma la prevenzione è semplice. Per proteggersi nei confronti di un attacco di questo tipo è necessario usare NTLMv2, fornito con il *Service Pack* 4 o 5, oppure un *tool* VPN (*Virtual Private Network*) come il protocollo PPTP (*Point To Point Tunneling Protocol*) di *Microsoft*. NTLMv2 dovrebbe essere sufficiente per proteggere i dati nel momento in cui viaggiano nella LAN interna, mentre il protocollo PPTP aiuta a proteggere le informazioni durante il loro viaggio attraverso le reti non affidabili (per esempio *Internet*).

Il processo di logon

La gestione della sicurezza dell'*account*, è a cura del SAM, che contiene informazioni su tutti gli *account* utente e gruppo. SAM viene utilizzato da LSA per autenticare gli utenti durante il processo

di collegamento, confrontando il nome utente e la *password* inserita dall'utente nella registrazione dell'utente nel *database* SAM. SAM crea anche gli identificatori di sicurezza (**SID**) che identificano univocamente ogni *account* utente e di gruppo. Un **SID** è un numero simile al seguente:

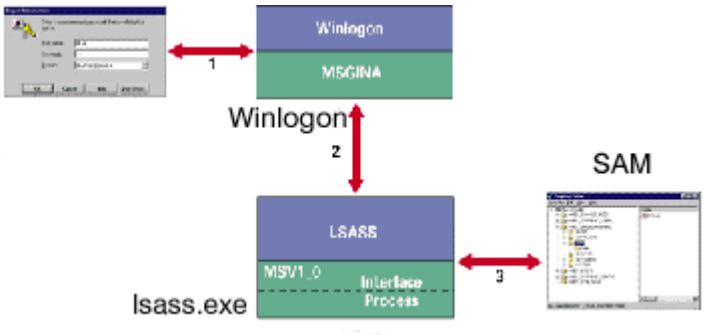
```
S-1-5-21-2087915065-388913830-1877560073-500
```

L'utente, o gruppo, è identificato nel sistema dal **SID**, non dal nome che viene fornito per comodità dagli utenti. Quando un *account* utente o di gruppo viene eliminato, il **SID** viene ritirato. È impossibile quindi ricreare lo stesso *account* utente o di gruppo con lo stesso **SID**. Un nuovo *account* potrebbe avere il vecchio nome, ma gli verrebbe assegnato un nuovo **SID**.

I **SID** sono il motivo per cui occorre particolare attenzione quando si elimina un *account* utente o di gruppo, perché non è più possibile ottenere lo stesso *account*. Se si reinstalla *Windows* NT, operazione diversa da quella di aggiornamento, dopo l'installazione il *computer* potrà avere lo stesso nome, ma avrà un nuovo **SID** e sarà essenzialmente un nuovo *computer* sulla rete. I *database* di sicurezza sulla rete conterranno il vecchio **SID** ma non riconosceranno la nuova configurazione del *computer*, a causa del cambiamento del **SID**.

Procedura di *logon*

I collegamenti obbligatori sono la chiave nel sistema di sicurezza di *Windows* NT. Il processo di collegamento di *Windows* NT segue le seguenti fasi:



1. In *Windows* NT, la combinazione `CTRL+ALT+CANC` produce l'esecuzione del programma *WinLogOn*, un componente di LSA, che visualizza la finestra di dialogo GINA, che richiede l'immissione di un nome utente un dominio e una *password*. Se il nome utente o la parola chiave non sono corretti, il sistema risponde con il messaggio *User Authentication Failure*, che avvisa del fallimento dell'autenticazione dell'utente. Il sistema non specifica se la causa dell'errore risieda nel nome utente o nella *password*.
2. Se il nome utente e la *password* sono validi, LSA passa i dati di *logon* al sottosistema di sicurezza.
3. Il sottosistema di sicurezza passa tali dati al SAM.
4. SAM consulta il *database* della sicurezza dell'*account* per determinare se il nome utente e la *password* corrispondano ad un *account* del *database*. Queste informazioni vengono restituite al sottosistema di sicurezza insieme alle informazioni sui permessi dell'utente, sulla posizione della *home directory* e sul profilo utente. Se l'utente ha un *file script* di *login*, questo viene eseguito.
5. Se è stata trovata una corrispondenza per il nome utente e la *password* e se l'*account* utente ha i permessi sul sistema, il sottosistema di sicurezza genera un *token* d'accesso che rappresenta l'utente e i *server* come una chiave che contiene le credenziali dell'utente sul sistema. Il *token* d'accesso viene passato al sottosistema *Win*32.
6. Il sottosistema *Win*32 genera un nuovo processo che è associato ai *token* d'accesso.
7. Il processo creato da *Win*32 è utilizzato per avviare `explorer.exe` di *Win*32 attivando il

desktop.

Nell'ambiente di domino di *Windows* NT *Server* si verificano due tipi di collegamenti:

- Collegamenti interattivi. Questi si hanno in risposta ad una richiesta di collegamento nella finestra di dialogo specifica. L'utente effettua il *logon* in un dominio o nel *computer* locale a seconda se il campo *Domain* specifichi un dominio o il nome di un *computer* locale.
- Collegamenti remoti. Quando un utente è già collegato ad un *account* utente e cerca di stabilire una connessione di rete ad un altro *computer*, si verifica un collegamento remoto. Un esempio di collegamento remoto è il tentativo di sostituire un'unità di disco di rete o di accedere ad una stampante condivisa.

Supportando i collegamenti remoti, *Windows* NT *Server* consente agli utenti di accedere alle risorse disponibili su altri *computer* e in altri domini.

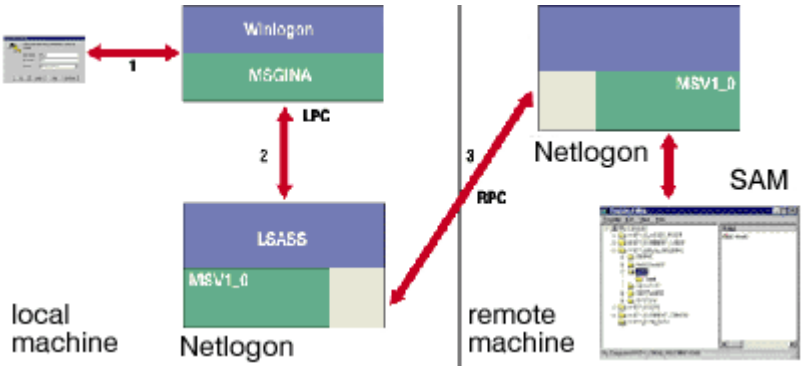
Il processo di net-logon

Il processo di collegamento è controllato dal servizio *NetLogon* che dipende dal servizio *Workstation*. Il servizio *NetLogon* ha tre compiti rispetto al processo di collegamento:

- **Ricerca**. Quando un *computer* con *Windows* NT viene attivato, il servizio *NetLogon* tenta di individuare un *controller* di dominio (un PDC o un BDC) per il dominio che è specificato nel campo *Domain* della finestra di dialogo *LogOn*. Il controllore di dominio scoperto viene quindi contattato per l'autenticazione del collegamento dell'utente.
- Attivazione di un **canale sicuro di comunicazioni**. I *computer* con *Windows* NT che comunicano richieste e risposte su questioni per stabilire l'esistenza di *account* utente validi. Dopo la verifica, viene stabilita una sessione di comunicazione tra i *computer* e vengono trasmessi i dati d'identificazione degli utenti.
- L'**autenticazione *pass-through***. Quando un utente tenta di accedere a una risorsa di rete e l'accesso richiesto non può essere autenticato da un *controller* di dominio del dominio di collegamento dell'utente, la richiesta viene passata a un *controller* di dominio nel dominio di destinazione per l'autenticazione.

L'autenticazione *pass-through* rende la struttura della rete considerevolmente più trasparente all'utente che sta tentando di accedere alle risorse condivise. L'utente viene esplicitamente autenticato durante il collegamento a un dominio. I tentativi di accedere alle risorse in altri domini sono gestiti attraverso l'autenticazione *pass-through*, eliminando la necessità di collegarsi esplicitamente a ulteriori domini. I domini a cui un utente può accedere attraverso l'autenticazione *passthrough* sono determinati da relazioni di fiducia che vengono stabilite tra i domini.

Il servizio *NetLogon* è responsabile anche della duplicazione dei cambiamenti nel *database* SAM sul PDC nei BDC nel dominio. Il processo di recupero della coerenza delle diverse copie del *database* SAM viene chiamato sincronizzazione.



Controllo d'accesso

Sebbene sia possibile configurare, a livello di sistema ed attraverso il *database* SAM, le aree di controllo relative a politica degli *account*, diritti degli utenti e politica di *audit*, molte altre aree di controllo (1 - **ACL**, 2 - *audit control lists*, 3 - autorità amministrativa e 4 - servizi di sistema) operano invece indipendentemente dal *database* SAM.

1. **ACL**. È possibile usare i permessi a livello di oggetto per accedere ad oggetti quali *file* e cartelle. Ciascun oggetto dispone di una lista di permessi, chiamata **ACL**, che definisce gli utenti ed i gruppi che possono accedere all'oggetto e in che modo possa avere luogo questo accesso. Non bisogna confondere i permessi con i diritti; si tratta infatti di elementi completamente diversi in NT. I diritti vengono assegnati a livello di sistema tramite *User Manager*. I permessi vengono invece definiti a livello di oggetto, tramite le liste **ACL**. Per visualizzare la lista **ACL** di un *file*, aprire *Windows Explorer*, fare clic sul *file* con il pulsante destro del *mouse* e selezionare *Properties*. Porsi in corrispondenza della scheda *Security* e fare clic su *Permissions*. NT accorda agli utenti un accesso cumulativo, a seconda dell'appartenenza ai gruppi.
2. ***Audit control list***. Ciascun oggetto dispone anche di una *audit control list*, che definisce i tipi di accesso all'oggetto che NT dovrebbe registrare nel *log Security* locale. In corrispondenza di ciascun tipo di accesso (per esempio, *Read*, *Write*, *Delete*), è possibile indicare se NT deve registrare gli eventi riusciti o falliti. Per esempio, per effettuare il *logging* dei tentativi effettuati dagli utenti per leggere un documento confidenziale al quale non hanno accesso, bisogna attivare l'*auditing* per quel documento in modo che NT registri i tentativi *Read* falliti da parte dei membri del gruppo *Everyone*. Per tener traccia dei cambiamenti apportati ad un *file*, attivare l'*auditing* di quel *file* in modo che NT registri i tentativi *Write* riusciti. Per visualizzare la *audit control list* di un *file* o di una cartella, aprire *Windows Explorer*, fare clic sull'oggetto con il pulsante destro del *mouse* e selezionare *Properties*. Porsi in corrispondenza della scheda *Security* e fare clic su *Auditing*. Per visualizzare la *audit control list* di una chiave del registro di configurazione, aprire `regedt32`, selezionare la chiave e quindi selezionare *Security*, *Permissions*, oppure *Security*, *Auditing* dalla barra del menu.
3. **Autorità amministrativa**. I gruppi *Administrators* e *Power Users*, incorporati in NT, offrono la possibilità di controllare l'autorità amministrativa. I membri del gruppo *Administrators* possono condividere *directory* e stampanti, creare e gestire *account* macchina locali e gruppi macchina locali nel *database* SAM locale del *computer*, assegnare diritti agli utenti ed impostare la politica di *audit*. I membri del gruppo *Power Users* possono condividere *directory* e stampanti, creare e gestire *account* macchina locali e gruppi macchina locali.
4. **Servizi di sistema**. I servizi rappresentano le porte di accesso al sistema NT per l'ingresso dalla rete; in conseguenza, è necessario identificare tutti i servizi in esecuzione su ciascun *computer* e stabilire se sia opportuno disattivare qualcuno di questi servizi. Per visualizzare i servizi del sistema locale, aprire la *applet Services* del pannello di controllo. Per visualizzare i servizi di un sistema remoto, aprire *Server Manager* (sotto *Administrative Tools*), selezionare il *computer* desiderato, quindi selezionare *Computer, Services* dalla barra del menu (se sul vostro sistema non riuscite a trovare *Server Manager*, copiate semplicemente `srvmgr.exe` da un *controller* di dominio).

Sicurezza a livello di dominio

Anche se i sistemi NT possono funzionare in modo indipendente e sicuro anche senza i vantaggi di un dominio, il fatto di basarsi unicamente sulla sicurezza a livello di *host* è accompagnato da alcune limitazioni che si manifestano quando la rete ha molti utenti che devono accedere alle risorse su vari *server*. Quando un *computer* non è membro di un dominio, per effettuare il *logon* gli utenti devono usare un *account* macchina locale nel *database* SAM di quel *computer*. In mancanza di un dominio, ciascun utente ha bisogno di vari *account* utente - uno per ciascun *server* al quale deve accedere. La presenza di molteplici *account* per ogni utente comporta vari problemi.

Gli utenti vengono sovraccaricati dal lavoro legato alla sincronizzazione delle *password* per ogni *account*. Quando i dipendenti abbandonano l'azienda, bisogna individuare e cancellare tutti i loro *account*. È facile dimenticarsi di un *account* e ritrovarsi con un problema di accesso residuo (in altre parole, gli utenti sono sempre in grado di entrare nella rete, anche molto tempo dopo che hanno lasciato l'azienda). Per risolvere con facilità questi problemi, è opportuno creare un dominio ed assegnare a ciascun utente un *account* di dominio che risulti valido per tutti i *server* ai quali deve accedere.

Quando una *workstation* o un *server* membro si unisce ad un dominio, non perde nessuna delle sue impostazioni di sicurezza specifica a livello di *computer*, ma ottiene ulteriori funzionalità a livello di dominio. Quando un *server* o una *workstation* si unisce al dominio, accorda la fiducia ai *controller* di dominio per l'autenticazione degli utenti. In conseguenza, questi ultimi possono effettuare il *logon* usando un *account* macchina locale, oppure un *account* di dominio. Quando un *computer* si unisce ad un dominio, è anche possibile aggiungere gruppi ed utenti del dominio alle liste **ACL** dei *file* e degli altri oggetti sul *computer*. NT aggiunge automaticamente il gruppo *Domain Admins* del dominio al gruppo *Administrators* locale del *computer* (ciò spiega perché i membri di *Domain Admins* abbiano l'accesso amministrativo a tutti i *computer* del dominio.

Mentre i *database* SAM delle *workstation* o dei *server* membri sono specifici e limitati ad un *computer*, il *database* SAM di un *controller* di dominio non lo è affatto. Il SAM di un *controller* PDC è il SAM per l'intero dominio, e *User Manager for Domains* lo apre automaticamente. Quando si crea un utente o un gruppo, oppure si cambia una politica in *User Manager for Domains*, l'*utility* registra la modifica nel *database* SAM del *controller* PDC. Qualche minuto dopo, il *controller* PDC replica questa modifica sul *database* SAM di ogni *controller* BDC (*Backup Domain Controller*). In conseguenza - tenendo conto della latenza tra i cicli di replicazione - tutti i *controller* in un dato dominio condividono un *database* SAM duplicato e dispongono degli stessi *account* utente, gruppi e politiche. Per esempio, quando si attiva l'*auditing* in *User Manager for Domains*, viene attivato l'*auditing* per il *controller* PDC. Quando avviene la replicazione, NT attiva l'*auditing* su ciascun *controller* BDC.

Controllo di dominio o locale

Oltre ad ottenere ulteriori funzionalità di dominio, la configurazione della sicurezza NT di un *computer* opera indipendentemente dopo l'aggiunta del *computer* ad un dominio. Le modifiche apportate tramite *User Manager for Domains* non hanno alcun effetto sulle politiche o sui diritti utente delle *workstation* e dei *server* membri. Per esempio, quando si usa *User Manager for Domains* per assegnare il diritto *Logon locally* a *Peter*, si autorizza questo utente ad effettuare il *logon* sulle *consolle* di tutti i *controller* del dominio. Se tuttavia l'utente deve effettuare il *logon* su un *server* membro del dominio, l'amministratore dovrà fare *logon* su quel *server* ed accordare localmente a *Peter* il diritto attraverso *User Manager*.

La politica degli *account* e la politica di *audit* che vengono specificate in corrispondenza del *controller* di dominio non hanno niente a che fare con la politica di *audit* e gli *account* definiti localmente sulle *workstation* ed i *server* membri. Per esempio, si supponga di fare *logon* su un

controller di dominio, aprire *User Manager for Domains* e configurare la lunghezza minima delle *password*. Questa azione non ha alcun effetto sugli *account* macchina locali definiti nel *database* SAM di una *workstation* o di un *server* membro; in conseguenza, se le politiche degli *account* locali sono piuttosto trascurate, la situazione è critica per la sicurezza. Sarà infatti possibile utilizzare un *account* locale per attaccare un sistema.

In conseguenza, si dovrebbe evitare la creazione di molti *account* macchina locali. È invece opportuno fornire a ciascun utente un *account* di dominio, valido per tutti i *computer* nel dominio. Se ci si trova nella situazione in cui un amministratore ha creato degli *account* macchina locali su una *workstation* o un *server* membro, sarà possibile riconfigurare il *database* SAM locale della *workstation* o del *server* membro attraverso la rete. Copiare semplicemente sulla propria *workstation User Manager for Domains* (`usrmgr.exe`) da qualsiasi *server* NT. Aprire il programma e selezionare *User, Select Domain*, ma non inserire un nome di dominio. Inserire invece due sbarrette rovesciate ed il nome di un *computer* remoto (per esempio, se si desidera configurare il *database* SAM su un *computer* chiamato `kramer`, digitare `\\kramer`). Per verificare che venga effettivamente usato il *database* SAM del sistema remoto e non quello del dominio, osservare la barra del titolo della finestra, che dovrà comprendere due sbarrette rovesciate.

La sicurezza dei domini NT non è centralizzata come potrebbe apparire a prima vista. Ogni *computer* dispone di una configurazione discreta, con centinaia di opzioni di sicurezza che richiedono attenzione. Anche negli ambienti di dominio, solo gli *account* ed i gruppi degli utenti risultano veramente centralizzati (e soltanto se si evita la creazione di *account* macchina locali sui *server* membri). In conseguenza, una completa conoscenza delle opzioni di sicurezza NT di base è una pietra miliare per avere una rete sicura.

La sicurezza nei sistemi basati su Windows 2000 e XP

Il cambiamento del livello di autenticazione di maggior risalto in *Windows* 2000 è costituito dall'uso di *Kerberos* e dal nuovo protocollo di autenticazione di *default*. Ogni *server* e *workstation Windows* 2000 è dotato di un *provider* di autenticazione *Kerberos client*. *Windows* 2000 non include tuttavia il supporto *Kerberos* per altre piattaforme *Microsoft*; se si desidera che i propri *client* NT 4, *Windows* 95, o *Windows* 98 effettuino l'autenticazione usando *Kerberos*, sarà necessario aggiornare le *workstation* a *Windows* 2000 *Professional*.

Due altri cambiamenti importanti nel sistema operativo che influenzano l'autenticazione di *Windows* 2000 sono costituiti dal supporto per *Active Directory* quale *database* per la convalida delle credenziali (ogni *controller* di dominio *Windows* 2000 dispone di una *Active Directory*) e l'inclusione di un nuovo *provider* SSP (*Security Support Provider*): il SSP *Negotiate*. Questo nuovo *provider* SSP attiva la negoziazione del protocollo di autenticazione tra il *server* e un *client* connesso tramite RPC (*Remote Procedure Call*).

Per il momento, il SSP *Negotiate* può gestire soltanto i protocolli di autenticazione *Kerberos* e NTLM (NT LAN *Manager*). La sua prima opzione è sempre costituita dal protocollo *Kerberos*. Se quest'ultimo non è disponibile, il sistema passa a NTLM.

Kerberos

Kerberos viene specificato nel documento RFC (*Request for Comments*) 1510 della IETF (*Internet Engineering Task Force*). Ciò ne fa uno *standard* aperto, che può essere usato per fornire lo schema SSO (*Single sign-on*) tra *Windows* 2000 e altri sistemi operativi che supportano un'implementazione *Kerberos* basata sul documento RFC 1510. L'uso di *Kerberos* presenta anche altri vantaggi:

- autenticazione più veloce;
- mutua autenticazione;

- supporto per la delega dell'autenticazione;
- relazioni di fiducia transitive;
- *logon* con *smartcard*.

Grazie al suo univoco sistema di *ticketing*, *Kerberos* non ha bisogno dell'autenticazione *pass-through* e può offrire un'autenticazione più veloce. L'autenticazione *pass-through* è un meccanismo che viene usato durante la sequenza di autenticazione NT 4 basata su NTLM.

Mutua autenticazione: ciò significa che il *client* si autentica con il servizio che è responsabile per la risorsa e che quest'ultimo si autentica con il *client*. Questa è un'altra importante differenza rispetto al protocollo NTLM. Lo schema *challenge*/*response* di NTLM offre unicamente l'autenticazione del *client*: il *server* interroga il *client*, e quest'ultimo calcola una risposta che viene convalidata dal *server*. Usando NTLM, gli utenti potrebbero fornire le proprie credenziali a un *server* fasullo.

Kerberos supporta delle **relazioni di fiducia transitive** interdominio, che permettono di ridurre il numero delle relazioni di fiducia necessarie per l'autenticazione. Ciò significa che se `europe.compaq.com` e `us.compaq.com` accordano la fiducia a *compaq.com*, allora *europe.compaq.com* accorda implicitamente la fiducia anche a *us.compaq.com*. *Kerberos* di *Windows* 2000 supporta inoltre un meccanismo noto come delega dell'autenticazione o inoltro delle credenziali.

La **delega** può essere considerata come un'ulteriore fase dopo l'impersonalizzazione: grazie a quest'ultima, un servizio può accedere alle risorse locali per conto di un utente; grazie alla delega, un servizio può accedere anche alle risorse *remote* per conto dell'utente. Il vero significato della delega è che l'utente A può assegnare alla macchina intermediaria B i diritti di autenticarsi su un *server* di applicazioni C come se la macchina B fosse l'utente A. Il *server* di applicazioni C basa le proprie decisioni di controllo dell'accesso sull'identità dell'utente A invece che sull'*account* della macchina B. Nella terminologia *Kerberos*, ciò significa che i ticket dell'utente possono essere inoltrati da una macchina all'altra.

Attraverso l'estensione *Kerberos* **PKINIT** (*Public Key cryptography for INITial authentication*), *Windows* 2000 include il supporto per il *logon* con *smartcard*. Quest'ultimo offre un'autenticazione molto più forte rispetto al *logon* con *password*, dal momento che è basato su un'autenticazione a due fattori: per effettuare il *logon*, l'utente ha bisogno di una *smartcard* e del relativo codice *PIN* (*Personal Identification Number*). In generale, il *logon* tramite *smartcard* offre anche una sicurezza più forte: permette infatti di bloccare gli attacchi dei cavalli di Troia che cercano di ottenere la *password* dell'utente leggendola dalla memoria del sistema.

Il protocollo *Kerberos* di base, così come viene definito nel documento RFC 1510, gestisce unicamente l'autenticazione. L'implementazione di questo protocollo compiuta da *Microsoft* permette di gestire anche i dati di controllo dell'accesso. I ticket *Kerberos* di *Windows* 2000 contengono le informazioni relative ai privilegi e all'appartenenza ai gruppi dell'utente, in un campo chiamato PAC (*Privilege Attribute Certificate*). *Kerberos* di *Windows* 2000 può inoltre fornire una chiave segreta, che può essere usata per l'autenticazione dei pacchetti, l'integrità e i servizi di riservatezza per i pacchetti scambiati dopo la sequenza iniziale di autenticazione.

Controllo dell'accesso

Windows 2000 include alcune nuove funzionalità per il controllo dell'accesso, ma il modello del controllo dell'accesso è fondamentalmente lo stesso che viene usato in NT 4. Questo modello si basa ancora sui seguenti concetti chiave:

- *access token*;
- *access mask*;

- *security descriptor*;
- *impersonation*.

Il controllo dell'accesso di *Windows* 2000 comprende alcune importanti modifiche relativamente alle liste **ACL** (*Access Control List*), agli elementi **ACE** (*Access Control Entry*) associati, al modo in cui viene amministrato, all'ereditarietà, alle regole di valutazione delle **ACL** e all'**ACL** *editor*.

Il nuovo ACL *editor*

La caratteristica più importante del nuovo **ACL** *editor* di *Windows* 2000 è costituita dall'indipendenza dagli oggetti (lo stesso *editor* viene usato per impostare il controllo dell'accesso su diversi tipi di oggetti che possono essere resi sicuri) e dal supporto per elementi **ACE** *deny* e le nuove regole di valutazione **ACL**. Mentre NT 4 supporta elementi **ACE** negativi, l'**ACL** *editor* non è invece in grado di gestirli. Quando in NT 4 si impostano gli elementi **ACE** *deny* a livello di programma, usando l'**ACL** *editor* di NT 4 veniva visualizzato un messaggio di errore. Le nuove regole di valutazione **ACL** verranno spiegate più avanti. Il nuovo **ACL** *editor* dispone di una **GUI** (*Graphical User Interface*) completamente nuova, che consiste in una vista di base e una vista avanzata.

Per accedere in maniera semplice e immediata all'**ACL** *editor*, è sufficiente selezionare una cartella del *file system* e fare clic con il tasto destro del *mouse* e attivare il tab Protezione.

ACE basati sul tipo di oggetto

Windows 2000 supporta gli elementi **ACE** *object-type*. Si tratta di una funzionalità della versione 4 del modello per il controllo dell'accesso di *Windows*. *Microsoft* ha implementato questa nuova versione unicamente per gli oggetti *Active Directory*. Il motivo principale per cui *Microsoft* ha incorporato questa nuova versione **ACL** è stato quello di attivare in un modo più granulare la definizione del controllo dell'accesso sugli oggetti *Active Directory*.

Queste modifiche attivano anche una delega amministrativa di grana fine sugli oggetti *Active Directory*, un'altra funzionalità chiave di *Windows* 2000. Usando gli **ACE** *object-type*, l'amministratore può creare delle impostazioni per il controllo dell'accesso a grana fine per gli oggetti *Active Directory*. Per esempio, questi elementi si possono usare per definire a quali tipi di oggetti (utenti, gruppi, o *computer*) verrà applicato un elemento **ACE** impostato su un'unità OU (*Organizational Unit*).

Gli **ACE** *object-type* attivano altri **ACE** basati su proprietà, insieme di proprietà e diritti estesi. Gli **ACE** basati sulle proprietà e insiemi di proprietà si possono usare per impostare il controllo dell'accesso su una proprietà o un insieme di proprietà di un oggetto. Alcuni esempi di proprietà dell'oggetto sono *first name*, *Home Directory*, *City* e *Manager's name* di un utente. Alcuni insiemi di proprietà esemplificativi di un oggetto utente sono *Phone and Mail options*, *Account restrictions* o *Personal Information*, *Public Information* e *Home Address*. Un esempio di insieme di proprietà è costituito dal *Public Information*: copre gli attributi *E-mail addresses*, *Manager Common Name* di un utente. Nel *Windows* 2000 *platform* SDK (*Software Development Kit*) viene spiegato come creare degli insiemi di proprietà personalizzati per la propria organizzazione.

I diritti estesi (*Extended Rights*) sono delle speciali azioni o attività correlate agli oggetti *Active Directory*, che non sono coperte da nessuno dei diritti di accesso *Windows* 2000 *standard* (*read*, *write*, *execute*, *delete*). Ciò che li rende così speciali è il fatto che non possono essere collegati alle proprietà degli oggetti. Alcuni validi esempi sono costituiti dai diritti estesi *send as* e *receive as* specifici delle caselle di posta. Anche se i diritti estesi non sono collegati alle proprietà degli oggetti, vengono visualizzati insieme ai permessi *standard* sull'oggetto nell'**ACL** *editor*. Per ottenere una panoramica degli *Extended Rights*, aprire il container *Extended-rights* del contesto di

assegnazione dei nomi di configurazione di *Windows* 2000 *Active Directory* (è possibile esaminare i contenuti di *Active Directory* usando *tool* ADSIEDIT).

Analogamente agli insiemi di proprietà, l'organizzazione può creare ulteriori diritti estesi; come per gli insiemi di proprietà, il modo per configurarli è dettagliato nella documentazione di sistema (*Windows* 2000 *platform* SDK).

Nuove regole di valutazione delle ACL

Tutte le modifiche al controllo dell'accesso che sono state elencate in precedenza hanno forzato *Microsoft* a rivedere le regole e l'ordine di valutazione delle *Discretionary* **ACL**. Nel gergo del sistema, l'ordine di tali regole viene detto ordine canonico e viene visualizzato mediante la vista avanzata dell'**ACL** *editor*.

Questo ordine di valutazione contiene tre regole fondamentali:

- Le impostazioni per il controllo dell'accesso definite in modo esplicito hanno sempre la precedenza sulle impostazioni ereditate. Come è stato indicato in precedenza, questa è una conseguenza diretta del modello di controllo dell'accesso discrezionale usato in *Windows* NT e in *Windows* 2000.
- Le impostazioni per il controllo dell'accesso padre *tier* 1 hanno la precedenza sulle impostazioni per il controllo dell'accesso padre *tier* 2.
- I permessi *deny* hanno la precedenza sui permessi *allow*. In caso contrario, un utente con diritto di accesso *deny* potrebbe ancora essere in grado di accedere a una risorsa sulla base, per esempio, di un **ACE** *allow* per uno dei suoi gruppi.

Auditing

Per quanto alle funzionalità di *auditing* ci si riferisce all'*event viewer* (modificato rispetto alla precedente versione) e l'inclusione del SCTS (*Security Configuration Tool Set*).

L'*event viewer* di *Windows* 2000

Rispetto al suo predecessore in NT 4, l'*Event Viewer* è stato esteso ed include una serie di nuove cartelle per raccogliere le informazioni di *auditing* relative ad alcuni servizi centrali del sistema operativo (come il *Directory Service*, *DNS Server* e il *File Replication Service*). Anche la porzione per la descrizione degli eventi è stata estesa, facilitando la risoluzione dei problemi. Alcuni eventi comprendono inoltre un puntatore HTTP al sito di supporto *on-line* di *Microsoft*. Infine, ora è possibile accedere ai *log* degli eventi usando l'interfaccia di gestione WMI (*Windows Management Instrumentation*).

Due funzionalità che ancora mancano dall'*Event Viewer* di *Windows* 2000 e dal suo sistema per il *log* degli eventi, sono la capacità di impostare il *logging* centralizzato e di archiviare le impostazioni del *log* degli eventi in modo professionale. In questo contesto, il termine *logging* centralizzato indica la raccolta in tempo reale e in un singolo *database* centralizzato di tutti gli eventi di *log* di ogni *computer*. Le organizzazioni che desiderano tenere traccia con l'andare del tempo dei contenuti dei propri *log* degli eventi devono implementare alcune speciali procedure di archiviazione. Inoltre, per attivare il *logging* centralizzato, sono necessari speciali *script* o specifici prodotti *software* di terze parti.

Il *Security Configuration Tool Set*

Usando l'SCTS, l'amministratore può controllare le impostazioni di sicurezza correnti di un

computer confrontandole con i valori definiti in un modello di sicurezza. I modelli di sicurezza si possono definire usando lo *snap-in* MMC (*Microsoft Management Console*) *Security Templates*. Questi modelli contengono ogni tipo di parametro legato alla sicurezza di *Windows* 2000: **ACL** su oggetti del registro di configurazione e *File system*, impostazioni dell'*Event Log*, gruppi *Restricted* (per impostare l'appartenenza ai gruppi), impostazioni dei servizi di sistema e impostazioni delle politiche sugli *account* di sistema.

Microsoft mette a disposizione due categorie di base di modelli per la configurazione della sicurezza:

- *default*;
- *incremental*.

I modelli di sicurezza *default* contengono le impostazioni di sicurezza *Windows* 2000 *default*, così come vengono applicate al sistema *Windows* 2000 durante la normale installazione. È possibile usare i modelli di *default* per configurare una macchina aggiornata da NT 4 a *Windows* 2000 al livello delle impostazioni di sicurezza *Windows* 2000 *default*.

I modelli di sicurezza *incremental* definiscono invece dei livelli di sicurezza *higher* o *lower*; è possibile usare questi modelli per portare una macchina dal livello di sicurezza di *default* a un livello di sicurezza più alto o più basso.

Per esempio, il modello compatws.inf riduce la sicurezza su una macchina *Windows* 2000 *Professional*, in modo da consentire alle applicazioni di scrivere su un maggior numero di chiavi del registro di configurazione.

Il modello hisecdc.inf rafforza invece la sicurezza di un *controller* di dominio *Windows* 2000. Un modello *incremental* non dovrebbe essere mai applicato senza aver prima applicato un modello di *default*. *Microsoft* definisce tre livelli:

- *compatible*;
- *secure*;
- *high*.

Usando lo *snap-in* MMC SCTS, è possibile caricare in SCTS i modelli appropriati a seconda delle proprie esigenze di sicurezza. SCTS usa uno speciale *database* (`secedit.sdb`) per archiviare le informazioni relative al modello di sicurezza che è stato caricato. È possibile eseguire il motore SCTS dallo *snap-in* MMC, oppure dal *prompt* dei comandi (usando l'eseguibile `secedit`). Lo stesso motore viene usato per la porzione di sicurezza delle impostazioni GPO (*Group Policy Object*) *Windows* 2000. SCTS si usa per definire le impostazioni di sicurezza locali. La porzione di gestione della sicurezza GPO può configurare le impostazioni di sicurezza non locali che vengono definite a livello di dominio *Active Directory*, di unità OU, oppure di sito.

Funzionalità di active directory

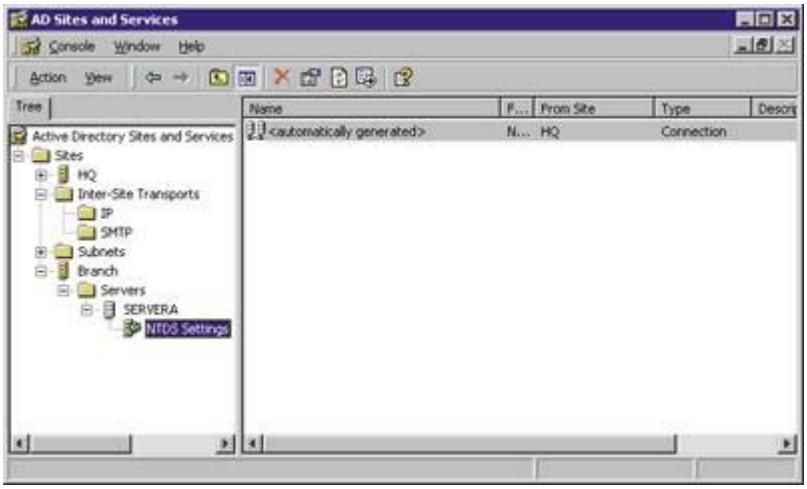
Active Directory fornisce a *Windows* 2000 molte funzionalità che diversificano questo sistema operativo da NT 4.0 e lo rendono più facile da usare nella gestione delle grandi aziende. Queste nuove funzionalità comprendono:

- il *Global Catalog*;
- le unità OU;
- i gruppi espansi;
- la replicazione della *directory*;

- una nuova struttura dei *controller* di dominio.

Global Catalog. Il *Global Catalog* è un nuovo concetto, introdotto per la prima volta con *Windows* 2000. Il catalogo è un indice separato di oggetti in una foresta *Active Directory*. Per *default*, questo indice non contiene tutti gli oggetti presenti nel *database* di *Active Directory*, ma soltanto una parte dei loro attributi. Il *Global Catalog* permette agli utenti di individuare rapidamente gli oggetti della *directory* all'interno della foresta aziendale, senza doversi recare presso il *controller* del dominio in cui risiede l'oggetto. Il *Global Catalog* viene usato al meglio quando sono presenti più domini e alberi, sparsi su varie reti. È necessario avere a disposizione sulla rete almeno un *server Global Catalog* affinché i *client* possano compiere l'autenticazione sui domini *Active Directory*.

Per *default*, il primo *controller* del primo dominio nel primo albero diventa il *server Global Catalog*. Per specificare manualmente altri *controller* di dominio come *server Global Catalog*, si può usare lo *snap-in* MMC (*Microsoft Management Console*) *Active Directory Sites and Services*.



Sebbene la maggior parte delle informazioni del dominio (come utenti e gruppi) venga replicata soltanto sul *controller* all'interno del dominio stesso, *Active Directory* replica il *Global Catalog* attraverso i confini del dominio e verso tutti i *controller* che sono dei *server Global Catalog*. Nel momento in cui viene implementato *Windows* 2000, è opportuno posizionare attentamente i *server Global Catalog*. Ciascuna macchina *client* deve avere un facile accesso al *Global Catalog*, in modo da ottimizzare la capacità del *computer* di compiere una ricerca all'interno della *directory*. Il *Global Catalog* sostituisce inoltre la lista GAL (*Global Address List*) in *Exchange* 2000 *Server* (in precedenza chiamato in codice *Platinum*).

Unità OU. Le unità OU permettono di delegare il controllo delle risorse di dominio *Windows* 2000. Quando si crea una unità OU nel dominio *Active Directory*, viene creato un confine amministrativo all'interno del quale è possibile delegare a un sottoinsieme di utenti la gestione degli oggetti contenuti in questa OU. Come è stato accennato in precedenza, ciascuna unità OU può contenere altre unità, oppure oggetti *leaf* come utenti, *computer*, o stampanti. È possibile nidificare qualsiasi numero di unità OU all'interno di altre; affinché la cosa risulti pratica, è tuttavia opportuno limitare a un massimo di dieci il numero massimo di OU nidificate nel dominio. Per creare le OU nidificate, si può usare lo *snap-in* MMC *Active Directory Users and Computers*.

Ad esempio si consideri l'unità OU chiamata US contiene quella California, che a sua volta contiene quella *Finance*. In quest'ultima è contenuto un oggetto utente chiamato *Joe User*. Si supponga che quest'ultimo sia l'amministratore locale del dipartimento *Finance* in California. Per delegare a *Joe User* il controllo dell'unità OU *Finance* su tutti gli oggetti posti al suo interno, si può usare la procedura di autocomposizione *Delegation of Control Wizard* dello *snap-in* MMC *Active Directory Users and Computers*. Per avviare questa procedura è sufficiente fare clic con il pulsante destro del *mouse* in corrispondenza dell'unità OU, quindi selezionare *Delegate Control*. Successivamente occorre scegliere l'utente o il gruppo a cui delegare il controllo e specificare quali diritti dovranno avere sugli oggetti contenuti nell'unità OU. In alternativa, è possibile selezionare *Custom task* per assegnare i diritti usando un completo elenco. I diritti che si possono assegnare corrispondono alle liste **ACL** (*Access Control List*) di sicurezza sugli oggetti OU cui è stato delegato il controllo. Anche se è possibile modificare manualmente le liste **ACL** su una unità OU, utente, oppure gruppo, in modo da assegnare i diritti di sicurezza per i singoli oggetti, la procedura di autocomposizione

Delegation of Control Wizard offre una semplice interfaccia **GUI** per delegare il controllo.

Gruppi *Windows* 2000. NT dispone di soltanto due tipi di gruppi: globale e locale. Questi gruppi esistono unicamente a fini di sicurezza (per esempio, per assegnare la sicurezza alle risorse) e possono contenere soltanto oggetti utente. *Windows* 2000 dispone invece dei gruppi globale e locale di dominio, oltre a un nuovo gruppo di sicurezza chiamato **gruppo universale**. I gruppi universali diventano disponibili quando i domini *Active Directory* vengono fatti passare dalla modalità mista a quella nativa. In modalità mista, un dominio *Windows* 2000 può contenere dei *controller* di dominio *Windows* 2000 e dei *controller* BDC (*Backup Domain Controller*) di NT 4.0. Nella modalità nativa, i domini non possono invece contenere i *controller* BCD di NT 4.0. Il passaggio alla modalità nativa è una funzione a senso unico: non sarà più possibile ritornare nelle condizioni precedenti.

I gruppi universali possono contenere quelli globali e altri gruppi universali provenienti da qualsiasi dominio della foresta. I gruppi globali sono invece specifici al dominio (un gruppo globale contiene utenti, *computer*, oppure altri gruppi globali provenienti unicamente dall'interno del medesimo dominio). Ovviamente, i gruppi globali di un dominio possono essere membri di gruppi locali di un altro dominio. I gruppi universali permettono inoltre di nidificare nella foresta i gruppi globali e universali di altri domini. In *Windows* 2000 è possibile creare dei gruppi di sicurezza che contengono oggetti macchina. In conseguenza, si possono impostare i permessi di accesso alle risorse usando gruppi basati sulle macchine e non soltanto sugli utenti.

Windows 2000 permette di creare dei gruppi non di sicurezza chiamati gruppi di distribuzione, che hanno un ambito analogo (ovvero locale, globale e universale) a quelli di sicurezza. Questi gruppi funzionano come le liste DL (*Distribution List*): non hanno un contesto di sicurezza ma permettono di raggruppare gli utenti per scopi come la posta elettronica.

Replicazione della *directory*. *Windows* 2000 usa un nuovo modello di replicazione per fare in modo che tutti i *controller* di dominio della foresta dispongano di informazioni aggiornate. Questo modello si basa sul concetto di replicazione *multimaster*. In NT 4.0 soltanto il *controller* PDC (*Primary Domain Controller*) mantiene una copia a lettura/scrittura del *database* SAM. In *Windows* 2000, invece, ciascun *controller* del dominio contiene una copia a lettura/scrittura dell'albero DIT. Gli utenti possono apportare a qualsiasi *controller* di dominio delle modifiche che vengono replicate sugli altri *controller*.

Windows 2000 fa uso di una funzione chiamata numero USN (*Update Sequence Number*) per determinare se è necessario replicare le modifiche da un *controller* di dominio all'altro. Ciascun oggetto e le sue proprietà in *Active Directory* contengono un numero USN, che viene usato dai *controller* di dominio per stabilire quando devono avvenire le modifiche su un *partner* di replicazione. Durante un ciclo di replicazione, le modifiche (e non l'intero oggetto) vengono replicate per ogni proprietà. Per esempio, se il numero di telefono di un oggetto utente cambia sul *controller* di dominio 1, viene replicato sul *controller* di dominio 2 soltanto il nuovo numero (non l'intero oggetto utente). Se la modifica a una proprietà si verifica su due *controller* di dominio, il contrassegno di *data* e ora aiuta a fare in modo che venga presa in considerazione soltanto la modifica più recente. Per replicare le informazioni *Active Directory* e di dominio, i *controller* di una foresta usano tre contesti di assegnazione dei nomi di replicazione. Si può pensare ai contesti di assegnazione dei nomi come ai percorsi seguiti dalle informazioni replicate. Ciascun contesto di assegnazione dei nomi può seguire un percorso differente tra i *controller* di dominio nella foresta, inoltre replica informazioni diverse a seconda del loro ruolo. Il contesto di assegnazione dei nomi di dominio replica le modifiche DIT sui *controller* poste in all'interno del dominio; il contesto di assegnazione dei nomi dello schema replica le informazioni di schema su tutti i *controller* di dominio all'interno di una foresta; il contesto di assegnazione dei nomi della configurazione replica le informazioni di configurazione (come la tipologia di replicazione) su tutti i *controller* di dominio della foresta.

Active Directory usa i siti per consentire di controllare il traffico di replicazione tra ubicazioni caratterizzate da collegamenti WAN lenti. I siti *Active Directory*, proprio come i siti *Exchange Server*, sono aree caratterizzate da un'elevata larghezza di banda di rete. All'interno di un sito, il processo KCC (*Knowledge Consistency Check*) che si trova in esecuzione su ciascun *controller* di dominio genera automaticamente la tipologia di replicazione del *controller* per ogni contesto di assegnazione dei nomi. Il processo KCC crea una tipologia ad anello tra i *controller* di dominio del sito. Con la crescita del numero dei *controller*, il processo KCC aggiunge tra di essi nuovi oggetti di connessione in modo da impedire un numero eccessivo di salti tra due *controller* di dominio qualsiasi. È possibile pianificare manualmente la frequenza di replicazione tra i siti, a seconda di quali siano le proprie esigenze di rete. Per definire i siti manualmente è necessario usare lo *snap-in* MMC *Active Directory Sites and Services*.

Occorre inoltre creare degli oggetti *subnet* che corrispondano a tutte le *subnet TCP/IP* presenti sulla rete, quindi associare queste *subnet* ai siti appropriati. Le *workstation* usano queste informazioni per individuare il *controller* di dominio più vicino ai fini di autenticazione, dal momento che preferiscono usare un *controller* all'interno del sito prima di iniziare a interrogare a caso il servizio DNS alla ricerca degli altri *controller* di dominio disponibili.

Struttura dei *controller* di dominio. In NT 4.0 il *controller* PDC è un singolo punto di modifica per il *database* SAM, oltre che un singolo punto di guasti. Come accennato in precedenza, *Windows* 2000 non richiede il *controller* PDC per le modifiche al *database* SAM, dal momento che il sistema operativo supporta la replicazione *multimaster Active Directory*. In ogni caso, il ruolo di *controller* PDC esiste ancora. Le foreste *Windows* 2000 richiedono i seguenti cinque ruoli *Operations Master* sui *controller* di dominio: PDC, **RID** *Pool, Infrastructure, Domain Naming* e Schema (in precedenza *Microsoft* faceva riferimento ai ruoli *Operations Master* usando l'acronimo FSMO - *Flexible Single - Master Operations*).

I ruoli *Operations Master* PDC, **RID** *Pool* e *Infrastructure* devono risiedere su almeno un *controller* in ogni dominio. Se il *server* che contiene un particolare ruolo si guasta, occorre elevare manualmente a questo ruolo un altro *controller* di dominio. Il ruolo PDC è piuttosto intuitivo: se si dispone di *client* e *controller* BDC NT 4.0 *downlevel*, il *controller Windows* 2000 che ottiene il ruolo PDC è il *controller* PDC del dominio. Il ruolo **RID** *Pool* si riferisce al valore **RID** (*Relative IDentifier*) nell'identificatore **SID** (*Security IDentifier*) di un utente. Dal momento che *Windows* 2000 permette a qualsiasi *controller* di dominio di apportare modifiche alla *directory*, ha bisogno di un metodo per coordinare l'assegnazione degli identificatori **RID** ai nuovi oggetti. L'*Operations Master* **RID** *Pool* riveste esattamente questo ruolo. Il ruolo *Infrastructure* è un processo che mantiene la coerenza interdominio tra gli oggetti replicati attraverso i confini del dominio (per esempio, connessioni di replicazione, configurazione del sito, *Global Catalog*).

I ruoli *Operations Master Domain Naming* e Schema risiedono almeno su un *controller* di dominio della foresta. Il ruolo *Domain Naming* garantisce l'univocità dei nomi di dominio all'interno della foresta quando vengono aggiunti nuovi domini. Il ruolo Schema definisce invece quale *controller* di dominio può apportare modifiche allo schema di *directory*, dal momento che il fatto di permettere a più *controller* di compiere modifiche allo schema della *directory* può generare dei problemi.

Migrazione verso *Active Directory*

Il metodo più semplice di migrazione verso *Active Directory* è quello di aggiornare i domini NT 4.0 in loco, ovvero aggiornare il *controller* PDC nel primo dominio *master*. Dopo l'aggiornamento del *controller* PDC, il primo dominio funziona in modalità mista: i *controller* di dominio *Windows* 2000 ospitano *Active Directory* ma, per i rimanenti dispositivi NT 4.0, hanno lo stesso aspetto dei *controller* di dominio NT 4.0. Non appena il primo *controller* PDC viene aggiornato a *Windows* 2000, tutte le *workstation* e i *server Windows* 2000 possono avvantaggiarsi di alcune funzionalità di *Active Directory* (come le unità OU e le *Group Policies*), senza influenzare i rimanenti dispositivi

NT 4.0. Ogni successivo dominio NT 4.0 che viene aggiornato entra a far parte della foresta che era stata creata in corrispondenza dell'aggiornamento del primo dominio. Se si usano dei domini delle risorse, sarà necessario aggiornare anche questi; successivamente, si possono spostare le risorse in unità OU all'interno di un altro dominio già esistente - riducendo in questo modo il numero totale di domini (con le unità OU non sono più necessari i domini delle risorse). Dopo la migrazione verso *Windows* 2000 di tutti domini, si possono usare dei *tool* di terze parti o le *utility* del *Microsoft Windows* NT *Server* 4.0 *Resource Kit* per consolidarli a seconda delle proprie esigenze amministrative e organizzative.

Un altro metodo per compiere la migrazione verso *Active Directory* è quello di creare da zero l'infrastruttura *Windows* 2000 e usare dei *tool* di terze parti (come DM/*Manager* di *FastLane Technologies*, *DirectMigrate* 2000 di *Entevo*, *OnePoint Domain Administrator* di *Mission Critical Software*), oppure le *utility* del *Resource Kit* (per esempio gli *script* di *history cloning* **SID**) per far migrare uno alla volta i gruppi di utenti. Questo approccio conservatore offre un metodo organizzato per aggiornare a *Windows* 2000 utenti e *computer*, senza necessità di ricorrere a un approccio di tipo tutto-o-niente. La possibilità di iniziare da zero permette di evitare il problema della gestione dell'eredità dell'infrastruttura NT già esistente, quando si cerca di muoversi in avanti. Questa strategia alternativa offre inoltre un'opzione di *back-out*, dal momento che i *tool* di terze parti e le *utility* del *Resource Kit* permettono di ricreare o clonare gli oggetti NT 4.0 nella foresta *Windows* 2000 lasciando intatti gli oggetti NT 4.0 già esistenti.

Componenti dell'infrastruttura PKI

La base dell'infrastruttura PKI di *Microsoft* è costituita dalla **API** di crittografia: la CryptoAPI 2.0. Questa **API** mette disposizione un servizio di crittografia e di gestione dei certificati per la sicurezza a chiave pubblica. Il servizio crittografico compie alcune funzioni, come quelle relative alla generazione delle chiavi, all'*hashing* dei messaggi, alle firme digitali e alla codifica.

Il servizio di gestione dei certificati mette invece a disposizione la gestione e l'archiviazione dei certificati digitali *X*.509v3. L'infrastruttura PKI offerta da *Windows* 2000 è formata da vari componenti:

- i *provider* CSP (*Cryptographic Service Provider*);
- il *Certificate Server*;
- il servizio *smartcard*;
- un canale sicuro;
- il *file system* EFS (*Encrypting File System*);
- il *Microsoft Exchange Server* KM (*Key Management*) *Server*;
- alcune applicazioni che utilizzano l'infrastruttura a chiave pubblica.

Il nuovo sistema operativo di *Microsoft* dispone di un'architettura modulare, che consente agli

amministratori di aggiornare, integrare, estendere e sviluppare con facilità l'infrastruttura PKI dell'azienda, senza modificare i sottostanti *kernel* di sistema operativo. Per esempio, *Exchange Server* 5.5 utilizza unicamente il suo *server* KM per creare e gestire i certificati *client Exchange Server*. Con il *Service Pack* 1 in *Exchange Server* 5.5, quest'ultimo utilizza *Certificate Server* invece di *Exchange Server* KM *Server* per creare e gestire i propri certificati *client*.

Gli sviluppatori possono creare applicazioni compatibili con l'infrastruttura PKI, che siano basate sui componenti PKI e sulla CryptoAPI forniti da *Microsoft*. Per esempio, è possibile impiegare la CryptoAPI e i certificati digitali per crittografare e autenticare i messaggi delle applicazioni MSMQ (*Microsoft Message Queue Server*). È anche possibile utilizzare selettivamente i vari componenti *Microsoft* PKI, in funzione delle effettive necessità aziendali. Per esempio, se l'azienda ha bisogno di disporre di un sito *Web* sicuro, si può utilizzare *Certificate Server* e la funzione a canale sicuro incorporata in *Internet Information Server* e in *Internet Explorer*.

Esaminiamo ora più da vicino alcuni componenti PKI di *Windows* 2000:

- i *provider* CSP;
- i *Certificate Server*, con i suoi *Certificate Manager*, ed il *tool Certificate Server Manager* e le politiche di certificato da esso generate;
- il servizio **smartcard** (di abbiamo disposto apposita sezione);
- la funzione a canale sicuro;
- lo schema **Authenticode**;
- il *file system* EFS.

Provider CSP

Lo schema CSP (*Cryptographic Service Provider*) implementa alcuni algoritmi e chiavi specifiche, in funzione dei requisiti di sicurezza richiesti (per esempio, la generazione di chiavi pubbliche RSA a 1.024 *bit*, la crittografia RC2 e RC4 a 128 *bit*, lo *standard* DES - *Data Encryption Standard* - a 56 *bit*). Le applicazioni possono essere configurate con vari *provider* CSP, a seconda delle esigenze aziendali e delle limitazioni legali nei confronti dell'esportazione al di fuori degli USA degli algoritmi di crittografia. Per esempio, si potrebbe utilizzare il *Microsoft Enhanced Cryptographic Provider* (che supporta le chiavi pubbliche RSA a 1.024 o più *bit*, oltre agli schemi RC2/RC4 a 128 *bit*, DES e 3DES) all'interno del Nord America, e il *Microsoft Base Cryptographic Provider* (che supporta le chiavi pubbliche RSA a 512 *bit* e lo schema RC2/RC4 a 40 *bit*) al di fuori degli USA, in modo da rispettare le leggi sull'esportazione degli algoritmi. Quando è necessario specificare un *provider* CSP all'interno di un'applicazione, è sufficiente scegliere quello desiderato dalla lista dei *provider* CSP installati in NT.

Oltre al *provider* CSP di base e a quello migliorato che sono stati inclusi da *Microsoft* in NT 4.0, *Windows* 2000 ne comprende anche altri, in grado di adattarsi meglio alle varie esigenze di sicurezza - per esempio, gli schemi RSA, DSS (*Digital Signature Standard*), *Diffie-Hellmann* e vari *provider* CSP di base per *smartcard*. Gli sviluppatori di terze parti possono scrivere nuovi *provider* CSP proprietari, che possono essere installati dopo avere ottenuto l'approvazione da parte di *Microsoft* per il loro utilizzo in NT. Per esempio, *Datakey* (una società che offre prodotti *smartcard*) ha reso disponibile un *provider* CSP approvato da *Microsoft* e chiamato *SignaSURE Cryptographic Provider*, che consente di eseguire tutte le principali funzioni di crittografia proprie delle *smartcard*.

Fisicamente, un *provider* CSP è costituito da un *file* .dll. Anche se la maggior parte è basata sul *software*, è tuttavia possibile implementarli anche nell'*hardware*. Per esempio, un produttore potrebbe implementare il suo *provider* CSP proprietario per *smartcard* e l'adattatore per la lettura di queste ultime, in modo da contribuire a migliorare prestazioni ed efficienza del sistema.

Certificate server

Anche *Certificate Server* è disponibile a partire da NT 4.0; ha sempre fornito le funzionalità di base di CA per richiedere, rilasciare, pubblicare e gestire i certificati. *Certificate Server* offriva l'integrazione S/*MIME* (*Secure Multipurpose Internet Mail Extensions*) per *Exchange Server*, ma successivamente *Microsoft* lo ha orientato soprattutto all'autenticazione *client* basata su chiave pubblica per IIS (*Internet Information Server*). Per controllare la configurazione di *Certificate Server*, si richiedeva agli amministratori delle modifiche dirette su alcuni *file* di testo. *Certificate Server* era privo di alcune funzionalità di gestione particolarmente importanti per l'uso *enterprise* dell'infrastruttura PKI (come i *tool* per personalizzare i tipi di certificato e le impostazioni delle politiche) e offriva il supporto unicamente per le gerarchie CA a due livelli (che risultava inadeguato per l'implementazione PKI su larga scala).

In *Windows* 2000, il nome *Certificate Server* è stato leggermente modificato in *Certificate Services*. Questi ultimi sono più potenti e meglio integrati con tutto il resto del sistema operativo. Gli *snap-in* della *consolle* MMC (*Microsoft Management Console*) offrono dei *tool* **GUI** per il lato *client* e per quello *server*. Anche se i *Certificate Services* possono mantenere un proprio *data store* indipendente, per ottenere la completa funzionalità *enterprise* usano *Active Directory* per archiviare e pubblicare i certificati. In questo modo si possono mappare con facilità i certificati sugli utenti e sfruttare le funzionalità di gestione dell'*editor* GPE (*group Policy Editor*) per controllare a chi, per conto di chi e a quale scopo i *Certificate Services* rilasciano i certificati. Infine, i *Certificate Services* ora supportano le gerarchie multilivello.

I *Certificate Services* sono diventati molto più leggeri in *Windows* 2000, dal momento che alcune porzioni del loro predecessore per NT 4.0 sono passate ad altri componenti. Per esempio, *Microsoft* ha aggiunto alla CryptoAPI le funzioni per la gestione dei certificati; ha inoltre spostato in *Active Directory* il *data store* dei certificati di *default*. Dal momento che i *Certificate Services* accedono al proprio *store* dei certificati attraverso la CryptoAPI, possono pubblicare i certificati in altre *directory* di terze parti.

Canale sicuro

Windows 2000 incorpora un canale sicuro che supporta il protocollo SSL, il quale viene utilizzato da molte aziende per rendere sicure le comunicazioni *Web* su *Internet*. L'ente IETF (*Internet Engineering Task Force*) ha ratificato lo schema SSL quale *standard Internet*, assegnandogli il nuovo nome di TLS (*Transport Layer Security*).

Il canale sicuro messo disposizione da *Microsoft* comprende anche la crittografia SGC (*Server Gated Cryptography*): si tratta di un'estensione dello schema SSL 3.0, che utilizza la codifica a 128 *bit* per rendere sicure le sessioni bancarie *on-line*. *Internet Explorer* 3.0 e le versioni precedenti, oltre a *Internet Information Server* 3.0, supportano lo schema SSL/TLS; *Internet Explorer* 4.0, *Internet Information Server* 4.0 e le versioni successive, oltre a *Money* 98, supportano invece lo schema SGC.

Rispetto a quest'ultimo, lo schema SSL/TLS è costituito da un protocollo generico, che può essere utilizzato per qualsiasi sito *Web*. A causa delle limitazioni insite nelle esportazioni al di fuori degli USA degli algoritmi di crittografia più potenti, lo schema SSL/TLS implementa due versioni di *Internet Explorer*: la prima dispone della crittografia a 128 *bit* per il Nord America, mentre la seconda utilizza la codifica unicamente a 40 *bit* per l'uso internazionale. Le banche e le istituzioni finanziarie hanno tuttavia bisogno di utilizzare una crittografia particolarmente sofisticata per proteggere i clienti che utilizzano l'accesso *on-line* tramite il *Web*; per questo motivo, il governo degli Stati Uniti consente alle organizzazioni finanziarie internazionali di utilizzare la crittografia SGC tra gli USA e praticamente qualsiasi altro paese del mondo. *Microsoft* ha incorporato questo protocollo nelle versioni 4.0 e 5.0 del *browser Internet Explorer*, a 40 e 128 *bit*. Il *browser* può utilizzare la funzione SGC per la crittografia a 128 *bit* soltanto se il *server Web* con cui è in comunicazione è a sua volta dotato di un certificato *server* SGC. Ricordiamo l'interfaccia SSPI (*Security Support Provider Interface*) e la CryptoAPI. Le **API** di *Windows* 2000 permettono agli sviluppatori di riutilizzare i servizi (per esempio quelli crittografici) forniti dal sistema operativo. Il fatto di essere in grado di astrarre le applicazioni dal *provider* permette inoltre di proteggerle dall'obsolescenza. È possibile aggiornare e migliorare i componenti *provider* di pari passo con l'evoluzione della tecnologia, senza per questo influenzare l'applicazione. Per esempio, un'applicazione che usa il *provider* CSP può sfruttare rapidamente l'algoritmo di crittografia di un

nuovo *standard*.

Esaminiamo ora il modo in cui gli schemi SSL/TLS e SGC utilizzano i certificati e le chiavi pubbliche al fine di instaurare un canale sicuro tra *client* e *server*. Nello *standard* SSL/TLS, il *client* invia per prima cosa un messaggio di *hallo* al *server*, mentre quest'ultimo saluta il *client* inviandogli il proprio certificato *server*. Il *client* autentica il *server* utilizzando la chiave pubblica dell'autorità CA (che viene estratta dal certificato CA archiviato nel *client*), per controllare la firma di quest'ultima nel certificato del *server*. Il *server* può a sua volta autenticare opzionalmente il *client*, chiedendogli e verificando il suo certificato (in *Internet Information Server* 3.0 e nelle versioni successive, gli amministratori possono fare sì che il *server* autentichi il *client*). Quando il *client* ha autenticato il *server*, genera una chiave di sessione a 40 o 128 *bit*, a seconda delle limitazioni per la sicurezza. Il *client* crittografa quindi la chiave di sessione usando la chiave pubblica del *server* (estratta dal certificato di quest'ultimo), quindi gliela invia. Il *server* la decodifica quindi utilizzando la propria chiave privata. A questo punto, il *server* e il *client* possono scambiarsi i dati crittografati con questa chiave di sessione.

Con lo schema SSL/TLS, è possibile usare i certificati rilasciati da *Certificate Server* di *Windows* 2000 oppure da altre autorità CA. Per utilizzare lo *standard* SGC, è invece necessario richiedere un certificato *server* SGC a un'autorità CA autorizzata (per esempio a *VeriSign*), la quale esaminerà i requisiti del richiedente. Il protocollo SGC utilizza una sequenza di *handshake* simile a quella adottata dallo schema SSL/TLS per impostare una sessione sicura, ma che differisce tuttavia da quest'ultima sotto due aspetti. Per prima cosa, con lo *standard* SGC il *client* resetta e fa ripartire la sequenza di *handshake* dopo il primo scambio di *hello*, quando rileva che il certificato del *server* è di tipo SGC. Secondariamente, il *client* genera sempre una chiave di sessione a 128 *bit* dopo aver resettato la sequenza di *handshake* e l'autenticazione del *server*.

Authenticode

L'infrastruttura PKI di *Microsoft* mette a disposizione una tecnologia chiamata *Authenticode* per la firma del codice, che assicura l'integrità e l'origine del *software* commerciale e di quello gratuito distribuito su *Internet*. Lo schema *Authenticode* è basato sulla tecnologia a firma digitale; aggiunge una firma digitale, un certificato di firma del codice e un contrassegno dell'ora e della data al codice che caratterizza il *software* come quello di *applet Java*, controlli *ActiveX*, *file* .dll, *file* eseguibili, *file cabinet* e *file catalog*. Questo schema consente anche di verificare il codice scaricato prima di eseguirlo sul proprio sistema. Per firmare e verificare il codice, il sistema *Authenticode* utilizza due tecniche differenti: la firma del codice e la sua verifica.

Prima di poter firmare il codice per garantirne l'integrità e l'origine, è necessario ottenere da un'autorità CA un certificato per la firma del codice o per la pubblicazione del *software*. Dopo questa operazione, per firmare il codice si potranno usare le funzioni di firma *Authenticode* offerte dal kit SDK *ActiveX*. Il codice che si desidera autenticare viene passato attraverso un algoritmo di *hashing*, mentre la propria chiave privata viene usata per firmare l'*hash* ottenendo una firma digitale. A questo punto viene creato il blocco della firma, che contiene la firma digitale e il certificato per la firma del codice. Lo schema *Authenticode* consente di contrassegnare il blocco della firma con l'ora e la data, in base ai dati che vengono forniti da un *provider* di servizi di data e ora, come per esempio *VeriSign*. Infine, il blocco della firma viene legato al *software* originale; a questo punto è possibile pubblicare sul proprio sito *Web* il *software* firmato, rendendolo disponibile per il *download*.

Il kit SDK *ActiveX* mette disposizione anche una funzione dedicata alla verifica del codice, che consente di verificare il *software* scaricato prima di eseguirlo. *Microsoft* ha incorporato questa funzione in *Internet Explorer* 3.0 e nelle versioni successive. Prima che *Internet Explorer* esegua il codice firmato, viene chiamata questa funzione per la verifica del codice che controlla la firma, il certificato dell'editore e la data e l'ora dell'autenticazione. Dopo questa verifica, la funzione

visualizza il nome del codice, quello dell'organizzazione o della persona che lo ha pubblicato, la data in cui l'editore lo ha autenticato e il nome dell'autorità CA che ha fornito il certificato per la firma del codice. L'utente a questo punto può decidere di accordare la fiducia all'editore.

Usando *Internet Explorer*, è possibile impostare una politica di sicurezza per *Authenticode*, caratterizzata da quattro diversi livelli di protezione: alto, medio, basso e personalizzato. Il livello più elevato fa sì che non venga eseguito il codice danneggiato; quello medio avverte l'utente prima di eseguire del codice potenzialmente danneggiato; il livello basso esegue in ogni caso qualsiasi tipo di codice, mentre quello personalizzato consente di scegliere le impostazioni di sicurezza: per esempio attivare il codice *ActiveX*, oppure disattivare le *applet Java*. È anche possibile definire diversi livelli relativi a zone di sicurezza differenti: per esempio *Internet*, intranet, siti affidabili e siti ad accesso limitato di *Internet Explorer*.