Sicurezza nelle applicazioni client-server
Servizi sicuri

Parlando di servizi sicuri, tipicamente ci si riferisce a servizi che forniscono due tipi di garanzie:

- il servizio non può essere utilizzato in nessun modo se non per le operazioni previste.
- non è possibile leggere e/o falsificare le transazioni che avvengo attraverso il servizio.

Tali garanzie non implicano che si possano eseguire transazioni con il servizio continuando ad essere al sicuro. Per esempio, si potrebbe utilizzare un HTTP (*HyperText Transfer Protocol*) sicuro per effettuare il *download* di un *file*, ed essere sicuri che si stia effettivamente effettuando il *download* del *file* a cui si è interessati, e che nessuno lo stia modificando nel transito. Ma non si possono avere garanzie che il *file* non contenga dei virus o programmi dannosi.

È possibile anche utilizzare servizi insicuri in modo sicuro, ma ciò richiede maggiore cautela. Ad esempio, la posta elettronica attraverso il protocollo SMTP (*Simple Mail Transfer Protocol*) è un classico esempio di un servizio insicuro.

Tutte le volte che si valuta la sicurezza di un servizio, bisogna contestualizzare le valutazioni al proprio ambiente e tener conto delle proprie configurazioni; non è interessante la sicurezza in astratto di un servizio.

Il World Wide Web

Il *World Wide Web* è divenuto così popolare che molte persone pensano che sia Internet stesso. Se non si appartiene al *Web*, non si è nessuno. Sfortunatamente, sebbene il *Web* si basi principalmente su un singolo protocollo (HTTP), i siti *Web* utilizzano spesso una varietà di protocolli.

Molte persone confondono le funzioni e le origini del *Web* con i *browser* (*Netscape, Microsoft Internet Explorer*), con un protocollo (HTTP) e con il linguaggio di pubblicazione dei documenti sui *server* Internet (HTML). Riteniamo importante quindi darne una sufficiente descrizione:

- Il *Web* consiste nell'insieme dei *server* HTTP in Internet.
- Il protocollo HTTP costituisce il protocollo principale su cui si basa il *Web*; consente agli utenti l'accesso ai documenti che sono resi disponibili dai *server Web*. Tali documenti possono assumere diversi formati (testo, immagini, audio, video, ecc.), ma il formato usato per fornire il collegamento tra essi è il linguaggio HTML (*HyperText Markup Language*).
- Il linguaggio HTML costituisce uno standard per la realizzazione di pagine *Web*. Fornisce delle funzionalità base per la formattazione dei documenti e per la definzione di link ipertestuali ad altre pagine e/o ad altri *server*.
- *Netscape Navigator* e *Microsoft Internet Explorer*, comunementi noti come *Netscape* ed *Explorer*, (anche altri *browser* sono discretamente diffusi: Lynx, Opera, Slurp, Go!Zilla e perlWWW) sono prodotti commerciali che realizzano il lato *client* dell'applicazione *Web*. Un *Web client*, chiamato anche *Web browser*, consente la lettura di documenti attraverso il protocollo HTTP oppure attraverso altri protocolli.

Sicurezza dei Web client

I *Web browser* forniscono una interfaccia grafica per un gran numero di risorse Internet. Le informazioni e i servizi che non erano disponibili o che erano accessibili solo agli esperti di informatica diventano grazie ai *browser* facilmente accessibili.

Sfortunatamente è difficile rendere sicuri i *Web browser* ed i *Web server*. L'utilità del *Web* è in larga

parte basata sulla sua flessibilità, ma tale flessibilità rende difficoltosi i controlli. Si pensi, ad esempio, quanto sia più facile trasferire ed eseguire un programma attraverso un *Web browser* rispetto al servizio FTP, ma si consideri anche la possibilità di trasferire ed eseguire un programma malizioso. I *Web browser* dipendono da programmi esterni, genericamente chiamati *viewer* (sono chiamati "visualizzatori" anche quando eseguono un brano sonoro invece di mostrare un'immagine), per gestire i tipi di *file* che non risultano decodificabili dal *browser*. Genericamente interpretano correttamente i principali tipi di *file* (HTML, il normale testo e le immagini in formato JPEG e GIF). *Netscape* ed *Explorer* attualmente supportano un meccanismo (progettato per rimpiazzare i *viewer* esterni) che consente a terze parti di produrre dei *plug-in* che possono essere scaricati per costituire una estensione del *Web browser*. Si deve fare molta attenzione a quali *viewer* vengono configurati e a quali *plug-in* vengono scaricati.

La maggior parte dei *Web browser* comprende uno o più linguaggi (*Java*, *Javascript* o *ActiveX*) che consentono di estendere le loro caratteristiche e le loro funzionalità. Questi linguaggi rendono i *Web browser* più potenti e più flessibili, ma introducono anche nuovi problemi. Mentre l'HTML è principalmente un linguaggio per la formattazione dei documenti, i linguaggi di estensione forniscono capacità di elaborazione locale delle informazioni, al pari di un linguaggio di programmazione. Tradizionalmente, quando si acquista un nuovo programma si sa da dove proviene e chi lo ha realizzato. Se si decide di copiare un programma da un sito Internet, non si hanno le stesse informazioni.

I progettisti di *Javascript*, *VBscript*, *Java* ed *ActiveX* hanno proposto diversi approcci per la soluzione a questo problema. Per quanto riguarda *Javascript* e *VBscript* si suppone semplicemente che non possano eseguire azioni dannose; tali linguaggi infatti non hanno, ad esempio, comandi per scrivere sul disco. *Java* usa un approccio chiamato *sandbox*. *Java* a differenza dei precedenti linguaggi contiene dei comandi che potrebbero essere dannosi, ma l'intrerprete *Java* blocca un programma non fidato ogniqualvolta tenta un'azione dannosa. Sfortunatamente, ci sono stati problemi di implementazione con *Java* e sono stati trovati diversi modi per eseguire delle operazioni che si credevano impossibili.

La tecnologia *ActiveX* invece di provare a limitare le possibilità di un programma, cerca di associare ad un programma le informazioni che ci consentono di individuare da dove e da chi proviene in modo da valutarne l'affidabilità. Tutto ciò è realizzato attraverso il meccanismo della firma digitale; prima dell'esecuzione di un programma *ActiveX* un *Web browser* mostra le informazioni relative alla firma digitale dell'autore del programma e l'utente può decidere se eseguirlo o meno.

Sicurezza dei Web server

L'attivazione di un *server Web* implica che tutta la comunità di utilizzatori che possono accedervi possano inviare dei comandi. Anche se il *Web server* è configurato per fornire solamente *file* HTML e quindi i comandi sono abbastanza limitati possono comunque verificarsi problemi di sicurezza. Ad esempio, molte persone ritengono che l'utente non possano vedere i *file* del *server* a meno che non esistano link espliciti ad essi; tale assunzione è generalmente falsa. È più corretto ritenere che se un *Web server* è in grado di leggere un *file*, è anche in grado di fornirlo ad un utente remoto. I *file* che non dovrebbero essere pubblicamente accessibili dovrebbero almeno essere protetti con dei permessi a livello di *file system*, e dovrebbero, se possibile, essere posti al di fuori dell'area di disco accessibile al *Web server*.

Molti *Web server*, inoltre, forniscono altri servizi rispetto alla semplice gestione di *file* HTML. Ad esempio, alcuni *Web server* forniscono dei servizi amministrativi che consentono all'amministratore del *Web server* di configurarlo attraverso un *Web browser* da remoto, senza necessità di lavorare sulla macchina *server*. Se l'amministratore del *Web server* può raggiungere il *server* attraverso un *browser*, chiunque può farlo; bisogna quindi essere certi che la configurazione iniziale del *server* sia impostata in un ambiente sicuro.

I *Web server* possono anche invocare l'esecuzione di programmi esterni in diversi modi. Tali programmi sono molto facili da scrivere ma molto difficile da rendere sicuri, poiché possono ricevere comandi arbitrari da utenti esterni. Il *Web server* non fornisce alcuna protezione significativa per tali programmi.

Protocolli HTTP sicuri

Allo stato attuale 2 protocolli forniscono *privacy* del contenuto all'HTTP usando meccanismi di cifratura e di autenticazione forte. Quello che comunemente viene adottato si chiama HTTPS ed è utilizzato inserendo nell'URL la parola chiave https. L'altro, per lo più sconosciuto, si chiama *Secure* HTTP ed utilizzato inserendo nell'URL la parola chiave shttp.

L'obiettivo del protocollo HTTPS è quello di proteggere il canale di comunicazione quando si ricevono o si spediscono dati. Attualmente HTTPS utilizza il protocollo SSL per ottenere tale obiettivo.

L'obiettivo del protocollo *Secure* HTTP è quello di proteggere i singoli oggetti che vengono scambiati piuttosto che il canale di comunicazione. Questo consente, ad esempio, che alcune pagine su un *Web server* possano essere associate ad una firma digitale e che un *Web client* possa verificare la firma al momento del *download* di tali pagine.

L'uso del *Secure HTTP* potrebbe avvantaggiare significativamente i consumatori nel mondo del commercio elettronico, infatti l'identità del consumatore e quella del venditore sono associate in maniera inscindibile agli oggetti che fanno parte di una transazione Secure HTTP, mentre nel caso del protocollo HTTPS l'identità del consumatore e quella del venditore sono associate al canale di comunicazione.

La posta elettronica

La posta elettronica e le *news* forniscono agli utenti un modo per scambiarsi informazioni senza la necessità di risposte interattive o immediate.

La posta elettronica è uno dei più popolari servizi di Internet. Solitamente è un servizio a basso rischio (ma non esente da rischi). Modificare la posta elettronica è banale, e le modifiche facilitano due differenti tipi di attacco:

- attacchi contro la propria reputazione;
- attacchi di manipolazione sociale (ad esempio, ad un utente viene inviato un messaggio da parte di un sedicente amministratore che lo invita ad impostare la propria *password* in un certo modo).

Accettare messaggi di posta elettronica significa consumare tempo di CPU e spazio di memoria e sul disco; anche su ciò si basano gli attacchi di tipo DoS (*Denial of Service*). In particolare, con gli attuali sistemi di posta elettronica multimediali, alcuni utenti possono spedire messaggi di posta elettronica contenenti dei programmi che possono essere eseguiti. All'interno di tali programmi possono celarsi dei *Trojan* (Cavalli di Troia).

Sebbene molti utenti si preoccupano degli attacchi diretti, in pratica, i problemi più comuni con la posta elettronica sono dovuti agli attacchi di *flooding* (incluse le cosiddette catene di Sant'Antonio) ed alle persone che inviano dati riservati fidandosi della confidenzialità del servizio. Se gli utenti sono informati adeguatamente ed il servizio di posta elettronica è isolato da altri servizi, in modo tale che gli attacchi DoS possano provocare il minor numero di danni, il servizio stesso è ragionevolmente sicuro.

Il protocollo SMTP (*Simple Mail Transfer Protocol*) è il protocollo Internet standard per spedire e ricevere posta elettronica. La posta che viaggia attraverso i *server* Internet viene gestita principalmente attraverso il protocollo SMTP. Il protocollo SMTP in sè non è un problema, i *server* SMTP invece possono esserlo. Un programma che consegna la posta agli utenti spesso deve essere in grado di accedere alle risorse dei singoli utenti.

Il più comune *server* SMTP in *Unix* è *Sendmail*. *Sendmail* è stato attaccato con successo in diversi modi, (si pensi, ad esempio al caso dell'Internet *worm*), scoraggia gli utenti ad usarlo. Comunque, molti *server* di posta che sono stati sviluppati per superare i problemi di *Sendmail* non sono certo migliori di *Sendmail*. L'evidenza suggerisce che subiscono meno attacchi perché sono meno popolari e non perché siano più sicuri o meno vulnerabili.

Il più comune *server* SMTP nei sistemi operativi della *Microsoft* è *Microsoft Exchange*; anche *Exchange* è stato più volte attaccato con successo.

Il protocollo SMTP viene utilizzato per scambiare messaggi di posta elettronica tra i *server*, e per trasferire la posta da un *client* al *server*. Per leggere la posta dalla propria casella ospitata da un *server*, gli utenti impiegano un protocollo diverso: i più utilizzati a questo scopo sono POP (*Post Office Protocol*) ed IMAP (*Internet Message Access Protocol*). *Mircosoft Exchange* e *Lotus Notes* utilizzano dei protocolli proprietari che forniscono caratteristiche aggiuntive.

I protocolli POP ed IMAP hanno le medesime implicazioni per quel che riguarda la sicurezza; entrambi trasferiscono i dati relativi all'autenticazione degli utenti ed al contenuto dei messaggi senza cifrarli, consentendo agli attaccanti di leggere la posta e di ottenere le credenziali degli utenti.

Usenet news

I *newsgroup* equivalgono a bacheche su cui si affiggono gli annunci e sono stati progettati per realizzare comunicazioni molti a molti. Anche le *mailing list* supportano le comunicazioni molti a molti, ma lo fanno meno efficientemente.

I rischi delle *news* sono molto simili a quelli della posta elettronica:

- gli utenti possono fidarsi inavvertitamente delle informazioni ricevute;
- gli utenti possono rivelare informazioni riservate;
- il proprio *server* delle *news* può subire attacchi di tipo DoS.

Poiché le *news* sono raramente un servizio essenziale, gli attacchi di tipo DoS contro un singolo *server* sono solitamente ignorati. I rischi di sicurezza delle *news* sono quindi abbastanza insignificanti.

Attualmente molti *Web server* consentono agli utenti di accedere al servizio delle *news* attraverso il protocollo HTTP. Questa soluzione non è molto efficiente se un numero elevato di utenti leggono le *news*.

Il protocollo NNTP (*Network News Transfer Protocol*) viene utilizzato per trasferire le *news* attraverso Internet. Nel configurare un proprio *server* delle *news*, si dovrà determinare il modo più sicuro per far giungere le *news* presso i sistemi interni in modo tale che il protocollo NNTP non possa essere utilizzato per attaccare i sistemi stessi.

Il trasferimento, la stampa e la condivisione dei file

La posta elettronica può essere utilizzata per trasferire dati da un sito ad un altro, ma è stata

progettata per piccoli *file* in forma testuale.

Anche se gli attuali sistemi di posta elettronica includono delle caratteristiche che consentono di trasferire ingombranti *file* in formato binario, suddividendoli in più parti codificate opportunamente dal mittente, e poi decodificati e riassemblati dal ricevente. Sfortunatamente, tali operazioni sono complesse e possono provocare facilmente degli errori. Inoltre, gli utenti sono interessati a cercare i *file* senza attendere che qualcuno spedisca loro ciò di cui necessitano. Per questi motivi, anche quando la posta elettronica è disponibile, è utile avere un metodo progettato per trasferire *file* a richiesta.

A volte, più che un trasferimento di *file* tra macchine, può essere interessante ed utile disporre di una singola copia del *file* e renderla accessibile a una serie di *client*. In questo caso si parla di condivisione. I protocolli per la condivisione dei *file* possono anche essere utilizzati come protocolli per il trasferimento dei *file*, ma principalmente consentono di usare un *file* come se fosse locale. Solitamente, la condivisione dei *file* risulta più conveniente per gli utenti, ma poiché offre maggiori funzionalità, è meno efficiente, meno robusta e meno sicura.

La stampa dei *file* è spesso basata sui protocolli per la condivisione o per il trasferimento dei *file*.

Il trasferimento dei file

Il protocollo FTP (*File Transfer Protocol*) è il protocollo Internet standard per i trasferimenti dei *file*. Molti *Web browser* supportano sia FTP che HTTP e usano automaticamente il protocollo FTP per accedere alle locazioni i cui nomi iniziano con ftp://[utente:*password*@]macchina.dominio/*file*.; in tal modo molti utenti usano il protocollo FTP senza neanche accorgersene. In teoria, consentire ai propri utenti il *download* di *file* non implica un aumento dei rischi rispetto all'uso della posta elettronica; infatti, alcuni siti offrono servizi che consentono agli utenti di accedere all'FTP attraverso la posta elettronica.

I principali problemi in molti siti sono relativi al fatto che gli utenti possono scaricare *software* contenente dei *Trojan*. Sebbene questo possa capitare, attualmente le maggiori preoccupazioni sono relative all'installazione di giochi per il *computer*, all'uso di *software* pirata o allo scambio di immagini pornografiche. Anche se questi non sono problemi direttamente collegati alla sicurezza, essi provocano una serie di altri problemi (incluso il consumo di tempo e di spazio su disco e l'introduzione di problemi di natura legale). Si consideri inoltre la possibilità di acquisire virus; i virus infatti potrebbero essere nascosti all'interno dei *file* che gli utenti copiano dal sito.

Seguendo le seguenti semplici regole si può essere sicuri che il traffico FTP in ingresso sia ragionevolmente sicuro:

- Gli utenti vengono educati a diffidare di qualunque *software* che possa essere scaricato attraverso l'FTP.
- Gli utenti vengono informati delle politiche relative al materiale sessuale e all'uso delle risorse dell'organizzazione a cui appartengono.

I servizi FTP anonimi sono un meccanismo estremamente popolare per consentire agli utenti l'accesso remoto ai *file* senza fornire loro la possibilità di avere un accesso completo alla propria macchina. Se si esegue un FTP *server* si può consentire agli utenti di reperire i *file* che sono stati collocati in un'area pubblica del proprio sistema senza consentire loro di accedere a qualunque risorsa del proprio sistema. L'area relativa al proprio *server* FTP anonimo può contenere gli archivi pubblici della propria organizzazione (articoli, *software*, immagini grafiche e informazioni di qualunque altro genere).

Per ottenere l'accesso ai *file* che sono stati resi disponibili, gli utenti effettuano il *log in* nel sistema

usando il servizio FTP e servendosi di un *login name* speciale (solitamente *anonymous* o ftp). Molti siti richiedono agli utenti che inseriscano per cortesia il proprio indirizzo di posta elettronica, in risposta al *prompt* della *password*, in modo tale il sito possa tracciare chi sta usando il servizio FTP anonimo, ma questo requisito può essere gestito raramente (soprattutto perchè non c'è un modo per verificare la validità di un indirizzo di posta elettronica).

Installando un servizio FTP anonimo, bisogna essere sicuri che gli utenti che lo usano non possano avere accesso ad altre aree o a *file* del sistema, e che non possano usare il servizio FTP per ottenere un accesso a livello di *shell* nel sistema stesso.

La condivisione dei file

Sono disponibili diversi protocolli per la condivisione dei *file*, che consentono ai *computer* di usare *file* che sono fisicamente collocati su dischi appartenenti fisicamente ad altri *computer*. Tutto questo è molto vantaggioso, perchè consente agli utenti di usare i *file* remoti senza l'*overhead* di trasferirli avanti e indietro e di cercare di mantenere le versioni sincronizzate. In ogni caso, la condivisione dei *file* è più complessa da implementare rispetto al trasferimento dei *file*. I protocolli per la condivisione dei *file* devono fornire trasparenza (il *file* sembra essere locale, e non ci si rende conto della condivisione) e completezza (si deve poter fare sul *file* remoto tutto ciò che si può fare su un *file* locale). Queste caratteristiche rendono vantaggiosa la condivisione dei *file*, ma la necessità di trasparenza pone dei limiti alla sicurezza, e la necessità di fornire completezza rende i protocolli complessi da implementare. Una maggiore complessità conduce inevitabilmente ad una maggiore vulnerabilità.

I protocolli per la condivisione dei *file* più comunemente utilizzati sono in ambito *Unix* il protocollo NFS (*Network file System*), in ambito *Microsoft* il protocollo CIFS (*Common Internet file System*) e in ambito *Macintosh* il protocollo *AppleShare*. Il protocollo CIFS fa parte di una famiglia di protocolli tra i quali SMB (*Server Message Block*), NetBIOS/NetBEUI e *LanManager*. Sono simili fra loro, in larga parte intercambiabili, e presentano gli stessi problemi per quel che riguarda la sicurezza.

Il protocollo NFS è stato progettato per essere utilizzato in area locale e assume che ci siano bassi tempi di risposta, elevata affidabilità, sincronizzazione temporale ed un elevato grado di fiducia tra le macchine. Se non si configura propriamente il protocollo NFS, un attaccante può essere in grado di accedere facilmente al *file system*. Con il protocollo NFS le macchine client hanno la possibilità di leggere e cambiare i *file* memorizzati in un *server* senza avere la necessità di effettuare il *login* sul *server* o inserire una *password*. Poichè il protocollo NSF non effettua il *logging* delle transazioni, potrebbe accadere che non si scopra mai che qualcuno ha accesso completo ai propri *file*. Il protocollo NSF fornisce un modo che consente di controllare quali macchine abbiano accesso ai propri *file*. Un *file* chiamato /etc/exports ci consente di specificare a quali *file system* si può accedere e quali macchine possano accedervi. Se si lascia un *file system* al di fuori del *file* /etc/exports, nessuna macchina potrà accedervi. Se lo stesso *file* si inserisce in /etc/exports, ma non si specifica quali macchine possono accedervi allora si consente a qualunque macchina di accedervi. Il protocollo NFS ha meccanismi di autenticazione dei *client* molto deboli, e un attaccante può essere in grado di convincere un *server* NFS che una richiesta proviene da uno dei *client* elencati nel *file* /etc/exports. Ci sono anche delle situazioni in cui un attaccante può effettuare l'*hijacking* di accessi NFS esistenti.

Entrambi i protocolli CIFS e *AppleShare* fanno affidamento su meccanismi di autenticazione a livello di utente, invece che a livello di *host*. Tale approccio costituisce un miglioramento per quel che riguarda la sicurezza. *AppleShare* tuttavia non è in grado di supportare dei metodi flessibili per l'autenticazione degli utenti. Si è costretti ad utilizzare *password* riusabili, il che significa che un attaccante può semplicemente limitarsi a catturare le *password*. Il protocollo CIFS fornisce buoni meccanismi di autenticazione e protezione nelle sue versioni recenti. Comunque, le caratteristiche

di compatibilità all'indietro del protocollo CIFS aumentano la sua vulnerabilità. Inoltre, allo stato attuale il protocollo CIFS fornisce una famiglia completamente nuova di servizi, alcuni dei quali più vulnerabili dei servizi relativi alla condivisione dei *file*.

Alcuni protocolli per la condivisione dei *file* sono stati progettati per essere utilizzati su reti quali Internet; ad esempio, il protocollo AFS (*Andrew file System*) usa Kerberos per l'autenticazione ed opzionalmente la cifratura. I protocolli NFS, CIFS ed *AppleShare* sono tutti parte di popolari sistemi operativi, mentre AFS è un prodotto di terze parti. A causa di questo e poichè AFS e Kerberos richiedono un'esperienza tecnica significativa per essere installati e manutenuti, il protocollo AFS non viene molto utilizzato.

La stampa dei file

Quasi tutti i sistemi operativi oggi forniscono la possibilità di stampare da remoto, attraverso lp o lpr sulle macchine *Unix*, attraverso il sistema di stampa SMB sulle macchine *Windows* oppure mediante i servizi di stampa *AppleTalk* sulle macchine *Macintosh*. La stampa da remoto consente ad un *computer* di stampare su una stampante che è fisicamente connessa ad un altro *computer*, oppure direttamente alla rete. Tale prestazione è molto vantaggiosa in una rete locale, ma tutte le opzioni di stampa da remoto sono non sicure e risultano inefficienti quando vengono utilizzate in reti geografiche. Se si ha la necessità di stampare su un sito attraverso Internet o consentire ad un altro sito di utilizzare la propria stampante, è possibile impostare uno speciale alias di posta elettronica che stampi il messaggio di posta elettronica non appena riceva il messaggio.

Sicurezza degli accessi remoti

In molti casi, anche per ragioni di performance è necessario che l'elaborazione dei dati avvenga in modalità *server-side*.

In origine, i programmi che fornivano un accesso remoto a *server/mainframe* consentivano agli utenti di utilizzare un sistema remoto come se il proprio *computer* fosse un terminale locale del sistema. Attualmente troviamo sistemi di elaborazione che supportano l'accesso remoto senza richiedere funzionalità di terminale ai *client*.

Il protocollo *Telnet* è il protocollo standard per l'accesso remoto in Internet con modalità di emulazione di terminale alfanumerico. Il protocollo *Telnet* in principio è stato considerato come un servizio abbastanza sicuro poichè richiede agli utenti di autenticarsi. Sfortunatamente, *Telnet* spedisce tutte le proprie informazioni in chiaro, il che lo rende estremamente vulnerabile ad attacchi di tipo *sniffing* e *hijacking*. *Telnet* è sicuro solamente se la macchina remota e tutte le reti tra la macchina remota e le altre sono sicure. Ciò significa che *Telnet* non è sicuro attraverso Internet.

Per risolvere i problemi del protocollo *Telnet* ci sono due modi. Il primo prevede di utilizzare in alternativa un programma che faccia uso della cifratura; lo standard Internet comunemente accettato in questo caso è SSH (*Secure SHell*) che fornisce una varietà di servizi di accesso remoto cifrati. Il secondo prevede l'attivazione di una connessione di rete cifrata (cioè una VPN, *Virtual Private Network*) ed eseguire *Telnet* utilizzando tale connessione.

Altri programmi come *rlogin, rsh* sono utilizzati per fornire accessi remoti senza necessità di autenticarsi nuovamente.

Interfacce grafiche remote per sistemi operativi Microsoft

Spesso l'utente preferisce che l'accesso ad una macchina remota avvenga mediante un'interfaccia grafica, piuttosto che a linea di caratteri.

Microsoft fornisce un'interfaccia grafica remota come parte dei *server Windows* 2000 in un *package* chiamato *Terminal Services*. (Esiste anche una versione di *Windows* NT 4.0 chiamata *Terminal Server*). Sia i *Terminal Services* di Windows 2000 che *Terminal Server* di Windows NT 4.0 impiegano un protocollo sviluppato dalla *Microsoft* chiamato RDP (*Remote Desktop Protocol*) per le comunicazioni tra *client* e *server* (impiega connessioni TCP con *port* 3389 sul *server*).

Diversi protocolli proprietari vengono utilizzati per realizzare interfacce grafiche remote per *Windows*, tra questi il più potente e il più utilizzato è il protocollo ICA (*Indipendent Computing Architecture*) sviluppato da *Citrix*. Il protocollo ICA è stato adottato da diversi produttori.

Molti programmi quali *LapLink, RemotelyPossible* e *PcANYWHERE* rendono possibile l'accesso remoto attraverso i protocolli TCP/IP. Anche BO2K (*Back Orifice* 2000) è un programma gratuito che fornisce accesso remoto e può essere considerato uno dei *tool* più potenti che sia mai stato realizzato per l'amministrazione remota, sebbene venga spesso citato nell'ambito di attacchi e intrusioni a sistemi informatici.

I Window System di rete

Molte macchine *Unix* attualmente forniscono dei *Window System* basati sul sistema X11. I *server* X11 sono anche disponibili come applicazioni di terze parti per molti altri sistemi operativi, incluse tutte le versioni di *Windows* e molte versioni di MacOS. I *client* X11 sono abbastanza rari ma anch'essi sono disponibili per *Windows* NT. L'accesso alla rete è un'importante caratteristica del sistema X11.

I *server* X11 sono bersagli appetibili per gli attaccanti. Un attaccante che guadagna l'accesso ad un *server* X11 può causare i seguenti danni:

- Catturare il *dump* dello schermo, in questo modo si riesce a leggere qualunque cosa sia visualizzata sullo schermo dell'utente.
- Leggere i tasti premuti dall'utente.
- Emulare la pressione dei tasti da parte dell'utente.

In passato, il sistema X11 usava principalmente come strumento di autenticazione l'indirizzo da cui provenivano le richieste di connessione. Oggi molti *server* X11 implementano strumenti di autenticazione più sicuri. Ad ogni modo, il sistema X11, come il protocollo *Telnet* è ancora vulnerabile ad attacchi di tipo *hijacking* e *sniffing*.