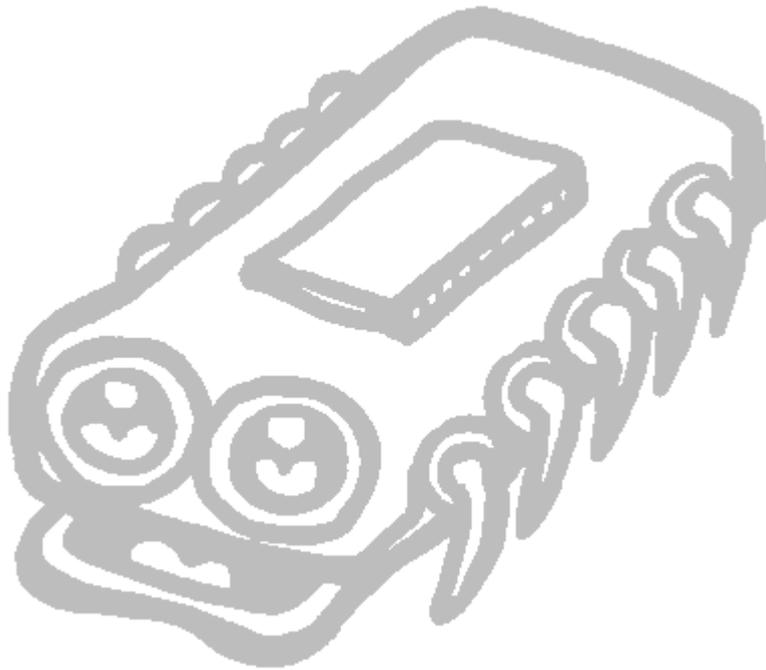


Hackmeeting 2000

CSOA Forte Prenestino

16-17-18 giugno 2000



Hackmeeting 2000 - Intro

L'Hackmeeting 2000 si terrà il 16, 17, 18 Giugno al CSOA Forte Prenestino di Roma (<http://www.forteprenestino.net> ; <http://www.hackmeeting.org>).

Negli ultimi mesi si è acceso nel mondo dell'informazione un grande interesse per l'hacking. E' un fenomeno sociale nuovo anche se ha le sue radici negli anni '60. Vi è però una profonda distorsione in quello che troppo spesso si dice.

L'hacking non è semplicemente l'intrusione distruttiva nei siti. Hackers erano gli informatici del MIT che creavano autonomamente un sistema operativo alternativo a quello venduto commercialmente sui primi PDP 10 nel '67, Hackers sono Richard Stallman (il creatore della Free Software Foundation), Linus Torvalds (il creatore del sistema operativo Linux), Lee Felsenstein (uno dei visionari ideatori del personal computer), Steve Wozniac (inventore e costruttore della Apple), e migliaia di sistemisti e programmatori che hanno costruito il software e i protocolli che fanno girare i computer di tutto il mondo.

Gli hackers sono coloro che hanno dato la possibilità ad ognuno di noi di accendere un computer personale nelle nostre case e di usarlo per scrivere o per inviare posta o per comunicare in generale. Internet stessa e' opera degli Hackers.

La criminalizzazione a cui spesso assistiamo viene dalla mancanza di conoscenza di un fenomeno sociale complesso. La Criminalizzazione nega come siano in genere hackers che segnalano i buchi di sicurezza nei sistemi pubblici e commerciali, nega che la maggior parte del software sulla rete nasce dal loro lavoro. L'equazione Hackers=Pirata è priva di significato, non perché stiamo dicendo che gli hackers sono "buoni", invece che "cattivi", ma perché questa semplificazione nega l'emergere di quella che viene chiamata "Società dell'Informazione", nega che fenomeni come la *new economy* (spesso osannata con troppa approssimazione) sono opera del lavoro a volte oscuro ed a volte eclatante di migliaia di hackers.

Alla domanda "Cosa e' un Hacker?" si puo' rispondere soltanto in un modo, una persona che ha una attitudine alla ricerca della conoscenza sulla e per mezzo della tecnologia e al suo uso libero da ogni tipo di Copyright.

Alla domanda "Cosa e' il movimento Hacker?" si deve invece rispondere che il movimento è basato sulla condivisione di sapere e sulla libertà di utilizzo della tecnologia e di tutte le sue forme.

Un Hacker è chi crea sapere, chi lo diffonde, chi lo cerca, a prescindere dai suoi scopi personali, giusti o sbagliati, legali o illegali, esclusivi o pubblici che siano.

Il movimento Hacker è formato invece da persone che vogliono scoprire il sapere e che lo vogliono condividere e rendere libero e accessibile a chiunque.

Hackmeeting è il meeting del Movimento Hacker Italiano, e per questo prende una connotazione di condivisione e di creazione del sapere per tutti e tutte.

Sommario:

<i>Hackmeeting 2000 - Intro</i>	1
<i>Sommario:</i>	4
<i>PROCLAMA HACKMEETING 2000</i>	5
<i>L'IDEA HACKER DI COMUNICAZIONE</i>	6
<i>L'ICATA 89 (International Conference on the alternative use of technology)</i>	7
<i>BREVE STORIA SUGLI HACKER di Eric S. Raymond</i>	10
<i>CALENDARIO SEMINARI HACKMEETING</i>	17
<i>DESCRIZIONE SEMINARI:</i>	19
VENERDI' 16 GIUGNO 2000	19
SABATO 17 GIUGNO 2000	39
<i>INSTALLAZIONI:</i>	63
<i>CONCERTI SERALI:</i>	64
Venerdi'	64
Sabato	64

PROCLAMA HACKMEETING 2000

La terza edizione dell'HACKMEETING avrà luogo il 16, 17, 18 Giugno 2000 a Roma, nel Centro Sociale Occupato e Autogestito Forte Prenestino (Via F.Delpino a Centocelle, Tram 14-19-516, URL: www.forteprenestino.net).

Dando continuità ai precedenti appuntamenti del 1998 a Firenze (CPA, Centro Popolare Autogestito) e del 1999 a Milano (Laboratorio Studentesco Deposito Bulk), questi tre giorni saranno un territorio libero di comunicazione, informazione, socializzazione, sperimentazione.

L'hackmeeting, è un evento collettivo autogestito e autorganizzato attraverso un'apposita mailing list (hackmeeting@kyuzz.org).

E' un incontro tra persone che hanno voglia di condividere le proprie esperienze e conoscenze e che si battono per una comunicazione telematica orizzontale.

E' un'occasione di aggiornamento tecnico di alto livello a costo quasi nullo, proprio per lo spirito di condivisione che permea i partecipanti.

E' uno stimolo per discutere su molti temi importanti tra i quali le possibilità di sviluppo del free software, i diritti in rete, la cooperazione sociale.

L'Hackmeeting è un incontro per chi concepisce la tecnologia come qualcosa di smontabile e ricomponibile, su cui agire consapevolmente e collettivamente, su cui mettere le mani, per chi crede che essere un hacker è un'attitudine.

Il 16, 17, 18 Giugno grazie alla freenet del Forte Prenestino ci sarà la connessione gratuita 24 ore su 24. Ci saranno numerosi seminari autogestiti e autorganizzati, concerti serali, campeggio libero, cene e pranzi collettivi, occasioni di incontrare vecchi e nuovi amici.

Tre giornate per collettivizzare le proprie idee sulla tecnologia e sul suo utilizzo pratico, politico e sociale, ma anche tre giorni (e tre notti) di gioco e contaminazione.

URL di riferimento: <http://www.hackmeeting.org>

Assemblea dei promotori di Hack00

L'IDEA HACKER DI COMUNICAZIONE

L'idea hacker di comunicazione è opposta a quella sottostante al medium televisivo: orizzontale, rizomatica, decentrata, non gerarchica né autoritaria, non controllata né censurata dove diventa possibile scambiarsi saperi in modo paritario.

L'idea hacker di comunicazione è anche una visione diversa e critica della tecnologia: non più pensata per pochi "sacerdoti" , ma comprensibile, smontabile e ricomponibile per adattarla a fini individuali e collettivi.

L'idea hacker di comunicazione prevede altresì una condivisione dei saperi e delle tecnologie: la grande opportunità rappresentata dai programmi di software libero, che creati collettivamente in rete consentono già oggi a milioni di utilizzatori di sottrarsi al giogo economico del software commerciale e possono diventare un'opportunità di occupazione e di organizzazione nuova della produzione.

L'idea hacker di comunicazione pone le basi per pratiche condivise di intelligenza collettiva, che sappiano amplificare le risorse degli individui verso finalità di bene comune.

L'idea hacker di comunicazione è' alla base del concetto di Lavoro in Rete, una pratica che non ha frontiere, ne' conosce etnie e che quindi per sua natura non può far altro che opporsi attivamente a qualsiasi logica di guerra.

Assemblea dei promotori dell'Hack IT 99
(Laboratorio Studentesco Deposito Bulk - Milano)

IL PRIMO HACKMEETING INTERNAZIONALE:

L'ICATA 89

(International Conference on the alternative use of technology)

DICHIARAZIONE FINALE DELL'ICATA 89

Adottata il 4/8/89

In questa dichiarazione programmatica finale, l'intera scena hacker internazionale ha concordato nell'agosto 1989 su alcuni principi base, al fine di riaffermare la propria pratica e di spezzare la canea montante repressiva, in corso contro di essi in quasi tutti i paesi del mondo. E' interessante notare che la pratica dell'hackeraggio viene letta come necessaria per infrangere il monopolio statale e delle multinazionali sull'informazione. Questo dominio, difatti, suona tanto più strano, se confrontato con l'oggettiva democraticità del mezzo "computer".

Noi, cittadini planetari e partecipanti alla FESTA GALATTICA DEGLI HACKERS e dell'ICATA 89 ad Amsterdam, abbiamo confrontato, durante tre giorni, le nostre idee, le nostre esperienze, le nostre speranze e rispettivi scopi per l'avvenire. Profondamente turbati dalla prospettiva di una tecnologia dell'informazione e degli attori economici e politici scatenati da essa, senza controllo democratico né partecipazione popolare efficace, noi abbiamo risoluto che:

Lo scambio libero e senza alcun ostacolo dell'informazione sia un elemento essenziale delle nostre libertà fondamentali e debba essere sostenuto in ogni circostanza. La tecnologia dell'informazione deve essere a disposizione di tutti e nessuna considerazione di natura politica, economica o tecnica debba impedire l'esercizio di questo diritto.

Tutta intera la popolazione debba poter controllare, in ogni momento, i poteri del governo; la tecnologia dell'informazione deve allargare e non

ridurre l'estensione di questo diritto.

L'informazione appartiene a tutto il mondo, essa è prodotta per tutto il mondo. Gli informatici, scientifici e tecnici, sono al servizio di tutti noi. Non bisogna permettere loro di restare una casta di tecnocrati privilegiati, senza che questi debbano rendere conto a nessuno del loro operato.

Il diritto all'informazione si unisce al diritto di scegliere il vettore di questa informazione. Nessun modello unico di informatizzazione deve essere imposto a un individuo, una comunità o a una nazione qualsiasi. In particolare, bisogna resistere alle pressioni esercitate dalle tecnologie "avanzate" ma non convenienti. Al loro posto, bisogna sviluppare dei metodi e degli equipaggiamenti che permettano una migliore convivialità, a prezzi e domanda ridotti.

La nostra preoccupazione più forte è la protezione delle libertà fondamentali; noi quindi domandiamo che nessuna informazione di natura privata sia stockata, né ricercata tramite mezzi elettronici senza accordo esplicito da parte della persona interessata. Il nostro obiettivo è di rendere liberamente accessibile i dati pubblici, proteggere senza incertezze i dati privati. Bisogna sviluppare delle norme in questo senso, insieme agli organismi e alle persone interessati.

Ogni informazione non consensuale deve essere bandita dal campo dell'informatica. Sia i dati che le reti devono avere libertà d'accesso. La repressione dei pirati deve divenire senza fondamento, alla maniera dei servizi segreti.

Parallelamente domandiamo che tutte le legislazioni, in progetto o già in applicazione, rivolte contro i pirati e che non perseguono scopi criminali o commerciali, siano ritirati immediatamente.

L'informatica non deve essere utilizzata dai governi e dalle grandi imprese per controllare e opprimere tutto il mondo. Al contrario, essa deve essere utilizzata come puro strumento di emancipazione, di progresso, di formazione e di piacere. Al contempo, l'influenza delle istituzioni militari sull'informatica e la scienza in generale deve cessare.

Bisogna che sia riconosciuto il diritto d'avere delle connessioni senza alcuna restrizione con tutte le reti e servizi internazionali di comunicazione di dati, senza interventi e controlli di qualsiasi sorta.

Bisogna stabilire dei tetti di spesa, per paese, per avere accesso a questi vettori di comunicazione di dati pubblici e privati. Si deve facilitare quei paesi senza una buona infrastruttura di telecomunicazione e la loro partecipazione nella struttura mondiale.

Noi ci indirizziamo agli utilizzatori progressisti di tecnologie di informazione nel mondo affinché socializzino le loro conoscenze e specializzazioni in questo campo con delle organizzazioni di base, al fine di rendere possibile uno scambio internazionale e interdisciplinare di idee e informazioni tramite delle reti internazionali.

Ogni informazione è al contempo deformazione. Il diritto all'informazione è al contempo inseparabilmente legato al diritto alla deformazione, che appartiene a tutto il mondo. Più si produce informazione, e più si crea un caos di informazione sfociante sempre più in rumore. La distruzione dell'informazione come del resto la sua produzione, è il diritto inalienabile di ognuno.

Bisognerebbe sovvertire i canali regolamentari e convenzionali dell'informazione grazie a dei detournaments e dei cambiamenti surrealisti degli avvenimenti, al fine di produrre del caos, del rumore, dello spreco i quali, a loro volta, saranno considerati come portatori di informazione.

La libertà di stampa deve applicarsi anche alle pubblicazioni tecno-anarchiche, che appaiono in giro, per reclamare la liberazione dei popoli, la fine delle tirannie della macchina e del sistema sugli uomini.

BREVE STORIA SUGLI HACKER

di Eric S. Raymond

Prologo: i Real Programmer

In principio furono i Real Programmer. Non è così che si definirono. Ma neanche "hacker", o qualcosa in particolare; il nomignolo "Real Programmer" fu coniato solo dopo il 1980. Fin dal 1945, ad ogni modo, la tecnologia informatica ha attirato molte delle menti più brillanti e creative del pianeta. A partire dall'ENIAC di Eckert e Mauchly in avanti, è sempre esistita cultura tecnica più o meno continua e autocosciente di programmatori entusiasti, di persone il cui rapporto col software era di puro divertimento.

I Real Programmer di solito provenivano dai settori dell'ingegneria e della fisica. Indossavano calzini bianchi, camicie e cravatte in poliestere e lenti spesse, programmavano in linguaggio macchina, in FORTRAN e in un'altra mezza dozzina di linguaggi ormai dimenticati. Si tratta dei precursori della cultura hacker, dei ben poco celebrati protagonisti della sua preistoria.

Dalla fine della Seconda Guerra Mondiale ai primi anni '70, nella grande era dei computer a linea di comando e dei mainframe chiamati "big iron", i Real Programmer dominarono la scena della cultura tecnica dei computer. Alcuni articoli del venerato folklore degli hacker risalgono proprio a quegli anni: la famosa storia di Mel (inclusa nel Jargon File), vari passi di Murphy's Laws (La legge di Murphy) e il manifesto caricatura tedesco "Blinkenlights" che ancora adorna molte sale di computer.

Molte persone formatesi nella cultura "Real Programmer", rimasero attive anche negli anni '90. Si narra che Seymour Cray, progettista della linea di supercomputer Cray, abbia una volta trasportato un intero sistema operativo di sua concezione in un computer di sua creazione. Col sistema ottale e senza un errore. Funzionò tutto alla perfezione. Macho supremo dei Real Programmer.

Su scala minore, Stan Kelly-Bootle, autore di *The Devil's DP Dictionary* (McGraw-Hill, 1981) e straordinario folklorista, programmò sul Manchester Mark I nel 1948, il primo computer digitale completamente operativo. Oggi scrive strisce tecnico-umoristiche per riviste di informatica, che spesso hanno il sapore di un energico e complice dialogo con la odierna cultura hacker.

Altri, come David E. Lundstrom, hanno scritto aneddoti su quei primi anni (*A Few Good Men From UNIVAC*, 1987).

Ciò che, comunque, ebbe origine dalla cultura "Real Programmer", è lo slancio innovativo che investì il computer interattivo, le università e le reti. Elementi che hanno avuto un ruolo fondamentale nella nascita di una tradizione tecnica che sarebbe sfociata nell'attuale cultura hacker Open Source.

I primi hacker

L'origine della cultura hacker, come oggi la conosciamo, può essere fatta risalire al 1961, anno in cui il MIT acquistò il primo PDP-1. Il comitato Signals and Power del Club Tech Model Railroad del MIT, adottò la macchina quale prediletto giocattolo-tecnologico creando strumenti di programmazione, linguaggi e quell'intera cultura che ancora oggi ci appartiene in modo inequivocabile. Questi primi anni sono stati esaminati nella prima parte del libro *Hackers* di Steven Levy (*Anchor/Doubleday*, 1984).

La cultura informatica del MIT sembra essere stata la prima ad adottare il termine "hacker". Gli hacker della TMRC divennero il nucleo dell'Artificial Intelligence Laboratory (Laboratorio di Intelligenza Artificiale) del MIT, il principale centro di ricerca AI (Intelligenza Artificiale) su scala mondiale, nei primi anni '80. La loro influenza si protrasse ben oltre il 1969, il primo anno di ARPAnet.

ARPAnet è stata la prima rete transcontinentale di computer ad alta velocità. Ideata e realizzata dal Ministero della Difesa statunitense come esperimento nelle comunicazioni digitali, crebbe fino a diventare un collegamento tra centinaia di università, esponenti della difesa e laboratori di ricerca. Permise a tutti i ricercatori, ovunque essi si trovassero, di scambiarsi informazioni con velocità e flessibilità senza precedenti, dando un forte impulso allo sviluppo del lavoro di collaborazione e accelerando enormemente il ritmo e l'intensità del progresso tecnologico.

Ma ARPAnet fece anche qualcos'altro. Le sue autostrade elettroniche misero in contatto gli hacker di tutti gli Stati Uniti e questi, finora isolati in sparuti gruppi, ognuno con la propria effimera cultura, si riscoprirono (o reinventarono) nelle vesti di vera e propria tribù di rete.

Le prime intenzionali azioni di hackeraggio - i primi linguaggi caratteristici, le prime satire, i primi dibattiti autocoscienti sull'etica hacker - tutto questo si propagò su ARPAnet nei suoi primi anni di vita. (Basti come esempio la prima versione del Jargon File, datato 1973.) La cultura hacker mosse i primi passi nelle università connesse alla Rete, in particolar modo (ma non esclusivamente) nei loro dipartimenti di scienza informatica.

Dal punto di vista culturale, l'AI (Intelligenza Artificiale) Lab del MIT è da considerarsi il primo tra laboratori di pari natura a partire dai tardi anni '60. Anche se istituti come il Laboratorio di Intelligenza Artificiale

dell'Università di Stanford (SAIL) e, più tardi, l'Università Carnegie-Mellon (CMU), divennero in seguito quasi altrettanto importanti. Tutti costituivano fiorenti centri di scienza dell'informazione e ricerca sull'intelligenza artificiale. Tutti attiravano individui brillanti che contribuirono al grande sviluppo del mondo degli hacker, sia dal punto di vista tecnico che folkloristico.

Per comprendere ciò che successe dopo, comunque, è necessario un ulteriore sguardo ai computer stessi, poiché sia la nascita del Laboratorio che il suo futuro declino furono fortemente influenzati dalle correnti di cambiamento nell'ambito della tecnologia informatica.

Fin dai giorni del PDP-1, le sorti dell'hacking si intrecciarono alla serie di minicomputer PDP della Digital Equipment Corporation (DEC). La DEC aprì la strada a prodotti interattivi di stampo commerciale ed a sistemi operativi time-sharing. La flessibilità, la potenza e la relativa economicità di queste macchine, portarono molte università al loro acquisto.

L'economicità dei sistemi time-sharing, costituì l'habitat ideale per lo sviluppo della cultura hacker e anche ARPAnet fu costituita, per la maggior parte della sua durata, da una rete di macchine DEC.

La più importante fra queste fu il PDP-10 che fece la sua comparsa nel 1967. Essa rappresentò la macchina preferita dagli hacker per quasi quindici anni; il TOPS-10 (sistema operativo DEC per la macchina) e il MACRO-10 (suo assemblatore), sono ancora ricordati con passione nostalgica nell'ambito della cultura hacker.

Il MIT, pur utilizzando lo stesso PDP-10, imboccò una strada lievemente diversa; rifiutò il software DEC del PDP-10 scegliendo di creare un proprio sistema operativo, il leggendario ITS.

ITS stava per "Incompatible Timesharing System", (Sistema Time-Sharing Incompatibile), sigla che rendeva perfettamente l'idea delle intenzioni insite nel progetto: volevano fare a modo loro. Fortunatamente per tutti noi, la gente della MIT possedevano un grado di intelligenza in grado di contrastare la sua arroganza. L'ITS, strambo, eccentrico e a volte perfino pieno di difetti, portò tuttavia una brillante serie di innovazioni tecniche, e ancora detiene senza dubbio il record di sistema operativo time-sharing più a lungo utilizzato.

Lo stesso ITS fu scritto in Assembler, ma molti progetti ITS furono scritti nel linguaggio LISP dell'AI. Il LISP si rivelò il più potente e flessibile linguaggio dell'epoca e, a distanza di vent'anni, si presenta ancora meglio congegnato rispetto a molti dei linguaggi odierni. Il LISP permise agli hacker di ITS di dare libero sfogo a tutta la loro creatività. Fu forse questa la formula del successo straordinario di questo linguaggio, che resta uno dei preferiti dagli hacker.

Molte creazioni tecniche della cultura ITS, sopravvivono ancora oggi; l'editor Emacs è forse il più conosciuto. Così come molto del folklore ITS è tuttora "vivo" per gli hacker, come dimostra il Jargon File.

Non si può certo dire che il SAIL e il CMU si fossero nel frattempo assopiti. Molti nuclei di hacker, sviluppatasi intorno al PDP-10 del SAIL, divennero più tardi figure chiave nel progresso dei personal computer e delle interfacce di software finestra/icona/mouse, come oggi le conosciamo. Gli hacker di CMU, dal canto loro, stavano portando avanti ciò che avrebbe dato vita alle prime applicazioni pratiche su larga scala di sistemi esperti e di robotica industriale.

Un altro luogo che ha giocato un ruolo fondamentale per il progresso culturale fu lo Xerox PARC, il famoso Centro Ricerche di Palo Alto. Per più di un decennio, a partire dai primi anni '70 fino alla metà degli '80, il PARC produsse un'impressionante quantità di innovazioni hardware e software. Le moderne interfacce di software costituite da mouse, finestre e icone, videro la luce proprio in quell'ambito, ma anche le stampanti laser e la local area network (LAN). La serie PARC di macchine D, anticipò di un decennio i potenti personal computer degli anni '80. Purtroppo, questi profeti non ebbero né onori né gloria in seno alla loro azienda e presto diventò un'abitudine descrivere sarcasticamente il PARC come un luogo caratterizzato dallo sviluppo di brillanti idee per chiunque altro, tranne che per se stessi. L'influenza di queste menti sulla cultura hacker fu comunque a dir poco pervasiva.

Le culture ARPAnet e PDP-10 crebbero in forza e varietà nell'arco degli anni '70. I programmi per le mailing list elettroniche, utilizzati fino ad allora per incoraggiare la cooperazione tra i diversi gruppi di interesse disseminati a quattro angoli del mondo, furono sempre più impiegati per scopi sociali e ricreativi. DARPA chiuse deliberatamente un occhio di fronte alle attività tecniche "non-autorizzate", ben comprendendo come queste spese extra fossero un piccolo prezzo da pagare rispetto all'effetto di convogliare l'attenzione di un'intera generazione di menti giovani e brillanti alla causa dell'informatica.

Probabilmente, la più nota delle mailing list a sfondo "sociale" di ARPAnet fu la SF-LOVERS, per gli appassionati di fantascienza; basti pensare che ancora oggi essa continua ad esistere in "Internet", l'erede naturale e senza confini della rete ARPAnet. In questo scenario, si contano numerosi altri pionieri di questo stile di comunicazione che più tardi venne commercializzato in servizi time-sharing a pagamento come CompuServe, Genie e Prodigy.

La nascita di Unix

Nel frattempo, comunque, nel selvaggio New Jersey, qualcos'altro era stato messo in cantiere fin dal 1969, qualcosa che avrebbe inevitabilmente adombrato la tradizione del PDP-10. L'anno di nascita di ARPAnet, fu anche l'anno in cui un hacker dei Laboratori Bell, di nome Ken Thompson, inventò il sistema Unix.

Thompson si era trovato coinvolto nella fase di sviluppo di un Sistema Operativo Time-Sharing chiamato Multics, che divideva la propria discendenza con ITS. Multics costituì un importante banco di prova su come la complessità di un sistema operativo potesse essere celata fino a essere resa invisibile all'utente e perfino alla maggioranza dei programmatori. L'idea fu quella di rendere l'uso di Multics molto più semplice e programmabile in modo da permettere di operare anche dall'esterno.

I Laboratori Bell si tirarono fuori dal progetto quando Multics iniziò a mostrare segni di crescita non giustificata (il sistema fu poi commercializzato da Honeywell, senza successo). Ken Thompson cominciò ad avere nostalgia dell'ambiente Multics, e pensò di giocare un po' miscelando alcune caratteristiche del sistema operativo naufragato con altre di sua concezione su un rottame di DEC PDP-7.

Un altro hacker, di nome Dennis Ritchie, inventò un nuovo linguaggio chiamato "C", da usare con una versione Unix di Thompson ancora allo stato embrionale. Come Unix, C fu progettato per essere piacevole e facile da usare oltre che flessibile. L'interesse per questi strumenti non tardò a crescere nell'ambito dei Laboratori Bell, e subì un'impennata nel 1971 quando Thompson e Ritchie vinsero un appalto per produrre quello che oggi chiameremmo sistema di office-automation per uso interno. Ma Thompson e Ritchie avevano in mente qualcosa di ben più ambizioso.

Per tradizione, i sistemi operativi erano stati, fino ad allora, scritti in Assembler in modo da ottenere la maggiore efficienza possibile dalle macchine host. Thompson e Ritchie furono tra i primi a capire che la tecnologia dell'hardware e dei compilatori aveva raggiunto un tale livello di maturità da poter scrivere in C un intero sistema operativo: nel 1974 l'intero ambiente operativo era regolarmente installato su numerose macchine di diversa tipologia.

Si tratta di un evento senza precedenti e le implicazioni che ne derivarono furono enormi. Se davvero Unix poteva presentare la stessa interfaccia e le stesse funzionalità su macchine di diverso tipo, era sicuramente in grado di fungere da ambiente software comune per tutte. Gli utenti non avrebbero mai più dovuto pagare per nuovi software appositamente progettati ogni volta che una macchina diventava obsoleta. Gli hacker erano in grado di utilizzare gli stessi strumenti software da una macchina all'altra, piuttosto che dover reinventare l'equivalente di fuoco e ruota ogni volta.

Oltre alla portabilità, Unix e C presentavano altri punti di forza. Entrambi si basavano sulla filosofia "Keep it simple, stupid!" letteralmente "Semplifica, stupido!". Un programmatore poteva senza difficoltà tenere a mente l'intera struttura logica di C (a differenza di molti altri linguaggi precedenti, ma anche successivi), e non dover più ricorrere continuamente ai manuali. Unix era un insieme flessibile di semplici strumenti che si mostravano complementari l'un l'altro.

Questa combinazione si rivelò adatta per una vasta gamma di operazioni, incluse alcune completamente nuove, non previste in origine dagli stessi progettisti. La sua diffusione in AT&T fu estremamente rapida, a dispetto della mancanza di programmi di supporto formale. Entro il 1980, il suo uso si era già allargato a un gran numero di università e siti di ricerca informatica, e centinaia di hacker la consideravano come la propria casa.

Le macchine da lavoro della prima cultura Unix furono i PDP-11 e il loro discendente fu il VAX. Ma, proprio per la sua caratteristica portabilità, Unix funzionava senza alcuna modifica su una vasta gamma di macchine che costituivano ARPAnet. Nessuno usava l'Assembler, i programmi creati in C erano facilmente utilizzabili su tutte queste macchine.

Unix aveva persino una propria rete non certo di qualità eccelsa: Unix-to Unix Copy Protocol (UUCP), bassa velocità, poco affidabile ma economica. Due macchine Unix qualsiasi potevano scambiarsi posta elettronica point-to-point attraverso le ordinarie linee telefoniche. Questa funzionalità era parte integrante del sistema e non solo un'opzione. Le postazioni Unix cominciarono a formare una rete a se stante, e una cultura hacker iniziò a crescere al suo interno. È del 1980 la prima Usenet board, che sarebbe rapidamente diventata più grande di ARPAnet.

ARPAnet stessa ospitò alcuni siti Unix. PDP-10 e le culture Unix e cominciarono a incontrarsi e fondersi, anche se, dapprima, senza grande successo. Gli hacker di PDP-10 consideravano la gente di Unix come una banda di principianti che utilizzava strumenti dall'aspetto primitivo, se paragonati alla squisita e perfino barocca complessità di LISP e ITS. "Coltelli di pietra e pelli d'orso!" brontolavano.

Ecco allora che si delineò un terzo scenario. Il primo personal computer fu immesso sul mercato nel 1975. La Apple fu fondata nel 1977, e il suo progresso avvenne con impressionante rapidità negli anni che seguirono. Il potenziale dei microcomputer era ormai chiaro e attrasse inevitabilmente un'altra generazione di giovani e brillanti hacker. Il loro linguaggio era il BASIC, talmente primitivo che i partigiani del PDP-10, e gli aficionados di Unix lo considerarono subito indegno di qualsiasi considerazione.

La fine del tempo che fu

Ecco la situazione nel 1980: tre culture, simili ma organizzate intorno a diverse tecnologie. La cultura ARPAnet/PDP-10 sposata a LISP, MACRO, TOPS-10 e ITS. Il popolo di Unix e C, con i loro PDP-11, VAX e connessioni telefoniche di modesta entità. E infine un'orda anarchica di appassionati dei primi microcomputer, decisi a portare al popolo la potenza del computer.

Tra tutte queste, la cultura ITS poteva ancora rivendicare il posto d'onore. Ma sul Laboratorio si stavano addensando nubi minacciose. La tecnologia PDP-10 dipendente da ITS, cominciava a essere datata e il Laboratorio stesso era diviso in fazioni fin dai primi tentativi di commercializzazione della tecnologia AI. Alcune delle migliori menti del Laboratorio (e di SAIL e CMU), si erano lasciate attirare da lavori molto ben retribuiti presso società di nuova costituzione.

Il colpo di grazia fu inferto nel 1983, quando DEC cancellò la sua adesione al PDP-10 per concentrarsi sulle linee VAX e PDP-11. ITS non aveva più un futuro. In virtù della sua scarsa portabilità, infatti, l'idea di trasportarlo da un hardware all'altro era impensabile per chiunque. La variante funzionante su Unix di Berkeley VAX, divenne il sistema prediletto dagli hacker, e chiunque avesse rivolto lo sguardo al futuro, si sarebbe reso conto di quanto rapidamente crescesse la potenza dei microcomputer e con quale velocità avrebbero spazzato via tutto quello che li aveva preceduti.

Fu all'incirca in questo periodo che Levy scrisse Hackers. Una delle sue principali fonti di informazione fu Richard M. Stallman (inventore di Emacs), una figura chiave del Laboratorio e accanito oppositore della commercializzazione della tecnologia del Laboratorio.

Stallman (meglio conosciuto con le sue iniziali e login name, RMS), creò la Free Software Foundation, dedicandosi alla produzione di free software di alta qualità. Levy lo elogiò quale "ultimo vero hacker", una descrizione che si rivelò fortunatamente errata.

Il grandioso progetto di Stallman riassunse chiaramente la transizione che subì la cultura degli hacker nei primi anni '80: nel 1982 egli iniziò la costruzione di un intero clone di Unix, scritto in C e disponibile gratuitamente. Si può quindi dire che lo spirito e la tradizione di ITS furono preservati come parte importante della più nuova cultura hacker, incentrata su Unix e VAX.

Sempre in quello stesso periodo, la tecnologia dei microchip e della local area network iniziarono a fare presa sul mondo degli hacker. Ethernet e il microchip Motorola 68000 costituirono una combinazione teoricamente molto potente e solo dopo numerosi tentativi si arrivò alla prima generazione di ciò che oggi conosciamo come workstation.

Nel 1982, un gruppo di hacker Unix di Berkeley, fondò Sun Microsystems con la convinzione che Unix funzionante su un hardware con base 68000, relativamente economico, sarebbe stata la combinazione vincente per una grande varietà di applicazioni. La previsione si rivelò esatta e la loro intuizione definì il modello che l'intera industria avrebbe seguito. Sebbene i loro prezzi non erano ancora alla portata della maggior parte degli utenti, le workstation erano relativamente economiche per università e grandi aziende. Reti formate da questa nuova generazione di computer (uno per utente), sostituirono rapidamente gli ormai sorpassati VAX e altri sistemi time-sharing.

L'era del free Unix

Quando nel 1984 la AT&T iniziò ad essere svenduta e Unix divenne per la prima volta un prodotto commerciale, il mondo degli hacker si divideva in una "network nation", relativamente coesiva e centrata su Internet e Usenet, in cui venivano per lo più usati minicomputer o workstation funzionanti con Unix, e una vasta ma disorganizzata "hinterland" di appassionati di microcomputer.

La classe di macchine workstation costruite da Sun e da altri, aprì nuovi orizzonti agli hacker. Queste erano concepite per realizzare grafica di livello professionale e trasferire e gestire dati condivisi attraverso una rete. Nel corso degli anni '80, il mondo degli hacker si mostrò attento alle sfide di software e strumenti per sfruttare al massimo queste caratteristiche. Il gruppo Unix di Berkeley sviluppò un supporto integrato per i protocolli ARPAnet che offriva una soluzione al problema delle reti favorendo un'ulteriore crescita di Internet. Numerosi furono i tentativi di semplificare l'uso degli strumenti di grafica delle workstation. Il sistema che prevalse fu l'X Window System. Uno dei fattori che determinarono il suo successo fu dato dalla disponibilità dei suoi sviluppatori a fornire gratuitamente i sorgenti, secondo l'etica hacker, e a distribuirli tramite Internet. La vittoria di X sui sistemi di grafica proprietari (incluso quello offerto dalla stessa Sun), fu un'importante messaggio di cambiamento che, pochi anni dopo, avrebbe profondamente influenzato lo stesso Unix.

La rivalità tra ITS e Unix generava ancora qualche occasionale manifestazione di collera faziosa (per lo più proveniente dalla parte dei sostenitori dell'ex-ITS). L'ultima macchina ITS cessò comunque di funzionare per sempre nel 1990. I suoi partigiani si ritrovarono senza più un posto dove stare e furono in larga parte assimilati dalla cultura Unix non senza lamentele.

Nell'ambito degli hacker della rete, la grande rivalità negli anni '80 era tra i sostenitori della versione Unix di Berkeley e quella di AT&T. Sono ancora oggi reperibili copie di un manifesto di quel periodo che riportava un combattente, in stile cartoon, con ali a forma di X, preso in prestito dal film Guerre Stellari, in fuga da una Death Star (stella morta) in esplosione contrassegnata dal logo AT&T. Gli hacker di Berkeley amavano vedersi come i ribelli contro i crudeli imperi aziendali. La versione Unix di AT&T non riuscì mai a competere sul mercato con il concorrente BSD/Sun, sebbene si aggiudicò la guerra degli standard. Nel 1990, le versioni AT&T e BSD divennero difficili da distinguere avendo l'una adottato molte innovazioni dell'altra e viceversa.

Agli inizi degli anni '90, la tecnologia delle workstation del decennio precedente cominciava a vedersi seriamente minacciata da nuovi personal computer, a basso costo e dalle alte prestazioni, basati sul chip Intel 386 e i suoi discendenti.

Per la prima volta, ogni singolo hacker poteva finalmente permettersi di disporre anche a casa di macchine paragonabili, per potenza e capacità di memoria, ai minicomputer di un decennio prima, macchine Unix in grado di supportare un ambiente di sviluppo completo e di comunicare con Internet.

In questo nuovo scenario, il mondo MS-DOS rimase beatamente allo scuro degli sviluppi in corso. Nonostante le fila degli appassionati di microcomputer della prima ora si ingrandirono rapidamente fino a diventare una popolazione di hacker DOS e Mac di dimensioni ancora maggiori rispetto alla cultura "network nation", essi non riuscirono mai a sviluppare una cultura consapevole. Il ritmo dei cambiamenti era talmente veloce che ben cinquanta diverse culture tecniche nacquero e cessarono di esistere con la rapidità di una farfalla, senza mai raggiungere la stabilità necessaria allo sviluppo di un gergo, di un folklore e di una storia propri. L'assenza di una rete realmente pervasiva, paragonabile a UUCP o a Internet, non permise loro di diventare una network nation. Il crescere degli accessi a servizi commerciali online, come CompuServe e Genie, ma parallelamente la non diffusione in bundle di strumenti di sviluppo per sistemi operativi non-Unix, significava poco materiale su cui lavorare. Questa situazione impedì lo svilupparsi di una tradizione di collaborazione tra gli hacker.

La corrente hacker più importante, (dis)organizzata intorno a Internet, e finora largamente identificata con la cultura tecnica di Unix, non era interessata ai servizi commerciali. I suoi adepti volevano solo strumenti migliori e più Internet, cose che l'economico PC a 32-bit promise di mettere alla portata di tutti.

Ma dov'era il software? Le macchine Unix commerciali restavano comunque costose. Nei primi anni '90, numerose società fecero una prova vendendo porting di Unix BSD o AT&T per macchine PC. Il successo si rivelò elusivo, i prezzi non erano scesi di molto e (ancora peggio) non si ottenevano sorgenti modificabili e ridistribuibili per il proprio sistema operativo. Il tradizionale modello di software-business non stava affatto fornendo agli hacker ciò che volevano.

Neanche con la Free Software Foundation la situazione migliorò. Lo sviluppo di HURD, il tanto sospirato kernel Unix gratuito per hacker promesso da RMS, rimase fermo per anni e non riuscì a produrre alcunché di utilizzabile fino al 1996 (sebbene dal 1990 la FSF avesse fornito quasi tutti gli altri complicati componenti di un sistema operativo simile a Unix).

Ciò che dava davvero motivo di preoccupazione era che, con l'inizio degli anni '90, si cominciava a vedere con chiarezza come dieci anni di tentativi di commercializzare Unix stessero dopotutto fallendo. La promessa di Unix, di rendere portabili le cross-platform si perse tra mezza dozzina di versioni proprietarie di Unix. I detentori di Unix proprietario diedero prova di tanta lentezza e inettitudine nel campo del marketing, che Microsoft fu in grado di inglobare la maggior parte della loro fetta di mercato con la tecnologia del sistema operativo Windows, incredibilmente inferiore a quella Unix.

Nei primi mesi del 1993, qualsiasi osservatore pessimista avrebbe avuto tutti i motivi per decretare l'imminente fine della storia di Unix e della fortuna della sua tribù di hacker, cosa tra l'altro predetta sin dai tardi anni '70 a intervalli regolari di 6 mesi.

In quei giorni, era pensiero comune la fine dell'era del tecno-eroismo individuale e che l'industria del software e la nascente Internet sarebbero state dominate da colossi come Microsoft. La prima generazione di hacker Unix sembrava invecchiata e stanca (il gruppo di Ricerca della Scienza Informatica di Berkeley chiuse i battenti nel 1994). Il periodo non era tra i più felici.

Fortunatamente, ci furono cose che sfuggirono all'attenzione della stampa specializzata e perfino alla maggior parte degli hacker, cose che avrebbero prodotto sviluppi positivi verso la fine del 1993 e l'inizio del 1994.

In futuro, questa situazione avrebbe portato la cultura a imboccare una strada completamente nuova, disseminata di insperati successi.

I primi free Unix

Dal gap provocato dal fallimento dell'HURD, era emerso uno studente dell'Università di Helsinki di nome Linus Torvalds. Nel 1991, cominciò a sviluppare un kernel free Unix per macchine 386 usando un kit di strumenti della Free Software Foundation. Il suo rapido successo nella fase iniziale, attrasse molti hacker di Internet, volenterosi di aiutarlo nello sviluppo del suo Linux, una versione Unix con sorgenti interamente free e redistribuibili.

Anche Linux aveva i suoi concorrenti. Nel 1991, contemporaneamente ai primi esperimenti di Linus Torvald, William e Lynne Jolitz stavano sperimentando il porting di Unix BSD sul 386. La maggior parte di coloro che paragonavano la tecnologia BSD agli sforzi iniziali di Linus, si aspettavano che i porting di BSD diventassero i più importanti free Unix su PC.

La caratteristica fondamentale di Linux, tuttavia, non era tanto tecnica quanto sociologica. Fino allo sviluppo di Linux, era pensiero comune che qualsiasi software complicato come un sistema operativo, dovesse essere sviluppato in modo attentamente coordinato da un ristretto gruppo di persone ben collegate tra di

loro. Questo modo di operare era, ed è tuttora, tipico sia del software commerciale che delle grosse cattedrali di freeware costruiti dalla Free Software Foundation negli anni '80; così come dei progetti freeBSD/netBSD/OpenBSD, che allargarono il campo di applicazione del porting originale 386BSD dei Jolitz. Linux si evolse in modo completamente differente. Fin quasi dalla sua nascita, fu casualmente "preda di hacking" da parte di un vasto numero di volontari collegati solo tramite Internet. La qualità fu mantenuta non da rigidi standard o autocrazia, ma dalla strategia semplice e naive di proporre settimanalmente delle idee e di ricevere opinioni in merito da centinaia di utenti ogni giorno, creando una sorta di rapida selezione darwiniana sulle modifiche introdotte dagli sviluppatori. Con stupore da parte di quasi tutti, il progetto funzionava piuttosto bene.

Verso la fine del 1993, Linux fu in grado di competere per stabilità e affidabilità, con molti Unix commerciali, ospitando una grande quantità di software. Esso stava perfino cominciando ad attirare il porting di applicazioni software commerciali. Un effetto indiretto di questo sviluppo, fu lo spazzare via la maggior parte dei piccoli fornitori di Unix commerciali - la loro caduta fu anche determinata dalla mancanza di hacker e potenziali utenti ai quali vendere. Uno dei pochi sopravvissuti, BSDI (Berkeley System Design, Incorporated), fiori offrendo sorgenti completi, con il suo Unix base BSD, e coltivando stretti legami con la comunità hacker.

All'epoca tali sviluppi non furono pienamente rilevati dalle comunità hacker e non lo furono affatto al di fuori di essa. La tradizione hacker, a dispetto delle ripetute predizioni su una sua imminente fine, stava proprio iniziando a riorganizzare il mondo del software commerciale a propria immagine. Trascorsero ancora cinque anni prima che questa tendenza iniziasse a palesarsi.

La grande esplosione del Web

L'iniziale crescita di Linux coincise con un altro fenomeno: la scoperta di Internet da parte del grande pubblico. I primi anni '90 videro l'inizio di una fiorente industria dell'Internet provider, che forniva connessioni al pubblico per pochi dollari al mese. Dopo l'invenzione del World Wide Web, la già rapida crescita di Internet accelerò a rotta di collo.

Nel 1994, anno in cui il gruppo di sviluppo Unix di Berkeley chiuse ufficialmente i battenti, molte diverse versioni di free Unix (Linux e i discendenti del 386BSD) catalizzarono l'interesse degli hacker. Linux era distribuito su CD-ROM, e andava via come il pane. Alla fine del 1995, le maggiori aziende informatiche cominciarono a promuovere i propri hardware e software giocando la carta della loro grande compatibilità con Internet!

Nella seconda metà degli anni '90, l'attività degli hacker si incentrò sullo sviluppo di Linux e sulla diffusione di massa di Internet. Il World Wide Web era riuscito a trasformare Internet in un mezzo di comunicazione di massa, e molti hacker degli anni '80 e '90, intrapresero l'attività di Internet Service Provider fornendo accesso a questo nuovo mondo.

La diffusione di massa di Internet, aveva perfino portato la cultura hacker ad essere rispettata in quanto tale. Nel 1994 e 1995, l'attivismo hacker fece naufragare la proposta Clipper che avrebbe posto sotto il controllo del governo un metodo di codifica. Nel 1996, gli hacker si mobilitarono per sconfiggere il "Communications Decency Act" (CDA), e scongiurare il pericolo di censura su Internet.

Con la vittoria sul CDA, si arriva ai giorni nostri, periodo in cui chi scrive diviene attore e non solo osservatore. Questa narrazione continua con "La vendetta degli Hacker".

Tutti i governi sono, chi più chi meno, coalizioni contro il popolo...e siccome i governanti non hanno maggiore virtù dei governati...il potere del governo può essere mantenuto nei propri confini costituiti dalla presenza di un potere analogo, il sentimento congiunto del popolo.

*-- Benjamin Franklin Bache,
in un editoriale del Philadelphia Aurora, 1794*

(Tratto da A.A.V.V., *Open Sources. Voci dalla rivoluzione Open Source, Apogeo OpenPress, 1999*)

Questo saggio è libero; è possibile distribuirlo e/o modificarlo secondo i termini della licenza [GNU General Public License](#) come pubblicata dalla Free Software Foundation; si applica la versione 2 o (a propria discrezione) qualsiasi versione successiva della Licenza.

CALENDARIO SEMINARI HACKMEETING

VENERDI' 16 GIUGNO 2000

- PUB

14.00 - 16.00: Telematica di Base -
Parte I (Maddler)

16.00 -18.00: Dibattito Cybersyn II (a
cura di Avana).

18.00 - 20.00: Seminario sul Packet
Radio (Andrea Borgnino)

- CINEMA:

14.00 - 21.00: Sezione ARTE – L'arte
come pratica reale: riflessioni sull'uso
della tecnologia nella sperimentazione
artistica dalle Avanguardie all'arte in
Rete - Parte Prima (a cura di T.Tozzi,
T_Bazz e Snafu)

21.00 – 24.00: Proiezione di:
Pi Greco (USA 1999)

Syrena di Mariano Equizzi
AgentZ di "Giubilaeum "

- CATTEDRALE:

14.00 -16.00: Seminario sull'obiezione
di coscienza a software chiusi
(freaknet media lab + Osservatorio
diritti) + Intervento sulle problematiche
giuridiche del copyright (Barbara)

16.00 - 18.00: Seminario sull'ATM
(Fausto Saporito)

dopo cena - notte: corso sull'hacking
delle reti x.25 (a cura di Ivan Verri).

Notte: Seminario a cura di
radiocybernet trasmesso via radio-ip:
radio su ip (jaromil)

- SALA DA TE'

14.00 - 16.00: Corso su Flash (a cura
di Magnum e C4)

16.00 - 18.00: Seminario sulla
Crittografia (Laudano e Cerin0)

18.00 - 20.00: Seminario sullo Zen e
l'arte della programmazione (Stefano
Penge)

- STANZA VICINO AL CINEMA

Candida: la TV elettrodomestica

SABATO 17 GIUGNO 2000

- PUB

11.00 - 13.00: workshop
sull'accessibilita' dell'informazione in
Rete (di Ferry Byte, Arcelele, Claudio
Parrini e Shine).

14.00 -16.00: Seminario sull'snmp di
Tho, Bluca e P@sky

16.00 - 18.00: Seminario
sull'accessibilita' dell'informazione in
Rete (di Ferry Byte, Arcelele, Claudio
Parrini e Shine).

18.00 - 20.00: Seminario sulle Licenze
Free (a cura di Franco Vite: relatori
Eugenia Franzoni e Cristopher R.
Gabriel).

dalle 20.30: Dibattito sulla
"Globalizzazione, Telematica e
Movimenti Antagonisti"

- CINEMA:

14.00 -20.00: Sezione ARTE – L'arte
come pratica reale: riflessioni sull'uso
della tecnologia nella sperimentazione
artistica dalle Avanguardie all'arte in
Rete - Parte Seconda (a cura di
T.Tozzi, T_Bazz e Snafu)

- CATTEDRALE:

10.00 - 13.00: Seminario sul TCP/IP
del LOA Hack Lab di Milano.

14.00 - 16.00: Telematica di Base -
Parte II (Maddler)

16.00 - 18.00: Seminario su reverse
engineering (a cura di Letale Malattia,
Little John, Tinman e chiunque abbia
voglia)

18.00 - 20.00: BlindFold di Shine
Notte: seminario a cura di

radiocybernet trasmesso via radio-ip:
come costruire un hacklan (asbesto)

- SALA DA TE'

14.00 - 16.00: Seminario sulla
persuasione di Enkidu

16.00 - 18.00: Seminario sulle scelte
etiche e politiche nell'hacking (The
Warlus)

18.00-20.00: Seminario sul
Tecnoc controllo (a cura di Tactical
Media Crew) e sulla Censura in Rete
(a cura dell'Osservatorio) + intervento
delegazione di nodo 50

- STANZA VICINO AL CINEMA

Candida: la TV elettrodomestica

- PUB ALL'APERTO:

Dalle 15.00: Presentazione di Carlo
Formenti del libro "Incantati dalla
Rete"

DOMENICA 18 GIUGNO 2000

ore 11.00: Workshop/dibattito
generale conclusivo

DESCRIZIONE SEMINARI:

VENERDI' 16 GIUGNO 2000

.PUB:**14.00 – 16.00: TELEMATICA DI BASE, PARTE 1 (a cura di William Maddler)**

DESCRIZIONE:

Scopo del corso e' fornire gli strumenti di base per l'uso delle risorse di rete, cosa sono e a cosa servono:

- Email e Mailing Lists
- Usenet e Newsgroups, la grande bacheca
- IRC e chat, chiacchierare in rete
- World Wide Web, a spasso per la ragnatela, come e dove cercare
- PGP e i suoi fratelli, nozioni di base per l'uso dei software di crittografia

16.00 - 18.00: DIBATTITO CYBERSYN II (a cura di AvANa)

DESCRIZIONE:

Av.A.Na.net al Forte Prenestino lavora da quasi due anni alla costruzione di una infrastruttura di rete chiamata Cybersyn II <<http://www.forteprenestino.net/cybersyn2>> che permettera' l'uso di una piattaforma tecnologica realizzata autonomamente.

Questa sperimentazione si prefigge due scopi particolari:

a) fornire strumenti di cooperazione utilizzabili sulla Rete dalle emergenti figure dei brainworkers (i lavoratori immateriali) che stanno rinnovando il panorama lavorativo attuale;

b) fornire strumenti per incrementare il grado di coordinamento di un Centro Sociale come il Forte Prenestino senza intaccare l'autonomia interna che lo caratterizza.

IL PROGETTO CYBERSYN II

Il Progetto Cybersyn II (il nome richiama l'originale Progetto Cybersyn realizzato nel Cile di Salvador Allende per fornire nuovi strumenti di cooperazione sociale in alternativa al libero mercato e alla pianificazione) si basa su quattro progetti integrati: FORTHnet, BWN - Brain Workers Network, OSSERVATORIO sui diritti della comunicazione.

Saranno presentati in occasione dell'Hackmeeting FORTHnet (la freenet del Forte Prenestino), BWN (la piattaforma freeware di cooperazione che è in fase di realizzazione con l'obiettivo di creare un framework aperto per i lavoratori. La piattaforma viene sviluppata con modalità imprenditoriali), l'OSSERVATORIO (il servizio di informazione e consulenza sulle norme e sui diritti della comunicazione telematica).

18.00 – 20.00: SEMINARIO SUL PACKET RADIO (a cura di Andrea Borgnino)

DESCRIZIONE:

- Packet Radio: Discussione sull'utilizzo dell'etere per la trasmissione dati

Vedere al:

<http://www.alpcom.it/hamradio/packet/>

. CINEMA:

14.00 – 24.00: SEZIONE ARTE (a cura di Tommaso Tozzi, T_Bazz e Snafu)

**L'ARTE COME PRATICA REALE:
RIFLESSIONI SULL'USO DELLA
TECNOLOGIA NELLA SPERIMENTAZIONE
ARTISTICA DALLE AVANGUARDIE
ALL'ARTE IN RETE ;
Parte I: Avanguardie, Neoavanguardie
e movimenti degli anni '70-80**

DESCRIZIONE:

Oggi appare decisamente necessario costruire un discorso critico sulla direzione che sta prendendo la sperimentazione artistica, in un panorama culturale in cui il termine "arte" non presenta più confini precisi e si riscontra una notevole confusione sulla terminologia e sui parametri interpretativi da adottare.

C'è in realtà un filo rosso che collega le sperimentazioni artistiche delle Avanguardie del Novecento, quelle delle Neoavanguardie degli anni Sessanta-Settanta, le pratiche punk, dei writers e degli hackers con la creazione di contesti interattivi ad opera di chi lavora creativamente con gli strumenti digitali. Queste pratiche hanno contribuito a collettivizzare la concezione dell'arte, dimostrando che non esiste un artista da elevare a genio ed un pubblico relegato nelle paludi della passività, e che è possibile autoprodurre la propria arte a partire dall'autogestione dei mezzi comunicativi. Appare evidente come l'arte e la tecnica si siano sempre più avvicinate e come, proprio da un utilizzo consapevole e autogestito dei mezzi di comunicazione e degli strumenti del linguaggio, sia possibile concepire un'arte "su cui mettere le mani", su cui "hackerare", dando vita ad un evento-flusso in evoluzione collettiva. Lo stesso concetto di "Arte" appare quindi obsoleto ed è andato ampliandosi a numerose pratiche che si fondono con la vita quotidiana, processo favorito in gran parte dalla sperimentazione creativa con i media e con la tecnologia.

La storia dell'arte del Novecento, ha progressivamente determinato la

dissoluzione del concetto di Arte come merce fissa da collezionare o da esporre in un museo soggetta alle regole di un mercato artistico chiuso e condizionata da un Sistema dell'Arte onnicomprensivo. Lo *spettatore* di un'opera d'arte ne è diventato il *co-autore*, andando a fare parte di un evento artistico impermanente; la figura dell'artista come unico creatore si è frammentata nell'azione dei fruitori invitati a vivere collettivamente l'happening artistico determinando la nascita di eventi processuali potenzialmente senza fine. Tale aspetto è ancora più evidente oggi in cui, attraverso l'interazione con l'interfaccia grafica di un computer e con installazioni di realtà artificiale e virtuale, l'individuo può costruire e decostruire i propri percorsi comunicativi, rendendo la sperimentazione artistica un'esperienza reale.

Un'arte che sta al passo con i tempi, è quindi quella che si rapporta ai cambiamenti (culturali, tecnologici, economici) in corso nella società e che permette al fruitore di essere il protagonista dell'evento artistico, dandogli la possibilità di autodeterminarlo.

L'artista in questo senso diviene il creatore di *contesti di scambio*, colui che dispone la piattaforma in cui il fruitore può agire attraverso il suo gesto creativo, mentre l'opera diviene un flusso di continue manipolazioni, contaminazioni, riproduzioni, alterazioni dei codici del linguaggio e delle icone. L'arte si va a fondere con il gesto degli individui, con i loro corpi, con gli universi virtuali autodeterminati, con le derive fluttuanti dei link, con l'azione della nostra mente che si fa protesi digitale, con le pratiche autogestite...originando continui episodi aperti da riempire con reti di relazioni individuali orizzontali.

L'uso della tecnologia nella pratica artistica ha quindi ulteriormente contribuito ad aprire il concetto di arte e a frammentarlo nei numerosi aspetti della vita quotidiana, ma ha anche determinato la crisi di molte categorie critiche. Oggi più che mai seguendo la scia dell'esplosione

Internet vengono pubblicati testi, pagine web, realizzati eventi culturali sull'arte che fa uso delle nuove tecnologie, ma in molti casi appare evidente la mancanza di strumenti interpretativi adeguati. Questo non sempre per scarsa preparazione, ma per la difficoltà nello stare al passo con un'accelerazione tecnologica, creativa e sperimentativa come quella attuale e con un panorama artistico e culturale in continua mutazione.

Che differenza e che rapporti ci sono fra l'ascii art, la computer art, l'hacker art, la cyberart, la webart, la .net art? Che influenza hanno avuto i movimenti antagonisti degli anni Settanta ed Ottanta sulla creazione artistica che fa uso delle nuove tecnologie? E come si rapportano le odierne sperimentazioni con quelle attuate durante le Avanguardie e le Neoavanguardie?

In più: come si deve rispondere alla creazione di un mercato artistico in Rete con il dilagare della New Economy? E come vanno interpretati i tentativi di apporre il copyright alle opere in Internet? E' corretto definire le gallerie virtuali in Rete come opere di .net art?

Appare quindi necessario ritrovarsi per discutere queste ed altre delicate tematiche e provare collettivamente a dare un senso ai cambiamenti in atto.

Il seminario sarà diviso in due sezioni, che avranno luogo nel pomeriggio del 16 Giugno e in quello del 17 Giugno. La prima sezione riguarderà interventi relativi al periodo della Avanguardie e delle Neoavanguardie e ai movimenti degli anni '70-80. La seconda giornata prenderà in considerazione le pratiche artistiche attuali e le ipotesi sul futuro.

ore 14-15

PRESENTAZIONE:

Tatiana Bazzichelli (T_Bazz):

Presentazione

Tommaso Tozzi: Rapporti arte-media e comunicazione dall'inizio del secolo agli anni settanta + concetto di hacker art presentazione

ore 15-16.30

APPROFONDIMENTI STORICI (PARTE PRIMA)

- DALLE AVANGUARDIE AGLI ANNI '80

Francesco Galluzzi: Falsificazione, strategia e sabotaggio nelle pratiche artistiche del '900

Lorenzo Taiuti: Arte e media nel XX secolo

Simonetta Fadda: Il video sociale degli artisti negli anni '70

Gomma: "Hacking California 1966-1979"- --circa 15 minuti con la proiezione di un video di Lee Felsestein

Ore 16.30-19.00

APPROFONDIMENTI E DISCUSSIONE (PARTE PRIMA)

- RIPRODUZIONE E PRODUZIONE A COSTO ZERO

Roberto Terrosi: Rapporto tra estetica e tecnologia nell'arte dalle neoavanguardie alla new economy

Giacomo Verde: Ridefinire le arti a partire dal rapporto autore-mercato-fruitore

Jaromil: Presentazione dell'ascii webcam

Mariano Equizzi: Cinema ed effetti speciali a costo zero

Mongrel (video intervista): Le interfacce soggettivate, Linker e National Heritage

- IL CORPO ILLUMINATO

Federico Bucalossi: Codice neurale evolutivo – dalla pittura, alla scienza alle immagini digitali

Luigi Ciorciolini: Problemi posti dal concetto dello spazio nella Rete

Francesca da Rimini: Dollspace -

L'erotica della digicarne: 30 minuti + Gettando i semi del nuovo disordine

mondiale: Presentazione di BAD CODE (videogioco a cura di VNS MATRIX),

Love Hotel di Linda Wallace (AUS 2000), cortometraggio ispirato a DollSpace

- NEI FRATTALI SONORI DELLA RETE

Alessandro Ludovico: Suoni in rete, spunti e paradossi di musicisti ed artisti del suono Norie Neumark e Maria

Miranda (Australia): Presentazione e ascolto di "Shock in the Ear"

Ore 19.00-21: DIBATTITO

Ore 21.00-24.00:

Proiezione di PI GRECO (USA 1999)

Syrena

AgentZ

Giubilaum di Mariano Equizzi

e politiche di formazione basate su piattaforme free-software quali appunto linux per motivi di opportunita' economica ma soprattutto per permettere alle nuove generazioni di non perdere l'opportunita' di conoscere un ambiente operativo che e' universalmente riconosciuto per la sua filosofia di funzionamento come il linguaggio informatico piu' affidabile, versatile e sicuro

. CATTEDRALE:

14.00 – 16.00: SEMINARIO SULL'OBIEZIONE DI COSCIENZA A SOFTWARE CHIUSI (a cura del freaknet medialab - Catania) + PROPOSTA DI INTRODUZIONE AL FREE-SOFTWARE (P.E. LINUX) NELLA PUBBLICA ISTRUZIONE a cura dell'Osservatorio Diritti <http://www.romacivica.net/oss.diritti>) + INTERVENTO SUL COPYRIGHT, ANALISI DELLE PROBLEMATICHE GIURIDICHE Barbara)

DESCRIZIONE:

A scuola di linux

In calce alla proposta "Campagna per l'obiezione di coscienza del software" che riportiamo piu' avanti proponiamo come osservatorio un'iniziativa all'interno dell'hackmeeting con lo scopo di diffondere linux e la filosofia-risorsa del free-software nella pubblica istruzione italiana

il progetto Osservatorio Diritti ~ <http://www.romacivica.net/oss.diritti/> organizza una giornata di discussione sull'opportunita' di introdurre risorse open-source e free-software quali linux nella pubblica istruzione

particolare attenzione viene posta nella necessita' di introdurre strutture

i vari accordi esistenti fra i vari ministeri (non ultimo quello della pubblica istruzione) e case software proprietarie (windows, ecc.) prevedono una distribuzione a cascata sulla pubblica amministrazione che da un lato prefigura un rapporto di dipendenza estrema in termini di know-how dalle case produttrici e distributrici di software e dall'altro spesso e volentieri sono sovradimensionate rispetto al reale utilizzo di questi strumenti

quante sono ad esempio le licenze di sviluppo cosi' generosamente concordate con le software house che si trovano all'interno delle scuole (che pure in virtu' dell'autonomia dovrebbero avere un occhio di riguardo ad esigenze di risparmi economici) quando spesso e volentieri il corpo insegnante non e' in grado di gestire nemmeno le piu' semplici licenze end user?

perche' non e' mai stato preso seriamente in considerazione di adottare software open source e free software quale e' linux all'interno delle scuole pubbliche quando anche solo in termini di sicurezza (eppure e' una argomento di moda di questi tempi...) rappresenta se ben gestito una risorsa economicamente valida e tecnicamente efficace?

essite un mondo intero di associazionismo rappresentato dai cosiddetti l.u.g. (linux user groups) ~

<http://linux.it> ed hacklabs che invitiamo da ora ufficialmente a partecipare a questa iniziativa che potrebbe essere coinvolto in un'opera di colonizzazione della pubblica istruzione con risorse e competenze basate su risorse open-source e free-software

ecco perché la necessità di introdurre di politiche di formazione su ambienti operativi quali linux nelle scuole al fine di realmente mettere a disposizione della cittadinanza tutta le competenze necessarie per affrontare la cosiddetta società dell'informazione e non solo a parole (e mai nei fatti) come spesso viene strumentalmente sbandierato ai quattro venti dai politicanti di turno

nella nostra proposta è anche insita la volontà di dare credito a una economia nascente di tipo orizzontale costituita da tutta quelle miriadi di persone esperte in rete che potrebbero essere attivate per installare, assistere e formare hardware, software e human-ware nella pubblica istruzione (corpo docente e studenti insieme) e parallelamente ricondurre alla calma un mercato verticistico costituito dal monopolio delle software house che oltre a imporre un tipo di sapere poco trasmissibile e riproducibile impongono in più un dazio non più giustificabile e rappresentato dal costo delle licenze software che nel caso di introdurre una politica di introduzione di risorse free software nella pubblica istruzione semplicemente non potrebbe esistere

ci rendiamo conto di non tirare fuori niente di nuovo, già significative giornate di studio sono state svolte in passato ~
<http://www.provincia.tn.it/istruzione/scuole/seminario.htm> , sul free software ~
<http://firenze.linux.it/philosophy/categorie.shtml> e sull'open-source ~
www.opensource.it il dibattito è molto

sviluppato in rete ed esistono già proposte simili alla nostra come quella di Di Meo che si spera possa partecipare a questa giornata, il mondo dell'associazionismo ma anche della piccola impresa si è interrogato spesso su come cercare di ottimizzare le risorse software nella pubblica istruzione ~
<http://www.aniene.net/virtual>

ci rivolgiamo anche a quei settori dell'autorganizzazione come gilda e cobas che potrebbero forse fare da cerniera fra noi e le istituzioni referenti

vogliamo cercare di dare fiato e gambe a tutte quelle proposte che si muovono nel senso di veicolare risorse e conoscenze free-software nella p.a. e in particolare nella pubblica istruzione non solo per risparmiare un po' di soldi comunque "nostri" ma soprattutto per dare credito a una nuova forma di economia diffusa e rivendicare il diritto a confrontarsi con un tipo di sapere sul quale è possibile metterci le mani sopra, hacking docet... ;-)

osservatorio diritti

Campagna per l'obiezione di coscienza del software

Un aneddoto istruttivo

Poche settimane fa alla facoltà di Informatica dell'Università di Catania è successo che uno studente di secondo anno va dal professore a concordare il lavoro da portare all'esame. Si tratta di un database e il docente chiede che l'esercitazione venga svolta con Access, programma incluso nel pacchetto Office della Microsoft, in grado di funzionare solo sui sistemi operativi Windows della stessa Microsoft. Allora lo studente fa presente al professore che non ha Windows sul computer e, soprattutto, che non ha alcuna intenzione di installarlo. Dice anche che sul suo computer c'è

Linux (il sistema operativo "free" e a "sorgente aperto"), che su questo funziona MySql (un gestore di database, anch'esso "free" e a "sorgente aperto") e che è dispostissimo a fare il tutto con questi strumenti.

Il professore tergiversa, non è convinto e insiste: "È roba da terzo o forse quarto anno... Meglio che lo fai con Access". Ma lo studente è fermo sulla propria posizione di sviluppare il compito solo con il Free Software.

Alla fine a cedere è il docente.

L'Università e il software

Spesso nei corsi di laurea di Informatica, Scienze dell'informazione, Ingegneria elettronica e Ingegneria informatica si costringono gli studenti a usare software commerciali e chiusi. Ciò ha almeno quattro effetti:

1. Ogni studente deve acquistare le licenze d'uso del software prescelto dal professore. Ciò significa, nel caso preso in esempio (che è anche un caso molto comune), 100 euro per il sistema operativo più economico della serie Windows (il 98) e circa 650 per il pacchetto Office nella versione che include anche il gestore di database (cioè Access). Se lo studente intende fare a meno delle licenze d'uso viene automaticamente ricacciato nell'area grigia della pirateria del software e minacciato di arresto dalle campagne dai toni terroristici della BSA (Business Software Alliance, il braccio lobbistico della Microsoft). In paesi come l'Italia, dove la BSA è riuscita a imporre una legislazione estremamente repressiva, i guai con la giustizia penale sono un rischio effettivo.

2. Gli studenti che usano sistemi operativi la cui costruzione è coperta dal segreto industriale non hanno modo di sapere esattamente come il sistema operativo stesso funzioni. La Microsoft, per esempio, non ha mai fatto vedere il codice con cui ha realizzato Windows né il codice scritto per gli

applicativi. In più la licenza d'uso vieta espressamente il "reverse engineering", cioè la tecnica con la quale è possibile analizzare un programma a partire dalla sua versione già compilata (l'eseguibile che viene distribuito). Paradossalmente uno studente che abbia avuto esperienza solo su questo tipo di ambiente operativo può laurearsi in informatica senza conoscere quali compiti debba assolvere un computer per funzionare correttamente. L'università, incoraggiando l'uso di prodotti con queste caratteristiche, abdica dal suo ruolo formativo e di ricerca.

3. Denaro pubblico e tasse degli studenti sono sperperati nell'acquisto delle licenze d'uso di software commerciale. Le risorse (spesso già scarse negli atenei dell'Europa meridionale) vengono così impiegate per fare arricchire sempre di più i soliti noti.

4. Quando un laureato (non) formato sull'utilizzo di software Microsoft (o di qualche altro solito noto) andrà a lavorare, proporrà l'impiego dello stesso tipo di sistema operativo e di applicativi. Cioè l'acquisto di nuove licenze dalle maggiori software house, contribuendo così alla sempre maggiore potenza economica e influenza politica di queste.

L'importanza di esser aperti

Non tutte le grandi case di software operano con la stessa politica. Per esempio la Sun mette a disposizione per uso personale o didattico i sorgenti del sistema operativo Solaris (formalmente gratis, ma chiedendo di fatto 75 dollari). Quindi uno studente che opera su questo sistema, seppure soggetto ad alcune limitazioni imposte dalla casa madre, ha la possibilità di capire effettivamente "come stanno le cose" (ed eventualmente di modificarle per il proprio studio). Aziende come la Microsoft tengono invece il più stretto riserbo sul codice sorgente. È innegabile che i

prodotti Microsoft abbiano un'ampia diffusione anche nel mondo della produzione e dei servizi (industrie, aziende varie, enti pubblici, ecc.) per cui è giusto che gli studenti e i laureati li conoscano. È sbagliato però considerarli come gli unici strumenti da utilizzare.

Non si può non tenere conto che dalla metà degli anni novanta, grazie alla facilità dell'accesso a Internet, la comunità dei programmatori ha rivitalizzato l'idea del Free Software e ha sviluppato migliaia di progetti secondo la licenza GNU/GPL. Per capire l'importanza che oggi ha assunto il software così concepito basta pensare all'enorme diffusione di Linux, progetto GNU/GPL per eccellenza. Solitamente questi progetti sono portati avanti da gruppi di programmatori che possono anche risiedere in aree geografiche lontanissime tra loro. Chi partecipa si scambia i sorgenti (cioè il codice con cui si realizza il programma) con il resto del team attraverso la rete. Alla fine del lavoro questi sorgenti non rimangono chiusi, ma vengono distribuiti assieme al programma stesso. Talvolta sono solo i sorgenti a essere distribuiti, permettendo così all'utilizzatore di controllare per bene cosa sta per essere compilato e fatto funzionare sul proprio computer.

In questo modo si è venuta a creare una vasta rete di solidarietà e di conoscenza. Ogni giovane programmatore ha la possibilità di poter esaminare le tecniche utilizzate dai più esperti. Questo da un lato costituisce l'opportunità di una crescita professionale per migliaia di giovani e dall'altro ricrea lo spirito originario della comunità dei programmatori,

un po' come una enorme Berkeley degli anni settanta, un luogo dove ogni tipo di conoscenza scientifica veniva condivisa e utilizzata da tutti.

Al di là dei discorsi dal sapore idealistico, oggi giganti del calibro della Ibm, della HP, della Compaq, solo per citarne

alcuni, stanno investendo ingenti risorse economiche nello sviluppo di software di questa concezione.

Il ritardo europeo

In effetti la nuova politica delle grandi case di software (Microsoft esclusa) sembra profondamente diversa da quella cominciata alla fine degli anni settanta. A quel tempo le software house iniziarono negli Stati Uniti a porre sotto brevetto le tecniche di programmazione, talvolta anche tecniche non "proprietarie" ma ampiamente conosciute e utilizzate dai comuni programmatori. Il brevetto americano tutela l'"invenzione" per diciassette anni: un periodo spropositato di fronte alla rapida evoluzione dell'informatica.

L'effetto dei brevetti nella programmazione è stato deleterio perché ha sia impedito il libero sviluppo della programmazione indipendente, sia messo in difficoltà i "piccoli" programmatori che spesso si sono visti citare in tribunale dai titolari dei brevetti sulle tecniche che avevano utilizzato (spesso perché semplicemente "riscoperte").

Proprio mentre la pratica di brevettare ogni semplice algoritmo cominciava a entrare in crisi negli USA, l'Unione ha approvato una legge che istituisce il brevetto europeo sul software. Di fatto un'operazione considerabile ormai in controtendenza storica, seppure ancora dannosa, perché pone un limite grave allo sviluppo della tecnica e quindi alla qualità della vita.

Il diritto all'obiezione

Sosteniamo, quindi, che ogni studente dei corsi di laurea che hanno direttamente a che fare con la Computer Science, cioè Informatica, Scienze dell'informazione, Ingegneria elettronica e Ingegneria informatica, possa obiettare davanti all'obbligo di sprecare tempo e denaro (soldi che come abbiamo visto servono ad alimentare un

apparato repressivo e liberticida) su sistemi operativi e applicazioni che non contribuiscono di fatto all'arricchimento della conoscenza nel campo dell'informatica "seria". Chi decide di seguire un corso universitario ha il diritto di ricevere un'istruzione adeguata al livello scientifico e chi insegna ha il dovere di fornirla. Se questi non ne è in grado, non deve penalizzare lo studente portandolo dopo quattro o cinque anni di lezioni universitarie a un livello concettuale non dissimile da quello di una moderna segretaria d'azienda. Per queste ragioni lanciamo una campagna di disobbedienza civile con la forma dell'obiezione di coscienza all'uso del software "proprietario" a sorgente chiuso.

Catania, 18 marzo 2000
FreakNet MediaLab
<http://www.kyuzz.org/freaknet>

16.00 – 18.00: SEMINARIO SULL'ATM (a cura di Fausto Saporito)

DESCRIZIONE:

- Uno sguardo generale su questo protocollo, di cui si è detto tutto e il contrario di tutto

Dopo Cena – Notte: CORSO SULL'HACKING DELLE RETI X.25 (a cura di Ivan Verri)

DESCRIZIONE:

Cosa è una rete x.25
Perchè è bello fare hacking sulle reti x.25
Come si usa una rete x.25
Le reti Itapac e Sprintnet
Come interpretare il formato di un NUA
Come accedere a x.25 Scanning di una rete x.25
Come riconoscere i sistemi
Come hackerarli

Mi piacerebbe fare degli esempi pratici (simulati)

Durante tutta la giornata in Cattedrale: RADIO CYBERNET

Di notte: SEMINARIO A CURA DI RADIOCYBERNET TRASMESSO VIA RADIO-IP: RADIO SU IP (Jaromil)

- come fin dalla prima edizione l'hackit è in broadcast via web e anche via etere in alcune zone tramite qualche stazione che farà da gateway. la radio è **radiocybernet** (kyuzz.org/radiocybernet) lo staff: asbesto, voyager, shining, topo, kyuzz, jaromil. Quest'anno saranno seguiti da vicino da radio golem (radio rai) e qualche altra radio locale anche fuori italia

. SALA DA TE':

14.00 – 16.00: CORSO SU FLASH (a cura di Magnum e C4)

DESCRIZIONE:

FLASH (raw tutorial) - flashare il web

- INTRODUZIONE

(overview sul tool di creazione in tecnologia Shockwave [macromedia Flash], caratteristiche principali del formato SWF, e sulla tecnologia Shockwave Flash, campi di applicazione e ambiti di utilizzo)

- approccio con Macromedia FLAH 4.0 (approccio con l'applicativo e principali parti funzionali dell'interfaccia, "filosofie" di utilizzo, introduzione allo sviluppo di filmati Flash, tecniche organizzative dell'animazione (semplice e/o complessa))

- utilizzo pratico dell'interfaccia di FLASH 4.0 (parte 1)

- * accenni sulla grafica vettoriale (brevi brevi tanto per capirsi)
- * disegno delle forme (e degli oggetti), proprietà e convenzioni del disegno, tecniche grafiche
- * utilizzo dei LAYERS
- * utilizzo della TIMELINE (concetti principali della TIMELINE, KEYframing dell'animazione, MASKing, TWEENing [motion & shape]. easing.... etc)
- * utilizzo di suoni (tipi di suoni, possibilità di sequencing)

pausa (forse :-)))

- utilizzo pratico dell'interfaccia di FLASH 4.0 (parte 2)

- * utilizzo delle SCENES
- * LIBRARYes, tipi di oggetti in libreria, concetti di dipendenza fra gli elementi etc
- * tecniche avanzate di disegno e di masking
- * importazione di oggetti/files/formati esterni
- * utilizzo delle ACTIONs principali
- * effetti con i layer (ombre, pseudo 3d etc)
- * utilizzo di TOOLS esterni (3D studio + illustrate, SWIFT-3D, SWIFT)
- * esportazione dei filmati, embedding nell'HTML (dreamweaver - NO FP2000 !!!!)

- VARIOUS

- * esempi di progetti flash presi in giro per la rete e da cui imparare e molto .-))) (eye4u docet)
- * animazioni al limite dell'arte
- * scambio di OPINIONI/RIFLESSIONI, di TOOLz e roba simile.. portare CD-R vuoti.... :-))
- * tool NON-Windows per Flash... (qui serve un po' di aiuto!! :-)

16.00 – 18.00: SEMINARIO SULLA CRITTOGRAFIA (a cura di Cerin0 e Laudano)

DESCRIZIONE:

Seminario sulla crittografia

Venerdì 16 Giugno 2000 -
Hackmeeting'00 - Centro Sociale Forte Prenestino (Roma)
Sala da thé - ore 16:00 / 18:00

Programma (I parte - introduzione alla crittografia)

- la crittografia/crittologia (cos'è, a cosa serve, quando è nata)
- la crittografia asimmetrica (cos'è, come funziona, cenni matematici alle funzioni unidirezionali)
- algoritmi crittografici (l'Idea, l'Rsa, ElGamal, DSA, 3DES, Blowfish, Twofish, CAST5, MD5, SHA-1, RIPE-MD-160 and TIGER)
- i software crittografici freeware ed open-source, il PGP ed il GNUPG

Programma (II parte - la steganografia)

- la steganografia (cosa è, le sue origini e i suoi principi fondamentali)
- steganografia e watermarking
- le caratteristiche e i problemi più frequenti
- i tools più comuni e il loro impiego

Riferimenti bibliografici e links

Libri:

- "Segreti Spie e Codici Cifrati" C.Giustozzi, A.Monti, E.Zimuel - edizioni Apogeo
- "Codici & Segreti" Simon Singh - edizioni Rizzoli
- "Crittografia" Andrea Sgarro - edizioni Franco Muzzio
- "Crittologia - come progettare le informazioni riservate" L.Berardi,

- A.Beutelspacher - edizioni FrancoAngeli
- "L'enigma di Fermat" Amir D.Aczel - edizioni Est
- "Kryptonite" Joe Lametta - edizioni Nautilus
- "The Official Pgp User's Guide" Philip R. Zimmermann - edizioni Mit
- "PGP : Pretty Good Privacy" Simson Garfinkel - edizioni O'Reilly & Associates

(Th3 Italian Gathering), un demo party, a Pescara.
Sta preparando un nuovo libro introduttivo e di programmazione sul PGP ed il GNUPG.

Laudano

Uno come tanti.
Deve a Fravia e al suo sito se ha incominciato ad interessarsi alla steganografia.

Web Links:

- <http://www.pgpi.org>
- <http://www.gnupg.org>
- <http://www.codicicifrati.com>
- <http://www.cryptography.com/resources/papers/>
- <http://www.cs.berkeley.edu/~daw/people/crypto.html>
- <http://www.jyu.fi/~paasivir/crypt/>
- <http://www.demcom.com/english/steganos>
- <http://www.ijtc.com/Steganography/>
- <http://www.stego.com/>
- <http://www.cl.cam.ac.uk/~fapp2/papers/>
- <ftp://ftp.funet.fi/pub/crypt/steganography>

I relatori

Enrico Zimuel (aka cerin0)

Sviluppatore Open Source, collabora con il progetto [GNUPG](#) - The GNU Privacy Guard.

Coautore del libro "[Segreti, Spie e Codici Cifrati](#)" edito da [Apogeo](#) si occupa di crittografia e sviluppo software.

Collabora con diverse riviste italiane del settore informatico: Computer Programming, Dev, Pc Professionale.

Fa parte del direttivo della [Metro Olografix Association](#) (dove tiene corsi di programmazione e crittografia gratuiti) ed è socio di [Alcei](#) (Electronic Frontiers Italy).

Ha organizzato nel '96, '97 e '98 il Tig

18.00 – 20.00: SEMINARIO SULLO ZEN E L'ARTE DELLA PROGRAMMAZIONE (a cura di Stefano Penge)

DESCRIZIONE:

<http://www.onlynx.it/hi/strumenti/zen.html>

Zen e arte della programmazione

Contributo per gli atti del THUG di Stefano Penge

Introduzione

Questo scritto è dedicato ai programmatori, professionisti o autodidatti, soggetti completamente trascurati dalla saggistica non tecnica (per esempio quelle pedagogica), in quanto generalmente considerati troppo lontani da interessi teorici.

Si riconosceranno, spero, in molti luoghi del testo, visto che molte frasi sono state scritte all'alba di nottate passate a cercare l'uscita da foreste condizionali, attraversando vortici ricorsivi e arrampicandosi su pareti di cicli da i all'infinito. E forse acquisteranno essi stessi una coscienza più alta del loro stesso "mestiere".

L'idea di base è quella di affrontare il rapporto con la programmazione da una prospettiva apparentemente paradossale, cioè appunto quella del pensiero orientale di matrice genericamente taoista, che qui da noi ha avuto più fortuna sotto

l'etichetta dello Zen.

Zen che, come in tutti i casi letterari paralleli a questo almeno nel titolo (da Robert Pirsig a Eugene Herrigel, da Fritjof Capra fino a Jacopo Fo), c'entra solo fino ad un certo punto, come pretesto, come artificio di straniamento, come rimando continuo ad un'altra sponda. Ma è un rimando necessario, se si vuole evitare di rimanere invischiati nella visione dualista, tecnicistico-umanistica.

Ho cercato in giro sulla rete per vedere se questa operazione era già stata tentata, e ho trovato dozzine di "Zen e l'arte di ..." (dalla programmazione parallela al web mastering, dalla programmazione grafica al supporto tecnico); purtroppo nessuno di questi parla di Zen, né ha a che fare con lo Zen nemmeno lontanamente. Il significato del termine "zen" in quei contesti è semplicemente "vertice assoluto della perfezione".

Personalmente, sono rimasto sempre affascinato dalla letteratura zen (almeno da quella tradotta in italiano, da "La porta senza porta" a "101 storie zen") proprio per questa visione integrale dell'universo: il distacco dall'io senza il distacco dal mondo, una proposta di seguire la "via" che non richiede accecamenti mistici, ma passa per le cose di tutti i giorni. Anni dopo, leggendo Lo Zen e l'arte della manutenzione della motocicletta, quello che mi ha colpito è che questo atteggiamento può essere esteso anche a ciò che è artificiale, compresi i computer e il nostro contorto rapporto con essi.

In questa prospettiva, bisogna anzitutto fare piazza pulita dei pregiudizi, e riconoscere che costruire programmi (le cose che rendono i computer diversi dai frigoriferi) è un'attività complessa, ricca, artistica, che richiede altrettanta unificazione della persona del tiro con l'arco e della cerimonia del tè e che promette altrettanta pace interiore, per quanto paradossale possa sembrare.

Il Manuale del Programmatore Zen

La programmazione secondo loro

Programmare è un'attività che viene in generale sopravvalutata dai programmatori e sottovalutata da tutti gli altri. Vista dall'esterno, è un'attività ripetitiva, fredda, legata alla logica e alla matematica. Fino a poco tempo fa, nella scuola dell'obbligo l'informatica era considerata una parte della matematica, e se ne occupavano solo gli insegnanti di materie tecnico-scientifiche. Gli altri la guardavano con sospetto.

E anche fuori le cose non vanno meglio. Dal telaio meccanico al PC, l'immaginario collettivo associato con le macchine è diviso in una parte positiva (il miraggio del servo meccanico che svolge rapidamente tutti gli incarichi più gravosi) e una negativa (lo spettro sociale del licenziamento, ma anche del prosciugamento dei rapporti all'essenziale/utile).

Questo immaginario è comune tanto agli apocalittici che agli integrati. Persino quella categoria particolare di utenti che vive in un rapporto di dipendenza quotidiana dalla macchina (i programmatori di computer) spesso coltivano un'immagine di questo tipo. Per il programmatore, ufficialmente il computer è la parte materiale di un algoritmo di calcolo. Fili, transistor, circuiti stampati che corrispondono a funzioni matematiche. "Il computer fa solo quello che gli si dice di fare."

Ci si aspetterebbe quindi che i programmatori fossero tutti ingegneri in camice bianco.

Invece la figura classica del programmatore è ben descritta in "Microservi": single, trascurato nel vestire - nella migliore delle ipotesi, con pessime abitudini alimentari (dalle patatine country alle caramelle sintoblob), nessuna coscienza sociale o politica e in generale un'identità personale piuttosto slavata. Vive di notte ed è abituato a lavorare ininterrottamente anche per due o tre giri

della lancetta corta. Poi sprofonda in un dormiveglia di sogni a finestre che non riesce a chiudere. La prima cosa che fa al risveglio è premere, con gli occhi semichiusi, il pulsante On.

Non tutti i programmatori sono (sempre) così. Ma alzi la mano chi non si riconosce in almeno uno di questi tratti.

La programmazione secondo noi

Vista dall'interno, la programmazione è un'arte.

E infatti ne presenta le caratteristiche tipiche:

- Lo stile. Epoche diverse hanno avuto stili diversi; ma anche ogni programmatore ha uno stile riconoscibile. Questo fatto non è mai stato studiato approfonditamente. Ma chi lavora in gruppo, o quanto meno collabora con altri, è in grado di riconoscere la provenienza di un pezzo di codice, o addirittura di segmenti di codice diversi all'interno di un programma. Dai nomi delle variabili e delle funzioni alle preferenze sull'uso di strutture (do... until anziché while... end), fino all'impaginazione e ai commenti, ogni programmatore lascia la firma sulla sua opera.
- La flessibilità. Ogni problema ha un'infinità di soluzioni diverse. Il teorema secondo il quale "dato un problema qualsiasi e data la sua migliore soluzione è sempre possibile fare di meglio" non è mai stato dimostrato, ma tutti concordano sulla sostanza.
- Le stranezze dei suoi adepti: gli "artisti" della programmazione possono essere catalogati come hackers, come guru, come star o semplicemente come mariti snaturati.
- La difficoltà nell'apprendimento. Per quanti corsi si facciano, per quanti libri "C++ per le casalinghe" si possano leggere, ci sono solo due tipi di

programmatori: quelli che hanno appena iniziato e quelli che hanno una bella esperienza alle spalle. La cosa difficile da imparare - e ancora di più da insegnare - non è la programmazione, ma la buona programmazione.

Contrariamente a quello che si pensa, la buona programmazione richiede un grandissima dose di intuizione, molta fantasia, moltissima pazienza (e un minimo indispensabile di logica). Cosa si intende per buona? Non è una domanda facile.

Tradizionalmente, la qualità del codice viene valutata secondo tre parametri:

1. Efficienza: occupazione di memoria, velocità di esecuzione
2. Robustezza: capacità di gestire situazioni eccezionali, compresi gli errori
3. Manutenibilità: possibilità di mettere le mani sul codice in un secondo tempo o da parte di altri

Un buon programma è senz'altro un programma efficiente, robusto e manutenibile.

Ma questi sono solo aspetti pratici, effetti collaterali anche se vantaggiosi, della qualità vera.

Nel seguito proveremo ad adottare altri punti di vista. Il che non vorrà dire che questo punto di vista si rivelerà errato, ma solo che è necessario saper cambiare punto di vista, o saperne adottare più di uno contemporaneamente.

Ecologia del codice

Un punto di vista più avanzato è quello che valuta il codice da un punto di vista "ecologico", cioè lo inserisce in un sistema più vasto che può essere il computer dell'utente o la rete intera. I parametri significativi diventano allora economia delle risorse, velocità e riciclabilità.

Economia delle risorse

Ai tempi dei Vic 20 e dei Commodore 64,

la lunghezza dei nomi delle variabili era un parametro significativo dell'occupazione di memoria di un programma e della sua velocità d'esecuzione. Questo portava ad un codice molto compatto ma non commentato e pochissimo comprensibile. Quei tempi sono ricordati con nostalgia dai "vecchi", che storcono il naso di fronte al disprezzo delle risorse altrui dimostrato da chi occupa megabyte e su megabyte del vostro hard disk senza neanche chiedervelo.

Per loro è un principio ovvio, che non richiede dimostrazione: occorre usare strutture di dati non ridondanti. Un dato deve essere rappresentato dal più piccolo contenitore possibile. Non è solo questione di etichetta: più RAM viene occupata, più è probabile che il sistema lavori in situazioni limite e quindi vada in crash.

Oggi che RAM, HD e tempo processore sono in svendita, il bene che scarseggia - e quindi il valore - è la larghezza di banda. E la netiquette vuole che non si occupi la banda inutilmente. Dalle raccomandazioni ai webmaster sulla leggerezza delle pagine, a quelle dei moderatori delle liste sugli attachments duplicati all'infinito, il principio è chiaro: occupare meno banda possibile.

Velocità degli algoritmi

Bisogna ottimizzare tempo d'uso del processore. Anche oggi che il time sharing non esiste più e ognuno può disporre del suo processore personale per tutto il tempo che crede? Anche oggi che i milioni di cicli di istruzione al secondo cominciano a diventare unità di misura insufficienti?

Un programma veloce non solo riduce i tempi di attesa da parte dell'utente, e quindi dà l'impressione del rapporto causa/effetto tra un click e un'animazione (mentre si tratta sempre di una triangolazione tra un evento e un pezzo di programma che gestisce quell'evento), ma consuma meno cicli macchina e quindi meno elettricità (meno uranio, meno carbone, etc.).

Riciclabilità: la programmazione orientata agli oggetti

L'informazione, come pochissime altre materie prime, può e deve essere riciclata. Il tempo necessario per scrivere un algoritmo (tradotto in ore di vita, consumo di ossigeno, etc.) può essere risparmiato molte volte se l'algoritmo viene incorporato in un oggetto riusabile in un punto diverso del programma o in un altro programma o da un altro programmatore.

Qui si tocca un punto delicato, quello della possibilità di duplicare più o meno liberamente un pezzo di codice. Non me ne occupo qui, ma chi fosse interessato, può seguire le piste seguenti : Stallman, licenza GNU, copyleft, opensource...

Zen e qualità

Gli aspetti fondamentali della qualità

Ma ancora non siamo arrivati alla qualità vera, precedente e indipendente dai suoi effetti pratici, ma loro presupposto necessario. Questa è la Qualità di cui parla Robert Pirsig nei suoi due libri (Lo Zen e l'arte della manutenzione della motocicletta e Lila, entrambi tradotti presso Adelphi), e ai quali rinvio. Sono libri seri, a dispetto dei titoli, che hanno poco a che fare con lo zen ma molto con la Qualità e con il rapporto dell'Occidente con la tecnologia.

Ci sono due tipi di qualità di un software: una qualità visibile (che ha a che fare col disegno dell'interfaccia, con la scelta di una metafora adeguata...) e una qualità nascosta, che è quella relativa al codice. La qualità visibile è quella relativa all'interazione con l'utente, che rende il programma facile o difficile da usare. La qualità nascosta è quella relativa all'interazione con il programmatore, e che rende la sua vita un inferno o un paradiso.

Per quanto si possano scrivere anche programmi brutti ma efficaci, il fatto che un programma sia anche un testo che

deve essere letto e riletto da almeno una persona (il programmatore) oltre al compilatore implica che un programma leggibile, chiaro, equilibrato, si capisca meglio e venga modificato più volentieri, quindi meglio.

In realtà, così come un compositore può giudicare un pezzo musicale anche solo guardando la partitura, così un programma può essere valutato "leggendolo", a prescindere dall'esecuzione.

Il primo patriarca venuto dall'occidente

Da noi (in occidente) le divisioni tra estetica e etica, tra intuizione e intelletto hanno prosperato per anni. È difficile trovare le sorgenti di questa divisione. Platone è uno degli maggiori responsabili (l'anima tripartita), ma più in generale tutta la filosofia greca presuppone questa divisione o meglio si fonda su essa. Ma non è stato sempre così e non è dovunque così.

Il buddismo zen è una forma giapponese di buddismo. È un incrocio, o meglio il frutto di una rielaborazione successiva attraverso tre culture.

Proveniente originariamente dall'India (sesto secolo a.C.), il buddismo si è spostato verso est, in Cina, intorno al primo secolo d.C. dove ha fatto i conti con Confucianesimo e Taoismo; quindi dalla Cina nel tredicesimo secolo passò ancora più a est, al Giappone. "Zen" infatti è la parola giapponese che corrisponde al cinese "chan", e viene tradotto con meditazione.

Il suo viaggio da occidente verso oriente lo ha portato negli Stati Uniti e quindi - attraverso l'atlantico - in Europa.

Zen e té

La fortuna dello zen è dovuta probabilmente ad una grande semplicità e ad un atteggiamento "rilassato" nei confronti della dialettica mezzo-fine. La non-definizione classica di Zen di Po-Chang "Quando ho fame mangio, quando ho sonno dormo" ha il fascino disarmante di una verità che però si capisce solo a

metà.

Qui non ho intenzione di descrivere compiutamente teoria e pratica dello Zen. Non ne sarei in grado e forse non interesserebbe a nessuno. Oltre ai testi in italiano già citati, ci sono scuole ovunque e persino siti più o meno ufficiali (date un'occhiata a "[Zen Stories](#)"

(<http://www.rider.edu/users/suler/zenstory/zenstory.html>), a "[List of available IRIZ Zen Text Files](#)"(<http://www.ijjnet.or.jp/iriz/irizhtml/zen texts/cd texts.htm>) dell'International Research Institute for Zen Buddhism o a "[Dark Zen](#)"(<http://www.darkzen.com>).

Più note dello zen stesso sono le applicazioni dello zen alla vita quotidiana, dalla cerimonia del tè alla calligrafia, dalla progettazione di giardini alle arti marziali. Questo folclore giapponese ha portato alla convinzione che sia "zen" tutto quello che è perfetto, semplice, naturale, ecologico, mentre è vero esattamente il contrario. Lo zen deve e può essere applicato soprattutto alle situazioni difficili, complesse, anche a quelle più artificiali.

Koan

Un koan è un indovinello paradossale, apparentemente senza soluzione. Viene usata dalla scuola Rinzai Zen per liberare la mente dello studente dalle pastoie della logica intellettuale (quella che da noi si chiamerebbe aristotelico-tomista). Questa situazione è di casa tra programmatori: il problema irrisolvibile è loro pane quotidiano. Un vero programmatore è annoiato dalle attività di routine (scrivere semplicemente programmi), e se è sufficientemente raffinato scrive programmi che le svolgano per lui. Ma il problema lo affascina e lo cattura anche di notte, fino all'illuminazione. Quel momento - in cui improvvisamente tutto quello che era oscuro diventa ovvio - ripaga dei mesi di buio vagare.

Zen e computer

Scuole

Come le scuole di Buddismo sono

diverse e anche molto lontane tra loro, così negli anni si sono susseguite varie scuole di programmazione, ognuna con un proprio decalogo e con un modello sottostante. Dal modello imperativo a quello funzionale, da quello logico a quello orientato agli oggetti. Ogni modello porta con sé caratteristiche psicologiche non indifferenti: adottare l'uno o l'altro non è mai una questione di pura efficienza, ma di preferenze estetiche, etiche o politiche. Perciò prima ancora di cominciare a scrivere codice occorre scegliere - almeno provvisoriamente - il modello di riferimento. A ciascuno il suo.

Istruzioni per il montaggio della bicicletta giapponese

"Il montaggio della bicicletta giapponese richiede una grande pace mentale" (questo aforisma è tratto da Lo Zen e l'arte della manutenzione della motocicletta di Robert Pirsig). Sembra uno scherzo, invece è una raccomandazione seria. In Giappone non vale solo per le biciclette. Per un programmatore dovrebbe essere il primo comandamento. Nessuno inizia a lavorare ad un nuovo programma analizzando i dati e gli obiettivi. Quello che serve è una visione d'insieme della situazione problematica che il programma è chiamato a risolvere. Da questa rappresentazione chiara emerge quasi automaticamente, o in ogni caso senza che si sappia bene in che modo, un'ipotesi di strategia di programmazione. La meditazione dovrebbe essere il primo passo nell'affrontare un programma nuovo. In questo contesto, meditare significa affidarsi alle proprie conoscenze implicite, all'esperienza precedente non codificata, che è sicuramente più ampia di quella cosciente chiamata in causa direttamente dal problema. Nella base di conoscenza globale la probabilità di trovare una strategia applicabile a questo problema è molto più alta. Croce e delizia della programmazione è la flessibilità della stessa descrizione del problema.

C'è sempre un altro modo di vedere il problema, in cui la base diventa il tetto e il tetto la base.

Un motto che si applica bene a questo stadio di lavoro è "Appoggiatevi saldamente ai vostri principi, in modo che cadano presto".

I sogni del programmatore

Ogni programmatore, se chiude gli occhi, si rappresenta l'opera a cui sta lavorando secondo una qualche metafora. Come spazio fisico, come organismo, come macchina, come sistema politico, come musica. È su questa rappresentazione interna che lavora realmente. È questa visione che è difficile comunicare e che rende così difficile continuare il lavoro di un altro. Quasi sempre, si preferisce ricominciare da capo, per avere la possibilità di crearsi la propria rappresentazione.

Ritmo: la forma degli algoritmi

Il ritmo è un elemento fondamentale della qualità del codice. È un elemento del tutto non significativo dal punto di vista dell'efficienza e dell'ecologia, eppure si potrebbe anche dire che è la caratteristica che più resta impressa di una porzione di codice.

Capita a tutti di cercare - e di trovare - un pezzo di codice di cui si ricorda la forma visiva, il ritmo, anche se non si è in grado di ricordare esattamente cosa contiene. Quando il programmatore rilegge un programma segue gli accapo, i termini riservati che segnano le strutture; in altre parole, tiene il tempo.

Per esempio:

un ritmo binario è semplice ma ossessivo;
 un ritmo ternario è mosso ma permette di porre accenti;
 un ritmo quaternario è riposante ma pesante, noioso;
 un ritmo quinario è aperto ma inquietante; eccetera...

Provate a leggere, a voce alta, questo:

```
ask "What is your Number?"
number = it
if number is 10
beep 4
else
if number is 7
beep 3
else
if number is 5
beep 2
else
if number is 3
beep 1
end if
end if
end if
end if
```

e adesso questo:

```
ask "Number, please?"
number = it
conditions
when number is 10
beep 4
when number is 7
beep 3
when number is 5
beep 2
when number is 3
beep 1
end
```

Non è solo questione di efficienza o di leggibilità. La prima soluzione, grazie all'uso dell'indentazione, è leggibile quanto la seconda. Ma se provate a leggerla a voce alta. Il when è binario, l'if ternario. La prima versione è drammatica: c'è un crescendo, un fortissimo (beep 1) e un finale. La seconda versione è leggera come una filastrocca: non enfatizza nessun punto particolare e la fine arriva a sorpresa.

Scegliere l'una o l'altra dipende da gusti letterari, esperienze musicali. Oppure da preferenze grafiche: la prima versione ha il profilo diagonale di una vela, la seconda di una colonna compatta.

A proposito dell'indentazione: uno standard ormai universale al cui inventore

non saremo mai abbastanza grati. È un micromodello che riportando in orizzontale la gerarchia dei livelli logici di una porzione di codice fornisce un appiglio visuale. Non si tratta solo di un aiuto visivo per non scordare parentesi aperte. Il modello sottostante, anche se implicito, è chiaro: il flusso del programma scende sui gradini delle tabulazione e degli accapo, fino ad un punto da dove non può far altro che risalire.

Eppure non è raro vedere codice (anche scritto da professionisti) del tipo:

```
if numero = 10; beep 3; else beep 1; end
```

che è più compatto ed equivalente per l'interprete, ma fa male alla pancia.

Equilibrio: le strutture di dati

Un programma resta in attesa dei dati, come un samurai in attesa dell'attacco dell'avversario. Attesa significa equilibrio. Ma se il suo equilibrio è statico, un flusso di dati superiore a quello previsto può causare un trabocco.

Invece un equilibrio dinamico significa capacità di adattarsi alle situazioni sconosciute.

Dichiarare le variabili all'inizio dell'handler è una prassi raccomandata da tutti.

Ancora meglio se si dichiara anche il tipo della variabile, soprattutto quando così facendo si limita la quantità di risorse che il sistema deve riservare per quella variabile.

Per esempio, usare una string per memorizzare un valore logical (che può essere solo vero o falso) è uno spreco di risorse. O usare un real per rappresentare un intero.

Un altro esempio: un array statico è più efficiente e più ecologico di uno dinamico, perché alloca esattamente la memoria necessaria. Questo è il livello di qualità ecologico.

Da un altro punto di vista, le strutture rigide sono costose a crearsi e a modificarsi, anche se sono economiche durante l'uso quotidiano. Le strutture

flessibili invece permettono al programma di adattarsi alla situazione, anche se richiedono risorse supplementari. La struttura dati più dinamica è la lista. In alcuni linguaggi (il LISP ma anche il bistrattato LOGO, che da questo punto di vista è un signor linguaggio) è praticamente l'unico tipo di dati complesso. Una volta detto che le funzioni in LISP sono esse stesse liste, si capisce la potenza di questa impostazione.

Una lista non ha dimensioni predefinite, può contenere qualsiasi tipo di dato, anche mescolati fra loro. Openscript fornisce diverse primitive per accedere alle liste, tra cui i classici push e pop (il loro nome è dovuto alla funzione standard in uso negli anni '60), che sono molto più veloci di put e get. La limitazione più grossa all'uso delle liste è il fatto che il separatore debba per forza essere una virgola, il che impedisce di inserire stringe che contengono virgole all'interno di liste.

Un programma deve essere in condizione di risolvere le situazioni problematiche che si possono presentare nel corso della sua vita: dati di tipo errato, flusso di dati eccessivo, inattività prolungata, scarsità di risorse, eccetera. Non è possibile prevedere tutte queste situazioni. Sarebbe anche contrario alla logica stessa della programmazione.

Il programmatore non deve risolvere tutti i problemi, ma costruire un programma che sia in grado di sbrigarsela da solo.

Flusso: il passaggio del controllo

Un buon programmatore si cura che il flusso del controllo nei suoi programmi sia regolare, forte, che tocchi tutti i punti del sistema, che fluisca attraverso di esso come il respiro.

La prima onda è l'inizializzazione. Alla partenza si eseguono tutte le operazioni che "risvegliano" il programma e lo preparano ad affrontare una nuova giornata.

Poi l'attesa - indefinita - di un evento; e in seguito all'evento, il flusso di messaggi che comunicano l'evento al sistema (su su su).

Come un boomerang torna al punto di partenza, così il messaggio dovrebbe tornare, in qualche forma, al mittente. C'è una ragione pratica, come al solito: così ci si assicura che tutto sia andato per il suo verso. Per questo spesso le funzioni ritornano, se non altro, almeno un codice d'errore. Ma c'è anche una ragione zen, per spiegare la quale mi lascerò andare a un po' di personalizzazione e animismo. Immaginare un programma che respira aiuta a sentirsi uniti a lui; e la separazione, soprattutto affettiva, del programmatore dal programma (e in generale dal computer) è il vizio di fondo che rende impossibile qualsiasi risultato pratico positivo nell'interazione con il computer. Un programmatore che miri a "fregare" il programma non farà mai un buon lavoro.

È lo stesso problema delle parti di codice commentate come test e poi lasciate per sempre lì, come cicatrici mai rimarginate. Non fanno male all'interprete, che le salta, ma dicono a chi legge che quel programma non è stato mai ripulito, è stato abbandonato non appena ha mostrato di reggersi sulle sue gambe. Come avrebbe detto Pirsig, c'è un solo modo di aggiustare una motocicletta: tenerci.

Torniamo al boomerang.

Ci sono due versioni di questo ritorno.

- Nel modello "funzione" ogni interazione tra due oggetti è uno scambio che restituisce un valore a chi ha iniziato la conversazione. Come in una staffetta, il corridore che ha consegnato il testimone resta fermo, congelato in attesa della restituzione di un oggetto in cambio del testimone.
- Nel modello "messaggio" il messaggero ha terminato il suo compito una volta consegnato il

messaggio; sarà cura del sistema inviare a sua volta un messaggio di risposta al primo mittente.

In pratica i due modelli possono essere implementati in openscript tanto con funzioni (to get) che con messaggi (to handle). La differenza sta più nel fatto che una funzione ha senso se il valore che torna è utilizzato. Per esempio, l'algoritmo classico ricorsivo per l'inversione di una lista lavora su di un elemento della lista per volta e richiede che il lavoro sul resto della lista venga fatto da qualcun altro; alla fine la funzione restituisce un assemblaggio del proprio lavoro e di quello dell'altra funzione. In questo caso particolare, la funzione passa il testimone a se stessa, ma questo affascinante avvistamento sul posto non crea nessun problema all'interprete, come potrete vedere se seguite l'esecuzione nella finestra di debug.

```
to get rovesciaFN lista
if itemcount(lista) is 1
return lista
else
pop lista into primo
nuovalista = rovesciaFN(lista) & ", "& primo
return nuovalista
end
end
```

Dal punto di vista dell'informatica tradizionale, il processore esegue una operazione alla volta (prefetch escluso). Nel modello classico, un processore ha una coda di istruzioni e indirizzi in ingresso e fornisce in uscita indirizzi.

Dal punto di vista del programmatore, il programma è fermo, ed è lo scenario, la mappa del territorio dove si svolge l'azione. In altre parole, c'è un punto di vista che si sposta in tutto il programma, in base ai percorsi che sono stati predisposti dal programmatore e in base al verificarsi di eventi. In questo senso una rete ipertestuale non è un'applicazione particolare dell'informatica, è un modello della stessa

programmazione.

La programmazione è un'opinione

Un problema può avere molte soluzioni diverse. E le soluzioni possono avere qualità diversa, dal punto di vista dell'efficienza, dell'ecologia e della qualità profonda.

Esempio: trovare il numero d'ordine di una lettera.

Confrontate queste tre soluzioni:

```
1. conditions
when lettera is "a"
set numero to 1
when lettera is "b"
set numero to 2
.....
```

```
2.
set numero to (CharToAnsi(lettera) - 43)
```

```
3.
set numero to
ItemOffset(lettera, "a,b,c,...")
```

La versione 1 è di comprensione immediata. Si può scrivere usando a ripetizione CTRL+C / CTRL+V. Però è ripetitiva, soggetta ad errore e poco manutenibile.

La 2 è efficiente, compatta, ma non evidente; si basa su di una proprietà (la tabella di codici ANSI) che è fortemente convenzionale e fissa.

La 3 è naturale, elegante e veloce. Si basa sull'ordine alfabetico, che è un fatto noto. Ma può essere modificata facilmente. Per esempio, se si dovesse usare un alfabeto diverso da quello americano (usato nell'ASCII) sarebbe facile modificarla. Crea un modello naturale della situazione (l'alfabeto) e manipola il modello.

Un programmatore esperto userebbe quasi sicuramente la 2. Questa soluzione consiste essenzialmente nel realizzare una black box, una macchina che risolva un problema dato.

La 3 invece è una rappresentazione di

una situazione e la progettazione di una strategia per muoversi all'interno di questo spazio. Se cambia la situazione problematica, può essere facilmente modificata tanto la rappresentazione quanto la strategia.

Toolbook è un linguaggio zen?

Un esercizio zen

In questo capitolo vi propongo un esercizio di qualità: provare a vedere TB come un linguaggio Object Oriented e usarlo il più possibile consistentemente con questa interpretazione.

Nella programmazione tradizionale si parte dagli obiettivi che si vogliono raggiungere, si studiano i dati disponibili e si cerca di costruire una macchina - astratta - che trasformi questi in quelli. La programmazione a oggetti consente invece di rappresentare direttamente la maniera in cui una persona vede una situazione (oggetti dotati di proprietà e connessi fra loro).

Sui vantaggi in termini di efficienza ed ecologia di un approccio del genere non voglio entrare. Il vantaggio che mi interessa non è, ancora una volta, nel codice di per sé, ma nella maniera di interagire con il codice da parte del programmatore.

Se posso pensare al mio programma come una versione ridotta del mondo, posso anche utilizzare tutta la mia esperienza reale per inventare soluzioni.

Di sicuro TB non è un ambiente di programmazione completamente ad oggetti, come C++ o più ancora Smalltalk. Tuttavia, la struttura di dati principale offerta dal sistema lo è: i libri possono essere pensati come classi di sfondi, gli sfondi come classi di pagine, le pagine come classi di oggetti.

Il sistema stesso potrebbe essere visto come un browser di classi di oggetti. Naturalmente ci sono grosse limitazioni. Per esempio, non è possibile creare una nuova classe, ma solo istanze di classi

già esistenti. Le classi non possono essere distrutte indipendentemente dai propri componenti. Eccetera.

Un modo semplice per simulare meglio le classi in TB è costituito invece dai gruppi di oggetti.

Gruppi come classi

I gruppi si comportano come classi da diversi punti di vista:

- possono essere creati
- possono essere distrutti indipendentemente dai propri componenti (a differenza delle pagine e background)
- possono contenere altri gruppi
- supportano un meccanismo di ereditarietà limitato alle proprietà di sistema: per esempio si può assegnare un colore ad un gruppo, col risultato di modificare il colore di tutti i componenti del gruppo
- supportano la gestione dei messaggi

Si comportano come sottoclassi delle pagine o dei background, nel senso che non possono contenere oggetti di pagine o background diversi.

Inoltre, possono contenere solo oggetti grafici (bottoni, campi, rettangoli, linee, etc.)

Ma è sempre meglio di niente.

Per estendere il meccanismo dell'ereditarietà delle proprietà di sistema anche a proprietà utente, si può usare un handler ricorsivo di questo tipo:

```
to handle heritate propertyP,valueP
get my objects
while it is not null
pop it into destVL
if object of destVL is group
send heritate propertyP,valueP to destVL
else
execute ("set"&& propertyP &&"of"&&
destVL&&" to"&& valueP)
end
end
end
```

Normalmente, TB gestisce i messaggi secondo la regola "se non sai come fare, chiedi a tuo padre", ovvero un messaggio che non può essere gestito viene passato automaticamente alla classe superiore. Un comportamento utile da simulare è quello per cui per inviare un messaggio a tutti gli oggetti di una classe è sufficiente inviarlo alla classe stessa.

```
to handle broadcast messageP
get my objects
while it is not null
pop it into destVL
if object of destVL is group
send broadcast messageP to destVL
else
execute "send"&& messageP&& "to"&&
destVL
end
end
end
```

Polimorfismo e altre diavolerie

A questo punto, se si vuole si può gestire il polimorfismo.

Per esempio, per assegnare una stringa agli oggetti di un gruppo, a prescindere dal fatto che siano bottoni o field, si può inserire il seguente handler nel gruppo.

```
to set mytext to stringaP
conditions
when object of target = button
set caption of target to stringaP
when object of target = field
set text of target to stringaP
end
end
```

L'uso di questo handler diretto è semplice:

```
set mytext of selection to "Questo è il mio
testo"
```

ma va fatto per tutti gli oggetti, uno ad uno.

Per eseguirlo in una sola volta su tutti gli oggetti del gruppo invece:

```
send heritate mytext, "Questo è il mio
testo" to group G1
```

È possibile anche mandare un messaggio a tutti i componenti di una classe, anche senza conoscerli esplicitamente.

Per esempio, se si volessero muovere tutti gli oggetti del gruppo a saltelli, uno per volta, si potrebbe scrivere questo handler:

```
to handle salta
x = 100
y = 100
move target by x,y
move target by -x,-y
move target by x,y
end
```

Ma invece di inviarlo oggetto per oggetto, è possibile passarlo a tutti gli oggetti del gruppo in una sola volta:

```
send broadcast hitch to group G1
```

Tutti questi handlers usano il comando `execute`, che invoca l'interprete su una stringa. `Execute` è molto lento, ma ha l'enorme vantaggio di permettere di eseguire codice openscript fornito dall'utente oppure generato a runtime. In teoria, questo vuol dire che se il programma incontrasse una difficoltà per la quale non è stato adeguatamente attrezzato, potrebbe costruirsi da solo il codice necessario.

Si apre qui la possibilità di insegnare le zen non più ai programmatori ma direttamente ai programmi ;)

Conclusioni

Questo testo non ha la pretesa di oggettività di un contributo tecnico. Ma non è neanche un divertissement fine a se stesso. La sua natura leggera è in parte dovuta al tema e in parte agli eventi: durante il THUG il seminario non si è tenuto, a causa di una *défaillance* del

suo relatore. Se ci fosse stato, alcune delle tesi proposte avrebbero potuto essere discusse direttamente, gli esempi sarebbero stati scritti a più mani e si sarebbe potute fare osservazioni sulle somiglianze e differenze.

Ma se questo testo avrà suscitato interesse o dubbi, allora se ne potrà parlare al prossimo THUG, nel corso del seminario - reale, questa volta - sullo "Zen e l'arte della programmazione".

. STANZA VICINO AL CINEMA:

Durante tutta la giornata: CANDIDA, LA TV ELETTRODOMESTICA

DESCRIZIONE:

In occasione della terza edizione dell'hackmeeting prevista per il giugno 2000 a Forte Prenestino, noi di "Candida. La prima televisione elettrodomestica." abbiamo intenzione, insieme ad alcune persone di avana.net, di occuparci dello streaming video in rete di momenti ed eventi particolari della tre giorni. Si tratterà di fare microdifferite o dirette del materiale che verrà ripreso dalle nostre telecamere nei giorni stessi.

Un lavoro simile e parallelo a quello che sarà proprio di radiocybernet, con la differenza che esploreremo con la trasmissione video piuttosto che con quella audio.

Abbiamo intenzione per questo di utilizzare una macchina con sistema operativo linux e programmi specifici per lo streaming e per la codifica video che abbiano la licenza di freeware (se ne esistono). Candida, il cui intento è quello di infestare l'etere e il mainstream televisivo, ha un interesse particolare per la trasmissione e la diffusione di materiale video via rete, per adesso abbiamo un archivio video delle

rubriche delle prime 9 puntate di Candida che sono andate in onda su teleambiente un'emittente regionale del Lazio, visibili all'URL:

<http://candida.kyuzz.org>.

I video sono stati codificati con il Real Producer G2 e sono visibili con il Real Player basic 7. Individuiamo nell'hackmeeting una splendida occasione

per creare un laboratorio che sperimenti con la codifica e la trasmissione mediante l'utilizzo di free software.

La trasmissione video via rete, quella che ora tutti chiamano Web Tv senza sapere assolutamente cosa sia, è un campo di grande interesse per il mercato al momento, nel quale si investono grandi capitali da parte delle televisioni pubbliche, private e satellitari. Chi ha deciso di puntare sul commercio elettronico mira alla creazione di utenze di rete per avere potenziali consumatori. L'interesse del mercato si interseca con il nostro, che è quello della rapida diffusione, e creazione collettiva di materiale visivo, e una linea di ricerca che non va lasciata sguarnita, ma che anzi va invasa al più presto in questo momento di incubazione, spingendo sull'utilizzo di programmi e sistemi operativi liberi.

SABATO 17 GIUGNO 2000

.. PUB:

11.00 – 13.00 E 16.00-18.00:

WORKSHOP/SEMINARIO SUL TEMA DELL'ACCESSIBILITA' DELL'INFORMAZIONE IN RETE IVI COMPRESA LA TEMATICA DELLA REPERIBILITA' DELL'INFORMAZIONE

**STESSA (a cura di Ferry Byte, Arcelele,
Claudio Parrini e Shine)**

i materiali in rete di riferimento sono

<http://strano.net/chaos> e

<http://www.ecn.org/xs2web>

IMPORTANTE: il seminario dovra' avere luogo in uno spazio del forte facilmente accessibile per i disabili

DESCRIZIONE:

"Perche' e' giusto e opportuno prevedere che esistano pagine web accessibili per chiunque indipendentemente dalle caratteristiche dell'interfaccia utente. Quindi per tutti coloro che non si possano permettere hardware o software di grido o che comunque per necessita' o per situazioni emergenziali si trovano a doversi arrangiare con attrezzature arrangiate

Anche per:

- chi e' incapace di distinguere i colori, e' ipovedente o non puo' vederci affatto
- per le persone sofferenti di disturbi epilettici insofferenti ai normali monitor (che secondo alcune teorie recenti si ritiene possono innescare le loro crisi, tali persone devono perciò ricorrere a schermi a cristalli liquidi, questi schermi non consentono grandi varietà di colore se non soltanto dei livelli di grigio ed inoltre permettono la visione su uno schermo con risoluzione generalmente non superiore a 800x600 pixels più comunemente 640x480 pixels) e soprattutto possono risultare danneggiate da repentini cambiamenti di luce e immagini
- chi ha disabilità motorie
- chi ha difficoltà nell'udire suoni
- chi ha uno schermo solo testuale, un piccolo schermo o una connessione Internet molto lenta
- chi ha la versione precedente di un browser, un browser completamente

diverso, un browser basato su dispositivi di sintesi vocale o un diverso sistema operativo

- chi ha inabilità cognitive o psichiche
- chi per scelta vuole avere il controllo sulla propria navigazione

Per pagina web accessibile intendiamo quindi una pagina facilmente reperibile (che si trova quindi nelle prime trenta posizioni nei risultati di ricerca di un qualsiasi motore di ricerca), tradotta in una lingua comprensibile dall'utente di riferimento e con un aspetto grafico e una struttura tali che sia facilmente comprensibile da chiunque. Per facilitare la comprensione di una materia abbastanza complessa e in evoluzione come quella dell'accessibilità alle pagine web abbiamo scelto di indicare solo quei suggerimenti tecnici che vanno verso un tipo di accessibilità universale e pressocché assoluta indipendentemente dagli sviluppi futuri e contemporaneamente accennare agli scenari prossimi elencando una serie di links utili per approfondimenti di vario tipo per informarsi e reperire quelle informazioni e quelle risorse, come sugli ausili (uno per tutti gli screen reader che traducono testo in braille o audio per chi ha problemi di vista), che non abbiamo avuto modo di trattare, per ora... Intendiamo inoltre promuovere una campagna di sensibilizzazione su questi argomenti in primo luogo cercando di sensibilizzare il settore pubblico, quello dell'autorganizzazione ma anche quello privato nell'adottare pagine "accessibili" rispetto al problema della visibilità sui motori di ricerca, noi, nel nostro piccolo, a livello per ora solo teorico proviamo a fornire alcune indicazioni che - utilizzando alcuni paradigmi del chaos - potrebbero essere utili per uscire da queste dinamiche, per lo più sconosciute alla maggior parte della cittadinanza telematica, ma a nostro parere segnate da evidenti e pesanti manipolazioni

dell'informazione e dell'accesso all'informazione.

prima proposta: SCANNERIZZAZIONE CAOTICA DELLA RETE.

Studiare sistemi che permettano una scelta casuale della parte di rete da indicizzare (qualsiasi motore di ricerca o portale può solo indicare-selezionare una parte della rete visto le sue dimensioni e caratteristiche raggiunte negli ultimi anni) e contemporaneamente cercare di privilegiare interessi non solo di business ma anche di cordate di interessi no-profit tentando per esempio di sviluppare motori di ricerca che indicizzino tratti specifici della rete come quello del volontariato, dell'antagonismo ecc. ; privilegiare cioè meccanismi (tutti da sperimentare) che siano orientati al reperimento di informazione in Internet partendo da indirizzi casuali ma possibili della Rete stessa. Un'ipotesi sarebbe quella di indicizzare parte della Rete partendo da IP numerici oppure nomi simbolici possibili. Auspichiamo quindi che per una certa (piccola) parte, i motori di ricerca prendano in considerazione la possibilità di generare indirizzi simbolici possibili della rete come e assumere l'eventuale sito trovato per caso con i suoi relativi links come parte delle loro basi di conoscenza.

Lo stesso potrebbe essere fatto ipotizzando indirizzi numerici ip della Rete fra quelli potenzialmente destinati ad ospitare server web. Una volta "azzeccato" un indirizzo IP numerico o indirizzo simbolico alfabetico il motore di ricerca esplora i LINKS ospitati arrivando così magari anche ad indirizzi che non si possano permettere di essere rappresentati direttamente da domini alfabetici o indirizzi ip numerici.

Seconda proposta: RANKING CAOTICO DEI RISULTATI DI RICERCA.

I motori di ricerca dovrebbero fornire risultati delle ricerche conseguite sulla base del semplice caso neutralizzando così eventuali strategie di marketing che vedono risultati di ricerca

viziati a favore dei "soliti raccomandati" ;-)

Soprattutto i meta-motori di ricerca che spesso sono gestiti da singoli o enti indipendenti potrebbero soppesare la possibilità di rimescolare casualmente la graduatoria dei risultati di una ricerca effettuata sui vari motori al fine di non privilegiare nessun* e non penalizzare nessun altr* nella presentazione dei risultati della ricerca effettuata.

Entrambe queste proposte si fondano su un'idea artistica, - accostando la pratica della ricerca a quella dell'arte ponendo il ricercatore nel ruolo di artista indagatore meta-diffusore dell'informazione, colui che si preoccupa in maniera laterale e creativa non tanto di limitarsi a produrre qualcosa in Rete, ma di offrire prospettive trasversali per fruire delle esperienze che già esistono -, per cui una possibile efficace democrazia è quella che si basa su una sorta di PREMIO CASUALE ammettendo che ogni entità ha le stesse probabilità di successo (amici più eruditi di noi ci hanno detto che gli antichi greci in passato applicavano in alcuni casi questa sorta di democrazia casuale). "

In preparazione a tale evento e' stao pubblicato su www.ecn.org/xs2web un pre_sito del nascente portale sull'accessibilita'. Fra le idee uscite c'e' anche quella di fare un'indagine sulle caratteristiche generali di alcuni tipi di "Pagine Web". Per ottimizzare gli strumenti di ricerca e iniziare a capire quali sono gli elementi comuni (ed inaccessibili) maggiormente utilizzati si e' deciso di testare 4 tipi di pagine:

- portali di accesso a internet
- motori di ricerca
- HomePages delle "realta' antagoniste" ospitate sul sito

<http://www.ecn.org/presenze/csc.htm>

-- Pagine di siti che si occupano di accessibilita' e web.

14.00 – 16.00: CONFERENZA SULL'SNMP (a cura di Tho, Bluca e P@sky)

18.00 – 20.00: SEMINARIO SULLE LICENZE FREE (a cura di Franco Vite; Relatori Eugenia Franzoni e Christopher R.Gabriel)

DESCRIZIONE:

Svolgimento:

Da qualche mese a questa parte è iniziato un gran parlare delle magnifiche und progressive sorti dell'Opensource. Ad alcuni di noi (a me in particolare ...) la cosa ha puzzaato assai, in particolar modo da quando colossi multinazionali come SUN, IBM, COREL, etc etc si sono buttati sulla cosa.

Il seminario in questione si pone l'obbiettivo di discutere delle differenze che stanno alla base delle diverse "licenze free". In particolar modo tra le licenze che sono state accettate nella "Opensource definition" ([Http://www.opensource.org](http://www.opensource.org)) e la piu' classica e antica delle licenze free: la GPL della Free Software Foundation ([Http://www.gnu.org](http://www.gnu.org)). E di cercare di capire perche' (sempre che ci sia un perche' ...) le multinazionali di cui prima si sono buttate sull'Opensource quando c'era gia' bella pronta (da anni) la GPL (e la LGPL, meno restrittiva). Ultimamente, e la cosa mi pare assai interessante, sono nate due licenze contrapposte (si fa per dire; ma rende l'idea :) per la documentazione, facenti parte, una della *famiglia* Opensource (la "Open Publication License"), l'altra della FSF (la "GNU Free Documentation License").

Il seminario prevede un paio di interventi

iniziali, a mo' di introduzione alla materia, svolti, uno da un'ospite che dovrebbe trattare delle licenze varie con grande cognizione di causa; e l'altro dal sottoscritto, che dovrebbe delirare su assurdi complotti delle SIM atti a fottere l'intero movimento del Free Software, con accenni al Pynchon, Dick e alla trilogia degli Illuminati, per concludersi con il mio svenimento mentre cerco di copiare Stalman nella preghiera a Emacs (ovviamente scherzo forse).

DALLE 20.30: DIBATTITO SU "GLOBALIZZAZIONE, TELEMATICA E MOVIMENTI ANTAGONISTI"

DESCRIZIONE:

GLOBALIZZAZIONE, TELEMATICA E MOVIMENTI ANTAGONISTI

I movimenti antagonisti si presentano essenzialmente come attori segmentati, acefali, reticolari, le cui caratteristiche di informalità e di bassa strutturazione organizzativa impongono una fluidità comunicativa molto accentuata, in grado di garantire forme di coesione e di coordinamento tra le varie isole dell'arcipelago o, meglio ancora, tra i "nodi" della rete.

Di fatto l'informalità e la debolezza nella struttura organizzativa, la mancanza di accessi diretti ai mezzi di comunicazione di massa, da sempre hanno spinto i movimenti sociali verso la sperimentazione di forme e tecniche inedite di comunicazione politica. Si tratta di una sperimentazione aperta, "tattica", nella quale rintracciare, pur evidenziando il carattere non neutro della tecnica, gli usi possibili, le eventuali potenzialità antagoniste insite nell'innovazione tecnologica. L'esperienza delle radio libere negli anni settanta rappresentò in questo senso il primo segnale di un incessante

tentativo da parte dei movimenti di adeguare e applicare le continue innovazioni tecnologiche rispetto alle proprie necessità comunicative, ai propri obiettivi politici e sociali.

Da questo punto di vista, il carattere orizzontale e rizomatico dei movimenti trova nella rete telematica uno strumento di valorizzazione della sua stessa natura, un sistema in grado non solo di superare i limiti organizzativi interni preservando informalità e orizzontalità, ma anche di raggiungere un livello di comunicazione verso l'esterno che nessun altro mezzo di comunicazione in precedenza era riuscito a garantire.

All'informazione telecomandata dall'alto, all'unidirezionalità e all'omologazione dei flussi comunicativi, alla costante marginalizzazione e criminalizzazione delle voci di dissenso dentro i circuiti della comunicazione di massa, la telematica antagonista ha contrapposto canali comunicativi autonomi e indipendenti, del tutto autogestiti e svincolati sufficientemente dalle norme di controllo e di censura politica (anche se con sempre più sistematicità si susseguono i tentativi di mettere bavagli alle voci antagoniste presenti su internet, come recentemente contro Nod50 e Burn). Non è quindi un caso che fin dalle origini delle prime BBS, ritroviamo l'antagonismo sociale in prima fila sulle frontiere elettroniche, così come - malgrado la tendenza sempre più feroce della mercificazione del cyberspazio - parallelamente alla diffusione "di massa" (bianchi, maschi, occidentali, of course) di internet sono cresciute nelle maglie della rete forme più o meno dispiegate di resistenza e di riappropriazione diretta delle potenzialità sociali e antagoniste della telematica.

Attraverso la rete, infatti, si è andata progressivamente rafforzando e ramificando la comunicazione dal basso tra gli attivisti dei movimenti sociali, tra movimenti geograficamente distanti ma anche dissimili nella natura e negli obiettivi, al punto tale da far

disvelare una dimensione internazionale finora del tutto estranea, per certi versi impossibile, per i movimenti antagonisti e rompere la logica delle compartimentazioni e del settorialismo; percorsi di sincretismo antagonista che fanno presagire, come all'indomani della "rivolta di Seattle", nuove dimensioni di opposizione e antagonismo sociale capaci di impattare contro le dinamiche della globalizzazione selvaggia. Sappiamo come le "giornate di Seattle" non nascano dal nulla, ma sono piuttosto il risultato di un lento, costante e sotterraneo lavoro di comunicazione, di contaminazione e di interconnessione tra la molteplicità di soggetti che vivono sulla loro pelle gli effetti nefasti del neoliberismo.

La telematica ha giocato certamente un ruolo fondamentale nel riannodare i fili di una trama comune (che ha oltrepassato e messo in discussione le stesse "gabbie" delle comunità virtuali), di una capacità di confronto e di mobilitazione collettiva attraverso la quale il coro dei "senza voce" si è fatto talmente assordante da azzittire e rimandare a casa i padroni del mondo con un pugno di mosche. Non sarà certo la telematica il "motore" di un processo di ricomposizione dei soggetti non garantiti su scala mondiale, il megafono planetario con cui urlare lo slogan "proletari di tutto il mondo unitevi", tuttavia oggi crediamo nell'utilizzo della telematica antagonista come uno strumento fondamentale per la costruzione di network di lotta che si contrappongono materialmente ai processi della globalizzazione neoliberista, capovolgendo la struttura di questo sistema globale fondato sulla libera globalizzazione dei dominanti e sulla localizzazione forzata dei dominati.

Pur partendo dalla materialità del conflitto sociale, con i piedi improcrastinabilmente

ancorati al proprio contesto locale, alle contraddizioni e alle lotte che si esprimono sul proprio territorio, dobbiamo tessere un sistema di comunicazione globale che diventi il sistema nervoso di un nuovo internazionalismo, espressione reale e diretta dei movimenti. In questo la telematica, internet, la posta elettronica, ci potranno essere di grande aiuto.

Alcuni punti di discussione :

1) INFORMAZIONE/CONTROINFORMAZIONE

ECN, TMC, AVANA e le altre situazioni della telematica antagonista in Italia già oggi svolgono un ruolo notevole e fondamentale sul terreno della comunicazione nei movimenti e dei movimenti.

In tutt'Europa crescono esperienze simili, una sorta di rete di "agenzie di comunicazione antagonista" che, pur nelle loro differenze, garantiscono un'informazione autonoma anche in tempo reale e una copertura rispetto gran parte delle iniziative di lotta e di mobilitazione, indispensabile oggi per la valorizzazione e la circolazione delle stesse.

Dobbiamo quindi ragionare e affinare le modalità di interconnessione tra le esperienze consolidate della telematica antagonista e le dinamiche della conflittualità sociale che si esprimono nei nostri territori, per moltiplicare visibilità ed efficacia, amplificare le rivendicazioni, i dibattiti, le contraddizioni di un agire antagonista sempre più complesso e multiforme.

Piuttosto che rincorrere il terreno della spettacolarizzazione e della notiziabilità imposta dai mass-media, è necessario quindi che i movimenti e i soggetti antagonisti inizino a interrogarsi e darsi anche risposte "sistematiche" sul rafforzamento dei propri canali di comunicazione autonoma e sul come far

uscire questi dal sempre incombente ghetto dell'autoreferenzialità.

Da questo punto di vista, le esperienze nordamericane dei media indipendenti ci mostrano - non solo rispetto alle mobilitazioni di Seattle e Washington - una capacità sistematica di utilizzo e di applicazione degli strumenti multimediali in grado di imporsi e lanciare una sfida di metodo e di contenuto anche nei circuiti tradizionali dei mass-media.

Del resto, le battaglie di denuncia e di controinformazione sulle politiche criminali delle multinazionali (Mc Donald's, Shell,) ci mostrano come sia possibile impattare nell'immaginario collettivo pur partendo da una considerevole disparità di mezzi a disposizione.

2) INTERNAZIONALISMO

Il 18 giugno 1999, il 30 novembre 1999, il 1 maggio 2000 sono state giornate di mobilitazione internazionali contro la globalizzazione, giornate in cui i movimenti hanno acquisito la sfida e la consapevolezza di giocare la partita sullo stesso tavolo della controparte.

Di fatto, la comunicazione telematica ha rafforzato l' "onda lunga" di Seattle, provocando una reazione a catena su tutto il pianeta e rafforzando mobilitazioni internazionali al tempo stesso unitarie e molteplici, così come ha avuto un ruolo imprescindibile nella costruzione delle mobilitazioni contro il MAI e contro tutti quei luoghi decisionali "oscuri" dei padroni del mondo.

Ci troviamo dunque dinanzi ad un salto di qualità non indifferente : ad un primo momento in cui la telematica ha garantito informazioni e notizie sulle lotte in tutti gli angoli del mondo - dalla Selva Lacandona alla Corea del Sud, passando per il Sud Africa, ora essa stessa si accinge a divenire "colonna vertebrale" di un'interconnessione globale, di network internazionali di lotta e di resistenza contro il capitale globale.

Gli zapatisti già da tempo avevano

evidenziato questa potenzialità, sottolineando tra l'altro come i flussi di comunicazione telematica stimolano, e non sostituiscono, la necessità di incontri, di discussioni "ravvicinate" : gli "Incontri Intercontinentali per l'umanità e contro il neoliberalismo" andavano letti in questo senso, ma probabilmente anticipavano una consapevolezza che solo oggi affiora in tutta la sua complessità.

Ora, sull' "onda lunga" di Seattle, è possibile rafforzare relazioni e comunicazioni a livello planetario, anche e soprattutto partendo dalle reti di comunicazioni antagoniste presenti sul globo.

Non è però nostra intenzione disquisire sulle teorie, sulla possibilità, attraverso la telematica, di costruire una quinta internazionale che superi - a partire da internet - i limiti e gli errori delle quattro internazionali precedenti in termini di burocratizzazione e verticalizzazione.

Ci interessa piuttosto partire dalla materialità delle sperimentazioni in atto, delle relazioni costruite e da costruire. Confrontarsi quindi sulla costruzione del Mayday, del controvertice di Bologna, e progettare insieme la costruzione, sul versante della comunicazione, dei prossimi appuntamenti internazionali, come i border camp che quest'estate "assedieranno" congiuntamente i "punti nevralgici" delle frontiere della Fortezza Europa.

3) ATTIVISMO TELEMATICO

Accanto alla comunicazione dal basso, la diffusione di massa della telematica ha aperto un ulteriore terreno di scontro e di conflitto che vede schierati, ancora una volta, da una parte i gruppi dominanti, che sostengono e sponsorizzano una sempre più selvaggia colonizzazione ed un incontrastato predominio delle leggi del profitto nel mondo virtuale, e dall'altra gli "attivisti telematici" che, a partire dalla valorizzazione della comunicazione interattiva, orizzontale, multidirezionale della rete, sperimentano

quotidianamente inedite strategie di resistenza e di disobbedienza elettronica: NetStrike, mailbombing, cybersquatting sono probabilmente forme ancora embrionali di una mobilitazione telematica che supporta e sorregge le mobilitazioni e le conflittualità sociali che si esprimono nel mondo.

Dobbiamo necessariamente discutere sull'efficacia, sulle possibili strategie di perfezionamento, sulle modalità per diffondere a macchia d'olio queste pratiche di lotta che ridefiniscono la "geografia situazionale" della protesta, configurandosi come primo esempio di un processo combinato di deterritorializzazione e mondializzazione della stessa.

. CINEMA:

14.00 – 20.00: SEZIONE ARTE (a cura di Tommaso Tozzi, T_Bazz e Snafu)

L'ARTE COME PRATICA REALE:

**RIFLESSIONI SULL'USO DELLA
TECNOLOGIA NELLA SPERIMENTAZIONE
ARTISTICA DALLE AVANGUARDIE**

ALL'ARTE IN RETE ;

**Parte II: Situazione attuale; Arte e
Nuove Tecnologie.**

DESCRIZIONE:

vedi sopra, Venerdì 16 Giugno 2000.

CALENDARIO sabato

Ore 14-14.30

PRESENTAZIONE

Riassunto di ciò che è emerso il giorno prima

Ore 14.30-16.00

**APPROFONDIMENTI STORICI (PARTE
SECONDA):**

- DAGLI ANNI'80 AD OGGI

Tommaso Tozzi: Le comunità virtuali italiane tra la metà degli anni ottanta ad oggi Francesco Galluzzi e Claudio

Parrini: Storia dell'arte in rete: storia di pratiche del soggetto – dalla nascita delle BBs alla New Economy
 Franco Berardi Bifo: intervento da definire

Ore 16.00-18.30

APPROFONDIMENTI E DISCUSSIONE
 (PARTE SECONDA):

- NET ART: AUTOMAZIONE,
 PLAGIARISMO E NETWORKING
 Introduzione di Snafu
 Mark Napier: Video intervista
 O1.org: Lettura di articoli di giornali
 FerrariStationWagon: Presentazione progetto PlayStationWagon + proiezione video + navigazione nel sito dell'operazione playstation
 Cornelia Solfrank: Net.art Generator e Old Boys Network
 RTMARK: video "Bringing to you"
 Massimo Contrasto: Il veleno dell'arte

Ore 18.30-20.00

DIBATTITO

. CATTEDRALE:

**10.00 –13.00: SEMINARIO SUL TCP/IP
 (a cura del LOA Hack Lab Milano)**

DESCRIZIONE:

Si tratta di due incontri, il primo a livello elementare ed il secondo un po' piu' avanzato, della durata di un paio di ore ciascuno. La prima parte e' un riassunto di argomenti che si trovano facilmente su libri e manuali (avendo anche un po' di voglia di leggere), mentre la parte successiva riprende gli stessi concetti esaminandone i punti deboli. Stiamo anche preparando dei CD con documentazione ed informazioni utili.

----- Scaletta
 ridottissima
 Seminario sul TCP/IP - LOA HackLab

Milano
 Primo Incontro:
 Introduzione, Hardware, Pacchetti, Indirizzamento, Protocolli e Servizi, Routing, "Standard" Microsoft
 Secondo Incontro:
 Porte e Servizi, Pacchetti, Connessioni

----- Scaletta
 ridotta
 Seminario sul TCP/IP - LOA HackLab
 Milano

Presentazione
 LOA HackLab Milano

Primo Incontro (Base)
 Introduzione, storia e diffusione Hardware
 Sistemi aperti e modello OSI
 Pacchetti ed incapsulamento
 Indirizzamento
 Risoluzione dei nomi
 Protocolli e servizi
 Interconnessione tra reti ed instradamento
 "Standard" Microsoft

Secondo Incontro (Intermedio)
 Porte e servizi
 Pacchetti
 Connessioni

**14.00 – 16.00: TELEMATICA DI BASE,
 PARTE II (a cura di William Maddler)**

DESCRIZIONE:

vedi sopra, Venerdì 16 Giugno 2000, Pub, ore 14.00 – 16.00-

**16.00 – 18.00: SEMINARIO SU
 REVERSE ENGINEERING (a cura di
 Letale Malattia, Little John e Tinman)**

DESCRIZIONE:

Windows Reversing:
 - Introduzione

Cracking/Reversing. Zen Cracking.
 Perche' esistiamo :)
 - Le basi di ASM
 Registri, Segmenti, Istruzioni, Routine, API
 - I tool
 Disassemblatori, Debugger, Hexeditor
 - Le protezioni/Le tecniche
 Serial, time limited, crippled, PE & co.
 - Esempio di reversing
 sorpresa =)

Linux Reversing:

- Introduzione
 (overview sul sistema operativo e differenze principali con Windows)
 - File Eseguibili
 (file binari e non, il formato ELF, approccio al reversing)
 - Tools necessari per il reversing:
 descrizione
 (debugging, disassembling, hex editing)
 - L'ambiente grafico X
 (differenze con Windows, layers)
 - Reversing in Linux
 (reverse engineering con e senza codice sorgente)

18.00 – 20.00: BLINDFOLD (a cura di Shine con la collaborazione di jaromil)

Si tratta di provare dei pacchetti IP, un po' "alla cieca", fino a trovarne alcuni che fanno crashare una macchina target...e quindi si puo' "rinforzare" lo stack tcp/ip di linux...

DESCRIZIONE:

Blindfold Project - Overview

L'obiettivo del Blindfold Project e' di trovare dei pacchetti IP (provando quasi tutti quelli possibili) che fanno andare in crash un computer "target" con Linux.
 Una volta trovati questi pacchetti, sara' possibile rinforzare lo stack TCP/IP di Linux.

Il sistema e' formato da tre parti: l'executor (cioe' l'host dove gira il programma che manda il pacchetti), il target (quello che riceve e pacchetti ed ogni tanti crasha :-)) ed il packet server (quello che da i pacchetti da provare e riceve quelli gia' provati o "interessanti").
 I pacchetti interessanti sono quelli che fanno crashare l'host target.

Mappatura dei pacchetti IP.

Un pacchetto IP, senza dati e senza opzioni, e' lungo 160 bit; escludendo i campi "destination address" e checksum che dovranno essere necessariamente validi, rimangono 112 bit.
 $2^{112} = 10^{33}$ (10^{34}). Un pacchetto IP e' lungo 160 bit (circa 10^2 bit), quindi ho 10^{34} (pacchetti) * 10^2 (bit) = 10^{36} bit.

Con un flusso di 10Mbit/s (quindi 10^7 bit/s) per provare tutti i pacchetti (10^{36} bit) mi servirebbero 10^{29} secondi.

In un anno di sono circa 10^7 secondi => $10^{29} / 10^7 = 10^{22}$ anni, sicuramente fuori dalla nostra portata....
 Bisogna quindi fare delle semplificazioni, sapendo che ogni semplificazione puo' tenere fuori dai risultati un certo numero di pacchetti "interessanti".

Vediamo quindi le semplificazioni nel nostro progetto:

Di ogni pacchetto IP ci saranno alcuni campi che non saranno utilizzati per il testing:

- destination address (deve essere valido)
- checksum (deve essere valido)

altri campi assumeranno pochi valori (limite):

- src address (0.0.0.0, "netid=0 subnetid=0 hostid", 127.0.0.1, dest address, 255.255.255.255, "netid subnetid=0 hostid=11.1", "netid subnetid

hostid=11...1", "netid subnetid=11...1
hostid=11...1") => servono 3
bit

(con dest address = netid - subnetid -
hostid)

- TTL (0, 1) => serve 1 bit

- fragment offset (0, 1, maxlen-1, maxlen)
=> servono 2 bit //
maxlen = 2^{13}

- identification (0, 1, maxlen-1, maxlen)
=> servono 2 bit //
maxlen = 2^{16}

- packet len (0...2047, 8192, 8192+1,
65535-1, 65535) => servono 11 + 2
= 13 bit

Per questi campi servono in totale 21 bit.
Gli altri campi assumeranno tutti i valori
possibili => 4 (ver) + 4 (header
len) + 8 (TOS) + 3 (flags) + 8 (protocol) =
27 bit.

In totale ci servono $21 + 27 = 48$ bit (x
pacchetto).

$(2^{48} * 160) / 10^7 / 10^7 = 10^3$ anni
(cioe' con 1000 macchine in parallelo
basta un anno per provarli tutti).

Se invece di "portarci dietro" tutto il
pacchetto IP, ci portiamo solo la
parte che cambia, abbiamo circa 48 bit
che possiamo mappare come se fossero
sequenziali.

L'importante e' che l'executor sappia
come rimappare il pacchetto "ridotto"
in quello completo.

Il Packet Server.

Il server e' un demone che passa ai
richiedenti, un "chunk" alla volta e
ritira i pacchetti controllati e interessanti.

Un chunk e' formato da 10^{10} pacchetti
(di 48 bit), cioe' quelli controllabili in circa

30 ore. (10^{12} bit / 10^7 bit/s = 10^5
secondi = 28 ore (circa))

Un chunk viene rappresentato come un
valore a 48 bit (lunghezza del
pacchetto) ed e' il valore del primo
pacchetto da controllare.

Per trovare il primo pacchetto da
controllare, si moltiplica il numero del
chunk per 10^{10} .

Volendo controllare 10^{15} pacchetti
(2^{48}), avremo 10^5 chunk di 10^{10}
pacchetti l'uno.

Rappresentando con una matrice di bit
(booleana) i chunk controllati e da
controllare, avremo una matrice di 10^5
bit, cioe' 100KByte.

Quindi possiamo anche scandirla
sequenzialmente senza fare alcuna
struttura dati di supporto (indice).

Ci servono N matrici (matrixC), una per
ogni versione del kernel da testare.
Inizialmente (e forse definitivamente)
supportera' i kernel 2.2 e 2.4
Ovviamente queste saranno serializzate
su disco (trattati come file binari).

Ci saranno dei metodi che lavorano su
queste strutture:

- turnon (num) // dice che un pacchetto e'
stato controllato (==1)
- turnoff(num) // dice che un pacchetto e'
da controllare (==0)

Ci sara' anche una matrice (per ogni ver
del kernel) (matrixR) per tenere
traccia dei chunk richiesti ed un metodo
per lavorarci:

- findfirst() // trova il primo pacchetto da
controllare (non richiesto) e
lo prenota (torna 0000...0 se prenotati
tutti o errore)

Quando tutti i pacchetti sono stati
prenotati, si rifa' la matrice matrixR
copiando matrixC.

Ci saranno poi N file (sempre uno per ogni ver del kernel) (matrixl) con i pacchetti "interessanti".

Ci sarà un metodo per lavorare su questi file:

- append(val, N)

Protocollo tra Executor e Packet Server.

I comandi che il Packet Server può ricevere sono formati da tre parti:

Azione (gime, take, done) - pacchetto - Ver (kernel)

L'azione è lunga 4 byte, il pacchetto è lungo 15 byte, e ver è un byte.

I 48 bit del pacchetto vengono codificati con i valori decimali ascii e quindi per rappresentare 2^{48} valori ci servono 15 cifre (byte).

Il valore del chunk è codificato con 5 char (arriva fino a 99999).

Per richiedere dei pacchetti da controllare (un chunk):

- Azione = gime; pacchetto = qualunque valore di 15 byte; Ver = quella del kernel dell'host target.

- Il packet Server richiamando la findfirst() trova il primo chunk da controllare (5 char) e lo passa al richiedente (o "erroR" in caso di problemi).

Per passare al Packet Server un pacchetto interessante:

- Azione = takeit; pacchetto = i 15 byte del pacchetto trovato; Ver = quella del kernel dell'host target.

- Il packet Server risponde inviandogli una stringa con "thankU" (o "erroR" in caso di problemi).

Per dire al packet server che ha finito di controllare dei pacchetti:

- Azione = done; pacchetto = i 5 byte del chunk controllato + 10 byte di padding; Ver = quella del kernel dell'host target.

- Il packet Server risponde inviandogli una stringa con "thankU" (o "erroR" in caso di problemi).

L'executor.

L'executor si appoggia a tre file: go.in, go.out e go.check.

go.in contiene il numero del chunk da controllare (stringa di 5 char).

go.out contiene i pacchetti interessanti (stringhe di 15 char).

go.check contiene l'ultimo blocco controllato (in caso di break del programma) (stringhe di 8 char).

Il chunk ha 10^{10} pacchetti. Questo si può dividere in 10^8 blocchi da 100 pacchetti l'uno.

L'executor prova i 100 pacchetti di ogni blocco e poi pinga l'host (target) per vedere se è crashato.

In caso positivo, continua a pingarlo (ogni 60 secondi) finché non torna su.

Inoltre salva il numero del blocco nel file go.check.

Quando l'host target è tornato su, si riprovano tutti i pacchetti di quel blocco pingando dopo ogni pacchetto finché non si trova quello che lo fa crashare.

Si appende quel pacchetto in un file (go.out) e si continua a provare gli altri pacchetti finché non finisce il blocco. Finito il blocco, si ricomincia col ping solo alla fine del blocco.

Quando sono finiti tutti i pacchetti, il programma esce, e con "nomeprogramma" --update (dopo essersi connessi ad internet) si flushano i buffer (go.out e go.in) e si riempiono quelli di input (go.in).

(si cancellano inoltre i file go.check e go.out).

I gestori per il SIGINT (ctrl-c o kill -9), il SIGTERM e il SIGQUIT salveranno in go.check il valore dell'ultimo blocco controllato e termineranno il programma.

**Durante tutta la giornata in Cattedrale:
RADIO CYBERNET**

**Di notte: SEMINARIO A CURA DI
RADIOCYBERNET TRASMESSO VIA
RADIO-IP: COME COSTRUIRE UN
HACKLAN (Asbesto)**

. SALA DA TE':

**14.00 – 16.00: SEMINARIO SULLA
PERSUASIONE (a cura di Enkidu)**

**16.00 – 18.00: SEMINARIO SULLE
SCELTE ETICHE E POLITICHE
NELL'HACKING: HACKER PIRATA O
GENTILUOMO? (a cura di The Warlus)**

DESCRIZIONE:

Discussione sulle scelte etiche e politiche nell'hacking:
presentazione

Per evitare equivoci va innanzitutto chiarito, al momento in cui si propone una discussione del genere, ambiziosa e per certi versi inconsueta nel contesto dell'hackmeeting, che chi la propone e` perfettamente consapevole che l'hacking e` un concetto e un tipo di

pratica che non si presta facilmente a definizioni univoche ne` tantomeno ideologiche, ma che si tratta al contrario di un contesto sociale culturale e psicologico estremamente fluido in cui individualita` e comunita` di vario tipo e natura, assai eterogenee tra loro anche se accomunate da un vivo interesse per l'utilizzo e l'appropriazione diretta delle tecnologie informatiche, sono venute via via collocando se stesse e la propria attivita`.

Le stesse modalita` di nascita e di autoorganizzazione dell'hackmeeting e le discussioni che ad esso si accompagnano periodicamente ne sono una riprova: il campo di interesse e di azione dell'"hacking" -- almeno nell'accezione che ne da` l'hackmeeting stesso -- e` quanto mai vasto: si va dall'attenzione per i problemi di sicurezza in rete e dall'attivita` di cracking dei siti ai "cyber-rights", alla controinformazione e alla difesa della liberta' di espressione in rete; dall'attivita` di produzione e diffusione di software libero, con tutta la riflessione sulle problematiche di autotutela giuridica e commerciale e di contraddizione od interazione con la logica del profitto che essa ha comportato, all'esplorazione degli aspetti piu' estremi ed inconsueti delle nuove tendenze culturali, comportamentali e sessuali poste in essere dalla "vita nel cyberspace"; dal reverse engineering del software e dalla messa in circolazione di programmi sprotetti alle nuove forme di creativita` artistica e culturale basate sulle nuove tecnologie; dall'interesse sempre vivissimo per tecnica e smanettamenti vari alla critica radicale del concetto di copyright.

Un grande calderone ribollente insomma, la cui storia il cui senso e le cui tendenze non sono riducibili a facili

appiattimenti, e che se non rappresenta certo quel "mostro" che governi e mass-media hanno cercato tante volte di additare all'opinione pubblica, non si è nemmeno mai caratterizzato per un eccessivo rispetto di leggi, norme, morali e convenzioni di vario tipo.

Potrebbe quindi sembrare un controsenso il tentativo di aprire una discussione sulle "scelte etiche e politiche nell'hacking", quando in esso convivono tanti sottoinsiemi e tante tendenze, perlopiù spesso scarsamente interessate se non insofferenti verso qualunque tentativo di normarle, anche dal punto di vista puramente sociologico o teorico.

Ma in realtà è l'evoluzione stessa della situazione che ci circonda a rendere necessario un momento di riflessione del genere: il ripetersi sempre più insistito e frequente di campagne di criminalizzazione da parte di governi polizie e mass media (che ha raggiunto una punta parossistica mai vista in precedenza in occasione dei cosiddetti "hacker days" durante gli attacchi ad Yahoo! e ad altri server commerciali di qualche mese fa) e il successivo stillicidio di episodi repressivi, di inchieste, di chiusure di server, di proposte di legge liberticide, di nuove campagne di stampa, non sono stati privi di effetti sulla comunità hacker. Tanto più che ad esse si è accompagnato un chiaro invito a trasbordare la figura dell'hacker sul versante dei "buoni" (dal punto di vista dello stato, ovviamente, e che questo invito è stato talvolta raccolto anche da figure che a torto o a ragione dell'hacking vengono considerate rappresentanti, perlomeno dai media.

Sono stati quindi posti sul tappeto -- e hanno dato luogo ad aspre discussioni -- una serie di problemi scottanti, da quello del rapporto

con la legge e con quei comportamenti illegali o quantomeno "malvisti" che del patrimonio hacker fanno tradizionalmente parte a quello del giudizio da dare su quegli hacker che prestano collaborazione ed assistenza a forze di polizia, agenzie di sicurezze, organizzazioni moralistiche.

Non crediamo sia possibile dare soluzioni univoche a tali problemi, né è obiettivo di questa discussione approdare ad una sorta di "guideline" etica o politica su di essi. Ma riteniamo sia comunque necessario iniziare a discuterne: il tentativo è quello di capire come e perché stanno mutando gli scenari politici e giudiziari rispetto ad Internet e all'hacking. Soprattutto è il tentativo di portare avanti un NOSTRO concetto di hacking che faccia giustizia dei luoghi comuni e delle falsità che mass media e governi danno in pasto all'opinione pubblica. Ma senza neanche ridurlo a quel concetto di "buon" hacker tagliato su misura per risultare ben accetto alle esigenze di governanti poliziotti e moralisti, che troppo spesso ci sembra sia stato portato avanti in questi ultimi tempi da chi si è espresso nei mass media su questo problema.

Il metodo seguito per la discussione sarà come sempre il più possibile orizzontale ed aperto; di conseguenza non sono previsti né scalette né interventi preordinati: chi partecipa all'incontro venga per portare un suo contributo ed un suo punto di vista, non per sentire tre o quattro "specialisti" o "personaggi famosi" discutere tra loro.

Verrà quindi preparata solo un'introduzione, che cercherà più che altro di fare il punto su una serie di fatti particolarmente emblematici, a partire dall'episodio degli hacker-days sino a fatti di cronaca recente. L'individuazione dei problemi etici e politici sollevati da questi fatti, e le conseguenze da trarre rispetto ad essi saranno affidati alla libera

discussione tra i presenti. Se saremo tutti d'accordo, la discussione potra' essere registrata -- o trasmessa via radio Cybernet -- in modo che gli interventi dei partecipanti possano essere pubblicati su web. Telecamere e macchine fotografiche non saranno ammesse durante l'incontro.

Eventuali interventi scritti pervenuti prima della discussione saranno bene accetti e saranno anch'essi pubblicati su web sul sito dell'hackmeeting. Come intervento preliminare, si rimanda a quello dell'hacklab Firenze sugli hacker days, disponibile all'URL <http://firenze.hacklab.it/DoS/Commento.txt>

Con questo intervento intendiamo aprire una discussione su come mutano per Internet gli scenari economici politici e giudiziari dopo il martellamento mediatico dell'immaginario collettivo che ha fatto seguito agli "Hacker's Days" (come sono stati ribattezzati dalla stampa USA gli attacchi che hanno bloccato per qualche ora Yahoo! e altri noti siti del commercio elettronico") e su come affrontare i problemi che questi mutamenti pongono a chi intenda mantenere quegli spazi di comunicazione libera e non omologata al "pensiero unico" che sinora e' stato possibile aprire dentro Internet.

La prima cosa che si impone anche all'osservatore superficiale e' quanto sia stata enorme la sproporzione tra quanto e' effettivamente successo -- chi bazzica la rete sa bene che il down di qualche ora di un server e' cosa abbastanza comune, e che la stragrande maggioranza degli utenti da se' non si sarebbe nemmeno accorta di quel che era successo -- il panico generale che si e' sollevato o che si e' preteso di sollevare, e le misure che si vorrebbero adesso far credere destinate a "ripristinare la sicurezza di tutti su Internet".

Nemmeno le conseguenze finanziarie dell'episodio giustificano il gigantesco spettacolo mediatico e gli strappamenti di

capelli di presidenti USA e governi messi in scena per l'occasione. Chi ha familiarita' con gli andamenti borsistici sa bene che i cali dell'indice Dow-Jones e Nasdaq che si sono verificati (seguiti del resto da pronti rialzi grazie alla vertiginosa impennata dei titoli delle aziende che si occupano di sicurezza) sono fisiologici, specie quando ci sono in ballo capitalizzazioni finanziarie gonfiatissime come quelle di cui sono stati oggetto i siti del commercio elettronico. E sa benissimo che in occasioni del genere, se c'e' qualcuno che ci perde c'e' anche qualcun altro che ci guadagna.

Ma si ingannerebbe anche chi volesse vedere in questa sproporzione solo la consueta isteria mediatica di giornalisti e mezzibusti televisivi a caccia di notizie strabilianti da vendere. Stavolta, ad aprire il gran ballo la cui solfa finale e' come sempre "piu' sicurezza in rete per difendere il commercio elettronico" (e vedremo cosa significa questo discorso) sono stati gli astri piu' luminosi nel firmamento del potere mondiale. Poche ore dopo l'accaduto, Janet Reno, ministro della giustizia USA, ha solennemente promesso al mondo la cattura dei "cybercriminali del terzo millennio" (suona bene, nevero?) Clinton ha esternato le sue stupefatte preoccupazioni. Parlamenti e governi di tutto il mondo si sono accodati ai leader del Nuovo Ordine Mondiale nelle sortazioni a identificare e scongiurare questa nuova minaccia all'umanita'.

A questo proposito, si potrebbe anche osservare che forse forse, agli albori del terzo millennio, ci sarebbero per l'umanita' dei pericoli un tantino piu' gravi di cui preoccuparsi: le guerre di sterminio, l'inquinamento ambientale e lo spreco delle risorse, la miseria, lo sfruttamento e la morte per fame e malattia di centinaia di milioni di uomini donne e bambini... Problemi che magari non sono cosi' cyberaffascinanti,

ma da cui l'umanita' -- ivi compresi quei due miliardi di persone che in vita loro non hanno mai fatto una telefonata -- e' magari un tantinello piu' afflitta che non dal simpaticone di turno che viola un sito Web per falsificare le parole di Clinton e mettergli in bocca che in rete ci vuole piu' pornografia, venendo promosso per qualche giorno al rango di spauracchio mondiale. Ma si sa, noi siamo degli irriducibili renitenti al pensiero unico, e siamo cosi' pericolosamente estremisti da credere che i sacri principi del business e della logica del profitto non bastino a giustificare la barbarie e gli orrori del "Nuovo Ordine Mondiale"...

Comunque, retoriche millenariste a parte, le ipotesi sugli autori di questo "vilissimo attentato" si sono sprecate. Si va dalle esilaranti dichiarazioni di Eric Holder (viceministro della giustizia USA) che ha dato la colpa ai genitori americani, che trascurano di sorvegliare i loro vispi pargoletti quando stanno al computer, alle serie ipotesi dell'FBI sul "nuovo terrorismo elettronico" nato sulla scia del complotto in rete ordito da cospiratori internazionali per fracassare le vetrine di Seattle. Su quest'ultimo punto in particolare si e' accanita la stampa, insistendo nel confondere (spesso con fraintendimenti tecnici tanto assurdi quanto divertenti, specie se messi in bocca all'hacker di turno frettolosamente scovato a scopo di intervista) consolidate e tradizionali pratiche di disobbedienza civile in rete quali il "netstrike" o l'invio massiccio di e-mail di protesta con un attacco tipo "distributed denial of service" come quello attuato contro Yahoo! e co.

A questo proposito, dobbiamo essere decisi nel denunciare che dietro "fraintendimenti" del genere puo' nascondersi il tentativo autoritario di equiparare alcune tecniche di disobbedienza civile -- tutto sommato

legalitarie -- a comportamenti che costituiscono reato penale. Ma dobbiamo anche essere altrettanto decisi nell'affermare che noi non condanniamo ne' ci dissociamo a priori da chi decide di "commettere reato" per motivi che ci sembrano politicamente ed eticamente condivisibili <<Anche se non tutti condividiamo la logica dell'azione individuale e del sabotaggio "mordi e fuggi">>. Sappiamo troppo bene quali abusi, orrori ed ignominie possono essere coperti dal manto della legge dello stato perche' il suo rispetto ci sembri l'unica o piu' importante considerazione da fare in circostanze del genere. Se questo suona sgradito a chi ha tentato affannosamente di ricoprire l'hacker col vestito della rispettabilita' a tutti i costi -- fingendo di scordare o ignorando quanti hacker con la legge hanno avuto problemi in nome della libera circolazione dei saperi e dell'informazione -- molto semplicemente non sappiamo che farci.

Ma nemmeno ci interessa giocare agli investigatori da romanzo giallo o rivendicare un movente politico di nostro gusto per un'azione che resta invece aperta a tutte le interpretazioni -- vista la totale assenza di ogni dichiarazione o rivendicazione -- e sulla quale ogni ipotesi e' legittima. Una volta sottolineato che la scelta dei siti da attaccare non sembra casuale (si trattava dei siti maggiormente rappresentativi del commercio elettronico e che avevano visto maggiormente crescere le proprie quotazioni in borsa) e che chi ha compiuto l'attacco ha avuto se non altro la competenza necessaria a non farsi rintracciare, ci interessa innanzitutto prendere in esame quale realta' stia prendendo forma dietro le roboanti dichiarazioni sulla necessita' di ripristinare "la sicurezza in rete" da cui siamo stati subissati nei giorni successivi.

E una prima realta' da prendere in considerazione, ignorata da quasi

tutti, e' questa: LE BANCHE USA
SAPEVANO IN ANTICIPO
DELL'ATTACCO.

Secondo una nota dell'Associated Press del 14 Febbraio, il "Financial Services Information Sharing and Analysis Center" (FSISAC) aveva diramato diversi allarmi urgenti agli esperti di sicurezza di varie banche USA almeno quattro giorni prima degli inizi degli attacchi, indicando anche alcuni indirizzi Internet di macchine compromesse da cui l'attacco sarebbe arrivato. Ma la notizia non e' stata trasmessa dalle banche all'FBI o ad altre agenzie di polizia USA.

Va sottolineato che il FSISAC, secondo l'Associated Press, e' un centro informazioni riservato, al punto che la sua stessa locazione fisica e l'elenco delle banche che usufruiscono dei suoi servizi - - che costano sino a 125.000 dollari -- sono tenuti segreti. Questo centro e' stato recentemente sviluppato e potenziato dietro diretto ordine presidenziale e in questa occasione si e' dimostrato molto efficace.

Le banche hanno potuto tenere segreta la notizia degli imminenti attacchi grazie a una precisa disposizione voluta dal Dipartimento del Tesoro USA, per cui non sono tenute a condividere con gli organismi di polizia statali e federali eventuali notizie e informazioni su reati ottenute attraverso questo di sorveglianza. I vertici USA hanno ritenuto che qualunque obbligo in questo senso avrebbe disincentivato le banche ed altri soggetti strategici per l'economia statunitense dall'attrezzarsi per una efficace vigilanza contro "impiegati disonesti, bug nel software, virus ed hacker".

Senza voler commentare se questa notizia possa dare particolare credibilita' alle ipotesi di chi ha visto negli attacchi ad Yahoo! etc.
una manovra volta a destabilizzare la borsa a fini speculativi, ci pare

evidente come essa getti una luce un tantinello diversa sulla stupefazione di Clinton all'indomani degli attacchi. In particolare ci porta a domandare quale senso abbia, ai fini della sicurezza reale, investire milioni di dollari in strutture come il NIPC (National Infrastructure Protection Center, un organismo gestito da FBI ed altre agenzie per la sicurezza delle reti USA contro i reati informatici) se poi a queste strutture non vengono trasmesse le informazioni che erano a disposizione dei piu' sofisticati centri di sorveglianza USA.

Una seconda realta' da prendere in considerazione e' questa: INTERNET, A DIFFERENZA DELLE RETI BLINDATE PER LA GESTIONE DELLE INFORMAZIONI STRATEGICHE DI NATURA ECONOMICA POLITICA E MILITARE, E' ALTAMENTE INSICURA, e gli alfieri della sicurezza in rete attraverso la repressione lo sanno benissimo, come sanno benissimo che le misure proposte non servono in realta' a modificare questa situazione.

Secondo un rapporto FBI, nell'anno trascorso, il 62% (sissignori, il sessantadue per cento) delle societa' americane ha subito violazioni degli impianti informatici senza che nessuno gridasse alla minaccia epocale del terzo millennio. Del resto, i programmi usati per l'attacco ad Yahoo! e compagni si basano sugli stessi principi che milioni di ragazzini usano in tutto il mondo per fare ai loro coetanei il dispetto di mettergli KO la macchina quando litigano in Internet Relay Chat.

L'esistenza di questa possibilita' di attacco e dei software utilizzati e' nota da anni agli esperti di sicurezza, e ponderose note tecniche su come prevenirle sono da anni disponibili su Internet. Quest'attacco ha quindi svelato a tutti che il re e' nudo, ma come nella favola tutti quanti in realta' lo sapevano

benissimo da tempo. Allora, come mai questa fragilita'?

La risposta e' semplice: in primo luogo perche' e' fisiologicamente impossibile garantire sicurezza totale a una rete globale di comunicazione aperta che -- sebbene tutt'altro che anarchica -- e' cresciuta in modo scarsamente pianificato e gerarchico, e che risente ancora dell'impostazione originaria di rete militare, che doveva continuare a funzionare anche se gran parte di essa fosse stata distrutta da un attacco nucleare.

Ma anche e soprattutto per pure e semplici ragioni di profitto. Una cosa che gli isterici commentatori sugli attacchi si sono ben guardati dal far rilevare e' che, se i router (le macchine usate per far uscire le reti locali su Internet) fossero stati ben configurati secondo le norme di sicurezza, l'attacco sarebbe risultato inefficace. Ma una volta comprato un router e installato, le successive modifiche di configurazione devono essere fatte da un esperto. E chiamare un esperto esterno o far addestrare un dipendente costa troppo caro.

Esistono poi in rete numerose macchine fisse che utilizzano tecnologie software come il Wingate (per sistemi Microsoft) che risultano penetrabili con estrema facilita' e da cui e' possibile far partire attacchi come quelli contro Yahoo! etc. senza lasciare alcuna traccia. Macchine da cui vengono pero' ricavati fior di profitti con il minimo di spesa.

La logica del commercio elettronico e' una logica che taglia il piu' possibile sui costi fissi di macchine, spazi e personale per puntare tutto sul profitto da pubblicita' e sulla capitalizzazione finanziaria gonfiata. La "febbre dell'oro cibernetico", il miracolo della "nuova economia", mostrano gia' i primi buchi e sotto i lustrini spunta la faccia dell'economia basata sulla logica di sempre: profitto

attraverso lo sfruttamento pesante del lavoro, e risparmio sulla sicurezza. Cosa succedera' quando il nuovo soggetto collettivo, i lavoratori dell'industria rete, scemati gli entusiasmi e l'infatuazione giovanile per il monitor -- intelligentemente sfruttati da chi incentiva i miti dei tredicenni divenuti miliardari grazie a Internet e poi paga un tozzo di pane chi lavora per lui -- prenderanno coscienza di questa situazione e agiranno di conseguenza?

Ma questo e' un altro discorso, per ora limitiamoci a registrare come anche da questo punto di vista la "sicurezza in rete" richiederebbe interventi ben diversi da quelli che Clinton e compagni si apprestano a farci trangugiare. L'inasprimento delle pene detentive contro i reati telematici (misura che tutti i governi stanno prendendo in esame, e che avra' per unico effetto concreto l'affibbiamento di qualche anno di galera in piu' a qualcuno abbastanza ingenuo da farsi pescare con le mani nel sacco e da diventare "monito per tutti") e l'investimento di milioni di dollari in centri di sorveglianza giganteschi quanto inutili, sono misure di PURA FACCIATA. Non servono a migliorare la sicurezza ma a mantenere la fiducia di massa nell'Internet "nuova frontiera del profitto", per assicurarsi che i giganteschi flussi finanziari messi in moto da decine di milioni di consumatori e di investitori scarsamente consapevoli di questa realta' non vengano meno.

Terza realta': POCHI GIORNI PRIMA DEGLI ATTACCHI, E PRECISAMENTE IL 4 FEBBRAIO, L'IETF (Internet Engineering Task Force, l'organismo che lavora sugli standard e sui protocolli su cui si basa il funzionamento di Internet) AVEVA SECCAMENTE RIFIUTATO DI PRENDERE IN CONSIDERAZIONE LA PROPOSTA DELL'AMMINISTRAZIONE CLINTON DI APPORTARE MODIFICHE AI NUOVI PROTOCOLLI IP CHE FACILITASSERO IL LAVORO DI

INTERCETTAZIONE DELLE AGENZIE DI SORVEGLIANZA. Sebbene tale rifiuto in realta' ostacoli in modo minimo dal punto di vista tecnico l'intercettazione, dal punto di vista politico ha costituito una secca sconfitta, che e' andata ad aggiungersi a quelle gia' collezionate dai governi su questo terreno, e che rimarca ancora una volta come il "popolo della rete" non abbia poi tanta fiducia nella paterna sorveglianza dello Stato. E' anche legittimo dedurre che evidentemente per il team IETF (e del resto anche per molte aziende) sorveglianza ed intercettazione non servono a garantire una briciola di sicurezza in piu' agli utenti Internet.

Malgrado questo, nei giorni immediatamente successivi all'attacco, abbiamo sentito Clinton riproporre con forza un ennesimo robusto finanziamento delle agenzie di sorveglianza USA (da 15 a 240 milioni di dollari destinati ad FBI ed NSA, a seconda di quanti gliene lasceranno spendere i suoi avversari repubblicani) come illusoria panacea contro questi attacchi. Abbiamo sentito parlare di nuove futuristiche cyberpolizie, mentre sappiamo che quelle esistenti sono gia' state in grado di prevedere gli attacchi, e che solo per scelta politica ed economica questi allarmi non sono stati diffusi oltre il circuito bancario. Abbiamo visto l'FBI offrire agli ingenui un proprio software per difendersi dagli hacker (ma chi ci difendera' dall'FBI?). Abbiamo visto persino la CIA -- questo campione dell'insicurezza di massa e del terrorismo su scala internazionale -- ergersi a baluardo della sicurezza in rete.

Intanto la Gran Bretagna (non a caso uno dei paesi beneficiari delle informazioni raccolte dal sistema d'ascolto clandestino Echelon) minaccia due anni di galera a chi si rifiutera' di rendere accessibili allo Stato le chiavi private del proprio sistema crittografico che impedisce a terzi indesiderati di leggere le proprie e-mail.

In Cina gli utilizzatori di software di crittografia vengono schedati. In Russia sono in atto nuovi tentativi di monitorare e limitare gli accessi alla rete.

In Germania abbiamo visto prima criminalizzare poi "collaborare con la polizia" (dizione ambigua che in questi casi puo' significare molte cose, da un'entusiastico sostegno al trovarsi in stato di semiarresto) l'autore di un software che consente di effettuare attacchi come quelli contro Yahoo! messo a punto per ragioni di analisi della sicurezza. Ancora una volta si cerca di sostenere il concetto pericolosissimo e nefasto della "security by obscurity", per cui chi svela al grande pubblico le debolezze di un sistema che si vorrebbero tenere nascoste dovrebbe essere considerato automaticamente responsabile degli attacchi portati contro questo sistema.

Proprio in questi ultimi giorni, Eric Holder (sempre lui, il geniale viceministro americano) ha informato il Congresso USA che, come contromisura contro altri possibili attacchi di questo tipo, il governo sta apprestandosi ad inasprire ulteriormente le pene contro gli "hacker", a responsabilizzare penalmente i genitori contro le marachelle al PC commesse dai loro frugoletti, a instaurare misure di monitoraggio sugli accessi e i comportamenti su Internet, e dulcis in fondo, naturalmente, a criminalizzare gli anonymous remailer, quei servizi di anonimizzazione della posta elettronica che rappresentano l'unico modo a disposizione degli utenti con scarse conoscenze tecniche per garantirsi un anonimato reale su Internet e per non far rintracciare a chiunque sia intenzionato a farlo l'elenco dei propri corrispondenti.

Questi servizi, normalmente e storicamente disponibili su Internet, dopo il fallimento dei tentativi passati di additarli come un covone di pedofili e

terroristi, vengono ora presentati come una minacciosa "risorsa degli hacker", come se chiunque fosse dotato di un po' di competenze tecniche non avesse in realta' a disposizione sistemi ben piu' efficaci per non farsi rintracciare, e come se non fosse assolutamente chiaro che con gli attacchi portati contro Yahoo! etc. non hanno nulla a che vedere.

Infine, nel nostro paese, Rodota' in persona ha pensato bene di informarci su "Repubblica" che e' l'ora di finirla con la "CyberAnrchia", che rifiutare le regole e la paterna sorveglianza dello stato e' infantilismo, e che un atteggiamento del genere e' infantile, perche' non solo mette a rischio la liberta' di tutti ma finisce per dare spazio alla "deregulation" che favorisce i poteri forti economicamente. Del resto tutti sanno che la liberta'puo' esistere solo attraverso le regole, e che queste regole non possono essere liberamente autoscelte ma devono essere necessariamente imposte dallo stato per il bene di tutti. Particolarmente per il bene dei poveri e degli oppressi, per i quali lo stato rappresenta l'unica difesa contro i padroni e gli oppressori, particolarmente se lo stato in questione e' a guida PDS...

A questo rispondiamo: NOSSIGNORI!

DEVE ESSERE CHIARO A TUTTI: NON E' LA SICUREZZA, E TANTOMENO LA DIFESA DEGLI INTERESSI DEGLI UTENTI CONTRO I POTERI FORTI CHE SI VUOLE INSEGUIRE CON MISURE DEL GENERE.

Qui si vuole sfruttare il panico per colpire quelle possibilità di sfuggire all'occhio onnipresente del controllo statale, che Internet ha dato a milioni di persone che hanno preso in contropiede stati e governi. Si profila un nuovo minaccioso attacco agli spazi di liberta' individuale in rete attraverso misure che in realta' con la prevenzione di questi attacchi non hanno

NULLA a che fare, e nuovamente siamo chiamati a rispondere.

Ma va preso atto che rispetto a tutto questo, proprio la risposta della "comunita' hacker" storica, dei "cyberpunk", di tutto il variopinto insieme degli utilizzatori e dei navigatori "alternativi" di Internet e' stata sinora debole ed ambigua. La reazione piu' comune e' stata: "un attacco del genere e' roba da "lamer" (termine dispregiativo per indicare chi danneggia siti o pirateggia software avvalendosi di strumenti tecnici messi a disposizione da altri senza avere la comprensione di quello che sta facendo e la padronanza tecnica che caratterizzerebbero il "vero hacker") noi che c'entriamo?. Una reazione del genere -- considerando il fatto che in realta' non si sa tuttora NIENTE sugli autori e sui moventi di chi ha compiuto l'attacco, e che non e' certo l'uso di un software piuttosto che di un altro a poter dare qualche indicazione sull'effettivo livello di "lameraggine" degli autori -- ci sembra piu' che altro estremamente ipocrita ed intenzionata ad eludere i veri nodi del problema per tenere bassa la testa in un momento di crisi.

Ancor piu' esplicite in questo senso ci sembrano le reazioni di chi sta tentando di convincere i media che l'hacker e' fondamentalmente un boy-scout che non danneggerebbe mai qualcosa, che si interessa solo di tecnica e che non domanda di meglio, una volta messa la testa a partito, che farsi assumere da qualche software house o da qualche agenzia di sorveglianza o polizia. Azzerando completamente il fatto che se davvero si intende agire in conformita' al principio base de "l'informazione vuole essere libera", allora spesso e volentieri ci si trova ad infrangere qualche legge dello stato (e di cio' ci sono numerosi esempi storici) e che lavorare per polizie,

agenzie di sorveglianza eccetera e' in totale contraddizione con questo principio.

In questo varco di confusione, di incertezza, e di equivoco spesso voluto, si sta inserendo l'avversario, che mentre criminalizza i comportamenti insubordinati cerca di enfatizzare a scopo di recupero politico ed economico (il talento dell'hacker e' una risorsa preziosa sotto entrambi gli aspetti) l'esistenza degli hacker "buoni". Buoni dal suo punto di vista, ovviamente: quelli che combattono la pedofilia in rete (magari al servizio di parroci isterici ed ossessionati come Don Fortunato di Noto), quelli che sorvegliano i terroristi, quelli che danno assistenza alle agenzie di polizia e se hanno commesso qualche "crimine" lo hanno fatto solo per leggerezza giovanile. Cosi' assistiamo al fatto che Kevin Mitnick, "cult-hero" della cultura underground su Internet, pluriprocessato e nemico pubblico numero uno USA per tanti anni, si trasforma di colpo in una figura "buona" per i media quando si appresta a fornire collaborazione al governo USA per indagare su questi attacchi. Vediamo la CIA fare appello agli hacker perche' ha bisogno di nuovi talenti e vediamo decine di migliaia di giovani rispondere all'appello...

Questo non deve stupire piu' di tanto. La cultura hacker e cyberpunk non e' sinora riuscita a sciogliere un nodo di fondo: l'hacking e' puro possesso di capacita' tecniche o e' anche scelta e responsabilita' personale etica e politica? e quale scelta?

Ma e' proprio su questo interrogativo che da oggi si gioca la partita.

Sinora e' stato possibile alimentare l'illusione -- per chi voleva nascondere a se stesso e agli altri questo problema -- che la diffusione di Internet, della "cultura di rete" senza altre specificazioni, delle capacita' tecnologiche "neutre", significasse automaticamente diffondere una nuova forma di liberta'. Questa

illusione -- come quella dell'anarchia intrinseca" di Internet -- e' stata consentita dal convergere di due fattori: il carattere di per se' scarsamente gerarchico e difficilmente controllabile della rete, che ha permesso l'insorgere in essa di comportamenti e spazi insubordinati, e l'interesse del capitale finanziario e commerciale a massificare Internet, liberalizzando gli accessi e minimizzando i controlli in nome dello sviluppo del commercio elettronico.

Questa fase e' vicina a chiudersi. Ha permesso lo sviluppo di una contraddizione notevole, perche' Internet si e' trovata ad essere insieme "nuova frontiera" per la logica del profitto e mezzo di comunicazione di massa effettivamente "man to many", aperto a tutti e scarsamente governabile, dove individui e piccoli gruppi possono tuttora concorrere con successo sul piano comunicativo con governi e grandi gruppi di potere politico-economico in misura inimmaginabile per gli altri media, ma ora la situazione e' mutata. L'ulteriore espansione del commercio elettronico e dei profitti non dipendera' piu' dalla liberalizzazione degli accessi e dalla scarsa sorveglianza, ma dipendera' dal recupero della capacita' di GOVERNARE la rete, o quanto meno dell'illusione di essere in grado di farlo.

Rispondere a questo necessita di una nuova maturita'. Non e' piu' possibile oggi fingere che il trastullarsi con i nuovi giocattoli tecnologici e cibernetici conduca da solo a risultati diversi dall'inserirsi in un trend commerciale. Non e' piu' possibile esorcizzare chi sottolinea quest'ambiguita' di fondo e la necessita' di rompere con la logica del profitto e del potere, con l'accusa sprezzante di essere un barboglio "veteroqualchecosista". Non e' piu' possibile illudersi ed illudere che lo sbarco in forze dell'industria

culturale e dello spettacolo su Internet (o lo sbarco di essa in questa industria, ricordiamo che e' stata America On Line a comprare la Warner Bros e non viceversa, alla faccia di chi ci raccontava che la comunicazione su Internet era di per se' irriducibilmente antagonista agli altri media) rappresentino chissà quale progresso ed elemento di innovazione se non di liberazione. E viceversa non e' nemmeno piu' possibile fingere che la smania di protagonismo e presenzialismo mediatico di tanti hacker, veri o presunti, alluda a qualcosa di sostanzialmente diverso dallo svendersi a questa industria e dall'accettare le regole che la macchina mediatica globale per la produzione del consenso al pensiero unico ti impone per darti visibilita' positiva.

Soprattutto, non e' piu' possibile fingere che la passione e la capacita' tecnologica, da sole o accompagnate tutt'al piu' da qualche orpello di irriverenza, da qualche comportamento e linguaggio "strani" che fanno tendenza, ma destituite di ogni altro senso e movente, possano condurre alla fin fine a risultati diversi dal buon posticino alla software-house o all'arruolamento nella CIA.

Per rispondere a questo scenario mutato occorre riflettere a fondo su cosa sia "hacking", senza eludere i problemi, senza accontentarsi di pararsi le chiappe da chi vorrebbe ridurlo a criminalita', senza preoccupazioni di notorieta' mediatica e di risultare graditi al giornalista o allo sbirro di turno. Occorre maturare e portare avanti un NOSTRO concetto di hacking, senza lasciare che siano altri ad imporcelo. Occorre anche chiarire -- e sino in fondo -- che chi fa danni immotivati, chi si diverte a danneggiare altri, chi impiega tempo risorse e saperi solo per cavarci il gusto di rompere le palle a un pacifico user per affermare la propria presunta superiorita',

o per danneggiare un sistema al solo scopo di esibirsi e godere di un'effimera notorieta', non per questo e' un hacker. Come non lo e' chi cracca e pirateggia solo per cavarci dei soldi. Ma non lo e' nemmeno chi supinamente accetta le leggi e le norme che vorrebbero imporci in rete governi e multinazionali, chi fa propria la logica del profitto innanzitutto, scegliendosi un lavoro legale invece del cracking, chi finisce per arruolarsi nella polizia o a fare il mercenario delle multinazionali. E non e' un caso che molti passino direttamente dal cieco vandalismo o all'esibizionismo alla richiesta di arruolamento nella CIA.

Noi non crediamo di stare dicendo con questo nulla di diverso come hacklab di Firenze da quanto abbiamo detto sino dalla prima nostra uscita pubblica. Ne' nulla di diverso da quanto siamo andati dicendo a titolo personale o in altre situazioni ancora prima di dar vita all'hacklab. Perche' queste posizioni e questi ragionamenti non sono frutto unicamente delle nostre intelligenze personali, sono frutto dell'intelligenza collettiva di rete. E condividiamo quanto questa intelligenza collettiva ha proposto in merito, da anni e anni.

Noi quindi riaffermiamo che:

1) Intendiamo studiare e praticare il modo per garantirci in ogni caso la possibilita' di comunicare e soddisfare i nostri bisogni e desideri anche attraverso le nuove tecnologie. Rivendichiamo innanzitutto il diritto di diffondere e acquisire liberamente le informazioni. In modo UNILATERALE se necessario, indipendentemente da leggi e normative. Ci assumiamo questa responsabilita' perche' riconosciamo a noi stessi la capacita' di autogestire scelte comportamenti e responsabilita' etiche e politiche in modo autonomo da quanto vorrebbero imporci stati, governi e maggioranze.

2) Intendiamo combattere la battaglia per mantenere Internet LIBERA il piu' possibile, che per noi significa anche il meno possibile sottomessa alle esigenze di stati, governi, gruppi di potere economico ed agenzie di controllo e di spionaggio. Intendiamo combattere questa battaglia anche sul piano legislativo e normativo, con intelligenza politica e scioltezza tattica, senza rifiutare a priori l'apporto di forze istituzionali, ma consapevoli innanzitutto che cio' che governi e capitali chiamano "insicurezza" per noi e' innanzitutto "liberta'" e mettendo sempre al primo posto questo fatto.

3) Intendiamo studiare e praticare liberamente la possibilita' di nuove forme di uso della tecnologia che abbiano in prospettiva lo stesso senso di liberta' e orizzontalita' della comunicazione e dell'informazione che ha avuto Internet nei suoi anni eroici, e collaborare con chi a questo sta gia' lavorando, che a nostro giudizio merita l'appellativo di "hacker" nella pienezza del suo senso originario piu' di chiunque altro.

Sulla base di quanto detto finora riteniamo infine importantissimo aprire una discussione il piu' possibile allargata in rete sugli aspetti e le responsabilita' etiche e politiche nell'hacking.

Proponiamo quindi che su quest'argomento venga aperto anche uno spazio fisico di discussione continuativa, aperta, orizzontale e non formalizzata da comportamenti politichesi quali ordini del giorno, mozioni finali eccetera -- che non ci appartengono perche' oscurano la libera espressione e l'assunzione di responsabilita' personale.

Proponiamo che questo spazio resti aperto per tutta la durata nell'hackmeeting 2000 e che i contributi di chiunque interverra' in esso siano ridiffusi in rete e che possano essere liberamente

riprodotti e distribuiti senza scopo di lucro.

Intendiamo naturalmente dare il necessario apporto, logistico e di dibattito, a tale spazio.

Hacklab Firenze - 6 Marzo 2000
<http://firenze.linux.it/~leandro/hacklab>
hacklab@firenze.linux.it

**18.00 – 20.00: SEMINARIO SUL
 TECNOCONTROLLO (a cura di Tactical
 Media Crew) E SULLA CENSURA IN RETE
 (a cura di Osservatorio Diritti) +
 INTERVENTO DELEGAZIONE DI NODO50**

- **Proposta** di Ferry Byte di invitare una **delegazione di nodo50** per relazionarci sulla loro vicenda (un sito web contenente info su denunce di torture e maltrattamenti effettuati da pubblici ufficiali in Spagna suscita l'interesse del nostro equivalente Garante della privacy che intima al provider nodo50 di togliere il materiale in rete e minaccia processi sanzionatori a riguardo: vedi http://www.nodo50.org/censura_act/). Il tutto all'interno di un dibattito contro la censura in rete (Osservatorio Diritti);

DESCRIZIONE:

credo che la vicenda nodo50 vs. Agencia de Protección de Datos sia molto importante e significativa di come le istituzioni si rapportano alla difesa dei diritti, umani e cyber.

premettendo che bisognerebbe saperne di piu' nei minimi particolari da nodo50 quello che comunque (anche e malgrado le difficolta' di lingua diversa) traspare dai comunicati e dalle info che sono in linea e' abbastanza preoccupante (x aggiornamenti vedi anche:

http://www.nodo50.org/censura_act/)

un sito web contenente info su denunce di torture e maltrattamenti effettuati da pubblici ufficiali in spagna suscita l'interesse del nostro equivalente Garante della privacy che

intima al provider nodo50 di togliere il materiale in rete e minaccia processi sanzionatori a riguardo

ovviamente tutto cio' scatena la reazione delle comunita' in rete che si concretizza nei seguenti mirrors della pagina web originale:

<http://www.nodo50.org/actortura>) e relativo pacchetto per realizzazione <ftp://ftp2.nodo50.org/pub/mirrors/>

<http://squat.net/mirrors/actortura>
<http://www.ecn.org/actortura>
<http://www.ecn.org/inr/mirrors/actortura.htm>
<http://mirrors.web.net/www.nodo50.org/actortura/>
<http://www.planet.org.nz/mirror/www.nodo50.org/actortura/>
<http://www.archivos.org/act/actortura/>
<http://www.netrights.org/actortura/>
<http://www2.inform.dk/actortura>
<http://www.cobra.org/actortura>
<http://www.xarxaneta.org/mirrors/actortura>
<http://igc.mirrors.apc.org/nodo50/actortura/>
<http://apcmirrors.ecn.cz/actortura/>
<http://www.cyberdark.net/actortura/>
<http://www.ecuanex.net.ec/mirror/actortura>

ma tutto questo non puo' bastare

non possiamo far passare senza colpo ferire il principio per cui la tortura (attuata normalmente da tutte le polizie europee come anche riconosciuto da amensty international) sia una pratica comune che non puo' neanche essere denunciata in virtu' di un ipotetico diritto dei torturatori (o magari dei torturati) a difendere la propria privacy!?!

non possiamo far passare in maniera innocua il concetto per cui le preoccupazioni delle istituzioni non siano quelle di guardare con obiettivita' al proprio interno ma sia quello di reprimere le forme di dissenso al proprio esterno

abbiamo il dovere di esprimere una solidarieta' internazionalista verso le nostre sorelle e fratelli di nodo50 in difficolta' anche perche' gli strumenti telematici ce lo permettono con una certa

facilita' per cui propongo di:

1) invitare una delegazione di nodo50 (pagandogli il viaggio) al prossimo hackmeeting per relazionarci sulla loro vicenda e all'interno di un dibattito contro la censura in rete

2) attuare con i tempi e le modalita' da discutere in rete un netstrike contro <http://www.ag-protecciondatos.es/>

. STANZA VICINO AL CINEMA:

Durante tutta la giornata: CANDIDA, LA TV ELETTRODOMESTICA

- In occasione della terza edizione dell'hackmeeting prevista per il giugno 2000 a Forte Prenestino, noi di "Candida. La prima televisione elettrodomestica." abbiamo intenzione, insieme ad alcune persone di avana.net, di occuparci dello streaming video in rete di momenti ed eventi particolari della tre giorni. Si trattera' di fare microdifferite o dirette del materiale che verra' ripreso dalle nostre telecamere nei giorni stessi.

DESCRIZIONE:

Vedi Venerdì 16 giugno 2000, Stanza vicino al Cinema.

. PUB ALL'APERTO:

dalle 15.00: PRESENTAZIONE DI CARLO FORMENTI DEL LIBRO "INCANTATI DALLA RETE", ECONOMIA E ANTROPOLOGIA DELLA RETE (Carlo Formenti + Isole nella Rete)

DESCRIZIONE:

INCANTATI DALLA RETE. IMMAGINARI,
UTOPIE E CONFLITTI NELL'EPOCA
DEL WEB

Prefazione

PARTE PRIMA.

L'IMMAGINARIO DELLA GRANDE
MUTAZIONE

CAPITOLO I.

SCHERMI: LE ICONE DEL SUBLIME
TECNOLOGICO

1. Un regime del voyeurismo universale?
2. Schermo delle mie brame
3. Tuffarsi nello specchio
4. Verso una realtà più leggera
5. Il sublime tecnologico

CAPITOLO II.

CYBERTEOLOGIA: GLI ADORATORI
DEL FUTURO

1. Teilhard de Chardin: il Dio
dell'Evoluzione
2. Pierre Lévy: l'Angelo dell'intelligenza
collettiva
3. Philip Dick: il mito di Valis
4. Tecnognostici: guerra all'entropia
5. Tecnopagani: lo spettro nella
macchina

CAPITOLO III. CYBERANTROPOLOGIA:
METAFISICA DEL CYBORG

1. Il fantasma di Cartesio
2. Fottere (con) gli Dei
3. La narcosi di Narciso
4. omuncoli elettronici

PARTE SECONDA.

ECONOMIA SOCIETA' RETI

CAPITOLO IV. I CHIERICI: EROI O
VITTIME DELLA RIVOLUZIONE
DIGITALE?

1. I sogni degli ingegneri-poeti
2. Intellettuali al bivio: opporsi o aderire?
3. L'ipertesto è un rizoma. O no?

CAPITOLO V. QUEGLI STRANI
ALLEATI: POPULISMO HACKER E
ANARCOCAPITALISMO

1. Computer liberation
2. L'individuo sovrano
3. E-money: neanche un soldo allo Stato
4. Un Menenio Agrippa postmoderno

5. Anomalie: neoluddisti e cyberpunk

CAPITOLO VI. COMUNITA' E

CONFLITTO: MARX OLTRE FORD?

1. Postfordismo e cyberfeudalesimo
2. Metafore territoriali
3. Nuovo soggetto o sincretismo
antagonista?
4. Postilla sulle comunità
5. Due parole al posto d'una conclusione

PER TUTTO L'HACKMEETING:

**- WORKSHOP PER LA RETE
DELL'H2K (a cura di tutti i
partecipanti)**

**- ESPERIMENTO CLUSTER BOWULF
(a cura dell'hacklab di firenze);**

DESCRIZIONE:

Presentazione del progetto del Ciclope
dell'HackLab di Firenze

In occasione dell'HackMeeting di Roma
una parte dell'HackLab di Firenze
presenta la realizzazione di un
cluster di computer, simpaticamente
battezzato Il Ciclope.

Che cos'è un cluster

Un cluster di computer è una rete locale
nella quale i vari componenti collaborano
allo svolgimento di un solo programma
dividendosi il lavoro. Questa
"collaborazione" avviene usando una
comune rete basata sul protocollo tcp/ip e
una serie di programmi di controllo.

Il risultato di quest'aggregazione è una
specie di singolo computer con una
capacità di calcolo estremamente
elevata usando comunque computer e

componentistica estremamente comuni e di bassissimo costo.

Praticamente useremo una serie di computer basati su Pentium con quantità di ram che vanno dai 16 mega ai 48 e una Sparc, con sistema operativo Linux su tutte e una connessione con una normalissima rete composta ethernet a 10 megabit (se va bene ;-))) su cavi bnc.

Come realizzazione software ci muoveremo su due strade differenti: la più semplice sarà quella di utilizzare software già scritto per sfruttare questo tipo di architettura, come ad esempio il programma di rendering Persistence of Vision. Strada interessante e anche, diciamo così, folkloristica data la bellezza delle immagini ottenibili con questo programma. Per questo tipo di elaborazione è molto utile questo tipo di architettura perché la quantità di calcoli da effettuare è molto elevata, specialmente per immagini molto complesse, e quindi la potenza di calcolo di un cluster è cosa buona.

L'altra strada è molto più "hacker": utilizzando un programma scritto da un componente dell'HackLab cercheremo di scoprire una chiave DES usando la tecnica del testo in chiaro. Praticamente usando un testo criptato di cui abbiamo anche la versione in chiaro cercheremo di trovare la chiave di crittazione. Anche in questo tipo di operazione l'architettura cluster è estremamente utile perché anche qui c'è in ballo una gran quantità di calcoli.

Chiaramente saremo estremamente contenti se altri partecipanti all'HackMeeting porteranno idee e realizzazioni software da dare in pasto al Ciclope, nonché se ci verranno portati altri computer da aggiungere al nostro equipaggio (nel caso fatevi sentire).

Senso

Il senso di tutto questo sta nel vedere e poi nel far vedere se e come sia effettivamente realizzabile un computer con capacità di calcolo estremamente elevate usando componenti comuni e poco costosi. Poi che ognuno ne faccia quel che maggiormente gli aggrada e buon pro gli faccia.

Il tutto a <http://firenze.hacklab.it/cluster>

INSTALLAZIONI:

CANDIDA TV – ASCII STREAMING

- Proposta di Jaromil: la mia proposta di presentare un ascii-streaming framework potrebbe diventare una "installazione performance quelchevuoi", la utilizzerà radiocybernet e candida tv.

SNIFFER SU FOSFORI VERDI:

- Altra istallazione e' quella che si sta cercando di realizzare con lo sniffer a fosforoni, sara' mooolto cyberpunk se vien fuori come la stiamo pensando :) - se qualcuno la vuol chiamare arte lo faccia pure (a cura di Jaromil, Odo ed Efglas)

- HACK-SHOP

- Punti di scambio hardware (magari suddivisi per tipo)
- Banchetti distribuzioni varie (libri, riviste, e-zine, materiale su cd, ecc...)

- TRACKER JAM:

una stanza con amplificazione e mixer a disposizione di chiunque voglia portare le sue tracce (tracker o al limite /dev/midi).

CONCERTI SERALI:

Venerdi'

- REF – Roman Electronic Fighters
Passaranibenedettimaxdurante
four decks live set
- Jam (secondo la proposta venuta fuori
dalla lista: portatevi i computer
per suonare)

Sabato

- Mike Dredd (rephlex records)
Techno exp break bit dj set.