FIREWALL E SISTEMI DI CONFERENCING REAL-TIME

Lo scopo di questa tesina è analizzare i problemi di sicurezza che possono insorgere utilizzando servizi di conferencing in tempo reale basati su Internet, ponendo l'accento su quello che si può e non si può fare con l'utilizzo di firewall.

I servizi di conferencing (che nel seguito tradurrò con servizi di conferenza) sono tutti quei servizi che permettono alle persone di interagire tra loro, in contrapposizione ai servizi di consultazione o di prelievo di materiale. Con tale termine non si designerà quindi solo audio e videoconferenza (oggi tanto di moda) ma anche servizi basati su testo come *talk* e *IRC*. Talk e IRC sono entrambi servizi di conferenza in tempo reale basati su testo, con la differenza che il primo apre una sessione tra due persone, mentre il secondo mette in collegamento tante persone contemporaneamente (un'altra differenza è che talk viene usato prevalentemente nelle reti locali, mentre IRC viene usato prevalentemente su Internet). Anche la posta elettronica e i newsgroup sono sistemi di conferenza, ma funzionano in modo asincrono e non in tempo reale (gli utenti coinvolti non sono per forza on-line), per cui non sono oggetto del nostro studio.

IRC ha fatto scuola e oggi come oggi sono molti i protocolli che offrono un servizio di conferenza multiutente in tempo reale basato su testo (ICQ, C6, NAPSTER e tanti altri). Tutti i servizi IRC-like presentano dal punto di vista dei firewall problematiche analoghe, per cui ci limiteremo a esaminare quello che rischia di rubare il primato di popolarità a IRC stesso: ICQ.

Daremo un'occhiata a MBONE (Multicast Backbone), un'infrastruttura nata su Internet con lo scopo di promuovere il multicasting (tecnologia quasi indispensabile per l'audio e la videoconferenza) e termineremo accennando a NetMeeting, la principale offerta Microsoft in ambito di conferencing real-time.

TALK

Talk è un sistema di conferenza in tempo reale per due persone basato su testo; permette a due persone di aprire una sessione di "chat". Ciascuno dei due ha lo schermo diviso in due sezioni; in una appare quello che si scrive, nell'altra quello che scrive l'altro.

Talk è molto complesso in quanto fa uso di UDP per negoziare la connessione tra i due siti e di TCP per muovere i dati avanti e indietro. UDP è usato tra il client chiamante e il server del chiamato e ancora tra il client dell'utente chiamato e il server del chiamante; TCP è usato tra i due client.

Per complicare ulteriormente le cose ci sono due versioni incompatibili del protocollo talk, chiamate talora *talk* e *ntalk* (new talk) e talora *otalk* (old talk) e *talk*, a seconda del modo di vedere di chi ve ne parla. La vecchia versione aveva un bug per cui solo macchine che usavano CPU con la stessa disposizione dei byte in memoria potevano parlarsi; la nuova versione ha fissato questo bug al costo di un'incompatibilità con la vecchia versione.

Il meccanismo di connessione esatto è il seguente. Il client dell'utente chiamante contatta il server dell'utente chiamato e, mentre attende la risposta, contatta anche il proprio server per informarlo che sta aspettando una chiamata e per informarlo della porta TCP su cui attenderà la connessione dall'altro client. Il server del chiamato comunica all'utente che il chiamante desidera stabilire una sessione di talk e lo informa di che riga di comando deve scrivere per rispondere. Questa riga di comando è qualcosa del tipo *" talk otheruser@otherhost"*, che fa partire il client del chiamato il quale contatterà il server del chiamante via UDP per informarsi sulla porta TCP su cui sta aspettando l'altro client. Ottenuta la risposta, il client del chiamato può finalmente stabilire la connessione col client del chiamante sulla porta TCP scelta da quest'ultimo.

Qui di seguito illustro schematicamente le operazioni con queste convenzioni: user1 è l'utente chiamante che chiama user2; user1 si trova su host1 (user2 su host2) e client1 e server1 sono il client e il server talk su host1 (client2 e server2 sono su host2).

1) User1 esegue client1 con una riga del tipo *"talk user2@host2"*
2) Client1 chiama server2 via UDP per richiedere la connessione
3) Client1 chiama anche server1 via UDP e lo informa su quale porta TCP attenderà
4) Server2 informa l'utente (user2) con un messaggio sulla console del tipo: *user1 is requesting a talk session - respond with: "talk user1@host1"*
5) User2 avvia il client2 con la riga di comando che gli è stata suggerita
6) Client2 contatta server1 via UDP chiedendo la porta TCP a cui si deve collegare
7) Server1 risponde a client1 via UDP fornendogli la porta che gli è stata comunicata al passo 3
8) Finalmente client2 stabilisce la connessione TCP con client1
9) Da questo momento c'è una comunicazione bidirezionale via TCP tra client1 e client2

Packet filtering

I server talk (che servono solo per mediare la connessione tra i client e poi escono di scena) usano la porta UDP 517 per le vecchie versioni di talk o la 518 per le nuove versioni. I client talk usano le porte TCP sopra alla 1023 per interagire tra loro. Questo significa che, se vuoi far passare talk attraverso il firewall, devi permettere connessioni TCP dove entrambe le parti stanno usando porte arbitrarie sopra alla 1023. Questo non è sicuro perché un hacker potrebbe approfittarne per portare un attacco servendosi di un server insicuro come quello di X11.

La tabella che segue descrive un set di regole che possiamo applicare al nostro sistema di packet filtering. Poiché l'azione di tutte queste regole è "Permit" (fai passare il pacchetto) è inteso che alla fine del set di regole (che comprenderà tra le altre quelle che elenchiamo qui) vi sia una regola che neghi tutto quello che non è esplicitamente dato (default deny stance).

#	Dir.	Src addr	Dst addr	Prot.	Src port	Dst port	ACK	Action
1	In	Ext	Int	UDP	>1023	517/518	-	Permit
2	Out	Int	Ext	UDP	517/518	>1023	-	Permit
3	Out	Int	Ext	UDP	>1023	517/518	-	Permit
4	In	Ext	Int	UDP	517/518	>1023	-	Permit
5	Out	Int	Ext	TCP	>1023	>1023	Any	Permit
6	In	Ext	Int	TCP	>1023	>1023	Any	Permit

1. un client esterno contatta il server interno (operazioni 2 e 6)
2. un server interno risponde a un client esterno (operazione 7)
3. un client interno contatta un server esterno (operazioni 2 e 6)
4. un server esterno risponde a un client interno (operazione 7)
5. un client interno comunica con un client esterno (operazioni 8 e 9)
6. un client esterno comunica con un client interno (operazioni 8 e 9)

Proxying

Non esistono proxies per talk. Anche se in linea di principio sarebbe possibile farne, fornendo client modificati che si appoggino, sia nella connessione TCP sia nella negoziazione iniziale via UDP, a un server che si collega all'altro client o all'altro server fingendo di essere lui stesso un client, di fatto il gioco non ne vale la candela. Attualmente, si tende ad usare molto talk nelle reti locali e quasi mai su Internet, ed è probabile che chi ancora lo usa in questo modo lo abbandonerà presto in favore di IRC o ICQ o qualcosa di analogo (entrambi i protocolli citati permettono conversazioni private tra due o più utenti).

Raccomandazioni finali

Poiché è impossibile far passare talk attraverso un sistema di packet filtering ed è impossibile farne un proxy, il consiglio è quello di non permettere l'uso di talk attraverso Internet. Se c'è una assoluta necessità di usarlo in questo modo, allora l'unica soluzione è quello di metterlo in una macchina che faccia da vittima sacrificale, ossia un host non sicuro esterno alla protezione del firewall (sul perimetro della rete) su cui gli utenti possano fare login.

INTERNET RELAY CHAT (IRC)

IRC è un sistema di conferenza in tempo reale basato su testo e multiutente.Gli utenti utilizzano programmi client IRC per collegarsi ai server IRC. I server sono connessi tra loro e si parlano per trasmettersi i messaggi di tutti i client, in modo da dare l'illusione di un unico ambiente virtuale (seppur diviso in canali). I client possono collegarsi indifferentemente a un qualsiasi server.

La maggior parte dei problemi di sicurezza sono causati dal modo in cui viene utilizzato IRC (e da chi!) e non da problemi del protocollo. Molti client permettono un accesso a risorse (file, processi, programmi,…) maggiore di quanto dovrebbero; un server malintenzionato può fare grossi danni se trova un client mal programmato o mal configurato. In più, alcuni degli utenti che frequentano regolarmente IRC hanno la cattiva abitudine di cercare di convincere i nuovi utenti a eseguire comandi che provocheranno danni ai loro sistemi.

Molti utenti IRC ben intenzionati, del resto, hanno una concezione naif della sicurezza. Per esempio, è normale vedere alcuni utenti distribuire software mettendo un piccolo server sulla loro macchina e informando tutti che è possibile scaricarlo con un comando del tipo "telnet myhost myport | sh". Tale comando permette sì di poter scaricare il software senza interazione dell'utente, ma permette anche di entrare nella macchina come utente autorizzato e di eseguire comandi che potrebbero danneggiare il sistema.

Utenti malintenzionati e utenti naif possono essere altrettanto pericolosi e un responsabile della sicurezza che decida di autorizzare l'utilizzo di IRC dovrebbe prendersi la briga di informare i propri utenti di non eseguire comandi, dentro o fuori al client IRC, suggeriti loro da estranei sui canali IRC.

Sebbene questi siano problemi non indifferenti per IRC, IRC fornisce un servizio utile e diffuso che permette alle persone di comunicare tra loro in tutto il mondo, per cui vale comunque la pena supportarlo, anche in considerazione del fatto che ha molti vantaggi propri della teleconferenza a un costo molto più basso.

Eseguire un server IRC in un ambiente ristretto su un bastion host non è una buona idea, visto che non è prudente permettere il login degli utenti su un bastion host. E' comunque fattibile far girare all'interno del firewall un IRC solo per uso interno e senza connessione verso l'esterno.

Molti client IRC supportano un meccanismo chiamato Direct Client Connections (DCC). Il DCC permette a due client IRC di comunicare tra loro direttamente senza passare dai server, utilizzando la connessione normale solo per la negoziazione iniziale.

Packet filtering

IRC è un servizio basato su TCP. I server si mettono di solito sulla porta 6667 per ricevere chiamate sia dai client sia dagli altri server (alcuni server comunque usano altre porte). I client e i server che contattano altri server usano le porte sopra alla 1023, cosa che fanno anche i client che fanno uso del meccanismo DCC.

Nel meccanismo DCC, il client chiamante passa un invito al client chiamato attraverso i normali canali IRC. L'invito include un numero di porta TCP dove il client chiamante si metterà ad aspettare. Il client chiamato, se sceglie di accettare l'invito, apre una connessione TCP a quella porta.

#	Dir.	Src addr	Dst addr	Prot.	Src port	Dst port	ACK	Action
1	In	Ext	Int	TCP	>1023	6667	Any	Permit
2	Out	Int	Ext	TCP	6667	>1023	Yes	Permit
3	Out	Int	Ext	TCP	>1023	6667	Any	Permit
4	In	Ext	Int	TCP	6667	>1023	Yes	Permit
5	In	Ext	Int	TCP	>1023	>1023	Any	Permit
6	Out	Int	Ext	TCP	>1023	>1023	Yes	Permit
7	Out	Int	Ext	TCP	>1023	>1023	Any	Permit
8	In	Ext	Int	TCP	>1023	>1023	Yes	Permit

1. Il client o server esterno contatta il server interno
2. Il server interno risponde al chiamante esterno
3. Un client/server interno contatta un server all'esterno
4. Il server esterno risponde al chiamante interno
5. Il client esterno esegue una connessione DCC col client interno in risposta al suo invito
6. Connessione DCC richiesta dal client interno
7. Il client interno esegue una connessione DCC col client esterno in risposta al suo invito
8. Connessione DCC richiesta dal client esterno

La regola 6 è in realtà inglobata nella 7, mentre la regola 8 è inglobata nella 5. Si noti che le 4 regole legate alle connessioni DCC (o anche solo le regole 5 e 7) permettono in realtà qualunque tipo di connessione su TCP al di sopra delle porte 1023 aprendo diversi buchi nella sicurezza del firewall.

Proxying

Ogni server IRC è un server proxy. Per configurare IRC per il proxying, basta mettere l'IRC nell'host che si desideri faccia proxying e far puntare i client interni a questo server.

Considerazioni finali

- Anche se è possibile fare proxy IRC o permettere IRC attraverso i filtri non è probabilmente una buona idea, vista la debolezza dei client. La soluzione migliore è quella di permettere

agli utenti di fare IRC su una macchina che sta sul confine della rete e al di fuori della protezione del firewall (una vittima sacrificale).

- Se si installa un server IRC interno, assolutamente non collegarlo ai server esterni. Se si fa così questo server diventerà un proxy per i vostri utenti per accedere al mondo esterno e consentirà loro di farsi attaccare dall'esterno
- Se malgrado le considerazioni fatte si decide di far passare IRC lungo il firewall, è impossibile permettere ai client interni di richiedere connessioni DCC a client esterni (regola 5). Si può invece permettere ai client interni di stabilire connessioni richieste da client esterni (regola 7), se si ha una certa fiducia nei propri utenti (che hanno la possibilità di stabilire una qualsiasi connessione TCP tramite una porta sopra alla 1023 con qualsiasi macchina).

ICQ

ICQ (che in inglese suona come "I seek you") è un servizio di conferenza multiutente in tempo reale basato su testo e molto di più (come vedremo). Come IRC, è costituito da una rete di server diffusi su Internet. La prima volta che ci si collega ad un server col client apposito, ci si registra lasciando i propri dati personali (first, last e nickname, gender, e-mail, …), quindi si riceve dalla procedura di registrazione un codice utente (il cosiddetto ICQ number) che identificherà univocamente l'utente. Sebbene sia possibile effettuare ricerche per nome, per cognome, per nickname e per e-mail, nessuno di questi campi garantisce l'unicità della persona cercata, e quello che si otterrà sarà una lista di utenti che fanno match con i parametri di ricerca specificati, ciascuno identificato dal proprio ICQ number.

La prima operazione che si effettua una volta registrati è solitamente quella di compilare una lista di amici, attraverso le procedure di ricerca viste sopra; tale lista resta sempre visibile nella finestra principale del programma ed è suddivisa tra la lista degli utenti attualmente connessi e quella degli utenti non connessi o di cui non si conosce lo stato di connessione (esiste una configurazione del client che permette di non far conoscere il proprio stato).

I vantaggi di ICQ rispetto a IRC sono una maggiore varietà di servizi e un client unico per tutti i sistemi operativi programmato dalla Mirabilis (l'azienda che ha inventato e sostiene il protocollo).

Che il client sia unico è un bene sia perché non si corre il rischio di incorrere in un client malprogrammato e poco configurabile come con IRC (dove accanto a pochi buoni programmi ci sono tanti programmi mediocri), sia perché si ha una consistenza nell'interfaccia grafica e non si deve reimparare ad usare un programma nuovo tutte le volte che si cambia sistema (laddove i client IRC sono testuali, c'è una certa consistenza nell'interfaccia a comandi, mentre per i client grafici è una vera e propria giungla). La giovinezza di ICQ fa sì comunque che sia disponibile solo per i sistemi operativi più comuni (Windows, Macintosh, Linux), mentre chi usa un sistema operativo più esotico potrebbe trovarsi nella condizione di non poterlo usare (mentre IRC è universalmente diffuso).

Per quanto riguarda la varietà di servizi, i servizi di ICQ che tutti i suoi utenti usano sono essenzialmente la chat (che può essere multiutente), il trasferimento di file e il sistema di messaggistica, che permette di inviare un messaggio a un utente o a un gruppo di utenti; tale sistema lavora proprio come la posta elettronica, permettendo a un utente che si connette di scorrere tutti i messaggi che gli sono arrivati mentre era off-line.

ICQ mette poi a disposizione tutta una serie di servizi minori quali:

- la possibilità di creare comunità ICQ basate su un comune interesse (specie di newgroup)
- un blocco note, un reminder (ossia un calendario che ricorda gli avvenimenti) e una todo list che verranno registrate nei server in modo da essere sempre a disposizione, indipendentemente dalla macchina usata

- la possibilità di far passare una connessione ICQ attraverso una normale chiamata telefonica tra due persone (servizio “Follow me”)
- la possibilità di un sistema automatico di risposta ai messaggi

ed altri ancora. In più, è possibile per terze parti fornire dei plug-in per ICQ che permettono di estendere ulteriormente le capacità del protocollo. Attualmente esistono plug-in per fare giochi on-line (Quake, Canasta, …), un sistema che imita in tutto e per tutto il funzionamente di IRC (IrCQ-Net), plug-in di video e audio conferenza (in maniera da utilizzare la rete come telefono o come videotelefono), un plug-in che permette di interfacciarsi a Microsoft NetMeeting ed altri ancora.

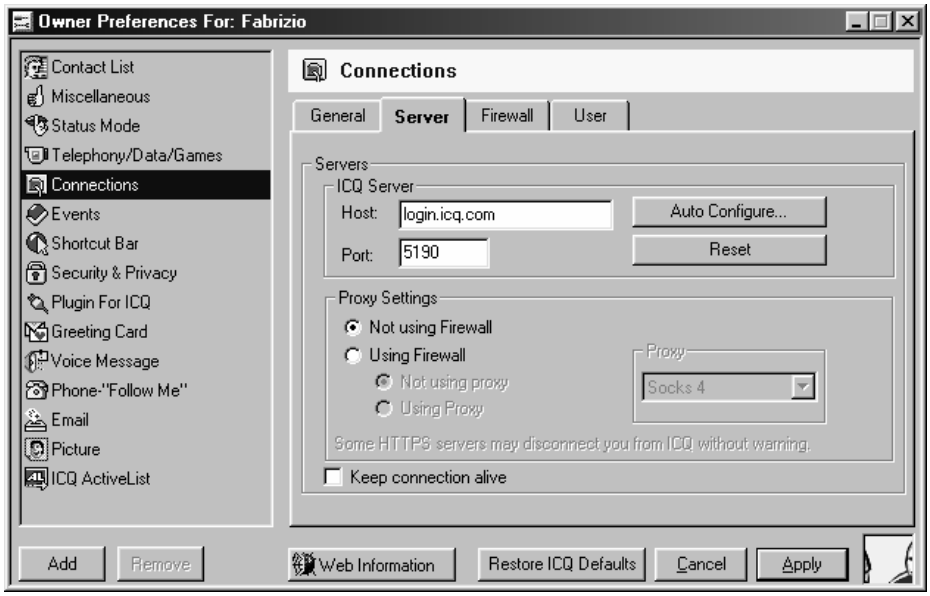
Non vogliamo qui entrare nei dettagli dei singoli servizi ma è chiaro come ICQ, come ogni prodotto moderno, si sia evoluto oltre i servizi inizialmente offerti tanto da sembrare di voler perseguire lo scopo di divenire un protocollo factotum, capace di soddisfare tutti i bisogni che spingono a fare conferenza su Internet (vedremo che questo è ancora più vero per Microsoft NetMeeting).

La nuovissima versione (ICQ 2000-a Beta Version) ha introdotto il supporto per i firewall, sia nelle capacità di configurazione del client, sia in forma di supporto tecnico (c’è una documentazione che spiega come configurare ICQ da dietro un firewall e un’e-mail di supporto tecnico per amministratori di rete). Nei prossimi due paragrafi esaminerò i consigli della ICQ Inc., mentre nelle considerazioni finali trarrò qualche conclusione finale mia.

Packet filtering

ICQ è un protocollo TCP-based che ha due tipi di connessioni: client-to-server (connessione normale) e Direct Connection (client to client).

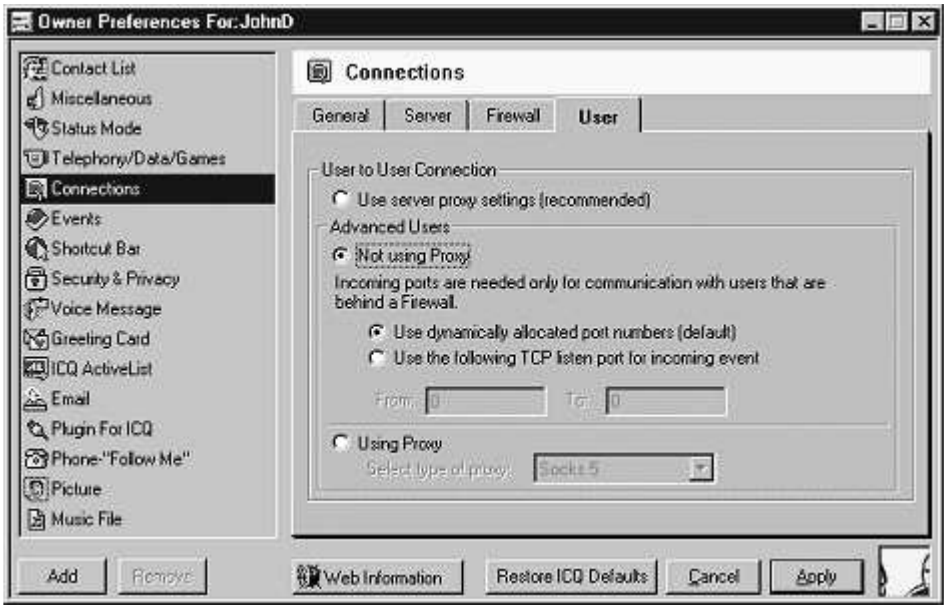
Le comunicazioni client-to-server avvengono collegandosi sulla porta TCP 5190 all’host login.icq.com. Un’annotazione della documentazione per amministratori di rete specifica di concedere connessioni bidirezionali verso login.icq.com e non a un indirizzo IP specifico, poiché tale hostname corrisponde a svariati indirizzi IP (bisogna avere un servizio DNS correttamente configurato).



Le comunicazioni client-to-client sono effettuate usando le porte TCP >1023. Poiché la maggior parte dei firewall che fanno packet filtering tendono ad escludere questa possibilità, il sito della ICQ Inc. prende in considerazione due possibili soluzioni:

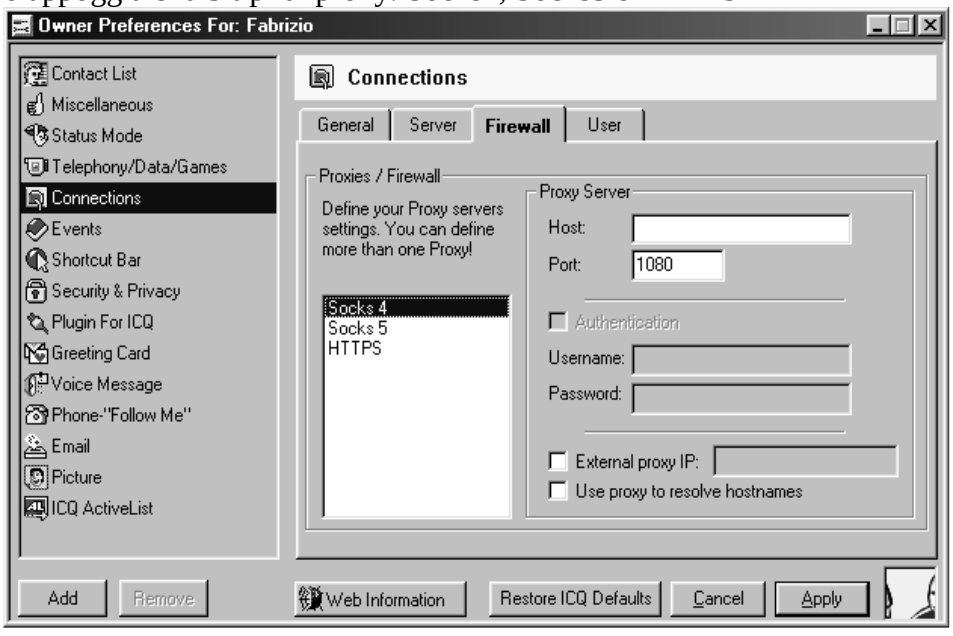
1. Restringere il range delle porte TCP attraverso cui è possibile fare connessioni ICQ
2. Usare un proxy

Qui di seguito mostro la finestra di configurazione del programma per la connessione client-to-client:



Proxying

ICQ può appoggiarsi a 3 tipi di proxy: Socks4, Socks5 e HHTPS



SOCKS4 e SOCKS5 sono dei protocolli che ritrasmettono sessioni TCP a un host del firewall per permettere agli utenti un accesso trasparente attraverso il firewall; è possibile effettuare un controllo di accesso all'inizio di ogni sessione TCP, dopodiché il proxy server non fa altro che ritrasmettere i dati dei pacchetti successivi tra il client e il server dell'applicazione.

Possono essere utili se il packet fltering di un firewall blocca i pacchetti TCP in ambo le direzioni o non permette di aprire porte in attesa. Nel primo caso, ICQ potrà funzionare nella modalità client-to-server ma non nella modalità client-to-client; nel secondo caso ICQ lavorerà ma gli utenti non

saranno capaci di comunicare con altri utenti nella stessa situazione (infatti nessuno potrà chiamare l'altro).

In questi casi il proxy server rappresenta un'alternativa in quanto, ponendosi come intermediario tra il client e l'ICQ server, garantisce una certa sicurezza. Si ricordi che il proxy server deve essere lasciato libero di aprire connessioni e riceverle.

Considerazioni finali

ICQ è a mio avviso molto pericoloso, per svariati motivi:

- È molto complesso da configurare, per cui rende molto probabile una cattiva configurazione da parte degli utenti e una loro conseguente facile esposizione agli attacchi
- È una realtà continuamente in movimento; se la nuova versione (ICQ 2000-a Beta Version) ha introdotto il supporto dei firewall e molti gingilli relativi alla sicurezza è anche vero che una nota della documentazione precisa che molte cose lì descritte non valgono per le versioni per gli altri sistemi operativi, per le versioni Java e per le vecchie versioni di Windows. Anche avendo l'ultima versione e trovandosi sul sistema operativo giusto (Windows), si ha pur sempre in mano un prodotto fatto da programmatori che hanno appena cominciato ad interessarsi di sicurezza e che però è abbastanza popolare da ricevere tutta l'attenzione possibile da parte degli hacker
- La grande estendibilità che deriva dalla possibilità di aggiungere plug-in, se da un lato amplia molto le capacità del servizio, dall'altro fa si che sia molto facile fare grossi danni, soprattutto su client configurati male o non configurati (la stragrande maggioranza degli utenti tenderà ad utilizzare la configurazione standard)

Di conseguenza il mio consiglio è quello di evitare i suggerimenti della ICQ Inc. e di non permettere l'utilizzo di ICQ all'interno di un firewall (a meno che non sia per un uso interno), ma di usare, proprio come per IRC, una vittima sacrificale sul bordo della rete su cui gli utenti possano fare login.

MBONE

Il multicasting è la possibilità che ha un host in certe tipologie di rete di inviare lo stesso messaggio a un gruppo di altri host. Si contrappone da un lato all'unicasting, la possibilità per un host di contattare un secondo host, dall'altro al broadcasting, la possibilità per un host di inviare un messaggio a tutti gli host della rete.

Il multicasting è particolarmente utile con applicazioni a largo consumo di banda, come audio e videoconferenza. Se si utilizza l'unicasting, occorrerà replicare i dati tante volte quanti sono i riceventi; ciascuna occorrenza dei dati percorrerà l'intero percorso dal trasmittente fino ai riceventi, limitando in maniera significativa il numero massimo di riceventi che la rete potrà sostenere. D'altra parte, se si utilizza il broadcasting, il consumo di banda sarà talmente elevato che una sola trasmissione rischierà di sovraccaricare l'intera rete.

Il mulitcasting IP fornisce un meccanismo per inviare pacchetti a gruppi di host IP. Con questo meccanismo si assegna un indirizzo IP a una particolare trasmissione; tutti gli host che vogliono riceverla condivideranno quella trasmissione. Gli indirizzi IP riservati al multicasting vanno da 224.0.0.0 fino a 239.255.255.255. Questi indirizzi possono essere usati solo come indirizzi destinazione, per cui non sono mai indirizzi sorgente validi (i pacchetti multicast partono da un indirizzo IP regolare dell'host trasmittente).

Alcuni gruppi multicast sono permanenti e assegnati sempre agli stessi usi: ad esempio la NASA ha un gruppo che utilizza per inviare video ogni volta che lo space shuttle è in orbita, c'è un gruppo per i meeting della Internet Engineering Task Force (IETF) e così via. Altri gruppi multicast sono temporanei, usati magari per singoli avvenimenti e poi riassegnati.

Il multicasting viene oggi utilizzato in Internet soprattutto per servizi di conferenza in tempo reale (audio e videoconferenza). Si comincia ad utilizzarlo anche per altri motivi, come l'invio efficiente dei newsgroup a un vasto gruppo di persone.

I prodotti commerciali (router e host) stanno cominciando ora a supportarlo. Alcune tecnologie come le reti Ethernet supportano il multicast direttamente. Altre tecnologie di rete non lo supportano, così occorre simularne il comportamento duplicando il pacchetto multicast in una serie di pacchetti unicast, per ciascuno dei riceventi.

Nel supportare il multicasting in una rete di reti (quale può essere Internet), alcune capaci di multicasting, altre solo di unicasting, si cercherà ove possibile di minimizzare la duplicazione di questi pacchetti unicast. Un approccio comune per collegare due reti capaci di multicast (come possono essere le Ethernet) attraverso una rete capace solo di unicast (come può essere una linea T1) è di creare un *tunnel* attraverso la rete unicast, mediante l'uso di multicast router (chiamati anche *mrouter*) alle due estremità del tunnel. Questi dispositivi prendono il pacchetto multicast IP in arrivo, lo incapsulano in un normale pacchetto IP unicast e lo inviano attraverso il tunnel all'altro mrouter, che lo disincapsulerà ritrasformandolo in un pacchetto multicast.

MBONE (Multicast Backbone) è, come dice il nome, la struttura portante del multicast su Internet, ed è una specie di web di mrouter e tunnel. I suoi partecipanti sono siti interessati al multicasting per una varietà di servizi su Internet.

Il multicasting IP solleva diverse questioni di sicurezza che riguardano i firewall. Un sito può partecipare a MBONE o attraverso un tunnel o con una tecnologia che utilizza direttamente il multicasting.

Se un sito utilizza un tunnel, dovrà incapsulare un pacchetto IP multicast in un normale pacchetto IP unicast. Il sistema di packet filtering deve allora essere progettato per riconoscere questa incapsulamento di un pacchetto IP dentro a un altro. Alcuni sistemi di packet filtering non riconoscono l'IP-in-IP per nome ma attraverso il numero di protocollo, che è 4 (ICMP è 1, TCP è 6, UDP è 17). I tunnel multicast IP usavano i pacchetti IP source-routed, ma questa pratica causa problemi (non ultimo quelli relativi ai firewall) per cui ora viene sconsigliata.

Se invece utilizza una tecnologia adatta al multicasting, il packet filtering non presenta grosse difficoltà, in quanti i pacchetti di multicast IP sembrano normali pacchetti IP con indirizzi particolari (da 224.*.*.* a 239.*.*.*), di conseguenza vanno trattati come gli altri. Per esempio, se un responsabile della sicurezza adotta una politica di default deny stance, li negherà tutti di default per poi far passare solo quelli che conosce e che sa cosa fanno. In ogni caso occorre un po' di attenzione perché il pacchetto multicast sarà diretto a molte macchine interne, e ciascuna di queste macchine avrà bisogno di protezione dagli attacchi esterni.

Anche se il tunnel è ristretto solo ai pacchetti multicast o se si sta utilizzando il multicast direttamente senza tuneling resta il problema di come gli host risponderanno a indirizzi riservati al multicast verso porte regolari. Sfortunatamente il comportamento varia da sistema operativo a sistema operativo e da release a release dello stesso sistema operativo. Occorre assicurarsi che il sistema operativo sia progettato per riconoscere gli indirizzi riservati oppure installarsi da soli estensioni per il multicast.

MICROSOFT NETMEETING

Microsoft NetMeeting è il programma di conferenza real-time proposto da Microsoft; distribuito insieme al sistema operativo, si può scaricare gratuitamente da Web. Qui conferencing va inteso nella concezione più ampia possibile: la Microsoft lo ritiene adatto sia per il mercato consumer (da usarsi per esempio come videotelefono) sia come strumento di produttività aziendale.
Con NetMeeting si può:

- Fare video e audio conferenza

- Usare una lavagna (whiteboard) comune su cui disegnare (una specie di Paint multipagina)
- Usare un sistema di chat multiutente (multipoint secondo la terminologia Microsoft)
- Consultare elenchi di persone iscritte al servizio
- Trasferire file
- Condividere programmi durante una conferenza (esecuzione remota)
- Controllare un computer in remoto vedendone il desktop
- Collegarsi direttamente via telefono a un altro utente

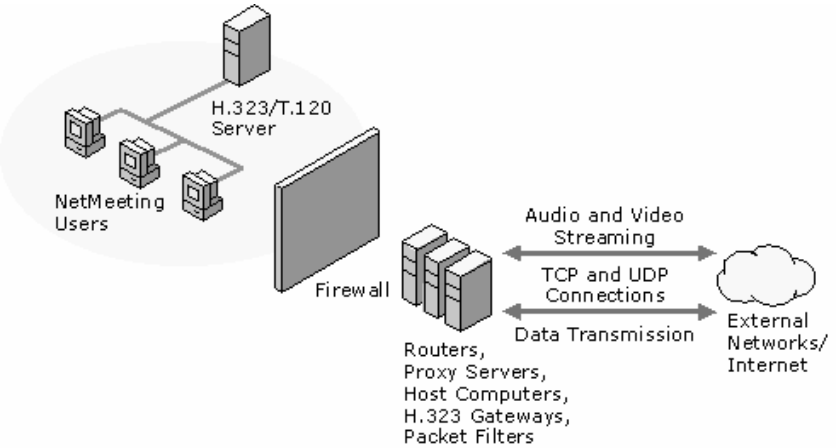
Fa quindi praticamente tutto quello che fa ICQ ed anche di più (ad esempio su ICQ non è possibile il controllo remoto, perlomeno non ho letto di nessun plug-in che lo faccia) e si propone come il tool definitivo di conferencing su Internet… certo, avendo un sistema operativo Microsoft.
Le informazioni che seguono sono tratte direttamente dalla documentazione di NetMeeting

Packet filtering

Microsoft NetMeeting è un pacchetto complesso che fa uso di diversi protocolli per funzionare. Ciascuno di questi protocolli fa uso di porte TCP o UDP.

Port	Function	Outbound Connection
389	Internet Locator Service (ILS)	TCP
522	User Location Service	TCP
1503	T.120	TCP
1720	H.323 call setup	TCP
1731	Audio call control	TCP
>1023	H.323 call control	TCP
>1023	H.323 streaming	Real-Time Transfer Protocol (RTP) over UDP

T.120 e H.323 sono entrambi due protocolli definiti dalla International Telecommunications Union. Il primo supporta lo scambio dei dati e la conferencing real-time multipoint; NetMeeting lo usa ad esempio per la whiteboard e la condivisione di file. Il secondo è specifico per la video e l'audio conferenza.



ILS (Internet Locator Service) è uno standard basato su LDAP che permette di trovare utenti su Internet facendo ricorso a liste generate dinamicamente ed è quello che NetMeeting usa per fare la lista di utenti attualmente connessi.

Affinché NetMeeting possa passare attraverso il firewall per una connessione verso l'esterno devono essere impostate le seguenti cose:

- Devono essere permesse connessioni TCP verso l'esterno sulle porte 389, 522, 1503, 1720 e 1731
- Devono essere permesse comunicazioni TCP e UDP bidirezionali sulle porte >1023

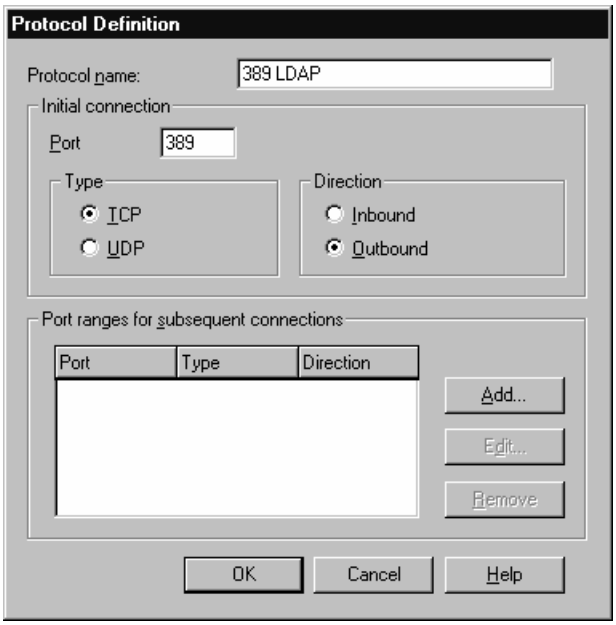
Il call setup protocol H.323 usa la porta 1720 per negoziare quale porta TCP >1023 dovrà utilizzare per il controllo dei dati (il call control protocol). Il call setup protocol e l'audio control protocol vanno poi a negoziare su quali porte UDP dovrà passare lo streaming dei dati; il primo negozierà le due porte UDP relative alla trasmissione video (una per direzione), il secondo negozierà le due porte UDP relative all'audio (una per direzione).

Poiché l'uso delle porte TCP e UDP al di sopra della 1023 riguarda la video e audioconferenza, filtrando questo tipo di pacchetti sarà ancora possibile utilizzare tutti gli altri servizi di NetMeeting (controllo remoto, chat, whiteboard,

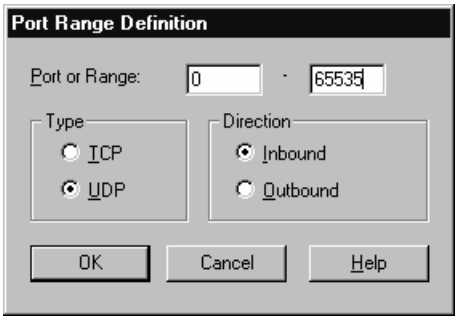
Proxying

La manualistica Microsoft parla di come configurare Microsoft Proxy Server per funzionare con NetMeeting.

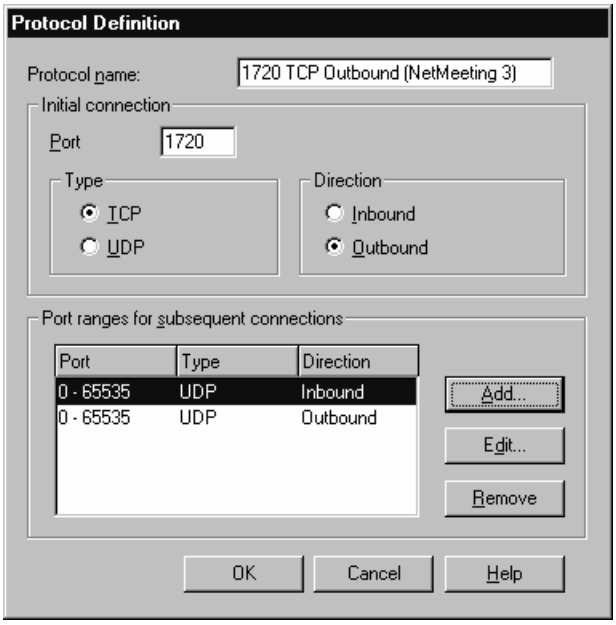
Questo è un proxy server generico multiprotocollo. Per ogni protocollo (scelto nella lista di quelli che supporta) è possibile specificare la porta su cui risiede il server, se sia basato su TCP o su UDP, la direzione che si desidera supportare (per supportare connessioni in ambo le direzioni occorre aggiungere due volte lo stesso protocollo) e il range di porte che si devono usare in quelle che la Microsoft chiama le connessioni successive alla prima (che in realtà dovrebbero essere i pacchetti successivi al primo nella stessa connessione TCP o nella stessa comunicazione UDP).



Il manuale di NetMeeting dice di specificare un range di porte tra 0 e 65535 per tutte le connessioni UDP; specificando il range di porte si apre una seconda finestra che chiede direzione e tipo di pacchetti.



Di seguito fornisco un esempio di come viene configurato il proxy per la connessione alla porta 1720 (H.323 call setup). Si ricordi che tale connessione serve per negoziare le due porte UDP (una interna al firewall e una esterna) su cui avverrà il passaggio dei dati video



Considerazioni finali

Abbiamo visto come sia pericoloso far passare attraverso il packet filtering system i protocolli che gestiscono l'audio e video conferenza, mentre col proxying dovrebbe essere possibile un controllo più fine e quindi, se si opta per il proxying, non si ha più questo problema.
D'altro canto, certe funzioni di NetMeeting (condivisione di programmi, controllo remoto), sono molto pericolose e non dovrebbero essere permesse su macchine che abbiano dati sensibili. Ogni funzione di NetMeeting può essere esclusa o limitata in una maniera molto raffinata (ci sono 20 pagine di manuale per questo), ma in generale un responsabile della sicurezza deve pensare che l'utente medio tenderà a tenere tutto abilitato, sia per la pigrizia di non cambiare le impostazioni di default e sia per avere un ambiente di lavoro migliore. La soluzione ottimale è ancora una volta la vittima sacrificale, ossia una macchina al bordo della rete messa lì apposta per supportare questo genere di servizi.

BIBLIOGRAFIA

Libri

- D. Brent Chapman and Elisabeth D. Zwicky: "Building Internet Firewalls", O'Reilly Associates (November 1995)
- Kevin Savetz, Neil Randall and Yves Lepage: "MBONE: Multicasting Tomorrow's Internet", IDG (1996, 1998)

Siti internet

- www.icq.com
- www.mbone.com
- www.microsoft.com/windows/NetMeeting/