

GIORNALinux 2.0

Rivista aperiodica per studenti

JUST KEEP ON HACKING

N10 | 12.11



Tutti i software CAD per LINUX - OpenVPN, parte 2 Proteggere i dati con la Crittografia Quantistica

Politecnico Open unix Labs

Editoriale

Daniele, Pietro, Stefano
<direttivo@poul.org>

CARI lettori, sono passati diversi mesi dall'ultimo numero del Giornalinux. Nel tempo che è trascorso ci sono stati dei cambiamenti ai vertici dell'associazione, è stato eletto infatti il nuovo direttivo che va a sostituire quello dell'anno precedente. Ora alla guida dell'associazione troviamo il Segretario uscente Daniele lamartino nel ruolo di Presidente e ad assisterlo il nuovo Vicepresidente Pietro Virgilio ed il nuovo Segretario Stefano Bouchs. Si sono inoltre svolti ad Ottobre i corsi GNU/Linux base. Anche se non c'è stata una vastissima partecipazione, ci sono stati interventi molto interessanti (virtualizzazione, linux per lavorare con tracce audio, ...). Come abbiamo già fatto per i Corsi GNU/Linux avanzati, abbiamo registrato ogni lezione del corso ed abbiamo reso disponibili i video sul nostro sito insieme al resto dei materiali, in modo da permettere a chi non ha potuto frequentare di assistere comunque alle lezioni. Questa pratica di registrare i nostri eventi va oramai consolidandosi e potete aspettarvi di trovare sul nostro portale i video di tutti i prossimi eventi organizzati dal POU.L. Ed è pro-

prio su quello che abbiamo in programma quello su cui questo nuovo direttivo sta impiegando le proprie energie. Non molto tempo fa alcuni mettevano in dubbio la stimolazione di eventi da parte del POU.L, ritenendola una associazione in via di declino. Invece, ora come non mai, siamo pieni di proposte, corsi, eventi e conferenze da pianificare! Inizieremo l'anno 2012 con l'ormai classico workshop/conferenza sulla sicurezza informatica a Gennaio, di cui stiamo ancora discutendo i temi da presentare (Se avete delle idee non esitate a mandarcele! Siamo sempre raggiungibili ad info@poul.org!). Seguiranno poi tra Marzo e Giugno dei corsi sulle librerie Qt, sul linguaggio Python, sull'editor LyX. Non possono ovviamente mancare gli incontri dei corsi avanzati GNU/Linux, che furono un enorme successo nella loro prima edizione e che nella prossima saranno ripensati e migliorati grazie ai vostri feedback. Anche il prossimo anno terremo i corsi GNU/Linux base verso Ottobre. Riproporremo poi, questa volta in modo più aperto, un laboratorio di programmazione su microcontrollori. Al prossimo numero!

Indice

OpenVPN: parte 2	3
Crittografia quantistica	7
CAD e Linux	10

Quest'opera è rilasciata sotto la licenza Creative Commons BY-NC-SA 2.5. Questo significa che sei libero di riprodurre, distribuire, comunicare al pubblico, esporre in pubblico, rappresentare, eseguire o recitare l'opera e creare opere derivate alle seguenti condizioni:

-  **Attribuzione.** Devi riconoscere il contributo dell'autore originario.
-  **Non commerciale.** Non puoi usare quest'opera per scopi commerciali.
-  **Condividi allo stesso modo.** Se alteri, trasformi o sviluppi quest'opera, puoi distribuire l'opera risultante solo per mezzo di una licenza identica a questa.

In occasione di ogni atto di riutilizzazione o distribuzione, devi chiarire agli altri i termini della licenza di quest'opera. Se ottieni il permesso dal titolare del diritto d'autore, è possibile rinunciare ad ognuna di queste condizioni.

Le tue utilizzazioni libere e gli altri diritti non sono in nessun modo limitati da quanto sopra.

Questo è un riassunto in linguaggio accessibile a tutti del Codice Legale:
<http://creativecommons.org/licenses/by-nc-sa/2.5/it/legalcode>



OpenVPN: parte 2

Daniele Iamartino
<danieleiamartino@gmail.com>

NEL numero di Dicembre 2010 del Giornalinux ci eravamo lasciati dopo aver parlato di come configurare il nostro server ed il nostro client OpenVPN. Vediamo ora un po' di modifiche molto interessanti per sfruttare la nostra VPN.

Uscire su internet tramite il server della VPN

Per prima cosa andiamo sul server e abilitiamo il forwarding dei pacchetti e il masquerade:

```
echo "1" > /proc/sys/net/ipv4/
ip_forward
iptables -t nat -A POSTROUTING -s
10.0.0.0/24 -j MASQUERADE
iptables -P FORWARD DROP
iptables -A FORWARD -s 10.0.0.0/24
-i tap0 -j ACCEPT
iptables -A FORWARD -o tap0 -m
state --state RELATED,
ESTABLISHED -j ACCEPT
```

Poi spostiamoci sul client e cambiamo la sua configurazione. Dobbiamo aggiungere la riga

```
redirect-gateway
```

Che sostanzialmente dice al client di fare questo: Crea una regola di routing statica di modo da restare collegato al server della VPN mediante il gateway default attuale, poi cancella il routing del gateway default e aggiungi un nuovo gateway di default. Se stiamo usando TUN allora OpenVPN riesce a trovare da solo quale gateway di default impostare, se usiamo TAP invece dovremo specificare ad esempio di usare 10.0.0.1 con il parametro:

```
route-gateway 10.0.0.1
```

In ogni caso, dobbiamo prestare attenzione ai DNS. Se il nostro computer già utilizza dei DNS pubblici non dobbiamo preoccuparci, mentre se ad esempio usiamo come server DNS il router interno alla nostra rete, una volta cambiate le regole di routing non riusciremo più a raggiungerlo. Uno dei tanti modi per risolvere questo problema è creare due script che cambiano i server dns. Per evitare conflitti e per comodità installiamo il programma resolvconf sulla nostra distribuzione e creiamo ad esempio il file /usr/bin/dnsvpn su con questo contenuto:

```
#!/bin/bash
echo "nameserver 8.8.8.8" |
resolvconf -a tun0
```

Che una volta eseguito aggiungerà come server DNS primario Google. Creiamo poi un altro file `/usr/bin/dnsvpngiu`

```
#!/bin/bash
resolvconf -d tun0
```

Che cancellerà il server DNS aggiunto dalla vpn quando la scolleghiamo. Diamo i permessi di esecuzione a questi due script:

```
chmod +x /usr/bin/dnsvpnsu
chmod +x /usr/bin/dnsvpngiu
```

E a questo punto torniamo nel nostro file di configurazione del client della vpn e aggiungiamo alla fine i comandi per lanciare questi due script rispettivamente all'avvio e all'arresto della vpn:

```
script-security 2
up /usr/bin/dnsvpnsu
down /usr/bin/dnsvpngiu
```

Kung-fu degli IP e delle porte di rete

Può capitare che siamo su un client della VPN che ha bisogno di esporre su internet un servizio, come un server web o qualcosa d'altro, però si trova dietro la NAT che abbiamo creato prima. Un' interessante modifica alla VPN potrebbe essere quella di aggiungere sul server VPN una regola che rediriga il traffico in ingresso da una certa porta TCP/UDP ad un'altra porta TCP/UDP di un client della VPN. Per fare ciò ci basta aggiungere sul server qualcosa del tipo:

```
iptables -t nat -A PREROUTING -i
eth0 -p tcp --dport 6001 -j
DNAT --to-destination
10.0.0.2:22
```

```
iptables -A FORWARD -o tap0 -d
10.0.0.22 -p tcp --dport 22 -j
ACCEPT
```

Che redirige il traffico in ingresso sulla porta TCP 6001 del server alla porta 22 dell'host 10.0.0.2 della VPN. Il secondo comando iptables serve invece per abilitare il forward dei pacchetti verso quella porta dell'host interno alla vpn. Su alcuni server può capitare anche di avere più indirizzi IP, perché ad esempio lo stesso server ha diverse interfacce di rete, ciascuna con un proprio IP pubblico. Potrebbe essere molto utile che un certo client della rete VPN utilizzi uno degli IP pubblici del server per se stesso, di modo da avere un IP pubblico ovunque si trovi e risultare praticamente come se fosse senza NAT. Per fare questo dovremo aggiungere due regole di iptables sul server:

```
iptables -t nat -A PREROUTING -d
123.45.6.7/32 -i eth0 -j DNAT
--to-destination 10.0.0.2
iptables -t nat -A POSTROUTING -s
10.0.0.2/32 -o eth0 -j SNAT --
to-source 123.45.6.7
iptables -A FORWARD -o tap0 -d
10.0.0.2 -j ACCEPT
```

Dove 123.45.6.7 è l'indirizzo ip che vogliamo reindirizzare e 10.0.0.2 è l'host della VPN a cui reindirizzarlo. È sottinteso che l'host della VPN su cui facciamo questa operazione deve per forza uscire su internet tramite il server della VPN.

Utilizzare OpenVPN dietro proxy

Può essere che qualche client della rete si trovi per qualche motivo dietro a un proxy,

come accade spesso in molte aziende. Per collegarci basterà specificare due opzioni aggiuntive nel file di configurazione del client:

```
http-proxy PROXY_HOSTNAME
          PORTA_PROXY
http-proxy-retry
```

La revoca dei certificati

La vostra configurazione funziona perfettamente, però avete appena scoperto che un certificato ha la sicurezza compromessa (è stato rubato, è stato perso). Non volete assolutamente che nessuno in possesso di quel certificato riesca a fare nulla, quindi decidete di revocarlo. Ovviamente avete ancora il certificato nella vostra cartella di easy-rsa sul server, quindi entrateci:

```
cd /etc/openvpn/easy-rsa/2.0/
source ./vars
./revoke-full client1
```

Usciranno una serie di messaggi, alla fine vedrete error 23, significa che la revoca è stata effettuata con successo... Vedrete un nuovo file keys/crl.pem che contiene l'elenco dei certificati revocati. Ora dovrete copiare il file nella cartella principale del server:

```
cp keys/crl.pem /etc/openvpn/crl.
pem
```

E quindi dovrete aggiungere una riga nel file di configurazione del server, se non l'avevate già:

```
crl-verify /etc/openvpn/crl.pem
```

Se avevate già aggiunto questa riga in precedenza e volete comunque aggiornare

la configurazione del server, per avvertirlo di non accettare le connessioni da quel client, potete lanciare (sui sistemi debian-like):

```
/etc/init.d/openvpn reload
```

Questo comando aggiorna la configurazione, nessuno dei client si accorgerà di nulla, tranne il client con certificato revocato.

Un modo un po' più sicuro di generare certificati

Come già detto prima, non è proprio la cosa migliore in fatto di sicurezza quella di generare chiavi pubbliche e private insieme, perchè se nel loro transito dal server a uno dei client venissero entrambe intercettate da un attaccante, questo potrebbe sfruttarle a suo favore per collegarsi alla rete. Tutto questo ipotizzando che siano chiavi non protette da password, ma anche se lo fossero potrebbe restare un po' di margine per tentare degli attacchi a forza bruta e la cosa non ci piace. Questo è il motivo per cui se si procede il quel modo bisogna essere assolutamente certi di trasferire le chiavi su un canale sicuro. Vediamo allora ora come riuscire a spostare comunque in modo sicuro le chiavi pur usando easy-rsa. Ci spostiamo su uno dei client e generiamo la sua chiave privata nella cartella di openvpn:

```
openssl genrsa -out client1.key
1024
```

Se vogliamo utilizzare chiavi di dimensioni diverse basta cambiare ovviamente l'ultimo parametro. A questo punto generiamo

la richiesta di certificato che invieremo al server.

```
openssl req -new -key client1.key
           -out client1.csr
```

A questo punto copiamo in qualche modo il file generato `client1.csr` nella cartella `keys/` di `easy-rsa` sul server e spostiamoci nella cartella principale di `easy-rsa`, dove ci basterà lanciare:

```
./sign-req client1
```

E dopo una conferma il server utilizzerà la CA per creare un certificato valido a partire dalla nostra richiesta. Probabilmente uscirà un errore di `chmod` a fine operazione perché non è presente la chiave privata (e non vogliamo che ci sia!), ma possiamo ignorarlo tranquillamente. A questo punto andiamo nella cartella `keys/` di `easy-rsa`, recuperiamo il certificato generato che si chiamerà ovviamente `client1.crt` e spostiamolo sul nostro client. Ora il nostro client avrà sia chiave pubblica che privata e anche se ci fosse stato un eventuale attaccante a intercettare il traffico, non potrà farsene nulla dei dati ricavati, poichè gli manca la chiave privata, che non conosce.

Link Utili

Per approfondire:

- **Documentazione OpenVPN:**
<http://openvpn.net/index.php/open-source/documentation.html>
- **Articolo completo sul mio blog:** <http://otacon22.wordpress.com/2011/01/08/guida-a-openvpn-per-gnulinux/>

Crittografia quantistica

Andrea Bontempi

<andreabont@gmail.com>

I Primi esempi di crittografia risalgono al 400 a.C. quando il messaggio veniva scritto lettera per lettera in colonne su un foglio arrotolato attorno ad un bastone, una volta srotolato il messaggio risulta illeggibile per chiunque non avesse a disposizione un bastone dallo stesso diametro (che fungeva da chiave). Da allora è partita l'eterna lotta tra crittografi e crittoanalisti, i primi che studiano come nascondere un messaggio, il secondi che studiano come leggerlo abusivamente. La crittografia quantistica potrebbe far propendere l'ago della bilancia a favore dei crittografi, rendendo impossibile (a detta dei sostenitori) violare una comunicazione protetta.

Le basi della teoria quantistica

La teoria quantistica nasce con gli studi di Planck sul corpo nero, che avevano lo scopo di evitare che il modello matematico prevedesse una emissione infinita di energia. Il problema venne risolto vedendo la luce come fatta di pacchetti d'onda,

quelli che noi chiamiamo fotoni. Ovvero la luce può essere emessa solo a particolari quantità discrete di energia. Un fotone è un oggetto quantistico, sia onda elettromagnetica che particella (Bosone mediatore della forza elettromagnetica nella teoria quantistica, privo di massa) ed è quindi soggetto al principio di indeterminazione di Heisenberg. Il principio afferma che è impossibile effettuare contemporaneamente, con precisione assoluta, più di una misura su un oggetto quantistico come il fotone, questo perché osservando il sistema lo si modifica. Ad esempio, cercando di conoscere la posizione esatta di un elettrone, ne andremo a modificare la velocità, la quale non potrà più essere misurata con la precisione voluta. Il fotone, comportandosi anche come onda elettromagnetica, può essere polarizzato trasversalmente, ovvero l'onda può oscillare in qualsiasi direzione che sia perpendicolare alla direzione di propagazione dell'onda stessa. Nel nostro caso useremo solo 4 tipi di polarizzazione: a 0° e 90° per avere la base + e a $+45^\circ$ e -45° per avere la base x.

Protocollo BB84

Questo protocollo venne pubblicato (come si intuisce dal nome) nel 1984 da Bennett e Brassard, e permette uno scambio sicuro di una chiave da utilizzare poi per cifrare il messaggio con un metodo classico. La forza di questo protocollo sta nel fatto che si usa un canale quantistico per generare casualmente una chiave condivisa tra Alice e Bob (le classiche figure che si vogliono scambiare un messaggio segreto) che non sia però accessibile, a causa del principio di Heisenberg, alla classica spia Eve. Il protocollo si basa sulla polarizzazione dei fotoni precedentemente accennata, si usano due filtri polarizzatori (Uno per base: $+$ e \times) che permette di capire quale è la direzione di polarizzazione del fotone (Nel caso di $+$ il filtro fa passare solo il fotone verticale, non quello orizzontale, o viceversa). Il trucco sta nel fatto che se io polarizzo un fotone con la base $+$ e cerco di misurare la polarizzazione con un filtro in base \times ottengo una misura perfettamente casuale, e non saprò mai quale è la sua reale polarizzazione, dato che una volta effettuata la misura il fotone viene perso. Scelgo quindi di dare un valore alle varie polarizzazioni: bit 0 per i fotoni verticali in base $+$ e per i fotoni a $+45^\circ$ in base \times , bit 1 per i fotoni orizzontali in base $+$ e per i fotoni a -45° in base \times . Posso quindi trasmettere dati in binario attraverso un canale quantistico, con questa procedura:

Base	0	1
$+$	\updownarrow	\leftrightarrow
\times	\nearrow	\nwarrow

1. Alice sceglie una sequenza casuale di

bit da inviare, e li invia a Bob usando per ogni bit una base scelta casualmente tra $+$ e \times .

- Bob ad ogni fotone sceglie casualmente quale base usare per la lettura e memorizza le varie misure, statisticamente avrà una parte di bit letti correttamente e una parte di bit casuali (fotoni letti con la base sbagliata).
- Bob comunica ad Alice la sequenza di basi da lui utilizzata per leggere i fotoni ricevuti (la comunicazione può avvenire pubblicamente) ma non le polarizzazioni misurate.
- Alice comunica a Bob quando ha usato una base sbagliata per la misura, quindi quali bit sono casuali e vanno scartati (senza comunicare la sequenza stessa di bit).
- Alice e Bob ora hanno una sequenza di bit condivisa che può essere usata come password per la comunicazione vera e propria, avendo la certezza che nessuno, tranne loro due, possiede quella particolare sequenza di bit.

Perchè BB84 è sicuro?

Eve, nel tentativo di intercettare la comunicazione sul canale quantistico è costretto a interrompere il flusso di fotoni, che vengono distrutti alla misura, questo permette a Bob di accorgersi della presenza di Eve, e di prendere provvedimenti. Inoltre Eve non conosce le basi utilizzate da Alice per polarizzare i fotoni, quindi, come

Bob, è costretto a scegliere casualmente una base e a provare, ma non avendo un riscontro diretto di quali basi siano giuste e quali siano sbagliate, può ottenere solo una sequenza casuale di bit (Bob, si spera che abbia avvisato Alice della presenza di Eve, che non risponderà ad una eventuale richiesta di Eve su quali basi fossero corrette e quali no, a meno che Alice non abbia voglia di scambiare qualche messaggio cifrato con Eve...) Inoltre è possibile fare dei controlli di integrità sulla chiave, controllando che un sottoinsieme casuale di bit coincida tra Alice e Bob (il sottoinsieme verrà poi scartato e non usato come chiave). Nel caso in cui le due chiavi non coincidano si ha un evidente indizio della presenza di Eve che cerca di intercettare la comunicazione.

Punti deboli

Il canale quantistico funziona grazie al principio di indeterminazione, il quale non permette l'amplificazione dei fotoni (ne causerebbe la modifica impedendo il corretto scambio di chiavi), questo relega le comunicazioni con questo protocollo fattibili solo su brevi distanze, anche se è parzialmente risolvibile con fibre ottiche sempre più evolute rimane un problema non da poco.

Il protocollo non implementa l'autenticazione, quindi è effettuabile l'attacco - Man in the middle, dove Alice si collega a Eve il quale si collega, separatamente, a Bob. Usando due distinti canali quantistici Alice e Bob non si accorgono dell'attacco in corso. Nella realtà il protocollo viene implementato attraverso quattro fo-

todiodi in cascata (uno per ogni polarizzazione possibile) che fungono da rilevatore, i quali, se colpiti da una luce troppo intensa, hanno il difetto di comportarsi come dei normali rilevatori. Il trucco sta proprio nel rendere il rilevatore cieco, in modo che non si accorga dei fotoni ricevuti. Eve intercetta i fotoni al posto di Bob e li legge scegliendo una base casuale, una volta letto il bit lo invia a Bob con la polarizzazione letta ma con un segnale molto più intenso in modo da confondere il rilevatore e obbligarlo a scegliere la base voluta. In questo modo Alice e Bob credono di avere una comunicazione sicura, ma nella realtà la comunicazione è gestita da Eve.

Link Utili

Per approfondire:

- **Demo on-line che simula il protocollo BB84:** <http://fredhenle.net/bb84/>
- **Vulnerabilità nell'implementazione:** <http://goo.gl/Na8gD>
- http://it.wikipedia.org/wiki/Crittografia_quantistica

CAD e Linux

Paolo Redaelli

<paolo.redaelli@gmail.com>

LINUX ormai dispone di programmi per tutte le necessità: dalla nascita del progetto GNU ad oggi i programmi liberi hanno coperto in modo efficace innumerevoli bisogni, dai nuclei ai compilatori fino agli ambienti da scrivania più completi. Purtroppo la disponibilità di strumenti liberi per il disegno e la progettazione CAD (Computer Aided Design -) è sempre stata piuttosto scarsa e le alternative disponibili per lungo tempo sono state poco soddisfacenti nonostante gli sforzi profusi. Numerosi sono i fattori che hanno generato e mantenuto questa carenza; in primo luogo i programmi liberi, specie quelli di maggior successo, sono spesso stati scritti per soddisfare i bisogni dei programmatori che li creavano: i primi utenti di un compilatore, di un editor sono i programmatori stessi mentre molti dei server web più diffusi sono usati in prima persona dai gruppi che li hanno realizzati; gli utenti dei CAD raramente sono anche programmatori - spesso sono architetti, ingegneri civili, edili e meccanici - e le conoscenze necessarie per un programma di disegno sono ampiamente multisettoriali - informatica, matematica, ingegne-

ria; con l'esclusione del settore elettronico sono ben pochi i settori dove le capacità di programmazione convive con la necessità di usare estensivamente programmi CAD.

I formati

DWG e DXF

Un'ulteriore barriera all'ingresso nel settore CAD di programmi liberi è stata per lungo tempo il formato dei dati: sin dagli esordi dei PC negli anni '80 il formato più diffuso è stato il DWG della Autodesk. Questo formato binario è sempre stato - ed è tuttora - proprietario e non ufficialmente documentato; questo rende necessario l'ingegnerizzazione inversa dei dati, pratica usualmente molto complicata e resa ancora più difficoltosa dalle innumerevoli versioni del formato stesso: ben 18 revisioni si sono succedute negli anni. Ad onor del vero Autodesk - leader indiscussa del mercato dei CAD - ha sempre affiancato al formato binario un formato di interscambio, il DXF (Drawing eXchange Format), per assicurare l'interoperabilità dei dati. Anche questo formato è però rimasto non documentato per molto tempo mentre ora Autodesk ne pubblica le specifiche relative alle versioni da AutoCAD

Release 13 ad AutoCAD 2010. Nonostante il fatto che DXF sia stato originariamente pensato per essere un testo ASCII in chiaro la mancanza di specifiche ha reso meno agevole l'effettivo interscambio e la creazione di programmi che leggessero anche solo questo formato. In modo simile ad un sistema operativo proprietario ben conosciuto la diffusione di AutoCAD è stata favorita negli anni da una politica di sostanziale tolleranza nei confronti delle copie illecite del programma da parte degli studenti che una volta divenuti professionisti trovavano più comodo continuare ad utilizzare lo stesso strumento nonostante la presenza di alternative. I concorrenti di AutoDesk - tra i più noti Bentley Systems, IntelliCAD, Intergraph, Nemetschek, SolidWorks - non riuscendo negli anni a scalzare AutoCAD ed il suo formato DWG hanno creato nel 1998 la OpenDWG Alliance con lo scopo di realizzare una libreria - sempre proprietaria - per la lettura e la scrittura di file DWG. Questa organizzazione non-profit ha cambiato nome in Open Design Alliance in seguito a varie iniziative di AutoDesk che considerava la dicitura DWG un marchio di sua proprietà; anche la libreria prodotta dal consorzio successivamente rinominata DWGdirect ha dovuto essere nuovamente rinominata come Teigha. Anche la Free Software Foundation ha sentito la necessità di creare una libreria simile: il progetto LibreDWG - considerato prioritario - è partito nel 2009 e nonostante sia ancora considerato di qualità alpha permette la lettura di molti disegni DWG in formato da R13 fino alla release 2004.

IGES e STEP

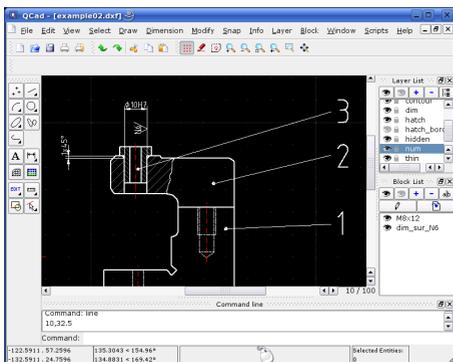
L'Initial Graphics Exchange Specification (IGES) è un formato neutrale per CAD originariamente sviluppato da Boeing, General Electric, Xerox ed altri per vari enti del governo statunitense; in seguito divenuto standard ANSI è adottato da praticamente tutti i CAD di alto livello a causa del requisito del dipartimento della difesa USA di supportarlo in lettura e scrittura. Nonostante le specifiche aperte non è mai stato particolarmente diffuso nel settore privato probabilmente per la maggior dimensione degli archivi.

Oltre alla distinzione fra programmi liberi e proprietari che tanto è cara al POUL i programmi CAD si dividono principalmente in software per il disegno bidimensionale o per la modellazione tridimensionale: i primi tendono ad essere una riproposizione informatica di strumenti tradizionali come riga, squadra e compasso e lasciano sempre trasparire questa impostazione mentre i secondi inevitabilmente abbandonano l'analogia cartacea per approcciarsi alla modellazione in modo più adatto ai requisiti di un ambiente tridimensionale con una logica orientata alla composizione dei solidi con operazioni booleane e tridimensionali che non hanno equivalenti nelle tradizionali tecniche limitate alla superficie del foglio da disegno.

CAD Bidimensionali

QCAD

QCAD della RibbonSoft è stato fra i primi CAD liberi ad essere ampiamente utilizzato; limitato alle due dimensioni usa come formato nativo il DXF e le librerie QT per l'interfaccia utente. Rilasciato con licenza proprietaria per le versioni 3 beta e 2.2 e GNU General Public License per la 2.0 è disponibile da molti anni - nella sua variante libera - per tutte le principali distribuzioni (Debian, Ubuntu, Fedora). Interfaccia utente e modalità di utilizzo sono molto simili a quella di AutoCAD LT e per questo risulta essere di facile utilizzo; la limitazione al solo disegno 2D rende tutto il pacchetto particolarmente leggero sia per gli standard odierni che rispetto alle alternative.



DraftSight

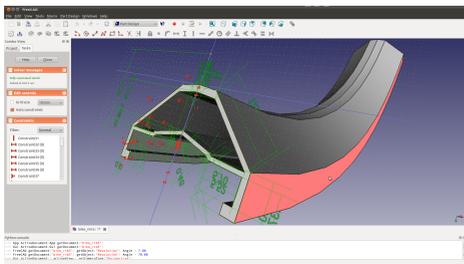
CAD bidimensionale gratuito ma proprietario è destinato ad ingegneri, architetti, progettisti, disegneri, studenti e docenti. Sviluppato Dassault Systèmes permette la creazione di file DWG e DXF. Come quasi

tutti i CAD 2D offre un sistema di coordinate cartesiane, un'interfaccia a linea di comando che integra la rappresentazione grafica del disegno e la suddivisione del disegno in livelli. Rispetto a QCAD offre un livello di compatibilità con Autocad molto più elevato.

CAD Tridimensionali

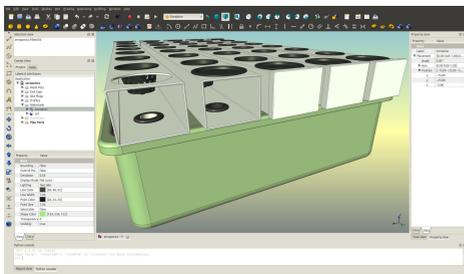
FreeCAD

FreeCAD è un modellatore CAD generico, rilasciato con licenze GNU General Public License e Lesser General Public License, destinato principalmente all'ingegneria meccanica ed alla progettazione prodotti sebbene sia utilizzabile con profitto anche in ambito architettonico e di ingegneria civile. Decisamente orientato alla modellazione 3D ha caratteristiche che fanno oltre quelle dei CAD tradizionali rendendolo un Computer-aided engineering (CAE) come equivalenti proprietari quali CATIA, SolidWorks e Solid Edge: la parte di modellazione solida è pienamente parametrica e la sua architettura modulare permette l'aggiunta di funzionalità ulteriori senza modifiche al progetto originale. Come altri modellatori solidi o CAD 3D ha componenti per il disegno bidimensionale necessari per creare sezioni e prospetti dei modelli destinati alla stampa ma il solo disegno 2D - come in Autocad LT - non è disponibile.



FreeCAD sfrutta pesantemente tutta una serie di software libero disponibile per la computazione scientifica come il potente nucleo CAD fornito dal progetto OpenCASCADE, Coin3D (un'implementazione di Open Inventor), le librerie Qt per l'interfaccia utente ed il popolare linguaggio di scripting Python per la programmabilità. Estremamente promettente nonostante sia ancora considerato di qualità alpha è disponibile per Linux, Windows e MacOS. Potete installarlo su Ubuntu con i seguenti comandi:

```
sudo add-apt-repository ppa:
    freecad-maintainers/freecad-
    daily
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install freecad
    freecad-doc
```

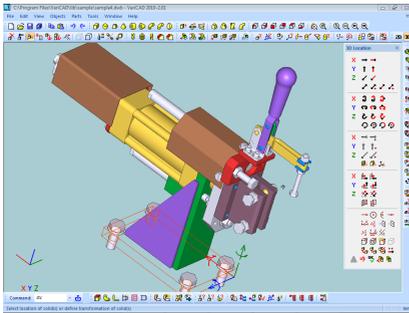


Bricscad

Sviluppato da Bricsys Bricscad è un pacchetto proprietario basato fino alla versione 10 sul motore di IntelliCAD; è uno dei pochi software proprietari disponibile con supporto commerciale per Linux. Utilizza le librerie della Open Design Alliance per leggere e scrivere il formato DWG ed implementa i linguaggi di scripting AutoLISP, VBA e BRX, tutti introdotti da AutoCAD. Di caratteristiche simili a quelle di AutoCAD fino al 2010 l'unica versione nativa era quella Windows che però girava con supporto ufficiale anche su Linux tramite Wine. Da settembre 2010 Bricsys ha rilasciato una versione nativa per Linux. È disponibile al download e senza licenza-chiave funziona per 30 giorni. A seconda delle opzioni richieste la licenza varia da 280€ a 515€.

VariCAD

VariCAD è un CAD bi-tridimensionale orientato alla progettazione meccanica sviluppato sin dal 1988. Dispone di supporto a vincoli geometrici e parametrici, strumenti per gusci, tubrazioni, piegature di lamiere, gestione assemblaggi di parti meccaniche, computo metrico ed altro. Fornito con una estesa libreria di viti, bullone ed altre parti meccaniche offre supporto nativo al formato DWG sempre attraverso le librerie della Open Design Alliance; supporta anche il formato STEP. Disponibile per il download in versione prova per 30 giorni si propone come potente ma abbordabile per la modica cifra di 499€.



Quale scegliere?

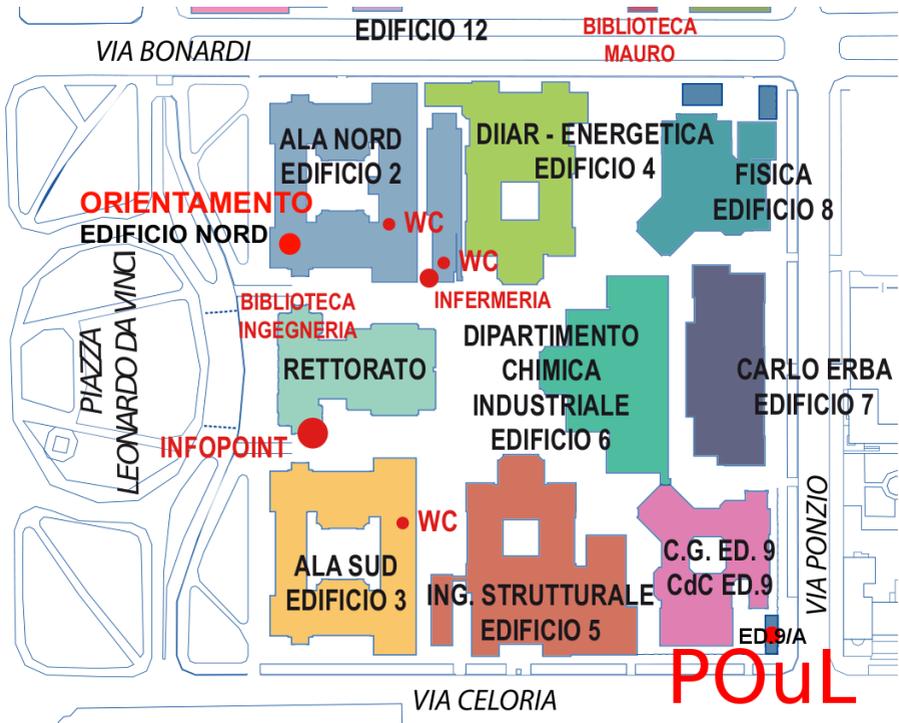
Fino a non molto tempo fa l'unica alternativa effettivamente praticabile per gli studenti era Qcad nonostante la forte limitazione dell'uso dei soli DXF. Ad oggi DraftSight con il suo modello commerciale basato sul pagamento dell'assistenza sembra essere l'unica soluzione di compromesso accettabile per poter lavorare in campo architettonico/ingegneristico con sistemi operativi liberi sempre di potersi limitare al solo disegno bidimensionale. In caso contrario sarete costretti ad metter mano al portafoglio per poter usare Bricscad e Varicad: software proprietari a tutti gli effetti ma notevolmente più abbordabili di AutoCAD e di pari livello. FreeCAD nonostante la giovane età del progetto poggia sulle solidissime basi fornite dalle librerie OpenCASCADE e potrebbe diventare nel breve periodo una soluzione validissima per l'uso professionale. L'offerta di programmi per la progettazione per Linux è in lento ma costante miglioramento: spesso è ancora necessario affidarsi a programmi proprietari gratuiti per interfacciarsi con un mondo - quello della progettazione - che neppure conosce il concetto di software

libero e tuttavia una nuova generazione di strumenti liberi è sul punto di essere usabile anche dal grande pubblico. Progetti come FreeCAD e LibreDWG hanno ancora bisogno di sviluppo, test e documentazione! Hanno bisogno anche del nostro aiuto!

Link Utili

Per approfondire:

- <http://en.wikipedia.org/wiki/.dwg>
- http://en.wikipedia.org/wiki/AutoCAD_DXF
- <http://opendesign.com/>
- <http://www.gnu.org/software/libredwg/>
- <http://www.qcad.org/>
- <http://free-cad.sourceforge.net/>
- <http://www.3ds.com/products/draftsight/free-cad-software/>
- <http://en.wikipedia.org/wiki/Bricscad>
- <http://en.wikipedia.org/wiki/VariCAD>



Vi è venuta voglia di conoscere il mondo di Linux? Volete partecipare più da vicino alle nostre attività? Volete scrivere un articolo su questa rivista? Iscrivetevi alla nostra mailing list oppure venite a trovarci nella nostra sede presso l'edificio 9/A!

sito Internet: www.poul.org
 informazioni: info@poul.org



La stampa della rivista è interamente finanziata dal Politecnico di Milano, che non si assume alcuna responsabilità sul contenuto.

Stampa a cura di GRAPHIC WORLD snc, Fizzonasco (MI), 2011.