Consiglio Nazionale delle Ricerche

Cambiamento dinamico della PASSWORD

G. Cresci - D. Lari

123

CNUCE

Divisione Servizio Elaborazione Dati

A cura di : Guglielmo Cresci - Diana Lari

Istituto del Consiglio Nazionale delle Ricerche

INDICE

INTRODUZIONE

Il lavoro descritto in questa nota e' rivolto a migliorare la protezione delle informazioni di proprieta' degli utenti del sistema VM/370.
Come e' noto questo sistema garantisce una protezione completa delle informazioni di ciascun utente contro interferenze o errori degli altri utenti e dello stesso sistema VM.
Tale sicurezza, che e' uno dei vantaggi piu' evidenti e caratteristici dei sistemi a macchine virtuali, puo' venire meno se l'utente non tutela adeguatamente la segretezza delle proprie informazioni sia perche' non lo ritiene opportuno che per negligenza o disattenzione.
Per accedere al VM/370 e' necessario essere proprietari di una macchina virtuale definita all'interno del sistema stesso; in pratica e' sufficiente conoscere nome e password (parola chiave) di una macchina per disporre, piu' o meno semplicemente, di tutte le informazioni contenute in quella macchina. Ancora, mentre i nomi delle macchine virtuali sono di pubblico dominio perche' utilizzati dal sistema, per esempio per distinguere files di output etc., le rispettive passwords sono "gelosamente" conservate dal VM in modo inaccessibile a qualunque utente: solo all'amministratore del sistema e' consentito accedere a queste informazioni.
Alla cura che il VM pone nella conservazione e nell'uso delle passwords delle macchine virtuali spesso non corrisponde un uguale impegno a tutelare la "privacy" del proprio lavoro da parte dell'utente che, per fretta, disattenzione o scarsa competenza, puo' lasciare la password della sua macchina virtuale a disposizione della curiosita' di altri.
Dalla nostra personale esperienza ci risultano molti casi in cui la perdita totale o parziale di protezione delle informazioni va ricercata nella disattenzione di un utente (e nella malizia di un altro); non ci risultano casi in cui questo inconveniente sia causato da deficienze del sistema operativo.
Situazioni di questo genere sono difficili da eliminare perche' presuppongono una collaborazione da parte dell'utenza che non sempre e' possibile trovare, anche a causa della sua scarsa sensibilita' a questa problematica.
Trovandoci nella impossibilita' pratica di evitare l'inconveniente ci siamo posti il problema di come sia possibile operare per limitarne i danni al minimo possibile.
L'unico valido provvedimento ci e' sembrato quello di cambiare le passwords delle macchine virtuali con una certa frequenza in modo da prevenire i furti di informazioni e limitare nel tempo quelli gia' perpetrati.
Il cambio di passwords e' una procedura gia' in uso da tempo

anche per motivi diversi da quelli finora discussi; esistono infatti macchine virtuali su cui lavorano, in tempi diversi, persone diverse che pure appartenendo allo stesso ente desiderano tutelare la "privacy" del loro lavoro e soprattutto avere la certezza di essere i soli che utilizzano la macchina virtuale per sapere, a lavoro ultimato, quale ne sia stato il costo effettivo.
Visto che la soluzione ai problemi prospettati e' sempre esistita viene da chiedersi come mai finora sia stata usata tanto poco. La risposta piu' plausibile e' che la procedura necessaria a sostituire la password di una macchina virtuale ha un iter burocratico che non giova alla sua diffusione.
Raramente l'utente e' disposto a sottostare a questo iter se non e' stimolato da motivi gravi per cui in pratica finora hanno fatto ricorso a questo provvedimento solo coloro che hanno subito in prima persona le conseguenze della propria disattenzione e dell'altrui malizia.
Cambiare la password di una macchina equivale in termini di tempo e lavoro a richiedere un cambiamento di configurazione; e' infatti necessario:

1) Compilare un modulo di cambiamento di configurazione della macchina virtuale specificandovi la nuova password.

2) Consegnarlo allo Sportello Utenti.

3) Attendere che la richiesta venga evasa.

Tutto il processo, con l'attuale organizzazione, puo' durare fino ad una settimana. Infatti questa procedura impegna anche l'amministratore del sistema che deve modificare il "directory" delle macchine virtuali per far riconoscere al VM la nuova password.
Concludendo possiamo sintetizzare in tre punti i difetti di questa organizzazione:

1) Il processo e' troppo lento. Se c'e' il dubbio, o peggio ancora, la certezza di una appropriazione indebita, il provvedimento-tampone dovrebbe essere il piu' rapido possibile. D'altra parte non e' possibile istituzionalizzare una procedura piu' rapida per i cambiamenti di passwords perche' questi richiedono l'aggiornamento del "directory": un lavoro lungo e laborioso che viene effettuato settimanalmente.

2) La nuova password passa per diverse mani gia' prima di diventare operativa. Questo non giova affatto alla sua segretezza.

3) L'attuale procedura richiede un certo dispendio di tempo da parte del personale del Centro di Calcolo, in particolare dell'amministratore del sistema.

Questi inconvenienti vengono superati se si fornisce all'utenza un metodo automatico di cambiamento di password che consenta di eliminare tutti gli intermediari e di rendere possibile un dialogo diretto utente - VM che garantisca rapidita' e segretezza.
Fornendo a ciascuno la possibilita' di cambiare la password della propria macchina con una procedura estremamente semplice (un comando) si raggiunge il duplice risultato:

- di abituare a questa operazione anche coloro che, per pigrizia, non avrebbero mai adoperato la precedente procedura burocratica.

- di investire l'utente della responsabilita' della sicurezza delle proprie informazioni. Infatti, mentre l'organizzazione precedente richiedeva l'intervento e quindi co-responsabilizzava anche il personale del Centro di Calcolo, ora l'utente rimane il solo abilitato a gestire la sicurezza del proprio lavoro.
  A nostro avviso, questo fattore puo' avere effetti largamente positivi perche' contribuisce a sensibilizzare l'utenza al problema della sicurezza delle informazioni ed e' nostra convinzione che un'utenza consapevole sia il presupposto chiave per poter ottenere risultati positivi in questo delicato settore.
  Senza queste premesse diventano inutili gli interventi operati sul sistema.

L'ORGANIZZAZIONE STANDARD

La possibilita' di cambiare la password della macchina virtuale con un comando era gia' stata implementata sul CP-67, il vecchio sistema conversazionale del CNUCE; anche per questo motivo abbiamo voluto ripristinare questa facility sul VM mantenendole una forma il piu' possibile simile a quella che essa aveva sul vecchio sistema.
Le similitudini tra le due implementazioni si limitano pero' al solo aspetto formale poiche' i due sistemi adottano criteri di gestione del directory delle macchine virtuali completamente diversi che hanno richiesto un lavoro completamente nuovo. Prima di accennare ai criteri seguiti nell'implementazione di questa facility e' utile dare alcuni cenni sulla modalita' con cui il VM gestisce il directory delle sue macchine virtuali.
L'amministratore del sistema dispone di un file CMS dove sono descritte tutte le macchine virtuali che possono accedere al VM. Tutte le volte che e' necessario modificare la configurazione di una macchina o aggiungere macchine nuove e' necessario aggiornare questo file. Esistono poi delle routines particolari che consentono di trascrivere queste informazioni su un'area disco preventivamente allocata in un formato direttamente accessibile al CP-VM: a quest'area fa riferimento il sistema operativo ogni qualvolta necessita di informazioni di directory.
Queste eventualita' sono molto frequenti poiche' gli accessi al directory sono richiesti da molti comandi di VM e non solo dal LOGON, per cui quest'area disco deve essere sempre accessibile.
Quando l'amministratore del sistema deve aggiornare il directory, e' necessario bloccare tutti gli accessi da parte del sistema operativo finche' in quell'area disco non sia stato riscritto il nuovo directory. Questa operazione richiede sempre alcuni minuti: per tutto questo tempo tutte le funzioni che necessitano di accessi al directory restano bloccate.
Situazioni di questo genere, ammesse dal CP-67, furono giudicate inaccettabili per il VM che fu quindi implementato in modo da evitare l'inconveniente.
Esso infatti dispone di due aree-disco per memorizzare le informazioni di directory: di queste una sola e' referenziata (copia attiva). Quando l'amministratore del sistema riscrive su disco un nuovo directory, questo va ad occupare l'area attualmente non attiva per cui la riscrittura della nuova copia procede senza bloccare gli accessi, che continuano a referenziare la copia attiva. Solo a riscrittura ultimata viene dichiarata attiva la copia appena creata e disabilitata l'altra.
Questa implementazione piu' sofisticata ed efficiente di

quella del CP-67 crea piu' notevoli difficolta' quando si voglia affrontare il problema di modificare il directory in modo autonomo, cioe' senza passare attraverso il file di CMS di cui dispone l'amministratore del sistema.
Infatti non e' piu' possibile modificare i campi di nostro interesse direttamente su disco perche' ogni modifica apportata dall'amministratore del sistema crea un nuovo directory che non si sovrappone all'attuale, ma occupa un'area nuova e quindi ad ogni modifica di directory andrebbero perdute tutte le modifiche effettuate dal sistema operativo in modo "autonomo".
Non e' neppure possibile trasferire sulla copia nuova le modifiche "autonome" apportate alla copia vecchia senza incorrere in gravi difficolta' per cui e' necessario ricorrere ad un tipo di organizzazione che non preveda modifiche al directory.

CRITERI DI IMPLEMENTAZIONE

Il primo criterio seguito nell'implementazione della facility e' stato quello di non modificare la gestione del directory.
Questa scelta, che ha condizionato molte scelte successive, ci impone:

a) di utilizzare una nuova area-disco (distinta da quella del directory) in cui conservare le passwords modificate a comando dalle macchine virtuali;

b) di scrivere ex-novo le routines di gestione di questa area;

c) di integrare la gestione del buffer delle nuove passwords coi riferimenti al directory eseguiti dal CP-VM.

Altro criterio a nostro avviso determinante nell'economia dell'implementazione, e' stato quello di evitare, per quanto possibile, l'interferenza del codice scritto localmente con i moduli standards del sistema operativo.
Questa scelta non e' prerogativa della sola implementazione a cui questa nota fa riferimento: essa e' una costante sempre presente quando si affrontano implementazioni locali ed e' giustificata dalla necessita' di ridurre al minimo le interferenze tra software locale ed aggiornamenti che la casa produttrice rilascia mensilmente.
In base alla nostra esperienza, possiamo affermare che e' sempre conveniente ricorrere ad implementazioni il piu' possibile svincolate dal software standard perche', anche se all'atto dell'installazione richiedono un impegno notevole (superiore a quello richiesto da implementazioni strettamente integrate nel sistema), consentono poi notevolissimi risparmi nella fase di manutenzione ordinaria (cioe' nell'inserimento degli aggiornamenti).
Nel caso specifico di questa implementazione i due criteri enunciati si adattano perfettamente l'uno all'altro infatti, impostando una gestione del tutto svincolata dal directory, ci e' anche semplificato il compito di raccogliere la quasi totalita' delle modifiche da apportare al CP-VM in un modulo locale indipendente dal sistema standard.
Le uniche interferenze sono pertanto limitate alla richiesta di intervento del nuovo modulo che il sistema deve eseguire.
La scelta di orientarci verso una implementazione svincolata dal sistema operativo ci lascia la massima liberta' nella strutturazione della modifica. Abbiamo tuttavia cercato di implementarla secondo i criteri logici tipici dell'ambiente in cui essa sarebbe stata inserita.
In ossequio a questo principio abbiamo strutturato la

gestione delle nuove passwords in maniera analoga a quella del directory. Infatti:

1) Manteniamo costantemente aggiornate le nuove passwords sia in memoria centrale che su un'area del disco-sistema definita a livello di generazione.

2) Considerata la frequenza dei riferimenti che il buffer delle passwords subisce, sia il buffer stesso che il modulo di gestione sono stati sistemati in memoria paginabile.
Questo accorgimento consente di risparmiare spazio in memoria centrale poiche' le aree meno utilizzate vengono trasferite su devices di paginazione, richiamate solo quando diventano indispensabili e solo per il tempo strettamente necessario.

3) Infine, quale ulteriore garanzia di sicurezza, i blocchi che descrivono le nuove passwords sono "mascherati", cioe' nome e password della macchina virtuale non sono memorizzate direttamente in formato-carattere, ma sono codificate dal modulo di gestione.
In questo modo si evita il pericolo che l'operatore di sistema o qualche altro utente privilegiato (con la classe "operatore di sistema"), esaminando il contenuto di zone di memoria centrale paginabile, possa venire a conoscenza delle informazioni "riservate" relative alle passwords delle macchine virtuali.
Questa possibilita' e' decisamente remota anche perche', il buffer e' allocato in memoria paginabile quindi non sempre e' presente in memoria e soprattutto la sua posizione (l'indirizzo reale) non e' fissa ma e' continuamente variata dalle routines di paginazione.

Gli accorgimenti ora elencati sono stati adottati in conformita' alle modalita' di gestione del directory che e' mantenuto in due copie "mascherate": una su disco, una in memoria paginabile.

Questi sono gli unici vincoli a cui abbiamo assoggettato l'implementazione che, in virtu' della sua indipendenza dall'ambinte in cui lavora, ha poi potuto essere sviluppata con criteri autonomi.

Il problema fondamentale che abbiamo dovuto affrontare e' stato quello del tipo di gestione secondo cui organizzare il buffer delle nuove passwords. La soluzione ottimale a questo problema deve garantirci:

1) La massima velocita' di accesso al buffer. Gli accessi al buffer sono essenzialmente di due tipi: per semplice consultazione (in lettura) oppure per aggiornamento del contenuto (in scrittura). E' importante osservare che i primi sono molto piu' frequenti poiche' sono invocati da tutti i processi di login delle macchine virtuali, mentre i secondi avvengono solo ad ogni cambiamento di password ed e' verosimile presumere che l'utente medio non cambi la password della propria macchina ad ogni sessione.
L'aspetto piu' interessante di questo punto e' pertanto la velocita' d'accesso in consultazione.

2) Dimensioni del buffer contenute nei limiti piu' ristretti possibili per evitare di occupare memoria altrimenti utilizzabile.

3) Massima semplicita' di implementazione subordinata pero' al raggiungimento dei due precedenti obiettivi.

Tenendo presenti queste specifiche abbiamo preso in considerazione tre possibili organizzazioni che descriviamo brevemente qui di seguito.

1) Buffer organizzato sequenzialmente.
Per ogni macchina virtuale che si modifica la password a comando, viene creato un blocco contenente il nome della macchina virtuale e la nuova password. Questo blocco va ad occupare la prima posizione libera nel buffer.
Quando una macchina virtuale torna alla sua password di directory, l'entry corrispondente del buffer viene dichiarata libera (per es. azzerandola).
A proposito della liberazione di un'entry preventivamente occupata si prospettano subito due alternative:

   a) Ripulire l'entry e lasciare che sia la prossima operazione di inserimento a riempirla.

b) Compattare subito il buffer in modo che le operazioni di inserimento siano semplificate al massimo.
Se il buffer non e' compattato, tutte le volte che si vuole inserire una nuova entry e' necessario scorrerlo tutto per trovare la prima azzerata; se il buffer e' compatto basta inserire direttamente dopo l'ultima entry occupata senza bisogno di analizzare tutto il buffer entry per entry.
A favore della prima soluzione gioca la semplicita' di implementazione, ma a suo sfavore pesano la ricerca meno spedita della prima entry libera e le maggiori dimensioni del buffer (dovute ai "buchi" disseminati tra le entries occupate).
A sfavore della seconda soluzione bisogna considerare le difficolta' di implementazione e l'aggravio di lavoro richiesto dal compattamento.
Tuttavia e' nostro convincimento che questi aspetti negativi siano ampiamente compensati dai vantaggi che questo tipo di soluzione consente di ottenere.

Finora abbiamo esaminato i casi che prevedono aggiornamenti del buffer, cioe' modifiche delle passwords; dobbiamo ancora considerare i casi di semplice consultazione che sono poi i piu' frequenti.
E' immediato riconoscere che l'organizzazione sequenziale richiede la scansione del buffer entry per entry fino a che non si trova il blocco relativo alla macchina cercata.
Se la macchina cercata non esiste, il buffer viene scandito fino alla prima entry libera.
Se tutte le macchine del directory fossero presenti nel buffer, il numero medio di ricerche necessarie a trovare un'entry specifica sarebbe pari a N/2, essendo N il numero di macchine virtuali definite.
In realta' nel buffer e' presente soltanto una parte delle macchine del directory e quindi il numero medio di tentativi aumenta notevolmente perche', per ogni macchina che non figura nel buffer e' necessaria la scansione di tutto il buffer.
Concludendo possiamo sicuramente affermare che l'organizzazione sequenziale si presta bene alle modifiche (inserimenti o cancellazioni di entries), ma richiede tempi piuttosto lunghi per la ricerca di un'entry specifica.
Questo pesa soprattutto nelle operazioni di consultazione a cui il buffer e' soggetto durante i processi di login delle macchine virtuali.

Come abbiamo gia' accennato questi processi sono molto piu' frequenti dei comandi di cambiamento di password per cui e' conveniente organizzare il buffer in modo da rendere le consultazioni piu' spedite.
Per questo motivo, e nonostante la sua semplicita', abbiamo preferito prendere in considerazione tipi di organizzazione piu' complessi ma in grado di effettuare ricerche piu' veloci.

2) Ricerca di tipo statistico.
Abbiamo considerato la possibilita' di organizzare il buffer secondo un metodo di ricerca a chiavi. Le chiavi possono essere estratte dal nome della macchina con semplici algoritmi, che tuttavia non garantiscono la biunivocita' della relazione chiave-macchina virtuale.
Questo crea qualche problema, peraltro di non difficile soluzione, sia nei casi di inserimento o cancellazione di un'entry che nei casi di consultazione.
Ogni nuova entry viene aggiunta nel buffer nella posizione specificata dalla chiave estratta dal nome della macchina a cui quell'entry si riferisce. Poiche' non c'e' biunivocita' nel rapporto chiave-nome della macchina virtuale, non e' detto che quell'entry sia libera; in questi casi l'inserimento avviene nella prima entry libera successiva a quella selezionata con la chiave.
Per quanto riguarda la ricerca di un'entry il processo seleziona inizialmente l'entry corrispondente alla chiave, poi esegue un'ulteriore controllo sul nome della macchina. Se questo non e' quello voluto inizia una ricerca sequenziale che si arresta quando si trova la macchina desiderata oppure alla prima entry libera. Questa eventualita' si verifica quando la macchina cercata non e' presente nel buffer.
Infine la cancellazione di un'entry comporta una ricerca del tipo di quella appena descritta e non richiede compattamento.
E' immediato riconoscere che il numero medio di accessi al buffer per selezionare un'entry specifica e' notevolmente inferiore a quello richiesto dalla ricerca sequenziale; questo miglioramento si paga con un'organizzazione piu' complessa e soprattutto con l'impiego di un buffer di dimensioni maggiori. Infatti maggiori sono le dimensioni del buffer, piu' alta e' la probabilita' di trovare una macchina nella posizione specificata dalla chiave e quindi minori risultano essere i tempi di accesso al buffer. Si puo' pertanto concludere che la ricerca a chiavi richiede un buffer molto grande per dare ricerche veloci.
Non solo, questo tipo di ricerca fornisce i risultati

migliori quando la maggior parte delle macchine definite in directory e' presente nel buffer; infatti, per affermare che l'entry cercata non esiste, e' necessario operare una scansione sequenziale a partire dall'entry selezionata con la chiave fino alla prima entry libera del buffer. Questo richiede un notevole numero di accessi che rallentano notevolmente le ricerche. Alla data odierna (sono passati alcuni mesi dall'installazione della modifica) le macchine virtuali presenti nel buffer sono circa 1/3 di quelle definite in directory per cui le richieste "a vuoto" sono piuttosto frequenti.
Una situazione simile fu prevista al momento dell'implementazione; per questo motivo e poiche' furono considerate eccessive le dimensioni del buffer che questo tipo di organizzazione richiede per garantire ridotti tempi di accesso, si e' preferito orientarci verso un'altra soluzione.

3) L'organizzazione a blocchi concatenati.
E' la soluzione che, a nostro avviso, concilia nel modo migliore la velocita' di ricerca con le dimensioni del buffer, anche se, probabilmente, e' quella che richiede l'implementazione piu' complessa.
Per ridurre i tempi di accesso, si sono organizzate le macchine virtuali presenti nel buffer in tante catene quante sono le lettere dell'alfabeto. Tutte le macchine il cui nome inizia con la stessa lettera alfabetica sono concatenate tra loro; ogni catena ha un indirizzo-ancora tenuto fuori dal buffer (vedi fig. 1).

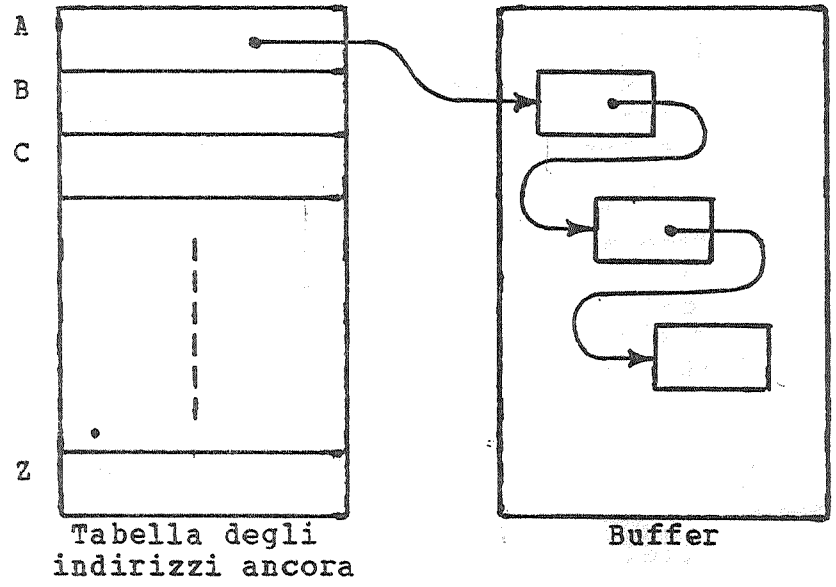

FIG. 1: Organizzazione a blocchi concatenati

Tutti gli indirizzi-ancora sono raccolti in una tabella (DISPTAB) avente tante entries quante sono le lettere dell'alfabeto. Le dimensioni di questa tabella sono ridotte al minimo (mezza voce per entry) associando ad ogni lettera non un indirizzo ma un displacement a partire dall'indirizzo di caricamento del buffer. In questo modo la selezione della prima entry della catena avviene sommando l'indirizzo base del buffer al displacement contenuto nella DISPTAB (vedi fig. 2).
Il processo di ricerca di una macchina avviene pertanto secondo le seguenti fasi:

1) Selezione dell'entry di DISPTAB corrispondente alla lettera alfabetica con cui inizia il nome della macchina da ricercare.

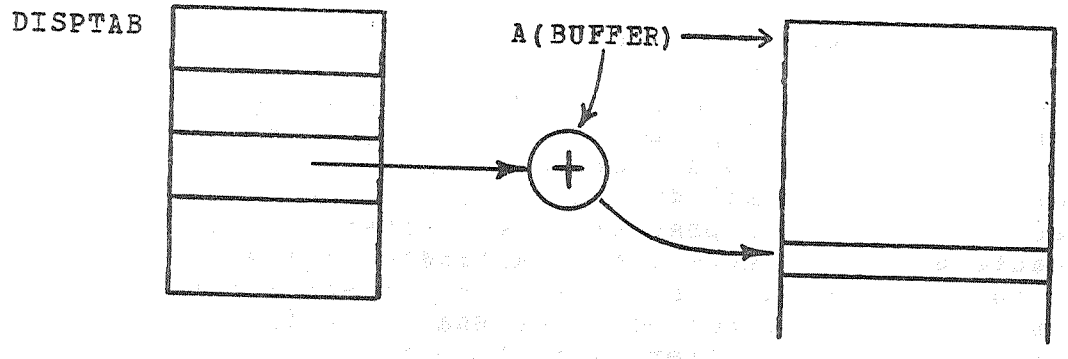

FIG. 2: Meccanismo di selezione della prima entry di una catena

2) Selezione della prima entry della catena come da fig. 2.
3) Scansione della catena fino al blocco che descrive la macchina cercata, ovvero fino all'ultimo blocco concatenato se la macchina non e' presente nel buffer.

Si osservi che la scansione e' limitata ad un numero ridotto di blocchi per cui il processo di consultazione risulta essere sufficientemente veloce, almeno quanto quello realizzabile con ricerca a chiavi.
Per semplificare al massimo anche il processo di aggiunta di un'entry, il buffer viene mantenuto sempre compatto, cosicche' e' possibile disporre sempre dell'indirizzo della prima entry libera.
Il processo di aggiornamento pertanto si puo' sintetizzare in tre fasi:

1) Selezione ed aggiornamento della prima entry libera (FRENTRY);

2) Riempimento del blocco con nome e password della macchina inserita;

3) Concatenamento del blocco alla rispettiva catena.

La fig. 3 schematizza le fasi di questo processo.

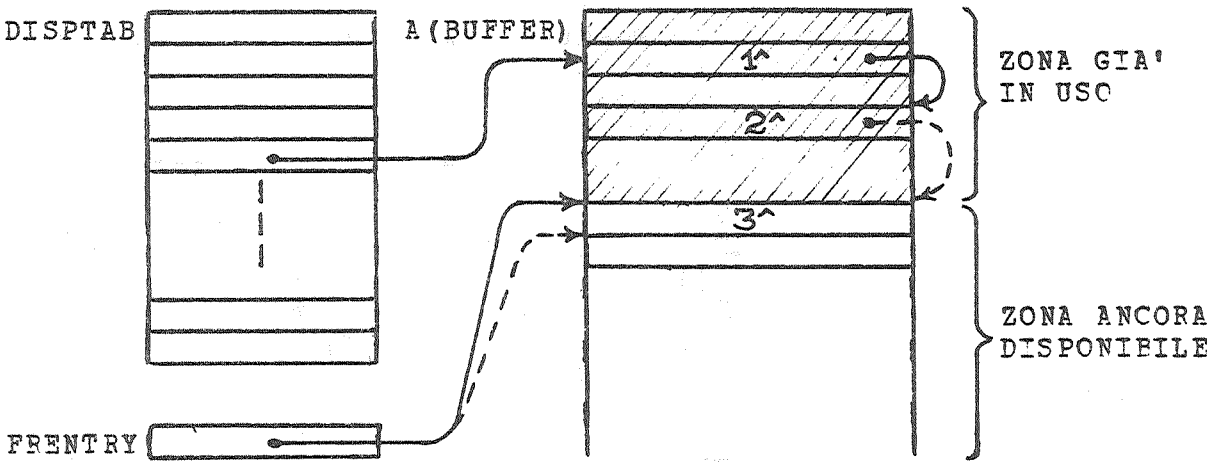

Fig. 3: Inserimento di una macchina:
——→ Situazione iniziale dei puntatori
- - - - - - → Situazione finale dei puntatori.

La fase di concatenamento puo' assumere aspetti diversi, infatti se la macchina aggiunta appartiene ad una catena che comprende gia' altri blocchi, l'aggiustamento del puntatore avviene tra l'ultimo blocco della catena e quello da inserire (vedi fig.3). Viceversa se la macchina appartiene ad una catena ancora vuota l'aggiustamento del puntatore e' effettuato nell'entry della DISPTAB riservata a quella catena.
La descrizione del processo di inserimento ne evidenzia la semplicita': l'aggiunta di una nuova macchina non richiede operazioni complesse ne' lunghe ricerche essendo immediata la selezione dell'entry libera.
Meno semplice e' il processo di cancellazione di un'entry preventivamente occupata; infatti esso richiede:

1) Aggiornamento dei puntatori della catena alterata
2) Compattamento del buffer

L'operazione di compattamento che e' quella piu' pesante, e' effettuata in 3 fasi:

2a) Spostamento dell'ultima entry occupata del buffer in quella appena liberata.

2b) Aggiustamento dei puntatori nella catena a cui appartiene l'entry spostata.

2c) Aggiornamento dell'indirizzo della prima entry libera (FRENTRY).

La fig. 4 schematizza le due fasi del processo di cancellazione di un'entry.

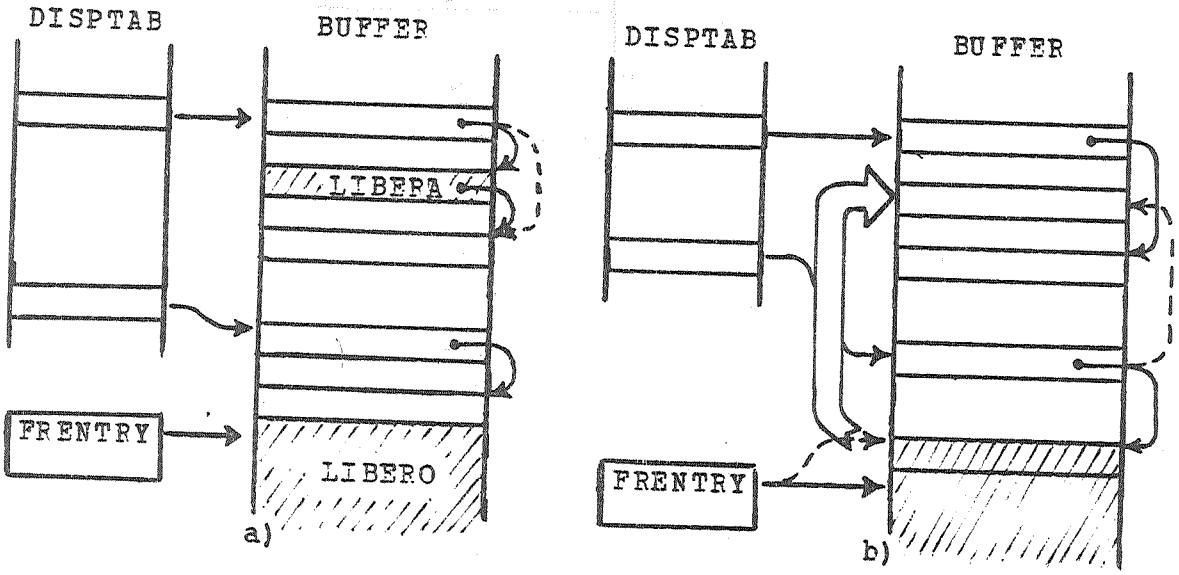

Fig. 4: Processo di cancellazione di un'entry:
a) Aggiornamenti puntatori
b) Compattamento del buffer

Anche in questo processo possono verificarsi casi particolari che necessitano di un trattamento leggermente diverso da quello illustrato; per esempio se l'entry liberata e' l'ultima del buffer o la prima di una catena.

L'analisi svolta finora sulle caratteristiche delle tre possibili implementazioni e' stata riassunta nella tabella di fig.5 che vuole confrontare le prestazioni delle singole organizzazioni considerate.

	TEMPO ACCESSO PER AGGIUNTA	TEMPO ACCESSO PER CANCELLAZIONE	TEMPO DI ACCESSO PER CONSULTAZIONE	DIMENSIONI DEL BUFFER	DIFFICOLTA' DI IMPLEMENTAZIONE	COMPATTAMENTO DEL BUFFER
ORG. SEQUENZIALE SENZA COMPATTAMENTO	LUNGO	MOLTO LUNGO	MOLTO LUNGO	MINIME	MOLTO SEMPLICE	NO
ORG. SEQUENZIALE CON COMPATTAMENTO	IMMEDIATO	LUNGO	LUNGO	MINIME	SEMPLICE	SI
ORGANIZZAZIONE A CHIAVI	BREVE	BREVE	MEDIO-BREVE	MOLTO GRANDI	MEDIA	NO
ORGANIZZAZIONE A BLOCCHI CONCATENATI	IMMEDIATO	MEDIO	BREVE	MINIME	COMPLESSA	SI

Fig. 5: Confronto di prestazioni tra le possibili implementazioni

La decisione di utilizzare l'organizzazione a blocchi concatenati deriva essenzialmente dal fatto che questa offre il miglior compromesso tra tempi d'accesso per consultazione e dimensioni del buffer.
Questi due parametri sono senza dubbio i piu' importanti tra quelli elencati in fig.5 ed i vantaggi ottenuti in questi settori compensano sicuramente le difficolta' d'implementazione che la soluzione adottata comporta.

Un altro grosso problema indipendente dal tipo di organizzazione prescelta e' quello di ridurre al minimo l'overhead di sistema connesso alla gestione della nuova facility.
L'allocazione del buffer e delle routines di gestione in memoria paginabile da gia' una soluzione soddisfacente poiche' evita di dover leggere da disco tutte le volte che e' necessario consultare il buffer.
Il ricorso ad accessi al disco-sistema dove e' conservata una delle due copie del buffer e' invece necessario quando si aggiorna la situazione delle passwords. E' stata nostra cura cercare di ridurre il piu' possibile l'overhead che simili operazioni comportano evitando di riscrivere tutto il buffer e scrivendo invece le sole pagine che hanno subito modifiche.
Questo ha richiesto un ulteriore complicazione delle routines di gestione che, ad ogni aggiornamento, devono selezionare e riscrivere le sole pagine modificate. In particolare abbiamo dovuto dimensionare il blocco che descrive un cambio di password (PASSBLOK) in modo che una pagina contenga un numero intero di blocchi cioe' non vi siano blocchi a cavallo tra due pagine. Questo fatto avrebbe complicato notevolmente le routines di gestione perche' pagine logicamente adiacenti del buffer non sono contigue in memoria (vedi fig.6) e quindi blocchi a cavallo tra due pagine sono spezzati in due aree diverse che devono essere contemporaneamente disponibili in memoria.
La certezza di non dover gestire situazioni di questo tipo consente di snellire notevolmente la gestione anche se ci obbliga a risparmiare il piu' possibile sulle dimensioni del PASSBLOK.
Esso deve infatti contenere almeno nome e password della macchina virtuale per complessivi 16 bytes; a queste informazioni minime va aggiunto un campo indirizzo (minimo 3 bytes) per il concatenamento dei blocchi richiesto dal tipo di implementazione prescelta. La dimensione del blocco, che risulta essere di 19 bytes non consente di realizzare l'allineamento alla pagina che si ottiene invece con blocchi di 16 o di 32 bytes; il problema e' quindi ricondotto alla scelta tra una delle due dimensioni suddette.
La seconda comporta uno spreco di 13 bytes per blocco e

quindi l'uso di un buffer di dimensioni superiori di circa 2/3 rispetto a quelle minime.

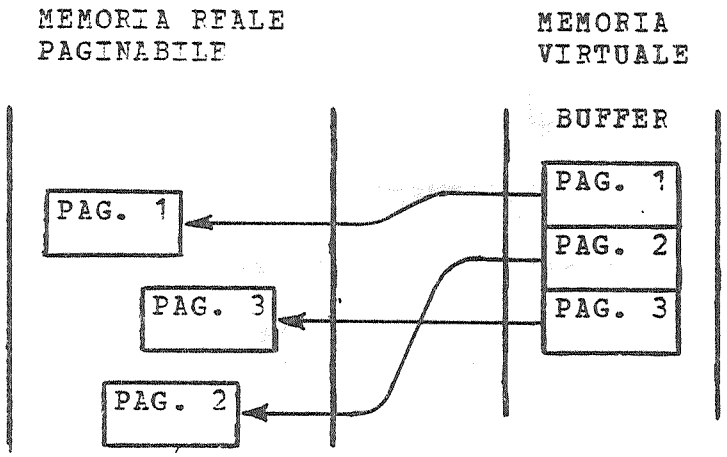

Fig. 6

La prima e' possibile solo se si riescono a compattare le informazioni necessarie in soli 16 bytes.
Questo compattamento e' stato ottenuto:

1) Sostituendo l'indirizzo del blocco successivo con un displacement a partire dall'indirizzo base del buffer (come gia' fatto per le entries della DISPTAB). Questo accorgimento riduce il campo-indirizzo a due soli bytes e torna utile anche perche' ci svincola dalle traduzioni di indirizzi virtuali in indirizzi reali che sarebbero altrimenti necessarie.

2) Compattando in 6 bytes il nome della macchina virtuale. Questa contrazione e' possibile poiche' sul nome delle macchine virtuali esistevano gia' limitazioni relative all'uso di caratteri particolari imposte dai programmi di riduzione dei dati di addebito.

La fig.7 mostra il formato definitivo del PASSBLOK che risulta dai compattamenti appena descritti.

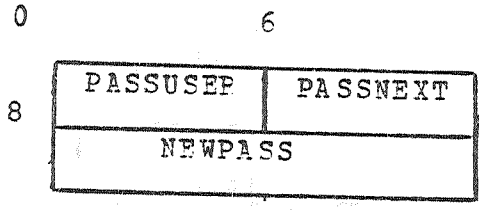

Fig. 7: Formato del PASSBLOK:

PASSUSEP	Userid compattata
PASSNEXT	Pointer al blocco successivo
NEWPASS	Nuova password

I NUOVI COMANDI

Abbiamo gia' accennato alla implementazione di un comando che consente agli utenti di modificare dinamicamente la password della propria macchina virtuale. Desideriamo ora riprendere questo argomento che insieme ai criteri di gestione del buffer rappresenta uno degli aspetti piu' importanti della implementazione.
E' bene notare che questa non puo' limitarsi al solo comando di utente a cui abbiamo accennato, ma deve fornire all'amministratore di sistema tutti i mezzi necessari a mantenere il controllo delle nuove passwords come gia' lo aveva su tutte le altre informazioni di directory.
Infatti l'amministratore di sistema deve essere sempre in grado di indagare sulla situazione delle nuove passwords e di intervenire su di essa qualora sia necessario.
In definitiva la nostra implementazione prevede:

1) un comando d'utente per il cambiamento dinamico di passwords

2) un comando privilegiato (di cui dispone solo l'amministratore di sistema) per conoscere la situazione del buffer

3) un meccanismo che consente all'amministratore di sistema di ricostruire la situazione delle nuove passwords se si verificano errori tali da invalidare il contenuto del buffer.

Nel presente paragrafo non ci soffermeremo sul funzionamento interno dei comandi poiche' si tratta dei meccanismi di ricerca, aggiunta o cancellazione di una entry nel buffer delle passwords gia' ampiamente descritti; ci interesseremo invece prevalentemente di formato ed uso dei comandi citati.
Il comando d'utente, come del resto quello privilegiato, sono stati realizzati sfruttando al massimo il codice standard; per questo motivo abbiamo preferito aggiungere nuovi parametri a comandi gia' esistenti piuttosto che crearne di nuovi.
In particolare il comando d'utente prevede il nuovo parametro PASSWORD (abbreviazione minima PASS) per il comando SET.
Per evitare che persone non autorizzate adoperino questo comando su macchine virtuali di cui non sono proprietarie, magari approfittando di terminali momentaneamente incustoditi, il sistema, appena riceve la richiesta, domanda all'utente la vecchia password. Se questi e' in grado di fornirla il comando viene eseguito altrimenti viene rifiutato.

In questo modo possono adoperarlo solo coloro che conoscono gia' le passwords, e quindi si ritiene che siano autorizzati ad usare la macchina virtuale.
Il SET PASSWORD e' un comando di classe G quindi disponibile a tutte le macchine virtuali; con l'unica eccezione delle macchine aventi codici di addebito che iniziano con le lettere I o E.
Queste macchine non possono modificare la propria password a comando poiche' sono destinate ad esercitazioni e quindi non sono di proprieta' di una singola persona ma a ciascuna di esse ha generalmente accesso tutto un corso di studenti.
In questi casi il nuovo comando darebbe la possibilita' ad un singolo di impadronirsi della macchina e di impedirne l'accesso a tutti gli altri legittimi co-proprietari. Per evitare l'insorgere di simili inconvenienti abbiamo impedito l'uso del SET PASSWORD alle macchine virtuali destinate ad esercitazioni.
La richiesta della vecchia password e' notificata all'utente col messaggio:

ENTER OLD PASSWOPD

che ammette risposta identica, anche formalmente, al messaggio di ENTER PASSWORD della sequenza di login.
Infatti, scritto il messaggio, la tastiera si sblocca e la stringa battuta a terminale e' interpretata come vecchia password; un ritorno di carrello a vuoto e' interpretato come una richiesta di protezione e provoca la stampa di una mascherina. Non appena il sistema dispone della vecchia password ne controlla la validita' e, se il controllo e' positivo, richiede la nuova col messaggio:

ENTER NEW PASSWORD

che ammette risposte analoghe al precedente. Se anche la nuova password e' valida (non supera gli otto caratteri di lunghezza) ha inizio il processo di scambio.
Inizialmente si esegue un confronto tra password vecchia, password nuova e password di directory: se sono tutte uguali la macchina virtuale riceve un messaggio di errore. Se la nuova e' uguale a quella di directory (ma la vecchia non lo e'), significa che la macchina desidera tornare alla sua password standard per cui si provvede ad eliminare dal buffer il blocco che descrive quella macchina.
Se la vecchia password e' uguale a quella di directory significa che e' necessario creare un blocco nuovo nel buffer delle passwords perche' la macchina che ha lanciato il comando non vi figurava ancora. Viceversa se la vecchia password e' diversa da quella di directory significa che la macchina figura gia' nel buffer per cui e' necessario

soltanto sostituire la password nel blocco che la descrive.

Il comando privilegiato e' stato ottenuto dotando di nuovi parametri il QUERY e consente di conoscere la password della macchina virtuale indicata come parametro.
L'amministratore di sistema dispone inoltre di una utility che qli consente di listare il contenuto di tutte le entries occupate del buffer. Questa facility e' utile in caso avvengano errori non rimediabili nella gestione delle nuove passwords per es. errori permanenti nella scrittura su disco delle pagine modificate.
Simili situazioni disabilitano tutte le nuove passwords per cui l'attivita' del sistema procede basandosi solo sulle passwords di directory e l'amministratore di sistema puo' usare l'utility di cui dispone per conoscere il contenuto del buffer in formato immediatamente interpretabile. Egli e' pertanto in grado:

1) Di localizzare il malfunzionamento.

2) Di ricostruire la situazione non appena sia stata rimossa la causa di errore.

Abbiamo avuto modo di verificare che situazioni simili sono molto rare. Infatti fino ad oggi l'unico uso che e' stato fatto dell'utility e' stato quello di controllare periodicamente il livello di occupazione del buffer.
A questo proposito si e' notato che le dimensioni inizialmente associate al buffer (16 K cioe' una capienza di 1024 PASSBLOKS) erano eccessive poiche' il numero di blocchi presenti non superava mai le 250 unita'.
Col passaggio di release (da Release 2 a Release 3) del VM avvenuto nel dicembre 76, abbiamo provveduto a ridurre le dimensioni del buffer a 8K che sono piu' che sufficienti per le esigenze attuali ed anche per quelle prevedibili per il prossimo futuro.

MODIFICHE AI MODULI DI VM

Si tratta, come gia' si e' accennato, di interventi abbastanza pesanti tanto da giustificare ampiamente la scelta, da noi operata, di conglobare il codice inserito in un modulo nuovo (DMKPAS) sistemato in memoria paginabile. Le funzioni svolte dal modulo sono essenzialmente tre e ad ognuna di esse e' associata una specifica entry:

- DMKPASEN esegue il controllo della password ad ogni login di macchina virtuale. Ritorna al modulo chiamante condition codes diversi a seconda che la macchina sia presente nel buffer e la password fornita sia corretta oppure sia presente nel buffer ma la password data da terminale sia errata, o ancora non sia presente nel buffer e quindi il controllo vada eseguito sul directory.

- DMKPASQP esegue tutte le funzioni connesse al comando di QUERY privilegiato.

- DMKPASSP esegue tutte le funzioni connesse alla gestione del comando SET PASSWORD.

Si noti che il lavoro svolto sul buffer e' essenzialmente di due tipi:

- semplice consultazione per le prime due entries di DMKPAS

- aggiornamento per l'altra.

Per quanto la funzione di consultazione si limiti ad una ricerca e quindi sia notevolemente piu' semplice di quella di aggiornamento, che comporta ricerca e modifica (sostituzione o aggiunta o cancellazione), tuttavia esistono due situazioni particolari che complicano notevolmente l'entry DMKPASEN.

1) Si e' detto che il buffer delle passwords e la tabella DISPTAB sono mantenuti in due copie sempre aggiornate e allocate rispettivamente in memoria paginabile e su un'area del disco-sistema. Questo e' sempre vero eccettuato il momento successivo all'IPL del VM quando la sola situazione aggiornata esistente sta sul disco-sistema. A questo punto e' quindi necessario preparare in memoria tutte le condizioni indispensabili al funzionamento dei meccanismi gia' descritti; in particolare e' necessario:

   - portare in memoria la tabella DISPTAB

- portare in memoria, nell'area occupata da DMKPWD, le informazioni del buffer delle passwords salvate sul disco sistema

- ricostruire il valore della prima entry libera del buffer (FRENTRY) che non viene aggiornato su disco.

L'esecuzione di queste operazioni avviene durante l'IPL del VM al momento del 1° login di macchina virtuale che e' solitamente quello dell'operatore di sistema.

2) L'altra situazione che l'entry DMKPASEN deve essere in grado di risolvere e' quella della inizializzazione del buffer nel caso non sia mai stato usato il comando SET PASSWORD. In questo caso e' prevista l'allocazione sul disco-sistema di tante pagine quante ne richiede la dimensione del buffer (attualmente due).
Notiamo esplicitamente che l'unica limitazione esistente a questo riguardo e' che i buffers delle macchine a tempo limitato e quello delle passwords non occupino piu' di 1 cilindro del disco-sistema.
All'interno del cilindro l'allocazione delle pagine all'uno o all'altro buffer non e' soggetta a limitazioni essendo descritta da una tabella di allocazione modificata automaticamente dalle routines di gestione delle 2 funzioni. Anche per questo motivo il cambiamento operato nelle dimensioni del buffer non ha richiesto interventi pesanti: e' stato sufficiente modificare una costante definita nel modulo DMKPAS.

Delle altre entries, quella che gestisce il comando d'utente SET ha richiesto il maggior sforzo di programmazione poiche' deve prevedere operazioni laboriose quali il compattamento e la riscrittura su disco delle sole pagine alterate. Per semplificare queste operazioni si e' deciso di "bloccare" in memoria il buffer all'inizio del processo e sbloccarlo solo al termine delle funzioni di aggiornamento.
Bloccare il buffer in memoria significa renderlo non paginabile e quindi snellire notevolmente il calcolo degli indirizzi reali.

Nonostante il grosso delle modifiche sia raccolto nel modulo ora descritto, si sono rese necessarie altre due categorie di interventi rivolte:

1) A definire i nuovi parametri per i comandi SET e QUERY.

2) Ad invocare l'esecuzione di DMKPASEN ad ogni processo di login.

Il primo punto ha richiesto semplici inserimenti nei moduli standards che gestiscono i suddetti comandi cioe': DMKCFC (comando QUERY) e DMKCFS (comando SET).
Il secondo punto ha richiesto modifiche un po' meno banali delle precedenti che vanno inserite nei moduli che curano il processo di login: DMKLOG e DMKLNK.
La funzione dei due updates e' identica ed e' quella di richiedere il test della password nel buffer (prima di effettuarlo nel directory) ed e' sdoppiata nei due moduli perche' la gestione del comando LOGIN e' diversa a seconda che la password sia specificata tra i parametri del comando stesso oppure sia fornita solo dopo la richiesta (ENTER PASSWORD) operata dal sistema.
Il primo caso e' gestito completamente da DMKLOG, il secondo implica anche l'intervento di DMKLNK e quindi entrambi i moduli hanno richiesto modifiche.
Ricordiamo infine che il buffer stesso costituisce un nuovo modulo (DMKPWD) paginabile non eseguibile in quanto definisce solo una zona di memoria che e' poi gestita da DMKPAS.

Anche da quanto detto in questo capitolo e' possibile riconoscere che l'impatto delle nuove funzioni sul CP-VM e' molto ridotto. L'analisi degli updates permette poi di constatare che questi si compongono di pochissime istruzioni (1 sola per DMKCFS e DMKCFC) convalidando definitivamente questa affermazione.

CONCLUSIONI

A circa un anno dalla installazione della nuova facility si possono trarre alcune conclusioni sul suo funzionamento e sul suo uso.
Per quanto concerne il funzionamento, dobbiamo registrare che e' sempre stato corretto, superiore alle nostre stesse aspettative tanto e' vero che alcuni meccanismi di recovery non sono mai stati adoperati.
Questi meccanismi ci hanno tuttavia consentito, come gia' abbiamo avuto occasione di dire, di tenere sotto controllo l'uso che l'utenza ha fatto della nuova facility.
Da quest'ultimo punto di vista i risultati sono stati meno brillanti anche se parzialmente previsti.
Ci risulta infatti che meno della meta' delle macchine virtuali definite in directory abbia usufruito della nuova possibilita', non solo, ma la maggior parte di coloro che l'hanno sfruttata e' costituita da personale interno dell'Istituto.
Questo fenomeno era prevedibile, tenuto conto che il personale del Centro dispone di macchine virtuali che lavorano molto e che spesso sono usate anche insieme ad utenti esterni e ancora che quelle macchine sono spesso privilegiate ed e' quindi estremamente pericoloso concederne l'uso ai "non addetti ai lavori".
Quello che invece e' venuto un po' a mancare e' stato l'interesse dell'utenza esterna che, pur essendo meno esposta al pericolo del furto di informazioni, tuttavia non ne e' completamente esente.
Uno degli scopi di questo lavoro era anche quello di sensibilizzare tutti al problema e di rendere l'utente direttamente responsabile della sicurezza del proprio lavoro; da questo punto di vista i risultati sono stati inferiori all'attesa.
Al ridotto interesse degli utenti ed anche ad una errata previsione di sviluppo del servizio conversazionale del Centro e' da imputare il sovradimensionamento iniziale del buffer delle passwords.