HAX

Hax Are eXperience

N° 1

Luglio 2007

SICUREZZA

- GPG e Crittografia

TUTORIAL

- In questo numero una guida all'uso di GIMP

PERSONAGGIO

- Linus Torvalds

GIMP vs PHOTOSHOP

Fare un paragone tra Photoshop e Gimp è un'impresa ardua, ma sembra che sia necessario visto l'interesse ad affrontare questo argomento da parte di chiunque conosca i due programmi.

Hackaserta 81100

L'Hackaserta 81100 nasce come hacklab nel 2002. Tre anni dopo, diventa associazione di promozione sociale col nome di "Hackaserta 81100" 25 Febbraio 2005.
Tutta la storia e le iniziative dell'Hackaserta in questo numero di HAX.

Perens e Stallman a ROMA

In occasione della Lection Magistralis presso la Cattedra del Prof.Arturo Di Corinto -Università La Sapienza, abbiamo incontrato Richard Stallman e Bruce Perens.
Un approfondito resoconto sui tre giorni che i Big del Free Software hanno passato nella Capitale.

HLCS HACKLAB COSENZA

http://hacklab.cosenzainrete.it/hax

H A X

HAX
HAX ARE EXPERIENCE

FRANCESCO MUSCIMARRO
“CICCIORAPTOR”
(GRAFICA, IMPAGINAZIONE)

FAUSTO NAPOLETANO
(REDAZIONE, ARTICOLISTA)

VINCENZO BRUNO
(REDAZIONE, ARTICOLISTA)

ENZO AGGAZIO
”JETPAC”
(ARTICOLISTA)

GIUSEPPE GUERRASIO
“LOBOTOMIA”
(REDAZIONE, 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)

IDREAMER
(ARTICOLISTA)

PUFFON
(REDAZIONE)

ALEXAIN
(ARTICOLISTA)

IGNAZIO FILICE
(ARTICOLISTA)

NUMERO 1

0...1 Eccoci arrivati al secondo e ultimo valore del primo bit.

Il numero 0 è stato un successo, è stato il prendere forma di un' idea, il vederla acquistare forza e sprigionare tutte le sue potenzialità. Il numero 0 ha avvicinato alla rivista molti altri nuovi collaboratori, molte nuove idee hanno trovato uno spazio dove concretizzarsi e, soprattutto, molte menti vaganti hanno trovato un porto dove fermarsi prima di ripartire.

Il numero 0 è stato questo, è stata la prova che l'antagonismo e l'underground informatico italiano è in continuo fermento e che HAX è parte integrante di tutto questo. Il nuovo valore del primo bit, l'uno, è nato dalle acque agitate del panorama informatico di questi ultimi mesi, che fra censure e incontri fra gli esponenti del mondo del Software Libero e la politica italiana. La tesi dell'HAX è quello di contribuire alla realtà filosofica, sociale ed economica italiana con la diffusione del Software Libero e la sua introduzione nelle Pubbliche Amministrazioni.

Il software libero perché, per il suo valore etico è fondamentale e per la libertà che esso concede all'utente. La potenza del software libero sta nella comunità, la comunità che cresce e contribuisce alla crescita dell'economia locale. l'intenzione della comunità del software libero è quello di creare dei punti di riferimento in ogni città e località ove possibile per dare assistenza alle imprese, scuole, università e pubbliche amministrazioni.

In poche parole, la comunità sostituisce le multinazionali monopolistiche che si introducono da migliaia di km di distanza senza contribuire minimamente l'economia locale.

La comunità del software libero in questo concetto crea anche nuove opportunità e quindi posti di lavoro lottando contro la precarietà.

Umit Uygur

HAX
HAX ARE EXPERIENCE

ANNO I
NUMERO I
LUGLIO 2007

PUBBLICAZIONE APERIODICA

RIVISTA IMPAGINATA CON OPENOFFICE.ORG

Hax Are eXperience

HAX SOMMARIO

OpenMoko, il telefono Open Source

È ufficiale , da oggi anche la Comunità Floss mondiale ha il suo telefono.
Più che telefono, in realtà, è una piattaforma indipendente dall'hardware denominata OpenMoko
che è usa il kernel Linux e che usa una sua GUI molto raffinata.

Il progetto OpenMoko si basa, a sua volta, su OpenEmbedded, e questo fa si che su OpenMoko possano girare applicazioni scritte per OpenZaurus, Familiar Linux e Ångström.
A dispetto delle altre soluzioni OpenMoko dispone di tutto il necessario per poter sviluppare nuove applicazioni, infatti è disponibile il codice del progetto in modo da poter accedere a tutte le applicazioni e scriverne di nuove ed è possibile sfruttare appieno le caratteristiche hardware senza restrizioni di alcun tipo.

OpenMoko ha già il suo primo telefono, il Neo 1973, che ha delle buone caratteristiche come il display touch-screen VGA da 2,8 pollici, utilizzabile con una stilo o con un dito (niente tecnologia multi-touch). Ha un processore Samsung “system-on-a-chip” da 266 MHz, ed è dotato di ricevitore AGPS, Bluetooth 2.0, codec audio e slot MicroSD. Per la parte telefonica il dispositivo è compatibile GSM 2.5G (quad-band). Al momento è disponibile in versione "developer preview" ed è ordinabile via internet per circa 300 dollari, versione base.

I modelli consumer arriveranno entro fine anno ed integreranno connettività WiFi, chipset per l'accelerazione grafica 3D e fino a 256 MB di memoria flash integrata. Al momento è tutto ma consiglio caldamente di stare in ascolto viste le potenzialità della piattaforma :-)

Happy Hacking .-)

http://www.openmoko.com

http://www.openembedded.org/

Ignazio Filice

MICROSOFT ALZA GLI SCUDI SU GPL3

Redmond (USA) - Microsoft non si è mai espressa ufficialmente sulla GPL3, ma la sua reazione alla recente approvazione della nuova licenza non si è fatta attendere. Sul finire della scorsa settimana BigM ha unilateralmente cambiato le condizioni del suo accordo con Novell con l'intento di dribblare i paletti posti da Free Software Foundation (FSF) nell'ultima revisione della GPL.

"Microsoft ha stabilito che le cedole di supporto per SUSE Linux che distribuisce ai suoi clienti non conferiscono al beneficiario il diritto di ricevere da Novell, o da qualsiasi altra parte, servizi di supporto e aggiornamento relativi a codice distribuito sotto licenza GPL3", si legge in un comunicato recentemente diffuso dall'azienda.

Per i clienti, ciò significa che i certificati di SUSE Linux Enterprise distribuiti da Microsoft non forniscono alcun servizio di assistenza per il software coperto da GPL3.

Con questa mossa BigM ha voluto assicurarsi di non rientrare in alcun modo nella definizione di distributore di software GPL, una posizione che con la GPL3 implicherebbe l'obbligo di fornire agli utenti una licenza libera da royalty di tutti i brevetti utilizzati nel codice distribuito. Tale clausola è stata fortemente voluta dal presidente di FSF, Richard Stallman, per impedire ai distributori di software (come Novell) di stipulare contratti di licenza esclusivi con i proprietari dei brevetti (come Microsoft).

"Microsoft non ha nulla a che fare con la GPL3 e nessuna delle sue azioni può essere interpretata come la partecipazione a questa licenza o l'assunzione di obbligazioni legali verso tale licenza", ha dichiarato perentorio il gigante di Redmond. "Non riteniamo che Microsoft necessiti di una licenza GPL per attuare la propria collaborazione con Novell, inclusa la distribuzione di certificati di supporto, e questo anche nel caso in cui Novell scegliesse, in futuro, di distribuire codice GPL3", ha affermato Microsoft.

Sebbene non ci si attenda che il kernel di Linux passi alla GPL3, diversi altri componenti chiave, come i tool di GNU, stanno progressivamente abbracciando la nuova licenza: a breve tutte le distribuzioni Linux, inclusa SUSE, conterranno sia software GPL2 che software GPL3.

In un comunicato pubblicato in concomitanza con l'approvazione della GPL3, Novell aveva precisato che "indipendentemente dalla posizione di Microsoft", continuerà a distribuire SUSE Linux Enterprise Server "con il suo completo insieme di caratteristiche e funzionalità, inclusi quei componenti che sono accompagnati dalla licenza GPL3".

Fonte: www.punto-informatico.it

Che differenza c'è tra sofware proprietario,open source e free software?

Open source e free software sono concetti che spesso vengono (erroneamente) confusi quando in realtà sono cose abbastanza diverse seppur non cosi distanti.Per software open source si intende qualsiasi programma che sia distribuito o venduto insieme al suo codice sorgente, cioè l'insieme delle istruzioni che lo compongono. Questo tipo di software viene spesso contrapposto a quello proprietario o a codice chiuso in cui il codice sorgente viene detenuto e non divulgato da chi sviluppa lo stesso.Andando in un ristorante e ordinando un piatto di pasta cio che si riceve è il prodotto finito di cui non si conoscono gli ingredienti e il tipo di preparazione (anche se si possono intuire): questo è il software proprietario. Invece la ricetta costituisce il codice sorgente in cui è indicato cosa è presente e in che modo va preparato. Avendo il codice sorgente si può sapere cosa si mangia :-) e modificarlo se è il caso.Il free software ha in comune con il tipo open source la disponibilità del codice sorgente, ma, in più, garantisce le 4 libertà fondamentali enunciate nella licenza GPL della Free Software Foundation

1) Libertà di eseguire il programma senza vincoli

2) Libertà di studiare il contenuto del programma (ovvero il suo codice sorgente)

3) Libertà di modificare il programma in ogni sua parte

4) Libertà di redistribuire copie identiche o modificate del programma sotto la stessa licenza

Come funziona l'interfaccia grafica nei sistemi GNU/Linux?

L'interfaccia grafica nei sistemi di tipo Unix-like (come GNU/Linux) viene gestita attraverso una architettura di tipo client-server in cui i termini, dal punto di vista dell'utente, sono "a rovescio": il server è quel programma che si occupa della visualizzazione dei contenuti sul pc "destinazione" (ovvero quello dell'utente) mentre il client è un qualsiasi programma che abbia bisogno di visualizzare il suo output su schermo.In questa ottica il server fornisce i componenti di base di una interfaccia grafica, cioè il disegno e lo spostamento delle finestre oltre all'interfacciamento con le periferiche di input come il mouse e la tastiera.Viceversa sarà il client a gestire la grafica e la funzione dei componenti usati nell'interfaccia grafica come i pulsanti, i menu, il titolo della finestra attiva etc...Per gestire l'apparenza e il posizionamento delle finestre viene usato un window manager come kwin o metacity rispettivamente per KDE e GNOME.L'architettura client/server rende il sistema trasparente rispetto all'uso su rete in modo che i client e il server possano trovarsi su macchine diverse collegate tra loro (ad es: attraverso ethernet o anche via internet usando un protocollo di comunicazione sicuro). Il gestore grafico che si occupa del rendering sullo schermo (cioè il server) è noto come X Window System (oppure X11 o semplicemente X). Attualmente l'implementazione della versione 11 (l'ultima disponibile) più usata è Xorg fornita dalla X.org Foundation.

Cosa significa ricompilare il kernel?

Un sistema operativo puà essere visto come un'automobile in cui il kernel è il motore affiancato dal resto dei componenti (telaio, sedili, finestrini, autoradio...). Per il buon funzionamento dell'auto è necessario che siano presenti e funzionanti tutti gli elementi costituenti ma il motore è il principale: se il motore non funziona l'auto non parte, è inutilizzabile.

Nel caso di GNU/Linux la parola "linux" indica appunto il kernel (che è il più diffuso ma non è l'unico disponibile: altri kernel possono essere HURD o Solaris). Esso fa funzionare un sistema operativo GNU (di tipo Unix-like).

In generale un kernel (e linux non fa eccezione) deve supportare una gran quantità di periferiche e funzioni di cui non tutti hanno bisogno: per questo si può pensare di modificarlo togliendo tutte le cose inutili che causano soltanto un degrado nelle performance del pc. Ciò è reso possibile dalla licenza GPL di Linux che fa si che sia fornito il codice sorgente e sia permessa la sua modifica.Tuttavia il codice sorgente è un linguaggio (teoricamente) comprensibile da un uomo ma non da un processore che si ciba solo di sequenze di 0 e 1. Per questo si ricorre alla compilazione che consiste nel trasformare il codice sorgente in un formato comprensibile dal calcolatore: le "parole" diventano sequenze binarie direttamente eseguibili.Quando si parla di ricompilazione non ci si riferisce solo alla mera traduzione del codice ma, in generale, a tutto il processo che va dall'ottimizzazione del codice sorgente del kernel alla compilazione vera e propria.Si dice ricompilazione perchè esistono versioni standard di kernel gia compilate e funzionanti che l'utente usa per far funzionare in un primo momento il suo sitema da cui effettuerà la seconda compilazione (la ricompilazione appunto) per adattare il kernel alle sue esigenze.

LINUS TORVALDS

Il creatore del kernel Linux ha iniziato tutto per divertimento. Oggi il suo lavoro fa funzionare milioni di computer e tutti lo considerano un eroe. Anche il Time.

Come Richard Stallman e' l'ideologo del movimento per il Software Libero, Linus Torvalds ne e' il braccio operativo. Linus ha creato il kernel che porta il suo nome ed ha inserito un mattone fondamentale nella costruzione avviata molti anni prima con il progetto GNU.
Oggi Linux e' il secondo sistema operativo più usato al mondo, guadagnando costantemente terreno sui sistemi proprietari. Se oggi abbiamo supercomputer, pc portatili e cellulari mossi da Software Libero, lo dobbiamo in buona parte anche a questo ragazzo finlandese, sempre sorridente, pacato ma deciso, che nel 1991 ha avviato lo sviluppo di Linux. Lo ha fatto sfidando all'inizio niente-popo-di-meno-che il prof. Tanenbaum, uno dei massimi esperti mondiali di programmazione di sistemi operativi. Egli aveva realizzato "Minix", un sistema operativo in qualche modo parente di Unix, che poteva essere eseguito su di un comune Personal Computer. Minix, la cui creazione aveva avuto scopi prettamente didattici, veniva distribuito con il codice sorgente, ma la sua licenza vietava di apportare modifiche al codice senza l'autorizzazione dell'autore. La divergenza tra i due era principalmente di carattere tecnico (kernel "monolitico" di Linus contro “microkernel” di Tanenbaum), ma presto divenne anche filosofica, sul modo di rendere disponibile il codice sorgente. E' in seguito a questa disputa che il nome di Linus Torvalds iniziò a circolare negli ambienti informatici.
Ma come e' nata l'idea di Linux, il progetto che ha reso Torvalds cosi' popolare a livello mondiale? Ce lo dice lui stesso nel titolo della sua autobiografia: "Just for Fun: The Story of an accidental revolutionary" (ossia "Solo per divertimento: storia di un rivoluzionario per caso") è il libro che racconta la nascita e la sorprendente crescita di Linux, dai primi messaggi sul newsgroup it.comp.minix ai successi di compagnie del calibro di Red Hat e Transmeta, dalle discussioni con Steve Jobs alle polemiche con Microsoft. Il libro è anche una riflessione sul sistema di regolamentazione della proprietà intellettuale, del copyright, dei diritto alla divulgazione delle conoscenze, argomenti su cui Torvalds ha sempre qualche seria parola da spendere.
Tra i fattori che hanno permesso il successo del software di Linus Torvalds, ci sono sicuramente: la licenza GPL e il modo di coinvolgere la comunita' di sviluppatori.
Uno dei passi migliori del libro è sulla filosofia open source e su Bill Gates: «Uno dei pezzi meno compresi del puzzle open source è il motivo che spinge ottimi programmatori a lavorare senza essere ricompensati. È indubitabile che le persone compiono i loro migliori lavori quando sono guidati da una passione, quando si divertono ... Il modello dell'open source dona alle persone l'opportunità di vivere le loro passioni, di divertirsi (To have fun) e di lavorare con i migliori programmatori del mondo. Sembra che Bill Gates non capisca tutto ciò».
La grande particolarità di Linux è a tutti gli effetti la sua licenza: essa impone che l'autore del software - Linus Torvalds per primo - anziché vietare, permette. Permette di usare liberamente il "prodotto del suo ingegno".
Permette che venga copiato, modificato, ampliato e persino venduto, senza imporre vincoli economici. Benché la licenza GPL dia grandi libertà, vi sono alcune limitazioni che sono anche i suoi principali punti di forza: quella fondamentale è che non è possibile prendere il software, modificarlo e rivenderlo senza che venga fornito anche il codice sorgente. Quindi chiunque voglia usare Linux (o un qualsiasi Software Libero), ad esempio in un cellulare, e gli apporta dei miglioramenti per farlo funzionare, questi miglioramenti devono essere anch'essi rilasciati con licenza GPL, quindi a beneficio dell'intera comunità.
Sara' stata questa lungimiranza a favorire il successo di Linux o l'abilita' di Linus come programmatore? Sicuramente molti fattori hanno contribuito e non ultimo il suo pragmatismo che lo porta sempre diritto verso gli obiettivi che si e' prefisso e che oggi lo porta a continuare lo sviluppo di Linux in seno all'Open Source Development Lab (OSDL) ora diventata Linux Foundation.
Un'ultima curiosità, sua moglie Tove e' stata ben sei volte campionessa nazionale finlandese di karate.

Un eroe, come Mandela, Gorbachev, Falcone e Borsellino. La prestigiosa rivista Time Europe ha incoronato Linus Torvalds come uno degli Eroi degli ultimi 60 anni, al fianco di Papa Giovanni Paolo II, Nelson Mandela, Falcone e Borsellino, Michael Gorbachev, i Beatles e Picasso. Un riconoscimento giustificato con la frase: “By giving away his software, the Finnish programmer earned a place in history “. Come non dargli ragione? http://www.time.com/time/europe/hero2006/torvalds.html

Links
http://it.wikipedia.org/wiki/Linus_Torvalds
http://www.ossblog.it/tag/torvalds
http://biografie.leonardo.it/biografia.htm?BioID=1152
http://www.linux-foundation.org

Vincenzo Bruno

HACKASERTA 81100

L'Hackaserta 81100 nasce come hacklab nel 2002. Tre anni dopo, diventa Associazione di promozione sociale col nome di “Hackaserta 81100”, 25 Febbraio 2005. La forma giuridica viene scelta per interagire con la pubblica amministrazione ed enti che richiedevano iter burocratici.

L'associazione condivide e difende l'etica Hacker e i valori del software libero, lo scopo è favorire la diffusione della filosofia del software libero promuovendone concetti, progetti e contenuti.
Dal 2002 Hackaserta 81100 organizza convegni, manifestazioni e corsi di formazione. Tra le attività svolte: recuperare vecchi computer dismessi e impiegarli per scopi di utilità sociale; diffondere la conoscenza dei problemi ecologici, sociali, culturali ed economici derivanti da un utilizzo improprio o diseguale delle tecnologie informatiche.
A marzo 2006 Hackaserta 81100 ha proposto alla Lug Conference la realizzazione di un'associazione nazionale degli utenti e degli sviluppatori del software libero.

Recentemente l’Associazione si è distinta in particolare per aver organizzato una petizione per l'adozione del software libero nella Pubblica Amministrazione (http://81100.eu.org/petizione) e per essere stata promotrice di Adunanza Digitale (http://adunanzadigitale.org).

La petizione è stata spedita al presidente del Consiglio Romano Prodi il 20 Luglio 2006 presso le Poste Italiane di Roma, contando circa 5000 firme. L’ evento è stato trasmesso dal format “Telecamere”, di rai3. (visibile al momento su http://www.youtube.com/watch?v=TBcpCKdKa9M). La lettera è stata riportata su diversi giornali online e cartacei. (http://81100.eu.org/wiki/LetteraRiportataSu - da Punto Informatico a Liberazione).

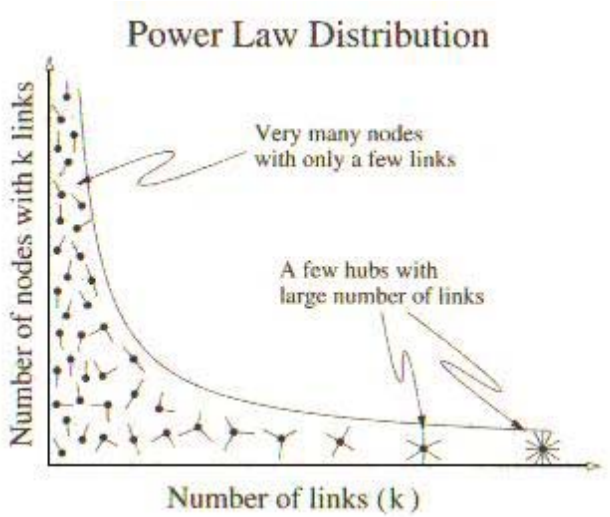
Adunanza Digitale è un'assemblea distribuita nel tempo e nello spazio. La manifestazione è divisa in vari tavoli di discussione. Ogni gruppo, che vuole aderire, può decidere di organizzare un tavolo di discussione. L'organizzazione è rigorosamente dal basso, Hackaserta 81100 si limita a fornire l'infrastruttura per lo streamming video e per lo screencast. Il motto è che la manifestazione è riuscita se un gruppo organizza una bella riflessione e permette ad altri di parteciparvi via Internet. Se poi si riesce a far procedere una riflessione comune allora non si può che esserne felici.

ph.d. Daniel donato
pres. Hackaserta 81100

I Motori di Ricerca a guardia del tesoro collettivo del web

Immaginiamo un drago buono che sta alla porta d'-entrata di un tesoro e ci dispensa ricchezze secondo le nostre esigenze. Noi non possiamo accedere alle ricchezze direttamente, ma entriamo in possesso di tutto quello che chiediamo. Gli unici sentimenti che proviamo sono la gratitudine per il drago e la gioia per la sua esistenza. Possiamo immaginare la ricerca sul web proprio a partire da questa metafora. Quando abbiamo curiosità o urgenza di trovare delle informazioni l'accesso ai motori di ricerca è un passaggio obbligato e inevitabile. La rete non sarebbe quello che è diventata se non fosse possibile esplorarla con l'aiuto di meccanismi di orientamento efficienti, veloci e accessibili che mettano in contatto le nostre esigenze con le risposte della ragnatela più grande del mondo.
Nonostante il ruolo fondamentale degli strumenti di ricerca, il triangolo da essi generato in relazione a web e navigatori non può dirsi privo di ombre. Recentemente un libro pubblicato dal gruppo Ippolita, The dark side of Google per Feltrinelli (ma è scaricabile gratuitamente o con un'offerta libera a questo indirizzo www.ippolita.net) mostra i lati oscuri del motore di ricerca più famoso del mondo, indicandone gli aspetti più pericolosi. Tra le caratteristiche più discutibili attribuite al motore di ricerca troviamo il rischio di danneggiare la privacy degli utilizzatori del servizio, quello della presunta oggettività dell'algoritmo di ranking utilizzato, oltre alla definizione ricorsiva dell'autorevolezza di una pagina sulla base della sua popolarità.
Il problema non è di facile soluzione, ma cerchiamo di far luce sui lati più imbarazzanti e intricati di uno strumento centralizzato per indicizzare il web che, invece, dovrebbe essere una struttura distribuita e disorganica, disarmonica e fuori controllo per qualsiasi autorità. Negli ultimi anni del secolo scorso alcune scoperte sconcertanti sulla struttura topologica della rete hanno minato l'idea che il web o la rete internet abbiano una struttura democratica o a distribuzione casuale. Dalle ricerche sperimentali effettuate risulta che la distribuzione dei link per i nodi segue la legge di potenza (vedi Fig.1).

Tale risultato tradotto in pratica evidenzia l'esistenza di pochissimi nodi iperconnessi (letteralmente con milioni di link) che vengono detti hub, mentre la maggior parte dei nodi sono sottoconnessi con pochi o un solo link. La struttura si riverbera inevitabilmente sulla natura facilmente egemonizzabile della rete. I pochi hub possiedono grande capacità di influenzare l'intera rete in quanto sono connessi con la maggior parte dei nodi e vengono conseguentemente molto frequentati. Altri studi sulla struttura del web in quanto grafo orientato (una struttura costituita da nodi e link orientati in un'unica direzione) hanno mostrato



Outlook.lnk Fig.1 Il grafico di una distribuzione che segue la legge di potenza http://www.macs.hw.ac.uk/~pdw/topology/Pictures/S-power.jpg

sperimentalmente, ma anche teoricamente, quale sia la struttura di una rete simile, che viene detta a papillon (vedi Fig.2). In essa convivono quattro zone: una zona centrale costituita da nodi ben connessi tra loro; una zona IN che consiste in nodi da cui è facile accedere al centro, ma difficile ritornare dal centro; una zona OUT accessibile dal centro, ma dalla quale non si torna verso il centro; una zona insulare sconnessa da tutto e anche poco connessa con se stessa.

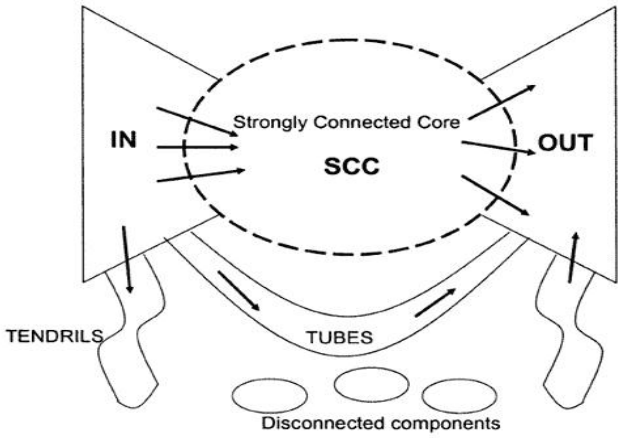


Fig.2 Struttura a papillon; fonte: http://habitat.igc.org/wealth-of-networks/figure-7-5.gif

Inoltre i crawler o spider del motore, cioè gli agenti software che ne costituiscono i "sensi", esplorano il web seguendo i link delle pagine ed evitando pagine dinamiche, form da riempire e altre aree simili. >

Tali meccanismi di accesso alle informazioni, insieme con la struttura topologica della rete, impediscono al motore di rappresentare oggettivamente tutto il web, senza considerare i notevoli problemi di freschezza delle informazioni reperite nei lunghi giri per pagine sempre più numerose.

Ora voi direte, ma questi sono problemi che devono risolvere i tecnici che si occupano di algoritmi di ranking degli indici inversi delle pagine web, di tecnologia per i softbot e di strategie di immagazzinamento dati. Ma non è così. Tutti/e siamo coinvolti/e da questo fenomeno perché quando ci affidiamo al motore per accedere alle informazioni non dobbiamo mai perdere di vista la grande quantità di vincoli che quei risultati (che per l'utente comune si limitano alle prime pagine) sopportano e l'esistenza di notevoli rischi di inquinamento delle informazioni che otteniamo. Possiamo riscontrare, tra le altre cose, la censura involontaria, la presenza di risultati parziali rispetto a questioni controverse, la mancanza di sistemi di protezione delle minoranze linguistiche nelle ricerche multilingue (l'italiano, per esempio, è estremamente simile allo spagnolo, lingua molto più diffusa della nostra), e altro ancora. Senza contare i problemi di privacy quando un'unica fonte detiene tutte le informazioni sul nostro conto (dai libri che leggiamo, alle ricerche che facciamo, alle zone che frequentiamo, ecc. ecc.) e ciò vale anche nella perfetta buona fede del nostro mediatore, che potrebbe comunque essere derubato di dati che ci riguardano.

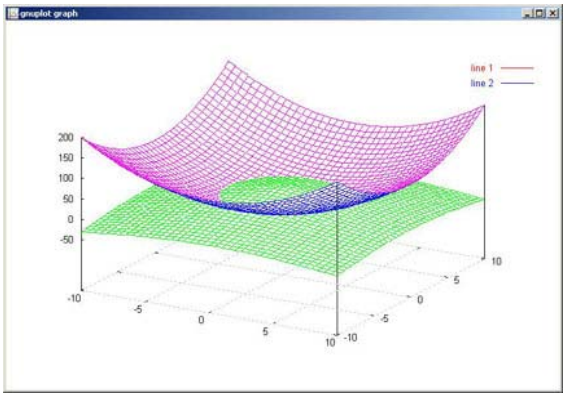
Cosa fare per evitare questi difetti degli strumenti di ricerca? Da soli possiamo fare ben poco, ma come collettività abbiamo più potere di influenza. Un comportamento ecologico consiste, per esempio, nel non utilizzare un unico motore, ma servirsi dei vari strumenti a nostra disposizione, enciclopedie, metamotori, motori specializzati nel vostro argomento di ricerca. Sarebbe molto importante e utile al nostro scopo anche sviluppare tecniche di ricerca dinamica simili ai metodi usati nelle reti P2P. Qui non esiste, di solito, un repository unico dei dati, le ricerche vengono istradate nella rete nel momento in cui vengono eseguite e per motivi di sostenibilità dell'istradamento si fermano a sette gradi di separazione dal punto di partenza. Un altro elemento importante della strategia di difesa sarebbe lo sviluppo della consapevolezza dei metodi utilizzati che ci permetterebbe di non scambiare la pagina dei risultati di un motore per un oracolo che pronuncia solo verità incontrovertibili. La grande risorsa del nostro spirito critico è uno strumento prezioso, forse in questo contesto, il più potente (come suggerisce anche Umberto Eco nei suoi interventi sull'argomento). Sarebbe poi, senz'altro, auspicabile chiedere che vengano resi pubblici i dettagli tecnici del funzionamento dei motori in modo da controllare il loro operato ed eventualmente migliorare collettivamente la tecnologia utilizzata.

Google, in particolare, persegue una politica un po' doppia rispetto all'apertura delle proprie procedure: utilizza piattaforme e tecnologie provenienti dall'open source adottando la licenza BDF (Berkeley Software Distribution) e anche quando mette a disposizione degli sviluppatori le API per la programmazione, lo fa con l'implicito obiettivo di fare talent-scouting, oltre che appropriarsi di eventuali miglioramenti significativi dei software modificati. D'altra parte il sistema usato da Google è sostanzialmente segreto se non per quanto riguarda la pubblicità dei brevetti presentati per difendere e proteggere le innovazioni prodotte all'interno dell'azienda, con l'argomentazione che altrimenti la qualità dei risultati sarebbe diminuita dall'eccesso di spam.
Se riprendiamo la metafora iniziale del drago la situazione potrebbe apparire ora molto cambiata. Abbiamo il tesoro che noi tutti contribuiamo a costruire, che è un bene pubblico e abbiamo un accesso regolato e mediato dal drago/motore di ricerca.
Vista in questa prospettiva non possiamo ignorare la delicatezza di una situazione nella quale alcune organizzazioni private, con un dichiarato scopo di lucro, sfruttano un bene pubblico, traendo guadagno notevole (attualmente Google si trova tra le prime aziende media al mondo per fatturato) dalla semplice – si fa per dire – organizzazione di quel materiale realizzato gratuitamente da quegli stessi fruitori del servizio, la cui attenzione viene messa in vendita al miglior offerente tra gli inserzionisti.

Teresa Numerico

OCTAVE il Calcolo Libero

Il programma GNU Octave è un programma di calcolo numerico' liberamente scaricabile dal sito www.octave.org.
Nacque presso l'Università del Texas come programma di calcolo per l'ingegneria chimica; il nome 'Octave' è il nome del docente di uno dei corsi presso i quali John W. Eaton, il principale autore del programma, ne iniziò lo sviluppo.
Poi fu esteso a coprire ambiti matematici più generali, come l'algebra lineare e le equazioni differenziali.
Attualmente Octave è un programma mantenuto e sviluppato dal progetto GNU, rilasciato con licenza GPL (è, quindi, software libero) ed è incluso nelle principali distro GNU/Linux, riceve contributi da un gran numero di sviluppatori indipendenti, spesso ricercatori e docenti universitari.
Tra gli ambiti di utilizzo di Octave si trovano:
l'aritmetica di base, l'algebra lineare, il calcolo matriciale, le equazioni algebriche e differenziali, la statistica, la ricerca operativa, la matematica finanziaria, la teoria dei controlli, l'analisi delle
immagini, l'analisi dei segnali, l'analisi del suono, e molti altri.



L'importanza del programma è notevole da vari punti di vista. Ne elenchiamo i principali:
1)E' uno strumento affidabile;
2) E' un ottimo sussidio didattico: a differenza dei programmi commerciali, permette di illustrare agli studenti anche il funzionamento del programma stesso. Si immagini l'utilità in corsi di informatica o di matematica applicata.
3)E' eticamente la migliore scelta dal punto di vista scientifico: nessuno dovrebbe potersi fidare
della "scatola nera" costituita da un programma proprietario in campo scientifico.La riproducibilità dei risultati scientifici non può essere affidata al numero di versione di un programma proprietario od alla speranza che gli sviluppatori di un tale programma non abbiano commesso errori.

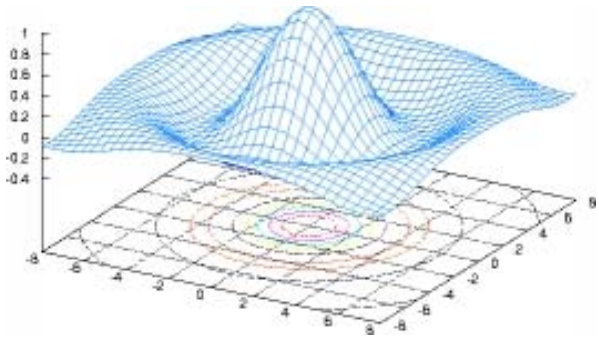
4)E' parzialmente compatibile con il linguaggio di Matlab ed è un programma a linea di comando ad esempio:

octave:3> a=5
a=5
octave:4> b=exp(a)
b = 148.41
octave:5> log(b)-a
ans = 0

Come *Matlab*, anche *Octave* permette di realizzare dei grafici, tuttavia non ha librerie grafiche. Esso si appoggia, infatti, su GNUplot, un programma per la realizzazione di grafici 2D e 3D, e presenta la possibilità di importare anche le librerie di *Matlab*.

Un'altra caratteristica certamente molto importante è l'opportunità offerta da GNU Octave di espanderne le potenzialità grazie al **supporto di moduli aggiuntivi**, analogamente a quanto è possibile fare con Matlab. Questo dà il vantaggio di non costringere l'utente (ad esempio qualcuno che lavora in ambito scientifico) a dover metter mano direttamente al codice sorgente del programma per aggiungere una feature: potrà creare un modulo apposito attenendosi alle istruzioni presenti nell'esaustivo manuale.

In conclusione, *Octave* è una valida alternativa a Matlab: versatile, facile da usare, con un'ampia documentazione semplice da reperire e soprattutto Open Source.



Mario Emanuele Lagadari

GIMP VS PHOTOSHOP

A confronto i due programmi di riferimento dei grafici

Fare un paragone tra Photoshop e Gimp è un'impresa ardua, ma sembra che sia necessario visto l'interesse ad affrontare questo argomento da parte di chiunque conosca i due programmi.
Innanzitutto Gimp è nato qualche anno dopo la comparsa di Photoshop come Creative Suite dell'Adobe. Gimp, di versione in versione, ha fatto enormi passi in avanti per colmare le difficoltà riscontrate da chi è abituato ad utilizzare Photoshop.

Il reale problema nel paragonare i due prodotti, al di là dei costi di acquisto (Gimp è un software gratuito), consiste nel decidere cosa realmente sia confrontabile. In altre parole: bisogna discutere su quando conviene utilizzare GIMP come sostituto di Photoshop (se ne state cercando uno) oppure bisogna discutere di Gimp semplicemente come software grafico? Gli impieghi di GIMP sono esattamente i medesimi: dal web al fotoritocco, all'animazione e alla pittura. Sicuramente GIMP è carente di alcune caratteristiche per un impiego tipografico, ad esempio la possibilità di creare immagini in quadricromia (CMYK) di contro Photoshop non gestisce animazioni, alcuni formati file (come l'MNG ossia un PNG animato che molti browser stanno introducendo come standard), ed è carente di un sistema di scripting avanzato (quindi non limitato dal puro e semplice batch processing o macro scripting).
Tenendo presente questo, con GIMP tutto quello che si può fare, lo si può fare bene e professionalmente, se si vuole.

L'impressione è che GIMP può diventare sicuramente il software leader nel campo della grafica; in altre parole quello che è mancato a GIMP è il tempo per crescere e distaccarsi da inutili dubbi, quale potrebbe essere quello sollevato dalla mancanza di un livello di astrazione nella renderizzazione di oggetti dinamici (sia esso del puro testo o un livello) o dall'incompletezza del tool di testo.

Per agevolare il passaggio da Photoshop a Gimp è possibile utilizzare 'Pspi', un plug-in che permette di utilizzare filtri di Photoshop. E' disponibile sia per sistemi Linux, che per Windows. Da una ricerca in Google si scoprirà la grande quantità di filtri disponibili sia in versione limitata, che "freeware". Comunque, diversamente dai plug-in di Gimp che utilizza le GTK, i plug-in di Photoshop usano librerie proprietarie. Questo perchè sono disponibili sia per Windows che per Macintosh.
In Windows: Ppspi è presente come eseguibile (pspi.exe), è sufficiente eseguirlo nella cartella dei plug-in di gimp.

Nel pacchetto scaricato per Linux sono presenti tre file:
README.linux

pspi, uno script da shell

pspi.exe.so, il binario che esegue wine

Copiare pspi e pspi.exe.so nella cartella dei plug-in di Gimp nella propria home, generalmente ~/.gimp-2.2/plug-ins . Quando si eseguirà Gimp comparirà un messaggio di errore "wire_read(): error" e pspi.exe.so non può avviarsi. Questo avvertimento non deve preoccupare (Gimp ignora quel file), ma se si desidera eliminare questo messaggio, si sposti pspi.exe.so in una diversa locazione e si aggiorni nello script pspi la nuova posizione.
Dopo aver avviato Gimp, nel pannello delle 'Impostazioni' dei Xtns:Photoshop Plug-in inserire la cartella da dove si prelevano i plug-in di Photoshop (con estensione .8bf) da usare in Gimp. Generalmente si usa una cartella inzialmente vuota, successivamente si installa una copia dei plug-in di Photoshop ad uno ad uno si dovrà verificare che funzionino.
Il progetto Gimp-Italia.org è molto giovane; è attivo da circa due anni e solo da quest'anno ha un sito web: http://www.gimp-italia.org. Siamo alla ricerca di persone che amano la grafica libera e che vogliono condividere con gli altri, senza risevarsi diritti, tutorial, tecniche e foto. Vogliamo valorizzare, attraverso Gimp e il progetto Gimp-Italia, le capacità dei grafici italiani.

Francesca Beatrice Cice, beatrix,
francescab.cice@gmail.com,
studia Sicurezza delle reti informatiche
presso l'Università di Milano,
manteiner di www.gimp-italia.org

Linus docet: sopravvivenza, vita sociale, intrattenimento

Una riflessione in merito alla legge elaborata da Linus Torvals (il padre del kernel Linux) e come essa si esprime nel modo di agire degli hacker.

Il tempo

Rileggendo `L'etica hacker` di Pekka Himanen mi sono trovato e riflettere sul tempo che molti dedicano all'informatica, il tempo che io ho dedicato all'informatica, il tempo che gli informatici dedicano all'informatica. Le notti, le giornate festive, le sere passate al computer a imparare... imparare a programmare, imparare ad usare al meglio il proprio sistema, imparare la differenza tra hacker e cracker, capire i sistemi operativi, capire la rete, i protocolli, la manipolazione dei dati, i paradigmi...

E se la propria passione diventa anche il proprio lavoro?

La mattina ti svegli e pensi al tuo codice o ai tuoi sistemi, dalle 9 alle 18 lavori sul tuo programma o sui tuoi server, torni a casa e non riesci a distogliere l'attenzione dai problemi che hai incontrato durante la giornata lavorativa... e' alienante, e' folle, e' terribile che esista qualcosa in grado di assorbire cosi' tanto una vita.

La vogliamo definire monotonia? Il pensare sempre all'informatica, fare dell'informatica il proprio lavoro, la propria passione e il proprio svago. Grazie all'informatica ho un lavoro, grazie all'informatica mi diverto, grazie all'informatica conosco gente interessante. Torvalds ha descritto in tre punti il modo di agire di molte persone che dedicano buona parte del proprio tempo - della propria vita - alla scienza con particolare riferimento all'informatica e ha denominato questi tre punti `La legge Linus` definendo cosi' il modo di agire degli hackers.

Sopravvivenza, intrattenimento, vita sociale

Volenti o nolenti viviamo in una societa' dove il supremo mezzo di scambio e' la moneta. La nostra sopravvivenza e' spesso (quasi sempre) legata alla disponibilita' di danaro. Con il danaro possiamo acquistare cibo, con il danaro possiamo procurarci un giaciglio, ecc... Il danaro e' di fatto una necessita' nella storia dell'umanita'. Anche gli appassionati di informatica, gli scienziati e perfino gli hacker hanno bisogno dei beni primari per sopravvivere, diventa quindi indispensabile procurarsi del danaro in modo da poter acquistare i beni - ed eventualmente i servizi - che ci interessano.
Non e' raro che chi si diletta nella programmazione trovi impiego come programmatore, come chi ha profonde conoscenze in ambito networking si adoperi come amministratore di rete. La passione e le conoscenze spingono gli individui a cercare lavoro in determinate direzioni in modo da garantire la propria sopravvivenza percependo una retribuzione (a prescindere dalla forma contrattuale) corrisposta ad un incarico lavorativo di proprio gradimento.
Quanto detto e' vero ma non rappresenta una costante, per quanto la sopravvivenza (fisica) sia indispensabile anche per gli hacker tale istinto e' spesso messo in secondo piano rispetto alla necessita' di ricercare legami sociali e intrattenimento. Possono sembrare concetti un po' difficili da digerire, chi metterebbe in secondo piano la propria vita/salute per un legame sociale o per la volonta' di divertirsi? Eppure non e' un fatto raro, basta pensare a cosa saremmo disposti a fare per salvare un rapporto di amicizia o un legame affettivo profondo. E che dire di chi pratica sport estremi rischiando di farsi male sul serio pur di divertirsi? La sopravvivenza e' certo un fattore non trascurabile ma non e' necessariamente il piu' importante.

L'importante e' divertirsi

Gli hacker del kernel Linux, gli hacker di internet, storicamente hanno passato (passano) molto tempo davanti al monitor del loro computer. Spesso spendono ore e ore del loro tempo libero per *divertirsi* con la programmazione, lo studio, affrontano nuove sfide informatiche, risolvono problemi. Affrontare problematiche legate al campo di proprio interesse diventa un modo per divertirsi, e' qualcosa di appagante e stimolante. L'hacker ha raggiunto un livello superiore di concezione dell'informatica, il suo sapere e la sua attitudine ad imparare e ricercare nuove soluzioni tecniche/scentifiche lo hanno portato ad utilizzare gli strumenti a sua disposizione non solo allo scopo di provvedere a se stessi in senso materiale (mi guadagno il pane programmando) ma anche per soddisfare la necessita' di divertirsi, per fare nuove conoscenze tramite la rete, per condividere il loro sapere e apprendere dalle esperienze altrui.

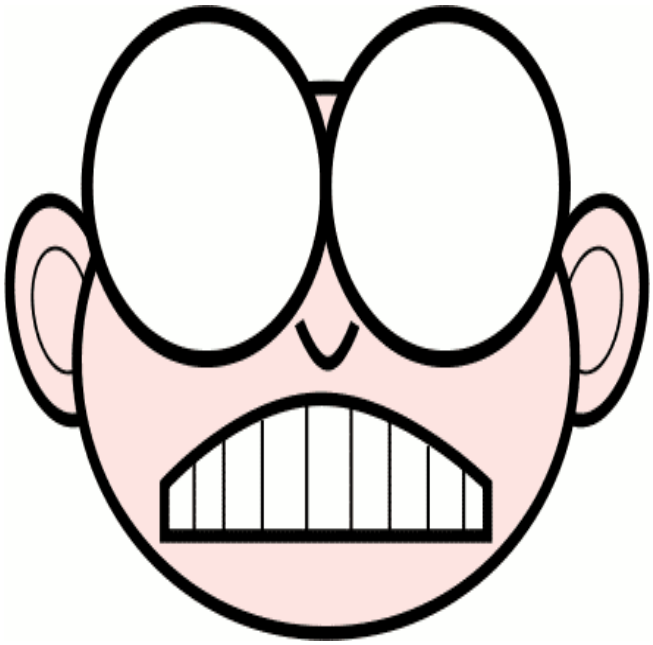
>

Non necessariamente nerd

Lo stereotipo vuole che un hacker sia un ragazzo adolescente, brufoloso, con gli occhiali che passa tutta la notte al computer mangiando schifezze e che non esce mai di casa perche' non conosce nessuno... sono certo che questa immagine in alcuni casi corrisponde a verita', sono anche certo del fatto che un hacker non e' necessariamente un nerd (su questo tema Raymond si espresse gia' in maniera esaustiva nel suo saggio `How to became a Hacker`). Osservando le realta' che oggi fanno hacking ci si puo' rendere conto che buona parte delle persone coinvolte nel tessuto sociale di internet sono ben disposte verso i rapporti sociali. La vita sociale e' un bisogno che viene soddisfanto in innumerevoli modi e che porta a rapporti interpersonali piacevoli quanto utili. Il mezzo comune a buona parte di questi rapporti interpersonali e' la rete: chi meglio di un hacker e' consapevole dell'immenso potenziale sociale che detiene internet? Dalle mail e le BBS siamo giunti in pochi anni alle community, le chat, i forum, la blogosfera (concedetemi il termine). Avere una vita sociale non e' qualcosa da dare per scontato, e' sicuramente piacevole, ci da la possibilita' di confrontarci non solo perciò che concerne l'informatica, ma anche per quanto riguarda la nostra vita.

La motivazione

Il fattore che in definitiva sembra spingere gli hacker ad adottare certi comportamenti e' il piacere di fare qualcosa di stimolante (intrattenimento) e di poterlo condividere con altri (vita sociale). Tutto il resto sembra passare in secondo piano, Torvalds ha di fatto osservato che la motivazione prescinde dalla sopravvivenza: dal momento che un hacker può fare ciò che più lo diverte assieme a chi come lui si diverte il resto non ha più importanza.



Rocco Sicilia

BRUCE PERENS E RICHARD M. STALLMAN
Roma 7-8-9 Giugno 2007

In occasione della Lection Magistralis presso la Cattedra del Prof.Arturo Di Corinto -Università La Sapienza, abbiamo incontrato Richard Stallman e Bruce Perens.

L'Aula Magna della Facoltà di Scienze delle Comunicazioni era talmente piena che la maggior parte delle persone stava in piedi per seguire Stallman e Perens (padri fondatori del Movimento del Software Libero e del software Open Source). La loro esposizione è stata semplice e lineare tale da permettere a tutti di comprendere le loro idee e il loro operato. Diversi sono stati gli interventi, tra cui quello di Daniel Donato dell'Hacklab di Caserta che ha esposto argomentazioni che gli avevo proposto io. Nel suo intervento ha presentato il progetto del Centro Studi sul Software Libero e l'idea di replicare progetti simili in ogni regione. Dai numerosi interventi che sono seguiti in proposito, è evidente che la proposta ha suscitato grande interesse.

Al termine dell'evento organizzato nell'Aula Magna, i due maggiori rappresentanti mondiali del movimento del Software Libero hanno incontrato il Presidente della Camera dei Deputati, l'On. Fausto Bertinotti, presso il Palazzo Montecitorio. Nonostante l'approccio pessimistico del Presidente verso le tecnologie in generale, l'incontro è andato abbastanza bene: si è dimostrato pienamente d'accordo con le idee di Richard Stallman, che ha discusso sul Software Libero unicamente da un punto di vista etico e filosofico. Purtroppo, l'On. Fausto Bertinotti, ha trovato qualche difficoltà a seguire in pieno l'intervento di Bruce Perens che ha affrontato l'argomento da un punto di vista strettamente tecnico.L'ultimo incontro della giornata,presieduto dall'On. Pietro Folena, si è tenuto presso la Commissione Cultura nel Palazzo Montecitorio. L'obiettivo era presentare il disegno di legge per l'introduzione del Software Libero nelle Pubbliche Amministrazioni (locali, Regionali e Nazionali). Stallman, nel suo intervento, ha esposto i pericoli che bloccano lo sviluppo della scienza informatica attraverso l'uso dei brevetti sul software (non in vigore in Italia). "Brevettare il software è come brevettare una sinfonia"- afferma Stallman - "se ciò fosse stato possibile, Beethoven avrebbe avuto molte difficoltà nel comporre le sue musiche". Mentre Perens, ha sottolineato come il software libero possa risultare migliore e più affidabile rispetto a quello proprietario, proprio grazie alle centinaia di migliaia di programmatori che, in tutto il mondo, lavorano attraverso internet. A chiusura del suo intervento, Stallman, si è espresso positivamente per quanto riguarda il progetto di legge presentato alla Camera dall'On. Pietro Folena, che afferma: "La pubblica amministrazione ha il dovere, verso i cittadini, di gestire i dati e le informazioni in modo trasparente e verificabile. Questo può assicurarlo solo il software libero".

Il progetto riguarda l'introduzione obbligatoria del Software Libero nelle pubbliche amministrazioni e per uso didattico. Proposte simili sono contenute anche in diversi progetti di leggi regionali (Emilia, Puglia, Campania, Toscana). La Commissione ha accolto positivamente gli intervento, chiedendo di dettagliare la proposta di legge ai fini di ripresentarla per l'approvazione.

Il giorno successivo si è tenuto l'ultimo incontro all'Ara Pacis. Alla tavola rotonda erano presenti anche Bruce Perens - Vice Presidente Open Source Movement, Carlo Daffara - Presidente Cirs, Emanuela Giannetta - Sun Microsystems Italia, >

Gabriele Ruffati - Direttore Sviluppo Engineering, Antonella Beccaria - Renomo, Pierpaolo Boccadamo - Responsabile Strategia di Piattaforma Microsoft, moderatore: Roberto Galoppino - Esperto Open Source Commerciale ed Ex Presidente del Cirs. Il contenuto dell'intervento del rappresentante della Microsoft riguardava la condizione di windows server sul mercato e la volontà della Microsoft a collaborare con le comunità del software open/free.

Ovviamente, io sono intervenuto incalzando con domande e precisazioni: il fatto che Novell collabori con loro non implica affatto che lo faranno anche le comunità, e ancora, quali sono i criteri legislativi che permettono alla Microsoft di imporsi come software pre-installato sui pc? Quelli che loro chiamano "accordi" sono "IMPOSIZIONI" e quello che loro chiamano "libero mercato" è "MONOPOLIO". A questi argomenti, Boccadamo non è riuscito a rispondere o a dare spiegazioni, mentre io ho avuto l'appoggio di Bruce Perens ed Henry Poole (Affero GPL), dimostrato in un sonoro applauso.

L'ultimo evento degno di nota è stata la partecipazione al corteo contro Bush, durante il quale, Stallman e Poole non hanno fatto mancare i loro slogan.

Bruce Perens

Umit Uygur

FOTOMONTAGGI CON GIMP

Una parte molto interessante nei programmi di manipolazione delle immagini è per l'appunto il fotomontaggio.
In questo tutorial vedremo come aggiungere il particolare di una foto, prendendolo da un'altra.
Nel nostro articolo, la prima foto sarà quella di un porto, a cui aggiungeremo un gabbiano in volo.

immagine: porto.png

Apriamo le due immagini con **File > Apri** oppure C**TRL+O**.
Iniziamo lavorando con l'immagine del gabbiano.

immagine gabbiano.png

Utilizziamo lo strumento **Forbici intelligenti** e clicchiamo sul contorno del gabbiano, creando diversi nodi.

Per attivare la selezione, occorre cliccare sul primo nodo creato e poi, con il tasto destro, all'interno dell'area selezionata.
In questo modo si crea la selezione attorno al gabbiano.

Invertiamo la selezione con **CTRL+I**, e dal menù **Livello** scegliamo la voce **Trasparenza > Alfa a selezione**.

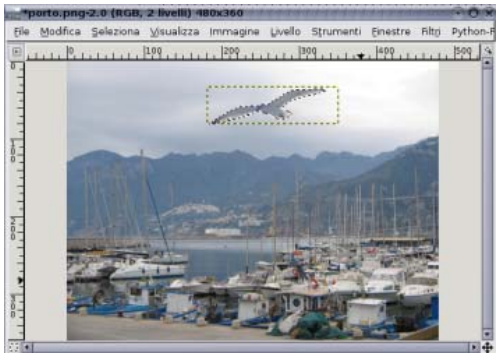
Premiamo **CTRL+X** per tagliare l'area che non ci interessa e avremo il gabbiano con lo sfondo trasparente.

Immagine: gabbiano-3.png

A questo punto, con un click del tasto destro sopra l'immagine del gabbiano, scegliamo **Modifica > Copia** oppure **CTRL+C** per copiare l'immagine.
Apriamo la finestra dell'altra immagine (in questo caso, del porto) e incolliamo l'immagine del gabbiano premendo **CTRL+V**, oppure cliccando con il tasto destro selezioniamo **Modifica > Incolla**.

L'immagine del gabbiano verrà applicata come **Selezione Fluttuante**, permettendoci in questo modo, tramite lo strumento **Sposta selezioni**, di posizionarla nel punto in cui vogliamo.

immagine:incollagabbiano.png



Una volta posizionata, per renderla immobile, dobbiamo cliccare con il tasto destro sul livello **Selezione Fluttuante** e scegliere la voce **Nuovo Livello**. In questo modo da immagine fluttuante diventerà un livello effettivo, che in qualunque momento potremo spostare.

Il risultato che abbiamo ottenuto è questo:

immagine: portogabbiano.png

Salviamo l'immagine in formato .xcf per poter effettuare future modifiche, e in formato immagine, ad esempio .jpg.

GPG E CRITTOGRAFIA

... Crittografia e Grande Fratello ...

Wikipedia insegna: “La crittografia tratta delle "scritture nascoste" (significato etimologico della parola) ovvero dei metodi per rendere un messaggio "offuscato" in modo da non essere comprensibile a persone non autorizzate a leggerlo” .
Usata fin dall'antichità, ha spesso deciso le sorti di guerre o più banalmente celato amanti a mogli tradite, ma il punto non è cosa si può fare con la crittografia,ma perché usarla! In un' era come la nostra dove, in nome di una fantomatica sicurezza, molte delle paure di Orwell si sono materializzate sotto il nome di Echelon e intercettazioni globali.
Quindi, usare la crittografia è uno dei pochi modi per assicurarci un po' di privacy. Al di là di questi già validissimi motivi, la crittografia è una misura di sicurezza formidabile per i nostri dati , l'unica che ci dà matematiche garanzie d'inviolabilità. Infatti, un sistema crittografico è ritenuto sicuro, se il miglior algoritmo esistente per violare la cifratura ci impiegherebbe, anche con ingenti risorse computazionali, un tempo irragionevolmente lungo (si parla di migliaia di anni).

Molte persone rimangono (inspiegabilmente) scettiche di fronte alla reale affidabilità di un programma di crittografia il cui codice è pubblico,ma il fatto che il codice sia pubblico non è una debolezza anzi!
Infatti come abbiamo già spiegato la sicurezza di un algoritmo di crittografia sta nella complessità computazionale che richiederebbe un attacco e non nel suo mantenere segreto il modo in cui un messaggio viene crittato. Inoltre, il fatto che il codice sia pubblico è un ulteriore sicurezza, poiché in caso esista una reale vulnerabilità dell'algoritmo, è probabile che questa venga scoperta da più persone ed è probabile che almeno una di queste la renda pubblica, aiutando a risanare la falla. Sono stati numerosi, invece, i casi in cui un algoritmo di crittografia “chiuso” si è scoperto contenere delle backdoor, grazie alle quali chi aveva fornito l'algoritmo poi spiava le comunicazioni (ne è un esempio eclatante Clipper Chip fornito dal governo statunitense!)

GPG e crittografia a chiave pubblica

Gpg (acronimo di Gnu Privacy Guard) rientra nella sfera della crittografia a chiave pubblica dove l'utente dispone di una coppia di chiavi: una pubblica e una privata. Per comprendere il funzionamente di questo meccanismo pensiamo alla chiave pubblica come ad una cassaforte aperta, nella quale, chi vuole scriverci lascia il suo messaggio e chiude la cassaforte. In un secondo momento, noi con la nostra chiave privata andremo ad aprire la cassaforte e a leggere il messaggio. La chiave pubblica va, in pratica, diffusa il più possibile (esistono appositi server su cui caricare queste chiavi in modo che chiunque possa trovarla), in modo che chi voglia scriverci possa usarla per crittare il messaggio e inviarcelo. La chiave privata invece va conservata gelosamente e protetta con una password robusta.

Installazione

Puntiamo il nostro browser su http://www.gnupg.org/ e scarichiamo i sorgenti che, al momento della stesura dell'articolo, sono arrivati alla versione 1.4.7 , o alla 2.0.4 per la versione modularizzata. Usiamo lo script “./configure” per vedere se il nostro sistema possiede tutti i programmi e le librerie richieste, fatto questo eseguiamo gli ormai noti comandi “make” e “make install”. Ora siamo pronti a salvaguardare la nostra privacy!

GPG

Usare gpg da shell

Per iniziare ad utilizzare gpg bisogna generare una coppia di chiavi, quindi aprite il vostro terminale e scrivete:

gpg --gen-key

viene chiesto di selezionare il tipo di chiave che si vuole generare ed è possibile scegliere tra tre opzioni.
Con l'opzione 1 vengono create due coppie di chiavi: una di tipo DSA (chiave primaria, con la quale è possibile solamente firmare) ed una di tipo ElGamal (con la quale è possibile criptare).
Con l'opzione 2 viene creata solamente una coppia di chiavi di tipo DSA mentre l'ultima opzione genera una singola coppia di chiavi ElGamal (con la quale si può sia criptare che firmare).

>

Con l'opzione 2 viene creata solamente una coppia di chiavi di tipo DSA mentre l'ultima opzione genera una singola coppia di chiavi ElGamal (con la quale si può sia criptare che firmare).

Selezionata una delle opzioni, ci viene chiesto di definire alcune caratteristiche sulle chiavi da creare, dobbiamo:

specificarne la dimensione (lunghezza da specificare in bit) e assegnare una data di scadenza.

E' da notare che, inserendo 0, la chiave non scade, cioè la rendiamo “immortale” ... :-) (almeno finchè non decidiamo di revocarla).

Infine ci viene chiesto di inserire una password per proteggere chiave primaria e secondaria.

Adesso può essere utile conoscere alcuni comandi utili per l'uso di gpg.

gpg --export [UID] ci permette di esportare la nostra chiave pubblica.
Il parametro UID (User ID) è opzionale: se non viene specificato verranno esportate tutte le chiavi presenti nel portachiavi. In più, con l'opzione -o è possibile dirigere l'output su un file, mentre utilizzando l'opzione -a le chiavi vengono scritte in un file ASCII a 7 bit invece che in un file binario.
Un altro comando utile è:

gpg --import [Nome file]

che ci permette di aggiungere la chiave di un amico al nostro portachiavi (l'ultimo parametro è opzionale e omettendolo leggiamo le chiavi da standard input).

Se la chiave privata viene persa può essere revocata con il comando:

gpg --gen-revoke

con il quale viene generato un certificato di revoca.
GnuPG ha messo a disposizione una serie di comandi per gestire le proprie chiavi:

gpg –list-keys vengono mostrate tutte le chiavi pubbliche esistenti.

gpg –list-sigs vengono mostrate le firme digitali delle chiavi pubbliche presenti.

gpg –fingerprint vengono mostrate le impronte digitali della chiavi pubbliche presenti.

gpg –list-secret-keys viene mostrata la lista delle chiavi private.

gpg --delete-key UID viene cancellata una chiave pubblica.

gpg --delete-secret-key UID viene cancellata una chiave privata.

gpg --edit-key UID viene modificata una chiave tramite un prompt interattivo da cui digitare i comandi.

Un comando importante è ***sign*** con il quale è possibile firmare una chiave (richiede l'inserimento della password).

Cifrare, Decifrare e Firmare

Fino a qui abbiamo creato chiavi ed aggiunto le chiavi degli amici al portachiavi. Adesso possiamo finalmente iniziare a cifrare e decifrare, impariamo due nuovi comandi:

gpg --encrypt destinatario [dati]

per cifrare, mentre per decifrare i dati usiamo:

gpg --decrypt [dati]
anche qui, utilizzando l'opzione -o reindirizziamo l'output in un file altrimenti verra stampato su *stdout* (standard output).
Per evitare che qualcuno si spacci per noi è consigliabile firmare i dati con la propria chiave, usiamo quindi il comando:

gpg --sign [dati]

però durante questa operazione i dati vengono compressi e, quindi, diventano illeggibili. Per lasciare i dati leggibili bisogna dare il comando:

gpg --clearsign [dati]

E' possibile inserire dati e firma in due file separati con il comando:

gpg --detach-sign [dati]
Supponiamo che un nostro amico ci abbia inviato un messaggio segretissimo: procediamo decifrando il file e poi verifichiamo la firma con il comando:

gpg -verify [dati]

(è necessario la chiave pubblica del mittente).
Per maggiori informazioni su GPG vi consiglio di visitare il sito web del progetto:

http://www.sito.org

Interfacce grafiche per gpg

L'uso di gpg è facilitato grazie a programmi che consentono la gestione delle chiavi con semplici click, qui sotto sono riportati i più conosciuti:

>

1. GPA (GNU Privacy Assistant)
2. Seahorse (interfaccia grafica per GNOME)
3. Geheimnis (interfaccia grafica per KDE)

Programmi di Posta Elettronica

Diversi programmi di posta elettronica supportano gpg (sono molti, è difficile elencarli tutti, non odiatemi se dimentico il vostro preferito :-D):

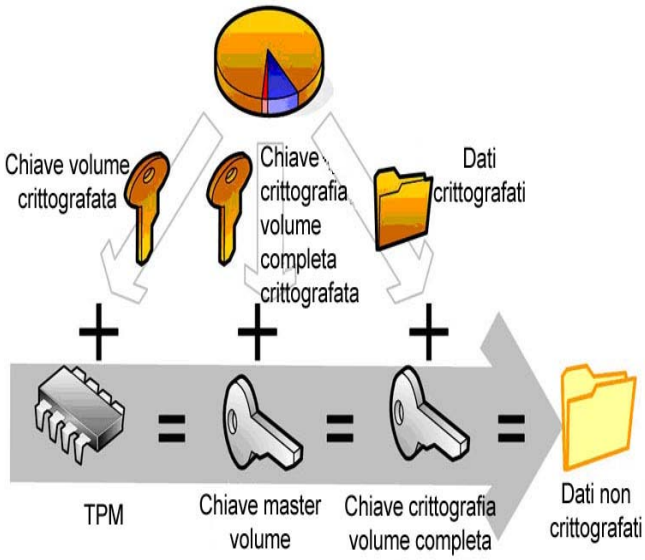
- Mozilla Thunderbird (con il plugin Enigmail)
- Kmail
- Eudora
- Pine
- Mutt

Estensione per FireFox

Da poco è stata rilasciata una ottima estensione per firefox ***FireGPG*** che permette di criptare, decriptare e firmare tutto quello che si invia con il browser, ad esempio, la posta elettronica , senza utilizzare un client. Da notare che, quando si compone una nuova email con gmail, accanto al pulsante *invia* troviamo in aggiunta altri quattro pulsanti: *cripta, firma, cripta e invia, firma e invia.* Per installare l'estensione basta accedere al sito sottostante con firefox e cliccare sul link *download FireGPG*:

http://firegpg.tuxfamily.org/index.php?page=install&lang=en

una volta cliccato partirà automaticamente l'installazione dell'estensione.



Semplici regole con Nmap e NetDiff

Due grossi problemi per gli amministratori di rete sono la mancata certezza che tutto quello che si è impostato nella propria rete funzioni e le possibili intrusioni esterne, rilevate anche dai log IDS (Intrusion Detection System)

Nmap è un portscanner open source con licenza GNU GPL realizzato da Fyodor. Nmap permette di rilevare nella rete sia la presenza di nuove postazioni che l'attivazione improvvisa di nuovi servizi sulle diverse porte del firewall di rete. Nel corso dell'articolo sono descritti alcuni esempi di scansioni possibili con Nmap, sottolineando quando è utile la sua implementazione.

Il ping

Gestire una rete ed essere sicuri dell'assenza o meglio dell'impossibilità che accessi non autorizzati abbiano libero accesso nella stessa è un compito arduo e molte volte difficile e non privo di sbagli. I pericoli sono sia esterni che interni; anche un utente all'interno della rete può utilizzare la dislocazione degli host e sfruttare incautamente la sua posizione per distribuire illegalmente contenuti a spesa dell'azienda.

Per l'installazione di Nmap in Debian (e derivate) è sufficiente digitare il comando: apt-get install nmap

L'installazione di Nmap in Windows parte dal download dell'ultima versione da http://insecure.org/nmap/download.html , scaricare l'ultima versione di cygwin da http://www.cygwin.com/ e nel terminale che si apre aprendo una sessione di cygwin digitare: tar xvjf nmap-version.tar.bz2 o tar xvzf nmap-version.tgz

Il primo test da effettuare potrebbe essere verificare quali postazioni sono attive, attraverso un ping di ogni host nella rete:

nmap -n -sP -oA output_file 192.168.10.0/99

Verrà prodotto un file in cui saranno elencati tutti gli host che hanno risposto al segnale di Nmap, -sP esegue il ping degli host, mentre -oA salva il file in formato txt, in formato gestibile con grep e XML. L'opzione -n indica ad Nmap di considerare gli IP e non i nomi DNS.

Il formato più interessante è quello gestibile con grep, poiché permette di analizzare l'output usandoi comandi cut e grep:

cat output_file.gnmap | grep ^[0-9]

Questo comando controlla che il file output_file.gnmap abbia un indirizzo IP valido.

Quando lanciamo il comando Nmap, per eseguire il ping delle macchine, possiamo anche utilizzare l'opzione -PS, ed ottenere una selezione di tutte le porte attive con il servizio TCP.

Ora sappiamo che le porte su cui è attivo il servizio TCP, sono la 21, la 22, la 23 e l'80; allora il comando precedente può essere scritto: nmap -sP -PS 21,22,23,80 -oA output_file 192.168.10.0/99

Con questo comando si eseguirà il ping degli host che hanno servizi attivi sulle porte 21, 22, 23 e 80 nell'intervallo degli indirizzi IP 0-99.

Fino a questo momento abbiamo verificato solo quante e quali macchine sono attive nella rete considerate; è il caso di iniziare ad effettuare la scansione delle stesse, ecco il comando:

nmap -PS 21,22,23,80 -p1-1000 -oA output_file 192.168.10.0/99

Praticamente, Nmap interroga la rete, considerando gli IP da 0 a 99 e tutte le porte da 0 a 1000, verificando dove sono attivi i servizi TCP.

Netdiff

In alternativa è possibile utilizzare un tool per l'analisi della rete, Netdiff, scritto in perl che esegue il portscan di Nmap di una specifica rete o reti e salva i risultati in un database MySQL. Effettua anche un report delle differenze tra successive scansioni, generando delle viste dei cambiamenti più recenti nella rete. Anche NetDiff è open source con licenza GNU GPL.

Prima di tutto è necessario modificare il file /etc/netdiff/netdiff.conf per settare le seguenti informazioni:

- inserire il *db_name*, *db_user*, e *db_pass per il database creato in MySQL;*
- in *report_to inserire l'email a cui inviare i report di NetDiff;*
- aggiungere i dati della rete o delle reti su cui effettuare la scansione nel formato *19-2.168.1.0/24*.

Queste informazioni possono essere settate attraverso il modulo WebTool di NetDiff.

NetDiff può essere invocato da linea di comando o si può utilizzare un web browser tramite il modulo WebTool ed eseguendo EnGarde Secure Linux.

NetDiff può essere impostato in cron per essere eseguito, ad esempio, durante le ore notturne.

Le opzioni di netdif sono:

netdif [OPTION][FILE]

-a, --all Perform full scan and report

-s, --scan Scan but do not report

-r, --report Report results of last scan

-i, --import Import FILE where FILE is an nmap XML report

>

In alternativa è possibile utilizzare un front-end grafico di NetDiff, disponibile con il modulo WebTool di EnGarde Secure Linux. Il modulo crea automaticamente il database (se non presente) e permette anche di settare le opzioni di scansione, di configurare i riceventi dei report, e di modificare la lista delle reti da scansionare. Inoltre, permette di esaminare tutti i report generati in passato, attraverso un'interfaccia semplice e diretta visualizzabile via browser.
La pagina principale del modulo WebTool contiene un link per la configurazione, una lista dei report generati in passato e una lista delle reti da scansionare.

Cliccando sul link per la configurazione si apre una pop up, in cui è possibile settare le opzioni Nmap e le email (separate da una virgola) a cui inviare i report.

Selezionando un report dalla lista viene aperta una finestra in cui possono essere analizzati i contenuti e nella stessa è possibile anche cancellare i report non più necessari. Nei report saranno indicate con un + le porte aperte e con un – quelle chiuse dall'ultima scansione.

Scegliendo una rete o cliccando sul link *Add Network* si aprirà una finestra dove è possibile modificare o cancellare una rete specifica dalla lista di scansione.

In conclusione, grazie a nmap e al tool di supporto come netDiff è possibile monitorare velocemente i propri sistemi e le proprie reti, con la possibilità di evitare in anticipo accessi indesiderati alle proprie informazioni.

Francesca Beatrice Cice

beatrix

HAX

Hax Are eXperience

Be Linux...

HLCS
HACKLAB COSENZA

H A X