



1

UNDER ATTHACK

iteration hacking ezine

UnderAttHack n.1 by Hackingeasy Team

In_questo_numero() {

Prefazione al n.1 [by adsmanet].....3

Sicurezza

Social engineering [by MRK259].....5

Pen testing in Mozilla Firefox (parte 1) [by Elysia].....18

Open Source

Trashware e seconda giovinezza [by Floatman].....24

Fluxbox su Ubuntu...as I like [by vikkio88].....38

Programming

Analisi virale: il caso WINE [by Floatman].....57

}

Prefazione al n.1

UnderAttHack è giunto alla sua seconda edizione e questo dimostra che il progetto sta proseguendo il suo cammino e non era un semplice schizzo di follia del Team di Hackingeasy (o forse è follia continuativa, ma non vorrei che ci vantassimo così tanto).

Questa prefazione è piuttosto importante per il gruppo, perchè coprendo i mesi di Aprile e Maggio cade a cavallo del nostro primo anno di attività. Nel periodo intercorso tra la pubblicazione del numero 0 ed oggi, il nostro forum si è trasferito nel circuito di mastertop, decisamente più tranquillo della bolgia infernale di forumcommunity a cui vanno comunque i dovuti ringraziamenti per il servizio impeccabile che ci è stato fornito.

Il lavoro di trasloco ha coinvolto in maniera molto attiva tutti i membri della nostra comunità ed è stato appena precedente rispetto alla stesura del numero dell'e-zine che state leggendo in questo momento; dimostrazione quindi di una notevole capacità organizzativa e realizzativa di cui possiamo andare orgogliosi.

Invito quindi tutti i lettori a venirci a trovare nella nostra nuova sede virtuale:

<http://hackingeasy.mastertopforum.com>

Ad un anno dalla nostra nascita vorrei fare qualche considerazione generale che non interessi solo la nostra utenza.

Innanzitutto posso dire che il materiale prodotto, di cui UnderAttHack è forse l'elemento più visibile, è di un livello medio decisamente buono.

Questo dimostra come anche oggi sia possibile trattare di quel buon hacking innovativo e sperimentale che sembra scomparso dalla scena nazionale.

Forse noi di Hackingeasy ci stiamo illudendo di questa possibilità ma ne siamo convinti; anche se fosse soltanto un sogno sarebbe comunque un sogno di ispirazione creativa, quindi di valore più alto della realtà stessa.

Se è possibile trovare un metodo preciso a questo faticoso cammino, io tenderei ad individuarlo in tre punti:

Per prima cosa la creazione di un nucleo duro ed attivo, che dimostri non tanto competenza tecnica superiore, quanto piuttosto la forza e la voglia di mettersi in gioco in nuove sfide.

La seconda cosa, legata alla precedente, è un forte spirito collaborativo che

non crei sottogruppi di serie A e B ma che spinga ad assorbire conoscenze che hanno gli altri e a ricercare nuove competenze in campi inesplorati.

La terza ed ultima considerazione riguarda la struttura stessa dei nostri forum, che devono essere concepiti come laboratori del sapere e non come portali commerciali dove il successo è dato dai numeri dei contatti, oppure dalla quantità di spazio data-base occupato o ancora dal posizionamento che viene ottenuto nei motori di ricerca.

Tutte quelle logiche del web commerciale poco si adattano agli argomenti trattati ed è bene lasciarle ad altre tipologie di web-site per concentrarsi sul contenuto di quello che viene offerto.

Anche in questo numero di UnderAttHack trova spazio l'unione di voglia di sapere ed esperienze personali di chi scrive.

Si tratta ancora di uno spirito creativo, applicato e documentato per essere diffuso a tutti in modo che i percorsi individuali di tutti diventino base di partenza per gli sviluppi personali degli altri.

Mi auguro quindi che il contenuto sia di vostro gradimento, però vi chiedo che questo materiale non venga considerato come un insieme di appunti da utilizzare in modo semplicistico ma anzi venga assorbito per il suo "significato", cioè nella forma di un metodo mentale e di azione, in cui un problema che ci si trova davanti viene analizzato alle sue basi per generare soluzioni innovative ed utili.

Buona lettura...

adsmanet

Social engineering

(ingegneria sociale, aka come diventare cyber-marionette)

Ormai tutto il cyberspazio si sta sempre più “proteggendo” contro gli attacchi sferrati da hacker, (che siano white, black o gray hat...).

Infatti i computer ormai stanno diventando sempre più sicuri, grazie agli antivirus sempre più efficaci, ai firewall sempre più “completi” e perché no...anche grazie ai nuovi tool che aiutano i programmatori e i web master nello sviluppo di programmi e siti web...

Infatti oggi come oggi la facilità con cui risulta possibile sfruttare vari bug sta diminuendo...E ciò sta provocando un calo di “punti di accesso” ai malintenzionati...

Ma quando non si può entrare in un computer tramite virus e bug...si ricorre all'accesso grazie al così detto anello più debole della sicurezza informatica, cioè all'utente..

E per questo si è sempre più portati nell'utilizzare l'ingegneria sociale.....

Cos'è l'ingegneria sociale??

In poche parole equivale a *fregare il prossimo con la psicologia.*

Il social engineering infatti è l'insieme delle tecniche (**attenzione!! le tecniche sono psicologiche, non informatiche!!**), usate dagli aggressori, oggi prevalentemente on-line, per farci fare quello che vogliono: per esempio indurci a dare loro i nostri codici di accesso, a cedere loro dati sensibili, ad aprire i loro allegati infetti o a visitare un sito che contiene *dialer* o altro materiale pericoloso.

Con una spiegazione più tecnica (ma comprensibile) riguardante i metodi di intrusione nei sistemi informatici, possiamo dire che il social-engineering è la modalità più complessa per acquisire diritti amministrativi, che però permette di bypassare stadi di una catena di permessi che dal punto di vista tecnico risulterebbero inaccessibili.

Il seguente schema ci fa comprendere meglio il meccanismo:

Analisi del sistema

Stadio 1 → richiesta primo elemento → bypassabile con vulnerabilità 1

Stadio 2 → richiesta secondo elemento → bypassabile con vulnerabilità 2

Stadio 3 → richiesta terzo elemento → nessuna vulnerabilità conosciuta = Social-engineering

Stadio 4 → richiesta quarto elemento → bypassabile con vulnerabilità 4

Acquisizione piena dei diritti

Beh, possiamo tranquillamente fare alcuni esempi più pratici...

Immaginate foste dei ladri, e voleste rubare una Ferrari, cosa fareste??

Penso che il 40% abbia risposto: “Per prima cosa provverei a disattivare l'antifurto, e poi forzerei la serratura..”

Invece il 60% (bravi...) suppongo abbia detto: “Per prima cosa cerco di rubare le chiavi al

proprietario...”

Ebbene facciamo un altro esempio più realistico.

Mettiamo il caso che ci presentassimo dalla segretaria di Bill Gates e chiedessimo un assegno da 100 milioni...

Adesso invece prendiamo l'ipotesi che la segretaria riceva una chiamata dal caro vecchio Bill in cui riceve l'ordine di intestare a nostro nome un assegno di 100 milioni..

Quale dei due casi credete abbia più possibilità di riuscita?

Questi sono solo alcuni esempi di ingegneria sociale.

Gli esempi più comuni sono i tentativi dei *virus writer* di inviare false mail allegandovi virus. Infatti ormai tra gli autori di virus sta dilagando la moda di infettare più macchine possibili, (Qualcuno paragona questi stupidi atteggiamenti a “una sorta di gara a chi pisca più lontano”).

Prima gli allegati alle mail venivano eseguiti dai programmi di posta automaticamente, ora invece non avviene più, anche grazie agli antivirus e i diversi firewall che ne bloccano l'esecuzione..

Ma solitamente (e stupidamente) l'utente tipo crede che questi tool bastino per “barricare” le porte di accesso a virus e malware...

Purtroppo questa è vera e propria **ignoranza informatica**, poiché così facendo si ci affida **TOTALMENTE** a questi stupidi programmi, non considerando che i computer (fortunatamente) non possono pensare e decidere autonomamente il da farsi.

E qui scatta il social...

Infatti una volta che nella testa di un utente si è insinuata l'idea della sicurezza derivata dagli antivirus si inizia a diventare spavaldi e ciò nella vita reale e virtuale è sempre un male...

Infatti l'utente stupidamente inizierà a fare dei ragionamenti tipo:

“Toh.. un'e-mail con un allegato strano, inviata da un account di nome pippo@mail.com.... Posso anche aprirla, tanto c'è Kaspersky che mi protegge...”

Questo è un esempio stupido, infatti nessuna persona intenta ad infettarvi userebbe mai un account così strano...

Infatti principalmente lo scopo dei cyber-aggressori è quello di indurre le vittime a fidarsi il più possibile del contenuto della mail, e degli allegati o link, in modo da poterne trarre vantaggio.

E per rendersi il più possibile degni di fiducia e far abboccare le vittime si usano principalmente 7 punti chiave:

- Autorevolezza
- Colpa
- Panico
- Ignoranza

- Avidità
- Buoni sentimenti
- Generalità

(Chiaramente non sempre verranno usati tutti contemporaneamente...)

Andremo ora ad approfondire il significato di questi punti chiave:

L'autorevolezza

Di solito il social viene attuato tramite mail...

Ormai falsificare una mail è diventato un gioco da ragazzi...

Per fare alcuni esempi...

Provate a pensare all'account di prima...: *pippo@mail.com*, pensate che se avesse provato a inviare un'e-mail dicendo di far parte dello staff di *hotmail* e che gli servissero i vostri dati per verificare l'uso dello spazio mail gli avreste fornito ciò che voleva?? (non penso..)

Invece se vi avesse contattato un account con il nome: *staff@hotmail.it* sono sicuro che molte persone ci sarebbero cascate e gli avrebbero fornito i loro dati..

Ad esempio, in questo caso l'aggressore fa leva sul cosiddetto **principio** d'autorità.

Infatti qui si è spacciato per uno dei componenti dello staff Microsoft addetto ad Hotmail.

Questo metodo di solito compare nella maggior parte delle mail false, non a caso in tutte le mail falsificate vi sono copie dei marchi, loghi ecc. che usano le vere fonti autorevoli...(infatti nella presunta mail ricevuta da *staff@hotmail.com* ci sarebbero SICURAMENTE stati inseriti dei loghi della microsoft..con eventuali spot pubblicitari...

Colpa

Solitamente tutti ci sentiamo un po' colpevoli di aver fatto qualcosa di male...

Ad esempio aver scaricato film dal "caro vecchio mulo" o canzoni e documenti tramite qualche *dork* dello zio Google...

E naturalmente non scordiamo la pornografia...infatti io non credo che al mondo esista una persona fornita di una connessione a internet che non abbia mai guardato anche per sbaglio un'immagine o video "hard"...

Adesso...immaginate di ricevere una mail avente come mittente un indirizzo "*Sorveglianza@network@gmail.com*" e come oggetto: "*Violazione dei termini di legge contro la pedofilia*"...

Naturalmente molti di voi direbbero: "Ma che diavolo sta insinuando??" sapendo di non aver fatto nulla...

Ma qui interviene il fattore *colpa* : infatti si insinuerà un dubbio: "Non è che si sono sbagliati?" o in alcuni casi...: "Non è che in quel sito di ieri..." e così si apre la mail e il rispettivo allegato per sapere cosa sia successo...

In questo stereotipo tipo di mail un ruolo importante è giocato anche da un altro fattore...

Panico

Infatti il fattore panico e colpa sono quasi sempre accomunati, poiché:

Di solito quando si viene incolpati di qualcosa si ha la paura di essere preso di mira da possibili conseguenze.

Moltissime volte il fattore del panico viene scatenato dall'aggressore, che può, per esempio, inviarti un e-mail in cui dice che in circolazione c'è un pericolosissimo virus che non viene ancora rilevato dai normali antivirus, ma che viene debellato dal programma allegato; però bisogna fare presto!!

Ancora una volta, se la richiesta di eseguire l'allegato giungesse in un messaggio normale, non abbochieremmo: ma siccome siamo spaventati dal contenuto del messaggio, tendiamo a cadere nella trappola.

Ignoranza

Beh si sa che nella vita non si può sapere tutto...

Un esempio è internet...NESSUNO può sapere tutto di internet...per il semplice fatto che ogni giorno nascono siti nuovi, tecniche nuove, e perché no... a volte anche linguaggi nuovi...

Alcune volte i malintenzionati non si curano nemmeno di dare informazioni pertinenti tra loro...infatti basterà buttare nomi e sigle a caso per far credere all'utente che abbia letto qualcosa di estremamente complicato..

Per esempio se io inviassi una mail ad una persona con un livello di conoscenza informatica abbastanza normale dicendo:

“Ciao, ho notato che sei stato infettato, il virus è programmato da un cracker che mira ad effettuare un buffer overflow su un cookie grabber per poi attuare un ping of death sul tuo sql injection, in modo da bruciare il settore 0 del tuo terminale, se non vuoi bruciare il pc scarica questa xss allegata...”

Sicuramente il ricevente dell'e-mail non si curerà di cercare questi termini, (e se lo facesse si accorgerebbe delle stronzate che ci sono scritte..xD) bensì si affretterà a scaricare l'allegato...

Questo stato di incompleta o totale mancanza di informazione viene sfruttata da un'eventuale malintenzionato che invece di inviarti una mail con scritto:

“Ciao sono MRK e sono interessato a infettarti...potresti farmi il piacere di aprire il virus in allegato all'e-mail??”

Sicuramente ne invierà una con scritto:

“Ciao..sono MRK, sono un tecnico informatico facente parte del team di sviluppatori di Kaspersy, (AUTOREVOLEZZA) e facendo uno scan del tuo Hard disk, ho notato che ultimamente hai scaricato alcune canzoni illegalmente (COLPA), e tra queste ce n'è una infetta da un trojan pericolosissimo, identificato con la sigla 7X52Q3Z, questo trojan

potrebbe portare ad un'inarrestabile perdita di dati, o ad un'eventuale buffer overflow (PANICO DOVUTO ALL'IGNORANZA..). Per rimediare a questo trojan ti consiglio di scaricare l'unico antivirus efficace per ora..”

Avidità

Beh parliamo chiaro...chi è che se avesse l'occasione di intascare gratuitamente 13.000 euro non lo farebbe??

E proprio sull'avidità si basano moltissime pubblicità..

In quale sito non appare almeno una volta una pubblicità con scritto un messaggio del tipo:

“Congratulazioni!!! Sei online dalle 12.00 del 27/03/20xx hai vinto 10000 euro!!! Clicca e scopri come intascare il premio!!!! Non è uno scherzo!!!!”

E magari cliccando ci re-indirizzerà sul sito della Banca Mediolanum...

Buoni sentimenti

Alla fine dei punti essenziali che ci portano a riconoscere un tentativo di ingegneria sociale c'è il più usato: L'uso di eventuali buone intenzioni...

Infatti se, come suppongo io non esiste nessun uomo avente a disposizione una connessione a internet che non abbia mai guardato anche per sbaglio un'immagine o video “hard”, è anche vero che di solito le donne non sono per niente attratte dai siti e dalle immagini hard, quindi per indurre una donna in un tranello sarà sconsigliabile usare un link ad un sito porno, bensì si userà un approccio più cauto, che si appellerà ai buoni sentimenti della ragazza, facendola cadere nel tranello...

Un esempio potrebbe essere l'invio di una catena con scritto:

“Scopri chi ti sta pensando,clicca sull'allegato e non crederai ai tuoi occhi!!!”

Sicuramente avrà più possibilità di un messaggio che dice :

“Raoul Bova and richard Gere xxx clicca l'allegato e vedrai le loro torri di babele!”

(Scusate il brutto esempio.. xD)

Un altro esempio di appello ai buoni sentimenti potrebbero essere alcuni appelli strazianti per donare denaro alla croce rossa, donare una quota per ritrovare una ragazza (in realtà mai scomparsa)...

ATTENZIONE!!! LA MAGGIOR PARTE DELLE VOLTE QUESTE SONO SOLO CATENE O STUPIDE TRUFFE, POICHÈ GLI ENTI BENEFICI E ASSOCIAZIONI DI VOLONTARIATO VERI MOLTO DIFFICILMENTE DISTRIBUIREBBERO APPELLI VIA MAIL...

Generalità del messaggio

Molte volte si possono riconoscere dei falsi anche da un modo di parlare diretto in generale... Infatti se ci vogliono far abboccare non potranno dirci:

“Signor Pippo abbiamo avuto una richiesta dal signor Topolino di un trasferimento dal suo conto di 100\$. Per confermare o annullare l'azione legga le istruzioni in allegato..”

Poiché gli autori della mail non possono conoscere il nome del ricevente, tanto meno può indovinare che conosca il signor “topolino”...

E quindi gli eventuali malintenzionati formuleranno la mail in modo da risultare vaghi e convincenti allo stesso tempo..

Quindi la mail potrebbe essere espressa così:

“Distinto cliente, abbiamo ricevuto una richiesta di trasferimento di 100\$ dal suo conto. Per conoscere il beneficiario e confermare e/o annullare l'azione di trasferimento segua le istruzioni allegate.”

E già ci sarà una possibilità infinitamente maggiore di poter fregare il signor “pippo”..

Bene...credo di aver illustrato una buona parte dei metodi usati nel social...

Ora che siete al corrente dei punti critici che usano gli attaccanti, saprete (spero) individuare una falsa mail da una vera..

Ma ci sono dei dettagli che potrebbero sfuggire....

Infatti finora ho trattato solo eventuali situazioni in cui un malintenzionato volesse far aprire un allegato...

Ma per estorcere dati sensibili e/o infettare il nostro computer non hanno necessariamente bisogno di un'allegato...

Infatti in precedenza quando ho fissato una “bozza” del termine social engineering ho detto che: **le tecniche sono psicologiche e non informatiche...**

Ribadisco quest'espressione, ma ci tengo a dire che le tecniche psicologiche nel social vanno perfettamente a braccetto con quelle informatiche...

Infatti il social può essere formato da pura psicologia, ma può anche essere misto a *fake*.

I *fake* sono dei “falsi”, e possono comparire in molte situazioni..

Infatti nelle community si chiamano fake gli utenti che si spacciano per altri copiandogli il nick...Ma in sostanza qualsiasi cosa che vuole spacciarsi per un'altra è un fake.

Nel social i fake sono molto importanti, perché gli aggressori molte volte creano finte pagine di login che invece di inviare i dati al database originale lo inviano ad altri contraffatti o addirittura scrivono semplicemente i dati di accesso su dei file di testo...

Ora andremo a vedere alcuni aspetti per difenderci proprio dai fake...

Per esempio ormai è diventata famosa la truffa di poste italiane... di cui cito qui il contenuto (attenzione ci sono diversi tipi di falsi!!):

OGGETTO: Poste Italiane premia il suo account con un bonus di fedeltà
DA: BPOL1@posteitaliane.it

Caro Cliente,

L'importo vinto le sarà accreditato sul Conto BancoPostaOnline o sulla carta Poste Pay.
Per ricevere il BonusFedeltà è necessario accedere ai servizi on-line entro 48 ore dalla ricezione di questa e-mail

Importo: 200,00
Commissioni: 1,00
Importo totale: 201,00

[» Accedi ai servizi online per accreditare il Bonus di Fedelta](#)

Per ulteriori informazioni consulta il sito www.poste.it o telefona al numero verde gratuito 8534160.

Distinti Saluti
Poste Italiane S.p.A.

Nota: il link è ovviamente stato modificato in nolink.net per motivi di sicurezza; il link originale della mail è questo:

*[http://www.sdherman.com/.redirectioni/redirectioni.html?
MfcISAPICommand=SignInFPP&UsingSSL=1&email=&userid=?index.php?
MfcISAPICommand=SignInFPP&UsingSSL=1&email=&userid=?
MfcISAPICommand=SignInFPP&UsingSSL=1&email=&userid=](http://www.sdherman.com/.redirectioni/redirectioni.html?MfcISAPICommand=SignInFPP&UsingSSL=1&email=&userid=?index.php?MfcISAPICommand=SignInFPP&UsingSSL=1&email=&userid=?MfcISAPICommand=SignInFPP&UsingSSL=1&email=&userid=)*

già di per sé molto invitante xD

Ora andremo ad analizzare questa mail e i punti che dovrebbero farci individuare la sua falsità...(scoprendo anche qualcosa che non ho detto fino ad ora..)

Iniziamo con i punti principali che ho illustrato in precedenza..

In questa mail possiamo riconoscere:

- Autorevolezza
- Panico
- Ignoranza
- Avidità
- Buoni sentimenti
- Generalità

Autorevolezza

dovuta all'ente delle poste..(*“o caccio mi hanno contattato le poste italiane!! chissà cosa vogliono...!”*)

Panico

dovuto al tempo disponibile per effettuare l'accredito..(*“Devo sbrigarmi altrimenti non riceverò quei 200 euro!!”*)

Ignoranza

infatti NON esiste un “bonus fedeltà” elargito dalle posteitaliane...

Avidità

dovuta alla vista della somma in regalo..(*“Che cu*o !!ho vinto 200 euro!!”*)

Buoni sentimenti

infatti la mail cerca di far credere ad un regalo elargito dalle poste..(*“No...che bravi hanno anche un bonus fedeltà?!”*)

Generalità

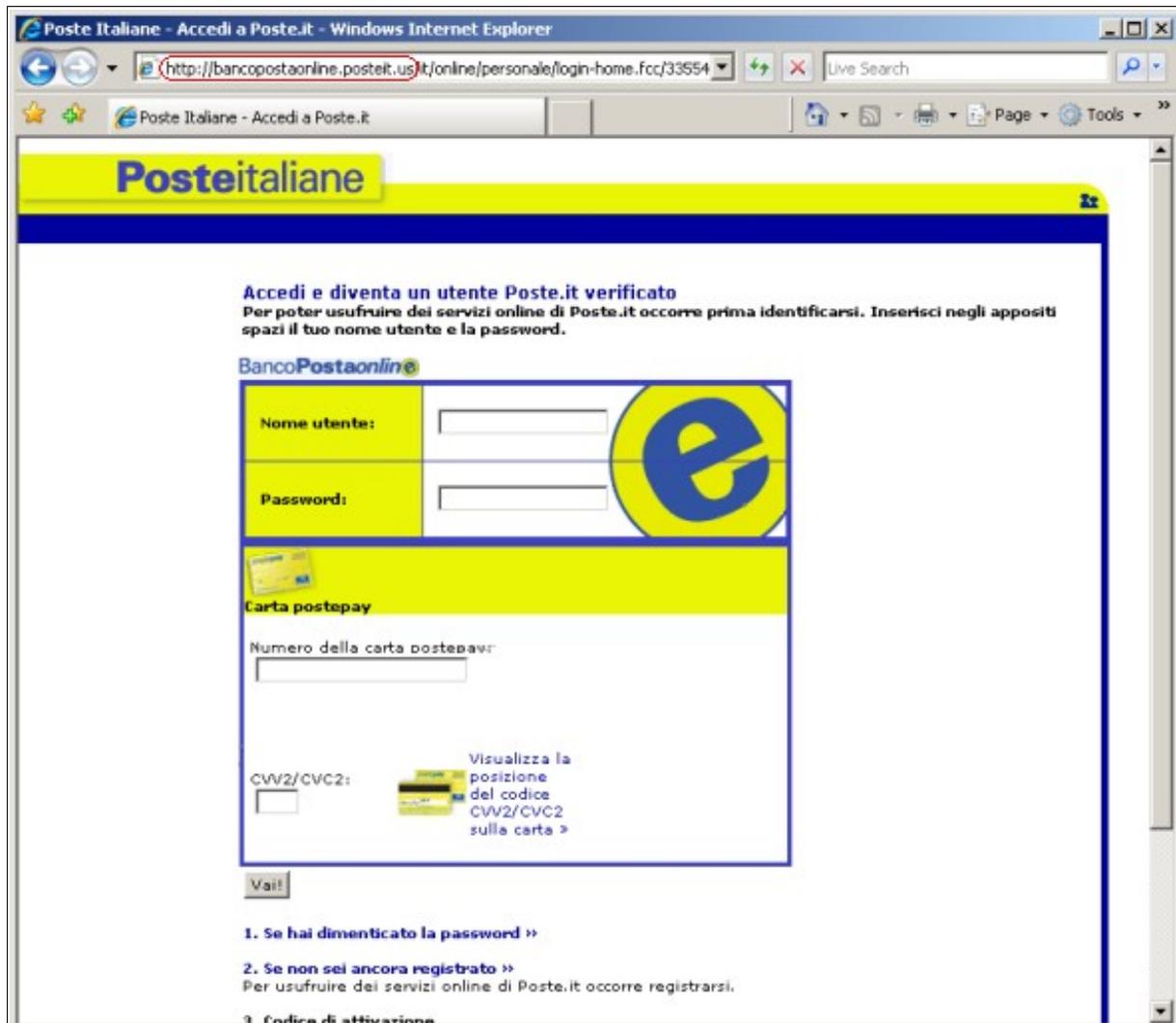
infatti la mail non fornisce nessun dato utile...inizia subito spedita con: *“L'importo vinto le sarà accreditato”* e non parla né del perché abbiamo vinto, né cita mai il nome di un account...

Infatti se fosse stata una mail vera sarebbe iniziata con un messaggio del tipo: *“Gentile MRK (o in alternativa: gentile cliente numero 12234..) lei ha vinto un bonus...”*

Oltre a questi punti però come si può notare vi è un link che, dal nome DOVREBBE reindirizzare ad una pagina delle poste italiane, invece basterà passare con il mouse al disopra del link da accorgersi che rimanda ad una pagina che non c'entra nulla con le poste italiane... Dopo essersi accorti di questo fatto si è certi al 100% che questa mail è un fake (per giunta fatto male..)

Ma dato che noi siamo molto curiosi andremo a cliccare sul link...

Cosa ne verrà fuori?



P.s. L'immagine è un po' vecchiotta..

Già..proprio così..il campo di accesso utente al sito delle poste...

MA un momento...Siamo sicuri che sia davvero delle poste italiane??

Io dico di no...

Già...infatti come possiamo notare nella barra degli indirizzi l' url è:

<http://bancopostaonline.posteit.US>....

La grafica è ok...il nome del sito è ok...allora perché dovrebbe essere un falso???

Semplice..Il dominio è Statunitense, e non italiano...E che senso avrebbe per le poste ITALIANE avere un dominio di primo livello registrato negli USA???

Questa pagina potrebbe trarre in inganno tantissime persone, poiché è stata falsificata molto bene..

Un'ulteriore errore però è nel protocollo utilizzato...

Infatti il comune http ovvero: Hyper Text Transfer Protocol in una connessione che richiede login o altri dati sensibili (password, nome di login) viene sostituita dalla sua versione “sicura” ..L' https ovvero:Hyper Text Transfer Protocol Security.

Come possiamo notare, questo sito utilizza l'http e **NON** l'https, poiché i dati non hanno bisogno di essere protetti, per un fatto molto semplice...

Di solito infatti i dati vengono criptati e inseriti nel database del sito...

Questa volta però, dato che si ha a che fare con un *fake login* i dati verranno inviati ad un falso database accessibile dai malfattori o semplicemente scritti su blocchi di testo uppati nel dominio web...

Ormai credo (e spero..) sappiate come difendervi...

Ma rileggendo la falsa mail qualcuno ora potrà pensare:

“Come mai però il mittente risulta essere delle poste???”.

Semplicemente inviando una mail anonima..

Per inviare mail anonime con windows si può ricorrere ad alcuni programmi o siti appositi;o in alternativa usare il telnet.

Ora andremo a vedere come usare il telnet per l'invio di una mail anonima:

Per inviare mail con telnet bisognerà prima connettersi al programma, per prima cosa apriamo il prompt di MS-DOS e digitiamo:

```
C:> telnet mail.tin.it 25
```

questo comando ci permetterà di collegarci al server mail sulla porta numero 25.

Dopodiché andremo a “presentarci al server” tramite il comando “HELO” seguito da un nome che ci identificherà:

```
HELO libero
```

Qui ho scelto il nome: libero e mi sono presentato...

Dopodiché andremo a inserire i dati di invio e ricezione per il server:

```
MAIL FROM: <destinatario@underatthack.it>  
RCPT TO: <mittente@underatthack.it>
```

il primo dato indica il destinatario della nostra mail; il secondo indica l'indirizzo del mittente.

Ora andremo ad inserire il comando “data” che ovviamente inserirà la data d'invio nella mail...

```
DATA inserisce la data nella mail
```

E ora andremo a inserire il nome dell'account con cui si vuole essere identificati:

FROM: <MRK@underatthack.259>

(Questo è il mittente *che comparirà nella mail*)

Il nome del destinatario:

TO: <nomedestinatario>

E l'oggetto della mail..

SUBJECT:bo

E da qui si potrà scrivere il corpo del messaggio:

Prova

Un punto seguito dal comando QUIT o da invio determina la fine del messaggio e l'invio della mail..

```
.  
QUIT
```

Ora basterà attendere qualche secondo, e poi potrete chiudere la schermata del cmd poiché l'e-mail sarà stata inviata con successo...Per comodità consiglio di inserire i comandi in un blocco note e in seguito incollarli.

Andiamo a riassumere il procedimento..

Apriamo il cmd di windows e digitiamo:

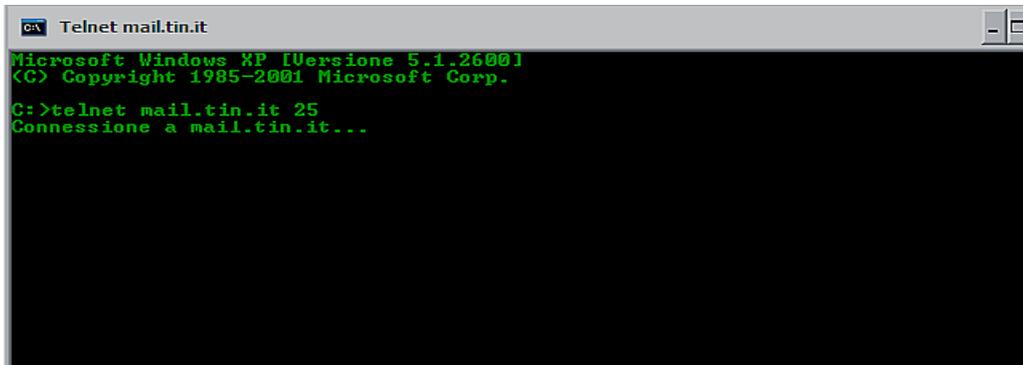
```
telnet mail.tin.it 25
```

Dopodiché per inviare la mail dovremo digitare:

```
HELO libero  
MAIL FROM:<destinatario@esempio.it>  
RCPT TO:<mittente@esempio.it>  
DATA  
FROM: <bo@esempio.com>  
TO: <nome>  
SUBJECT:bo  
  prova  
.  
QUIT
```

Qui di seguito ci sono degli screen dei passaggi fondamentali...

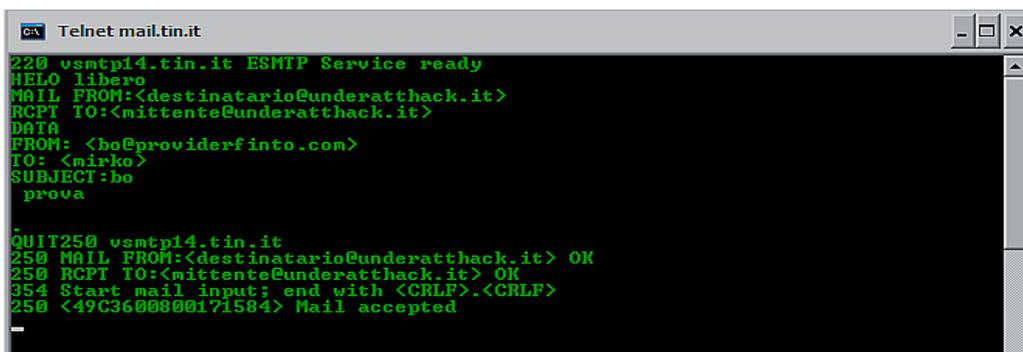
1. Accediamo a telnet:



```
Microsoft Windows XP [Versione 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:>telnet mail.tin.it 25
Connessione a mail.tin.it...
```

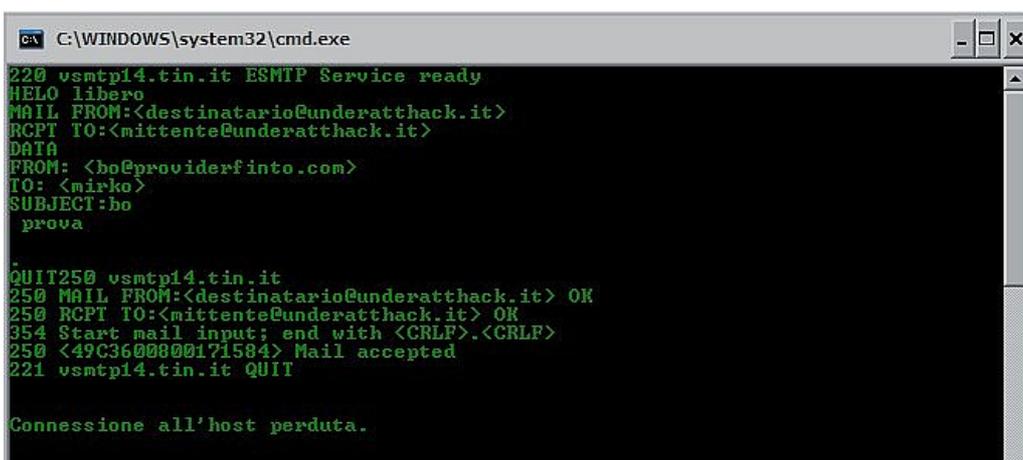
2. Incolliamo tutti i comandi:



```
220 vsmtpl4.tin.it ESMTP Service ready
HELO libero
MAIL FROM:<destinatario@underatthack.it>
RCPT TO:<mittente@underatthack.it>
DATA
FROM: <ho@providerfinto.com>
TO: <mirko>
SUBJECT:ho
prova

QUIT250 vsmtpl4.tin.it
250 MAIL FROM:<destinatario@underatthack.it> OK
250 RCPT TO:<mittente@underatthack.it> OK
354 Start mail input; end with <CRLF>.<CRLF>
250 <49C3600800171584> Mail accepted
```

3. Premiamo invio concludendo la connessione al server:



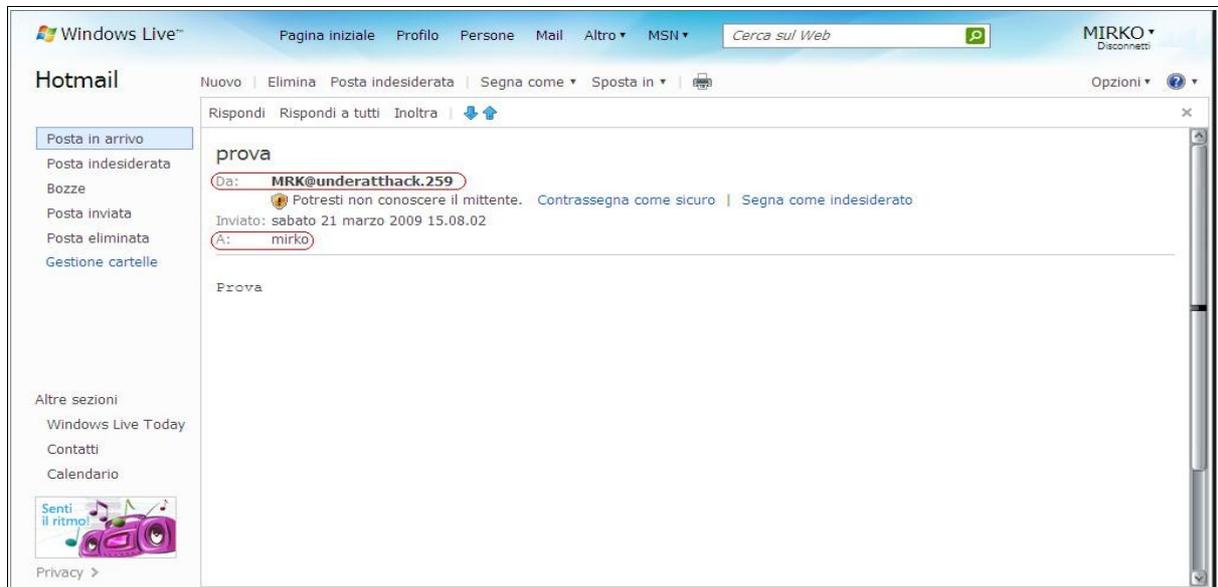
```
C:\WINDOWS\system32\cmd.exe

220 vsmtpl4.tin.it ESMTP Service ready
HELO libero
MAIL FROM:<destinatario@underatthack.it>
RCPT TO:<mittente@underatthack.it>
DATA
FROM: <ho@providerfinto.com>
TO: <mirko>
SUBJECT:ho
prova

QUIT250 vsmtpl4.tin.it
250 MAIL FROM:<destinatario@underatthack.it> OK
250 RCPT TO:<mittente@underatthack.it> OK
354 Start mail input; end with <CRLF>.<CRLF>
250 <49C3600800171584> Mail accepted
221 vsmtpl4.tin.it QUIT

Connessione all'host perduta.
```

Ed infine il risultato:



Ora spero che abbiate capito con quale facilità avrei potuto spacciarmi per un account diverso, come quello delle poste..

Vorrei specificare che non ho parlato dell'invio di mail anonime con GNU/Linux poiché per l'utenza GNU/Linux ci sono a disposizione molti programmi di Anonymous Remailer, ovvero dei programmi che sfruttano dei server che inviano le mail senza rivelare la loro provenienza.

Ora che avete letto questo manuale spero che prima di seguire le istruzioni di una mail vi assicuriate prima che sia autentica, e che non diventiate “marionette” pilotate dai fili di qualche astuto ingegnere sociale.

MRK259

Pen testing in Mozilla Firefox (parte 1)

Musica ascoltata:

magrudergrind – emo holocaust
napalm death – caught.. in a dream
hatred surge – old and tired
insect warfare – human slaughterhouse

Come al solito non ho una mazza da fare, e voi a quanto pare pure... quindi siamo in sintonia e possiamo cominciare xD

In questo viaggio che ci seguirà per alcuni numeri vorrei darvi una panoramica su vari plugin di firefox che ci consentono di testare applicazioni web (ovviamente vostre ;-)) cominciando con questo numero dal famigerato *tamper data* (la versione per donne si chiama *tampax data* ^^) dico famigerato perchè consente di trovare falle e fare danni all'inverosimile nelle applicazioni che non prevedono l'esistenza di questo giocattolo stesso, ovvero quelle cosiddette colabrodo.

Se deciderete di scolare la pasta sappiate che viene cotta con acqua bollente, quindi non venite a rompermi poi per le ustioni di quarto o quinto grado ;-) ... a buon intenditore... a buon intenditore!! (bitta docet) ok, basta con le cavolate.. let's go!

Tamper data (se non sapete come scaricarlo e installarlo è inutile proseguire con la lettura, non siete in grado)

Per illustrarvi il funzionamento del *tampax* (d'ora in poi lo chiamerò così), partirò da una mia esperienza che mi ha consentito di trovare una serie di bug nelle.. rullo di tamburi... *flashchat*.

Sì, avete capito bene... quelle odiose chat che proliferano tra i bimbininkia che, non contenti dei loro blog di cristina d'avena, si dedicano pure a ritrovarsi per chattare assieme ad una caterva di lamerz... ognuno vuole la sua, così le chat vulnerabili sono milioni... a dover di cronaca vi dico pure che ho segnalato il bug più grave su *bugtraq* ma non mi pare che l'abbiano letto i programmatori della chat ed in ogni caso le chat aperte ormai sono troppe per poter fixare il tutto.

Ma cos'è e a cosa serve il *tampax*??

Il *tampax* in pratica ci consente di modificare gli header delle richieste *http* ed *https*, ed inoltre di modificare i parametri di una richiesta effettuata mediante *metodo post* del protocollo *http*; se vi ricordate, la differenza tra la *get* e la *post* è che i parametri della *get* vengono concatenati all'url (quindi facilmente modificabile) della richiesta mentre i parametri della *post* risultano “nascosti” nel pacchetto stesso.

Il *tampax* si comporta da proxy intercettando il pacchetto in locale, facendoci fare tutte le modifiche che vogliamo e spedendolo a destinazione con i dati modificati!!

Vi lascio immaginare il bordello che può succedere se non viene effettuato *input filtering* (filtraggio dell'input mediante white list) e i dati che immettiamo vengano dunque accettati dall'applicativo sul server.

Dopo questa lunga meditazione per la risposta apriamolo dal menù “Strumenti” del firefox cliccando su “Modifica con tamper data”.

A questo punto si aprirà una finestra in cui premendo su “Avvia la modifica” il tampax comincerà a bloccare i pacchetti chiedendoci per ognuno se vogliamo inviarlo senza modifiche o modificarlo; proviamo a vedere cosa succede aprendone uno qualsiasi per modificarne i campi...

Andiamo su un sito qualsiasi con una form (per esempio un forum) e modifichiamo il pacchetto di invio dei dati della form stessa dopo averla compilata (es. scriviamo un nuovo messaggio in tag).

Per lo screenshot mi sono recato sul forum di hackingeasy, ho scritto un messaggio “FOD” e premuto su invia; il tampax mi ha chiesto se volevo modificarlo e gli ho detto di sì.

Ecco cosa ci appare: a sinistra i valori dell'header della nostra richiesta, a destra il contenuto della post in cui si vede chiaramente il valore del campo msg che contiene il messaggio da noi digitato; provando a cambiare il testo del messaggio, cancellando “FOD” e scrivendo “fuck %20off%20and%20die” (vi ricordo che la codifica è sempre URL quindi per gli spazi si usa %20), vedremo che verrà scritto in tag “fuck off and die” e quindi che la modifica ha avuto successo.

Richiedi il nome dell'intestazione		Richiedi il valore dell'intestazione		Inserisci il nome dell'intestazione		Inserisci il nome dell'intestazione	
Host		hackingeasy.mastertopforum.com		fcolor		none	
User-Agent		Mozilla/5.0 (Windows; U; Windows NT 5.1; it; rv		name		Anonymous	
Accept		text/html,application/xhtml+xml,application/xml		sb_user_id		-1	
Accept-Language		it-it;q=0.8,en-us;q=0.5,en;q=0.3		message			
Accept-Encoding		gzip,deflate		msg		FOD	
Accept-Charset		ISO-8859-1,utf-8;q=0.7,*;q=0.7		mode		submit	
Keep-Alive		300		submit_button		Invia	
Connection		keep-alive					
Referer		http://hackingeasy.mastertopforum.com/shoutl					
Cookie		phpbb2mysql_data=a%3A2%3A%7B%3A1%3A					

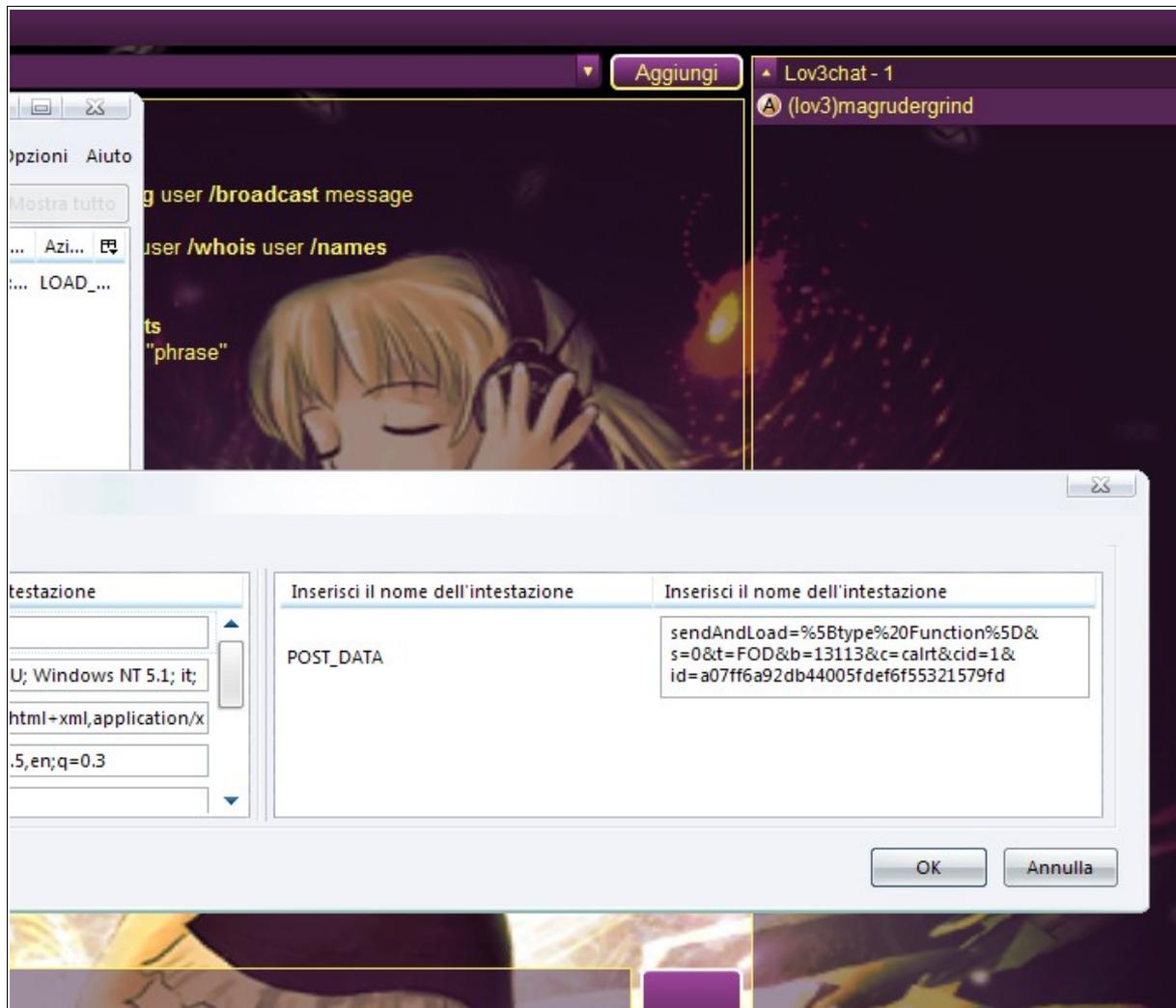
Possiamo cambiare il *referer* (pagina referente della richiesta), l'*user-agent*, i *cookie*, insomma tutto ciò che è modificabile.

Tutto qui??

Se vi sembra poco continuate con la lettura per capire la reale potenza del tampax...

Andiamo in una flashchat in cui siamo amministratori di sistema e proviamo a mandare un allarme alla chat, catturandone il pacchetto: scriviamo /chatalert FOD.

Questo è il pacchetto catturato:



Come vedete è praticamente una get nascosta in un solo parametro di una post chiamato POST_DATA.

I parametri che ci interessano sono:

$s=0$

$t=FOD$ che è il testo dell>alert

$c=calrt$ che è il nome del comando, ovvero un chat alert.

Proviamo un po' a spulciare il sorgente per vedere come vengono interpretati questi parametri che passiamo; vi incollo direttamente le linee che ci servono, prese dal file "connection.php" nella cartella /chat/inc/classes di una installazione di default della flashchat.

```
<!-- inizio codice pastato-->

//admin commands

if(
  ChatServer::userInRole($this->userid, ROLE_ADMIN) ||
  ChatServer::userInRole($this->userid, ROLE_MODERATOR) ||
  ($req['s'] == 7)
)
{

  switch($req['c']) {
    case 'alrt' : $this->doAlert($req['u'], $req['t']); break;
    case 'ralrt' : $this->doRoomAlert($req['r'], $req['t']); break;
    case 'calrt' : $this->doChatAlert($req['t']); break;
    case 'banu' : $this->doBanUser($req['u'], $req['b'], $req['r'],
$req['t']); break;
    case 'nbanu' : $this->doUnbanUser($req['u'], $req['t']); break;
    case 'gag' : $this->doGag($req['u'], $req['t']); break;
    case 'ngag' : $this->doUnGag($req['u'], $req['t']); break;
    default: addError("Unhandled admin request: {$req['c']}");
break;

  }

<!--fine codice pastato-->
```

Questi sono i comandi dell'admin, notate bene la funzione *if* perchè contiene una cosa abbastanza stupida; i più svegli di voi avranno notato che è un filter-bypass che ci permette di usare i comandi di admin anche se non siamo né admin né moderatori ^^

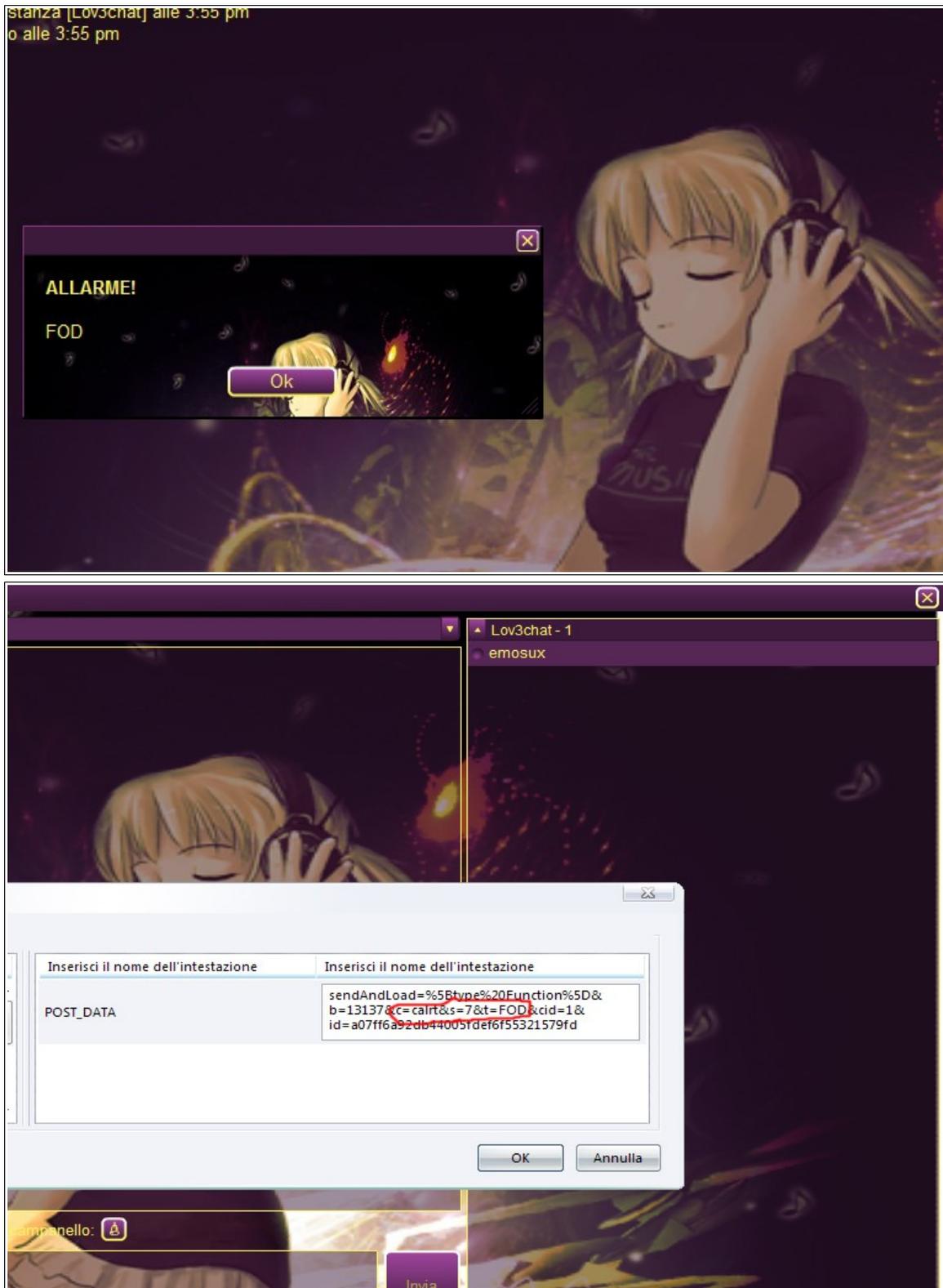
Basta semplicemente impostare il parametro *s* della richiesta al valore 7!!!

Probabilmente gli sarà servito per permettere ad un bot il ban e la possibilità di mandare allarmi.

Dove sta il problema in tutto ciò???

Il problema sta nel fatto che noi grazie al tampax il valore di *s* lo possiamo cambiare anche se è nascosto in una post e quindi comportarci esattamente come se fossimo admin della chat da utenti normali, sfruttando il bypass!!

Per verificare, proviamo a riconnetterci alla chat da utenti normali e a mandare un allarme; questo normalmente non ci è concesso e visualizziamo un messaggio di errore; proviamo allora a modificare il pacchetto col tampax settando il valore di *s* pari a 7:



il parametro della richiesta è più che ben accetto e siamo riusciti a mandare l'allarme da utente normale.

Questo stesso meccanismo è sfruttabile per tutti i comandi degli admin che sono presenti nel codice pastato semplicemente cambiando il valore di *s*, ban compreso.

Abbiamo dunque visto come sia utile poter modificare i dati da inviare al server per far buon pen testing sulle nostre applicazioni web ed evitare che dati erronei, malformati e volutamente cambiati non previsti siano accettati dando origine a problemi anche gravi.

Applicazioni mal progettate potrebbero ad esempio consentirci di cambiare i prezzi della merce per gli acquisti on-line, effettuare attacchi DoS, inserire codici anomali DOPO che il filtro è entrato in esecuzione bypassandolo, fingerci chi non siamo, eludere alcune tipologie di filtri basate sull'user-agent, e così via...

Elysia

Trashware e seconda giovinezza

Premessa: l'articolo che segue non sarebbe mai stato realizzato senza la consulenza di una persona a me molto cara che ha chiesto esplicitamente di non essere nominata. Se questo documento raccoglie il vostro favore complimentatevi più con R che con Floatman.

Una delle particolarità di GNU/Linux maggiormente apprezzate è la sua estrema configurabilità.

Purtroppo (o per fortuna...) negli ultimissimi anni credo che la spinta maggiore alla diffusione di questo sistema operativo in ambiente desktop sia da attribuire alle mirabolanti diavolerie di Compiz.

Nella realtà la possibilità di plasmare un sistema GNU/Linux in base alle proprie esigenze comporta possibilità di utilizzo decisamente più nobili e tecnicamente molto più apprezzabili.

Con il termine *Trashware* si intende il recupero di hardware obsoleto per il suo riassetto o la sua rivitalizzazione diretta tramite l'utilizzo di software libero.

Apparentemente questa attività potrebbe sembrare l'ennesimo hack compiuto per gioco dal tipico gruppo di smanettoni su un vecchio pc dismesso; in realtà il trashware comporta ragionamenti ben più complessi:

Innanzitutto ci si chiede cos'è l'hardware obsoleto, o meglio si ragiona su come la commercializzazione di sistemi sempre più esosi di risorse porti ad un invecchiamento precoce di una macchina, oltretutto su pc della fascia più bassa di utenza.

Ha senso che nel 2009 in una macchina ad utilizzo server l'hardware invecchi mediamente dalle tre alle cinque volte più lentamente che in un pc di casa?

La situazione reale è che l'hardware non invecchia ed è la diffusione di nuovo software che lo rende non più utilizzabile. Quanti di voi hanno “cestinato” pc con Windows 9x perfettamente funzionanti, per il solo fatto che ormai avevano un sistema inutile?

Un secondo fattore è invece di tipo strettamente economico. Il costo di un sistema operativo proprietario è oggi a carico dell'acquirente, che nella realtà non si rende conto della spesa, in quanto l'onere sostenuto risulta inserito nel costo di acquisto della macchina; nel momento in cui il sistema operativo diventa obsoleto automaticamente lo diventa anche l'hardware.

Quando infatti il sistema operativo invecchia, il costo complessivo per avere un pc “al passo con i tempi” risulta composto da due fattori: l'acquisto di un nuovo SO (questa volta a carico pieno dell'utilizzatore) più la spesa necessaria per l'aggiornamento dei componenti hardware, normalmente non più in grado di sostenere le richieste del sistema più evoluto.

Il risultato è ovviamente che la somma di questi due valori rende di solito più conveniente l'acquisto di una nuova macchina.

Visto i rapidissimi tempi di “evoluzione” dei sistemi operativi, ci si trova di fronte ad una situazione in cui anche macchine decisamente nuove risultano obsolete (ma quindi anche acquistabili per cifre irrisorie).

Un terzo fattore è invece più etico e riguarda il divario tecnologico esistente a livello mondiale in cui il mezzo informatico rimane possibilità di chi se lo può permettere.

Questo punto di vista potrebbe apparire correlato ad una serie di altri problemi di ordine politico-economico di cui l'informatizzazione globale rappresenta un fattore decisamente trascurabile. In realtà questo fenomeno non ha natura tecnologica ma è puramente legato ad interessi commerciali dei produttori di sistemi proprietari e ci tocca tutti!

Farò un esempio che fa molto riflettere; fate il favore di seguirmi per comprenderlo...

Oggi la fascia laptop ha un'esplosione di vendite; fermatevi un secondo, pensate mentalmente ai requisiti hardware necessari ad avere un sistema operativo molto moderno...

Bene, a questo punto pensate grosso modo al costo che questo pc potrebbe avere...

Ok. Sappiate che lavorando su software libero per avere un pc dalle prestazioni decisamente superiori a quelle che avete in mente vi basta un vecchio portatile che stia intorno a valori di RAM sui 128 MB; anche su 64 mega potreste viaggiare senza problemi.

Quanti soldi vi hanno fregato?

Lo scopo di questo documento è quello di introdurre le tecniche e le potenzialità del trashware; fare un esempio concreto sarebbe infatti fuorviante in quanto ogni macchina presetterebbe le proprie caratteristiche individuali, considerando anche il fatto che si andrebbe a comprendere hardware con una distanza temporale di 10-15 anni di possibilità.

Prima però di addentrarci in questo mondo è però il caso di vedere la nota dolente:

il trashware è un'attività che necessita di una buona conoscenza dei sistemi GNU/Linux, non pensate di alzarvi dalla sedia alla fine di questo articolo e rivitalizzare il vostro pc che sta in cantina dall'oggi al domani se non avete competenze su questo sistema operativo.

Concetti generali

Chi oggi conosce GNU/Linux sa che ormai tutta l'analisi dell'hardware supportato che veniva fatta un tempo non è più necessaria.

Anche se in giro per la rete, nei forum di aiuto all'utenza GNU/Linux, si trovano spesso richieste relative ad hardware supposto non compatibile, chi utilizza questo OS da lungo periodo sa bene che la compatibilità raggiunta oggi è a livelli decisamente ottimi.

Adesso possiamo dire che con l'esclusione di casi particolarmente "esotici" ogni tipo di hardware risulta bene o male accettato in modo automatico o dopo un processo di configurazione più o meno complesso.

Completamente diverso è il caso di hardware datato, in cui può accadere che i moduli del kernel delle comuni distribuzioni tendano a sfiorare i parametri di compatibilità necessari ad un corretto funzionamento, trascurando componentistica sorpassata (ovviamente parliamo di macchine con una decina di anni di vita).

Il controllo accurato dei moduli presenti nel kernel scelto per la procedura di trashware si accompagna alla necessità di una ricompilazione molto più spesso di quanto oggi si è abituati

a vedere.

Spesso si preferisce per ovvi motivi lavorare su versioni 2.4 di Linux, in modo da avere meglio garantita quella compatibilità hardware che predilige caratteristiche più vecchie a discapito di quelle legate alle ultimissime generazioni di dispositivi.

Le soluzioni normalmente adottate nella progettazione e realizzazione di opere di trashware sono riassumibili in tre grandi categorie:

- Utilizzare versioni delle distribuzioni precedenti a quelle attuali, almeno al momento della prima installazione, cercando di individuare dal punto di vista temporale un modello di hardware a cui quella distro può fare riferimento.
Ovviamente solo le distribuzioni più storiche si adattano a questo scopo per il fatto di avere vecchie versioni in grado di supportare hardware più vecchi; considerando comunque il fatto che queste distro sono quelle più conosciute e meglio supportate la cosa risulta tutto sommato positiva.
Una volta effettuata la prima installazione e configurazione, risulta possibile anche un relativo aggiornamento di determinati software, sempre prestando attenzione alle possibilità del sistema e ovviamente non permettendo mai l'upgrade di versione.
- La seconda possibilità è quella di lavorare da scratch o quasi-scratch.
Si procede cioè alla costruzione di un sistema fatto completamente a mano dalle basi, oppure si inizia da una micro-distribuzione di quelle contenute in un floppy per poi procedere all'ampliamento del software incluso tramite la normale installazione da sorgenti. Tale metodo risulta ovviamente quello più lento ma più valido; da notare il fatto che l'operazione di inserimento di nuovi sorgenti può avvenire in modo automatico tramite l'utilizzo di appositi script, allo stesso modo in cui la micro-distribuzione iniziale può essere a sua volta prodotta da scratch in maniera specifica per svolgere questo lavoro.
- La terza possibilità è quella di appoggiarsi a particolari sistemi GNU/Linux nati appositamente per questi scopi. Questo secondo metodo può comportare molti pregi ma anche parecchi difetti:
Da un lato l'installazione può risultare decisamente più semplice, sia per la presenza di un sistema già configurato per hardware obsoleti, sia per i numerosi tool appositi distribuiti dalle stesse distro che facilitano notevolmente il lavoro.
Come altro lato della medaglia spesso ci si trova di fronte a distribuzioni piuttosto specifiche, normalmente dotate di software apposito e caratteristiche particolari adottate per restringerne il volume, che poi non è detto offrano buone possibilità di configurazione che fuoriescano dai parametri stabiliti dal produttore.
Queste particolari distro però hanno comunque una funzione molto importante, cioè quella di farci comprendere quali software possano venire utilizzati per la realizzazione di un sistema operativo adatto a macchine vecchie e poco potenti.

In riferimento a questo secondo metodo, non possiamo non citare le due distribuzioni che meglio rappresentano questi principi.

Damn Small Linux: <http://www.damnsmalllinux.org>

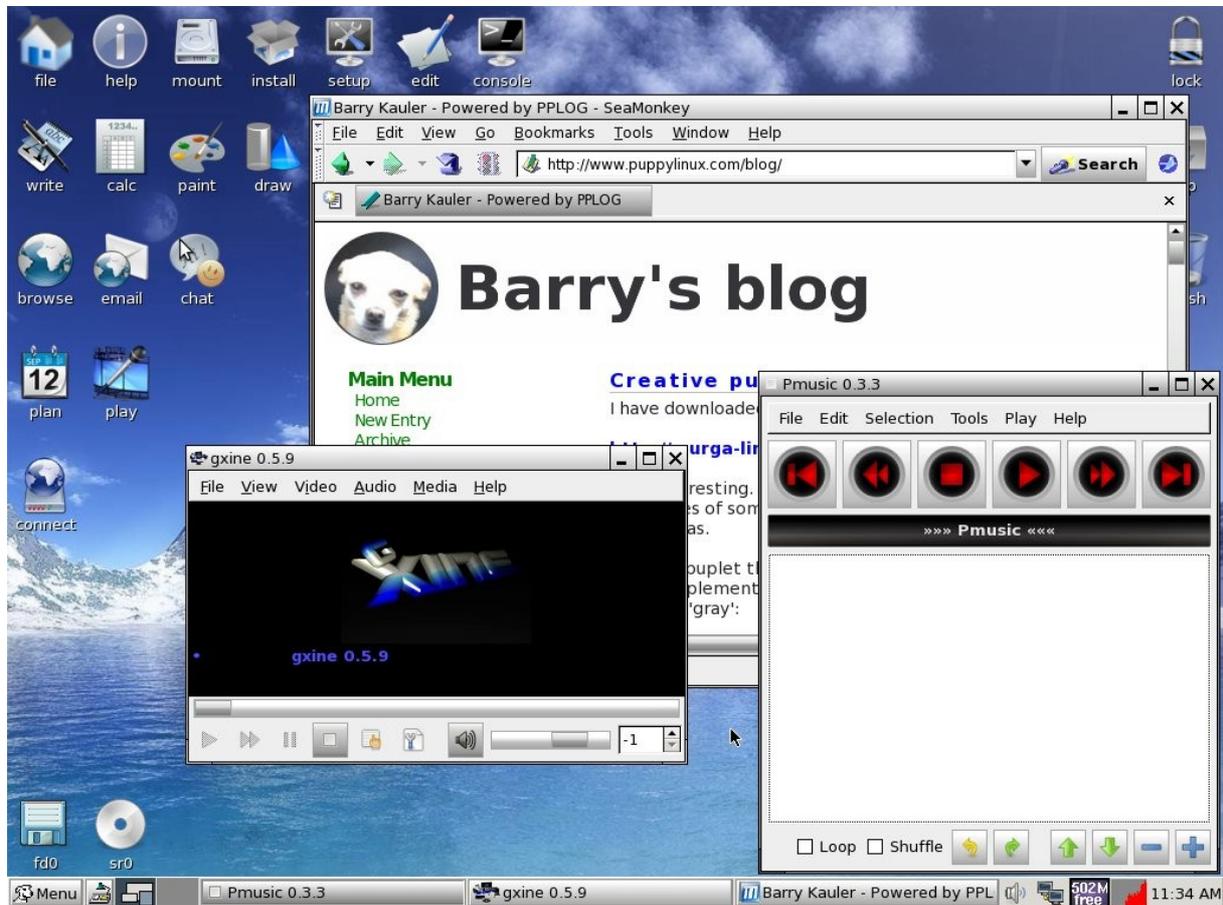


Una distribuzione derivata dal remastering di Knoppix 3 e quindi basata su Debian, dotata di uno spettacolare kernel 2.4 ricompilato meravigliosamente anche sfruttando la compatibilità hardware di Knoppix, in modo da funzionare perfettamente su sistemi che vanno dai più nuovo a quelli ormai più che sorpassati.

DSL utilizza l'abbinamento dei leggerissimi window-manager Fluxbox e JWM uniti ad una serie decisamente ampia di leggerissime applicazioni basate sulla prima versione di Gtk, oltre alla possibilità di installazione di software con l'utilizzo dei repository di Debian Woody e Sarge.

La possibilità di utilizzare pacchetti di Etch e superiori (come sempre senza possibilità di upgrade) deve essere valutata con cura, per il fatto che porta il sistema a crescere notevolmente di dimensioni a causa delle dipendenze necessarie, una cosa apparentemente influente per i canoni odierni, però fondamentale per macchine molto vecchie in cui la capacità del disco è ridotta a pochi GB.

Puppy Linux: <http://www.puppylinux.org>



Distro derivata dalla sempre ottima Slackware, a differenza della precedente utilizza un kernel 2.6 e JWM come DE.

Leggermente più voluminosa di DSL (siamo sui 70 MB), comprende un proprio gestore di pacchetti molto comodo ed estremamente ben fornito.

Con qualche risultato in meno dal punto di vista della gestione hardware, offre decisamente un aspetto ed una modernità superiore.

Un grosso problema dell'installazione di GNU/Linux su macchine molto vecchie è la mancanza del boot da cd-rom, a cui è necessario ovviare tramite l'avvio da floppy-disk.

Alcune distribuzioni permettono ancora il download dei dischetti necessari all'installazione, che risultano comunque reperibili per le vecchie versioni.

Chi ha qualche anno in più del dovuto ricorderà l'utilizzo di *loadlin.exe* per Windows 9x che rendeva possibile il boot di GNU/Linux su sistemi Windows utilizzando "l'avvio in modalità MS-DOS".

Dopo che mi è stata rinfrescata in memoria anche a me, credo sia bene spiegare la tecnica

generale di funzionamento, partendo con l'occorrente:

- un sistema Windows 9x avviabile in modalità MS-DOS
- un dischetto floppy (ho visto che a casa ne ho uno...incredibile)
- il kernel della distro che vogliamo installare (lo chiameremo *vmlinux*)
- l'immagine dello stesso kernel (la chiameremo *initrd.img*)
- il file *loadlin.exe*, ancora facilmente rintracciabile in rete

Dopo aver posizionato i tre file, riavvieremo il sistema sotto DOS e ci porteremo nella directory (anzi...cartella xD) dove risiedono i file; quindi daremo il comando:

```
> loadlin vmlinux initrd=initrd.img
```

per creare il nostro floppy di avvio.

In ogni caso, è consigliato seguire le guide relative all'installazione della distro scelta.

Un esempio piuttosto interessante è offerto nel sito di Damn Small Linux, che comunque dispone sia dei floppy di avvio, sia della versione .iso con avvio da Syslinux.

Questo metodo prevede l'abbinamento di due meravigliose distro: la stessa DSL e *tomsrtbt*

Per chi non lo sapesse, *tomsrtbt* (*TOM'S floppy which has a RooT filesystem and is also BooTable*) è una minidistro Linux su kernel 2.4 che sta in un floppy e che si adatta proprio a lavorare per il recupero di macchine decisamente vetuste.

Questo piccolo gioiello è scaricabile dal sito dell'autore: <http://www.toms.net>

e permette la creazione di floppy di avvio sia sotto Windows 9x tramite *loadlin.exe*, sia sotto GNU/Linux.

Una volta avviato il pc con una connessione ethernet, DSL propone il suo script *frugal.sh*

```
#!/bin/ash
# (C) 2004 Robert Shingledecker <robert@damnsmaillinux.org>

URL="ibiblio.org/pub/Linux/distributions/damnsmaillinux/current"
PROTOCOL="ftp"

clear
echo "DSL poorman's Install and Boot Floppy via Net"
echo
echo "No responsibility for data loss or hardware damage."
echo
echo "You must have created and formatted two empty and unmounted Linux
partitions."
echo "1. One large enough to hold the downloaded iso."
echo "2. The other large enough for poorman's install."
echo
echo "You must also have a no bad sectors unmounted floppy inserted into
drive."
echo
echo -n "Are you ready to being (y/..): "
```

```
read ANS
if test "$ANS" != "y"; then
    exit 1
fi
echo -n "Enter the partition to hold the iso (eg: hda1): "
read SOURCE
if test "$SOURCE"; then; else
    echo "no source partition entered."
    exit 1
fi
echo -n "Enter the partition to install into (eg: hda2): "
read TARGET
if test "$TARGET"; then; else
    echo "no target partition entered."
    exit 1
fi
mke2fs /dev/$SOURCE
mkdir /mnt/$SOURCE
mount -t ext2 /dev/$SOURCE /mnt/$SOURCE
cd /mnt/$SOURCE
echo
echo "Standby fetching the iso..."
wget "$PROTOCOL://$URL/current.iso"
if test "$?" -eq 1; then
    echo "Could not get iso file at this time."
    echo "Try again later."
    exit 1
fi
mke2fs /dev/$TARGET
mkdir /mnt/$TARGET
mount -t ext2 /dev/$TARGET /mnt/$TARGET
mkdir /mnt/iso
mount /mnt/$SOURCE/current.iso /mnt/iso -t iso9660 -o loop=/dev/loop0
echo
echo "Installing the compressed image..."
cp -r /mnt/iso/KNOPPIX /mnt/$TARGET
cp -r /mnt/iso/boot /mnt/$TARGET
rm -f "/mnt/$SOURCE/current.iso"
echo
echo "Creating the boot floppy..."
wget "$PROTOCOL://$URL/bootfloppy.img"
if test "$?" -eq 1; then
    echo "Error trying to get the boot floppy."
    echo "Try again later."
    exit 1
fi
dd if=bootfloppy.img of=/dev/fd0
if test "$?" -eq 1; then
    echo "Error trying to create the boot floppy."
    echo "Try again later."
    exit 1
fi
```

```
rm -f "bootfloppy.img"
echo "DSL installation complete."
rm -f "/mnt/$SOURCE/current.iso"
echo "Standby for reboot..."
reboot
```

versione originale scaricabile da:

<ftp://ibiblio.org/pub/Linux/distributions/damnsmall/current>

Come si vede, lo script richiede la presenza di due partizioni; una per il download del .iso (di circa 50 MB, che potrà poi essere eliminata) e un'altra per l'installazione del sistema.

Tutto il necessario per verrà direttamente scaricato da internet e quindi trattato tramite le utility presenti in *tomsrftb*.

Il tutto rende questo piccolo script molto simile al *net-install* di “mamma Debian”.

Il software: cosa non vede il linux-user

Quando mi sono addentrato un minimo nel mondo del trashware per scrivere questo articolo, sono rimasto decisamente impressionato e mi sono posto parecchi quesiti su come noi utenti GNU/Linux gestiamo le nostre macchine.

Chi utilizza Windows (o almeno chi lo usa bene) è decisamente abituato a porsi problemi relativi alla “pesantezza” del software usato, alla gestione del file-system e alla manutenzione del proprio sistema.

Sotto GNU/Linux molti di questi problemi sono del tutto sconosciuti, mentre altri risultano piuttosto irrilevanti.

L'idea, forse errata e puramente personale, che mi sono fatto è che la superiore gestione delle risorse di una macchina sotto Linux siano talmente date per scontate che non ci si pone mai il problema del reale consumo del pc.

Nel mio caso, vi sto scrivendo da un laptop modello HP Compaq nx6110 con 512 MB di RAM in dual-boot tra Debian testing (Squeeze) e Windows XP; mentre il sistema di casa Redmond gira discretamente, probabilmente Vista anche in versione “minimale” farebbe parecchi capricci.

Alla fine secondo i moderni canoni di una macchina comune il mio computer inizia ad invecchiare, anche se con la mia Debian ottengo performance decisamente superiori ad un pc con Vista e i suoi 2 GB di RAM, anche utilizzando Compiz-Fusion, che non uso ma ho perfettamente installato e configurato.

Nel momento in cui nell'ambito del trashware si inizia a lavorare su sistemi decisamente di frontiera rispetto gli standard di oggi il controllo del consumo di risorse è un'attività essenziale.

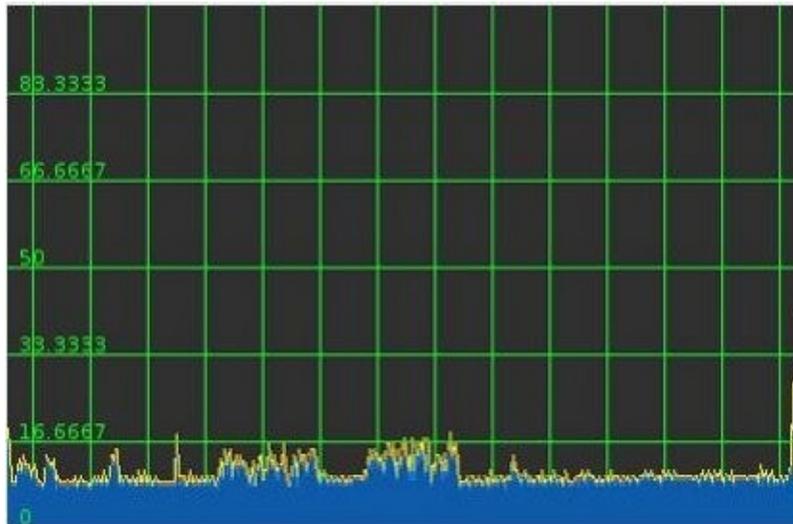
I seguenti test sono stati compiuti sul mio pc, con un sistema non all'ultimo grido ma comunque non vetusto e con il mio KDE 3.5

Se qualcuno volesse obiettare che KDE 4 rende il pc più lento, sono prontissimo a

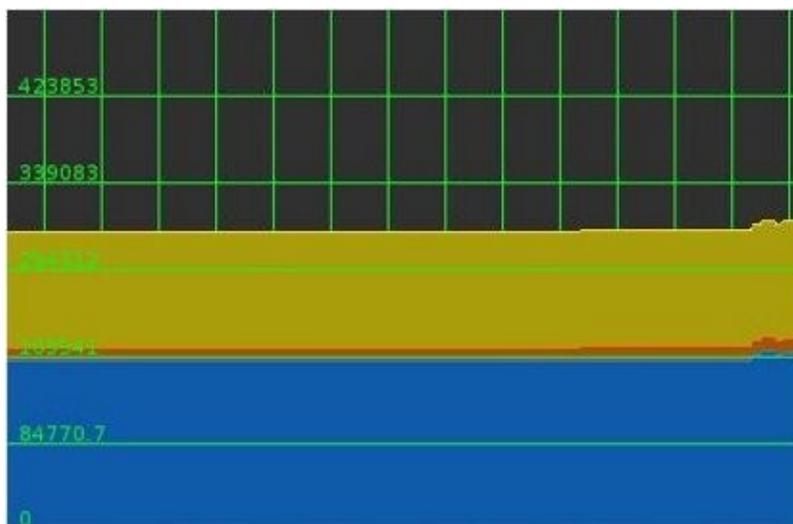
rispondere, facendo solo presente che l'ho montato un po' di tempo fa in un vecchio Asus L2000 da 256 MB di RAM con risultati nettamente invidiabili rispetto a Windows XP.

Per fare i seguenti confronti utilizzerò i dati che mi sono forniti da Ksysguard sui consumi di memoria fisica delle applicazioni testate.

Carico alla CPU



Carico di memoria



come si vede la memoria resta un po' sotto 300 MB, quindi siamo attorno al 55-60% di RAM, con swap nulla. Considerando il fatto che ho parecchi demoni di stampa che tengo attivi perchè mi servono il risultato è discreto.

Prima di iniziare facciamo qualche piccola premessa importante per comprendere il test.

In primo luogo è necessario focalizzarsi sui rapporti di consumo tra le varie applicazioni e non sui loro valori assoluti; cioè se da me un processo consuma 10 MB di memoria non è detto

che in un'altra macchina produca gli stessi valori, per l'efficienza dell'hardware, la gestione dei quanti di memoria ecc.

A questo punto passo dalla schermata di Ksysguard con i grafici dell'utilizzo a quella con i dati dei singoli processi; da qui in avanti inizierò a testare i consumi di varie applicazioni. I risultati come vedrete sono piuttosto sbalorditivi...

Come immaginavo *Kwrite* (l'editor da dove sto scrivendo) riserva per sè ben 18.880 KB, una cifra ragguardevole che vado confrontare con gli altri editor standard di KDE.

Il potente *Kate* consuma i suoi 21.956 KB, il 15% in più del fratello minore.

Il piccolo *Kedit* si ferma a 13.976 KB, cioè il 15% in meno di *Kwrite*.

Una scaletta piuttosto interessante direi...

A questo punto andiamo sul pesante, anzi sul leggero; apro *Nano* e faccio fatica a trovarlo, sono costretto a mettere i processi in ordine alfabetico...segna 1.436 KB, cioè 13 volte meno!

Passiamo a qualcos'altro...

Sicuramente tra i software di un qualunque sistema, uno dei più importanti è il file-manager, infatti sono proprio questi programmi che tolgono all'attività da terminale il maggior carico di lavoro.

Figuriamoci se uno come me, amante di KDE, non apprezza il suo *Konqueror*, un fantastico software tutto fare decisamente parsimonioso di risorse.

Infatti, dati alla mano, decisamente non sfigura con i suoi 23.204 KB di memoria fisica consumata.

Un'altro file manager che apprezzo molto è *PCManFM*, che so di per certo essere più leggero di *Konqueror* pur facendo molto bene il suo lavoro.

Vado subito a testare il suo consumo: 17.300 KB, cioè ben il 25% in meno, mica poco.

Rimanendo sempre nell'ambito dei file manager, vorrei fare un esempio piuttosto evidente che faccia capire bene di che problema stiamo parlando.

Nei casi fatti fino ad ora sono state confrontate applicazioni diverse tra loro; si potrebbe far notare che è facile per *Floatman* truccare il test confrontando *Kwrite* con *Nano*...

Un file manager molto simpatico che ho scoperto da poco è *GnomeCommander*, con un'interfaccia a doppia scheda stile mc ma una grafica in Gtk molto piacevole.

Nella sua interfaccia piuttosto curata ricorda *MidnightCommander* e darebbe quindi l'idea di non essere eccessivo nei consumi...in realtà viaggia a 23.852 KB appena *sopra* *Konqueror*.

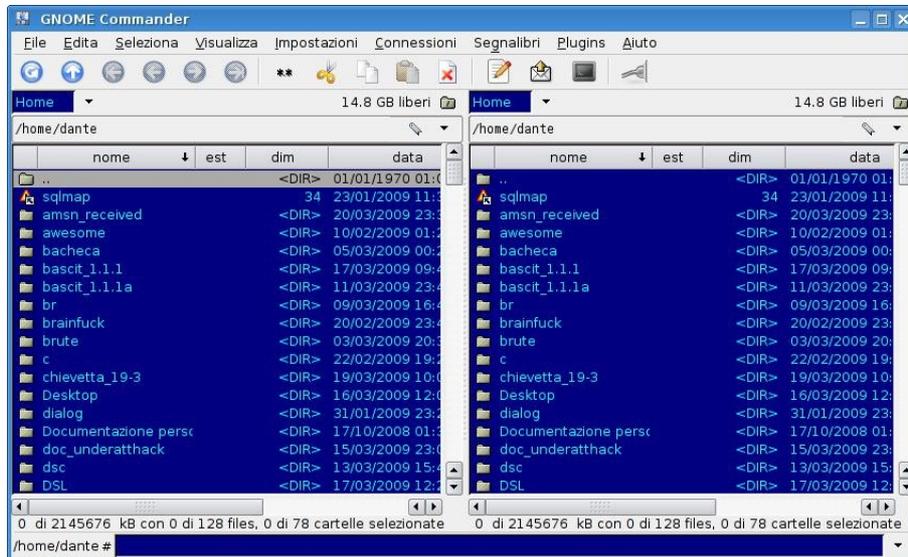
Anche *TuxCommander* è un bel file-manager a doppia scheda a modello mc, la grafica è un po' meno curata di quella di *GnomeCommander* ma le funzioni sono le stesse.

Al test il programma si ferma a 19.100 KB, cioè il 18% in meno di *Konqueror* e addirittura il 20% in meno del suo quasi-gemello *GnomeCommander*.

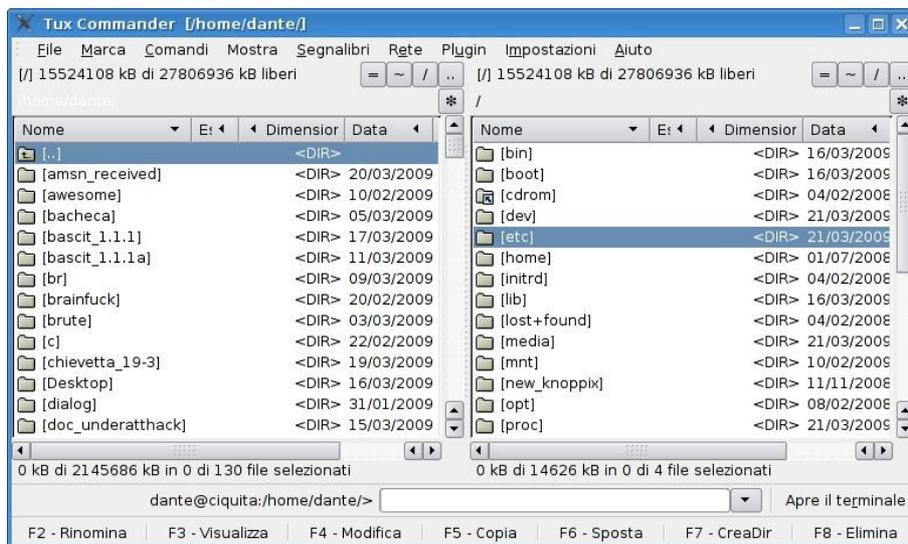
La cosa non sembrerebbe particolarmente degna di nota, se non fosse per il fatto che quando uso l'espressione "quasi-gemello" non lo faccio per esagerare i valori del risultato, e per far vedere che non sto forzando il confronto vi presento qui di seguito le schermate dei due

programmi:

Gnome Commander



TuxCommander



Siete convinti adesso? Uno sfondo blu vale il 20% di consumo in più? Secondo me no...

Altro file manager piuttosto utilizzato per la sua leggerezza è *ROX*, che infatti mi risulta consumare 13.240 KB di memoria, cioè ben il 43% in meno di Konqueror.

Quindi a questo punto quanto consumerà *MidnightCommander*? Metà di Konqueror? Forse addirittura un terzo viste le differenze... a questo punto sono curioso...

Aprò mc, scopro che lo devo cercare come Nano è che consuma 2.212 KB cioè addirittura 10 volte di meno del mio Konqueror!

Immaginavo che potesse essere decisamente leggero, però confrontare i dati in modo preciso

mette di fronte a realtà che vanno ben oltre il pensiero comune.

Il pensiero di avere uno schermo pieno di sessioni di mc per ottenere il consumo di una finestra di Konqueror mi lascia piuttosto sconcertato.

Proviamo ad andare a fondo del sistema, andiamo cioè a verificare il consumo di un terminale.

La componente grafica in questo caso “dovrebbe” essere abbastanza ininfluente e le percentuali non si dovrebbero scostare molto tra i vari emulatori.

Abbastanza convinto di quello che ho detto sopra, vado subito al confronto Davide e Golia:

Per prima cosa apro *x-term* e verifico i suoi 3.488 KB, tutto sommato nemmeno poco a dire il vero...

Dopo essermi segnato i consumi, vado ad aprire *Konsole* e scopro che mi occupa una quantità di 13.892 KB cioè 4 volte tanto; per un semplice terminale!

Come ho detto all'inizio, il mio pc è appesantito da demoni di stampa che possono essere disabilitati dall'utente comune, primo tra tutti il demone GhostScript.

Per curiosità personale vado a vedere i consumi di alcuni visualizzatori di .pdf, cosa che può comunque essere utile anche all'utente comune visto che oggi il formato .pdf risulta abbastanza utilizzato.

Vado quindi ad aprire il num. 0 di UnderAttHack con il veloce e leggero *Xpdf* ottenendo un consumo segnalato di 10.924 KB; alla luce dei precedenti test un risultato decisamente ottimo confrontato ad esempio con *Kwrite* nella sua trattazione di file di testo.

Apro lo stesso file con il visualizzatore di KDE, *Kpdf* (come altro si poteva chiamare...) e come previsto ottengo un consumo di 17.016 KB del 36% più elevato per svolgere la stessa funzione...mi pare che la cosa debba far riflettere parecchio.

Io comunque ho un utilizzo professionale di questo tipo di file, quindi non posso non avere installato il mio originale *Adobe Reader*, con cui è sicuramente il caso che faccia il test.

All'apertura del file ottengo “leggere differenze”, cioè un consumo di 44.644 KB...più di 2 volte e mezzo rispetto a *Kpdf*, e ben 4 volte superiore alla visualizzazione con *Xpdf*.

Rimaniamo su applicazioni ad alto consumo di memoria, e rimaniamo vicini ai .pdf e ad UnderAttHack.

Se in questo momento state leggendo al versione .pdf dell'e-zine, potrete immaginare come questa sia realizzata a partire da un file .odt di Open Office per poi essere convertita.

Un'altro programma abbastanza utilizzato per lavorare su file .odt è *Abiword*, che potrebbe fare benissimo lo stesso lavoro.

Aprendo il documento che sto scrivendo in questo momento tramite *Abiword* ottengo un consumo di 46.876 KB...e la cosa mi porta a pensare che forse sto facendo un buon articolo.

Apro lo stesso documento da *Open Office Writer* ottenendo un valore di 71.012 KB; un valore superiore del 34% a quello di *Abiword*...ma un articolo ancora migliore.

A questo punto non potevo che fare un confronto tra i browser più usati.

Il fatto che *Firefox* non sia un programma tra i più leggeri è noto a tutti, però quello che ci

chiediamo è: che distanza c'è tra il browser più usato su GNU/Linux ed altre applicazioni che svolgono la stessa funzione?

Nel caso di Firefox i consumi variano notevolmente in base al numero di add-on utilizzate, alla cache, alle impostazioni ecc.

Per fare questo test guarderemo solo la semplice apertura del programma, disconnessi da internet e con tutti gli add-on disabilitati esclusa la lingua Italiana.

Il risultato che otteniamo è un consumo di 40.140 KB, risultato decisamente elevato che però (credo) ci aspettassimo tutti.

Il browser che credo sia l'unico concorrente di Firefox credo sia *Epiphany*, il browser "ufficiale" di Gnome, anch'esso basato su Geko che segnala un consumo di 37.184 KB, appena inferiore a quello di Firefox.

Un altro browser su Geko, conosciuto come versione light di Firefox è *Kazehakase* che con 24.876 KB risparmia un buon 40% di memoria...non male direi.

Nel confronto ho pensato di testare anche due browser basati su webkit anziché su geko, ovviamente diversi da Konqueror.

Ho quindi testato *Netsurf* con i suoi 20.212 KB e *Arora* con 18.900 KB, entrambi quindi intorno al 50% di memoria occupata. E' anche possibile che il mio utilizzo di KDE possa incidere in qualche modo su questi valori, però non credo sia tale da giustificare il doppio dei valori.

Il confronto con i browser testuali è ovviamente inutile, però ho pensato che sarebbe stato interessante confrontare Firefox con un mini-browser minimalista ma grafico come *Links2*, che sebbene non possa essere paragonato direttamente, può comunque darci un'idea dei valori di scarto raggiungibili.

Il risultato ottenuto è di 3.664 KB, vale a dire 11 volte in meno del browser di casa Mozilla; i numeri credo non richiedano commenti.

Per concludere in bellezza e togliermi lo sfizio, voglio vedere quanto consuma Firefox con i vari Torbutton, PDFDownloader, GetRight, switch-user-agent e la mia serie di add-on accumulati nel tempo e per la maggior parte totalmente inutili...

Alla semplice apertura del browser, raggiunge un valore di 52.096 KB; quindi quando si dice che gli add-on appesantiscono Firefox non si parla decisamente a caso.

Credo che si possano concludere qui i nostri confronti, fatti su hardware decisamente moderno rispetto ai canoni del trashware, su pochi programmi oltretutto soltanto tra quelli dei repository attuali di Debian Testing.

Credo che quello che si pretendeva di spiegare sia stato compreso a dovere.

Conclusioni

Essendo questa una rivista dedicata all'hacking, per concludere è bene chiedersi cosa si è voluto dimostrare in questo documento.

Nelle poche pagine qui esposte, non era certo possibile spiegare tutto ciò che è necessario per operazioni di trashware, per le quali come si è visto non è possibile dare un metodo generale

preciso per la varietà delle macchine esistenti, delle modalità di azione e delle caratteristiche finali richieste al pc da trasformare.

Le due considerazioni sono una di ordine tecnico e una di tipo etico.

Dal punto di vista tecnico il trashware non dimostra semplicemente le potenzialità del software libero. Queste tecniche di progettazione di un sistema così minuziose ed elaborate, mostrano il significato della tanto decantata configurabilità di GNU/Linux.

La possibilità di prendere in mano una macchina data per spacciata secondo i parametri dei sistemi commerciali, analizzarne i dati hardware in ogni sua componente, elaborare un pacchetto completo di software intrecciati tra loro per dipendenze reciproche, quindi portarla ad un nuovo modello utilizzabile è un processo decisamente affascinante.

La quantità enorme di software libero disponibile, che qui abbiamo appena esemplificato con applicazioni ben visibili all'utente, ci permette veramente di superare limiti che sembrano imposti dalla natura stessa della base su cui si lavora.

La seconda considerazione che mi pongo è di tipo etico ed economico.

La diffusione del mezzo informatico ad ogni livello, se da un lato ha creato il fenomeno "utonto" dall'altra parte ha comunque permesso di ampliare la fascia di persone che è in grado di usufruire delle potenzialità ancora non ben conosciute di questa tecnologia.

Viene da chiedersi come procede l'idea di *progresso* dopo aver raggiunto questi risultati.

In una situazione dove praticamente in ogni casa esiste almeno un computer, non esiste un certo *dovere sociale* a far comprendere all'utenza come utilizzare nel modo migliore quelle macchine?

Riprendendo l'esempio fatto sopra dei due file-manager praticamente identici, ma con consumi nettamente differenti, mi chiedo se sia *progresso* sostituire macchine perfettamente funzionanti per il fatto che ad esempio non consentono le finestre trasparenti (soprattutto se in realtà le possono reggere senza problemi...).

Credo che le logiche strettamente commerciali del settore informatico attuale siano nettamente fallimentari e nel futuro prima o poi invertiranno quello che oggi è il gap tecnologico mondiale; prima o poi qualcuno tornerà a scoprire come il mezzo informatico sia fonte di progresso e non un volto commerciale dello stesso.

Come sempre, per imparare abbiamo internet; io spero solo di aver fatto riflettere il lettore.

Floatman

Fluxbox su Ubuntu...as I like

Per chi si avvicina al mondo di GNU/Linux negli ultimi anni, per necessità, per curiosità, per sentito dire e per cubi rotanti, Ubuntu è un must...

Il Linux for human being è uno degli OS più conosciuti e da qualche mese, sta riscuotendo un successo micidiale, che ne premia la stabilità e l'usabilità.

Tra coloro che si immergono nel pianeta GNU/Linux grazie a questo OS, si possono riconoscere due filoni principali: i cubisti-compizzisti-gnomisti e i minimalisti vecchia maniera...

Se i primi sfruttano circa 500 megabyte di RAM a sessione, i secondi preferiscono avere un DE minimale, lasciando libere (forse inutilmente) grandi quantità di memoria, e ottenendo un sistema forse non troppo human-friendly, ma di sicuro geek-friendly.

Personalmente, preso dall'avvento di Beryl prima e di Compiz-Fusion in seguito, sono stato a lungo un esponente della corrente di pensiero cuborotantistica, in seguito vi posso assicurare che ho preferito molto avere il controllo di come funzionava il tutto da dentro.

Soprattutto nel momento in cui installando la cara vecchia Slackware (distro decisamente meno user-friendly) in un pc non troppo potente ho avuto poca scelta, il DE che ho selezionato e provato allora è stato Fluxbox, di cui mi sono innamorato subito vista la sua incredibile leggerezza e la sua impressionante semplicità di funzionamento.

Così tornando al mio notebook-Ubuntu...e facendo roteare il mio cubo mi sono detto....proviamo a fare una Fluxubuntu!

A dire la verità tra le centinaia di remaster di Ubuntu esiste già una Fluxbuntu, però secondo me un DE come Fluxbox è bene configurarlo da soli. Tra l'altro anche se installassi Fluxbuntu dovrei comunque riprenderla in mano da capo.

Insomma, chi fa da se fa per tre, quindi lascio Fluxbuntu a chi ha un pc che non regge un'installazione su Gnome, motivo per cui oltretutto nasce quella distro.

Comunque veniamo al punto...

Configurare Fluxbox su Ubuntu:

Siete stanchi del zuccheroso Gnome?

Del fastidiosamente bello Compiz-Fusion?

Volete capire come funziona un DE semplice e leggero?

O avete semplicemente voglia di smanettare fino a scolorire i tasti della vostra keyboard?

Se la risposta ad una di queste domande (o anche a tutte xD) è sì, questo articolo-guida fa per voi...

Parte I

Installare Fluxbox

Come installiamo Fluxbox? semplicissimo, avete un grandissimo package-manager, Synaptic, ma avete soprattutto la potenza della shell, quindi digitate in console:

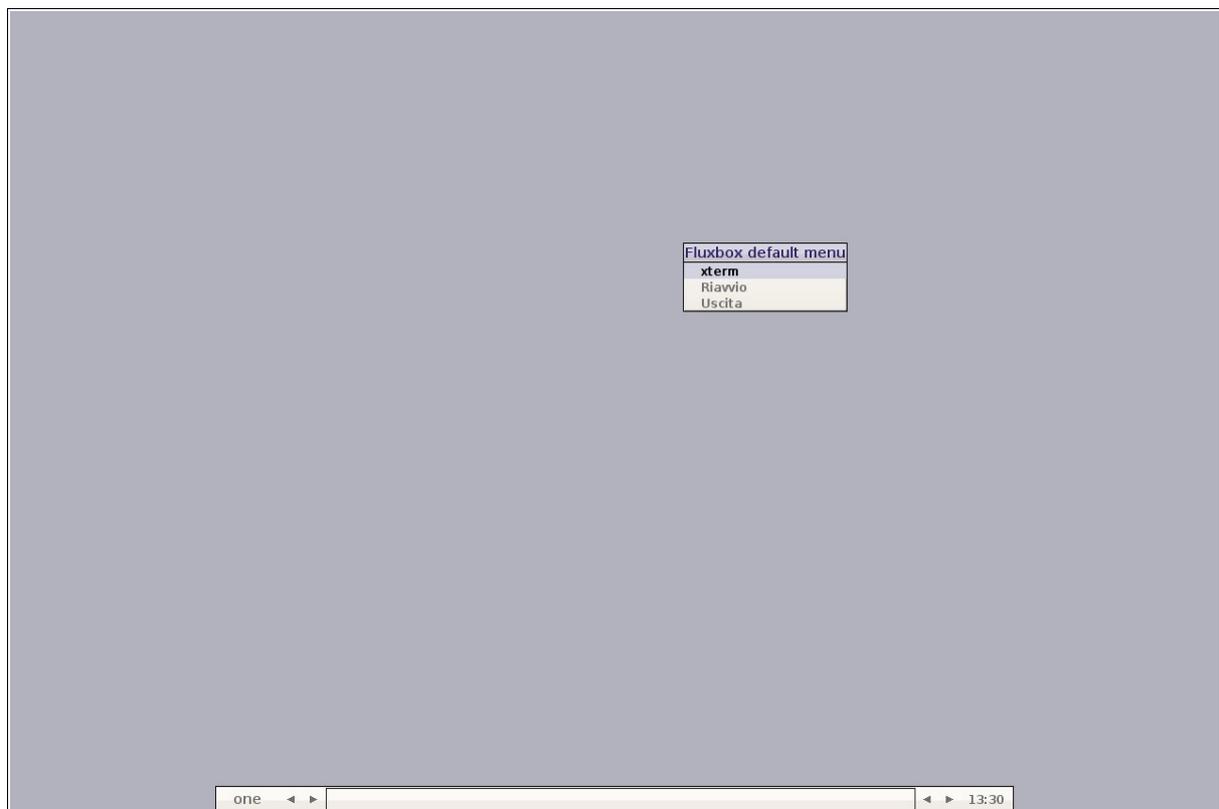
```
$ sudo apt-get install fluxbox
```

quando aptitude avrà finito di installare il leggerissimo DE, sarete pronti a smanettare come i pazzi, ma non prima di aver riavviato GDM....

Premete Control+Alt+Backspace e aspettate...ecco come per magia la schermata che tante volte avete visto per accedere al solito Gnome...ma prima di digitare la vostra *qwerty* nel campo password, selezionate dall'opportuno menu (dipende dal tema della schermata di login), la sessione Fluxbox... ora digitate username e password e...

Parte II

...Ecco Fluxbox!....ma...ma...che schifo!



Si...fa un po' schifo per dirla tutta...blu? Verde? Bianca? che colore la avete?...è spoglia, silenziosa e cattiva...sta ferma e vi guarda... aspettando che siate voi a farvi avanti. Il primo passo tocca a voi!

Nota: Problemi all'installazione

Ho riscontrato che molti segnalano problemi durante il primo avvio di Fluxbox ed in particolare questo errore:

chmod: impossibile accedere a /home/lost/.fluxbox/

il fix per questo problemino è accedere con Gnome (ancora una volta xD) e da terminale digitare:

```
$ chmod 755 /home/lost/*
```

per impostare i permessi necessari al vostro utente, in maniera da accedere ai file di configurazione di Fluxbox della vostra /home.

Riavviate GDM come ormai sapete fare...e riaccedete a sessione Fluxbox.

Se il problema dovesse ripresentarsi, dovete accedere nuovamente da Gnome e cominciare ad editare i file di configurazione di Fluxbox da Gnome stesso.

Dopo aver effettuato l'accesso, se cliccate col tasto sinistro su qualsiasi punto del desktop apparirà il menu della Fluxbox (come nello screen sopra).

Se sarete fortunati si sarà generato automaticamente e avrà già dentro molti dei programmi che usavate in Gnome... diciamo con criteri di categorizzazione un po' difficili da capire: doppiatori di lanciatori in menu diversi... programmi di cui nemmeno avevate sentito parlare e che invece erano installati e lanciabili sul vostro sistema operativo: jeditra?? W3m!? Eog? gmix?! programmi che magari usavate e nemmeno sapevate come si chiamavano come Nm-applet, che sarebbe l'utility di gestione delle connessioni di rete in Gnome (l'icona dei due monitor in alto a destra, per farvi capire).

Se invece sarete sfortunati il menu sarà spoglio, vuoto... e sarà quello di default della Fluxbox, con i tasti del menu: xterm, Riavvia, Spegni .

Insomma, in entrambi i casi siete su un desktop minimale finalmente, brutti smanettoni!...

Però minimale purtroppo equivale ad inutilizzabile nel vostro caso.

Che fare? Mano agli editor di testo e giù via a smanettare con le configurazioni di Fluxbox.

Tutti i file di configurazione si trovano nella cartella `~/.fluxbox/`

Analizziamoli:

~/.fluxbox/menu

Questo file vi consente di editare il menu che vedete apparire quando cliccate col tasto sinistro del mouse sul desktop.

Di default dovrebbe essere composto con l'inclusione di un file esterno, esattamente il file di configurazione di default del menudi Fluxbox, contenuto in qualche sperduta parte della cartella */etc/*...* ma voi non volete nulla di default, non ora che si può smanettare!

Via con gli editor di testo...

Aprite con qualsiasi editor di testo il file sopra indicato, cancellate l'inclusione del file esterno e cominciate a scrivere tra i tag *[begin]* e *[end]* la seguente stringa:

```
[begin] (fluxbox)
[exec] (Gnome-Terminal) {/usr/bin/gnome-terminal}
</usr/share/pixmaps/gnome-terminal.xpm>
[end]
```

Cliccate col sinistro sul desktop e vedrete il risultato del vostro lavoro... nel vostro menu troverete l'iconcina del terminale di default di Gnome, provate a cliccarla, che accade?

Fluxbox tramite Eterm lancia: *exec /usr/bin/gnome-terminal*

E vi appare il terminale che avete amato su Gnome!

Ecco, ora il potere è nelle vostre mani!

Spieghiamo un po' la sintassi:

Tutti i tag del menu, ovviamente vanno tra *[begin]* ed *[end]*.

Le cose scritte tra parentesi tonde tipo: *(fluxbox)*; *(gnome-Terminal)*, sono i *label* con cui vedrete scritti nel menu i comandi da eseguire.

Questi sono inseriti tra parentesi graffe “{}” (Potete anche mettere solo il comando invece di inserire tutto il percorso, ma è più pulito farlo con il percorso completo).

Infine tra le parentesi acute “<” ci vanno le iconcine; fatevi un giro nella cartella */usr/share/pixmaps* per vedere le icone che potete inserire nei vari comandi, le iconcine devono essere in formato .xpm o in .png, dimensioni 28x28px.

Continuando ad editare il file menu, vi consiglio vivamente di inserire i seguenti comandi di base di Fluxbox, utili per lasciare la sessioneX, restartarla, ed uscire a gdm:

```
[begin] (fluxbox)
[exec] (gnome-Terminal) {/usr/bin/gnome-terminal}
</usr/share/pixmaps/Gnome-terminal.xpm>
[config] (Configuration)
[submenu] (Style) {}
    [stylesdir] (/usr/share/fluxbox/styles)
    [stylesdir] (~/.fluxbox/styles)
[end]
[workspaces] (Workspaces)
[reconfig] (Riconfigure)
```

```
[restart] (Restart)
[exit] (Exit)
[end]
```

Potete editare i label in italiano se volete, cosa molto elegante.

Vi consiglio altresì di provare qualche style diverso nel submenu “*Style*” che avete appena incollato, vedrete quanti stili sono disponibili, ancora di più ne potete scaricare ad esempio dal sito ufficiale di Fluxbox, e oltretutto ne potrete realizzare voi stessi, essendo (come quasi tutto in Fluxbox e in GNU/Linux) semplici file di testo.

Osserviamo però la comparsa di un'altro tag: `[submenu] [end]`; tra questo tag (sempre assegnando un *label*), possiamo ottenere un sottomenu a cascata, ottimi per separare le categorie delle applicazioni che vorrete inserire nel menu. Per esempio, la categoria “internet” nel mio `[submenu] (Internet)` è molto interessante per gli utenti di Ubuntu:

[...]

```
[submenu] (Internet) {}
[submenu] (IM e Comunicazione) {}
[exec] (Pidgin) {/usr/bin/pidgin} </usr/share/pixmaps/pidgin-
menu.xpm>
[exec] (X-Chat Gnome) {/usr/bin/xchat-Gnome} <>
[exec] (emesene) {emesene} </usr/share/pixmaps/emesene.xpm>
[exec] (Terminal Server Client) {/usr/bin/tsclient -f}
</usr/share/pixmaps/tsclient.xpm>
[end]
[submenu] (Mail) {}
[exec] (Evolution) {/usr/bin/evolution}
</usr/share/pixmaps/evolution.xpm>
[end]
[submenu] (Ftp) {}
[exec] (gFTP) {/usr/bin/gftp-gtk} </usr/share/pixmaps/gftp.xpm>
[end]
[submenu] (P2p) {}
[exec] (Transmission) {/usr/bin/transmission}
</usr/share/pixmaps/transmission.xpm>
[end]
[submenu] (Monitoraggio) {}
[exec] (Firestarter) {gksu -g /usr/sbin/firestarter}
</usr/share/pixmaps/firestarter.xpm>
[exec] (Wireshark) {/usr/bin/wireshark}
</usr/share/pixmaps/wsicon32.xpm>
[end]
[end]
```

[...]

Come vedete ho diviso il sottomenu, in ulteriori sottomenu che mi hanno migliorato notevolmente la qualità della ricerca di un'applicazione di cui ho bisogno.

A sua volta ho fatto anche dei sottomenu per Open Office, e per la programmazione:

```
[submenu] (Office) {}
[exec] (OpenOffice Base) {/usr/bin/oobase}
</usr/share/icons/Gnome/32x32/apps/openofficeorg24-base.xpm>
[exec] (OpenOffice.org Calc) {/usr/bin/oocalc}
</usr/share/icons/Gnome/32x32/apps/openofficeorg24-calc.xpm>
[exec] (OpenOffice.org Impress) {/usr/bin/ooimpress}
</usr/share/icons/Gnome/32x32/apps/openofficeorg24-impress.xpm>
[exec] (OpenOffice.org Writer) {/usr/bin/oowriter}
</usr/share/icons/Gnome/32x32/apps/openofficeorg24-writer.xpm>
[exec] (OpenOffice.org Draw) {/usr/bin/oodraw}
</usr/share/icons/Gnome/32x32/apps/openofficeorg24-draw.xpm>
[exec] (OpenOffice.org Math) {/usr/bin/oomath}
</usr/share/icons/Gnome/32x32/apps/openofficeorg24-math.xpm>
[end]
```

```
[submenu] (Programmazione) {}
[exec] (DDD Debugger) {/usr/bin/ddd} <>
[exec] (Eclipse) {/usr/bin/eclipse}
</usr/share/pixmaps/eclipse32.xpm>
[exec] (Gambas2) {/usr/bin/gambas2.gambas}
</usr/share/pixmaps/gambas2.xpm>
[exec] (Python (v2.5\)) { x-terminal-emulator -T "Python (v2.5)"
-e /usr/bin/python2.5} </usr/share/pixmaps/python2.5.xpm>
[end]
```

Per Python, come per tutte le shell e i programmi che funzionano nel terminale è interessante osservare la sintassi:

```
{ x-terminal-emulator -T "Python (v2.5)" -e /usr/bin/python2.5 }
```

La Fluxbox avvia l'`x-terminal-emulator` con il titolo (`-T`) "Python (v2.5)" ed eseguendo (`-e`) il bin di pythonshell.

Potete prendere esempio da questa sintassi se per esempio volete un lanciatore per Elinks, Lynx, w3m, sirc...e altri programmi basati su ncurses e che funzionano sul terminale.

Per i soliti nostalgici delle funzionalità di Gnome, e per la gestione del sistema ho creato un piccolo submenu molto interessante, il submenu Sistema:

```
[submenu] (Sistema) {}
[exec] (Synaptic Package Manager) {/usr/bin/gksu /usr/sbin/synaptic}
</usr/share/synaptic/pixmaps/synaptic_32x32.xpm>
[exec] (vumeter (Gnome 2.0 Volume Meter\)) {/usr/bin/vumeter} </usr/
```

```

share/pixmaps/Gnome-vumeter.xpm>
[submenu] (Gnome System vari) {}
[exec] (Pannello di controllo) {/usr/bin/Gnome-control-center}
</usr/share/pixmaps/control-center2.xpm>
[exec] (Network Tool) {/usr/bin/Gnome-nettool}
</usr/share/pixmaps/Gnome-nettool.xpm>
[exec] (Monitor Processi) {/usr/bin/Gnome-system-monitor} <>
[end]
[submenu] (Hardware) {}
[exec] (Gnome Floppy Formatter) {/usr/bin/gfloppy}
</usr/share/pixmaps/gfloppy.xpm>
[exec] (HPLIP Toolbox) {/usr/bin/hp-toolbox}
</usr/share/pixmaps/HPmenu.xpm>
[exec] (Xvidtune) {xvidtune} <>
[end]
[submenu] (Monitoraggio) {}
[exec] (Gnome Log Viewer) {/usr/bin/Gnome-system-log}
</usr/share/pixmaps/Gnome-system-log.xpm>
[end]
[exec] (Mappa Caratteri) {/usr/bin/gucharmap} <>
[end]

```

Con molti interessanti lanciatori per programmi che tutti utilizzavamo su Gnome: Synaptic, Il Pannello di controllo, il system monitor, e roba simile...enjoy it!

Se premete la voce “*exit*” nel menu, non spegnete il pc, così come se premete “*Restart*”, non riavviate il sistema. Infatti questi comandi si riferiscono esclusivamente alla Fluxbox, non al pc stesso... come ovviare? Semplice, con un submenu!

Ecco le due righe che uso per spegnere e riavviare il pc da Fluxbox:

```

[submenu] (Spegni PC) {}
[exec] (Riavvia) {gksu reboot}
[exec] (Spegni) {gksu halt}
[end]

```

con il gtk based su, inserite la password di root nell'apposito textbox, e il vostro pc si riavvierà o si spegnerà.

Per i più pigri, esistono altri modi per ottenere lo spegnimento/riavvio del sistema senza la richiesta di password.

Quello più “istituzionale” riguarda la modifica del file */etc/sudoers* aggiungendo ad esempio questa riga:

```
%vikkio ALL=(root) NOPASSWD: /sbin/shutdown
```

Cosa è stato fatto?

È stato richiesto al sistema di gestione permessi che tutti gli utenti del gruppo *vikkio* non debbano digitare la password per il comando *shutdown*.

In questo caso il nostro menu per lo spegnimento diventerà questo:

```
[submenu] (Spegni PC) {}
[exec] (Riavvia) {sudo shutdown -r now}
[exec] (Spegni) {sudo shutdown -h now}
[end]
```

Ovviamente faccio notare come questa configurazione, sebbene più comoda, vada a discapito della sicurezza di sistema.

Alla fine della spiegazione sulla configurazione del menu, come esempio vi mostro la mia personalizzazione a lavoro finito:



~/.fluxbox/keys

Il file *Keys*, secondo me, è ancora più interessante ed importante del file menu.

Analizzatelo con un editor di testo...

sarà una cosa del genere:

```
OnDesktop Mouse1 :HideMenus
OnDesktop Mouse2 :WorkspaceMenu
OnDesktop Mouse3 :RootMenu
OnDesktop Mouse4 :NextWorkspace
OnDesktop Mouse5 :PrevWorkspace

Mod1 Tab :NextWindow
Mod1 Shift Tab :PrevWindow
```

```
Mod1 F1 :Workspace 1
Mod1 F2 :Workspace 2
Mod1 F3 :Workspace 3
Mod1 F4 :Workspace 4
[...]
```

Sembra un set di istruzioni in inglese...ed effettivamente lo è!

Le strane scritte prima dei due punti indicano una combinazione di tasti, dopo i due punti invece abbiamo l'azione che quella combinazione di tasti compie sulla nostra Fluxbox.

La prima parte è molto chiara per chi capisce un po' l'inglese:

Se premo sul desktop il tasto1 del mouse (sarebbe il tasto destro), cara Fluxbox, nascondimi i Menu.

Se vi piace potete cambiare ordine dei tasti, se volete che col destro si aprano i menu e col sinistro si chiudano invertite *Mouse1* con *Mouse2*, semplice no?

Consiglio di commentare la quinta e la quarta riga (*OnDesktop Mouse5 :PrevWorkspace*), perchè potrebbe darvi fastidio se usate un touchpad, infatti basterebbe sfiorare leggermente l'area deputata ad emulare la rotellina del mouse per vedervi scomparire l'area di lavoro corrente.

Peccato comunque che le modifiche a questo file non le possiamo vedere in diretta come per il file menu, dobbiamo restartare la Fluxbox, e permettergli di caricare nuovamente il *keyfile*.

La seconda parte tratta della sequenza di tasti della keyboard, essi non devono essere premuti in aree su desktop vuoti, ma dovunque e comunque, osserviamo la prima riga:

```
Mod1 Tab :NextWindow
```

ci ricorda una cosa vista e rivista, Alt+Tab per scorrere tra le finestre attive... indi per cui, si può dedurre che *Mod1* corrisponde al tasto *Alt*.

Per assegnare l'esecuzione di alcuni comandi a determinate combinazioni di tasti la sintassi è la seguente:

```
Mod4 F1 :ExecCommand gnome-terminal
```

Osservare che ho usato *Mod4*, che corrisponde all'odioso tasto di windows sulla nostra tastiera, al prossimo restart di Fluxbox appena premerò Win+F1, mi si avvierà automaticamente il terminale, il mio amato terminale che tanto ho usato su Gnome e che ancora di più vi sarà utile su un desktop minimale come quello configurato con Fluxbox.

Per farvi qualche altro esempio dell'utilità dei tasti rapidi vi scrivo qui una configurazione utile per il keyfile di Fluxbox:

```
Mod1 Sys_Req :ExecCommand gnome-screenshot
Mod4 F1 :ExecCommand gnome-terminal
Mod4 W :ExecCommand firefox
Mod4 F :ExecCommand nautilus --no-desktop
```

```
Mod4 E :ExecCommand gedit
Mod4 V :ExecCommand gnome-volume-control
Mod4 M :ExecCommand totem
Mod4 R :ExecCommand fbrun
```

Potete osservare che ho assegnato molte utility ad alcuni tasti, alcune le avete sentite prima, altre magari suonano nuove...

La prima in particolare indica che con la combinazione *Alt+Stamp* (o *Rsist*, o *PrintScreen* insomma quella sopra i tasti direzionali che serve solitamente a prendere screenshot del desktop), avvio il programma *Gnome-screenshot*, che sarebbe appunto il programma che in *Gnome* serve a fare screenshot del proprio desktop, lo troverete però leggermente lento in *Fluxbox*, ma pur sempre di grande aiuto.

il secondo lo conoscete già.

Il terzo avvia *firefox*, siate fantasiosi io ho messo *Win+W* (come dire *WWW*, non lo so), voi mettere cosa vi viene meglio.

Il quarto avvia il file manager di *Gnome*, *nautilus*, ma gli impedisce di disegnare il desktop sullo schermo, questo infatti vi appesantirebbe il desktop stesso, cosa che vogliamo evitare avendo un desktop minimale...(o no?)

Il quinto è l'editor di testo che preferite.

Il sesto è l'utility grafica di controllo del volume.

Il settimo è il player di default di *Gnome*, *totem*, che trovo più che spiacevole (meglio *vlc*?).

Last but don't least, *fbrun*... un utilissimo lanciatore che utilizzo spesso e volentieri, alla combinazione *Win+R* (*run*, per ricordarlo meglio), si avvia questa notevole utility che vi permette di avviare un comando senza lasciare aperto il terminale. Utile no?



Potete sbizzarrirvi come i pazzi a personalizzare questo file a seconda delle vostre esigenze, io l'ho fatto e poter vedere decine di minuti di configurazione via file di testo, diventare semplici e interessanti scorciatoie da tastiera è quantomeno gratificante per un comune smanettone.

In giro su internet si trovano interessantissimi file di esempio, ed è molto interessante vedere come tanta altra gente ha risolto problemi di usabilità con combinazioni di tasti sbalorditive.

~/.fluxbox/startup

Il file *Startup*, è un file molto importante ai fini della corretta esecuzione di *Fluxbox*, ed è molto utile altresì per avviare programmi che possono semplificarci la vita in ambienti desktop come *Fluxbox* stesso. La configurazione minima del file *startup* è la seguente:

```
exec /usr/bin/fluxbox
```

Questo avvia semplicemente il bin di Fluxbox... senza questa riga avviando Fluxbox, non si avvierebbe niente, tenetelo a mente.

Ora cosa possiamo avviare con Fluxbox? Cosa ci servirebbe?

Vi posso dire che ci sono una marea di widget, di programmini, di utility per desktop minimali, ma non me la sento proprio di elencarli tutti, vi posso invece spiegare i programmi che ho io nel mio startup, che ritengo utilissimi, se non indispensabili per la corretta usabilità di Fluxbox:

```
nm-applet &  
conky &
```

Nm applet è il network manager applet, che avevo già nominato sopra, la applet di gestione della configurazione della rete.

Risulta molto utile per chi usa il wifi, o l'ethernet e vuole gestirla in modo semplice dalla traybar di Fluxbox (subito a sinistra dell'orologio).

La seconda è *Conky*.

Questo programma è un system-monitor immancabile in un sistema con Fluxbox.

Anche in questo caso, *Conky* è personalizzabile sempre scrivendo lunghi file di testo, dove gli indicherete la posizione, se volete bordi e cosa deve farci vedere.

In giro su internet si trovano interessantissimi file di esempio.

Questo programma non è installato di default quindi se volete utilizzarlo dovete digitare nel terminale:

```
sudo apt-get install conky
```

Cercate un file di configurazione su internet, i più semplici ma utili li potete trovare direttamente nel sito ufficiale di Conky, a questo indirizzo:

<http://conky.sourceforge.net/screenshots.html>

Per chi non lo conoscesse, faccio notare come Conky nonostante la sua apparente semplicità sia quasi un sistema dentro il sistema, con possibilità infinite di utilizzo su cui si potrebbe scrivere un articolo apposito.

Oltre a visualizzare i dati del pc, esso può inoltre essere accoppiato a script in Perl, in Bash, in Python, utili ad eseguire particolari azioni che possono rendere ancora migliore l'attività di monitoring.

Per esempio io ho trovato in giro per la rete uno script in Perl, che ricava le condizioni meteo della mia zona, collegandosi al servizio meteo di yahoo.it, e restituisce grazie ad un particolare font un'iconcina con le previsioni meteo (il giorno che ho fatto lo screen non c'erano previsioni :D) oltre alla temperatura esterna.

Chi metterà fine all'inventiva? Io stesso ho creato uno scriptino per visualizzare il mio ip esterno alla LAN. E grazie a Conky lo vedo sempre sul mio desktop.

Dopo aver trovato una configurazione di base, salvatela in un nuovo file: `.conkyrc`, da lasciare nella propria `/home`.

Smanettandoci e capendo come funziona il file potete ottenere interessanti monitor di sistema. Se volete potete anche inserire ad esempio dei piccoli promemoria, in modo da potervi ricordare le combinazioni di tasti da utilizzare.

Come ho fatto io in questo mio conky:



Il file completo potete trovarlo nel forum di [hackingeasy](#), nella discussione che ho tinyurlato qua: <http://tinyurl.com/fluxboxconf>

Modificando il file startup si può anche aggiungere una semplice riga di comando

```
fbsetbg -f PATHDIUNIMMAGINE
```

che serve a impostare un background al nostro desktop, sbizzaritevi anche qui a sperimentare. Io per esempio ho alcune immagini in una sottocartella di `./fluxbox`, la directory `/backgrounds`, dove ho salvato 2-3 immagini di dimensioni pressochè simili al mio

widescreen, le ho chiamate back.jpg back1.jpg back2.jpg, le imposto come voglio all'avvio e se mi stufano le cambio con un semplice comando usando fbrun (con al combinazione Win+R). Ecco il mio startup, per farvi un'idea:

```
fbsetbg -f /home/vikkio/.fluxbox/backgrounds/back2.jpg
conky &
nm-applet &
exec /usr/bin/fluxbox
```

(Si deve aggiungere una e commerciale "&" dopo i programmi altrimenti la Fluxbox non si avvia)

Esistono diverse altre utility come ho detto prima, per i più nostalgici delle icone sul desktop esiste anche *idesk*, installabile sempre dai repo.

Questo programma serve per creare icone attive al click del mouse, semplici lanciatori configurabili sempre con un file di testo, se lo installate ricordatevi che per lanciarlo automaticamente dovete aggiungerlo in coda al file startup prima di "exec /usr/bin/fluxbox", come per conky, "*idesk &*".

Lui si avvierà, mostrandovi un lanciatore-icona di default chiamato idesk, serve a farvi capire la sintassi da utilizzare per scriverne delle altre.

Io personalmente ho odiato idesk, per i conflitti che dà con alcuni background, e con il mio keyfile e con il menu editato a puntino mi trovo benissimo.

PARTE III

Consigli finali per una maggiore usabilità e leggerezza.

Avete ora una bella Fluxbox e sapete come funziona, credo che vi siete anche informati in giro per vedere se altri ce l'hanno più bella di voi!... e credo che molti c'è l'hanno più bella.

Non è bello ciò che è bello, è bello ciò che piace...dicono i cessi...

Ma a noi smanettoni non deve interessare l'estetica della nostra Fluxubuntu, nossignore! ci interessa l'usabilità e la leggerezza!

Quindi vi do gli ultimi consigli per migliorare ancora e ancora il nostro desktop minimale.

Tutti i consigli sono racchiusi in una semplice frase: Rimpiazzare i programmi di Gnome.

- Nautilus -

Nautilus, il file manager di Gnome! ... è bello sì, ne sono certo: navigazione a schede, la conchiglietta disegnata, quanto è dolce...

Ma purtroppo è pesante come 200 grammi di trippa alle 7:30 del mattino, la nostra povera Fluxubuntu è triste quando avviate questo filemanager. Rimpiazzatelo ad esempio con *pcmanfm*!

Molti consigliano *Rox*, un'altro leggero filemanager, solo che a parità di velocità di avvio io consiglio e straconsiglio *pcmanfm*, installatelo con il solito:

```
sudo apt-get install pcmanfm
```

e avviatelo con `pcmanfm` da terminale o da fbrun...

Ma...che accade? errore? *"impossible blablabla icontheme blablabla"*?

Non abbiate paura è `pcmanfm` che vi dice che non trova un icontheme selezionato di default per tutte le applicazioni che usano `gtk`, voi fatelo stare tranquillo basta creare il file che ci dice lui... avviate il terminale e digitate:

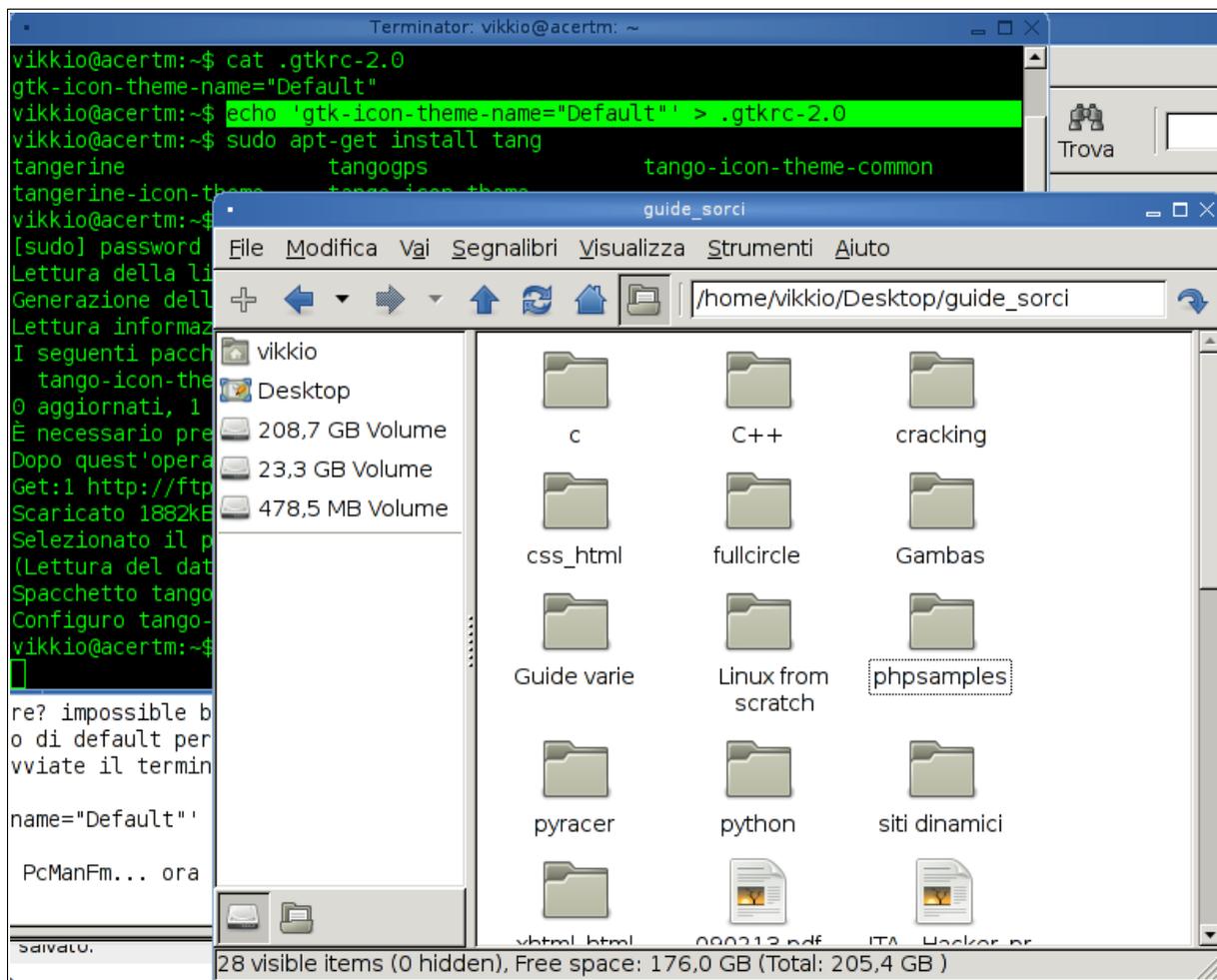
```
echo 'gtk-icon-theme-name="Default"' > .gtkrc-2.0
```

Se usate anche il file-manager da root, dovete fare la stessa cosa per quell'opzione:

```
sudo echo 'gtk-icon-theme-name="Default"' > /root/.gtkrc-2.0
```

e provate a riavviare `PcManFm`... ora vi parte correttamente.

Se non vi piace il tema di icone di default:



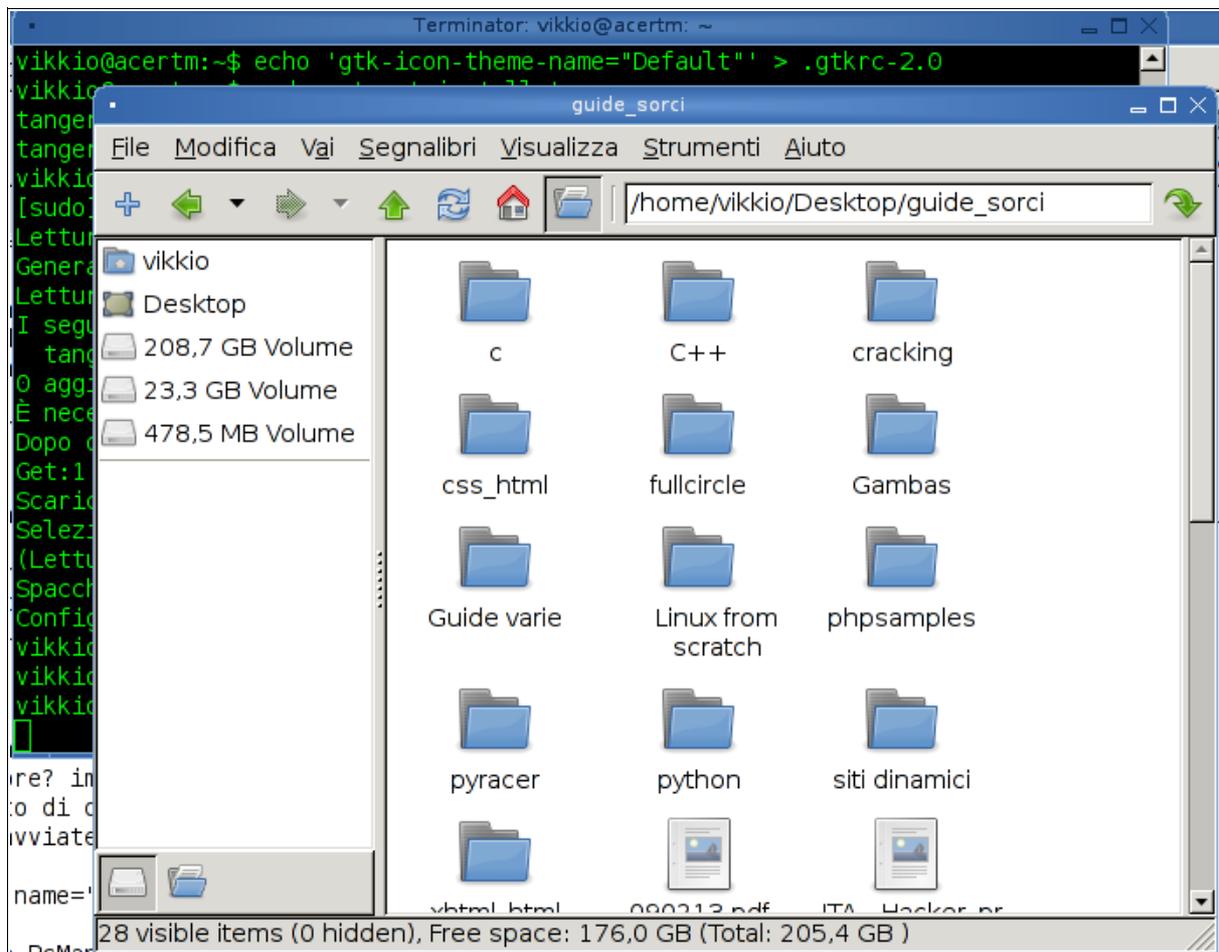
installate il tema tango per esempio...

```
sudo apt-get install tango-icon-theme
```

editate allo stesso modo il file `.gtkrc-2.0` e scrivete “Tango” al posto di “Default”.

Avrete il filemanager veloce e con un tema gradevole di icone.

(Ovviamente sostituite la voce `nautilus --no-desktop`, nel `keyfile`, altrimenti non abbiamo fattonulla :D)



- Gnome-Terminal -

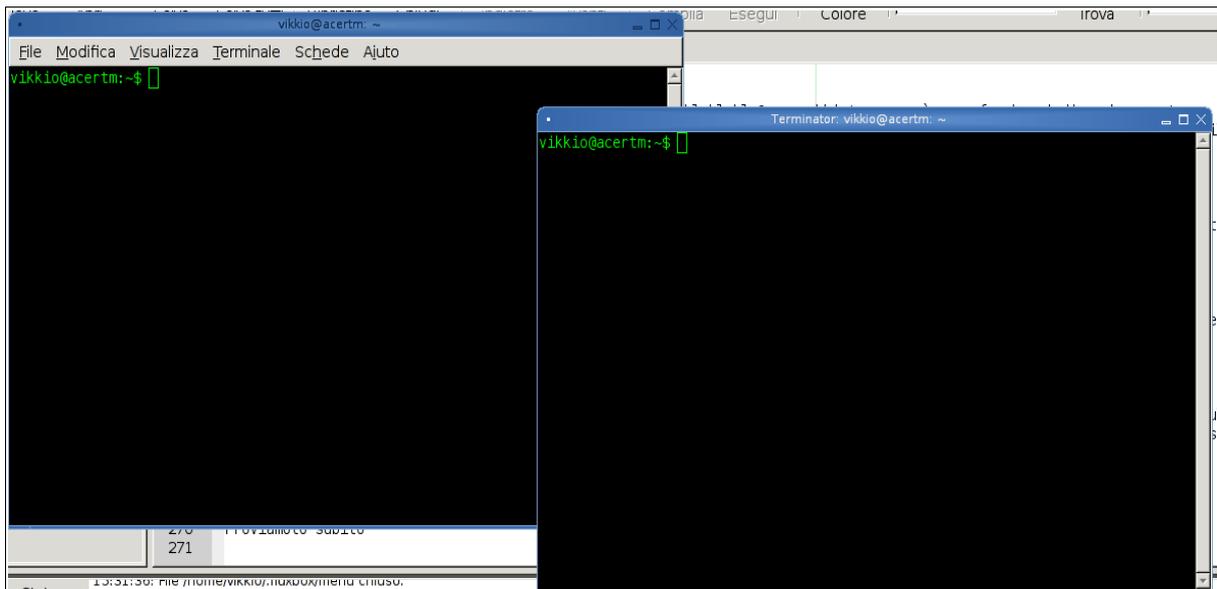
Il terminale di default di Gnome fa i capricci su Fluxbox? I suoi tempi di avvio si intensificano rispetto al DE di default di Ubuntu?

Niente paura vi aiuta Terminator! La famosa frase di Arnold Schwarzenegger nell'omonimo

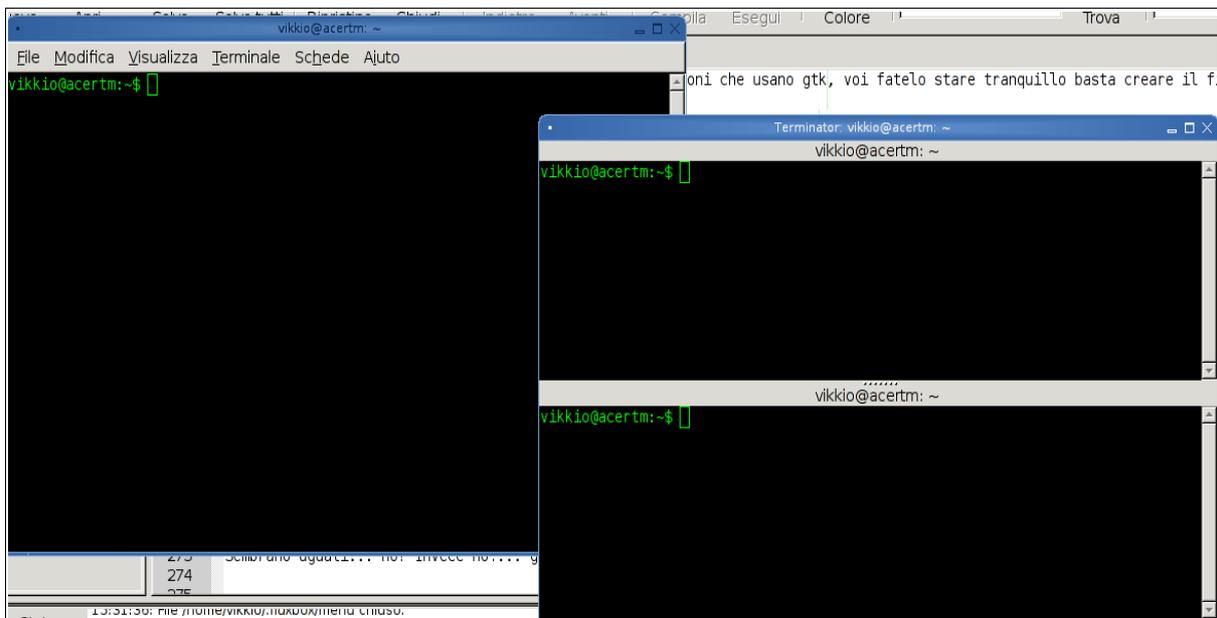
film: *Vieni con me se vuoi vivere!* non potrebbe essere più azzeccata! La nostra Fluxbox rinascerà appena installeremo terminator:

```
sudo apt-get install terminator
```

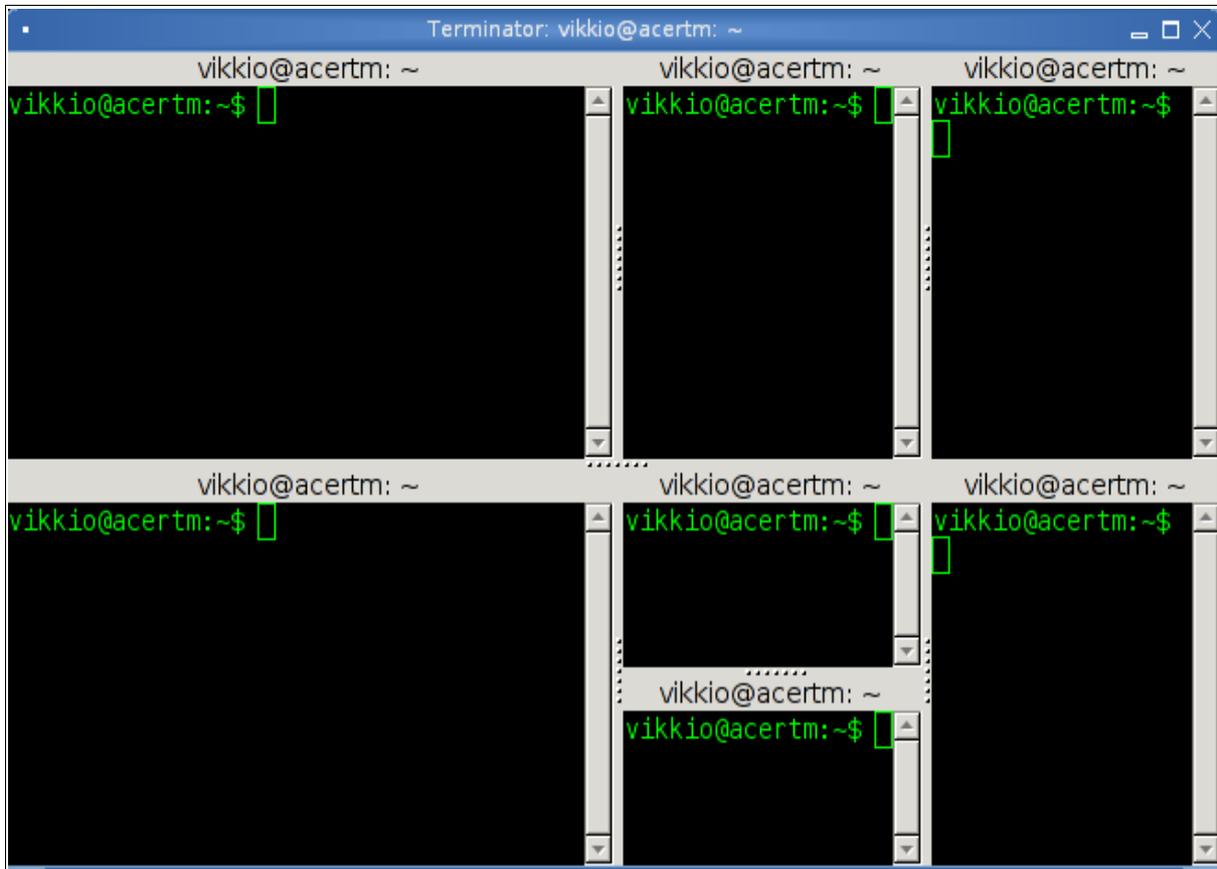
Proviamolo subito (*Gnome-terminal a sinistra e Terminator a destra*):



Sembrano uguali... Invece no!... guardate cosa posso fare con Terminator:



Posso dividere il terminale in tanti altri piccoli terminali ed usarli tutti nella stessa finestra. A molti ricorderà l'uso di *screen* in Gnome-terminal, ma qua è molto più semplice. Bastano pochi click col sinistro del mouse per avere un terminale del genere:



Ovviamente completamente inutilizzabile :D

- EOG -

EyeOfGnome...mamma mia cos'è questo?

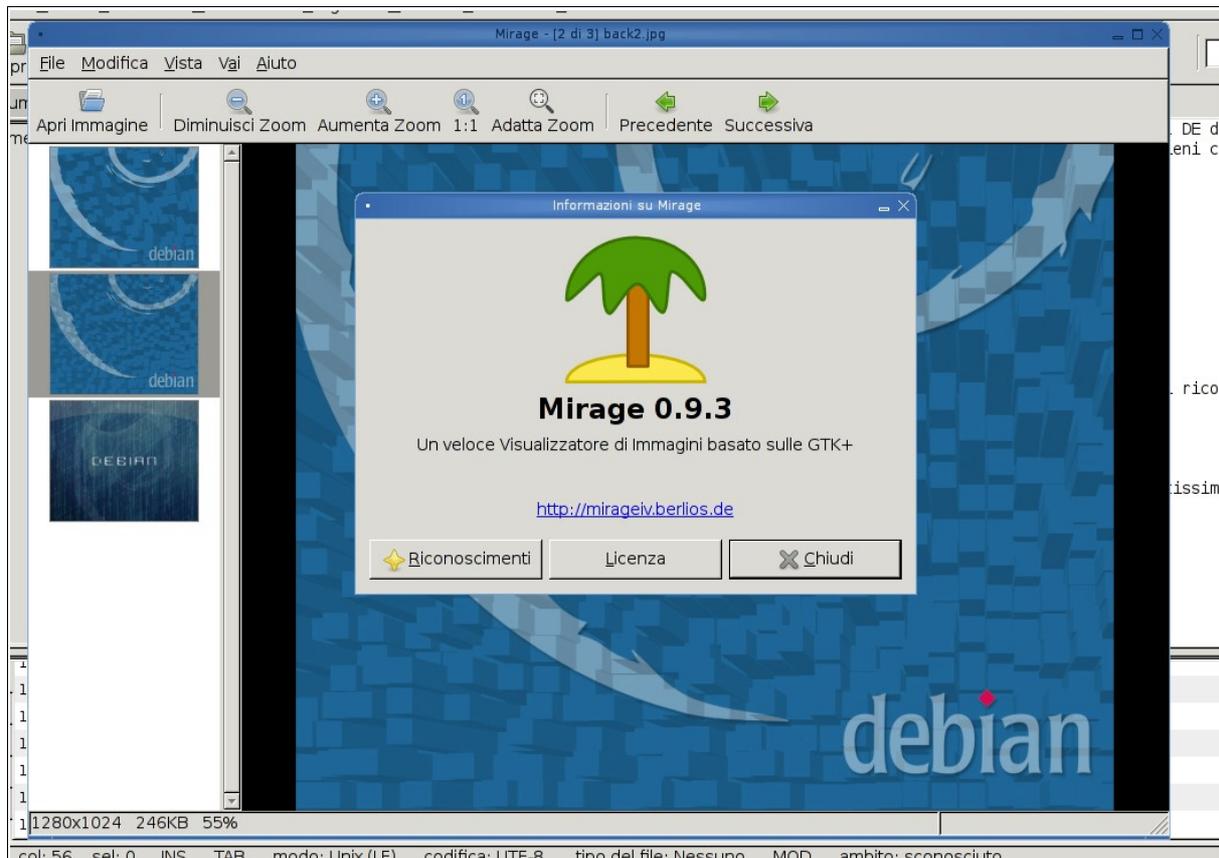
Eye Of Gnome altro non è che il visualizzatore di immagini per Gnome.

Apparentemente un semplice visualizzatore non sembrerebbe così pesante per il sistema, invece usando un DE così leggero come Fluxbox il peso di EOG si fa sentire.

Installiamo al suo posto: Mirage!

```
sudo apt-get install mirage
```

E avremo questo leggerissimo visualizzatore di immagini, ottimizzato per Fluxbox.



- Gedit -

Gedit è bello, utile, ma come gli altri gtk-Gnome based è lento... io preferisco *Geany*

```
sudo apt-get install geany
```

Bello e leggero, da provare!

- Gnome-screenshot -

Gnome-screenshot come visto nel file key è molto utile per prendere screenshot del proprio desktop, la sua lentezza, come gli altri programmi per Gnome, è disarmante, quindi sostituitelo con *scrot*.

```
sudo apt-get install scrot
```

E aggiungete nel file keys la seguente riga (sostituendola a Gnome-screenshot se vi va)

```
Mod1 Sys_Req :ExecCommand scrot -q 50
```

Gli screen si salveranno nella vostra home, con qualità di default del 50%.

Potete modificare la qualità a vostro piacimento incrementando o decrementando il valore dopo lo switch `-q`.

Conclusioni

Ovviamente non si finisce mai di imparare e di sperimentare, non ci si arrende mai con Fluxbox!

Sperimentate e smanettate come ho fatto io, che dopo aver usato Ubuntu per due soli anni già mi crogiolavo nella finzione di un sistema che “fa tutto da sè”.

Grazie a questo mio passaggio a Fluxbox, ho finalmente riscoperto la potenza del terminale e l'efficacia delle utility minimali.

Tutto questo mi ha finalmente riportato ad apprezzare il cuore del mondo GNU/Linux, dove si inizia ricercando solamente la semplicità e quindi si passa alla sperimentazione di nuove distro sempre più complicate ma quindi sempre più personalizzabili.

Per chi utilizza questo OS da un po' di tempo la passione della scoperta tende a perdere slancio e la gestione della complessità diventa quasi un “dato di fatto”, in cui ogni problema va risolto con due click su Google.

Sono sicuro che i così detti power-user Linux hanno capito di cosa sto parlando e spero stiano riflettendo sulle mie parole.

Mi auguro inoltre che questo sarà anche il cammino di tutti coloro che si apprestano ad entrare in GNU/Linux o che comunque ci sono entrati da poco.

Chiedo che rimangano con la mente aperte a nuove distro, a nuove sperimentazioni, e non si fermino ai cubi rotanti e alle finestre molleggianti, che sono semplici dolcetti per strappare utenza a Windows.

Quella stessa utenza che sminuisce e rende sempre più informi e simili le varie distribuzioni, facendo certe volte passare il piacere di sviluppare software più avanzato in favore di software banale per l'utenza media (altrettanto banale), che come sempre tende a comandare come ogni forma di massa non pensante.

vikki088

Analisi virale: il caso WINE

Chiunque si interessi di rimozione manuale di malware sotto Windows, conosce la necessità di individuare file e chiavi di registro create dai vari applicativi virali.

Un giorno mi sono chiesto se esisteva un modo semplice e pulito di testare un malware sotto WINE, per ricavarne quante più informazioni possibili.

La prima domanda che ci si può porre è: perchè WINE?

La prima risposta che salta in mente sarebbe quella di dire "perchè non si prende il virus"; in realtà per chi si diletta in questo lavoro il problema è ininfluente, si può come minimo utilizzare un secondo pc appositamente per bruciarlo e formattarlo.

Le due particolarità che invece rendono WINE utile ai nostri scopi sono principalmente due:

- In primo luogo la struttura di WINE si basa sulla presenza di una directory `./wine` all'interno della `/home` utente in cui è racchiusa tutta la configurazione e tutti i programmi installati.

Per ripulire tutto è sufficiente eliminare questa directory e riavviare WINE in maniera da creare una nuova directory di configurazione di base.

In questo modo è possibile tentare più installazioni, su differenti configurazioni, tenendo da parte una configurazione di default per le prove ecc.

- La seconda caratteristica di WINE è la simulazione del registro di Windows. Ovviamente installando un programma per Windows sotto GNU/Linux si pone la necessità di gestire la creazione delle chiavi di registro; a differenza di Windows il registro di Wine è perfettamente identificabile da tre file presenti all'interno della directory di configurazione:

- `system.reg` che corrisponde alla sezione HKLM
- `userdef.reg` e `user.reg` che simulano HKCU/Configuration

Questa particolarità ci permette di lavorare direttamente sul registro trattandolo come un comune log in forma di testo.

M.A.S.U.W (Malware Analyzing Software Under WINE) è un piccolo programma che risponde proprio a queste richieste in maniera tanto semplice quanto utile.

Il lavoro di questo software è quello di loggare i dati dell'albero delle directory di `./wine` e del suo registro prima dell'installazione di un determinato programma e al termine della stessa.

Eseguendo un'analisi delle differenze rilevate nel confronto tra i due log si ottengono gli estratti dei file e delle chiavi create dall'eseguibile.

Ogni programma come si sa vive di vita propria...e soprattutto di problemi propri.

La particolarità della progettazione di MASUW è stata quella di riuscire ad aderire il più possibile alle necessità richieste dall'ipotetico utilizzatore.

È infatti probabile (o probabile) che ci si presenti un utente abituato a lavorare su Windows, che non apprezzi la riga di comando e che non desideri impiegare molto spazio su disco per

un sistema GNU/Linux.

E se non sa nemmeno installare GNU/Linux? Dubito che possa fare analisi virale...

Per mantenere una assoluta leggerezza ma garantire un minimo di interfaccia grafica ho voluto utilizzare i servizi di Zenity, un buon software di cui ho apprezzato molto le qualità in rapporto all'eccezionale immediatezza d'uso.

Il programma è disponibile di default in numerose distribuzioni basate su Gnome ed è normalmente installabile senza dipendenze anche nei sistemi non strettamente basati su Gtk.

Il codice

MASUW si compone di due file principali: masuw e logger

Il primo file gestisce le impostazioni principali dell'interfaccia primaria attraverso un loop che utilizza Zenity:

```
function select_operation
{
OP_SELECT=$(zenity --title=M.A.S.U.W. --width=480 --height=225
--window-icon=/usr/share/pixmaps/masuw.png --list --radiolist
--column="" --text="<b>Welcome to M.A.S.U.W. Malware Analyzing
Software Under WINE</b>\n\nPlease choose one of the following
options:\n" --column="Options" FALSE "Install a program" FALSE
"Help" FALSE "About")

case $OP_SELECT in
    "Install a program" ) check_winedir
                        x-terminal-emulator -e $MASUW/logger
                        exit
                        ;;
    Help )               helptext
                        exit
                        ;;
    About )             version
                        exit
                        ;;
esac
}
```

Come si vede la variabile OP_SELECT prende il valore di uscita generato dalla finestra elenco (`--list --radiolist`) di Zenity.

È possibile impostare larghezza e altezza della finestra, il suo titolo (cioè l'elemento "caption") il percorso della sua icona, il tipo di lista (`--radiolist` dell'esempio oppure `--checkboxlist`) e quindi settare le impostazioni della lista stessa:

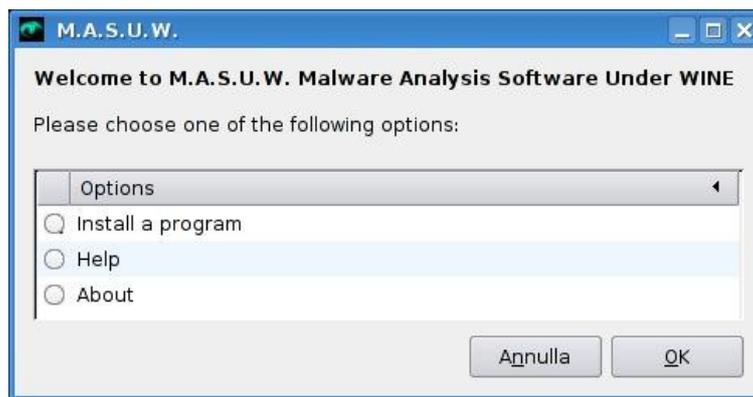
`--text` ci permette di inserire del testo esplicativo nella parte superiore della lista. Da notare come sia possibile anche l'inserimento di testo in grassetto.

--column gestisce il titolo delle colonne dell'elenco e può ovviamente venire utilizzato più volte in base alle necessità.

Nel mio caso ho scelto di non dare un titolo alla prima colonna (quella con i bottoni di checklist) e di titolare "Option" la seconda.

Gli elementi dell'elenco, il cui testo costituisce l'uscita di Zenity come si vede dalla funzione, possono aver attribuito i valori TRUE o FALSE in base alla scelta di selezionare o meno opzioni di default.

Il risultato all'avvio del programma sarà il seguente:



Certo non siamo ai livelli di una buona interfaccia realizzabile in "Gtk classico", però la semplicità d'uso di Zenity è veramente impressionante.

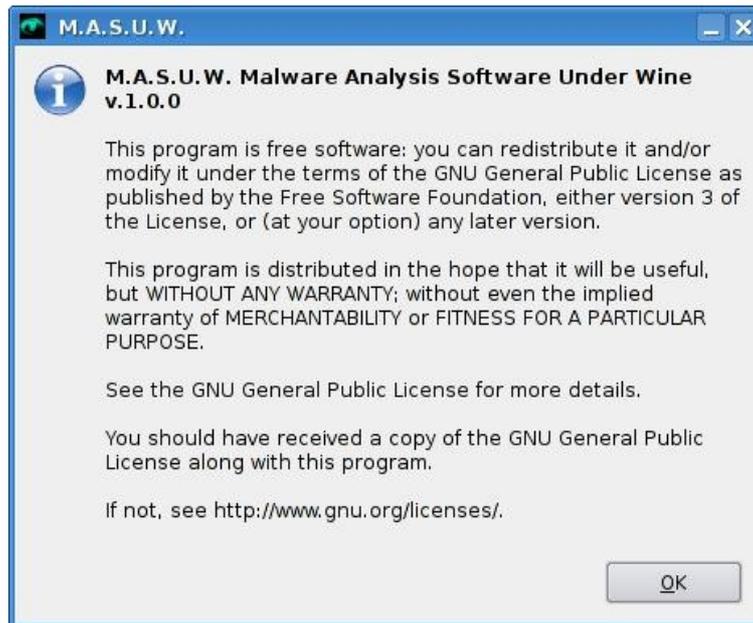
Con una manciata di impostazioni inseribili in un qualunque script per Bash si riesce a creare una discreta GUI per applicazioni da terminale.

Una cosa che mi è venuta in mente a fine programma è ad esempio l'aggiunta dell'opzione "About" che riporta la GPL3 come le comuni applicazioni grafiche richiamando la funzione "version", che altro non è che una finestra informativa di Zenity.

```
function version
{
$ZEN_DEF --info --text="<b>M.A.S.U.W. Malware Analysis Software
Under Wine\nv.1.0.0</b>
\nThis program is free software: you can
redistribute it and/or modify it under the terms of the GNU General
Public License as published by the Free Software Foundation, either
version 3 of the License, or (at your option) any later version.
\nThis program is distributed in the hope that it
will be useful, but WITHOUT ANY WARRANTY; without even the implied
warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
\nSee the GNU General Public License for more
details.
\nYou should have received a copy of the GNU General
```

```
Public License along with this program.  
        \nIf not, see http://www.gnu.org/licenses/."  
select_operation  
}
```

La variabile `$ZEN_DEF` contiene già dei valori di default di larghezza, altezza, icona. L'uso di `--info` ci permette di inserire il nostro testo, creando il seguente risultato...



Bello vero? Un giochino molto divertente...

Dalla funzione principale `select_operation` si può vedere come l'intero lavoro di analisi e produzione del log finale sia fatto dal file `logging`, che viene richiamato in un'apposita istanza di terminale.

Anche questo secondo file utilizza comunque Zenity per alcune attività di interazione con l'utente, oltre ad avviare una seconda istanza di terminale per l'installazione di un programma sotto WINE.

È possibile durante l'esecuzione creare un backup della directory di configurazione corrente `.wine` per poi eventualmente sostituirla alla fine del processo in maniera da ritornare alla corretta configurazione iniziale, tenendo sempre pulito il nostro sistema.

La funzione principale `main_log` fa un uso massiccio di `diff`, `sed` e `grep` per la gestione dei file temporanei (trattati in un'apposita directory `.masuw`) creati e la generazione del log finale `masuw.txt` che risulterà presente nella propria home utente.

La procedura utilizzata in questa operazione, sebbene sia abbastanza complessa, è decisamente lineare e quindi facilmente comprensibile a chiunque conosca i comandi

utilizzati comunemente per trattare file di testo tramite GNU/Linux.

L'unica cosa importante da far notare è il meccanismo di eliminazione continua di ogni log temporaneo appena questo non risulti più utile al corretto svolgimento della procedura.

La preferenza di questo metodo, contrapposto ad un'apposita funzione di *cleanup* al termine del programma (decisamente più comoda per lo sviluppatore) deriva dalla notevole variabilità di dimensione dei log in base al numero di applicazioni installate sotto WINE dall'utente.

Ogni programma installato aumenta infatti di molto la quantità di file e chiavi di registro presenti nella directory *.wine* creando file di log che possono facilmente crescere di parecchio.

Un caso concreto

A questo punto è bene testare MASUW per vedere come lavora.

Per fare la prova ho bisogno di un'applicazione virale qualunque, quindi apro il fedele Google e tento una richiesta per andare a colpo sicuro: *porn raped babe download freeware*

Al primo risultato trovo un sito vetrina di rinvio ad altri url (decisamente molto più interessante di quello che state leggendo), al secondo link arrivo ad una serie sterminata di fi...ehm link ad uno splendido *adult.exe*

Scarico quindi il file, pronto a testare la nuova creazione di floatman...

Come prima cosa è necessario ovviamente installare il programma tramite la classica procedura:

```
dante@debian:~$ make && sudo make install
```

per il suo avvio è a questo punto sufficiente digitare *masuw* da terminale, o creare un apposito lanciatore sul desktop.

All'attivazione di MASUW, da cui otterremo lo screen indicato nella prima immagine di questo articolo, scegliendo l'opzione "*Install a program*" inizieremo la procedura guidata.

All'apertura della prima istanza della shell verranno verificate le impostazioni del programma e verrà quindi richiesto se eseguire o meno il backup della directory *.wine*

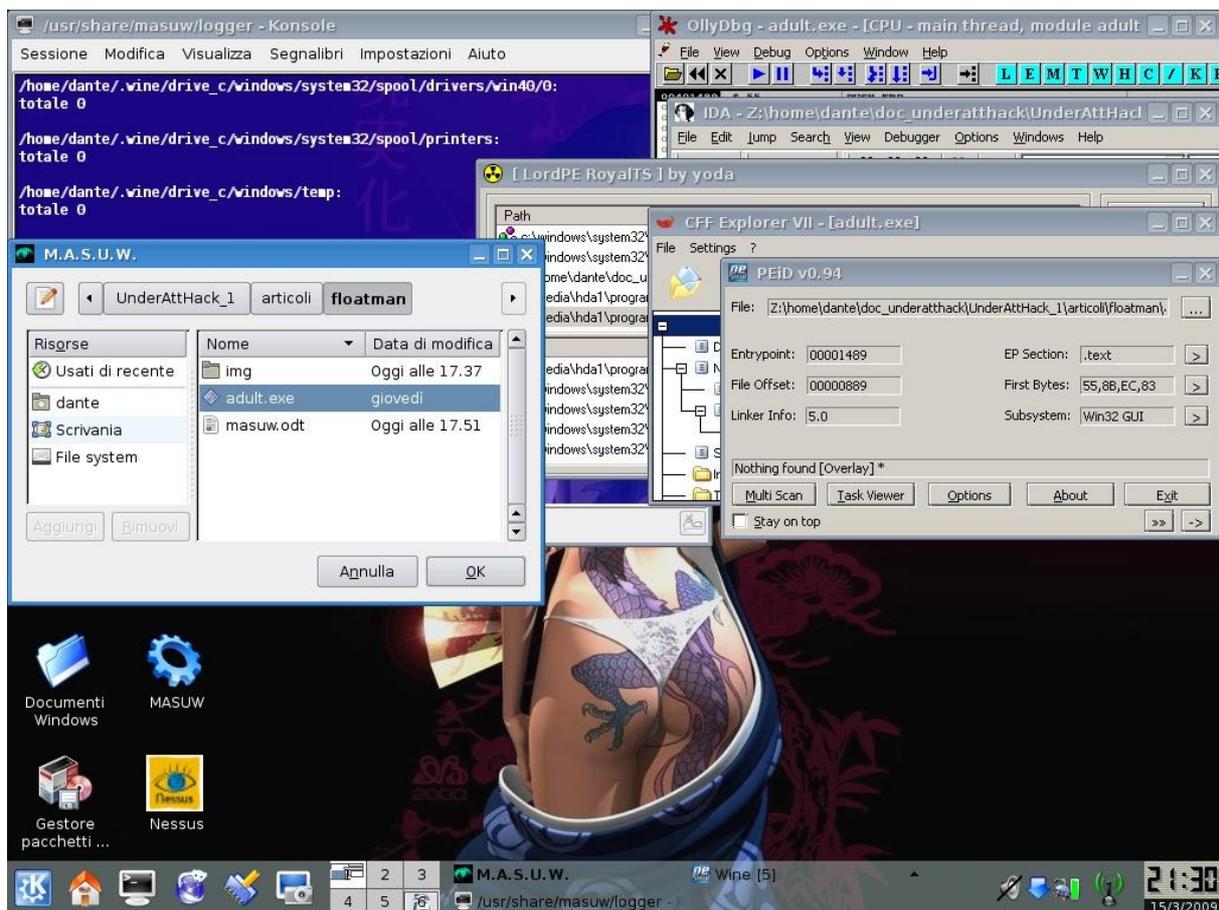
Nel caso MASUW individui una directory di backup creata precedentemente, richiederà all'utente se sovrascriverla o rinominarla.

In questo modo saranno possibili test e modifiche successive, mantenendo intatto il nostro sistema principale prima "dell'infezione".

Per quanto verrà trattato in questo documento di esempio la procedura di backup non sarà affatto utile, almeno per il fatto che per questo test sto lavorando su una directory minimale per accorciare l'analisi; in ogni caso è importante segnalare come nel caso si scelga la procedura con backup, alla fine dei lavori MASUW richiederà all'utente se desidera sostituire la directory "infetta" con quella precedentemente salvata.

Rispondendo negativamente sarà possibile effettuare altri test sui file infetti, ad esempio con un debugger come OllyDBG (perfettamente funzionante sotto WINE), lo stesso *winedbg*

disponibile di default con possibilità di funzionare tramite GDB, oppure con una vasta serie di programmi utilizzabili per mezzo dell'emulazione con WINE come si può chiaramente vedere nell'esempio visualizzato dalla schermata successiva, che è stata presa nel momento in cui il terminale di MASUW richiede, ancora attraverso l'uso di una finestra creata da Zenity la selezione del file da installare.



Dopo aver concluso la procedura troveremo nella nostra home un file di log *masuw.txt*

Però non è ancora il momento di avviare il programma, prima è infatti necessario prepararsi bene, oltre a scoprire tutti i limiti di questo nostro semplice esempio.

Come avevo spiegato in questo momento sto utilizzando una directory di WINE perfettamente pulita, senza nessuna applicazione installata.

Un sistema del genere non comprende parecchie .dll ed è strutturato allo scopo di permettere l'installazione di applicazioni per Windows (il lavoro che fa WINE) ma non precisamente al nostro, che è quello di simulare una struttura di sistema vera e propria di un sistema Microsoft con le varie directory, la serie di librerie presenti di default ecc.

Nel caso qui studiato sarà ad esempio rilevante la mancanza di Explorer che verrebbe

sicuramente sfruttato da qualunque malware.

L'ideale per l'utilizzo di MASUW è quello di dargli in pasto una directory di configurazione organizzata attentamente; ci sono parecchie applicazioni dedicate alla configurazione di WINE e all'installazione di svariati strumenti per riprodurre un sistema Windows.

Darne un elenco e spiegare il loro funzionamento non è il compito di questo documento, dove invece ci limiteremo ad un utilizzo minimo di MASUW.

Sebbene questi termini possano sembrare delle carenze, considerate che il solo elencare i limiti di quanto si sta facendo permettere di capire (o pretende di farlo...) i fondamenti su cui si basa il caso qui esposto.

Per prima cosa tento l'installazione del nostro *adult.exe* da terminale, in modo da vedere eventuali errori:

```
$ wine ~/doc_underatthack/UnderAttHack_1/articoli/floatman/adult.exe
```

Come era prevedibile l'installazione trova parecchi errori e WINE avvia il proprio lungo debugging (che quindi non sarà postato) permettendo di individuare le librerie mancanti:

- *uxtheme.dll*
- *mpr.dll*
- *imm32.dll*
- *iphlpapi.dll*

Avendo una macchina in dual-boot, vado a copiarle dalla partizione di Windows e le inserisco in system32 nella directory di WINE.

Nota: molte .dll di un sistema Windows non sono distribuibili, nel caso di macchine senza dual-boot la loro copia da altri pc contravviene alla licenza d'uso.

Faccio nuovamente la prova e vedo che la situazione è migliorata di molto.

Come previsto *adult.exe* cerca Explorer alla fine dell'installazione; comunque va bene così...

Dopo aver cancellato e ricreato la directory di configurazione, procedo all'installazione tramite MASUW.

A dire il vero la cosa che più mi sconvolge è la mole di icone che mi si creano sul desktop, dalle suonerie per cellulare ai casino on-line...però ho appena fatto uno screen, quindi per motivi editoriali tralasciamo questo fatto.

Andiamo invece a vedere MASUW cosa ha scoperto nel suo lungo log (500 KB).

La prima parte presenta una serie di dati generali del processo di logging svolto, più precisamente vengono individuati:

La data e l'ora di creazione del log, utile nel caso di log multipli

l'autore del log, cioè l'utente

Il percorso completo del file installato, utile a chi come floatman ha un leggero disordine nella home, tale da non ritrovare più il file installato

Ultimo ma più importante, la versione di WINE installata (nel mio caso è la 1.0.1 che è un po' vecchiotta). WINE infatti è un progetto che cresce piuttosto velocemente ed è quindi bene avere installata sempre l'ultima versione disponibile.

```
# M.A.S.U.W. - MALWARE ANALYSIS SOFTWARE UNDER WINE v.1.0.0 #  
  
Log file creation: Sun, 15 Mar 2009 20:38:11 +0100  
  
Author: dante  
  
Installation file logged:  
/home/dante/doc_underatthack/UnderAttHack_1/articoli/floatman/adult.  
exe  
  
Wine version in use: wine-1.0.1-174-gc4039bd
```

La parte successiva del log viene spiegata direttamente:

```
[ 1.0 ] - Directory Tree and Registry Complete Overview  
  
Lines indicated by the arrows are those which were changed by the  
installation.  
See Final Report (you can search it with your favourite editor) for  
changed lines list only.  
  
[ 1.1 ] - Modifications in directory structure:
```

Il log di MASUW comprende infatti un indice in cui vengono prima indicati tutti i dati globali (1.0) suddivisi tra albero delle directory (1.1) e registro (1.2), quindi si procede al riassunto (2.0) dei valori individuati, sempre suddivisi tra file/directory (2.1) e modifiche alle chiavi di registro (2.2).

Come scritto, verranno indicate con una freccia (“==>”) le modifiche effettuate dall'installazione, di cui ad esempio la prima individuata è nella directory Programmi

```
/home/dante/.wine/drive_c/Programmi:  
4,0KCommonFiles/  
4,0KInternetExplorer/  
==> 4,0KXPPoliceAntivirus/
```

Come si vede è stato aggiunto un nuovo programma *XPPoliceAntivirus* dentro una nuova cartella. Direi che è un tipico caso di downloader...

Se avessimo avuto Explorer installato avremmo molto probabilmente individuato un hijacking nel log delle chiavi di registro, con una serie di *trusted-url* aggiunti.

I siti inseriti sarebbero probabilmente dotati di codice iniettivo *ActiveX* per avviare il nostro bellissimo downloader. In definitiva siamo alle solite:

Internet Explorer come porta ufficiale delle infezioni su client Windows.

Pagine fasulle con ottime tecnologie che permettono il controllo di una macchina quando un browser attraversa una pagina.

Floatman che scrive da Debian, parlando di aspetti di sicurezza su Windows totalmente ignoti alla quasi totalità dell'utenza di quel sistema.

Rispetto a dieci anni fa i computer sono cento volte tanti, mentre il numero di chi li sa usare è lo stesso.

Però noi non abbiamo Explorer installato, quindi sto facendo solo supposizioni azzardate, probabilmente errate...

La parte più importante del log non è però quella iniziale ma quella riassuntiva, indicata nell'indice come capitolo 2

[2.0] - Final Report:

Those are lines (with number) modified by the install process.

[2.1] - In directories structure:

Lines: Modifications:

Di seguito verranno elencate le varie voci individuate, precedute dal numero di riga presente nel nostro log.

Imposto la visualizzazione dei numeri di riga sul mio Kwrite e inizio a vedere cosa mi dice il log.

```
43:     4,0KXPPoliceAntivirus/  
50:     /home/dante/.wine/drive_c/Programmi/XPPoliceAntivirus:  
51:     0setup.dat
```

Queste prime tre righe mi spingono a ritenere che si tratti dello stesso programma, loggato prima nella directory superiore e poi in profondità; vado quindi nella parte iniziale del log e vedo a cosa si riferiscono le righe segnalate.

```
/home/dante/.wine/drive_c/Programmi:  
4,0KCommonFiles/  
4,0KInternetExplorer/  
  ==> 4,0KXPPoliceAntivirus/
```

```
/home/dante/.wine/drive_c/Programmi/CommonFiles:
```

```
/home/dante/.wine/drive_c/Programmi/InternetExplorer:  
4,0Kiexplore.exe
```

```
  ==> /home/dante/.wine/drive_c/Programmi/XPPoliceAntivirus:
```

```
==> 0setup.dat
```

Come si vede viene loggato un setup.dat come se fosse un file nullo, evidentemente è solo un file di transito necessario come base per il downloader.

Ci si possono porre varie domande, ad esempio viene da chiedersi come possa un antivirus individuare un file di 0 byte, magari con un nome random...però passiamo avanti perchè abbiamo altro da fare.

Infatti cosa ti scopre MASUW? Ladies & Gentlemen, here you are!

```
58: 16Kiehost.dll
```

Che riporta a

```
/home/dante/.wine/drive_c/windows:  
[.....]  
==> 16Kiehost.dll
```

il cuore pulsante del nostro *adult.exe* su cui si concentrerebbe tutto il lavoro successivo con le tecniche tipiche dell'analisi di eseguibili, momento in cui MASUW si ferma.

Il resto del log presenta solo icone e vari .lnk che infatti mi hanno impostato il desktop

```
155: 4,0KCheapPharmacyOnline.LNK  
156: 4,0KCheapSoftware.LNK  
157: 4,0KMP3Download.LNK  
159: 4,0KSearchOnline.LNK  
160: 4,0KSMSTRAP.LNK  
161: 4,0KVIPCasino.LNK  
175: 16Kc.ico  
198: 16Km3.ico  
199: 8,0Km.ico  
214: 12Kp.ico  
228: 24Ksf.ico  
233: 8,0Ks.ico  
274: 8,0K3550_c.png  
275: 4,0K3550_m3.png  
276: 4,0K3550_m.png  
277: 4,0K3550_p.png  
278: 8,0K3550_sf.png  
279: 4,0K3550_s.png
```

così ad occhio potrei dire che il “coso” scaricato dal downloader vada ad iniettarsi nella nostra .dll, però indagare non è affare di questo articolo quindi procediamo con le modifiche al registro di sistema.

Queste come detto all'inizio sono individuate all'indice 2.2 del log di MASUW e seguono lo stesso principio indicato in precedenza, cioè con i numeri di riga che rimandano alla prima

parte del log. Indovinate un po' come inizia?

```
1075: [HKEY_LOCAL_MACHINE\Software\Classes\CLSID\
{12c7290a-157b-4f43-b109-97e792c598ed}]
1076: @="WinGDIClass"
1077:
1078: [HKEY_LOCAL_MACHINE\Software\Classes\CLSID\
{12c7290a-157b-4f43-b109-97e792c598ed}\InprocServer32]
1079: @="C:\\windows\\iehost.dll"
1080: "ThreadingModel"="Apartment"
1081:
1082: [HKEY_LOCAL_MACHINE\Software\Classes\CLSID\
{12c7290a-157b-4f43-b109-97e792c598ed}\ProgID]
1083: @="WinGDIApp.WinGDI.1"
1084:
1085: [HKEY_LOCAL_MACHINE\Software\Classes\CLSID\
{12c7290a-157b-4f43-b109-97e792c598ed}\Programmable]
1086:
1087: [HKEY_LOCAL_MACHINE\Software\Classes\CLSID\
{12c7290a-157b-4f43-b109-97e792c598ed}\TypeLib]
1088: @="{8a10fc9b-8d76-4e95-a9be-acda2f665c30}"
1089:
1090: [HKEY_LOCAL_MACHINE\Software\Classes\CLSID\
{12c7290a-157b-4f43-b109-97e792c598ed}\VersionIndependentProgID]
1091: @="WinGDIApp.WinGDI"
```

la nostra brava *iehost.dll* pronta a piazzarsi nel sistema in maniera direi più che persistente...
Proseguendo con le segnalazioni di MASUW...ricordate le mie lagnanze su Explorer e sul probabile hijacking?

Beh, forse ci eravamo scordati qualcosa, cioè gli utilissimi *helper object* del sempre ottimo browser di Windows! Come abbiamo fatto a non pensarci subito...fortuna che MASUW lo ricorda immediatamente

```
6167: [HKEY_LOCAL_MACHINE\Software\Classes\Interface\
{967A494A-6AEC-4555-9CAF-FA6EB00ACF91}]
6168: @="_IBhoAppEvents"
6169:
6170: [HKEY_LOCAL_MACHINE\Software\Classes\Interface\
{967A494A-6AEC-4555-9CAF-FA6EB00ACF91}\ProxyStubClsid]
6171: @="{00020424-0000-0000-C000-000000000046}"
6172:
6173: [HKEY_LOCAL_MACHINE\Software\Classes\Interface\
{967A494A-6AEC-4555-9CAF-FA6EB00ACF91}\ProxyStubClsid32]
6174: @="{00020424-0000-0000-C000-000000000046}"
6175:
6176: [HKEY_LOCAL_MACHINE\Software\Classes\Interface\
```

```
{967A494A-6AEC-4555-9CAF-FA6EB00ACF91}\TypeLib]
6177: @="{8A10FC9B-8D76-4E95-A9BE-ACDA2F665C30}"
6178: "Version"="1.0"
6179:
6180: [HKEY_LOCAL_MACHINE\Software\Classes\Interface\{9692BE2F-
EB8F-49D9-A11C-C24C1EF734D5}]
6181: @="IBhoApp"
6182:
6183: [HKEY_LOCAL_MACHINE\Software\Classes\Interface\{9692BE2F-
EB8F-49D9-A11C-C24C1EF734D5}\ProxyStubClsid]
6184: @="{00020424-0000-0000-C000-000000000046}"
6185:
6186: [HKEY_LOCAL_MACHINE\Software\Classes\Interface\{9692BE2F-
EB8F-49D9-A11C-C24C1EF734D5}\ProxyStubClsid32]
6187: @="{00020424-0000-0000-C000-000000000046}"
6188:
6189: [HKEY_LOCAL_MACHINE\Software\Classes\Interface\{9692BE2F-
EB8F-49D9-A11C-C24C1EF734D5}\TypeLib]
6190: @="{8A10FC9B-8D76-4E95-A9BE-ACDA2F665C30}"
6191: "Version"="1.0"
```

Come potrebbe altrimenti il buon pornomane sapere che deve scaricare un trojan serio invece che questo giocattolo?

Dopo questa prima configurazione del registro è giusto andare a rifinire l'opera di infezione, prima impostando a dovere la nostra amica .dll

```
8133: [HKEY_LOCAL_MACHINE\Software\Classes\TypeLib\
{8A10FC9B-8D76-4E95-A9BE-ACDA2F665C30}]
8134:
8135: [HKEY_LOCAL_MACHINE\Software\Classes\TypeLib\
{8A10FC9B-8D76-4E95-A9BE-ACDA2F665C30}\1.0]
8136: @="WinGDI1.0TypeLibrary"
8137:
8138: [HKEY_LOCAL_MACHINE\Software\Classes\TypeLib\
{8A10FC9B-8D76-4E95-A9BE-ACDA2F665C30}\1.0\0]
8139:
8140: [HKEY_LOCAL_MACHINE\Software\Classes\TypeLib\
{8A10FC9B-8D76-4E95-A9BE-ACDA2F665C30}\1.0\0\win32]
8141: @="C:\\windows\\iehost.dll"
8142:
8143: [HKEY_LOCAL_MACHINE\Software\Classes\TypeLib\
{8A10FC9B-8D76-4E95-A9BE-ACDA2F665C30}\1.0\FLAGS]
8144: @="0"
8145:
8146: [HKEY_LOCAL_MACHINE\Software\Classes\TypeLib\
{8A10FC9B-8D76-4E95-A9BE-ACDA2F665C30}\1.0\HELPPDIR]
8147: @="C:\\windows"
```

Quindi settandone bene i parametri d'uso nel sistema

```
8193: [HKEY_LOCAL_MACHINE\Software\Classes\WinGDIApp.WinGDI]
8194: @="WinGDIClass"
8195:
8196: [HKEY_LOCAL_MACHINE\Software\Classes\WinGDIApp.WinGDI\CLSID]
8197: @="{12c7290a-157b-4f43-b109-97e792c598ed}"
8198:
8199: [HKEY_LOCAL_MACHINE\Software\Classes\WinGDIApp.WinGDI\CurVer]
8200: @="WinGDIApp.WinGDI.1"
8201:
8202: [HKEY_LOCAL_MACHINE\Software\Classes\WinGDIApp.WinGDI.1]
8203: @="WinGDIClass"
8204:
8205: [HKEY_LOCAL_MACHINE\Software\Classes\WinGDIApp.WinGDI.1\CLSID]
8206: @="{12c7290a-157b-4f43-b109-97e792c598ed}"
```

Bravo *adult!* Ottimo lavoro ^^

Fino a questo punto però non possiamo a dire il vero trarre grandi conclusioni, se non il fatto che l'infezione è tutta effettuata a partire dalla .dll che comunque avevamo già verificato in precedenza.

Da qui in avanti, dopo che il nostro applicativo virale ha portato a termine tutto questo lavoro per la propria auto-definizione, non resta altro che settare i quei parametri del registro necessari per attivare l'infezione vera e propria:

1° - impostare le notifiche di allerta per far attivare il downloader

```
9140: [HKEY_LOCAL_MACHINE\Software\Microsoft\SecurityCenter]
9141: "AntiVirusDisableNotify"="1"
9142: "FirewallDisableNotify"="1"
9143: "UpdatesDisableNotify"="1"
```

2° - avviare il BHO

```
9165: [HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\
Explorer\BrowserHelperObjects]
9166:
9167: [HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\
Explorer\BrowserHelperObjects\{12c7290a-157b-4f43-
b109-97e792c598ed}]
9168: "NoExplorer"=dword:00000000
9169:
9170: [HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\
Explorer\BrowserHelperObjects\{12c7290a-157b-4f43-
```

```
b109-97e792c598ed}\1]
```

3° - posizionare il perfetto antispyware virale nel Pannello di controllo

```
12666:      [HKEY_USERS\S-1-5-4\ControlPanel\don'tload]
12667:      "scui.cpl"="No"
12668:      "wscui.cpl"="No"
```

4° - Abilitare il nostro presunto Hijacker

```
12784:      "ProxyEnable"=dword:00000000
```

in riferimento a:

```
[HKEY_USERS\S-1-5-4\Software\Microsoft\Windows\CurrentVersion\Intern
etSettings]
@=""
    ==> "ProxyEnable"=dword:00000000
```

log concluso...

Ci rimarrebbe la nostra .dll da analizzare, però la cosa non riguarda più né questo programma né questa guida.

Per chiudere il tutto, butto via quella schifezza che ho installato alla maniera di GNU/Linux

```
dante@ciquita:~$ rm -r .wine && winecfg
```

clikko "Annulla" nella finestra di winecfg ed ho formattato e reinstallato Windows!

Per chi fosse interessato all'uso, lo studio, la modifica ecc. di MASUW, il programma lo trovate a questo indirizzo:

<http://myville.altervista.org/software/masuw.html>

Conclusioni

Bene, abbiamo concluso l'analisi e abbiamo recuperato un numero notevole di informazioni. Certo come detto in precedenza il test ha avuto un valore limitato, sarebbe bello ripeterlo nuovamente installando Explorer, magari importando un registro originale di Windows e produrne uno apposito (ricordo che il registro di WINE è modificabile come file di testo) per questo lavoro.

Insomma si potrebbe fare tutto e molto di più, come ogni volta accade si può sempre migliorare e creare qualcosa di migliore. Questo è lo spirito del vero hacking.

Come ogni volta, nel momento in cui concludo queste prove, adoro il fatto che alla fine del test mi pongo più domande di quelle che mi avevano spinto a realizzarlo.

Il primo problema che mi sono posto è quanto sia possibile duplicare dentro WINE un sistema Windows, cioè se tramite questo splendido programma sia possibile ottenere una sorta di doppio sistema come avviene con Cygwin.

Pochi mesi fa, nel momento in cui ho scoperto WINE quasi per gioco, avevo installato *bblean* (Blackbox per Windows) sotto WINE in una finestra di emulazione 800x600px; il menu di *bblean* puntava ovviamente alla directory Programmi, cioè all'interno di WINE. Quindi non ero andato molto distante dall'aver un sistema virtuale vero e proprio dove questo test potrebbe essere condotto in modo migliore...ok, basta fantasticare.

Una seconda cosa che mi sono chiesto alla fine di questo articolo, molto più inerente all'argomento, è come si comporterebbe un software con capacità di rootkit in una simile analisi.

A livello teorico risulterebbe impossibile per il rootkit nascondere al propria esistenza, d'altra parte però per chi analizza il log si porrebbe il problema di come poter riconoscere una di queste infezioni rispetto ad una di tipo classico.

Ok. Adesso mi fermo...

Spero come sempre che questa guida e questo sorgente in GPL3 possano far muovere le menti per creare qualcosa di decisamente migliore del mio piccolo e semplice MASUW.

Floatman

Note finali di UnderAttHack:

Per informazioni, richieste, critiche, suggerimenti o semplicemente per farci sapere che anche voi esistete, contattateci via e-mail all'indirizzo underatthack@gmail.com

Siete pregati cortesemente di indicare se non volete essere presenti nella eventuale posta dei lettori.

Allo stesso indirizzo e-mail sarà possibile rivolgersi nel caso si desideri collaborare o inviare i propri articoli.

Per chi avesse apprezzato UnderAttHack, si comunica che l'uscita del prossimo numero (il num. 2) è prevista alla data di:

Venerdì 29 Maggio 2009

Come per questo numero, l'e-zine sarà scaricabile o leggibile nei formati PDF o xHTML al sito ufficiale del progetto: <http://underatthack.altervista.org>

Tutti i contenuti di UnderAttHack, escluse le parti in cui è espressamente dichiarato diversamente, sono pubblicati sotto [Licenza Creative Commons](#)

