

# HAX

*Hax Are eXperience*

# N° 2

OTTOBRE  
2007

## STICUREZZA

- Truecrypt



## GRAFICA

- Realizziamo un logo con GIMP

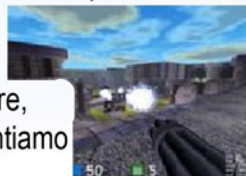
## PERSONAGGIO

- Kevin Mitnick



## GIOCHI

Da un pò di tempo i produttori di schede grafiche stanno dedicando più attenzione agli utenti di Linux, ed ecco che i giochi cominciano a fioccare, questo mese presentiamo CUBE.



## SPECIALE



Linux e il software libero all'interno delle pubbliche amministrazioni e nelle forze armate.

L'adozione e il passaggio, le ragioni economiche, il lato tecnico, tutto analizzato dal nostro Umit Uygun.

www.bhioart.com

<http://hax.cosenzainrete.it>

**HACS**  
HACKLAB COSENZA



**HAX****HAX ARE EXPERIENCE**FRANCESCO MUSCIMARRO  
"CICCIORAPTOR"  
(GRAFICA, IMPAGINAZIONE)VINCENZO BRUNO  
(REDAZIONE, ARTICOLISTA)GIUSEPPE GUERRASIO  
"LOBOTOMIA"  
(ARTICOLISTA)PIETRO LAUNI  
(ARTICOLISTA)UMIT UYGUR  
(REDAZIONE, ARTICOLISTA)DANIELE DI VITO  
(ARTICOLISTA)MARIO LAGADARI  
(ARTICOLISTA)ROCCO SPANÒ  
(ARTICOLISTA)ROCCO FOLINO  
"LORD ZEN"  
(ARTICOLISTA)ANTONIO GENTILE  
"ANTOFRAGE"  
(ARTICOLISTA)ONO-SENDAI  
(ARTICOLISTA)ALESSANDRO AVOLIO  
IDREAMER  
(ARTICOLISTA)IGNAZIO FILICE  
(ARTICOLISTA)ENZO AGGAZIO  
(ARTICOLISTA)FRANCESCA BEATRICE CICE  
(ARTICOLISTA)**FUORI DAI BINARI**

Siamo al numero 2. Sì, fuori dallo schema 0 e 1, fuori da quello che viviamo ogni giorno. Fuori con questo nuovo numero.

Innanzitutto mi permetto di rinumerarlo in numero 00000010 (permettetemi la geekkata). Fuori dai Binari anche perchè ormai HAX è una realtà che riscuote sempre più successo e questo grazie a tutti coloro che ci stanno seguendo, a tutti coloro che ogni giorno ci aiutano a realizzare HAX, a tutti i nuovi arrivati tra i collaboratori.

In questa uscita voglio focalizzare in particolare due cose: Hackmeeting e Microsoft.

Settembre è stato il mese di Hackmeeting, giunto ormai alla decima edizione, si rinnova, cresce e, ancora una volta, diventa luogo di incontro e di scambio (contro)culturale.

Chi quest'anno è riuscito ad essere presente nei tre giorni di attività presso il centro sociale Rebeldia di Pisa ha potuto vivere la realtà italiana del momento per quanto riguarda l'hacking. Niente sponsor, niente multinazionali con i loro soldoni, solo gente che ha voglia di fare e di collaborare, di conoscere ed esplorare.

Si è parlato come sempre di Linux, si è parlato di hacking di codice, si è parlato di hacking di vita, si è parlato di musica e di video, si è parlato del Sistema. Un solito punto comune: stravolgere il normale, uniti da uno slogan "code is written, future is not!".

Purtroppo non ci sono stato, ma leggendo qua e là e seguendo la mailing list, a cui vi invito a iscrivermi, è stato anche questa un'ottima annata.

Dall'altra parte del modo arrivano invece strane notizie, Novell che sviluppa un Silverlight Open Source, Novell che stringe accordi con Microsoft, Microsoft che vuole a tutti i costi un proprio formato "aperto" per i documenti, ma che per fortuna al momento sembra solo una loro vana speranza, ed infine Microsoft che rilascia il codice delle librerie di base di .Net.

Al di là dei vari giri "legali" in realtà .Net è stato rilasciato sotto una licenza NON Open Source e non solo, tale licenza impone stretti legami con Microsoft stessa, di fatto limita la libertà di chi legge il codice sorgente di .Net ed inoltre presenta una clausola sui brevetti, Microsoft non potrà essere in alcun modo citata in giudizio.

Oltre questo fattore, il codice potrà essere visto, ma non modificato nè compilato.

Lascio alla vostra immaginazione il giudizio su tale mossa.

Vi lascio alla lettura di questo numero in cui tra l'altro si parla di Kevin Mitnick, di grafica e giochi, di programmazione ed infine nell'ampio speciale si parla del software libero nelle forze armate.

Buona Lettura popolo dell'underground digitale

Giuseppe Guerrasio

**PUBBLICAZIONE  
APERIODICA**

**RIVISTA IMPAGINATA CON OPENOFFICE.ORG**

**HAX — HAX ARE EXPERIENCE**

è garantito il permesso di copiare, distribuire e/o modificare  
questo documento seguendo i termini della Licenza per  
Documentazione Libera GNU, Versione 1.1 o ogni versione  
successiva pubblicata dalla Free Software Foundation.

[HTTP://WWW.GNU.ORG/LICENSES/FDL.HTML](http://www.gnu.org/licenses/fdl.html)

Si ringrazia per la grafica della copertina di questo mese Antonio Pizzonia  
[www.bhioart.com](http://www.bhioart.com)



# SOMMARIO

## NEWS

Linux Day CZ, Ubuntu 7.10 \_\_\_\_\_ pag.4

## FAQ

Le domande più frequenti \_\_\_\_\_ pag.6

## PERSONAGGIO

Kevin Mitnick \_\_\_\_\_ pag.7

## WEB

Radiohead \_\_\_\_\_ pag.11

## COMMUNITY

Free Hardware Fondation \_\_\_\_\_ pag. 12

## GIOCHI

Cube \_\_\_\_\_ pag. 13

## PROGRAMMI

Grass \_\_\_\_\_ pag. 14

## SPECIALE

Software Libero nelle pubbliche amministrazioni \_\_\_\_\_ pag. 16

## GRAFICA

Realizziamo un logo con GIMP \_\_\_\_\_ pag. 21

## SICUREZZA

Truecrypt \_\_\_\_\_ pag. 22



## LINUX DAY CZ

Il neonato Hacklab CZ organizza il Linux Day a Catanzaro, Il tema principale sarà:

### Legalità e il Software

Qui di seguito trovate il programma dell'evento inviatici da uno degli organizzatori, Giuseppe Leone.

Il luogo e la data dove si svolgerà l'evento è:

Catanzaro Lido, Hotel Niagara, 27 Ottobre

Questo e' il programma:

9,00 - Apertura

9,30 - Presenzazione HackLab - Nuccio Cantelmi ( HackLab Catanzaro )

10,00 - Legalità e Software - Nuccio Cantelmi ( HackLab Catanzaro )

10,30 - Introduzione al Software Libero - Vincenzo Bruno ( HackLab Cosenza )

11,00 - Pausa

11,15 - Imprese e Software Libero - Leonardo Aggazio ( HackLab Catanzaro )

e Dario Berretto

12,00 - Dibattito libero

13,00 - Pausa pranzo

15,00 - Dimostrazione distribuzioni GNU/Linux ed applicativi Free Software

con dimostrazioni pratiche

16,30 - Install party - Installazione distribuzioni e assistenza tecnica;

18,00 - Chiusura



Ubuntu 7.10, nome in codice Gutsy Gibbon, è l'ultima versione della distribuzione Ubuntu basata sullo GNOME Desktop, un ambiente desktop multi-piattaforma molto popolare.

Ubuntu viene rilasciato regolarmente ogni sei mesi, comprendendo tutte le migliori tecnologie informatiche che il mondo del software libero mette a disposizione. Ubuntu è software libero e in quanto tale fornisce tutte le libertà derivanti dal software libero, oltre a essere completamente gratuito.

I principali miglioramenti di Ubuntu 7.10 comprendono: miglior supporto alle lingue, strumenti di ricerca integrati nell'ambiente grafico, sistema delle preferenze semplificato, effetti visivi all'avanguardia, nuovi strumenti per la configurazione di schermi e grafica e stampanti, una gestione energetica migliorata e molto altro

### BulletProof-X

Con questa nuova caratteristica, Ubuntu 7.10 ora non presenterà più delle schermate incomprensibili nel caso in cui non venga riconosciuto il proprio monitor. Oltre a ciò, questa nuova funzionalità è in grado di analizzare il CD contenente i driver del proprio monitor alla ricerca dei valori ottimali di configurazione.

### Scrittura NTFS

Con Ubuntu 7.10 è ora possibile leggere e scrivere sulle partizioni NTFS in modo predefinito grazie all'integrazione del progetto NTFS-3g. In questo modo la condivisione di documenti e file con Windows è ancora più facile.

### Nuovo kernel Linux

Con l'ultimo kernel Linux, oltre alle tante migliorie apportate, è ora presente dynticks, sistema che consente ai microprocessori di consumare e scaldare meno. Per i computer portatili questo significa durate batteria più lunga e per i PC domestici un ambiente meno rumoroso.

## Linux contiene tutte le applicazioni di cui ho bisogno?

Si. La maggior parte delle applicazioni sono distribuite insieme a Linux, tuttavia se ciò che vi serve non è compreso nella distribuzione che possedete, potete tranquillamente scaricarla dalla Rete, compilarla e voilà... il gioco è fatto.

Generalmente i grossi database relazionali, sofisticati programmi di grafica, di gestione di suono e video non vengono distribuiti insieme ma potete scegliere il prodotto che vi interessa e downloadarlo voi stessi. Applicazioni desktop quali world processor, sofisticati fogli elettronici, database, analisi statistica o matematica sono generalmente sviluppati da terze parti quali Applixware, StarOffice, Corel, Empress, Wolfram Research.. e possono essere acquistate a parte (il loro costo è comunque di gran lunga inferiore alle versioni per Windows).

## QUESTO MESE UN PICCOLO GLOSSARIO

**Compilatore** Per realizzare un'applicazione occorre dapprima scrivere il codice (detto *sorgente*) con un linguaggio di programmazione quale il C o il C++, quindi elaborarlo da parte di un programma (detto, appunto, compilatore) che trasforma le istruzioni nel linguaggio binario utilizzato dal calcolatore. Occorre quindi collegare (o *linkare*) il file ottenuto con le librerie per ottenere un file eseguibile o binario.

**File System** Per immagazzinare le informazioni sotto forma di file su un supporto fisico, quest'ultimo deve essere strutturato mediante un filesystem (letteralmente *sistema di archiviazione*). I più diffusi sono FAT16, FAT32 (Dos/Windows), HTFS (Windows NT/2000/XP), ext2,ext3 (Linux), ISO9660 (CD-Rom)HOWTO Letteralmente: "Come si fa a...". Una serie di testi che spiegano come si fa a compiere una particolare operazione. Sono presenti solitamente nella directory /usr/share/doc/HOWTO. Raramente sono utili ai principianti dato l'alto tasso di termini tecnici utilizzati

**Libreria** Pessima traduzione italiana dell'inglese *library* (biblioteca). Si tratta di svariati files binari o eseguibili necessari per l'esecuzione dei programmi. Le librerie statiche sono incluse nei programmi all'atto della compilazione, creando programmi molto voluminosi; quelle dinamiche sono invece utilizzate solo al lancio del programma che le richiedono.

**LILO Linux LOader**, o "caricatore di Linux". È un programmino grazie al quale è possibile lanciare Linux o un altro sistema operativo installato.

**Man** Comando mediante il quale è possibile visualizzare il manuale di moltissimi comandi. Si avvia



da terminale, ma ne esiste una versione grafica (*xman*)

**Modulo** Una sezione del kernel, spesso dedicata al pilotaggio di un particolare dispositivo. Noto sotto altri sistemi come *driver*.

**Montare** L'atto di aggiungere un dispositivo (floppy disk, CD-Rom,...) al sistema. Normalmente va eseguito manualmente con il comando *mount* e smontato con il comando inverso, *umount*, ma le distribuzioni orientate ai principianti forniscono un metodo per il montaggio automatico (*automount* o *supermount*).

**Pacchete(pacchetti)** Un file unico, compresso, che contiene quanto occorre per installare e configurare un programma. I pacchetti possono essere di binari o di sorgenti e nel primo caso sono specifici per ogni distribuzione.

**Partizione** È una sezione del vostro disco rigido dedicata a un sistema operativo o una funzione. Nei sistemi Linux, vi sono almeno due partizioni, la partizione principale e quella di swap (per la memoria virtuale). Un sistema Windows o Macintosh usa solitamente un'unica partizione.

**Percorso** Stabilisce dove si trova il file. È posto davanti al nome del file e consiste in una serie di nomi separati da barre. Può essere assoluto o relativo: in quest'ultimo caso è privo di barra all'inizio e indica uno spostamento rispetto al punto in cui ci troviamo.

**Redattore testi(editor)** Semplice programma che permette di modificare un file di testo. Particolarmente

utile per la configurazione dei programmi. I più diffusi, sotto Linux, sono *edit* e *gedit*.

**Root,directory** Cartella che contiene tutti i file del nostro computer. È indicata con '/'.

**Root,utente** Utente speciale, detto anche amministratore o superutente. È l'unico che può lavorare sul computer senza alcun tipo di restrizioni.

**RPM** Red Hat Package Manager. Uno dei formati per la distribuzione del software sotto Linux.

**Script** Letteralmente "sceneggiatura". È un programma, solitamente breve, composto da una serie di comandi da eseguire in sequenza.

**Shell** Il programma che gestisce l'interazione con l'utente in modalità non grafica, cioè quando si lavora su terminale. Nel mondo Linux la shell più diffusa è *bash*

**Swap** Una o più partizioni del disco possono essere preparate del tipo "Linux Swap" e sono utilizzate dal sistema come memoria aggiuntiva o "virtuale".

**Tar** Il più diffuso programma di archiviazione nel mondo UNIX e Linux. È anche l'estensione dei files creati dal programma.

# KEVIN MITNICK

*Kevin Mitnick, il CONDOR: il lato avventuroso dell'Hacking*

Kevin Mitnick è uno degli Hacker più famosi della storia dell'informatica. Mitnick non è un filosofo come Stallman, né un programmatore come Torvalds, Mitnick esplora le tecnologie e la mente umana come mai nessuno aveva fatto prima di lui. La sua immensa curiosità, la bravura e la sua spregiudicatezza lo hanno portato verso terre inesplorate, terre molto pericolose, dove alla fine è scivolato nella rete dei suoi inseguitori.

Già da giovanissimo la sua vita ispirò Wargames [1], lo storico film dell'84 che raccontava le gesta di un ragazzino capace di entrare nei sistemi informatici della NASA. Ciò che fece tra l'81 e il 95 è ormai storia e nessuno l'ha saputo raccontare meglio di Raoul Chiesa in un articolo del 1999 quando Mitnick era ancora in carcere. Per cui lascio a lui la parola per questo numero.

[1] Wargames, <http://www.imdb.com/title/tt0086567/>

## **Vola Condor, vola** di Raoul Chiesa

La storia infinita di Kevin David Mitnick, l'hacker più famoso del mondo, sotto processo negli Stati Uniti. È in carcere da quattro anni e gli è stata respinta ogni richiesta di libertà su cauzione.

## **This is the End. My only friend, the End...**

È il 14 febbraio del 1995, giorno di San Valentino. In una cittadina americana sta accadendo qualcosa che scatenerà ribellioni, dimostrazioni, appelli via rete, la nascita di siti spontanei di controinformazione, la creazione di una colletta per realizzare un fondo spese legali...

Arriva un camioncino blu, un Van, con la scritta Sprint Telecommunications. Dal Van esce un giapponese, due tecnici della Sprint, le forze speciali dell'FBI con giubbotti antiproiettile ed armi alla mano. Sembra di vedere un film di azione. Ma non è così. A Raleigh, piccolo centro nello Stato del North Carolina, stanno per arrestare Kevin Mitnick. Le conseguenze di questo arresto saranno molteplici e non solo per le comunità underground ed hacker.

Iniziata nel 1981 con l'attacco da parte di Kevin, allora diciassettenne, ai sistemi Cosmos della Pacific Bell (compagnia telefonica americana), la "caccia all'uomo" dell'FBI si conclude nel 1995, dopo 14 anni. Centosessantotto mesi di appostamenti, intercettazioni, false piste, arresti mancati. Il 14 febbraio si arriva alla conclusione di una lunga corsa, un



inseguimento interminabile attraverso le reti di mezzo mondo. Quel giorno nasce il mito di Kevin Mitnick, l'hacker più famoso al mondo.

## **Gli inizi**

Kevin nasce in California. I suoi genitori divorziano quando lui ha appena 3 anni. Nella sua adolescenza rispecchia lo stereotipo classico dell'hacker: a 13 anni è un ragazzino solitario, grassottello. Inizia con i "CB", a 8 anni è già radioamatore. Da ex-hacker, interpreto questa sua passione come una ricerca comunicativa: là fuori c'è qualcuno e lui ci vuole parlare, la distanza fisica non è più importante, a otto anni può dialogare con degli adulti che si trovano in altre città. Credo che in quel periodo sia nata in lui, forse inconsciamente, la passione per l'hacking, le reti, la comunicazione.

Nella più classica immagine americana della prima metà degli anni '70, il ragazzino brufoloso, occhialuto e ciccione ha 9 anni e vagabonda per i negozi di elettronica della sua città, prende pezzi usati, li ricicla, costruisce, assembla tecnologia trasmissiva tutto solo nella sua stanzetta, mentre i coetanei giocano a basket o "simply, they are hangin' on around the school" (cazzeggiano davanti alla scuola): diventa cliente assiduo di alcuni negozi, fa amicizia con i proprietari, forse inizia a fare qualche lavoretto da bravo teen-ager americano, ricevendo come paga dell'hardware usato.

Il ragazzino cresce, scopre i PC. Va oltre, fin da subito, scoprendo i modem: a 13 anni viene cacciato da scuola dal preside, perché entrava negli archivi degli altri istituti. Trasferisce la bolletta telefonica di un ospedale (30.000 US\$) sul conto di uno che detestava. Si battezza Condor, dal film "I tre giorni del Condor", con Robert Redford. Credo che un alias, un nickname, non sia mai stato più azzeccato: il Condor è solitario, lavora da solo, non si fida di nessuno, vive con la solitudine. E probabilmente Kevin/Condor trova nella Rete e nell'hacking una compagna ideale, un qualcosa che lo rende meno solo.

Verso la fine degli anni '70, a 16 anni, come ogni bravo ragazzo americano, prende la patente: la targa

della sua auto è X-HACKER. A 17 anni viene arrestato per la prima volta: furto di manuali informatici. Immagino Kevin, andatura insicura, occhiali da vista spessi, andare a cercare manuali per imparare, per capire dei sistemi informatici ai quali non poteva accedere, a cercare l'informazione. Io facevo trashing (letteralmente, rovistare nella spazzatura; farlo davanti alle sedi di aziende di informatica, università e compagnie di telecomunicazione produce spesso informazioni molto riservate), per trovare i manuali della Digital e imparare a programmare su VAX/VMS.

Seguono altri arresti, nell'83, nell'87 e nell'88, sempre per reati informatici. Un giudice di Los Angeles, la signora Mariana Pfaelzer, lo mette in carcere emettendo una condanna superiore a quella richiesta dal D.A (District Attorney, il Pubblico Ministero). Prima di farlo uscire dall'aula, gli dice: "Questa è l'ultima volta che fa una cosa simile, signor Mitnick". Indubbiamente una frase profetica. Viene successivamente rilasciato, ma gli viene imposto il divieto di svolgere lavori che richiedano l'uso di un personal computer.

Oggi, nel 1999, ad alcuni anni di distanza, quello stesso giudice dovrà decidere se le richieste di risarcimento - presentate da multinazionali informatiche e pari ad un totale di ben ottanta milioni di dollari (sì, avete letto bene: hanno chiesto 80.000.000 US\$ di danni al Signor Kevin David Mitnick) - dovranno essere soddisfatte o meno. Buona fortuna, giudice Mariana.

Le aziende che hanno richiesto il risarcimento sono la Motorola, la Fujitsu, la Nokia, La Sun Microsystems, la Novell, la Nec. Il solo utile netto della Motorola nel 1995 è stato di 22.247.000.000 di dollari.

Con che coraggio queste aziende chiedono ottanta milioni di dollari ad un carcerato sotto processo, il quale non ha fatto altro che copiare delle informazioni per propria cultura personale, senza rivenderle, modificarle o distruggerle?

## Il "Condor" vola troppo in alto

Anni '90. Kevin è cresciuto. E' sempre più Condor. E' un fantasma, come scrissero in seguito. Non esiste. Vive dirottando i propri conti su altre utenze. Gira gli States, notebook e cellulare modificato. Pone molta attenzione durante le connessioni, cambia spesso numeri telefonici, appartamento. Si sposta di continuo. Esplode, probabilmente, il suo odio verso le "Big Companies": IBM, Digital, Sun Microsystems, Fujitsu. Tutte hanno dei segreti da custodire. Il Condor cerca la libertà d'informazione. Vuole la verità, vuole i bug, i famosi difetti, errori di programmazione compiuti dalle software house, per poter accedere ai sistemi informatici protetti.

Dalla prima metà degli anni '90, sino al suo arresto, Kevin cresce ancora. E' molto attirato dai sistemi VAX della Digital: sono i soli a non avere pratica-

mente bug, a non essere "sfondabili". Allora il Condor agisce, silenzioso. Utilizza Social Engineering, una tecnica hacking per carpire telefonicamente informazioni spacciandosi per un'altra persona, un collega di una filiale. Ottiene tutto quello che vuole. Viola il sistema di un Internet Provider inglese. A quel sistema è abbonato, come utente regolare, un consulente della Digital.

E' stato tra i creatori del VMS, il sistema operativo proprietario dei VAX Digital, ed ora effettua consulenze alla Digital sulla sicurezza. I bug ci sono. Vengono scoperti da quest'uomo. E Kevin gli spia le e-mail. Apprende i segreti più segreti della Digital, il suo intento era quello: se non posso ottenere le informazioni in un modo, le ottengo in un altro.

L'FBI è sui suoi passi. Lui lo sa. Spia le comunicazioni tra la sede centrale dell'FBI e gli agenti dislocati, i quali lo stanno cercando per mezza America. Non appena l'FBI dà l'ordine "ok, andate a prenderlo", lui sparisce. Li prende in giro. Falsifica le comunicazioni. E' un'ombra sulla Rete, nessuno sa dove sia fisicamente.

Kevin nel frattempo è entrato ovunque: multinazionali, società d'informatica, agenzie governative. Entra e copia: progetti, piani, budget, business plan, contatti, consulenze esterne. Non vende nulla, non baratta, non cancella: apprende, impara, conosce. Per lui la conoscenza è importante. Sa come funzionano le cose. Capisce che il nuovo business sta partendo: telecomunicazioni, telefonia cellulare, satellitare, pay-Tv. I bit avanzano, l'analogico scompare. Kevin lo sa. Forse inizia a capire il potere che ha in mano.

Lo capiscono anche altre persone. Kevin ha accesso ad informazioni riservatissime, e questo dà molto fastidio alle multinazionali. Le lobby USA si muovono, l'FBI lo inserisce ufficialmente tra i "Top Wanted", come per i peggiori criminali.

L'Fbi ha capito, le multinazionali anche, le lobby hanno provveduto: mancano i mass-media.

Come per magia, appare un articolo sulla prima pagina del New York Times, il 4 luglio 1994: racconta dell'esistenza del Condor. Kevin diventa un personaggio. Ma è sempre più braccato.

John Markoff, l'autore dell'articolo, fa di tutto per incontrarlo. Corrompe alcuni suoi "amici", collabora con l'FBI per incastrarlo.

## Inizia la guerra: Davide e Golia

Nel dicembre del 1994, appare un messaggio sul computer di Tsutomu Shimomura, nippo-americano, super esperto di sicurezza, consulente del Governo USA.

Non c'è scritto molto, solo un "Found me: I am on the Net". Trovami, sono sulla Rete, gli dice Kevin.

La sfida ha inizio. Per la prima volta, le armi sono diverse: è una caccia all'uomo on-line. Le maggiori compagnie di telecomunicazione americane collabo-



rano con l'FBI. La Sprint Corporation fornisce manuali, schede, tecnici specializzati. Il Condor è braccato.

Kevin ha utilizzato, tra i primi al mondo, la tecnica dell'IP-spoofing, nel dicembre del 1994, per attaccare i server di Shimomura con sede a San Diego. Shimomura commenta questa tecnica ad una conferenza americana (CMAD), nel gennaio del 1995. Pare dunque che inizi, sin da subito, lo sfruttamento del Condor, delle sue tecniche, della sua abilità, del suo stile e delle sue competenze.

Markoff scrive altri articoli, accusa Kevin, lo dipinge come il "criminale". Kevin ama, come molti hacker, la stampa. Vuole dire la sua. Non accetta giudizi senza poter ribattere. Contatta Jonathan Littman. Arriva a chiamarlo tre volte al giorno. Forse, da questo momento in poi, il Condor perde la sua freddezza, la sua lucidità, i suoi attenti calcoli. Compie degli errori. Deve cambiare città sempre più spesso. Non capisce le motivazioni di tanto clamore attorno al suo caso. Discute con il giornalista "buono" di hacking, di politica, di tecnologia, di donne, di costume. Littman lo definirà "una mente esplosiva, incontrollabile, incredibilmente potente".

Littman sbaglia: come in ogni "grande rovina", all'origine c'è un errore. Parla con Markoff, si confida. Gli rivela dove si trova Kevin. Lo rivela al collega giornalista, all'amico John. Markoff, il quale informa immediatamente Shimomura. Il cerchio si stringe. Kevin crede di essere tranquillo. Ha fiducia nel suo confidente. Una volta un hacker mi disse: "Trust no 1: non fidarti mai di nessuno". Kevin avrebbe dovuto incontrare quell'hacker, forse sarebbe andata diversamente. John Markoff o Shimomura - ma è indifferente - informano l'FBI. Il Condor sta per cadere.

Torniamo al 14 febbraio del 1995. Kevin viene arrestato. Non uscirà mai più dal carcere. Amici, conoscenti, hacker, amanti della libertà d'espressione, dell'open source, della libera comunicazione, anarchici, hanno fondato un sito web, <http://www.kevinmitnick.com>.

Sull'home page c'è un contatore. Non è il classico counter per gli accessi. Scorre veloce, di continuo. Purtroppo non testimonia una cosa allegra, come l'alto numero di visitatori. Le cifre scorrono, a rotazione, e contano:

**4 ANNI, 4 MESI, 22 GIORNI, 13 ORE, 8 MINUTI, 33 SECONDI**  
**4 ANNI, 4 MESI, 22 GIORNI, 13 ORE, 8 MINUTI, 34 SECONDI**  
**4 ANNI, 4 MESI, 22 GIORNI, 13 ORE, 8 MINUTI, 35 SECONDI**

La frase sopra recita: "Kevin Mitnick è stato imprigionato dal Governo Americano, prima della sentenza, da:..."

### **Vola mio Condor, vola sempre più in alto.**

Kevin Mitnick è la punta di un iceberg, a mio parere. E' diventato il capro espiatorio. Gli Usa, le multinazionali, le aziende d'informatiche, vogliono una vittima. Un caso esemplare. Una condanna altrettanto esemplare. Vogliono un esempio, un precedente.



Negli Stati Uniti la detenzione media per omicidio colposo è di 3 anni: Kevin è dentro da 4; non ha ancora avuto un processo; il numero dei capi d'imputazione assegnatigli farebbe impallidire Al Capone; è stato messo in isolamento per otto mesi; ci sono migliaia di persone al mondo che lottano per i suoi ideali, ma lui non lo sa; gli è stata rifiutata ogni richiesta di libertà su cauzione; gli furono sequestrati computer, modem, persino la radio: avrebbe potuto modificarla per comunicare con l'esterno, dissero.

L'amato Mr. Shimomura, insieme al degno compare Markoff (sembrano il gatto e la volpe), nel frattempo, hanno incassato un anticipo di 750.000 US\$ per il libro che hanno scritto, "Sulle tracce di Kevin" (edizione Sperling & Kupfer). Recentemente hanno venduto i diritti per il film, e Kevin marcirà in una prigione americana, e continua a non sapere cosa gli succede intorno, cos'è diventato il Web - quelle "3 W" che tanto ci stanno cambiando la vita - che lui ha contribuito a rendere più sicuro.

Il ragazzo che voleva il "cyber-world" libero, gratuito ed accessibile a tutti, l'uomo che voleva dei sistemi sicuri, l'uomo che ha ispirato due generazioni di hackers, guarda dalle sbarre i fili telefonici, immagina i segnali satellitari, le reti GSM a 1800 MHz che spingono i segnali. Io telefono, e la "centrale" sa dove sono fisicamente, voi vi collegate al Web, e pagate la connessione, l'abbonamento, gli scatti, i megabytes scaricati. L'hacker che voleva l'informazione come un diritto innegabile dell'uomo, l'informazione gratuita, vera e totale, il condor che voleva volare in alto, è stato rinchiuso, è stato ridotto al silenzio.

Non posso dire altro, a voi che leggete, se non farvi riflettere su una cosa: se i diritti costituzionali possono essere messi da parte nella Grande America per Kevin Mitnick, cosa vi fa pensare che non sarebbe lo stesso con voi al suo posto?

E non posso augurare altro, al buon Kevin Mitnick, se non di volare. Vola mio Condor, vola sempre più in alto.

Qui finisce l'articolo di Chiesa, ma al momento della stesura Kevin Mitnick era ancora in carcere ma oggi l'Hacker più famoso del mondo che cosa fa? >

## Kevin Mitnick oggi

Kevin Mitnick e' uscito dal carcere americano nel gennaio 2000 ma col divieto assurdo di usare Internet fino al 21 gennaio 2003. Attualmente Kevin è CEO dell'azienda di consulenza e sicurezza informatica "Mitnick Security Consulting LLC"[2], famosa anche per aver subito un defacement da parte di alcuni cracker desiderosi di facile fama.

Nel 2000 esce nelle sale un film sulla sua storia, Operation Takedown [3], diretto da Joe Chappelle e scritto dai suoi inseguitori. Film molto criticato in quanto orientato a screditare l'Hacker focalizzando l'attenzione sull'investigatore che lo insegue in giro per l'America.

Mitnick ha anche pubblicato due libri finora: uno sull'ingegneria sociale che ha scritto insieme a William Simon dal titolo "The art of deception", uscito in Italia nel 2003 con il titolo "L'arte dell'inganno" (Feltrinelli, ISBN 8807170868), ed uno del 2005 dal titolo "The Art of Intrusion - The Real Stories Behind the Exploits of Hackers, Intruders & Deceivers", uscito in Italia nel maggio 2006 con il titolo "L'arte dell'intrusione" (Feltrinelli, ISBN 8807171228).

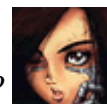
Il Condor continua a volare ....



Una foto recente di Mitnik

[2] <http://www.mitnicksecurity.com/>

[3] Operation Takedown, <http://www.imdb.com/title/tt0159784/>



## RADIOHEAD – TESTA DI RADIO

Era una sera alla radio di qualche anno fa, quando beccai un concerto su radiodue di una band annunciata dallo speaker come: "Radiohead - Testa di radio!".

Le due ore del concerto live dei Radiohead sono finite su un nastro bootleg, che avrò conservato da qualche parte dopo averlo consumato quasi fino alla distruzione.

Questa mattina, a distanza di anni da quella magica serata, ho aperto il browser sull'indirizzo [www.radiohead.com](http://www.radiohead.com). La band ha sempre avuto il pallino dell'informatica e di internet, riuscendo sin dai tempi pionieristici della rete a parlarne come nessun altro il linguaggio e costruire siti originali e ben fatti. Le teste di radio hanno deciso di rompere gli indugi e distribuire il loro nuovo album interamente in digitale, secondo le regole del software "shareware". Il prezzo di copertina non è fissato, ma si effettua una libera donazione per scaricare le dieci canzoni in mp3.

La modalità di scambio Shareware del software funzionava all'inizio proprio così. Al posto del prezzo, il software riportava un messaggio di cortesia, "se trovi questo programma utile per i tuoi scopi, per favore fai una donazione all'autore" seguito dalle modalità di versamento della donazione.

Per una manciata di sterline (i britannici hanno snobbato l'euro in uno storico referendum), ho scaricato un file zip contenente le tracce mp3 dell'album "in raimbows", che potrò liberamente ascoltare su tutte le fonti digitali che possiedo: computer, stereo CD, telefonino, lettore portatile, autoradio e tutto quanto fa parte del corredo di aggeggi mediatici di un perfetto hacker technopervertito.

I Radiohead avevano deciso di utilizzare la modalità di file sharing dietro donazione soprattutto per evitare il meccanismo di intermediazione delle avide major, riunite nella famigerata lobby RIAA.

Le multinazionali della musica e dell'entertainment, difatti, continuano a legare la produzione musicale al supporto (il cd), condannando l'uso della duplicazione personale e della condivisione.

Un atteggiamento avido, cieco e legato alla conservazione delle proprie posizioni di privilegio, la maggior parte delle volte sono le case discografiche a ritagliarsi la fetta più grossa dei ricavi delle vendite del supporto musicale, lasciando all'artista una misera percentuale.

E' vero anche che i Radiohead hanno dovuto fare un mezzo passo indietro, distribuiranno comunque il loro CD secondo i canoni tradizionali, e potremo acquistare il supporto Cd con tanto di copertina e custodia nei negozi di musica. Il motivo è legato alla promozione e ai passaggi in radio e nelle videoTV, ancora assoggettate alla industria discografica.

Il loro esempio resta comunque una pietra miliare che segna l'avvento del futuro della distribuzione musicale, rivoluzionaria quanto l'apertura dello store musicale di Apple qualche anno fa.

L'esempio dei radiohead sarà seguito da altre band, il primo nome certo è quello dei Nine Inch Nails, altra band alternative, ma si vocifera che il prossimo lavoro di Madonna sarà autoprodotta e seguirà la stessa politica di vendita.

E se a farlo è la numero uno Madonna, ci sono buone speranze che le cose nella industria musicale e dell'intrattenimento digitale debbano per forza cambiare.

## FREE HARDWARE FOUNDATION



Da chi dovrebbe ricevere ordini il nostro computer? La maggior parte della gente ritiene che il computer dovrebbe obbedire all'utente, non a qualcun altro. Ma così non la pensano tutti dato che con il progetto denominato "informatica fidata" (Trusted Computing), le grandi aziende mediatiche, incluse l'industria cinematografica e quella musicale, insieme alle solite aziende informatiche già abituate a fare ciò quali Microsoft e Intel, stanno cercando di fare in modo che il computer obbedisca a loro anziché all'utente.

Contro tutto ciò da anni una delle più grandi associazioni americane di appassionati di elettronica sta ultimando la definizione di una licenza open source specifica per l'hardware. A sostenerla c'è Bruce Perens, guru storico dell'open source.

Open Hardware License (OHL) 1.0 è una licenza che adotta molti dei principi legali alla base delle equiparabili licenze software, quali il copyleft di GNU, e le applica alla documentazione e alle specifiche tecniche dei componenti hardware, come schemi di progetto, diagrammi, file CAD/CAM ecc.

Ad affiancare tutto ciò è nata da pochi anni la Free Hardware Foundation che ha tra i suoi punti cardini:

1) Il difendere e diffondere i valori dell'universalità della rete internet quale luogo di libertà, di fratellanza e di uguaglianza.

2) La libera disponibilità degli strumenti tecnologici e della rete, affinché l'evoluzione della Comunicazione e della Informatizzazione siano opportunità di emancipazione e promozione umana

3) Difendere i diritti e le libertà digitali.

Principi lodevoli su cui in molti ci basiamo comunque il tentativo di una licenza open source hardware non è certo nuova vi sono i casi del:

Progetto Chumby che non è semplice da definire: parte come radiosveglia digitale, di cui ha ereditata vagamente la forma, ma con due porte usb, connessione wireless, un robusto processore a 266 mHz, 32 mb di ram e soprattutto la possibilità di installarci dei "widget" programmati in Flash, capite bene che rappresenta qualcosa di assolutamente diverso: questo device potrà essere usato ANCHE come sveglia, ma usando mp3 e radio online; si collegherà alla rete per mail e download di news, quotazioni di borsa e Rss;

lo potrete usare come cornice di foto (con Flickr) e con centinaia altre microapplicazioni ; ma soprattutto è completamente hackerabile....

Progetto Neuros OSD è il primo Media center basato su Linux, e già questo la dice lunga.

Fa tutto quello che fanno i media center ovvero registra la televisione, ma anche una serie di altre cose interessanti tra cui digitalizzare gli home movie, convertire video da moltissime sorgenti e fare uno screencapture di un video game. Insomma fa le stesse cose di Windows Media Center e l'Apple Tv, ma può essere completamente programmato dall'utente. Ma la novità vera è che non solo può essere hackerato, ma la Neuros invita a farlo offrendo dei soldi agli hacker !



Clumby la radiosveglia del futuro



Neuros OSD il primo media center basato su Linux

## E ORA GIOCHIAMO

### Nuova sezione dedicata ai giochi, questo mese: CUBE

Si discute ormai da tempo sulla scelta, fatta dai due colossi del mercato delle schede video (ATI ed Nvidia), di rilasciare i driver dei propri prodotti anche per Linux.

Addirittura AMD/ATI ha deciso di rilasciare i driver in maniera open source.

Questa scelta, che è la dimostrazione dell'enorme bacino d'utenza di cui gode ormai il "Pinguino", avrà sicuramente un impatto positivo sulle prestazioni dei processori grafici in ambiente Linux. Fino ad oggi ci sono state grandi differenze di prestazioni e di riconoscimento hardware rispetto al mondo Windows, dovute appunto al rilascio dei driver esclusivamente per sistemi operativi Microsoft.

Il codice aperto dei driver ATI consentirà alla community Linux di colmare il gap prestazionale in breve tempo e, quindi, noi di Hax, abbiamo deciso di aprire una rubrica dedicata interamente ai videogames che rappresentano sicuramente uno dei modi migliori per sfruttare a pieno le enormi potenzialità delle schede video presenti nella maggior parte dei computer (desktop e non) degli utenti del "Pinguino". Da non moltissimo tempo sono comparsi molti giochi open source anche per il nostro sistema operativo preferito, alcuni dei quali con un'ottima grafica 3D creati sfruttando le librerie XGL ed OpenGL. Possiamo dire quindi che è finito il tempo in cui gli utenti GNU/Linux erano costretti alla politica non corretta dell'emulazione, tramite wine o Cedega, dei videogiochi che erano pensati esclusivamente per Windows.

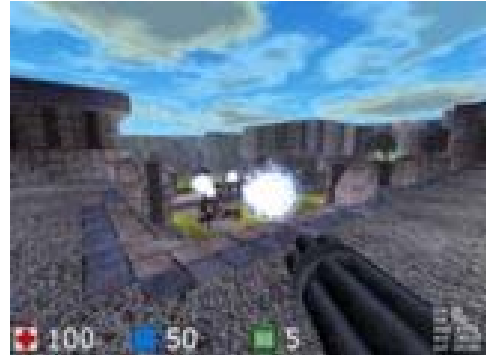
In questo numero consigliamo Cube, un videogame in pieno stile Doom3 ed Unreal Tournament.

Cube si può giocare in modalità singleplayer e multiplayer grazie al CubeEngine, un motore che consente di giocare in rete in multiplayer.

Cube non è l'unico gioco basato sul CubeEngine, ma ci sono altre edizioni di questo strepitoso first person shooter game che si differenziano per i landscape (gli ambienti in cui avvengono le sfide).



Screenshot di gioco



Screenshot di gioco

La grafica è fluida ed a dir poco strepitosa, come si può vedere dagli screenshots, ma non è l'unica cosa che colpisce di questo videogame.

La velocità di risposta ai comandi dati dall'utente è sbalorditiva e la longevità è garantita dall'imprevedibilità di ogni sfida (soprattutto quando si è in multiplayer) e da oltre 100 mappe nelle quali le battaglie sono sempre diverse.

L'intelligenza dei nemici guidati dal computer nella modalità singleplayer è sicuramente elevatissima ed alcune volte si rischia di rimanere accerchiati da queste orribili creature, pronte a tutto per avere la vostra testa!

Consigliamo a tutti di provare questo gioco (ma non se siete troppo occupati, altrimenti trascurerete i vostri impegni rimanendo incollati a Cube per ore ed ore!) che è liberamente scaricabile dalla rete.



Screenshot di gioco

L'indirizzo del sito ufficiale è:

<http://www.cubeengine.com/index.php4>

qui potete trovare anche le altre versioni di Cube.

## Gestione del Territorio "LIBERO"

Il panorama del software Floss dispone di uno dei migliori software per la gestione del territorio. Il software in questione è Grass (Geographic Resources Analysis Support System) che è un Gis (Geographic Information System) cioè una famiglia di software usati per la gestione e l'analisi del territorio.

Il software Grass è inizialmente (1982-1995) sviluppato dall'U.S. Army Construction Engineering Research Laboratories (US-Cerl) per scopi prettamente militari, come la gestione e l'uso del territorio per scopi militari ma poi il suo uso si è "spostato" verso usi "civili" come la gestione e l'uso del territorio.

Viene usato da Enti statali e commerciali come la NASA, il NOAA, l'USDA, il DLR, il CSIRO, il National Park Service statunitense, l'U.S. Census Bureau e l'USGS questo solo per citare alcuni enti statali americani che lo usano per non contare gli enti commerciali che lo usano.

In Italia viene usato in vari corsi universitari e in varie Università. Per citare solo alcuni corsi ed Università: corso di Fotogeologia (Laurea in Scienze Geologiche dell'Università di Parma), Cartografia Numerica (Facoltà di Ingegneria del Politecnico di Milano) e Fotogrammetria, Cartografia numerica e GIS e del corso integrato di Pianificazione ecologica del territorio forestale, analisi e valutazione ambientale e cartografia numerica (Facoltà di Ingegneria dell'Università di Trento).

### Caratteristiche di Grass

Grass è un software Gis che supporta sia i file di tipo raster che di tipo vettoriale ed il processamento delle immagini per la produzione di elaborati geografici.

Grass per fare ciò usa circa 350 programmi per la renderizzazione di immagini e mappe sia a monitor che su carta, per manipolare file di tipo raster e di tipo vettoriale e per processare immagini multi spettrali e creare, gestire e recuperare dati spaziali.

Grass usa la classica struttura Unix basata sulla shell ma dispone anche di una comoda interfaccia grafica che facilita le operazioni. Grass riesce ad interfacciarsi con stampanti, plotter e digitalizzatori ma anche con i database per recuperare, gestire e manipolare archivi che contengono dati geografici.

### Programmazione di Grass

Il codice sorgente di Grass è rilasciato sotto licenza GNU GPL e ciò consente un suo rapido sviluppo oltre ad una risoluzione degli errori di programmazione ma la presenza del codice sorgente, scritto in C con parti in Fortran, consente anche all'operatore

esperto anche in programmazione di poter inserire delle funzioni per lui importanti ma non ancora presenti.

Grass è attualmente supportato e potenziato da un gruppo di sviluppo presso la Baylor University e l'ITC-IRST di Trento con il centro principale di sviluppo situato presso quest'ultimo.

Piattaforme supportate da Grass

Grass viene sviluppato, proprio perché programmato in gran parte in C, per numerose piattaforme, giusto per citarne alcune:

Intel x86  
 Motorola PPC  
 SGI MIPS  
 Sun SPARC  
 Alpha AXP  
 HP PA-RISC  
 CRAY  
 altre.

### Sistemi supportati da Grass

Grass non supporta solo diverse architetture hardware ma anche diversi Sistemi Operativi. Lo sviluppo iniziale di Grass era, ma possiamo dire è, in ambiente Unix ma da qualche tempo Grass viene sviluppato anche per Windows tramite le librerie Cygwin, che forniscono librerie e applicazioni standard Unix, e questa soluzione consente l'uso di Grass anche su Windows.

Solo per citare alcuni Sistemi Operativi per il quale Grass viene sviluppato:

Linux/Intel  
 Linux/PowerPC  
 Solaris/SPARC  
 Solaris/i86  
 SGI IRIX  
 HP UX  
 Mac OS X (Darwin)  
 IBM AIX  
 BSD-Unix varianti  
 FreeBSD  
 CRAY Unicos  
 iPAQ/Linux  
 altre piattaforme Unix sia a 32 bit che a 64 bit  
 Ms Windows NT tramite Cygwin.

Come si può notare per la sua alta compatibilità a livello di sorgenti Grass gira su gran parte dei Sistemi Operativi \*nix, sia versioni commerciali che libere, ma gira anche, tramite le librerie Cygwin sviluppate dalla Red Hat, su Ms Windows.



Logo di Grass

## Importazione ed esportazione di dati

Grass supporta molti formati di file, sia in importazione che in esportazione:

2D raster data

3D raster data

topological vector data (2D, esteso anche al 3D)

point data

In dettaglio:

Raster: ASCII, ARC/GRID, E00, GIF, GMT, TIF, PNG, ERDAS LAN, Vis5D,

SURFER (.grd) ...

Questa compatibilità si ha usando la libreria GDAL (Geospatial Data Abstraction Library, `r.in.gdal`) con cui si possono usare molti formati come i formati come quelli CEOS (SAR, LANDSAT7 etc.). Image (satellite and air-photo): AVHRR, BIL/BSQ, ERDAS LAN, HDF, LANDSAT TM/MSS, NHAP aerial photos, SAR, SPOT, ...

Vector: ASCII, ARC/INFO ungenerate, ARC/INFO E00, ArcView SHAPE (with topologic correction), BIL, DLG (U.S.), DXF, DXF3D, GMT, GPS-ASCII, USGS-DEM, IDRISI, MOSS, MapInfo MIF, TIGER, VRML, ...

Sites (point data lists): XYZ ASCII, dBase

Le capacità di Grass vanno dall'Analisi Spaziale, alla generazione di mappe, alla visualizzazione dei dati in 2D, 2.5D e 3D, alla generazione di dati tramite la modellizzazione, collegamento a DBMS (PostgreSQL, e altri tramite ODBC, ...), alla memorizzazione di dati.

## Conclusioni

In conclusione Grass rappresenta un valido Gis che può essere usato sia per applicazioni commerciali che di ricerca. Uno dei problemi, risolvibile, che gli utenti di altri Gis in ambiente Windows potrebbero trovare nell'uso di Grass, almeno nel primo impatto, è dato dal suo intrinseco sviluppo in un ambiente operativo di tipo Unix, tra l'altro primeggiante in ambito di ricerca e di sviluppo, che privilegia l'uso della riga di comando, chiamata nei Sistemi Operativi di tipo Unix "shell" (CLI, Command Line Interface). Questo, come accennavo prima, non è un grosso problema visto che si stanno sviluppando comode interfacce grafiche (GUI, Graphical User Interface) che ne rendono l'uso abbastanza facile a patto di, come per ogni programma tecnico, conoscere ciò che si sta facendo.

<http://grass.itc.it/>

<http://it.wikipedia.org/wiki/GRASS>

[http://en.wikipedia.org/wiki/GRASS\\_GIS](http://en.wikipedia.org/wiki/GRASS_GIS)

[http://www.ing.unitn.it/~grass/docs/tutorial\\_60/italiano/generalita.htm](http://www.ing.unitn.it/~grass/docs/tutorial_60/italiano/generalita.htm)

## SOFTWARE LIBERO, PUBBLICHE AMMINISTRAZIONI E FORZE ARMATE

### ADOZIONE E PASSAGGIO AL SOFTWARE LIBERO

#### Cos'è il Software Libero (Brevemente)?

È un fenomeno che riguarda la libertà della conoscenza, non il suo valore economico. Non è importante che il software sia gratuito, ma invece che la sua struttura e il suo funzionamento siano pubblici. Con il software libero l'utente può eseguire, copiare, distribuire, studiare, cambiare e migliorare il software. Più precisamente, sono riconosciuti quattro tipi di libertà per gli utenti del software:

Libertà di eseguire il programma, per qualsiasi scopo (libertà 0).

Libertà di studiare come funziona il programma e adattarlo alle proprie necessità (libertà 1).

Libertà di ridistribuire copie in modo da aiutare il prossimo (libertà 2).

Libertà di migliorare il programma e distribuirne pubblicamente i miglioramenti, in modo tale che tutta la comunità ne tragga beneficio (libertà 3).

#### RAGIONI:

##### LATO TECNICO/SCIENTIFICO/SICUREZZA

Le Forze Armate di ogni paese (polizia, forze speciali, esercito, servizi di sicurezza nazionale, etc.) sono molto attenti nella scelta del sistema operativo in uso da loro. Vi sono svariati motivi di questa scelta. Uno dei motivi principali è quello di poter avere il pieno controllo sul proprio sistema informativo, cioè il controllo e la piena autonomia per lo stabilimento della sicurezza la tutela dei dati riservati. Con un sistema operativo proprietario e/o applicativi proprietari ciò non è possibile, per il semplice motivo che, il sistema operativo in questione ha un codice chiuso e all'utente finale (le forze armate nel nostro caso) viene consegnato solamente un eseguibile per far funzionare il proprio sistema, quindi l'utente e/o ente in questo è all'oscuro di tutto ciò che accade quando viene eseguita un'operazione su un supporto magnetico. Non vi è alcuna garanzia di cosa ci possa essere all'interno del sistema operativo e dell'applicativo in uso, vista l'impossibilità di avere il codice sorgente per visionare e autotutelarsi. È un concetto basato sulla fiducia, cioè devi fidarti è basta (trusted computing). Uno stato non può affidare i propri dati ad una multinazionale (tra l'altro non appartenente a tale stato) e

fidarsi ciecamente.

Con un sistema operativo e/o applicativi di piattaforme liberi invece (software libero e software open source) il concetto è diverso. L'Ente che adotta tale piattaforma ha tutto l'applicativo con i codici sorgenti a disposizione. Gli è concesso il diritto di esaminare il contenuto dell'applicativo e verificarne l'affidabilità. Quando parliamo dell'affidabilità ci riferiamo a due significati differenti:

l'affidabilità e l'efficienza per il funzionamento del software e/o applicativo

l'affidabilità per la tutela e sicurezza dei dati

Nel caso delle forze armate, quest'ultimo elemento è fondamentale. Nessun ente governativo affida i suoi sistemi propri a delle multinazionali per la gestione.

Il software libero può essere adottato e adattato all'interno delle forze armate trasformandola in un sistema di proprietà dell'esercito per la tutela dei dati e per garantire la sicurezza dei dati in questione (vedi le 4 libertà). D'altronde possiamo vedere gli esempi e realtà simili della maggior parte delle forze armate presenti nel mondo.

##### LATO ECONOMICO

Dal punto di vista economico il passaggio dal software proprietario a quello libero ha l'obiettivo di far risparmiare milioni di euro all'anno. Ciò è possibile ricorrendo al software libero. La migrazione permetterà alle forze armate (o altri enti e amministrazioni del governo) di svincolarsi dai costi ricorrenti per le licenze, conseguendo inoltre risparmi a fronte di una prolungata vita utile dell'hardware impiegato.

La migrazione può essere assistita dall'Hacklab Co-senza in collaborazione con la comunità del Software Libero e realtà presenti sul territorio nazionale, promotori del software libero, con costi iniziali e non ricorrenti sui grandi cifre (come quelli affrontati attualmente per il settore ICT) per la formazione del personale e per gli adattamenti informatici:

- gli investimenti dell'amministrazione, quindi, vengono rivolti all'arricchimento professionale del proprio personale, piuttosto che alle licenze di impiego di software importato dall'estero.



"E' più difficile cambiare le abitudini dei funzionari che il sistema operativo".

La migrazione si realizzerà quindi in due fasi:

nella prima, verrà diffuso l'impiego di OpenOffice;

quindi, si passerà al cambio del sistema operativo, una volta che il personale si sarà avvicinato agli strumenti liberi.

L'adozione di software aperto risponde non solo alle raccomandazioni emerse in europa, ma anche ad indirizzi emessi dal ministero dell'amministrazione pubblica, che propone l'open source per aiutare la libertà di scelta, la protezione degli investimenti, il miglior rapporto rendimento/costo e la garanzia di comunicazione ed interoperabilità.

### Analisi microeconomica sulla migrazione a GNU/Linux

Software libero contro software proprietario: Un tentativo di analisi microeconomica della migrazione a GNU/Linux in ambito aziendale.

La migrazione da un sistema operativo proprietario, a quello libero per eccellenza, cioè GNU/Linux, offre al giorno d'oggi numerosissimi vantaggi, purtroppo ancora ignorati dalla maggior parte degli utenti e dei dirigenti aziendali.

Dal punto di vista etico, il raggiungimento della totale libertà informatica produce un profondo appagamento per gli animi più "romantici", dal punto di vista tecnico invece, un sistema "UNIX like" facilmente reperibile è una manna dal cielo per gli smanettoni, e infine i più pragmatici possono essere decisamente interessati al risparmio connesso a una scelta del genere, risparmio che può raggiungere anche una ragguardevole consistenza quando chi compie la migrazione è un'azienda.

Però purtroppo, dato che l'attenzione dei sempre più numerosi sostenitori dei movimenti free software e open source, sembra quasi completamente rivolta verso implicazioni etico-filosofiche o prettamente tecniche, pochi si occupano di produrre stime verosimili, dei vantaggi economici derivanti da un radicale cambiamento nelle scelte riguardanti i mezzi informatici.

Facendo una ricerca in rete, si trovano con molta fatica, poche e frammentarie informazioni che possano dare la spinta decisiva a chi, eventualmente incaricato da qualche azienda, stia operando valutazioni su un possibile passaggio al software libero.

Grave mancanza questa, anche perché, all'interno dei mercati sembra ci sia l'abitudine a operare nel rispetto di principi diversi da etica e filosofia, o che purtroppo questi ultimi, producano risultati non suffi-

cientemente convincenti.

Per chiunque operi nei mercati (settore pubblico e privato) l'obiettivo da perseguire dovrebbe essere sempre la minimizzazione dei costi: è risaputo che, chi non minimizza i costi rischia gravemente di diventare inefficiente.

E le aziende, terrorizzate da questa possibilità (che può significare l'uscita dal mercato), impostano i propri processi decisionali applicando un modello di scelta fondato sull'analisi di costi e benefici.

Fondamentalmente ci si attiene alla regola secondo la quale è conveniente intraprendere una determinata azione se e solo se, i benefici che ne scaturiscono sono maggiori dei costi che bisogna sostenere per intraprenderla.

A prima vista, un principio del genere sembrerebbe portare banalmente a scelte scontate: invece è quasi un'arte (peraltro ben remunerata), rintracciare tutti i costi e i benefici connessi a una qualsiasi azione, e non di rado purtroppo, si ha la dimostrazione di come un'analisi non accurata, conduca a risultati abbastanza fuorvianti, con conseguenze nefaste.

Nel nostro caso, può essere incredibilmente illuminante mettere in risalto un semplice dettaglio: il non sostenimento di un costo è perfettamente equiparabile all'ottenimento di un beneficio.

Per estensione quindi, dall'analisi dei costi che scaturiscono direttamente o indirettamente dall'utilizzo di software proprietario, si possono ricavare i benefici economici derivanti dall'utilizzo di software libero.

Il software proprietario produce costi che possono essere raggruppati in almeno sei classi (tutte completamente o in larga parte evitabili utilizzando software libero):

1. costi di licenza;
2. costi hardware;
3. costi di assistenza e manutenzione;
4. costi di protezione;
5. costi di formato e di interazione aziendale;
6. costi in termini di tempo.

La prima di queste classi di costo, quella dei costi di licenza, che di solito è l'unica menzionata nelle analisi di risparmio (ma che come si vedrà tende ad essere non per forza la più incisiva), nasce dal fatto che, la scelta del software proprietario, comporta il pagamento di licenze di utilizzo che raggiungono importi anche abbastanza rilevanti, appena il numero di elaboratori serviti diventa considerevole; questo tipo di costi è totalmente inesistente con il software libero, che può essere adoperato oltre che liberamente, anche gratuitamente.

I costi hardware invece, possono presentarsi qualora

si abbia una miglione nel software che, come spesso accade (specialmente col software proprietario) può significare una maggiore richiesta di risorse fisiche, alla quale si deve inevitabilmente far fronte con un potenziamento degli elaboratori.

Il software libero, offre in moltissimi casi un'efficientissima struttura client-server: quasi sempre si potrà rispondere a una crescente richiesta di risorse, potenziando un solo elaboratore che funzionerà da server, e facendo in modo che le macchine che cominciano a diventare obsolete, accedano ai servizi operando come client.

Non bisogna dimenticare inoltre, che GNU/Linux nasce per sfruttare appieno macchine economiche, facilmente reperibili, e con scarse risorse, e che quindi, anche un elaboratore di qualità medio/bassa può dare tranquillamente ottimi risultati in termini di prestazioni.

L'argomento dei costi di assistenza e manutenzione, genera molte considerazioni, la prima delle quali è di natura strettamente empirica, scaturisce infatti dalle esperienze personali di moltissimi utenti: un sistema GNU/Linux ben configurato non da pressoché alcun problema, il ricorso all'assistenza per malfunzionamento software può essere nella stragrande maggioranza dei casi evitato, considerazione che invece notoriamente, molto spesso non vale per il software proprietario.

Qualora, invece che il malfunzionamento, i servizi di assistenza, o più propriamente (in questo caso) di manutenzione, riguardino esigenze più complesse che non possono essere evitate (come delle modifiche software specifiche), il software libero offre ancora moltissime possibilità di risparmio proprio per la sua caratteristica fondamentale: la libertà.

Questo perché il mercato dell'assistenza/manutenzione del software libero è concorrenziale dato che non ci sono barriere all'entrata: per apprendere le conoscenze necessarie a diventare un esperto GNU/Linux infatti, non servono né il consenso, né la certificazione di una ditta che detiene diritti sul codice sorgente e il codice sorgente stesso; al contrario il mercato dell'assistenza del software proprietario è monopolistico. L'implicazione che deriva da questa considerazione dovrebbe essere tristemente nota a tutti: i prezzi dei servizi in un mercato monopolistico sono sensibilmente più alti rispetto a quelli che genera un mercato concorrenziale.

Ancora, grazie al software libero, un'azienda con richieste specifiche, potrebbe anche decidere di rispondere da sé alle proprie esigenze: dal momento che GNU/Linux mette il suo codice sorgente a disposizione per essere studiato e adattato, si può tran-

quillamente investire nella formazione di uno o più specialisti che si occupino dell'assistenza/manutenzione del proprio sistema informatico, eliminando totalmente anche i costi derivanti dal ricorso a ditte terze, specializzate nel settore. E la possibilità appena descritta, non richiede neanche grossi investimenti in manuali informatici (notoriamente costosi): la rete, è stracolma di informazioni tecniche dettagliatissime consultabili da tutti senza costi aggiuntivi.

Continuando a scendere tra i punti della lista, si arriva ai costi di protezione, che al giorno d'oggi tendono a rivelarsi in molti casi tra i più significativi per le aziende.

Rendere il più sicuro possibile il proprio sistema informatico da qualsiasi tipo di intrusione (virus, cracker ecc.) è diventato ormai, più che una condivisibile pratica, un vero e proprio obbligo.

In Italia, chiunque tratti digitalmente dati personali oggetto di tutela da parte del garante della privacy, è tenuto a porre in essere una serie di misure di sicurezza (che diventano più rigide all'aumentare della sensibilità dei dati trattati), come dispone il recente decreto legislativo 196/2003 che ha integrato la normativa sulla riservatezza (privacy).

E tra queste misure, vengono annoverati un professionale accesso multiutente ai sistemi informatici aziendali, e la protezione degli elaboratori da attacchi esterni, tramite antivirus, firewall ecc.

E' lampante che, se per adeguarsi alla normativa si ricorre al software proprietario, si ha una vera e propria impennata dei costi, dato che da sempre, la sicurezza è una merce che costa abbastanza cara, e comunque, non si avrà la certezza di raggiungere l'obiettivo sperato.

I dati parlano chiaro, la differenza tra l'incidenza che il malware ha sui più diffusi sistemi operativi proprietari e quella che ha su sistemi GNU/Linux è abissale: è di poco tempo fa la notizia che gli ultimi due anni sono trascorsi lasciando GNU/Linux completamente indenne da virus.

E un'ulteriore riprova si ha osservando il mercato dei sistemi per server internet, dove GNU/Linux ha ormai conquistato e mantiene prepotentemente il predominio, proprio grazie alla sicurezza garantita, abbinata alla stabilità (altro requisito caratteristico).

Andando oltre, e analizzando il problema della sicurezza con un po' di malizia, si può anche arrivare a ipotizzare in modo abbastanza verosimile, che i costi più impegnativi, facendo uso di software proprietario, si possano sostenere non nella messa in sicurezza del sistema, ma bensì nel ripristino dei dati, qualora il sistema fosse violato (ipotesi come detto non così remota...).

Ancora nella lista, si trovano i costi di interazione aziendale, che possono manifestarsi, quando due aziende che comincino a collaborare a uno o più progetti, si scambino dati tramite mezzi informatici. Per prima cosa, prima di procedere nel discorso, è necessario ricordare come, l'utilizzo diffuso del software proprietario, abbia negli anni, aiutato il selvaggio proliferare di formati proprietari associati a programmi specifici, a discapito degli standard che vengono adottati e rispettati nella comunità del software libero, anche perché, ottimizzano l'utilizzo dello spazio.

Non sarebbe per nulla strano che, al momento dello scambio di dati dello stesso tipo (testi, immagini, fogli elettronici, ecc.) tra aziende, ci si accorgesse di usare due programmi proprietari diversi, magari con le stesse identiche funzioni, che producono però risultati incompatibili, avendo come riflesso l'obbligo per uno dei due partecipanti di utilizzare gli stessi software dell'altro, con dispendio di denaro, oltre che con fastidiosi sprechi di tempo.

Se, infatti, come dicono molti economisti, il tempo è la risorsa scarsa per eccellenza, possiamo ipotizzare che i costi in termini di tempo, siano osservati sempre con gran riguardo dai dirigenti aziendali.

E' chiaro che anche l'instabilità di un sistema informatico (caratteristica per la quale alcuni sistemi operativi proprietari sono divenuti negli anni, tristemente famosi), sottrae tempo a chi utilizza l'elaboratore, tempo che potrebbe essere utilizzato invece in maniera produttiva.

La proverbiale stabilità di GNU/Linux consente un significativo risparmio di tempo, ma anche in presenza di problemi di instabilità, la scelta di GNU/Linux si rivelerà oltremodo vincente: ancora una volta i dati ci vengono in soccorso, GNU/Linux, garantisce un uptime (intervallo temporale necessario al riavvio della macchina a causa di crash di sistema) che ha pochi concorrenti nel mercato.

Ora, dopo aver tentato di analizzare le più rilevanti implicazioni in termini di risparmio, derivanti dall'utilizzo di software libero in ambito aziendale, non resta altro che verificare se da questo possano nascere costi che di per sé il software proprietario non produce.

A questo punto i maligni potrebbero riesumare la stra-inflazionata critica, mossa da sempre contro l'utilizzo di GNU/Linux: il possibile inasprimento dei costi per il personale, dovuto alla necessità di corsi di formazione.

Ma la proverbiale difficoltà di utilizzo dei sistemi "UNIX Like" non è più un argomento convincente, e trova la sua smentita nella miriade di distribuzioni GNU/Linux, diversissime tra loro, ma con gli stessi

punti di forza, e in grado ormai di rispondere alle esigenze di qualsiasi tipo di utente.

A margine di queste considerazioni, probabilmente l'unica difficoltà si manifesta qualora si cerchi di capire il motivo per il quale il software libero non abbia già cominciato a impadronirsi dei desktop delle aziende pubbliche e di quelle private, specialmente in un Paese come il nostro, nel quale il settore più florido dell'economia è costituito da realtà principalmente medio-piccole (e quindi con disponibilità finanziarie non elevatissime), e dove la Pubblica Amministrazione si barcamena, nella triste attesa di grossi e sistematici tagli che, prima o poi a quanto pare, diverranno improrogabili e indispensabili per il benessere del sistema, ma che presumibilmente, con una gestione più attenta (almeno dei sistemi informatici), se non scongiurati, potrebbero essere almeno ridotti nell'entità.



I loghi delle forze armate italiane con Tux e lo Gnu

## CREIAMO UN LOGO ANIMATO CON GIMP

Nella rete tutto ciò che colpisce è animato; pur considerando i principi di accessibilità, banner, loghi, immagini hanno maggiore effetto se presentano animazioni.

Obiettivo di questo articolo è realizzare un logo per il prossimo Linux Day.

Iniziamo recuperando un'immagine del nostro caro Tux.

Fonte: <http://commons.wikimedia.org/wiki/Image:Baby.tux-800x800.png>



**1**

Creare un immagine di circa 520x502 px, sfondo bianco;

- CREARE UN BORDO INTORNO ALL'IMMAGINE:
- creare un layer trasparente, **Livello | Nuovo Livello**;
- selezionare l'intera immagine con **ctrl+A** e colorarlo di nero **ctrl+**,
- ridurre l'aria selezionata di 1 px: **Seleziona | Restringi ...**
- cancella la parte selezionata con **ctrl+K**

**2**

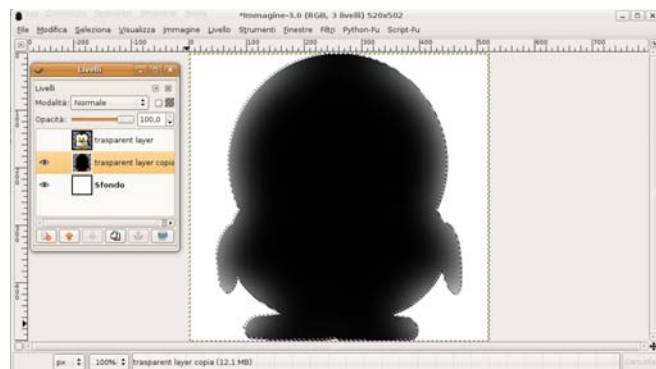
Inserire un'immagine:

CREARE UN'OMBRA INTORNO ALL'IMMAGINE: inserire l'immagine senza sfondo nell'immagine appena creata, **ctrl+v**, dopo averla copiata (**ctrl+c**) dal file di origine;

nel layer appena creato, tasto destro del mouse e ancorarlo al layer sottostante, scegliendo Ancora livello

scegliere l'oggetto inserito e duplicare il layer, inserirlo al di sopra dello sfondo

applicare Livello | Trasparenza | Alfa a selezione, valore scelto per Raggio di sfocatura: 100 px e come Metodo di sfocatura il valore RLE e spostare l'intero oggetto di 4 px a destra e in basso.



**3**

APPIATTIRE L'IMMAGINE:  
Immagine | Immagine appiattite

**4**

CREARE TESTO ANIMATO  
creare tre copie della stessa immagine appiattita;

in ognuna di esse inserire un testo;

utilizzare le guide magnetiche (Visualizza | Guide magnetiche) e la griglia (Visualizza | Mostra la Griglia) per centrare il testo nella stessa posizione in ogni layer.

il ritardo di ogni frame può essere impostato sia in fase di esportazione della gif, sia in GIMP, inserendo un commento ad ogni layer nella finestra dei layer, racchiuso tra ().

**5**

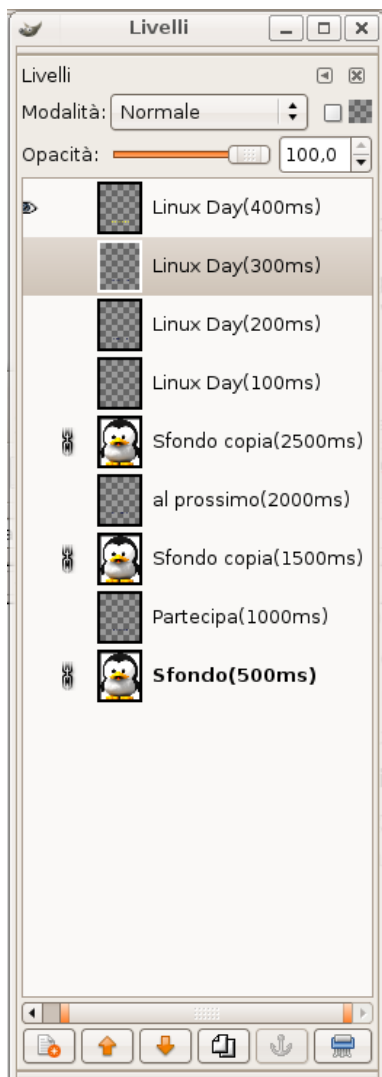
UTILIZZO DI GAP (GIMP animation package). Gap è un potente strumento per la produzione di animazioni in GIMP:

- installare GAP, in Ubuntu (`apt-get install gimp-gap`). A installazione avvenuta rieseguire `gimp` per rendere effettivi i cambiamenti;
- nella finestra principale comparirà un'altra etichetta Video e in Filtri saranno presenti due voci Filtra tutti i livelli... e Macrofiltri;

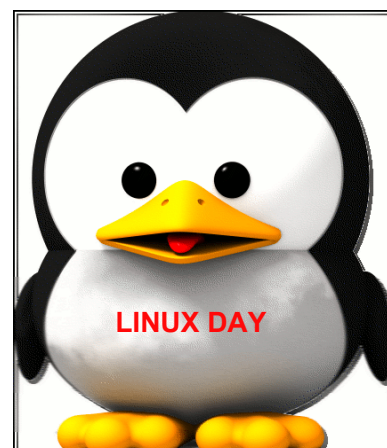
per poter applicare uno di questi script è necessario creare un'altra immagine. Questo in GIMP è molto facile, basta selezionare un layer e trascinarlo nella barra degli strumenti. Si aprirà un'altra finestra con la nuova immagine. A questo punto è necessario duplicare per 4 volte il layer. Con lo strumento di selezione rettangolare, selezionare l'area interessata dai layer. Filtri | Filtra tutti i livelli... e selezionare `plug_in_cubism`, cliccare il tasto applicare variabile.



- Nelle finestre successive verranno settati i parametri nelle diversi layer, presenti nello stack. Dopo aver testato che la funzione applicata produca il risultato sperato (Filtri | Animazione | Esecuzione), sarà necessario copiare nel senso inverso i layer nell'immagine originaria; aiutandosi con la trasparenza e i tasti cursore si potranno sistemare in posizione corretta i nuovi layer.
- Il passo successivo è esportare l'immagine in formato gif animato, durante la procedura sarà necessario inserire il tempo di latenza per ciascuna immagine e se il ciclo deve essere infinito o finito.



Ed ecco il risultato:



[http://www.gimp-italia.org/tux\\_mah.gif](http://www.gimp-italia.org/tux_mah.gif)

## TRUECRYPT

### ... Difendiamo la nostra privacy...

In questa epoca dove la nostra privacy è costantemente minacciata e la sicurezza dei nostri dati è sempre in bilico, quale migliore soluzione potrebbe esserci della crittografia?

Però spesso non è affatto comodo utilizzare un software di crittografia che è in grado di lavorare solo sui singoli file, pensiamo infatti a quanto tempo ci farebbe risparmiare l'aver un volume interamente criptato, ma utilizzabile con la praticità di un volume normale. In questo articolo andremo ad analizzare le funzionalità offerte dai quei software che si occupano di costruire e gestire dei volumi criptati "al volo" e in particolare faremo riferimento al software open source truecrypt.

Quando parliamo di crittografia "al volo" (on the fly), ci riferiamo a quei sistemi dove i dati vengono automaticamente criptati o decriptati prima di essere salvati o caricati senza il bisogno di alcun intervento da parte dell'utente...comodo no?!

Pensiamo a quanto tempo ci farebbe risparmiare il poter mettere al sicuro un file riservato facendo un semplice copia e incolla di questo dal nostro desktop al volume criptato!! Ovviamente come in qualsiasi sistema dove c'è in ballo la crittografia anche qui avremo la nostra password o il nostro file-chiave senza il quale non sarà possibile fare alcuna operazione sul nostro volume criptato, dato che ogni cosa sarà, appunto, criptata! Infatti a partire dal contenuto dei file e dei nomi delle cartelle, finendo allo spazio libero e ai meta-data ogni cosa sarà pesantemente protetta dall'algoritmo di crittografia scelto.

Veniamo ora a Truecrypt.

Truecrypt come ogni buon software di crittografia on the fly, non salva alcun file decriptato sul disco, ma le operazioni di criptaggio/decriptaggio utilizzano la RAM dove i file vengono temporaneamente tenuti fino a quando vengono utilizzati. Fino a quando il nostro volume creato con truecrypt rimarrà smontato, non ci sarà modo di identificare in questo un volume criptato, poiché non apparirà che come un file contenente dati casuali! Quindi fin quando noi non monteremo il nostro volume sarà assolutamente impossibile per chiunque dimostrare che quel file contenga effettivamente un volume criptato (anche se un'immagine jpg di 40GB possa essere molto sospetta ;)

Ma, a questo punto qualcuno potrebbe chiedersi, e se qualcuno ci obbligasse, magari con la forza, a rilevare la nostra password per montare il volume? Qui viene il bello! Perché truecrypt ha la possibilità di creare all'interno di un volume criptato un ulteriore volume criptato nascosto!! Il livello nascosto sarà accessibile con una password diversa dal livello superiore, in questo modo possiamo, per esempio, cre-

are un volume di 40 GB che contiene al suo interno un volume nascosto di, per esempio, 39GB (ovviamente non può essere più grande del volume che lo contiene), e mettere nel volume "contenitore" dei file non importanti. In questo modo anche se saremo costretti a rivelare la password, potremo svelare quella che apre solo il livello "contenitore", attraverso il quale non sarà comunque possibile rendersi conto dell'esistenza del livello nascosto, poiché anche lo spazio libero è criptato.

Ovviamente dopo aver creato il livello nascosto, stiamo attenti a non scrivere altri file sul volume "contenitore" poiché questi potrebbero sovrascrivere una porzione del nostro volume nascosto. Ma come protegge truecrypt i nostri dati? Per farlo si avvale di potenti e sicuri algoritmi: l'AES, il Serpent e il Twofish con la possibilità di combinare fra loro questi algoritmi due alla volta per farli lavorare in cascata. Le uniche minacce per i nostri dati conservati su un volume creato con truecrypt sono, la RAM e soprattutto l'ibernazione, e i filesystem journaling. Infatti ricordiamoci che i dati in ram sono contenuti in chiaro e potrebbero rimanerci lì fino allo spegnimento della macchina, quindi ricordiamoci di fare un bel riavvio dopo aver lavorato con il nostro volume criptato!

Soprattutto ricordiamo di non ibernare per nessun motivo il computer dopo che abbiamo aperto dei file contenuti nel volume criptato poiché l'ibernazione andrebbe a scrivere sul disco il contenuto della ram e quindi potrebbe permettere a qualcuno di recuperare dal disco i nostri file in chiaro! Inoltre i filesystem journaling salvano spesso alcune informazioni (per esempio l'ora di accesso ai file) che potrebbero compromettere la nostra sicurezza... teniamolo a mente! Truecrypt è semplice da usare, comodo, e soprattutto sicuro... quindi ora non abbiamo più scuse per non preoccuparci seriamente della nostra privacy e della sicurezza dei nostri dati!!

A seguire nella prossima pagina una dimostrazione pratica dell'uso di Truecrypt.

## Pratica di Truecrypt

Ora iniziamo ad esplorare il lato pratico installando truecrypt.

Scarichiamo i sorgenti tar.gz dal link <http://www.truecrypt.org/downloads.php>, poi li scompattiamo con il comando

```
tar -xvzf truecrypt-4.3a-source-code.tar.gz
```

e poi portandoci nella directory del file scompattato digitiamo i comandi

```
cd Linux
```

```
./install.sh
```

Se avete Debian per installarlo dovete digitare il comando

```
apt-get install truecrypt
```

Ora siamo pronti per incominciare ad utilizzare truecrypt.

Per farlo funzionare assicuratevi di avere il kernel 2.6.5 o compatibile e i programmi Device mapper (dmsetup, <http://sources.redhat.com/dm>) e loop device (losetup), che sono disponibili per tutte le distribuzioni linux.

Per creare un volume utilizziamo il comando

```
truecrypt -c nomevolume
```

che visualizza una serie di menu per configurarlo.

In questi menu dovrete scegliere:

il tipo di volume

il filesystem

la dimensione del volume

l'algoritmo di hash

l'algoritmo di crittazione

la password

il keyfile (non obbligatoria)

Per creare un volume nascosto ricordatevi di passare come parametro il nome di un volume già creato in precedenza.

Dopo tutto ciò avrete creato il vostro primo volume!!!

Vi starete chiedendo cos'è il keyfile??

E' un file che potete utilizzare come chiave per montare il vostro volume.

Ora per utilizzarlo dobbiamo montarlo con il comando

```
truecrypt nomevolume puntodimount
```

dove vi sarà richiesta la password (se la password è composta da spazi bianchi deve essere inserita tra virgolette es. "mia passwd").

Se un bel giorno vi viene in mente di cambiare la password dovete utilizzare il comando

```
truecrypt -C nomevolume
```

dove vi sarà richiesto di inserire prima il keyfile e la password corrente e poi il keyfile e la password nuova.

Per smontare un volume utilizzate il comando

```
truecrypt -d volumemappato
```

```
lordzen@debian: /home/lordzen
File Modifica Visualizza Terminale Scheda Aiuto
lordzen@debian:~$ truecrypt -c vol1
Volume type:
1) Normal
2) Hidden
Select [1]: 1
Filesystem:
1) FAT
2) None
Select [1]: 1
Enter volume size (bytes - size/sizeK/sizeM/sizeG): 1M
Hash algorithm:
1) RIPEMD-160
2) SHA-1
3) Whirlpool
Select [1]: 3
```

Due screenshot di shell mentre si utilizza Truecrypt

```
lordzen@debian: /home/lordzen
File Modifica Visualizza Terminale Scheda Aiuto
lordzen@debian:~$ truecrypt -c prova
Enter current keyfile path [none]:
Enter current password for 'prova':
Enter new keyfile path [none]:
Enter new password for 'prova':
Re-enter new password:
Truecrypt will now collect random data.
To enable mouse movements to be used as a source of random data,
please do one of the following:
- Run Truecrypt under administrator (root) account.
- Add read permission for your user to device /dev/input/mice.
Please type at least 320 randomly chosen characters and then press Enter:
rivubvqierubveriuvbirubvqèiurevbrüvebwiquvbiqurbvbeiqurvbqiruvncrivubeirubve
iuvnervieubrviurbeviurvbiqurebvüervneirubveirubvüerbvüqernvtbviuqweücnriuvb
eqiruvnitubtiuvrnbviuqbvyebriuqbveriuvrubveriuqvriqurbiquvntieurbvirubvèurbvr
Characters remaining: 78
fuvbwpiuefrubewrivubewèriuvberiuwbirubveèirubvbitubvweèiuvneriuvtbivèirubvetiuv
entibutnbitrubvüvbeiwutvbeiwubriuwneriuvewbrviuqbriuernvieuwrbvqi rubqeirvu
```

**HAX**  
Hax Fire eXperience



**be linux...**

www.bhioart.com



<http://hax.cosenzainrete.it>

