Tutto quello che un utente Unix vorrebbe sapere sulla riservatezza dei suoi dati e non ha mai osato chiedere

Alberto Cammozzo

15 gennaio 1997

Sommario

Esplorerò brevemente il tema della sicurezza e della riservatezza dei dati su Unix in un ambiente di rete TCP/IP, in particolare una rete connessa a Internet. Il punto di vista sarà quello dell'utente piuttosto che quello dell'amministratore di sistema.

Dopo alcuni elementi fondamentali introduttivi (riservatezza in Unix e sulle reti, chi è `root`) verranno elencate tutte le cose che altri possono sapere su un dato utente, a seconda di chi sono e di che poteri hanno. In seguito sarà spiegato se e come un utente può difendersi, e cercare di fare in modo che i suoi dati restino esclusivamente suoi. In conclusione il problema verrà brevemente toccato dal punto di vista etico e legale.

1 Convenzioni e termini

Essendo questo documento orientato all'utente e non allo specialista, cercherò di evitare i tecnicismi per quanto possibile.

Non sono richieste conoscenze specifiche di Unix e reti, ma è indispensabile una conoscenza di base dei calcolatori. Non verranno spiegati termini quali *directory*, *file*, periferica, linea comandi, PC, posta elettronica, *World Wide Web*, *home directory*.

Userò le seguenti convenzioni:
- il termine *file*,se non altrimenti specificato, denoterà sia i files veri e propri che le directories.
- i nomi dei files che non comincino con "/" sono riferiti alla *home directory* dell'utente.
- collocherò a piè di pagina le informazioni non indispensabili o di carattere maggiormente tecnico.

Con il termine *riservatezza* dei dati intendo la garanzia che il proprietario dei dati è l'unico a poter decidere se e a chi divulgarli. Con il termine *sicurezza* intendo la garanzia che i dati e il sistema che li ospita non possano essere alterati o distrutti da altri che dal proprietario. In gergo, il termine *security* raccoglie il significato di riservatezza e sicurezza insieme, spesso aggiungendo anche quello di *autenticità*, cioè la garanzia dell'origine dei dati; ad esempio nei messaggi di posta elettronica l'autenticità è una proprietà che garantisce al ricevente l'identità del mittente.

2 Elementi fondamentali su UNIX e le reti

- Distinzione root/user e schema di sicurezza di UNIX.

  Il sistema operativo multiutente UNIX è costruito con in mente due livelli fondamentali di priorità distinti: quello di livello 0 e tutti gli altri.
  In questi due livelli vengono inseriti gli identificativi degli utenti, o *uid*: l'*uid* 0 corrisponde all'unico utente di livello 0 il cui nome è `root`. Nell'altro livello si trovano tutti gli altri codici corrispondenti agli altri utenti, un codice per ogni utente. I nomi convenzionali cui corrispondono i codici vengono chiamati *login*, e ogni utente può accedere alla propria *login* tramite una parola d'ordine, o *password*.
  La conoscenza della *login* e della *password* consente l'accesso al sistema, compresa l'area dati personale, chiamata *home directory* nella quale l'utente può scrivere i propri files.

  UNIX è strutturato internamente in modo che all'utente `root` sia consentito ogni possibile accesso a ogni possibile periferica e ad ogni file, mentre gli altri utenti possono accedere alle periferiche ed ai files secondo uno schema o matrice di permessi di cui ogni file è munito.
  Questo schema consente o nega i permessi di lettura, scrittura, esecuzione (o accesso nel caso delle directory) all'utente proprietario del file, al suo gruppo di appartenenza o a tutti gli altri[1].

  È importante fare immediatamente la distinzione tra ciò che ciamerò `root` e l'*amministratore di sistema*: l'amministratore di sistema è la persona fisica o il gruppo di persone che normalmente esercita la funzione interna di `root`.

  Chiamerò invece `root` chiunque abbia modo di accedere alla login di `root` eai suoi privilegi, sia che questa possibilità derivi dal possesso legale della password, sia che derivi da attività fraudolente che abbiano fruttato questo accesso.

- Le mansioni e i privilegi di `root`.

  Unicamente `root` ha il potere di creare e cancellare gli utenti, collegare e scollegare periferiche, fare delle copie di sicurezza che coprano tutti i files presenti nel sistema e in generale qualsiasi mansione tipicamente tecnica.
  In certi casi alcuni di questi poteri possono essere delegati ad altri *uid* ovvero altri utenti, ma questa prassi introduce sovente problemi di sicurezza, in quanto risulta difficile assicurarsi che i poteri delegati si limitino a quelli desiderati, e non vi sia nessuna possibilità di ottenerne altri.

[1] Le caratteristiche dei permessi sono descritti in dettaglio nel manuale UNIX sotto le istruzioni d'uso del comando `chmod`.

Basti dire che buona parte dei difetti di sicurezza di UNIX attualmente scoperti coinvolgono questa possibilità di delega di poteri[2].

- Le reti locali e geografiche, Internet e sicurezza.

  Le reti sono degli insiemi di apparati e di sistemi di comunicazione che mettono in contatto i calcolatori tra di loro: molto sommariamente si può dire che la rete *Internet* è composta da un insieme di reti locali (che coprono un edificio o alcuni edifici), e reti geografiche che connettono le reti locali tra loro. Ogni calcolatore connesso e ogni apparecchio destinato a collegare le reti locali tra loro viene detto *nodo*.

  Quando due calcolatori devono scambiarsi dei dati, ad esempio un messaggio di posta elettronica, vengono attivati una serie anche cospicua di apparati quali cavi di rete, linee telefoniche e calcolatori dedicati all'instradamento (detti *routers*) lungo tutto il percoso che collega i due calcolatori: ad esempio un messaggio che vada dall'Italia alla Svizzera passerà attraverso una dozzina di *routers*.

  La natura di questi apparati fa si che non sia impossibile per persone motivate e dotate di mezzi tecnici e conoscenze adatte captare il traffico di messaggi che passa per un nodo locale. Questa attività di intercettazione viene chiamata in gergo *wiretapping* e *packet sniffing*.

  Talvolta questa attività può essere particolarmente facile, per esempio nelle reti locali non segmentate, vale a dire reti in cui i calcolatori si collegano tra loro attraverso un unico cavo non interrotto: un semplice PC munito di scheda di rete e dei programmi adatti può captare, filtrare e registrare tutto quello che passa per il cavo di rete cui è collegato.

3 Chi può sapere cosa

Questa sezione è strutturata in modo da fornire l'elenco delle informazioni che ogni possibile soggetto, da una persona esterna a `root`, possono ottenere su un dato utente, e come.

1. Qualsiasi persona con **accesso alla rete Internet** e la conoscenza dell'indirizzo di posta elettronica dell'utente può ottenere:

   (a) Le informazioni disponibili con il comando `finger <utente>@<macchina>`. Queste solitamente sono: il nome di login, il nome e cognome, la data dell'ultimo collegamento, la data dell'ultima volta che l'utente ha letto la mail, la data di ricezione dell' ultima mail, altre informazioni presenti nel quinto campo del file `/etc/passwd` e il contenuto del file `.plan` se presente. Talvolta l'amministratore di sistema ritiene opportuno disabilitare `finger` per motivi di sicurezza, o comunque ridurre la quantità di informazioni disponibili.

   (b) Il contenuto dei files presenti in una eventuale area personale ftp o WWW: si suppone che questi siano files messi a disposizione intenzionalmente.

[2] Per un approfondimento su questa possibilità di delega si può consultare la documentazione sul *setuid bit*, ad esempio la pagina del manuale UNIX della libreria `setuid`.

(c) L'indirizzo della macchina personale dell'utente, nel caso il nome della macchina corrisponda a quello dell'utente, o che la posta venga spedita direttamente da lí.

In caso di *scanning*[3] da parte di soggetti esterni, potrebbe essere rivelata la matrice dei collegamenti e delle garanzie accordate tra le macchine[4]. Queste riguardano l'utente nel caso la macchina da cui e' collegato venga identificata come sua e nel caso lo *scanning* abbia rivelato delle faglie nella sicurezza.

Se qualche soggetto esterno venisse a conoscenza di passwords degli utenti o addirittura di quella `root` potrà ottenere tutte le informazioni che sono disponibili all'utente o a `root`. Lo stesso avverrà nel caso che qualcuno penetrasse nel sistema per un difetto in qualche programma o per una negligenza dell'amministratore di sistema.

2. Qualsiasi persona in possesso della **password**[5] **di un utente registrato** può ottenere:
   (a) Il contenuto di tutti i files di proprietà dell'utente stesso, come è ovvio.
   (b) Il contenuto di tutti i files di proprietà altrui di cui sia consentita almeno la lettura.
   (c) La sovrascrittura del contenuto di tutti i files di proprietà altrui di cui sia consentita la scrittura.
   (d) La sovrascrittura e l'aggiunta di files nelle directories di proprietà altrui in cui sia consentita la scrittura.
   (e) La lettura del solo elenco dei files nelle directories di proprietà altrui di cui sia consentita la lettura.
   (f) Le informazioni sull'utente presenti in `/etc/passwd`: nome, cognome e altre informazioni, in molti casi anche la password cifrata. Se non e' abbastanza complessa la password può essere "indovinata" (anche se non decifrata) da programmi appositi.
   (g) Le informazioni su tutti i processi in esecuzione e in certi casi su quali files sono coinvolti (se presenti in linea comando). Tramite il comando `ps`[6] ogni utente può sapere quali comandi sono stati lanciati (e sono ancora attivi) dagli altri utenti e con quali argomenti. È tecnicamente possibile monitorizzare un utente registrando praticamente tutti i comandi lanciati.
   (h) Le informazioni registrate dal sistema che riguardano chi si è collegato con i programmi `login`, `telnet` e `ftp`. Con il comando "`last` *nome utente*" si può sapere quando, da dove e per quanto tempo l'utente si è collegato. Queste registrazioni solitamente coprono un periodo di alcuni mesi.
   (i) Le informazioni presenti nei files leggibili nella directory `/tmp`. Questi sono files temporanei generati dai programmi in esecuzione. Solitamente i programmi non rendono questi files leggibili da estranei, ma può capitare anche il contrario.

[3] Si tratta un vero bombardamento della rete con messaggi tendenti a acquisire informazioni sulla sicurezza

[4] Ovvero collegamenti NFS, utenti presenti su diverse macchine,etc.

[5] O comunque che abbia accesso alla *login*

[6] `ps -auwwx` oppure `ps -fea` a seconda se lo UNIX locale sia di tipo BSD o System V. Il comando `ps` fornisce anche la dimensione del programma in esecuzione, la quantità di CPU utilizzata, ecc...

(j) Alcune informazioni relative ai messaggi di posta elettronica spediti dall'utente ma non ancora arrivati a destinazione, tramite il comando `mailq`.

(k) L'elenco delle stampe inviate e non ancora stampate tramite il comando `lpq`.

(l) Alcune informazioni sulle connessioni attive in rete: non sempre è possibile associare univocamente la connessione (ad esempio risultanti dai comandi `ftp` o `telnet` ad un certo indirizzo) ad un utente, ma se il numero di utenti non è enorme, si può risalire a i collegamenti aperti da un dato utente attraverso il comando `netstat`. Certi dialetti di UNIX (come Linux) indicano esplicitamente l'autore della connessione, oltre alla destinazione e al protocollo usato. Anche in questo caso è possibile monitorizzare i collegamenti attuati dall'utente.

(m) L'accesso all'identità stessa (cioè la *login*) dell'utente, se viene consentito l'accesso in scrittura alla propria *home directory* o ad alcuni particolari files. Ad esempio i files `.rhosts`, `.login` e la directory `bin` sono particolarmente pericolosi se scrivibili.

3. Una persona con accesso alla **password** [7] **di root** può:

(a) Leggere il contenuto di qualsiasi file nel sistema (compresa la mail di chiunque).

(b) Accedere al contenuto di qualsiasi transazione in rete che sia possibile intercettare (con *packet sniffer*).

(c) Accedere al contenuto di qualsiasi schermo e alle battute dei tasti degli utenti collegati (con programmi adatti).

(d) Accedere al contenuto del file `.history` o equivalente contenente il tracciato dei comandi eseguiti interattivamente da qualsiasi utente faccia uso di interpreti di comandi (*shells*) che prevedano la registrazione dei comandi eseguiti. L'utente può cancellare periodicamente il file in questione, mantenere il tracciato a pochi comandi o anche scegliere altre shells che non prevedono questa possibilità.

(e) Accedere al contenuto delle directory ove `netscape` e altri *Web Browsers* tengono l'indice delle pagine recentemente consultate dall'utente (ad esempio `.netscape-history`) e le sue preferenze (`.netscape-bookmarks.html`).

(f) Accedere al contenuto del file `.newsrc`, che contiene l'elenco dei *newsgroups* che l'utente legge, e quali articoli sono stati letti per ciascun gruppo. In generale tutti i programmi che devono tenere un indice di preferenze o una qualche registrazione per conto dell'utente lo fanno in files "nascosti" (in cui primo carattere del nome è un punto) nella *home directory*

(g) Accedere alle registrazioni (*logs*) di vari programmi in uso sulla macchina. Ad esempio il programma responsabile della posta elettronica conserva una registrazione a scopo diagnostico unicamente della provenienza e della destinazione della posta elettronica. Altri programmi possono tenere una registrazione di altri eventi, quali ad esempio il prelevamento di files via `ftp` o la richiesta di usare particolari programmi, come `su`.

[7] O comunque che abbia accesso alla *login*

(h) Accedere alle registrazioni sul traffico della rete.
In certe reti sono attivi dei programmi che registrano dettagliatamente ora, provenienza, destinazione e protocollo usati per ogni connessione di rete rivolta verso l'esterno della rete locale, o provenienti dall'esterno. Queste registrazioni vengono effettuate per motivi di sicurezza, in modo da poter identificare o ricostruire accessi alla rete di natura insolita o sospetta. Combinando insieme registrazioni diverse (uso della rete, dei programmi, spazio su disco occupato, tempi di collegamento) è possibile creare degli indici che classifichino le varie tipologie di utente.

4. Una persona con **accesso fisico (visivo) alla console**[8] può ottenere informazioni contenute nelle registrazioni di eventi (*log*) inviati alla console. Tipicamente questi sono: errori di sistema, notifiche del collegamento di utenti, a volte l'origine e la destinazione dei messaggi di posta elettronica, o anche la registrazione dei collegamenti di rete da e per l'esterno della rete locale, a seconda di come e' configurato il *logging* (vedere il file `/etc/syslog.conf`).

5. Qualsiasi persona sullo **stesso spezzone di rete** che abbia completo controllo di una macchina[9] e sia munito di programmi e conoscenze tecniche adatti può intercettare e quindi leggere, filtrare e registrare qualsiasi comunicazione, compreso lo scambio di passwords, la posta elettronica, i collegamenti remoti, cioè verso macchine non appartenenti alla rete locale.

   Esistono addirittura dei programmi che permettono ad un terzo di **inserirsi** nella comunicazione e di inviare comandi a proprio vantaggio. Questa tecnica si chiama *ip-spoofing*, e si sta gradualmente affermando, benchè richieda conoscenze approfondite sui meccanismi interni delle reti.

6. Le persone con accesso ai **backup** (nastri di salvataggio dei dati) possono leggere, senza la necessità di conoscere alcuna password, il contenuto qualsiasi file sia stato salvato, ammettendo che abbiano accesso ad apparecchiature in grado di leggere i supporti su cui è stato effettuato il salvataggio.

7. Le persone con accesso fisico ai **dischi** estratti dalla macchina hanno accesso, senza la necessità di conoscere alcuna password, a qualsiasi file situato sul disco stesso. Alcuni calcolatori hanno dischi esterni la cui rimozione è decisamente agevole.

4 Come difendersi

I pericoli per la riservatezza e la sicurezza dei dati sono rappresentati principalmente da tre fattori: l'abuso dei privilegi di *root*, la curiosità degli altri utenti del sistema e la possibilità che i dati trasmessi in rete vengano intercettati.

La cifratura dei dati, ovvero la codifica a mezzo di una parola o una frase chiave rappresenta una buona soluzione, anche se non sempre è agevole.

[8] La *console* è il terminale o lo schermo principale del calcolatore, solitamente situato vicino all'unità centrale.

[9] Basta un qualunque PC munito di scheda di rete.

1. Cifratura.
   Il fatto di poter leggere i files o di poter captare i messaggi che i calcolatori si scambiano non è sufficiente se quello che viene letto è stato reso inintellegibile a terzi con la cifratura.
   Esistono diversi sistemi di crittografazione molto seri secondo i quali sono stati scritti numerosi programmi, anche di pubblico dominio:

   Per garantire la segretezza dei **files** si possono usare diversi programmi di cifratura, tra cui alcuni che utilizzano sistemi avanzati come `DES`, `RSA` e altri.

   Uno dei sistemi più interessanti per lo scambio sicuro di **messaggi di posta elettronica** è quello a "chiave pubblica", per cui è possibile garantire autenticità e/o segretezza senza dover scambiare preventivamente delle chiavi segrete, ma semplicemente prelevando la chiave pubblica (cioè non segreta) della persona con cui si vuole corrispondere. Un programma di pubblico dominio molto diffuso su tutti i principali sistemi operativi è `PGP` (*pretty good privacy*),. che ha riscosso un tale successo da allarmare le autorità di mezzo mondo (in particolare quelle statunitensi) essendo a quanto sembra non decifrabile nemmeno dalle ben attrezzate agenzie governative.

   Più difficile è rendere sicure le **comunicazioni interattive**, quali quelle necessarie per collegarsi o trasferire files da macchine remote (telnet, ftp) oppure su *World Wide Web*: esistono molti sistemi, talvolta piuttosto complessi e non sempre così sicuri come a prima vista sembrano.

2. Proteggersi dagli abusi di **root**.
   Il fatto che `root` goda di privilegi straordinari rende la riservatezza dei dati degli utenti dipendenti quasi esclusivamente dal rispetto della deontologia professionale o dal disinteresse della persona o delle persone che ne conoscono la password, e che essi siano i soli ad avere accesso alle funzioni di `root`[10].
   Esistono chiaramente delle possibilità di controllare l'operato di `root` tramite appositi programmi, ma `root` avrà comunque la possibilità di accedere ai programmi stessi e ai dati da essi generati; il problema è antico: "quis custodiet ipsos custodes?".

   Chi volesse proteggere alcuni files dagli occhi di root dovrà fare ricorso alla cifratura, ma va comunque detto che `root` ha la possibilità di leggere le battute sulla tastiera, e di conseguenza può teoricamente leggere i codici di cifratura mentre questi vengono digitati. L'utente dovrebbe quindi cifrare e decifrare i files *off-line*, cioè su una macchina "sicura[11]".

3. Proteggere i propri files dagli **altri utenti**.
   Mettere al riparo i propri dati dalla curiosità degli altri utenti è possibile con uno sforzo minimo. L'utente deve periodicamente verificare i permessi dei files nella propria *home directory* e quelli dei files eventualmente generati in `/tmp`[12] dai programmi che usa.

[10] Quasi tutta la sicurezza di un sistema UNIX dipende dalla capacità di limitarne l'accesso a `root` unicamente al legittimo detentore della password

[11] A prima vista potrebbe sembrare che un PC non collegato alla rete sia sicuro, ma va notato che solitamente i PC non hanno validi controlli di accesso ai dati: la disponibilità fisica della macchina fornisce automaticamente la disponibilità ai dati in essa contenuti

[12] La directory /tmp viene comunque periodicamente svuotata, di solito ogni notte.

Chi volesse precludere l'accesso a tutti i files contenuti nella *home directory* può farlo una volta per tutte, chiudendo alla lettura e all'esecuzione la directory stessa fuorchè al proprietario[13].

Un altro modo è quello di ordinare al sistema di creare i nostri files secondo uno schema di permessi più rigido del solito, che assegna la lettura e la scrittura unicamente al proprietario[14].

Alcuni programmi rispettano la convenzione di generare i files temporanei nella directory temporanea indicata dalla variabile di ambiente `$TMP` anzichè in `/tmp`. Volendo si può fare in modo che questa variabile indichi una directory all'interno della propria *home directory*.

4. Proteggersi dall'**intercettazione** lungo la rete.

   Proteggersi dal *packet sniffing* è arduo. Le uniche soluzioni veramente efficaci sono al momento due: la prima è l'estrema **frammentazione della rete** fino al limite di un segmento per ogni macchina, a mezzo di apparecchiature chiamate *switch*, la seconda è la **cifratura** della comunicazione dal principio alla fine.

   Il beneficio dell'uso degli *switch* nella rete è soprattutto quello di renderla molto veloce; quello di impedire l'intercettazione delle comunicazioni è un salutare "effetto collaterale". Restano comunque soggette a intercettazione tutte le comunicazioni con reti esterne alla rete frammentata con lo *switch* (collegamenti remoti), che possono essere rese sicure unicamente con la cifratura.

   Quest'ultima introduce dei livelli di complicazione ulteriori ad ogni connessione, in particolare se ci colleghiamo remotamente: potrebbero essere necessari piccoli apparecchi che forniscono la parola chiave adatta al momento, o elenchi di passwords diverse, o altri sistemi non sempre confortevoli. Inoltre non esistono standards affermati per le comunicazioni cifrate, e in altre reti raramente troveremo programmi di cifratura compatibili coi nostri.

Un *minimo* di "paranoia" aiuta talvolta a stare attenti, a cambiare spesso la password, soprattutto se ci siamo collegati da fuori rete locale, a non lasciare in giro files il cui contenuto non vogliamo assolutamente che vada in giro, ecc. Ricordo che fa parte dei compiti dell'utente controllare i permessi dei propri files, cambiare frequentemente le passwords, eventualmente gestire i propri sistemi di cifratura ed autenticazione.

5 L'etica dell'utente e dell'amministratore

"I files leggibili sono pubblici, i files protetti sono privati, root può fare tutto". Questa è la massima che definisce ogni comportamento basandosi unicamente sulla struttura di Unix. In pratica dovrebbe valere **tra utenti** la cortesia di non andare a rovistare tra i files degli altri, specie se si sa che non sono utenti esperti. Nel caso qualcuno lasciasse la *mailbox* leggibile a tutti non è detto che desideri che tutti leggano la sua posta.

[13] Dando il comando " `cd ; chmod 700 .` " la *home directory* verrà chiusa.
[14] Il comando è " `umask 007` ".

L'etica dell'**amministratore di sistema** è più ferrea: deve di norma rispettare, pur potendola violare, la matrice di permessi assegnata dall'utente ai suoi files. Esistono situazioni in cui l'amministratore deve operare sui files dell'utente, e normalmente questo avviene con l'esplicito consenso dell'utente stesso. In casi eccezionali l'amministratore opererà sui files anche senza autorizzazione[15], senza però occuparsi del loro *contenuto*.

Vi sono poi altri campi in cui l'operato dell'amministratore di sistema va regolamentato con protocolli di comportamento secondo i casi:

- Profondita' del *logging* e accesso ai *logfiles*.
  Nel caso vi siano numerose registrazioni delle attività dei computers e della rete è possibile ricostruire con un certo dettaglio quello che un utente ha fatto o sta facendo. Le politiche di utilizzo delle registrazioni devono indicare in quali occasioni, per quali motivi e con quali garanzie vanno fatte queste indagini.
  Un buon sistema per mantenere un livello anche elevato di registrazioni senza violare la *privacy* dell'utente è quello di usare programmi appositi che analizzino il contenuto delle registrazioni e notifichino all'amministratore unicamente gli eventi anomali o sospetti.
- Emergenze. L'amministratore deve conoscere quale accesso ai files dell'utente gli è consentito in caso di emergenza, in caso di sospetta intrusione da parte di estranei, etc.
- Controlli di sicurezza. La responsabilità della sicurezza complessiva del sistema obbliga l'amministratore a controllare periodicamente il contenuto di alcuni particolari files di ogni utente, come ad esempio `.rhosts`, che possono accordare garanzie ad altri utenti di altre reti, e dietro ai quali spesso si nascondono le intrusioni già avvenute.
  È buona prassi controllare la solidità delle passwords degli utenti tentando di indovinarle come farebbe una persona intenzionata alle intrusioni.
  Vanno anche controllati i permessi di tutti i files, per evitare che possano compromettere la sicurezza del sistema.

6 Considerazioni legali

L'art.4 della legge n.547 23/12/93 sulla criminalità informatica (Art 615-ter del Codice penale) dice:

> Chiunque abusivamente si introduce in un sistema informatico o telematico protetto da misure di sicurezza ovvero vi si mantiene contro la volontà espressa o tacita di chi ha il diritto di escluderlo, è punito con la reclusione fino a tre anni.

...

[15] Ad esempio se un programma dell'utente sta generando così tanti dati da riempire i dischi e mettere in pericolo la stabilità o funzionalità del sistema, oppure nel caso nei confronti dell'utente vengano presi provvedimenti amministrativi.

e L'art.6 della stessa legge (Art 617-quater del Codice penale) dice:

> Chiunque fraudolentemente intercetta comunicazioni relative a un sistema informatico o telematico o intercorrenti tra più sistemi, ovvero le impedisce o le interrompe, è punito con la reclusione da sei mesi a quattro anni.
>
> ...
>
> Tuttavia si procede d'ufficio e la pena è della reclusione da uno a cinque anni se il fatto è commesso:
> 1) in danno di un sistema informatico o telematico utilizzato dallo Stato o da altro ente pubblico o da impresa esercente servizi pubblici o di pubblica utilità.
> 2) da un pubblico ufficiale o da un incaricato di un pubblico servizio, con abuso dei poteri o con violazione dei doveri inerenti alla funzione o al servizio, ovvero con abuso della qualità di operatore del sistema;
> 3) da chi esercita anche abusivamente la professione di investigatore privato."

In sintesi chiunque, non essendo autorizzato a farlo, intercetti delle comunicazioni, si introduca in un calcolatore o abusi dei privilegi di amministratore di sistema commette un reato.

Rimane il dubbio se sia legale o no utilizzare sistemi di cifratura per inviare dati attraverso le reti.

7 Conclusioni

Da quanto detto fino ad ora, risulta evidente che la riservatezza dei dati personali in una macchina **Unix** *tecnicamente* non esiste. Ed esiste in pratica solo se si ripone completa fiducia nella discrezione dell'amministratore di sistema e nelle sue capacità di rimanere unico detentore dei privilegi di `root`.

La riservatezza nelle **reti** e in particolare in Internet è compromessa gravemente dalla possibilità di intercettazione. Questa tuttavia è una possibilità teorica non molto nota, e non sempre facile da mettere in pratica senza i mezzi adeguati e una esatta conoscenza del tipo di informazione da ottenere. Non è pensabile di intercettare tutto il traffico che passa per un grosso nodo senza mezzi molto potenti e senza una certa idea di quello che veramente si sta cercando.
Più facile è "collezionare" passwords su un ramo della rete locale, per poterle sfruttare con calma.

Se ogni persona nella rete conoscesse e volesse sfruttare tutte le opportunità messe a disposizione dalla conoscenza dei protocolli di rete e dal vertiginoso aumento dei difetti gravi (*bugs*) scoperti nel software correntemente usato, avrebbe inizio un saccheggio al termine del quale nessuno lascerebbe più un bit con un minimo di valore su un disco fisso connesso alla rete; tuttavia, visto che questo non succede, sembra che manchino sia la conoscenza diffusa dei problemi, sia la volontà di sfruttarli su larga scala.