Protokollbeschreibungen und Portlisten für Firewalls

Ernst Pisch

17. Mai 2003

Inhaltsverzeichnis

Dieses Dokument soll eine Hilfe für die Konfiguration von Firewalls darstellen. Ich hoffe, dass dem Firewall-Administrator die Arbeit damit erleichtert wird. Es werden Probleme aufgezeigt, mit denen bei manchen Protokollen zu rechnen ist und Tipps, welche Protokolle besser gar nicht erst über eine Firewall geleitet werden sollen. Man kann sich nicht erwarten, dass für jede Protokoll fertige, direkt einsetzbare Konfigurationen angeboten werden. Letztendlich wird man sich bei jeder Konfiguration etwas intensiver mit dem jeweiligen Protokoll beschäftigen müssen.

Bemerkung zu den Tabellen: Die Schreibweise lehnt sich an die Syntax von 'iptables' an, welche Linux ab Kernelversion 2.4 unterstützt. So bedeutet z.Bsp. '1024:' in der Spalte Quell- oder Zielport, dass die Portnummer größer oder gleich der Portnummer 1024 sein kann. 'iptables' unterstützt zustandsgesteuerte Paketfilterung. Bei TCP entspricht 'NEW' einem Paket, dessen SYN-Bit gesetzt und ACK-Bit nicht gesetzt ist. 'ESTABLISHED' sind alle weiteren Pakete, deren ACK-Bit gesetzt ist. 'RELATED' sind Pakete, welche von einer bereits bestehenden Verbindung abhängen. Das wären z.Bsp. ftp-Verbindungen für die Datenübertragung. Dazu muss die Firewall in der Lage sein Protokolle zu 'verstehen'. 'iptables' ist in der Lage, auch bei UDP neue Verbindungen von bestehenden Verbindungen zu unterscheiden. Deshalb findet man in der Spalte 'Status' auch bei UDP entsprechende Einträge wo es sinnvoll ist.

1 Auth und identd

Auth dient dazu, einen Benutzer, der eine Verbindung zu einem Service herstellen möchte, zu identifizieren. Aufgrund des unter UNIX zugehörigen Dämonprozesses wird dieses Protokoll auch *identd* genannt. SMTP-, IRC- und FTP-Server verwenden diese Informationen häufig. Da diese Informationen jedoch auch für Angreifer hilfreich sein können, sollte man Auth-Anfragen nicht durch eine Firewall hindurch lassen. Man sollte diese Pakete an der Firewall aber nicht verwerfen, sondern entweder mit Fehler quittieren (ICMP-Paket des Typs 3 'port unreachable') oder die Verbindung zurücksetzen (TCP Reset). Andernfalls kommt es zu starken Verzögerungen beim Verbindungsaufbau.

Auth verwendet das TCP-Port 113. Es gibt spezielle Auth-Proxies, die solche Pakete nicht an den Client weiterleiten, sondern sofort beantworten. Die Antworten können Zufallsdaten sein. Der Einsatz von NAT ist schwierig, da der Verbindungsaufbau in umgekehrter Richtung erfolgt. Wird z.Bsp. eine Verbindung zu einem Mailserver aufgebaut, so schickt dieser eine Auth-Anforderung an den Client zurück. Diese kann aber nicht dem Client zugeordnet werden, sodass das NAT-System die Zieladresse nicht weiß.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	113	NEW, ESTABLISHED	Anfrage an identd Server
	<	TCP	113	1024:	ESTABLISHED	Antwort von identd Server

2 chargen – Character Generator

chargen ist ein Zeichengenerator. Wird eine Verbindung zu TCP- oder UDP-Port 19 hergestellt, erhält man vom chargen-Server einen endlosen Zeichenstrom. *chargen* ist zwar ungefährlich, aber auch unnützlich[1] und sollte durch die Firewall blockiert werden.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		UDP, TCP	1024:	19	NEW, ESTABLISHED	Client Anfrage

[1] Unnützlich aus der Sicht eines normalen Anwenders. Und über eine Firewall sollten nur die Dienste ermöglicht werden, die unbedingt nötig sind.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
	<	UDP, TCP	19	1024:	ESTABLISHED	Server Antwort

3 CORBA – Common Object Request Broker Architecture IIOP – Internet Inter-Orb Protocol

CORBA ermöglicht verteilte Anwendungen und verwendet zur Kommunikation ein Programm namens *Object Request Broker* oder *Orb*. Im Internet wird *IIOP* (*Internet Inter-Orb Protocol*) verwendet. *IIOPS* verwendet SSL oder TLS zur Verschlüsselung. IIOP ist mit Paketfilterung kaum zu kontrollieren. Es werden keine bestimmten Ports verwendet. Die Sicherheit hängt überwiegend von der Anwendung selbst ab. Durch Verwendung von Verschlüsselung kann es sehr sicher sein. Praktische Anwendung findet IIOP und IIOPS beim Zugriff auf Sybase-Datenbanken (siehe Kapitel 60 auf Seite 43).

Man sollte CORBA nicht über Firewalls betreiben. Wenn es nötig ist, müssen spezielle Proxies eingesetzt werden!

IIOP verwendet eingebettete IP-Adressen und Portinformationen. NAT kann nur dann durchgeführt werden, wenn das NAT-System die eingebetteten Informationen anpassen kann.

4 daytime

Wird eine Verbindung zu TCP- oder UDP-Port 13 hergestellt, wird die aktuelle Systemzeit im Klartext ausgegeben. *daytime* ist zwar ungefährlich, aber auch unnützlich[2] und sollte durch die Firewall blockiert werden.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		UDP, TCP	1024:	13	NEW, ESTABLISHED	Client Anfrage
	<	UDP, TCP	13	1024:	ESTABLISHED	Server Antwort

5 DHCP – Dynamic Host Configuration Protocol bootp

bootp wird verwendet, um einem Rechner, der startet, Informationen wie IP-Adressen mitzuteilen. Es wird auch eingesetzt, um zum Beispiel zusammen mit *tftp* das Betriebssystem über Netz zu laden. *bootp* verwendet Broadcast und wird nur über speziell konfigurierte Router darübergelassen. Bei *bootp* ist jeder MAC-Adresse eines Clients eine IP-Adresse und ev. weitere Information zugeordnet. *DHCP* erweitert die Möglichkeiten von *bootp*. Mit DHCP ist man in der Lage, IP-Adressen dynamisch an Clients zu vergeben, deren MAC-Adressen nicht in einer Konfigurationstabelle registriert sind. Per DHCP kann eine Vielzahl an Konfigurationsdaten übermittelt werden, was eine zentrale Administration von größeren Netzen erleichtert. DHCP verwendet, wie auch *bootp* die UDP-Ports 67 und 68. Bei DHCP werden sowohl Broadcast als auch Unicast verwendet.

[2] Unnützlich aus der Sicht eines normalen Anwenders. Und über eine Firewall sollten nur die Dienste ermöglicht werden, die unbedingt nötig sind.

Obwohl es nicht direkt zum DHCP-Protokoll gehört, verwenden viele DHCP-Server auch eine ARP-Anfrage oder ein ICMP-EchoRequest, um festzustellen, ob eine IP-Adresse schon vergeben ist. Das muss bei der Konfiguration einer Firewall berücksichtigt werden.

DHCP sollte nicht über Firewalls geführt werden. Weder Clients noch Servern sollte getraut werden. Wird ein Bastionhost eingesetzt, sollte dieser nicht per DHCP konfiguriert werden, sondern mit fix konfigurierten Parametern eingestellt sein.

Es gibt eine Vielzahl an DHCP-Proxies, welche aber nicht entwickelt wurden, um Sicherheitsbelange zu erfüllen, sondern nur um die Pakete weiterzuleiten.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		UDP	68	67	Broadcast!!	Anfrage Client -> Server
	<	UDP	67	68	Unicast	Positive Antwort des Servers
	<	UDP	67	68	Broadcast!!	Negative Antwort des Servers
>		UDP	68	67	Broadcast!!	Client akzeptiert DHCP-Angebot von Server
	<	UDP	67	68	Unicast	DHCP-Server bestätigt Lease

6 discard

Wird eine Verbindung zu TCP- oder UDP-Port 9 hergestellt, werden alle eingegebenen Zeichen vom Server angenommen und verworfen. *discard* ist zwar ungefährlich, aber auch unnützlich[3] und sollte durch die Firewall blockiert werden.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		UDP, TCP	1024:	9	NEW, ESTABLISHED	Client Anfrage
	<	UDP, TCP	9	1024:	ESTABLISHED	Server Antwort

7 DNS – Domain Name Service

Domain Name Service oder auch *Domain Name System* genannt, hat nichts mit Microsoft Windows-Domänen zu tun! DNS ist ein verteiltes Datenbanksystem, in dem Rechnernamen und IP-Adressen gespeichert sind. Durch Anfrage eines DNS-Servers kann die richtige IP-Adresse für einen bestimmten Rechner (oder umgekehrt) in einem Netz ermittelt werden. Auch für den Versand von Mail ist das Domain Name System wichtig – es gibt nämlich Auskunft darüber, welcher Rechner einer Domain Mails entgegennimmt.

DNS-Server verwenden Port 53 sowohl mit TCP als auch mit UDP. Clients verwenden sowohl über TCP als auch über UDP beliebige Ports oberhalb von 1023. Einige Server verwenden Port 53, andere Ports oberhalb 1023[4]. Normalerweise werden DNS-Lookups per UDP durchgeführt. Klappt das nicht, so wird die Anfrage per TCP wiederholt. Zonetransfers zwischen Servern erfolgen immer per TCP. Befindet sich ein DNS-Server in einer DMZ und ein anderer im lokalen Netz, so kann man den Verkehr auf TCP beschränken, falls

[3] Unnützlich aus der Sicht eines normalen Anwenders. Und über eine Firewall sollten nur die Dienste ermöglicht werden, die unbedingt nötig sind.

[4] Bei manchen DNS-Servern (z.Bsp. bind9) kann das Verhalten beeinflusst werden.

beide Server 'DNS NOTIFY' unterstützen. Stellt ein Client eine Anfrage an den DNS-Server, so kümmert sich dieser üblicherweise selbst um die Auflösung der Namen – man nennt das 'Rekursiven Lookup'. Es könnte aber auch passieren[5], dass der DNS-Server dem Client einen anderen DNS-Server nennt, bei dem er es versuchen soll. In einer Firewallumgebung ist das nicht erwünscht. Damit die Firewall DNS-Pakete nur zu bestimmten DNS-Servern zulassen muss, können DNS-Server auch so konfiguriert werden, dass Anfragen nur über vordefinierte Server erfolgen. DNS-Server arbeiten immer wie Proxies. NAT ist problemlos einsetzbar.

Ist der Zugriff auf einen eigenen DNS-Server aus dem Internet erforderlich (weil man selbst Serverdienste zur Verfügung stellt), so sollte man unbedingt einen sogenannten 'Fake DNS-Server' in der DMZ einrichten, der nur die wirklich benötigten Daten nach außen gibt. Im lokalen Netz wird dann ein eigener DNS-Server eingerichtet, der alle lokalen Daten uneingeschränkt verwaltet. Der DNS-Server auf dem Bastion-Host sollte keine 'HINFO'-Datensätze nach außen sichtbar machen. Es sollten keine Zonetransfers nach außen möglich sein.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		UDP	1024:	53	NEW, ESTABLISHED	Client-Anfrage über UDP an DNS-Server
	<	UDP	53	1024:	RELATED, ESTABLISHED	Serverantwort
>		TCP	1024:	53	NEW, ESTABLISHED	Client-Anfrage über TCP an DNS-Server
	<	TCP	53	1024:	RELATED, ESTABLISHED	Serverantwort
>		UDP	53	53	NEW, ESTABLISHED	Anfrage über UDP von Server an Server
	<	UDP	53	53	RELATED, ESTABLISHED	Antwort des Servers an Server
>		TCP	1024:	53	NEW, ESTABLISHED	Anfrage oder Zonetransfer zwischen 2 Servern
	<	TCP	53	1024:	RELATED, ESTABLISHED	Antwort oder Zonetransfer zwischen Servern

Bei 'iptables' unter Linux habe ich festgestellt, dass Antworten von Servern mit UDP nicht immer der Anfrage zugeordnet werden können (Status 'RELATED') und verworfen werden, wenn nur Pakete mit dem Status 'RELATED' anstatt 'NEW' zugelassen werden. Ich habe das noch nicht genauer untersucht, es scheint aber trotzdem problemlos zu funktionieren, ohne neue Verbindungen von außen mit 'NEW' zuzulassen.

8 echo

Wird eine Verbindung zu TCP- oder UDP-Port 7 hergestellt, werden vom echo-Server alle eingegebenen Zeichen zurückgegeben. Manche Websites benützen das, um den dem Client nächsten Server zu ermitteln, indem die Durchlaufzeiten ermittelt werden. *echo* ist zwar ungefährlich, aber auch unnützlich[6] und sollte durch die Firewall blockiert werden.

[5] Bei 'bind' lässt sich die Verwendung der 'recursion' konfigurieren.

[6] Unnützlich aus der Sicht eines normalen Anwenders. Und über eine Firewall sollten nur die Dienste ermöglicht werden, die unbedingt nötig sind.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		UDP, TCP	1024:	7	NEW, ESTABLISHED	Client Anfrage
	<	UDP, TCP	7	1024:	ESTABLISHED	Server Antwort

9 GSSAPI – Generic Security Services API

GSSAPI stellt eine Schnittstelle für kryptographische Dienste zur Verfügung. Die GSSAPI stellt Anwendungen Schnittstellen für Auf- und Abbau von Verbindungen, Ver- und Entschlüsselung von Meldungen und Signierung und Echtheitsüberprüfung von Meldungen zur Verfügung. Meistens werden für die Protokollebene Kerberos oder RSA-Public-Keys verwendet. Es ist gut einsetzbar für Anwendungen, die nur eine einzelne Verbindung zwischen Client und Server benötigt. Es ist für jede Verbindung eine eigene GSSAPI-Session erforderlich. GSSAPI eignet sich schlecht für UDP, es sollte immer nur mit TCP arbeiten. Die Einsatzmöglichkeit von Firewalls hängt von der jeweiligen Applikation ab.

10 finger

finger dient Systemverwaltern, Informationen über angemeldete User zu erhalten. Es können damit auch normale Benutzern, Benutzer auf anderen Rechnern finden. Da *finger* keinen Kommandokanal besitzt, ist das Protokoll an sich ziemlich sicher. Gefahr könnte in umgekehrter Richtung entstehen durch bösartig konfigurierte finger-Server. Bedenklich ist eher, dass Angreifer wertvolle Informationen für Angriffe erhalten könnten.

Der Einsatz von Proxies und NAT ist möglich. Jedoch ist es nicht unbedingt anzuraten, die Informationen, die *finger* liefert ins Internet durchzulassen.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	79	NEW,ESTABLISHED	Clientanfrage
	<	TCP	79	1024:	ESTABLISHED	Serverantwort

11 FTP – File Transfer Protocol

FTP ist 'das' Protokoll, um im Internet Dateien zu übertragen. Trotz seiner Beliebtheit und Verbreitung ist FTP in Verbindung mit Firewalls nicht einfach zu handhaben. Es wird aber aufgrund der Wichtigkeit von Firewalls in der Regel gut unterstützt. FTP verwendet 2 getrennte TCP-Verbindungen: einen Kommandokanal für die Steuerung durch den Client und einen Datenkanal für die Übertragung der Daten. FTP kenn 2 verschiedene Modi, die in der Praxis auch beide zu finden sind:

Aktiver oder normaler Modus

Beim Verbindungsaufbau werden auf Client-Seite zunächst 2 TCP-Ports oberhalb 1023 reserviert. Nachdem eine Verbindung zum Server auf dessen TCP-Port 21 hergestellt wurde, sendet der Client im 'Normalmodus' über den soeben hergestellten Kommandokanal den FTP-Befehl 'PORT' und teilt dem Server seine Portnummer mit, die er für den Datenkanal geöffnet hat. Daraufhin stellt der Server eine Verbindung von TCP-Port 20 zum genannten Client-Port her. Das heißt, dass im Normalmodus eine neue Verbindung nicht, wie sonst üblich vom Client, sondern vom Server initiiert wird. Und das bereitet ein Problem bei der

Konfiguration einer Firewall, denn man will nach Möglichkeit keine Verbindungen von außen zulassen. Erschwerend kommt hinzu, dass man die Ziel-Portnummer nicht kennt, die für den Datenkanal geöffnet wird. Man müsste alle Ports über 1023 öffnen, damit ein Datenkanal aufgebaut werden kann.

Aufgrund dieser Schwierigkeiten sind die meisten Firewalls in der Lage, den Inhalt einer FTP-Verbindung zu interpretieren, um nur das jeweils benötigte Port zu öffnen.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	21	NEW, ESTABLISHED	FTP Anfrage von Client zum Server
	<	TCP	21	1024:	ESTABLISHED	Serverantwort auf Clientanfrage
	<	TCP	20	1024:	RELATED, ESTABLISHED	Öffnen eines Datenkanals für FTP Anfrage (Aktiv-Modus)
>		TCP	1024:	20	ESTABLISHED	Antworten im Datenkanal vom Client

Damit nicht generell neue Verbindungen von außen erlaubt werden müssen, muss die Firewall zustandgesteuert sein und erkennen, dass der Verbindungsaufbau für den Datenkanal abhängig vom zuvor eingerichteten Kommandokanal und deshalb zulässig ist.

Der aktive Modus stellt eine weitere mögliche Gefahr dar: Ein speziell präparierter FTP-Client kann einen FTP-Server als Portscanner missbrauchen. Er weist den Server mittels PORT-Kommando an, einen Datenkanal zu einem anderen Rechner herzustellen. Je nach Fehlermeldung, die der FTP-Client dann vom Server erhält, lässt sich darauf schließen, ob ein Port des angegriffenen Rechners offen ist oder nicht. Manche FTP-Server (u.a. Microsofts FTP-Server) lassen deshalb nur jene IP-Adresse zu, zu dessen Rechner der Kommandokanal besteht. Diese Art Angriff kann auch bei intelligenten Paketfiltern durchgeführt werden. Manche Paketfilter lassen ein Port längere Zeit geöffnet nachdem ein ftp-Client das Kommando PORT ausgeführt hat.

Passiver Modus

Obwohl alle üblichen Firewalls eingehenden ftp-data Verbindungsaufbau richtig zuordnen können, bereitet dieser Ablauf einem sicherheitsbewussten Administrator Kopfschmerzen welche gelindert werden können. Und zwar kennt das FTP-Protokoll das Kommando *PASV*, weshalb der Passive Modus manchmal auch PASV-Modus genannt wird. Beim passiven Modus sendet der ftp-Client nach dem Aufbau des Kommandokanals das Kommando PASV und fordert den FTP-Server damit auf ein TCP-Port zu öffnen und dessen Nummer zurückzumelden. Sobald der Client die Zielportnummer (>1023) für die Serverseite des Datenkanals kennt, öffnet der Client seinerseits ein TCP-Port (>1023) und stellt die Verbindung zum Server her. Dadurch geht die Initiative für den Verbindungsaufbau sowohl beim Kommandokanal als auch beim Datenkanal vom Client aus und es muss kein Port für neue Verbindungen von außen zugelassen werden.

Glücklicherweise verwenden fast alle Browser (Netscape, Internet Explorer, ...) den passiven Modus. Damit aber auch andere FTP-Clients keine Probleme haben, ist es empfehlenswert, einen FTP-Proxy zu installieren. Dieser kann aus dem lokalen Netz sowohl aktive wie auch passive Verbindungen entgegennehmen, stellt aber bei entsprechender Konfiguration ins äußere Netz nur passive Verbindungen her. Ein weiterer Vorteil eines FTP-Proxy ist, dass alle Verbindungen inklusive Dateinamen und Kommandos protokolliert werden können, was mit einem reinen Paketfilter nicht möglich wäre. FTP-Proxies lassen meistens auch eine Einschränkung der zulässigen Kommandos zu. Voraussetzung für den Aufbau einer passiven Verbindung ist, dass dies auch der FTP-Server unterstützt.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	21	NEW, ESTABLISHED	FTP Anfrage von Client zum Server
	<	TCP	21	1024:	ESTABLISHED	Serverantwort auf Clientanfrage
>		TCP	1024:	1024:	NEW, ESTABLISHED	Öffnen eines Datenkanals für FTP Anfrage (PASV-Modus)
	<	TCP	1024:	1024:	ESTABLISHED	Antworten im Datenkanal vom Client

Beim Einsatz von FTP-Servern ist äußerste Vorsicht geboten! Es muss darauf geachtet werden, dass FTP-Clients keinen Zugriff auf Dateien erhält, die sie nicht sehen sollen. Vor allem bei 'anonymous' FTP-Zugängen gilt dies. Muss Schreibzugriff gewährt werden, so muss sichergestellt werden, dass Clients nur schreiben, aber nicht mehr lesen können. Andernfalls ist Jenen Tür und Tor geöffnet, die diesen Mailserver für die Verbreitung von illegalen Daten missbrauchen wollen.

Im Zusammenhang mit FTP gibt es noch etwas zu sagen: Viele FTP-Server versuchen beim Verbindungsaufbau mittels 'ident' die Identität des Clients zu eruieren. Wird dieses Paket von der Firewall verworfen, so wartet der FTP-Server einige Zeit bis er weitermacht. Das kann unangenehme Zeitverzögerungen verursachen. Es ist nicht nötig, am Client einen ident-Server zu installieren um dieses Problem zu umgehen. Man muss die Firewall lediglich so konfigurieren, dass eingehende ident-Anfragen mit einem ICMP-Pakete des Typs 3 (Port unreachable) beantwortet werden. Man muss allerdings darauf achten, dass man dadurch nicht Portscannern die Arbeit erleichtert, indem auf alle ungeöffneten Ports so geantwortet wird.

12 Gopher

Gopher ist ein kaum mehr verwendetes Protokoll, um Datenbestände zu durchsuchen. Manche Webserver unterstützen noch Gopher. Es ist darauf zu achten, dass dieser Dienst entweder zu deaktivieren ist, wenn man ihn gar nicht benützen will bzw. dass keine ungewollten Daten angeboten werden.

Proxies machen i.A. keine Probleme. NAT ist auch problemlos möglich.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	70	NEW, ESTABLISHED	Anfrage von Client an Server
	<	TCP	70	1024:	ESTABLISHED	Server Antwort

13 H.323

H.323 ist ein Protokoll, das von der ITU (International Telecommunications Union) standardisiert wurde. Es wird für Audio-[7] und Videokonferenzen verwendet. Weil H.323 sowohl ausgehende als auch eingehende Verbindungen aufbaut, ist der Einsatz über eine Firewall nur zusammen mit einem speziellen Proxy sicher.

H.323 verwendet für den Verbindungsaufbau eine TCP-Verbindung auf Port 1720. Eine weitere, dynamisch zugewiesene TCP-Verbindung, wird für die Verbindungsüberwachung benötigt. Die Daten selbst, werden über eine UDP-Verbindung übertragen, die ebenfalls dynamisch zugewiesen wird. Audio- und Videosignale werden über getrennte Kanäle übertragen. Für eine Videokonferenz werden deshalb in eine Richtung 5

[7] Für IP-Telefonie (Voice over IP) wird auch H.323 verwendet.

Verbindungen (Anruf – TCP 1720, Audiosignal – UDP, Audioüberwachung – TCP, Videosignal – UDP und Videoüberwachung – TCP) und in die andere 4 Verbindungen (Audiosignal – UDP, Audioüberwachung – TCP, Videosignal – UDP und Videoüberwachung – TCP) aufgebaut! Würde man den Anweisungen von Microsoft folgen, müsste man alle UDP- und TCP-Verbindungen in beide Richtungen für Ports oberhalb 1023 zulassen. Da wäre nicht mehr viel übrig, was eine Firewall absichern könnte. Da bei H.323 die üblichen Methoden eingehende von ausgehenden UDP-Verbindungen voneinander zu unterscheiden nicht funktionieren, können die meisten Paketfilter keine Sicherheit bieten. Da es nur wenige Paketfiltersysteme gibt, die das H.323 Protokoll verstehen, wird man spezielle H.323-Proxies benötigen. Es gibt auch sogenannte *MCU*'s (Multipoint Control Unit), die Verbindungen von mehreren Teilnehmern zu mehreren Teilnehmern kontrolliert. Man müsste diese in der DMZ stationieren. Allerdings funktioniert das nur, wenn nur eine Seite eine derartige MCU einsetzt – beidseitig klappt das nicht.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	1720	NEW, ESTABLISHED	Anrufer baut Verbindung auf
	<	TCP	1720	1024:	ESTABLISHED	Angerufener antwortet
>		TCP	1024:	1024:	NEW, ESTABLISHED	Verbindungsüberwachung für Daten (nur eine Richtung)
	<	TCP	1024:	1024:	ESTABLISHED	Antworten für obige Verbindungsüberwachung
>		UDP	1024:	1024:	[8]	Datenübertragung (eine Richtung)
	<	UDP	1024:	1024:		Datenübertragung (andere Richtung)

14 HTTP – Hypertext Transfer Protocol

HTTP selbst ist ziemlich sicher und mit der Firewall einfach zu handhaben. Das Problem ist, dass vielfältigste Daten über HTTP transportiert werden können. Es können Viren oder andere schädliche Programme 'downgeladen' werden. Andere Gefahren entstehen im Zusammenhang mit den Erweiterungen der Browserprogramme. Solche Erweiterungen sind z.Bsp. Java, JavaScript und ActiveX. Da die Browser die im WWW angebotenen Dateien nicht alle selbst darstellen kann, werden externe Programme dafür gestartet. Das birgt weitere Gefahren in sich. Z.Bsp. können Postscriptdateien ungewünschte Befehle enthalten, Word-Dokumente können Macros ausführen, speziell aufbereitete Daten können 'buffer overflow' in Programmen verursachen, … HTTP-Clients können auch ungewollt Informationen preisgeben.[9] Die korrekte Konfiguration von Webbrowsern ist extrem wichtig für die Sicherheit!!!

Der Einsatz von Proxies ist nicht nur möglich, sondern sehr sinnvoll. Cache-Proxies können die die Leistung steigern durch Einsparen von Bandbreite. Es ist damit eine Kontrolle und Protokollierung von Zugriffen möglich. Die meisten der üblichen Webbrowser unterstützen die Verwendung von Proxies.

NAT ist problemlos möglich, da keine eingebetteten Adressen verwendet werden.

[8] siehe Anmerkungen im vorigen Absatz

[9] *From*: Header können die Mailadresse des Users enthalten, wenn diese Daten in der Browser-Konfiguration enthalten sind. Proxies können *Via:* Header hinzufügen, woraus die IP-Adresse der Proxies zu erfahren ist.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	80[10]	NEW, ESTABLISHED	HTTP Client-Anfrage an Webserver
	<	TCP	80	1024:	ESTABLISHED	Server Antwort

Neben HTTP gibt es noch *HTTPS* und *Secure HTTP*. HTTPS verschlüsselt die übertragenen Daten und schützt vor dem Ausspähen von übertragenen Daten (Kreditkarten-Nummern etc.). Es verwendet ein eigenes TCP-Port (Port 443).

Secure HTTP ist kaum bekannt. Es dient dazu die Seiten auf einem Webserver zu signieren. Damit können gefälschte Seiten erkannt werden.

Sowohl Secure HTTP als auch HTTPS können mit Proxies verwendet werden, sie können aber aufgrund der Verschlüsselung den Inhalt der Daten beurteilen. Auch NAT ist bei beiden Protokollen möglich.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	443	NEW, ESTABLISHED	HTTPS Client-Anfrage an Webserver
	<	TCP	443	1024:	ESTABLISHED	Server Antwort

15 HTTP Cache-Proxies

Cache-Proxies werden eingesetzt, um die Leistung zu erhöhen und Bandbreite zu sparen. Daten von Webservern müssen nicht immer vom Originalserver über das 'langsame' Internet geholt werden, sondern können von einem Cache-Proxy bezogen werden, der sich im schnellen lokalen Netz befindet. Manchmal ist ein einzelner Cache-Proxy überfordert, sodass man mehrere davon einsetzt. Diese müssen aber untereinander kommunizieren, da die gewünschten Daten sich auf irgend einem der Proxies befinden können.

15.1 ICP – Internet Cache Protocol

ICP ist ein Protokoll für HTTP-Cacheproxies. Damit kommunizieren mehrere Cache-Proxies, die sich die Last aufteilen. Anfragen werden über UDP gestellt, die Daten selbst werden per TCP aus dem Cache weitergeleitet.

ICP ist selbst in der Lage als Proxy zu dienen.[11] Möchte man ICP über Firewallgrenzen hinweg betreiben, so sollte man auf der Firewall selbst einen Cacheserver mit ICP betreiben, der dann als Proxy dient und Cache-Server im lokalen Netz bedient.

NAT ist möglich, obwohl eingebettete IP-Adressen vorkommen. Diese werden jedoch nicht verwendet. Man muss sich aber im Klaren sein, dass dadurch Informationen aus dem lokalen Netz nach außen gelangen.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		UDP	1024:	3130	NEW, ESTABLISHED	ICP Anfrage an Cache
	<	UDP	3130	1024:	ESTABLISHED	ICP Antwort

[10] Port 80 ist das Standardport. Einige Server laufen auf anderen Ports.

[11] Anfragen werden wie auch bei SMTP und NNTP von Server zu Server weitergereicht.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	3128	NEW, ESTABLISHED	HTTP Anfrage an Cache
	<	TCP	3128	1024:	ESTABLISHED	HTTP Antwort vom Cache

15.2 CARP – Cache Array Routing Protocol

CARP ist ein anderes Protokoll, welches von Cache-Servern verwendet wird. Das Proxy-Verhalten ist gleich wie bei ICP. NAT ist wegen eingebetteter IP-Adressen nicht einfach möglich.

15.3 WCCP – Web Cache Coordination Protocol

WCCP ist ein proprietäres Protokoll von CISCO. Es verwendet UDP-Zielport 2048. Das Quellport ist nicht definiert, scheint aber auch Port 2048 zu sein. Der HTTP-Verkehr wird über GRE gekapselte Pakete weitergeleitet. GRE kennt weder Ports noch ACK-Flags, weshalb die entsprechenden Spalten in der Tabelle leer bleiben.

Der Einsatz von Proxies ist wegen GRE kaum zu realisieren.

WCCP benützt eingebettete IP-Adressen, weshalb NAT nicht funktioniert. Das Konzept geht davon aus, dass zwischen Router und Cacheserver durch ihre Nähe niemals NAT nötig wäre.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>	<	UDP	x[12]	2048		WCCP-Update
>	<	GRE				umgeleitete HTTP-Anfrage an Cache-Server

16 ICA – Independent Computing Architecture

ICA ist ein Protokoll der Firma Citrix, welches für den Zugriff auf Windows-Server entwickelt wurde. Standardmäßig verwendet ICA bei der Benutzerauthentifizierung keine wirklich sichere Methode. Es gibt Programme, die die Passwörter lesen können! Bei *Secure ICA* wird eine RC5-Verschlüsselung verwendet. Der Datenfluss vom Server zum Client, auf dem das Bild dargestellt wird, enthält nur die Veränderungen der Bilddaten.[13]

ICA verwendet 2 Protokolle. Für das Suchen von ICA-Servern im Netz werden Broadcasts auf UDP-Port 1604 eingesetzt. Die Verbindung zwischen ICA-Client und ICA-Server wird über TCP/IP auf Port 1494 hergestellt. Die Suche nach ICA-Servern ist nicht zwingend notwendig und kann über die Firewall verhindert werden.

Es gibt Anleitungen für den Einsatz von ICA über generische Proxies. NAT kann in Verbindung mit ICA eingesetzt werden, allerdings ist eine Anpassung nötig, damit die Suche nach ICA-Servern funktioniert.

[12] scheint 2048 zu sein, ist aber im WCCP-Protokoll nicht definiert

[13] ICA reagiert laut meiner eigenen Erfahrung kritisch auf Übertragungsprobleme.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	1494	NEW, ESTABLISHED	ICA-Verbindung Client -> Server
	<	TCP	1494	1024:	ESTABLISHED	ICA-Verbindung Server -> Client
>		UDP	1024:	1604	NEW, ESTABLISHED	ICA-Server Suche
	<	UDP	1604	1024:	ESTABLISHED[14]	ICA-Suchantwort von ICA-Server

17 ICQ – (I seek you)

ICQ ist ein von Mirabilis speziell für Konferenzen entwickeltes Protokoll. Abgesehen von den Problemen, die mit IRC durch Vertrauensmissbrauch entstehen, sind bei ICQ tatsächliche Sicherheitsprobleme aufgetreten.

ICQ verwendet UDP-Port 4000 beim Verbindungsaufbau zum Server. Die Daten werden über eine TCP-Verbindung geleitet. Am Client können Bereiche für die zu verwendenden TCP-Ports konfiguriert werden. Es ist eine direkte Kommunikation zwischen zwei ICQ-Clients möglich. Beim Einsatz von Proxies sollte man den ICQ-Client entsprechend konfigurieren, damit Daten immer über den Server geleitet werden. Andernfalls treten Probleme auf. Für die direkte Client-Client Kommunikation werden eingebettete IP-Adressen verwendet, was den Einsatz von NAT erschwert.

ICQ sollte niemals über eine Firewall geleitet werden.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		UDP	1024:	4000	NEW, ESTABLISHED	Verbindungsaufbau Client -> Server
	<	UDP	4000	1024:	ESTABLISHED	Serverantwort
>		TCP	1024:[15]	1024:	NEW, ESTABLISHED	Nachrichtenübertragung Client -> Server/Client
	<	TCP	1024:	1024:	ESTABLISHED	Nachrichtenübertragung Server/Client -> Client

18 identd

Siehe Kapitel ’Auth’ auf Seite 5.

19 ICMP – Internet Control Message Protocol

ICMP setzt weder auf UDP noch auf TCP auf, sondern ist ein eigenständiges Protokoll. Anwendung findet es vor allem bei der Diagnose von Netzwerkproblemen und häufig werden ICMP-Pakete übertragen, ohne dass sich der Anwender dessen bewusst ist.

[14] Ob das mit *iptables* auch bei Broadcast funktioniert??
[15] Am Client kann ein Portbereich definiert werden.

19.1 ping

Das bekannteste Werkzeug, das ICMP verwendet ist *ping*. ping schickt ein ICMP-Paket des Typs 'echo request' zu einem Zielhost, der dann mit 'echo reply' antwortet. *ping* selbst ist an sich recht sicher. Allerdings lässt es sich missbrauchen, um DoS-Angriffe durchzuführen indem es das Netz überflutet. Manche Betriebssysteme (z.Bsp. Linux) können so konfiguriert werden, dass innerhalb eines bestimmten Zeitraumes nur eine gewisse Anzahl von 'echo request'-Paketen beantwortet werden. Wenn es die Firewall zulässt, sollte man die Paketgröße der eingehenden 'echo-request'-Pakete limitieren. Mit *ping* kann ein Angreifer auch feststellen, welche Maschinen sich in einem Netz befinden. Verstümmelte ICMP-Pakete können schlecht implementierte IP-Stacks zum Absturz bringen und damit das ganze Betriebssystem blockieren. Da ICMP-Pakete häufig nach außen zugelassen werden und der Datenteil für das Protokoll selbst nicht relevant ist, wird es von Angreifern manchmal dazu verwendet, Informationen durch die Firewall nach außen zu schmuggeln.[16] Falls die Firewall ein zustandsgesteuertes Paketfiltersystem besitzt, sollte man nur Antworten erlauben, die zu einer vorherigen Anfrage passen.

Es exisitieren Proxy-Systeme für *ping*.[17] Setzt man solche Proxies ein, sollte man den Datenteil der *ping*-Pakete bereinigen, um eventuelles Datenschmuggeln zu verhindern. Der Einsatz von NAT ist unproblematisch.

Richtung		Proto-	Meldungs-	Anmerkung
Client	Server	koll	typ	
>		ICMP	8	echo request
	<	ICMP	0	echo reply

19.2 traceroute

traceroute (bei Windows-Systemen *tracert*) ermöglicht die Verfolgung des Weges zu einem Zielrechner. Dazu wird mit einem Trick gearbeitet. *traceroute* versendet ein UDP-Paket an den Zielhost und setzt dabei die TTL zunächst auf 1 und erhöht diesen Wert schrittweise. Jeder Router vermindert die TTL um 1. Erreicht die TTL den Wert Null, wird das Paket verworfen und der Router schickt ein ICMP-Paket des Typs 11 ('time to live exceeded'). Auf diese Weise erhält man die Adresse des Routers, der das UDP-Paket verworfen hat. Durch Erhöhen der TTL kommt das UDP-Paket schrittweise jeweils um einen 'Hop' weiter zum nächsten Router und nähert sich so dem Ziel.

Die UDP-Pakete werden von *traceroute* üblicherweise an Ports im Bereich von 33434 bis 33523 geschickt. Das hängt aber von Implementierung und den Angaben in der Kommandozeile ab.

Richtung		Proto-	Meldungs-	Port		Anmerkung
Client	Server	koll	typ[18]	Quelle	Ziel	
>		UDP		[19]	[20]	UDP traceroute-Test
>		ICMP	8			ICMP traceroute-Test (echo request)
	<	ICMP	0			ICMP 'echo response'
	<	ICMP	11			ICMP 'time to live exceeded'
	<	ICMP	3			ICMP 'destination unreachable'

[16] Zum Beispiel lässt sich das Fernsteuerungswerkzeug *BO2K* so konfigurieren, dass es ICMP verwendet.
[17] SOCKS5 stellt z.Bsp. einen angepasstes ping-Programm zur Verfügung.
[19] Meist oberhalb von 32768.
[20] Meist zwischen 33434 und 33523.

19.3 Vorschläge für Filterung von ICMP-Paketen

Die meisten ICMP-Meldungen werden durch Fehlersituationen erzeugt und dienen dazu, die Datenübertragung effizienter zu gestalten. Andererseits können von Angreifern bewusst falsch eingesetzte ICMP-Meldungen die Datenübertragung behindern.

Die Typen 0 (*echo reply)* und 8 (*echo request*) wurden weiter oben im Zusammenhang mit *ping* und *traceroute* bereits besprochen. Man sollte diese Typen nur eingeschränkt zulassen, um ein Ausspionieren der im lokalen Netz vorhandenen Rechner zu verhindern.

Typ 3 (*destination unreachable*) besitzt zur genaueren Unterscheidung zusätzlich verwendete Codes. Pakete mit den Codes 0 (*network unreachable*), 3 (*port unreachable*), 4 (*fragmentation needed*), 11 (*nework unreachable for TOS*) und 12 (*host unreachable for TOS*) sollten im Allgemeinen – aus dem externen Netz kommend – zugelassen werden. Mit ICMP-Meldungen nach außen sollte man hingegen restriktiv sein, damit das Herausfinden vorhandener Rechner und offener Ports erschwert wird.

Der Meldungstyp 4 (*source quench*) veranlasst einen Rechner dazu, die Übertragungsgeschwindigkeit zu verringern, wenn die zu hohe Datenrate unnötige Sendewiederholungen verursacht. Böswillig eingesetzt, drosselt es die Sendegeschwindigkeit unnötigerweise. *Source Quench* sollte im Allgemeinen aber doch in beide Richtungen zugelassen werden, da die Vorteile vermutlich überwiegen.

Typ 5 (*redirect*) ist eine Aufforderung an den Absender, eine andere Route zu wählen. Solche Pakete sollten ignoriert werden. Angreifer könnten eine Umlenkung des Datenweges für sich ausnützen.

Typ 9 (*router announcement*) und Typ 10 (*router selection*) werden beide von *Router Discovery* verwendet und sollten blockiert werden. Näheres dazu im Kapitel 24 auf Seite 20.

Der Meldungstyp 11 (*time to live exceeded*) wurde weiter oben schon besprochen und sollte in allen Richtungen erlaubt werden.

Typ 12 (*parameter problem*) wird gesendet, wenn irgendwelche Fehler im Zusammenhang mit dem IP-Header auftreten. ICMP-Pakete dieses Typs sollten in allen Richtungen erlaubt werden.

Die Typen 13 (*timestamp request*), 14 (*timestamp reply*), 15 (*information request*), 16 (*information reply*), 17 (*address mask request*) und 18 (*address mask reply*) sollten alle geblockt werden. Sie haben kaum praktischen Nutzen, ermöglichen es aber Angreifern Informationen über das Netz zu sammeln.

20 IGMP – Internet Group Management Protocol

IGMP dient der Verwaltung von Multicast-Gruppen. Multicasting wird z.Bsp. eingesetzt für Konferenzdienste oder für administrative Zwecke (Softwareverteilung,...). Damit ein Router nur dorthin Multicast-Pakete schickt, wo sich auch Teilnehmer der entsprechenden Gruppe befinden, benötigt er Informationen, welche per IGMP übermittelt werden.

IGMP setzt direkt auf IP auf und verwendet die Protokollnummer 2. Es verwendet Pakettypen anstatt Portnummern. IGMP ist ein Datagramm-Protokoll. Alle IGMP-Pakete tragen den TTL-Wert von 1, sodass die Pakete nicht weitergereicht werden. Sitzt die Firewall zwischen Multicast-Router und Gruppen-Teilnehmern, so muss die Firewall so konfiguriert werden, dass die TTL nicht verringert wird.[21] Günstiger wäre es allerdings, die Firewall selbst als Multicast-Router zu konfigurieren, da sonst der gesamte IGMP-Verkehr durchgelassen werden muss.

Zieladresse	Protokoll	Pakettyp	Anmerkung
224.0.0.1	2	0x11	Anfrage nach Host-Mitgliedschaft
Multicast	2	0x12	Host Membership-Report Version 1

[21] *iptables* unter Linux ermöglicht dies mit Hilfe des Match-Operators '-m ttl ...'.

Zieladresse	Protokoll	Pakettyp	Anmerkung
Multicast	2	0x16	Host Membership-Report Version 2
224.0.0.1	2	0x17	Gruppe verlassen

21 IMAP – Internet Message Access Protocol

IMAP wird ebenso wie POP für den Zugriff auf Mailboxen verwendet. Es ist ein neueres Protokoll und ermöglicht mehr Flexibilität beim Umgang mit Mailboxen. IMAP ermöglicht den Einsatz von nicht wiederverwendbaren Kennwörtern (falls es der Server unterstützt). Und auch bei IMAP ist Verschlüsselung mit SSL oder TLS möglich.

IMAP kann auch mit generischen Proxies eingesetzt werden. NAT ist problemlos einsetzbar.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	143	NEW, ESTABLISHED	IMAP Verbindung Client -> Server
	<	TCP	143	1024:	ESTABLISHED	IMAP Serverantwort
>		TCP	1024:	993[22]	NEW, ESTABLISHED	IMAP über SSL Verbindung Client -> Server
	<	TCP	993[23]	1024:	ESTABLISHED	IMAP über SSL Serverantwort

22 IPsec – IP Security Protocol

IPsec wird für die Ende-zu-Ende-Verschlüsselung bei IPv6 und auch IPv4 verwendet. Es wird auf der IP-Schicht implementiert und kann deshalb für jedes IP-Protokoll eingesetzt werden. IPsec legt sich nicht auf die Verschlüsselungsalgorithmen fest (dadurch können zukünftig möglicherweise gefundene Schwächen behoben werden). IPsec besteht selbst wiederum aus 3 Protokollen:

AH Authentication Header – gewährleistet: Sicherheit, dass vom angegebenen Absender stammen und nicht verändert wurden; optional auch Sicherheit, dass erneut gesendete (von illegalem Mitlesen stammende) Pakete nicht akzeptiert werden;
AH setzt direkt auf IP auf und ist dem IP-Protokoll 51 zugeordnet (weder TCP noch UDP!)

ESP Encapsulating Security Payload – kümmert sich um die Verschlüsselung der Daten selbst, sodass abgefangene Daten nicht gelesen werden können;
EH stellt ebenso ein eigenständiges Protokoll auf der IP-Schicht dar und ist der IP-Protokollnummer 50 zugeordnet

ISAKMP Internet Security Association Key Management Protocol – regelt den Schlüsselaustausch zwischen Sender und Empfänger
ISAKMP verwendet das UDP-Protokoll und dessen Portnummer 500 sowohl auf Client- als auch auf Server-Seite

[22] ältere Versionen verwenden Port 585
[23] oder Port 585

Da weder AH noch ESP die Protokolle TCP oder UDP verwenden und auch keine Ports kennen, sind die entsprechenden Spalten in der folgenden Tabelle leer. Eine Firewall, die AH oder ESP kontrollieren will, muss die Angabe von IP-Protokolltypen ermöglichen. Mit 'iptables' unter Linux (ab Kernel V2.4) erfolgt das durch die Option '-p', wie im folgenden Beispiel gezeigt:

```
# iptables -A XXX -p 50 -j ACCEPT        lässt ESP-Protokoll durch
# iptables -A XXX -p 51 -j ACCEPT        lässt AH-Protokoll durch
```

Einsatzfähigkeit von Proxies: Der Einsatz von Proxies ist nur insofern möglich, wenn eine IPsec-Verbindung vom Client zum Proxy und eine weitere vom Proxy zum Server aufgebaut wird (das heißt, es müssen praktisch 2 Ende-zu-Ende-Kommunikationen aufgebaut werden). Theoretisch wäre es möglich, dass sich der Proxy bei der Aushandlung der Security Assosiaction-Parameter beteiligt.

NAT: AH und ESP sorgen für Integritätsschutz des kompletten Paketes einschließlich Header. Es ist somit kein NAT möglich! Dennoch kann man NAT und IPsec-Tunnelung zusammen verwenden, wenn zuerst NAT durchgeführt wird und dann der IPsec-Tunnel eingerichtet wird.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		AH (51)				AH – Client an Server
	<	AH (51)				AH – Server an Client
>		ESP (50)				ESP – Client an Server
	<	ESP (50)				ESP – Server an Client
>		UDP	500	500	NEW, ESTABLISHED	ISAKMP Anfrage
	<	UDP	500	500	ESTABLISHED	ISAKMP Antwort

23 IRC – Internet Relay Chat

IRC wird für das beliebte 'chatten' im Internet verwendet. Es ermöglicht die Kommunikation auf Textbasis in Echtzeit und kann teure Konferenzsysteme zumindest teilweise ersetzen. Während der Betrieb eines IRC-Servers relativ sicher ist, steckt die Gefahr bei diesem Protokoll im IRC-Client. Wobei die Hauptgefahr daher rührt, dass sich gutgläubige 'Chat-Teilnehmer' von anderen 'Profis' gefährliche Dinge einreden lassen, die entweder dem Angreife alle Tore öffnen oder das System zerstören.

Der übliche Weg der Kommunikation führt immer über IRC-Server, wobei am Server TCP-Port 6667 und am Client ein beliebiges Port >1023 verwendet wird. Viele IRC-Clients unterstützen sogenannte *Direct Client Connections* (*DCC*). Damit können 2 IRC-Clients eine direkte Verbindung ohne Server aufbauen. Bei diesen Verbindungen werden beidseitig TCP-Ports oberhalb 1024 verwendet. Ein Client teilt dem andern eine Portnummer mit unter der er erreichbar ist. Daraufhin wird vom Partner eine Verbindung zu diesem Port aufgebaut.

Die meisten IRC-Server versuchen vom Client per 'ident' dessen Identität einzuholen. Wird das von den Clients nicht unterstützt, verweigern einige IRC-Server einen Verbindungsaufbau.

In den Anfängen waren alle IRC-Server in einer Baumstruktur miteinander verbunden und jeder Server verhielt sich zugleich wie ein Proxy. Inzwischen arbeiten die meisten Server unabhängig voneinander, weshalb echte Proxies eingesetzt werden müssen. *mIRC* ist ein SOCKS-fähiger IRC-Client. Ein häufig eingesetzter IRC-Proxy ist *tircproxy*.

NAT zwischen Client und Server macht keine Probleme. Schwierig ist hingegen eine DCC-Verbindung, da das NAT-System in diesem Fall das IRC-Protokoll verstehen muss, um IP-Adresse und Portnummern korrekt zuordnen zu können.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	6667[24]	NEW, ESTABLISHED	IRC-Verbindung Client -> Server
	<	TCP	6667	1024:	ESTABLISHED	IRC Serverantwort
>	[25]	TCP	1024:	1024:	NEW, ESTABLISHED	Client fordert DCC-Verbindung bei anderem IRC-Client an
	<	TCP	1024:	1024:	ESTABLISHED	DCC-Verbindung

24 IRDP – Router Discovery Protocol

IRDP ist eigentlich kein eigenes Router-Protokoll, sondern lediglich ein bestimmter Meldungstyp (Typ 9 und 10) des ICMP-Protokolles.[26] Es ermöglicht Rechnern in einem Netzwerk auf simple Weise zu erfahren, welche Router im Netz existieren. Rechner können gezielt solche Anfragen stellen (*Router Solicitation*) und Router können von sich aus zyklisch sogenannte *Router Announcements* aussenden, um sich bekannt zu machen. Router Discovery verwendet Multicast, kann aber auch Broadcast verwenden. Das sollte aber vermieden werden.

Router Discovery sollte von der Firewall auf jeden Fall geblockt werden.

Zieladresse	Protokoll	Pakettyp	Anmerkung
224.0.0.2 / Broadcast	ICMP	10	Router Solicitation
224.0.0.1 / Broadcast	ICMP	9	Router Announcement

25 Kerberos

Kerberos ist ein Dienst zur Authentifizierung in einer Netzwerkumgebung. Es gibt zwei offizielle Versionen von Kerberos. Die ältere Version 4 sollte nicht mehr eingesetzt werden. Aktuell ist Version 5. Ab Windows 2000 wird Kerberos viel verwendet. Da aber Microsoft-spezifische Erweiterungen verwendet werden, können Kerberos-Server, die nicht auf Windows 2000 laufen, nicht alle erforderlichen Funktionalitäten erfüllen, die ein Windows 2000 Client benötigt. Diese Erweiterung bei Windows 2000 hat auch zur Folge, dass viel mehr TCP eingesetzt wird als UDP.[27] Kerberos Authentifizierungs-Server werden auch *Key Distribution Center* oder *KDC* genannt.

Kerberos Version 5 verwendet sowohl über TCP als auch über UDP Port 88. Kerberos-Clients verwenden beliebige Ports oberhalb 1023. Da Kerberos Authentikator-Pakete die IP-Adresse des Absenders enthalten und diese mit der Quelladresse verglichen wird (bzw. soll – nicht alle Kerberos-Implementierungen tun das), ist der Einsatz eines Proxy bzw. NAT kaum möglich.

[24] Manche Server verwenden andere Ports.
[25] Anfrage geht an andern IRC-Client
[26] Siehe Kapitel 19 auf Seite 15.
[27] Die maximale Paketlänge bei UDP wird durch die MTU beschränkt. In einem Ethernet-LAN beträgt die MTU normalerweise 1500 Bytes. Ist die zu übertragende Datenmenge zu groß, um in ein einzelnes Paket zu passen, muss TCP verwendet werden.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		UDP	1024:	88	NEW, ESTABLISHED	Anfrage an KDC
	<	UDP	88	1024:	ESTABLISHED	KDC Antwort
>		TCP	1024:	88	NEW, ESTABLISHED	Anfrage an KDC mit Überlänge
	<	TCP	88	1024:	ESTABLISHED	KDC Antwort mit Überlänge

26 L2TP – Layer 2 Transport Protocol

L2TP ist wie PPTP eine Erweiterung des Point-to-Point Protokolls (PPP) und wird normalerweise für VPN's verwendet. L2TP verschlüsselt die Daten selbst nicht, weshalb es meist zusammen mit IPsec verwendet wird. Gegenüber PPTP verbirgt es jedoch die beim Verbindungsaufbau laufenden Verhandlungen. L2TP ist im Gegensatz zu PPTP nicht auf IP angewiesen, es kann auch über andere Protokolle laufen.

Wenn L2TP über IP betrieben wird, verwendet es UDP-Port 1701. Da es meist mit IPsec zusammen eingesetzt wird, kommt ESP-Kapselung zum Einsatz. Es gelten die im Kapitel IPsec beschriebenen Paketfiltereigenschaften.

Der Einsatz eines Proxy erhöht die Sicherheit gegenüber einem Paketfilter nicht wirklich. Auf jeden Fall ist L2TP nur zusammen mit einer Verschlüsselung sicher.

NAT sollte problemlos möglich sein sofern die Antwort ebenfalls vom UDP-Port 1701 abgeschickt wird. Andernfalls hat die Firewall keine Chance, die Antwort richtig zuzuordnen.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		UDP	1024:	1701	NEW, ESTABLISHED	Client an Server
	<	UDP	1701[28]	1024:	ESTABLISHED	Server-Antwort

27 LDAP – Lightweight Directory Access Protocol

LDAP wird für den Zugriff auf Verzeichnisdienste verwendet. Anwendungsbeispiele dafür sind Benutzerauthentifizierung, Adressbücher, öffentliche Schlüssel etc. LDAP ist ein einfacher TCP-Dienst, der gut von Paketfiltern kontrolliert werden kann. Neben LDAP, das TCP-Port 389 verwendet, existiert noch eine verschlüsselte Form mit dem Namen LDAPS. LDAPS verwendet TCP-Port 636. LDAPS verwendet TLS zur Verschlüsselung und Authentifizierung. Clients verwenden ein beliebiges Port oberhalb 1023.

Ab Windows 2000 verwendet Active-Directory auch LDAP, jedoch verwendet es TCP-Port 3268 bzw. TCP-Port 3269 für den SSL-gesicherten Dienst.

Beim Einsatz von LDAP-Proxies ist darauf zu achten, dass viele Produkte mit dieser Bezeichnung nicht Proxies im herkömmlichen Sinn sind, sondern eigentlich Gateways für die Umwandlung in andere Verzeichnisdienste darstellen. LDAP ist aber ein einfaches Protokoll, das leicht zusammen mit Proxies (z.Bsp. SOCKS) eingesetzt werden kann. NAT kann verwendet werden. Man muss berücksichtigen, dass LDAP-Server einen Client anweisen können, die Daten von einem anderen LDAP-Server zu holen. Solche Anwei-

[28] Der Standard erzwingt keine Antwort vom Port 1701, die meisten Server tun dies jedoch, was Firewalladministratoren sicherlich freut.

sungen können IP-Adressen enthalten. Bei NAT darf eine solche Umlenkung nicht auf einen unerreichbaren Server verweisen.

Beim Einsatz von LDAP für Zugriff aus dem Internet sollte man entweder einen eigenen LDAP-Server einsetzen, oder Zugriffe durch einen Proxy kontrollieren.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	389	NEW, ESTABLISHED	LDAP-Anfrage, Client -> Server
	<	TCP	389	1024:	ESTABLISHED	Antwort des LDAP-Servers
>		TCP	1024:	636	NEW, ESTABLISHED	LDAPS-Anfrage, Client -> Server
	<	TCP	636	1024:	ESTABLISHED	Antwort des LDAPS-Servers
>		TCP	1024:	3268	NEW, ESTABLISHED	LDAP-Anfrage für Global Catalog des Active Directory
	<	TCP	3268	1024:	ESTABLISHED	Antwort des Active Directory Servers
>		TCP	1024:	3269	NEW, ESTABLISHED	LDAPS-Angrage für Global Catalog des Active Directory
	<	TCP	3269	1024:	ESTABLISHED	Antwort des Active Directory Servers

28 lpr – Line Printer System

Um Daten auf einem Drucker im Netzwerk auszudrucken, gibt es verschiedene Verfahren. Manche Drucker wie z.Bsp. HP-Drucker mit JetDirect-Box verwenden TCP-Port 9100 um Druckdaten entgegenzunehmen. Die Daten werden ohne speziellem Protokoll – einfach über TCP – zum Drucker übertragen. Jede TCP-Verbindung, beginnend mit SYN und endend mit FIN, stellt einen Druckjob dar. Andere Drucker verfahren ähnlich, 'hören' jedoch auf andere Portnummern.

Ein anderes Verfahren, welches von den meisten Druckspoolsystemen unterstützt wird, ist *lpr*. *lpr* verwendet TCP-Port 515. Es unterstützt weder Benutzerauthentifizierung noch die Verschlüsselung von Druckdaten. Berechtigungen können nur für einen gesamten Rechner verliehen werden. *LPRng* (lpr next generation) ist eine neuere Entwicklung, die zu *lpr* kompatibel ist, aber um Verschlüsselung und Authentifizierung erweitert ist. *LPRng* verwendet dasselbe TCP-Port 515.

Im Gegensatz zu den meisten anderen Protokollen, verwendet der Client ein zufälliges Port unterhalb 1024. Manche Implementationen verwenden nur Ports zwischen 721 und 731. Da *lpr* Aufträge von einem Server zum andern weiterreicht, arbeitet dieses Drucksystem selbst wie ein Proxy (ähnlich wie SMTP oder NNTP). NAT kann eingesetzt werden, der Dateninhalt kann aber Informationen über Hostnamen des lokalen Netzes verraten.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	:1023	515	NEW, ESTABLISHED	lpr Druckauftrag an Server
	<	TCP	515	:1023	ESTABLISHED	Serverantwort

29 Microsoft SQL Server

Microsoft SQL-Server unterstützt neben TDS (siehe Kapitel 65 auf Seite 46) Microsoft RPC und SMB für den Zugriff der Clients auf die Datenbank. Die Authentifizierungsart kann konfiguriert werden. Bei *SQL-Authentifizierung* werden die Daten unverschlüsselt gesendet, besser ist die Windows NT-Authentifizierung. Wenn es sich nicht vermeiden lässt, Datenbankzugriffe über Firewalls zuzulassen, sollte man TDS mit Windows NT-Authentifizierung einsetzen. Es ist zu berücksichtigen, dass nicht alle Versionen von Microsoft SQL-Server die Datenbank-Replikation über TDS unterstüzen[29]. Zusammen mit TDS ist der Einsatz von Proxies und NAT einfach möglich.

30 MySQL

MySQL ist eine weit verbreitete freie Datenbank. Sie wird häufig zusammen mit Webservern eingesetzt. Bezüglich der Sicherheit ist wichtig zu wissen, dass direkt nach der Erstinstallation als Normaluser ohne Superuser-Rechten eine Datenbankanmeldung mit 'root' (das ist der Datenbank-Superuser, nicht UNIX-Superuser) ohne Passwortabfrage möglich ist! Bis zur Vergabe eines Passwortes besteht also die Gefahr eines Angriffes. Solange man sich dessen bewusst ist, sollte das auch kein Problem darstellen. Es ist möglich den Zugang durch MD5 verschlüsselte Passwörter zu schützen. Die Datenübertragung selbst kann mit SSL verschlüsselt werden.

MySQL arbeitet über das Netzwerk standardmäßig mit dem TCP-Port 3306.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	3306	NEW, ESTABLISHED	MySQL-Verbindung Client -> Server
	<	TCP	3306	1024:	ESTABLISHED	Serverantwort

31 NetBT – Netbios über TCP
CIFS – Common Internet File System
SMB – Server Message Block

Netbios über TCP stellt die Dienste 'NetBIOS-Namensdienst', 'Datagrammdienst' und 'Session-Dienst' zur Verfügung. Der Namensdienst arbeitet auf UDP-Port 137. Der Datagrammdienst verwendet UDP-Port 138. Der Sessiondienst läuft über TCP-Port 139. Es existiert prinzipiell die Möglichkeit auch als 'Locator-Dienst' zu fungieren (ähnlich portmapper), was in der Praxis aber praktisch nicht vorkommt. Es werden auch eingebettete IP-Adressen verwendet, diese haben aber nur Informationscharakter und werden nicht für die Verbindung selbst benötigt. Deshalb ist NAT prinzipiell möglich. Der Datagramm-Dienst erfolgt häufig per Broadcast (Probleme beim Routen!!).

NetBT sollte möglichst nicht über eine Firewall geführt werden. Es kann sehr unsicher sein, weil die Übertragung von sicherheitskritischen Daten (je nach Betriebssystem-Version und Konfiguration) unverschlüsselt bzw. mit schwacher Verschlüsselung erfolgen kann. Der Einsatz von Proxies erhöht die Sicherheit kaum, kann aber einige bekannte DoS-Angriffe verhindern.

[29] Normalerweise verwendet Microsoft SQL-Server für die Replikation SMB.

31.1 NetBT-Namensdienst

Microsoft hat im Laufe der unterschiedlichen Windows-Versionen mehrmals die Methoden der Namensauflösung verändert, musste aber immer kompatibel zu den vorherigen Versionen bleiben. Das hat dazu geführt, dass die Namensauflösung bei den Windows-Betriebssystemen immer komplizierter wurde. Es werden Methoden verwendet, die den Namen durch Broadcast versuchen aufzulösen und welche, die das durch Unicast erreichen. Windowsrechner verwenden oft beides zugleich.

NETBIOS-Namen dürfen bis zu 15 Zeichen lang sein und dürfen keinen 'Punkt' im Namen enthalten[30]. Es werden - nicht wie bei DNS - Hierarchien unterstützt. Damit in einem größeren Netz die Probleme gleicher Rechnernamen leichter gelöst werden können, hat Microsoft NETBIOS-Bereiche eingeführt. Bereichsnamen und Rechnername zusammen dürfen maximal 255 Zeichen lang sein. Ursprünglich wurden Namen durch Broadcast-Anfragen aufgelöst. In größeren Netzen führt das zu einer erheblichen Belastung weshalb Microsoft WINS (Windows Internet Name Service) eingeführt hat. Die Namen werden auf einem WINS-Server verwaltet. Clients wenden sich per Unicast an den WINS-Server um Namen aufzulösen. Jedoch müssen trotz Einsatz von WINS auf Rechnern weiterhin Server laufen, die Broadcast-Anfragen behandeln. Ab Windows 2000 wird bevorzugt DNS zur Namensauflösung verwendet. Für Namen, die länger als 15 Zeichen sind, funktioniert die Namensauflösung nur über DNS.

Es kann vorkommen, dass Clients DNS-Antworten bekommen, obwohl keine DNS-Anfrage gestellt wurde. Das kommt daher, dass WINS-Server auch Gateways zum DNS-Dienst sein können. Umgekehrt können DNS-Server auch Gateways zu WINS-Server sein. Der 2. Fall wird deshalb verwendet, weil sich Rechner, die soeben hochgefahren werden, beim WINS-Server anmelden und deshalb dort bekannt sind. Dadurch bleiben die Daten aktuell.

In Windows stehen also folgende Möglichkeiten der Namensauflösung zur Verfügung:

- Der Name befindet sich im lokalen Namecache, sodass keine Anfrage übers Netzwerk erforderlich ist.
- Es wird ein WINS-Server kontaktiert.
- Die lokale Datei *lmhosts* wird durchsucht
- Der Name wird mittels NETBIOS-Broadcast aufgelöst
- Die lokale Datei *hosts* wird durchsucht
- Es wird ein DNS-Lookup durchgeführt

Da sich Rechner je nach Betriebssystem und Konfiguration unterschiedlich verhalten (die Reihenfolge der Auflösungsversuche betreffend), verwendet Microsoft in der Literatur spezielle 'Node-Bezeichnungen':

B-Node (Broadcast) – der Rechner führt nur Broadcast-Abfragen durch

P-Node (Punkt-zu-Punkt) – der Rechner führt nur WINS-Abfragen durch

M-Node (Mixed) – der Rechner versucht zuerst einen Broadcast und dann erst eine WINS-Abfrage

H-Node (Hybrid) – der Rechner versucht zuerst eine WINS-Abfrage und dann einen Broadcast (das ist die übliche Methode bei Rechnern ab Windows-NT, welche WINS verwenden)

Um mehr Ausfallssicherheit zu erreichen, kann man mehrere WINS-Server betreiben. Die Daten werden zwischen den WINS-Servern repliziert. WINS-Server senden zunächst IGMP-Pakete, um mögliche Partner zu finden und um Multicast-Adressen anzumelden. Die Replikation selbst erfolgt über TCP-Port 42. Es

[30] Das 16. Byte eines NETBIOS-Namen enthält ein Typkennzeichen.

werden immer Verbindungen in beiden Richtungen aufgebaut weshalb eine Firewall nicht so konfiguriert werden kann, dass Verbindungen nur in eine Richtung ermöglicht werden.

Für die Administration von fernen WINS-Servern kann ein WINS-Manager verwendet werden. Dieser benützt Microsoft-RPC (siehe Kapitel 54 auf Seite 39).

Der NetBT-Namensdienst verwendet TCP und UDP auf Port 137. TCP wird nur von Servern verwendet, oder von Clients, wenn UDP keine oder eine verstümmelte Antwort liefert. Microsoft verwendet Port 137 sowohl für Anfragen als auch für Antworten. Zur Auflösung von Namenskonflikten müssen Verbindungen in beiden Richtungen erlaubt werden. Wird kein WINS verwendet, so werden Anfragen an die Broadcastadresse gesendet. Antworten erfolgen über Unicast. WINS-Server kommunizieren untereinander über Multicast und verwenden zusätzlich IGMP. Die WINS-Replikation verwendet TCP-Port 42.

Microsoft stellt einen WINS-Proxy-Dienst zur Verfügung, welcher Broadcast-Anfragen in WINS-Anfragen umwandelt. NAT ist in der Praxis nicht verwendbar. Es werden nicht nur eingebettete Adressen verwendet, sondern es muss sich ja auch jeder Rechner registrieren wodurch keine Adressen eingespart werden können.

WINS-Anfragen und WINS-Replikationen sollen nicht über Firewalls geführt werden. Damit keine Probleme mit Bastion-Hosts entstehen, sollten die Namen entweder statisch in den WINS-Servern eingetragen werden, oder deren Namen müssen länger als 16 Zeichen sein, damit auf DNS ausgewichen werden muss.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		UDP	137, 1024:	137	Broadcast!!	Anfrage an NetBT-Namensdienst über UDP, Client -> Server
>		UDP	137, 1024:	137		WINS-Anfrage über UDP, Client -> Server
	<	UDP	137	137, 1024:		Antwort auf UDP-Anfrage, Server -> Client
>		TCP	137, 1024:	137	NEW, ESTABLISHED	Anfrage über TCP, Client -> Server
	<	TCP	137	137, 1024:	ESTABLISHED	Antwort auf TCP-Anfrage, Server -> Client
>		TCP	1024:	42	NEW, ESTABLISHED	Replikationsanfrage eines WINS-Servers
	<	TCP	42	1024:	ESTABLISHED	Antwort auf Replikationsanfrage

31.2 NetBT-Datagrammdienst

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		UDP	1024:	138	NEW,ESTABLISHED	Client-Anfrage an NetBT-Datagramm-Server
	<	UDP	138	1024:	ESTABLISHED	Antwort von NetBT-Datagramm-Server an Client

31.3 NetBT-Sessiondienst

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	139	NEW, ESTABLISHED	Client-Anfrage an NetBT Session-Server
	<	TCP	139	1024:	ESTABLISHED	Antwort von NetBT Session-Server

31.4 CIFS und SMB

SMB (Server Message Block) ist ein Protokoll, welches für Datei- und Druckdienste verwendet wird. Es setzt auf NetBT auf. Neuere Varianten verwenden TCP/IP direkt und werden in diesem Zusammenhang meist CIFS (Common Internet File System[31]) genannt. Neben dem 'Filesharing' wird CIFS auch für entfernte Transaktionen eingesetzt. Anwendungen dafür sind u.a. NT-Lanmanager Authentifizierung, Server-Manager, Registrierungseditor, Ereignisanzeige und Druck-Spooler. Öffnet man die Firewall für Filesharing, so ist sie auch für eine Menge anderer Dinge offen!! Die Sicherheit hängt von jedem einzelnen Rechner im Netz ab!

SMB wird meist über NetBT Session-Dienst (TCP-Port 139) geführt. Der NetBT-Datagrammdienst (UDP 138) wird dafür nur sehr selten verwendet. Ab Windows-2000 kann SMB direkt auf TCP/IP laufen und verwendet hierfür die Ports 445 auf TCP (Sessiondienst) oder UDP (Datagrammdienst – nur sehr selten).

SMB-Clients verwenden häufig den NetBIOS-Namensdienst.

Der Einsatz von Proxies ist möglich, bringt aber sicherheitsmäßig nur dann etwas, wenn er die sehr komplexen Internas beherrscht (verkettete Operationen, Überwachung von Dateinamen, . . .).

NAT ist möglich, da zwar eingebettete IP-Adressen vorkommen, diese aber nicht für die Kommunikation verwendet werden. Ab Windows 2000 läuft SMB direkt über TCP/IP und verwendet keine eingebetteten IP-Adressen.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	139	NEW, ESTABLISHED	Verbindungsaufbau Client -> Server
	<	TCP	139	1024:	ESTABLISHED	Server-Antwort
>		UDP	1024:	139	NEW, ESTABLISHED	Verbindungsaufbau Client -> Server (sehr selten)
	<	UDP	139	1024:	ESTABLISHED	Server-Antwort (sehr selten)
>		TCP	1024:	445	NEW, ESTABLISHED	≥W2k Verbindungsaufbau Client -> Server
	<	TCP	445	1024:	ESTABLISHED	≥W2k Server-Antwort
>		UDP	1024:	445	NEW, ESTABLISHED	≥W2k Verbindungsaufbau Client -> Server (sehr selten)
	<	UDP	445	1024:	ESTABLISHED	≥W2k Serverantwort (sehr selten)

[31] Der Begriff 'Filesystem' verwirrt, weil es sich nicht um ein wirkliches Filesystem, sondern um ein Protokoll handelt.

31.5 Windows-Browser

Der Windows-Browser oder Computer-Suchdienst ermöglicht es einem Benutzer an einem Windowsrechner aus einer Liste von Computern auszuwählen, um anschließend eine Verbindung dorthin herzustellen. Es ist völlig normal, dass in den Listen Rechner vorkommen, die nicht erreichbar sind oder umgekehrt Verbindungen zu Rechnern hergestellt werden können, die gar nicht angezeigt werden. Das kommt daher, dass es immer einige Zeit braucht, bis Browser-Server von Änderungen im Netz erfahren. In der Praxis ist es dann oft so, dass die Änderungen im Netz sich rascher ändern als Zeit benötigt würde um die Daten zu aktualisieren.

Browser-Server erhalten die Informationen von Maschinen, die sich beim Hochfahren bei der Domäne oder Arbeitsgruppe anmelden. Jeder Rechner, der Browserdienste unterstützt, kann 'Master-Browser' im Netz werden. Wer es letztendlich wird, hängt vom Betriebssystemtyp und von der Uptime (die Zeit, die ein System bereits läuft) ab. Alle Rechner, die einen Dienst haben, der im Browser sichtbar sein soll, senden alle 12 Minuten eine Benachrichtigung an die jeweilige Gruppe.

Der Windows-Browser verwendet den NetBT-Namensdienst auf Port 137 (sowohl TCP, als auch UDP), den NetBT-Datagrammdienst auf UDP-Port 138 und den NetBT-Sessiondienst auf TCP-Port 139. Das lässt erahnen, wie komplex der Windows-Browser ist. Da Vieles mittels Broadcast passiert, sind Proxies praktisch nicht einsetzbar. Auch NAT kann nicht verwendet werden. Ein Betrieb über eine Firewall sollte nicht durchgeführt werden.

31.6 Windows-Authentifizierung

Die Art und Weise, wie sich Windows-Rechner untereinander authentifizieren, hängt von Betriebsystemtyp und Konfiguration ab. Generell sollte keine Authentifizierung über eine Firewall geführt werden.

Die folgenden Listen zeigen, was eine Firewall unterstützen muss, um unterschiedliche Authentifizierungen zuzulassen:

Firewall zwischen Domain-Controller und Win95, Win98 etc. Weiterleitung von NetLogon-Broadcast oder funktionstüchtige WINS-Konfiguration, um den Domain-Controller zu finden.

Firewall zwischen Domain-Controller und Domänenmitglied (z.Bsp. Win-NT) Weiterleitung von NetLogon-Broadcast oder funktionstüchtige WINS-Konfiguration.
SMB zum Domain-Controller
Microsoft-RPC zum Domain-Controllern

Firewall zwischen PDC und BDC SMB zwischen Domain-Controllern
Mircrosoft-RPC zwischen Domain-Controllern

Firewall zwischen Domänen mit Trustbeziehung Weiterleitung von NetLogon-Broadcast oder funktionstüchtige WINS-Konfiguration, um den Domain-Controller zu finden.
Mircrosoft-RPC zwischen Domain-Controllern

32 NetMeeting

NetMeeting wird von Microsoft für Konferenzdienste verwendet. Es ermöglicht neben Audio- / Videokonferenzen den Datenaustausch, Chat und Whiteboard. NetMeeting verwendet T.120 und H.323. Zusätzlich wird eine Audio-Verbindungsüberwachung auf TCP-Port 1731, ein LDAP-Suchdienst mit dem Namen *Internet Locator Service* (ILS) auf TCP-Port 389 und ein proprietärer Suchdienst mit dem Namen *User Location Service* (ULS) auf TCP-Port 522. Es gelten daher die Sicherheitsprobleme von T.120, H.323

und jene von NetMeeting selbst. NetMeeting-Clients können zwar gewisse Zugriffe einschränken und Authentifizierung erzwingen, aber es ist für den Administrator schwierig, diese Sicherheitsmechanismen zu erzwingen.

Im Vergleich zu T.120 und H.323 sind die zusätzlichen Protokolle relativ einfach von Proxies zu verarbeiten. Wenn ein Proxy H.323 korrekt behandeln kann, so kann auch NetMeeting damit kontrolliert werden. Für den Einsatz von NAT ist ein H.323-fähiger Proxy nötig.

Der Einsatz über eine Firewall sollte vermieden werden.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	1731	NEW, ESTABLISHED	Audio-Steuerung; Anrufer kontaktiert Angerufenen
	<	TCP	1731	1024:	ESTABLISHED	Audio-Steuerung; Angerufener antwortet
>		TCP	1024:	389	NEW, ESTABLISHED	Verbindung von Client zu ILS-Server
	<	TCP	389	1024:	ESTABLISHED	ILS-Server Antwort
>		TCP	1024:	522	NEW, ESTABLISHED	Verbindung von Client zu ULS-Server
	<	TCP	522	1024:	ESTABLISHED	ULS-Server Antwort

33 NetOp – Remote Konsolsteuerungsprogramm für Windows

NetOp ist eines der vielen verfügbaren Programme mit denen von der Ferne ein Windows-Rechner bedient werden kann. Es wird der Bildinhalt der fernen Konsole quasi auf den lokalen Client kopiert. NetOp kann sowohl über UDP als auch über TCP laufen. TCP-Verbindung funktioniert erst ab Version V6.50 zuverlässig. Die Portnummer kann bei beiden Protokollen konfiguriert werden. Standardport ist 6502. Bei Verwendung von UDP sollte in 'NetOp Remote Control' 'keep-alive' aktiviert werden. Wird TCP verwendet, sollte man in der Datei 'NetOp.INI' in der Sektion überprüfen, dass 'BindToRecvPort=TRUE' gesetzt ist.

NAT ist möglich. Es dürfte die IP-Adresse jedoch eingebettet sein, da nicht die maskierte IP-Adresse, sondern die ursprüngliche Adresse angezeigt wird.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		UDP	1024:	6502	NEW, ESTABLISHED	Daten von Client zum Server
	<	UDP	6502	1024:	ESTABLISHED	Server-Antwort
>		TCP[32]	1024:	6502	NEW, ESTABLISHED	Daten von Client zum Server
	<	TCP	6502	1024:	ESTABLISHED	Server-Antwort

34 NFS – Network File System

NFS ermöglicht den Zugriff auf Dateien eines fernen Rechners. NFS-Server laufen meist auf UNIX-Systemen, Clients gibt es praktisch für alle verbreiteten Betriebssysteme. Das NFS-Protokoll ist zustands-

[32] erst ab Version 6.50

los um größtmögliche Fehlertoleranz zu ermöglichen. Das heißt, dass der NFS-Server nicht weiß, was ein Client vorher getan hat. Um den Protokolloverhead zu minimieren, wird i. A. UDP eingesetzt, NFS Version 3 unterstützt sowohl UDP als auch TCP. NFS macht Gebrauch vom RPC Protokoll, dennoch wird immer Port 2049 verwendet. Da die Authentifizierung bloß auf der IP-Adresse des Clients beruht, gilt NFS als ziemlich unsicher. Es sind nicht nur die Serverdaten gefährdet, sondern auch der Client. Da das Einhängen eines übers Netz freigegebenen Verzeichnisses privilegierte Rechte benötigt, können bösartige Programme, die sich am Server befinden, Schaden auf dem Client anrichten.

Man sollte es möglichst vermeiden, NFS über Firewalls einzusetzen. Ist es dennoch nötig, sollte man wenigstens aktuelle Versionen verwenden, die viele Sicherheitsprobleme älterer Versionen vermeiden. Zusammen mit NFS ist es meist nötig, ein Protokoll zur Synchronisation der Systemuhren einzusetzen (z.Bsp. ntp), weil es sonst zu Fehlverhalten kommen kann.

Wegen der Verwendung von RPC wird Port 111 für den Portmapper benötigt. *mountd* ist ein RPC-Protokoll und verwendet eine zufällige Portnummer. Werden Dateisperren verwendet, weil konkurrierende Schreibzugriffe kontrolliert werden müssen, so muss auch noch *lockd* und *statd* eingesetzt werden. *lockd* und *statd* sind ebenfalls RPC-Protokolle auf zufälligen Portnummern.

Der Einsatz von Proxies bei NFS ist nicht nur wegen der Vielzahl an beteiligten Protokollen sehr schwierig, sondern auch aufgrund der großen Datenmengen zeitkritisch. Auch NAT ist kaum einsetzbar, weil *mountd* die IP-Adressen zur Authentifizierung verwendet. Bei manchen Versionen wird auch noch das NFS-Filehandle zusammen mit der IP-Adresse auf Konsitenz überprüft.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		UDP	1024:	111	NEW, ESTABLISHED	Portmapper Anfrage von NFS-Client
	<	UDP	111	1024:	ESTABLISHED	Antwort von Portmapper
>		UDP	:1023	2049	NEW, ESTABLISHED	NFS-Client Anfrage an NFS-Server
	<	UDP	2049	:1023	ESTABLISHED	NFS-Server Antwort
>		TCP	1024:	111	NEW, ESTABLISHED	Portmapper Anfrage von NFS-Client
	<	TCP	111	1024:	ESTABLISHED	Antwort von Portmapper
>		TCP	:1023	2049	NEW, ESTABLISHED	NFS-Client Anfrage an NFS-Server
	<	TCP	2049	:1023	ESTABLISHED	NFS-Server Antwort

35 NIS, NIS+ – Name Information Service

NIS ist vor allem in der UNIX-Umgebung im Einsatz. Früher trug es den Namen 'Yellow Pages'. Es dient dem Austausch von Administrationsdaten (Benutzerdaten, Rechnernamen, Netzwerkadressen etc.) zwischen Rechnern. Geschützt ist das Protokoll nur durch einen 'NIS-Domänennamen'. Da keine Veschlüsselung der Daten erfolgt, ist der Einsatz nur in einer vertrauenswürdigen Umgebung einsetzbar. Zudem wird RPC verwendet, was auch sicherheitskritisch ist (siehe Kapitel 54 auf Seite 39).

NIS+ vermeidet einige Sicherheitsschwächen von NIS. Z.Bsp. werden die Daten verschlüsselt. Die Sicherheit ist aber nur dann gegeben, wenn alle Rechner NIS+ unterstützen, da NIS+ zu NIS kompatibel ist. NIS+ arbeitet mit Secure-RPC.

NIS sollte nie über eine Firewall betrieben werden.

36 NNTP – Network News Transfer Protocol

Für *NNTP* gelten ähnliche Sicherheitskriterien wie für SMTP und HTTP. Die Gefahr kann von den Daten selbst ausgehen (Viren, illegale Daten, etc). Wie bei SMTP sollten Anwender im lokalen Netz nicht direkt auf einen NNTP-Server im Internet oder auf einen Bastionhost zugreifen können. Man sollte sowohl im lokalen Netz selbst als auch einen Bastionhost einsetzen. NNTP-Server reichen die News von einem Server zum anderen und verhalten sich selbst wie Proxies. NAT ist prinzipiell möglich, da keine eingebetteten IP-Adressen verwendet werden. Es kann aber sein, dass Probleme dadurch auftreten, dass Hostnamen und IP-Adressen zur Authentifizierung verwendet werden.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	119	NEW, ESTABLISHED	Verbindung Client -> Server
	<	TCP	119	1024:	ESTABLISHED	Serverantwort

37 NTP – Network Time Protocol

NTP ermöglicht die Synchronisation von Systemuhren auf Genauigkeiten im Millisekundenbereich. Für viele Anwendungen ist eine genaue Uhrzeit und vor allem synchrone Zeit zwischen vernetzten Rechnern notwendig. Zum Beispiel setzt NFS synchrone Systemzeit aller beteiligten Systeme voraus. NTP kommt aus der UNIX-Welt, ist aber für viele Betriebssysteme (auch für Windows) verfügbar.

Die Verfolgung von Angriffen oder illegaler Verbreitung von Daten (Viren, SPAM-Mails, Kinderpornographie, etc.) ist häufig erst dadurch möglich, wenn Logdateien unterschiedlicher Server im Internet verglichen werden können. Durch eine genaue Systemuhr wird dies wesentlich erleichtert.

Manche Sicherheitsprotokolle verwenden die Systemzeit, um Angriffe durch ein wiederholtes Senden von Datenpaketen zu verhindern. Durch Veränderung der Systemzeit könnte ein solcher Angriff ermöglicht werden. Ältere NTP-Versionen unterstützen keine Authentifizierung. Das wird erst ab Version NTPv3 unterstützt. Bezieht man die Zeit aus dem Internet, so sollte man NTP auf einem Bastionhost einrichten und unbedingt mehrere NTP-Server im Internet wählen. NTP vergleicht ständig die Zeiten aller in der Datei *ntp.conf* konfigurierten Server und akzeptiert keine größeren Abweichungen. Um gegen Angriffe von außen besser geschützt zu sein, ist eine Funkuhr[33] oder GPS-Uhr im lokalen Netz zu bevorzugen. Alle anderen Rechner im Netz beziehen die Zeit dann von diesem Zeitserver und können nicht aus dem Internet beeinflusst werden.

NTP verwendet das UDP-Protokoll auf dem Port 123. NTP-Clients verwenden Quellports oberhalb 1023. NTP-Server verwenden für die Kommunikation untereinander beidseitig Port 123. NTP-Server können auch über Broadcast oder Multicast (Adresse 224.0.1.1 ist dafür reserviert) kommunizieren.

Da NTP über eine Hierarchie von Zeitservern arbeitet, ist kein eigener Proxy nötig. NTP-Server verhalten sich selbst ähnlich wie Proxies. NAT kann problemlos eingesetzt werden.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		UDP	1024:	123	NEW, ESTABLISHED	NTP-Anfrage; Client -> Server
	<	UDP	123	1024:	ESTABLISHED	Antwort von NTP-Server
>	<	UDP	123	123		Anfrage und Antwort zwischen NTP-Servern

[33] Ein in der Nähe stationierter Langwellensender könnte natürlich auch in diesem Fall eine falsche Uhrzeit simulieren.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		UDP	123	123	Multicast: 224.0.1.1	Multicast Anfrage oder Antwort von NTP-Servern
>		UDP	123	123	Broadcast	Broadcast Anfrage oder Antwort von NTP-Servern

38 Oracle SQL*Net Net8

*SQL*Net* und *Net8* sind die Netzwerkschnittstellen für Oracle7 bzw. Oracle8. Die Portwahl ist stark von der Konfiguration der Clients und des Servers ab. Sowohl SQL*Net als auch Net8 setzen auf den *Transparent Network Substrate* (oder auch *TNS-Listener* genannt) von Oracle auf. Oracle benutzt auch einen eigenen Namensdienst – *Oracle Names*. Der TNS-Listener und auch Oracle Names übertragen Befehle und Daten über denselben Port. Befehle können per Passwort geschützt werden – jedoch nur jene, die Oracle selbst als gefährlich einstuft. Version 1 von SQL*Net verschlüsselt weder Daten noch Benutzerauthentifizierung. SQL*Net Version 2 verschlüsselt die Authentifizierung, sie kann aber wiederverwendet werden. Ab Net8 verwendet auch der Namensdienst ein Kontrollpasswort. Die Daten können durch die Option *Oracle Advanced Networking Option* (*ANO*) verschlüsselt werden. Damit werden auch Einmal-Passwörter unterstützt.

Die Kontrolle von SQL*Net und Net8 ist nur mittels Proxies einigermaßen sicher über eine Firewall zu führen. Man ist aber auf Proxies von Oracle angewiesen, die mur den Binärcode einigen Firewall-Herstellern zur Verfügung stellen. Für SQL*Net gibt es von Oracle den *Oracle Multiprotocol Interchange Server*. Der *Oracle Connection Manager* unterstützt den Proxy-Einsatz für SQL*Net– und Net8–Clients. Die Möglichkeiten der Zugriffskontrolle sind ziemlich eingeschränkt. Einfaches NAT kann nicht eingesetzt werden, weil eingebettete Adressen verwendet werden. Da Oracle die Details des verwendeten Protokolls nicht freigibt und die Proxies von Oracle kein NAT unterstützen, existiert derzeit keine Lösung für den Einsatz von NAT.

Als Quellport verwendet Oracle TCP-Ports oberhalb 1023. Wie schon erwähnt, können diese durch die Konfiguration geändert werden. Das Standardport für den TNS-Listener ist Port 1521, der Oracle Multibprotocol Interchange Listener verwendet Port 1526, Oracle Names verwendet Port 1575 und der Oracle Connection Manager Port 1600.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	1575	NEW, ESTABLISHED	Clientanfrage an Namensdienst Server
	<	TCP	1575	1024:	ESTABLISHED	Serverantwort von Namensdienst
>		TCP	1024:	1600	NEW, ESTABLISHED	Clientanfrage an Server über Connection Manager
	<	TCP	1600	1024:	ESTABLISHED	Serverantwort über Connection Manager
>		TCP	1024:	1521	NEW, ESTABLISHED	Clientanfrage an Server über TNS-Listener
	<	TCP	1521	1024:	ESTABLISHED	Serverantwort über TNS-Listener
>		TCP	1024:	1526	NEW, ESTABLISHED	Clientanfrage an Server über Multiprotocol Interchange Listener

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
	<	TCP	1526	1024:	ESTABLISHED	Serverantwort über Multiprotocol Interchange Listener
>		TCP	1204:	1024:[34]	NEW, ESTABLISHED	Umgeleitete Anfrage von Client an Server
	<	TCP	1024:	1024:	ESTABLISHED	Antwort von Server auf umgeleitetet Anfrage

39 OSPF – Open Shortest Path First

OSPF ist wie RIP ein Routingprotokoll. Es ist neuer und unterstützt Authentifizierung und sogenannte kryptographische *Message Digests*. Der Verschlüsselungsalgorithmus ist allerdings nicht definiert und verwendet in der Praxis oft nur ein acht Zeichen langes Klartextpasswort.

OSPF verwendet weder TCP noch UDP, sondern setzt direkt auf IP auf und trägt die Protokollnummer 89. Es verwendet deshalb auch keine Ports, sondern Pakettypen. OSPF verwendet sowohl Unicast- als auch Multicast-Adressen. Es werden die beiden Multicast-Adressen 224.0.0.5 (AllSPFRouters) und 224.0.0.6 (AllDRouters) verwendet. Da OSPF-Pakete mit einer TTL=1 versandt werden, sind sie nicht für eine Weiterleitung vorgesehen.

Zieladresse	Protokoll	Pakettyp	Anmerkung
224.0.0.5	89	1	Begrüßung der Nachbarrouter
Nachbarrouter	89	2	Bekanntgabe der Daten aus 'Link-State-Tabelle'
Nachbarrouter	89	3	Link-State-Anfrage für bestimmten Pfad
Nachbarrouter	89	4	Link-State-Aktualisierung für bestimmten Pfad
224.0.0.5	89	4	Link-State-Aktualisierung; Informationen über alle Link-States per Flooding-Verfahren von einem designierten Router
224.0.0.6	89	5	Bestätigung einer Link-State-Aktualisierung von nichtdesigniertem Router
224.0.0.6	89	4	Link-State-Aktualisierung von nichtdesigniertem Router
224.0.0.5	89	5	Bestätigung einer Link-State-Aktualisierung von designiertem Router

[34] Port wird vom Server dynamisch zugewiesen.

40 pcAnywhere – Remote Konsolsteuerungsprogramm für Windows

pcAnywhere ist ein weit verbreitetes Programm, um Windows-Rechner von der Ferne aus zu bedienen. Es werden 2 parallele Verbindungen zwischen Client und Server aufgebaut. Der Steuerungskanal verwendet das UDP-Protokoll (Port 22 oder 5632), der Datenkanal läuft über TCP-Port 65301 oder 5631.

Ältere Versionen (V2.0, V7.0, V7.50, V7.51 und CE) verwendet TCP-Port 65301 und UDP-Port 22. Ab Version 7.52 werden TCP-Port 5631 und UDP-Port 5632 verwendet. Es ist zu beachten, dass die Portwahl in der Registry-Datenbank verändert werden kann.

41 POP – Post Office Protocol

POP ist ein Protokoll zum Zugriff auf Mailboxen. POP überträgt Benutzerkennung, Passwort und Daten unverschlüsselt. Es gibt auch sichere Versionen von POP: KPOP (mit Unterstützung von Kerberos) und APOP (ein Challenge-Response System). Darüberhinaus kann POP auch in Verbindung mit SSL oder TLS eingesetzt werden. Für alle sicheren Varianten sind aber sowohl spezielle Server als auch Clients nötig.

Die aktuelle POP Version ist POP3 und verwendet TCP-Port 110. POP2 benutzte TCP-Port 109. Server, die SSL anbieten, verwenden TCP-Port 995.

Die Verwendung von Proxies wird von den meisten Mail-Clients unterstützt und sollte keine Probleme bereiten. Auch NAT ist einfach einzusetzen, da keine eingebetteten IP-Adressen verwendet werden.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	110	NEW, ESTABLISHED	POP3 Verbindung Client -> Server
	<	TCP	110	1024:	ESTABLISHED	POP3 Serverantwort
>		TCP	1024:	109	NEW, ESTABLISHED	POP2 (veraltet) Verbindung Client -> Server
	<	TCP	109	1024:	ESTABLISHED	POP2 Serverantwort
>		TCP	1024:	995	NEW, ESTABLISHED	POP über SSL Verbindung Client -> Server
	<	TCP	995	1024:	ESTABLISHED	POP über SSL Serverantwort

42 PostgreSQL

PostgreSQL ist eine OpenSource Datenbank und neben *MySQL* die unter den kostenlos verfügbaren Datenbanken am meisten verbreitete relationale Datenbank. In niedrigeren Versionen trug sie noch den Namen *Postgres 95*. Sie erlaubt auch Datenbankzugriffe über Netzwerke. Unter anderem unterstützt PostgreSQL Authentifizierung mittels MD5-Verschlüsselung oder Kerberos und Datenübertragungen können mit SSL verschlüsselt werden.

Zur Kommunikation wird TCP-Port 5432 verwendet, das kann aber geändert werden. Ob Proxies für PostgreSQL existieren, ist mir derzeit nicht bekannt. Die Behandlung mit Paketfiltern ist problemlos. Auch NAT kann verwendet werden. Dabei muss jedoch auf die Berechtigungen aufgepasst werden. Da in der Datei *'$PGDATA/pg_hba.conf'* Zugriffsberechtigungen für ganze Netze bzw. einzelne Rechner laut IP-Adresse vergeben werden, wird jener Eintrag wirksam, der auf die IP-Adresse des NAT-Systems passt. Einschränkungen, die sich hinter dem NAT-System befinden, können deshalb nur mehr durch Benutzerkennung und Passwort gemacht werden. Es ist auch darauf zu achten, dass sich 'hinter' dem NAT-System ältere Clients

(Version <7.2) befinden können, welche die MD5-Verschlüsselung noch nicht beherrschen. Jenen Clients würde der Zugriff verweigert.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	5432	NEW, ESTABLISHED	PostgreSQL-Verbindung Client -> Server
	<	TCP	5432	1024:	ESTABLISHED	Serverantwort

43 PPTP – Point-to-Point Tunneling Protocol

PPTP verwendet das Point-to-Point Protocol (PPP), welches die Übertragung von IP-Paketen über Wählverbindungen ermöglicht und das Generic Routing Encapsulation Protocol (GRE), das Daten verschlüsselt. PPTP erweitert PPP dahingehend, dass es PPP-Pakete mittels GRE verschlüsselt und versendet. Es wird für den Aufbau von VPN's verwendet.

GRE ist ein Protokoll, das direkt auf IP aufsetzt und die Protokollnummer 47 besitzt.[35]

PPTP hat abgesehen von mehreren Sicherheitslücken welche Microsoft bei der Implementierung passiert sind (die meisten davon inzwischen behoben) prinzipielle Schwächen. Es schützt zwar die Daten durch Verschlüsselung, die Verhandlungen beim Verbindungsaufbau passieren jedoch unverschlüsselt. Es sollte deshalb keinesfalls die 'schwache' LanManager-Authentifizierung verwendet werden. Es gibt bessere Möglichkeiten VPN's zu realisieren!

Sind die Quelladressen bekannt, kann ein Paketfilter die Sicherheit erhöhen.

Der Einsatz eines Proxy ist nur dann sinnvoll, wenn dieser das Protokoll soweit beherrscht, dass die kritische Phase des Verbindungsaufbaues geschützt werden kann.

NAT ist möglich, das NAT-System muss aber neben TCP und UDP auch das Protokoll GRE beherrschen!

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		GRE (47)				Tunneldaten – Client an Server
	<	GRE (47)				Tunnelantwort – Server an Client
>		TCP	1024:	1723	NEW, ESTABLISHED	Setup-Anfrage von Client
	<	TCP	1723	1024:	ESTABLISHED	Setup-Antwort vom Server

44 quotd

Wird eine Verbindung zu TCP- oder UDP-Port 17 hergestellt, wird ein 'Zitat des Tages' aus einer 'Zitatdatei' ausgegeben. *quotd* ist zwar ungefährlich, aber auch unnützlich[36] und sollte durch die Firewall blockiert werden.

[35] Mit 'iptables' unter Linux kann dieses Protokoll wie folgt gefiltert werden: `# iptables -A xxx -p 47 -j ACCEPT`

[36] Unnützlich aus der Sicht eines normalen Anwenders. Und über eine Firewall sollten nur die Dienste ermöglicht werden, die unbedingt nötig sind.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		UDP, TCP	1024:	17	NEW, ESTABLISHED	Client Anfrage
	<	UDP, TCP	17	1024:	ESTABLISHED	Server Antwort

45 r – Befehle rsh, rlogin, rcp, rdump, rrestore, rdist

Die BSD-Unix *r-Befehle* ermöglichen unkomplizierten Fernzugriff ohne Passwortabfrage. Leider sind sie in sicherheitsgefährdeten Bereichen gefährlich und sollten dort nicht zum Einsatz kommen. Die Authentifizierung erfolgt nur durch die IP-Adresse. Das Resource-Kit von Microsoft Windows NT unterstützt auch r-Befehle. In Windows-Systemen sind r-Befehle noch gefährlicher, da sie nicht ins Sicherheitskonzept des Betriebssystem passen. *rexec* verhält sich anders und wird deshalb getrennt behandelt.

rlogin benützt TCP-Port 513. *rsh, rcp, rdump, rrestore* und *rdist* verwenden alle TCP-Port 514. Ungewöhnlich ist, dass alle Clients beliebige Ports unterhalb 1024 verwenden. Manche Clients des Servers auf Port 514 verwenden für Fehlermeldungen eine zusätzliche TCP-Verbindung, welche beidseitig ein beliebiges Port unterhalb von 1024 benützt. Da diese Verbindung vom Server aufgebaut wird, erschwert es die Kontrolle durch die Firewall zusätzlich.

Aufgrund des Risikos, r-Kommandos im Internet einzusetzen, gibt es außer für *rlogin* kaum Proxies. Da die Authentifizierung durch die IP-Adresse erfolgt, ist NAT problematisch.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	:1023	513	NEW, ESTABLISHED	rlogin Client -> Server
	<	TCP	513	:1023	ESTABLISHED	rlogin Serverantwort
>		TCP	:1023	514	NEW, ESTABLISHED	rsh, rcp, rdump, rrestore, rdist Client -> Server
	<	TCP	514	:1023	ESTABLISHED	rsh, rcp, rdump, rrestore, rdist Serverantwort
	<	TCP	:1023	:1023	NEW, ESTABLISHED	rsh Fehlerkanal Server -> Client
>		TCP	:1023	:1023	ESTABLISHED	rsh Fehlerkanal Clientantwort

46 RADIUS – Remote Authentication Dial-in User Service

Radius wird verwendet, um den Zugriff eines Benutzers, der sich über eine Wahlleitung zu einem Server verbinden möchte, zu kontrollieren. Dieser Server wendet sich an den Radius-Server und ist somit Radius-Client. Im ersten Schritt erfolgt die Authentifizierung des Benutzers - dafür kontaktiert der Radius-Client (aus Sicht des Benutzers, der Server) den Radius-Server über UDP-Port 1812. Anschließend erfolgen Nutzungsbenachrichtigungen über UDP-Port 1813. Ältere Radius-Versionen verwendeten die Ports 1645 und 1646 dafür.

Generische Proxies sind beim Radius-Protokoll nicht verwendbar. Ebensowenig kann NAT eingesetzt werden, weil die Quell-IP-Adresse Teil der Authentifizierung ist. Radius erfordert spezielle Proxies.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		UDP	1024:	1812	NEW, ESTABLISHED	Authentifizierungsanfrage an Raidus-Server
	<	UDP	1812	1024:	ESTABLISHED	Authentifizierungsantwort von Radius-Server
>		UDP	1024:	1813	NEW, ESTABLISHED	Nutzungsbenachrichtigung an Radius-Server
	<	UDP	1813	1024:	ESTABLISHED	Nutzungsantwort von Radius-Server

47 RAS – Remote Access Service

RAS wurde von Microsoft für Windows entwickelt, um Clients den Zugang zu einem Server über eine Modem (oder ISDN, X25) zu ermöglichen. Der Server kann entweder so konfiguriert werden, dass der Client eine Verbindung nur zum Server herstellen kann. Oder er kann als Router arbeiten und den Zugang zum gesamten Netz ermöglichen. Mittlerweile wird *RAS* häufig verwendet um VPN's zu realisieren. Dabei wird als Protokoll entweder PPTP (Point-to-Point Tunneling Protocol) oder L2TP (Layer 2 Transport Protocol) ab Windows 2000 verwendet.

Details siehe Kapitel PPTP oder L2TP.

48 RDP – Remote Desktop Protocol

RDP wird für Microsoft Terminal Services verwendet und ist eine Erweiterung des 'Telecommunications Union T.120 Standards'. RDP bietet unterschiedliche Verschlüsselungsstufen an (40 Bit bis 128 Bit RC4-Verschlüsselung). Es ist ein sich 'ganz normal' verhaltendes TCP-Protokoll und kann einfach über generische Proxies geführt werden. Es ist auch zusammen mit NAT verwendbar.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	3389	NEW, ESTABLISHED	RDP-Verbindung Client -> Server
	<	TCP	3389	1024:	ESTABLISHED	RDP-Verbindung Server -> Client

49 RealNetworks Protokolle RealAudio und RealVideo

RealAudio und RealVideo verwenden im Gegensatz zu den meisten anderen Protokollen UDP für die Übertragung der Daten. Es ist weniger kritisch, wenn Pakete verloren gehen, als wenn der Datenfluss verzögert wird. Für die Sitzungskontrolle wird das TCP-Port 7070 verwendet. Um den Durchsatz zu erhöhen, können mehrere Streams geöffnet werden. Das macht die Kontrolle durch eine Firewall schwierig. Es ist jedoch (mit dem Nachteil von Performanceeinbußen) möglich, den RealNetworks-Client so zu konfigurieren, dass nur TCP auf Port 7070, oder auch TCP Port 7070 und dazu ein einzelnes UDP-Port im Bereich von 6970 – 7170 verwendet werden.

RealNetworks stellt eigene Proxies zur Verfügung.

NAT kann nur dann verwendet werden, wenn TCP alleine (ohne UDP für die Daten) verwendet wird. Bei UDP werden IP-Adressen eingebettet.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	7070	NEW, ESTABLISHED	Anfrage Sitzungskontrolle von Client an Server
	<	TCP	7070	1024:	ESTABLISHED	Antwort Sitzungskontrolle von Server
	<	UDP	6970:7170	1024:		Daten von Server zum Client

50 REMOTE, RCMD und RCONSOLE – Windows Fernzugriffshilfen

Im Resource Kit von Windows NT werden Programme mitgeliefert, die den Fernzugriff ermöglichen. Ab Windows 2000 werden diese Funktionen vom Terminal Server unterstützt. Alle diese Dienste verwenden das SMB-Protokoll, sodass bei Firewalls mit dessen Problemen zu kämpfen ist. Der Einsatz über eine Firewall sollte vermieden werden.

Keines dieser Tools ermöglicht ein Arbeiten wie an der Konsole selbst, da keine graphische Oberfläche unterstützt wird.

Bei *REMOTE* muss am Server ein Programm gestartet werden, welches dann von der Ferne kontrolliert werden kann. Es erfolgt keine Authentifizierung!

Um *RCMD* zu verwenden, wird am Server ein Dienst gestartet und man kann dann aus der Ferne beliebige Kommandos ausführen. Sowohl REMOTE als auch RCMD haben nur eingeschränkte Ein- Ausgabefunktionen, sodass Programme wie *edit* nicht funktionieren. RCMD verwendet die Windows NT-Authentifizierung. Die Befehle werden mit den Rechten des Client-Benutzers ausgeführt (bei älteren Versionen war das fälschlicherweise nicht so!!).

Das sinnvollste dieser 3 Kommandos ist *RCONSOLE*, da es ein Arbeiten mit dem vollen Funktionsumfang eines DOS-Fensters ermöglicht. Außerdem wird die Authentifizierung verschlüsselt und die Datenübertragung kann auf Wunsch verschlüsselt werden. Standardmäßig können nur Mitglieder der Administratoren RCONSOLE verwenden. Durch Hinzufügen eines Users in die Gruppe der 'RConsole User' ist es auch anderen Usern möglich RCONSOLE zu verwenden.

51 RemoteView

RemoteView ist ein Produkt, welches bei Intel-Servern von Fujitsu-Siemens eingesetzt wird. Ein spezielles 'RSB-Board' ermöglicht den Zugriff auf die Maschine selbst im ausgeschalteten Zustand. Man kann damit den Server aus der Ferne ein- oder ausschalten, BIOS-Einstellungen durchführen, sich das aktuelle Bild der Konsole anzeigen lassen, etc. Die Verbindung des RemoteView-Frontends und des Servers kann über LAN oder auch über eine Modemstrecke erfolgen. 'RomPilot' (eine Funktion des BIOS) kann per SNMP Traps an das Frontend senden. Dafür wird standardmäßig UDP-Port 9162 verwendet. Die Telnet-Verbindung zum RSB-Board erfolgt über TCP-Port 2307. Die Telnet-Verbindung kann über SSL gesichert werden. Es ist auch möglich, eine Verbindung über einen Webbrowser herzustellen.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	2307	NEW, ESTABLISHED	RemoteView-Frontend -> Server
	<	TCP	2307	1024:	ESTABLISHED	Serverantwort
	<	UDP	1024:	9162		SNMP-Trap von Server zum RemoteView-Frontend

Der Einsatz eines generischen Proxy ist möglich. Es kann auch NAT verwendet werden, allerdings muss das NAT-System in der Lage sein, eintreffende SNMP-Traps an den Rechner mit dem RemoteView-Frontend weiterzuleiten.[37]

52 rexec

rexec wird meist mit den r-Befehlen zusammen genannt, verhält sich aber aus Sicht einer Firewall anders als diese. Der rexec-Dämon wird auf vielen UNIX-Systemen automatisch gestartet (/etc/inetd.conf) obwohl er kaum verwendet wird. Bei IRIX (das Unix-Derivat von SGI) verwendet rexec für die Softwareinstallation. Im Gegensatz zu den r-Kommandos erfolgt die Authentifizierung nicht über die IP-Adresse des Clients, sondern über Benutzerkennung und Passwort. Allerdings werden diese Daten unverschlüsselt übertragen und die meisten Implementierungen unterstützen keine Protokollierung.

rexec verwendet das TCP-Port 512. Da es wenig verwendet wird, existieren auch kaum Proxies. Die Anpassung von rexec-Clients sollte aber nicht schwierig sein, sofern die Quellen vorhanden sind. NAT kann eingesetzt werden - es werden keine eingebetteten IP-Adressen verwendet.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	512	NEW, ESTABLISHED	Verbindung Client > Server
	<	TCP	512	1024:	ESTABLISHED	Serverantwort

53 RIP – Routing Information Protocol

RIP war das erste Routing-Protokoll, das eingesetzt wird, um Router darüber zu informieren, welche Netze über welchen Weg am günstigsten erreichbar sind. RIP bietet keinerlei Schutzmechanismen. Eine neuere Version RIP-2 unterstützt den Einsatz von Passwörtern, welche aber unverschlüsselt und wiederverwendbar übertragen werden. Erst noch neuere RIP-2 Varianten unterstützen MD5-Verschlüsselung.

RIP verwendet UDP auf Port 520 und macht intensiven Gebrauch von Broadcastadressen.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		UDP	1024:	520	NEW, ESTABLISHED	RIP-Anfrage Client -> Server
	<	UDP	520	1024:	ESTABLISHED	Serverantwort
	<	UDP	520	520	Broadcast!!	Aktualisierungsdaten von Server

[37] Mit 'iptables' kann die Weiterleitung von SNMP-Traps mit folgenden Kommandos erreicht werden. Das NAT-System hat die Adresse 192.168.0.1 auf der Schnittstelle eth0 nach außen. Der Rechner, auf dem sich die Überwachungssoftware befindet, hat die Adresse 172.16.0.10. Es wird Destination-NAT durchgeführt.

iptables -A PREROUTING -t nat -p udp -d 192.168.0.1 –dport 162 -j DNAT –to 172.16.0.10:162 -i eth0
iptables -A PREROUTING -t nat -p udp -d 192.168.0.1 –dport 9162 -j DNAT –to 172.16.0.10:9162 -i eth0

54 RPC – Remote Procedure Call

Wenn irgendwie möglich, RPC-Protokolle niemals durch Firewall hindurch verwenden! Wenn überhaupt, dann über Proxy. Reine Paketfilter können solche Protokolle nicht behandeln, weil die Portnummern beliebige Werte haben können. RPC-Dienst ev. auf eigenen 'Opferhost' verlagern.

Es existieren Firewalls, die mit dem Location-Dienst (*portmapper* bei sun-rpc bzw. *RPC-Locator* bei Microsoft) kommunizieren können. Bei Microsoft-RPC können Portbereiche begrenzt werden (HKEY_LOCAL_MACHINE\-Software\Microsoft\RPC\. . .).

Auch für Proxies ist RPC schwierig zu meistern, aber es gibt Proxies dafür. Z.Bsp.: TIS-FWTK

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		UDP	1024:	111	NEW, ESTABLISHED	Client-Anfrage an sun-rpc portmapper
	<	UDP	111	1024:	ESTABLISHED	Antwort des sun-rpc portmappers
>		TCP	1024:	111	NEW, ESTABLISHED	Client-Anfrage an sun-rpc portmapper
	<	TCP	111	1024:	ESTABLISHED	Antwort des sun-rpc portmappers
>		UDP	1024:	135	NEW, ESTABLISHED	Client-Anfrage an Microsoft RPC-Locationserver
	<	UDP	135	1024:	ESTABLISHED	Antwort des Microsoft RPC-Locationserver
>		TCP	1024:	135	NEW, ESTABLISHED	Client-Anfrage an Microsoft RPC-Locationserver
	<	TCP	135	1024:	ESTABLISHED	Antwort des Microsoft RPC-Locationserver
>		UDP	1024:	any	NEW, ESTABLISHED	Client-Anfrage an RPC-Server
	<	UDP	any	1024:	ESTABLISHED	Antwort des RPC-Server
>		TCP	1024:	any	NEW, ESTABLISHED	Client-Anfrage an RPC-Server
	<	TCP	any	1024:	ESTABLISHED	Antwort des RPC-Server

RPC verwendet keine eingebetteten IP-Adressen, sodass NAT prinzipiell möglich ist. Aber es werden Portnummern zurückgemeldet, weshalb NAT-Systeme welche Portnummern verändern in der Lage sein müssen, die Antworten des Location-Server zu interpretieren! Außerdem können Protokolle, die über RPC kommunizieren sehr wohl IP-Adressen im Datenfeld eingebettet haben können. NIS und NFS verwenden die Absender-IP-Adresse zur Authentifizierung! DCOM (bei Microsoft sehr viel verwendet) verwendet eingebettete IP-Adressen! Es gibt die Möglichkeit, DCOM über HTTP zu betreiben.

55 rsync

rsync ist ein Programm, um Daten über langsame Netzwerke zu synchronisieren. Es verwendet einen ausgeklügelten Algorithmus, um nur die Differenzen zu übertragen und dadurch Bandbreite zu sparen. *rsync* kann zusammen mit rsh oder ssh verwendet werden. Zudem gibt es *rsyncd*, einen eigenen rsync-Server. Bei der Übertragung von vertraulichen Daten sollte unbedingt ssh verwendet werden.

rsyncd verwendet TCP-Port 873. Die Sicherheitsrisiken und Eigenschaften von rsh werden in Kapitel 45 auf Seite 35 besprochen. Der rsync-Client unterstützt den Einsatz von HTTP-Proxies (der Proxy muss dazu Verbindungen zu Port 873 herstellen können). ssh wird in Kapitel 59 auf Seite 43 beschrieben. Das rsync-Protokoll lässt sich einfach über Proxies führen. NAT kann zusammen mit dem rsync-Protokoll problemlos verwendet werden.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	873	NEW, ESTABLISHED	Client Anfrage
	<	TCP	873	1024:	ESTABLISHED	Server Antwort

56 RTP – Realtime Transport Protocol RTCP – Realtime Transport Control Protocol

RTP ist ein Protokoll für die Übertragung von Echtzeit-Daten und wird meist in Zusammenhang mit H.323 als Lowlevel-Protokoll verwendet. *RTCP* wird zusammen mit RTP für die Steuerung verwendet. RTP verwendet bei IP-Implementierungen im Normalfall UDP. RTP verwendet meist UDP-Port 5004. RTCP verwendet UDP-Port 5005 (Manchmal auch 24032 und 24033).

RTP und RTCP sind einfache UDP-Protokolle, die durch simple generische Proxies zu behandeln sind. NAT kann verwendet werden, die Daten können aber Hostnamen und Adressen verraten. Der Einsatz dieser beiden Protokolle würde keine große Gefahr darstellen, meist werden sie jedoch zusammen mit höheren Protokollen verwendet.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		UDP	1024:	5004	NEW, ESTABLISHED	RTP-Verbindung Client -> Server
	<	UDP	5004	1024:	ESTABLISHED	RTP Serverantwort
>		UDP	1024:	5005	NEW, ESTABLISHED	RTCP Client -> Server
	<	UDP	5005	1024:	ESTABLISHED	RTCP Serverantwort

57 SMTP – Simple Mail Transfer Protocol

SMTP ist eines der einfachsten, aber auch am meisten verbreiteten Protokollen. SMTP-Server sind aber aufgrund der Verbreitung auch häufig Angriffsziele. Das von SMTP verwendete TCP-Port 25 ist bei vielen Firewalls offen. Es sollten niemals SMTP-Pakete vom Internet direkt ins lokale Netz gelangen, sondern immer über einen Bastion-Host geführt werden. Für den internen Mailverkehr sollte ein Mailserver im lokalen Netz betrieben werden, damit keine vertraulichen Daten in die gefährdete DMZ gelangen können.

SMTP fungiert selbst schon als Proxy, da ein Mailserver die Mails immer von einem Server zum anderen weiterreicht.

NAT ist möglich, da ein Mailserver aber mit dem eigenen Rechnernamen begrüßt wird, könnte es Probleme geben, falls die IP-Adresse nicht zum Hostnamen passt. Werden die Mails über einen Mailserver des Internetproviders versandt und dieser akzeptiert solche Unstimmigkeiten, ist das OK. Möchte man seine Mails aber direkt an den Ziel-Mailserver senden, so muss sichergestellt werden, dass IP-Adresse und Hostnamen einem Test mittels 'double-reverse DNS-Lookup' bestehen. Sonst werden Mails von manchen Servern nicht angenommen.

Der Mailer am Bastionhost darf nur Mails aus dem Internet annehmen, deren Zieladresse sich im lokalen Netz befindet (andernfalls wird der Server für SPAM-Mail missbraucht!). Mails, die vom internen Netz nach außen gehen, sollen an den Bastionhost geschickt werden, der sie dann erst weiterleitet.

Auf Bastionhosts sollte kein Mailsystem eingesetzt werden, das üblicherweise proprietäre Protokolle verwendet und SMTP lediglich aus Kompatibilitätsgründen unterstützt (z.Bsp. Microsoft Exchange oder Lotus Notes). Die Implementierung von SMTP ist manchmal schlecht und unsicher[38], im Internet ist aber nur SMTP von Nutzen.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	25	NEW, ESTABLISHED	Sender an Empfänger
	<	TCP	25	1024:	ESTABLISHED	Empfänger an Sender

Eines der ätesten, mächtigsten und am weitesten verbreiteten Mailsysteme ist 'sendmail'. Es hat traurige Berühmtheit durch eine Vielzahl von entdeckten Sicherheitslücken erlangt. Teils durch seine starke Verbreitung, teils aber auch durch den monolithischen Aufbau und seiner Komplexität. Sehr viele der Sicherheitsprobleme sind behoben und es ist (gute Konfiguration vorausgesetzt) vermutlich nicht unsicherer als kommerzielle Produkte. Vor allem im Zusammenhang mit 'sendmail' bietet sich die Verwendung des *smap*-Paketes an, das als Wrapper für sendmail arbeitet. Es erhöht die Sicherheit dadurch, dass der Prozess *smap*, der die Mails entgegennimmt, keine root-Rechte besitzt. Erst *smapd* reicht die Mails an sendmail weiter. Ein Angreifer hat dadurch keinen direkten Kontakt mit dem komplexen SMTP-Server.

Einige Mailserver, wie z.Bsp. Postfix vermeiden schon aufgrund ihrer Architektur viele der Probleme von sendmail und sind einfacher zu konfigurieren.

57.1 biff

biff[39] ist ein E-Mail Benachrichtigungsdienst. Er soll Anwender benachrichtigen, sobald eine Mail für sie eingetroffen ist. Er sendet dem Anwender über UDP-Port 512 Teile des Nachrichtenheaders und die ersten Zeilen der Nachricht. Da ein Bastion-Host niemals UDP-Pakete ins interne Netz schicken soll, darf biff in einer sicheren Umgebung nicht verwendet werden.

57.2 Microsoft Exchange

MS Exchange ist weit mehr als nur ein Mailserver. Er stellt sonstige Dienste wie News, Kalender, Dokumentenaustausch, Termin- und Adressverwaltung zur Verfügung. Es macht intensiven Gebrauch von Microsoft RPC (siehe Kapitel 54 auf Seite 39) und verwendet abhängig von der Konfiguration noch folgende Protokolle: SMTP, POP, IMAP, NNTP, LDAP, LDAP über SSL und X.400.

Es kann kaum sicher über eine Firewall gebracht werden. Man benötigt protokollfähige Proxies und NAT-Systeme. Wird Microsoft-RPC gesperrt, so ist die Verwendung von MS Exchange nur eingeschränkt möglich (u.a. kein Exchange Administrator). Will man MS Exchange trotz aller Warnungen auf einem Bastionhost einsetzen, sollte man es so konfigurieren, dass die Dienste über eine HTTP-Schnittstelle angeboten werden. Anstelle von Microsoft RPC sollten dann sogenannte *Connectoren* eingesetzt werden, welche die nötigen Daten in andere Protokolle (z.Bsp. SMTP) einbetten. Das mindert aber die Performance.

[38] Mit 'schlecht' ist gemeint, dass sich der SMTP-Server teilweise nicht RFC-konform verhält und mit manchen Kombinationen Probleme bereitet. 'Unsicher' deshalb, weil ein komplexes System, welches viele Features bietet, anfälliger ist, als ein System, das nur eine gewisse Tätigkeit durchzuführen hat.

[39] Biff hieß der Hund des Programmierers, der den Postboten immer anbellte.

57.3 Lotus Notes und Lotus Domino

Auch Lotus Notes (neuere Versionen – Lotus Domino) ist mehr als nur ein E-Mail System. Zur Authentifizierung wird Verschlüsselung mit öffentlichen Schlüsseln verwendet. Daten selbst werden normalerweise nicht verschlüsselt, können aber bei entsprechender Konfiguration doch verschlüsselt. Das kann durch den Server erzwungen werden. Notes-Dokumente können *Lotus-Scripte* enthalten, welche externe Programme aufrufen können. Vor Version 4.5 gab es keine Sicherheitsvorkehrungen! Was Programme tun dürfen, hängt von der Signatur der Dokumente ab. Es sollte maximale Einschränkung auf den Notes-Clients konfiguriert werden!

Notes verwendet *Notes RPC* auf TCP-Port 1352 und ist mit der Firewall einfach zu handhaben. Lotus liefer selbst einen Proxy mit, der Anwendungen erkennt. Notes-Clients können so konfiguriert werden, dass Notes-RPC über HTTP getunnelt wird. Es gibt kommerzielle Firewalls, die Proxies für Notes beinhalten. Für die Verwendung von generischen Proxies können die Notes-Clients konfiguriert werden.

Notes verwendet zwar eingebettete Hostnamen, aber keine IP-Adressen. Das macht die Verwendung von NAT möglich, es können aber unter Umständen ungewollte Informationen nach außen gelangen.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	1352	NEW, ESTABLISHED	Notes Verbindungsaufbau
	<	TCP	1352	1024:	ESTABLISHED	Notes Serverantwort

58 SNMP – Simple Network Management Protocol

SNMP ist ein einfaches Protokoll, das UDP verwendet. Es ist nicht nur ein mächtiges Werkzeug in den Händen eines Administrators, sondern kann auch ein gefährliches Instrument für Angreifer sein. Die meisten Geräte, die den Anschluss an ein Netzwerk zulassen, unterstützen SNMP. Man kann damit den Zustand von Geräten (Auslastung von CPU oder Netzwerk, Temperaturen und Lüfterdrehzahlen, ...) überwachen, Informationen von Betriebssystem und installierter Software erhalten aber auch administrative Aufgaben (Shutdown, Stop- oder Start von Diensten, etc.) erledigen. Man setzt dafür eine sogenannte Management-Station ein, die die gerätespezifischen Informationen kennt und interpretieren kann. Die Station kann dann zyklisch oder bei Bedarf die SNMP-Agenten der Geräte im Netz um die Herausgabe von Informationen bitten oder durch Setzen von bestimmten Werten im fernen Gerät Reaktionen auslösen.

Das Schlimme an SNMP ist, dass die am meisten verbreitete Version 'SNMPv2' als Schutzmechanismus lediglich eine Art Kennwort (Community-Name) unterstützt. Dieser Community-Name wird aber im Klartext über die Leitung gesendet und wird zudem von vielen Administratoren auf dem Standardwert ('public') belassen. Erst SNMPv3 unterstützt Benutzerauthentifizierung und Verschlüsselung von Daten. Version 2 oder gar noch älter sollten keinesfalls in Netzen verwendet werden, denen man nicht vertrauen kann.

Rechner, auf denen mehrere SNMP-fähige Dienste laufen, verwenden oft zusätzlich zum üblichen UDP-Port 161 ein anderes Port. Häufig ist dies UDP-Port 1161. SNMP-fähige Geräte können wichtige Ereignisse mittels SNMP-Trap an eine Überwachungsstation an das UDP-Port 162 senden. Clients verwenden Ports oberhalb 1023.

Der Einsatz von Proxies ist im Prinzip möglich, man kann aber keine allgemein gültige Aussage machen, da die Informationen der Geräte und damit auch die Anwendungen und Reaktionen auf den Managementstationen unterschiedlichster Art sein können. Ein zusätzliches Problem kann die Weiterleitung der Traps darstellen. Ähnliches gilt für NAT: Es ist einsetzbar, da keine eingebetteten IP-Adressen verwendet werden. Jedoch wird häufig sicherheitskritische Information übertragen und viele Managementstationen reagieren auf ein erhaltenes Trap indem sie versuchen, Verbindung zum SNMP-Gerät aufzubauen. Das stößt dann auf Probleme. Ob lösbar oder nicht, hängt vom Einzelfall ab und muss im Detail erarbeitet werden.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		UDP	1024:	161	NEW, ESTABLISHED	Management-Station -> SNMP-Gerät
	<	UDP	161	1024:	ESTABLISHED	Antwort vom SNMP-Gerät
>		UDP	1024:	162		Trap von SNMP-Gerät zur Management-Station

59 SSH – Secure Shell

Die Secure Shell *ssh* ist DER Ersatz für telnet, kann aber noch mehr. Zum Beispiel ist es möglich mittels *ssh* andere Protokolle zu 'tunneln'.[40] Das gelingt mit Hilfe der sogenannten Port-Weiterleitung. Ein Sonderfall der Port-Weiterleitung ist die Unterstützung von X11 Window-Systemen. ssh verhindert auch sogenannte 'man in the middle Attacken'.[41]

ssh existiert derzeit in 2 Versionen. Da in Version 1 Sicherheitslücken erkannt wurden, sollte man nach Möglichkeit nur Version 2 einsetzen (in den beiden Dateien ssh_config und sshd_config durch den Parameter *Protocol 2*). ssh unterstützt auch die Authentifizierung durch rhosts, wie sie bei den r-Kommandos verwendet wird. Das ist allerdings ein Sicherheitsrisiko und deshalb auch nicht standardmäßig aktiv. Die nächste Sicherheitsstufe wird durch eine Passwort-Authentifizierung erreicht. Gegenüber telnet- oder rlogin-Verbindungen werden Benutzer- und Kennwortdaten bereits verschlüsselt übertragen. Die sichersten Methoden sind jedoch entweder RSA-Authentifizierung oder Kerberos V5-Authentifizierung.

Der Einsatz eines Proxy ist möglich, aufgrund der verschlüsselten Daten kann aber kaum an Sicherheit dazugewonnen werden. NAT funktioniert.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:[42]	22	NEW, ESTABLISHED	SSH-Verbindung Client -> Server
	<	TCP	22	1024:	ESTABLISHED	Server-Antwort

60 Sybase Datenbankprotokolle

Sybase unterstützt verschiedene Netzwerkprotokolle für den Zugriff auf die Datenbanken: TDS (siehe auch Kapitel 65 auf Seite 46), IIOP und HTTP. IIOPS und HTTPS bieten gesicherte Übertragung. IIOP verhält sich ähnlich wie CORBA (siehe Kapitel 3 auf Seite 6). Alle diese Protokolle verwenden einfache TCP-Verbindungen über Ports oberhalb 1023. Die Portnummern können am Datenbankserver konfiguriert werden. Die Standardports lauten: 7878 bei TDS, 8080 bei HTTP, 8081 und 8082 bei HTTPS, 9000 bei IIOP, 9001 und 9002 bei IIOPS.

Die allgemein verbreiteten HTTP-Proxies können zusammen mit der HTTP-Unterstützung von Sybase verwendet werden. Für TDS gibt es Proxies von einigen Firewall-Herstellern. Muss der Zugriff auf eine

[40] Beim Einrichten eines Tunnels ist darauf zu achten, dass u.U. die Firewall umgangen wird!

[41] Kritisch ist ein erster Verbindungsaufbau zu einem Server, wenn nicht zuvor die öffentlichen Schlüssel ausgetauscht wurden. Zu diesem Zeitpunkt könnte sich ein Angreifer dazwischenhängen, selbst die benötigten Schlüssel generieren und alle Daten einschließlich Benutzernamen und Kennwort mitlesen. Ein späterer Versuch, sich mit dem Server zu verbinden, würde eine Warnung erzeugen, wenn der Angreifer nicht mit seinem Verschlüsselungsystem dazwischen hängt. Solche Warnungen sind deshalb immer aufmerksam zu behandeln!

[42] Bei Verwendung von rhosts, werden Ports unterhalb von 1024 verwendet!

Sybase-Datenbank über eine Firewall zugelassen werden, sollte aber HTTPS oder IIOPS verwendet werden. Bei Verwendung von TDS und HTTP ist NAT problemlos einsetzbar.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	7878	NEW, ESTABLISHED	Clientanfrage an Server über TDS
	<	TCP	7878	1024	ESTABLISHED	Serverantwort über TDS
>		TCP	1024:	8080	NEW, ESTABLISHED	Clientanfrage an Server über HTTP
	<	TCP	8080	1024:	ESTABLISHED	Serverantwort über HTTP
>		TCP	1024:	8001, 8002	NEW, ESTABLISHED	Clientanfrage an Server über HTTPS
	<	TCP	8001, 8002	1024:	ESTABLISHED	Serverantwort über HTTPS
>		TCP	1024:	9000	NEW, ESTABLISHED	Clientanfrage an Server über IIOP
	<	TCP	9000	1024:	ESTABLISHED	Serverantwort über IIOP
>		TCP	1024	9001, 9002	NEW, ESTABLISHED	Clientanfrage an Server über IIOPS
	<	TCP	9001, 9002	1024	ESTABLISHED	Serverantwort über IIOPS

61 syslog

syslog wurde ursprünglich in UNIX dazu verwendet, Meldungen des Betriebssystemes oder Programmen zu protokollieren. Inzwischen wird *syslog* auch in Geräten wie Router implementiert. syslog kann auch Meldungen von anderen Rechnern entgegennehmen und weiterleiten (als Proxy) oder selbst verarbeiten. Aus der Sicht der Sicherheit kann syslog-Server in der Art angegriffen werden, dass man ihn mit Daten überflutet, um Plattenplatz zu verbrauchen oder Angriffe zu verstecken. Manche syslog-Implementierungen lassen es zu, Meldungen vom Netz nicht entgegenzunehmen oder auf bestimmte Quelladressen zu beschränken.

syslog verwendet UDP-Port 514. NAT würde zwar funktionieren, da aber im Protokoll der Absender von Interesse ist, würde man dann immer nur die Adresse des NAT-Systems vorfinden. Deshalb ist NAT zusammen mit syslog nicht sinnvoll einsetzbar. Stattdessen kann aber das NAT-System selbst einen syslog-Server betreiben und die Meldungen weiterleiten, was vermutlich dem entspricht, was man wollte. Die meisten Systeme verwenden als Quellport ein beliebiges Port oberhalb 1023. Werden Meldungen von einem syslog-Server zu einem anderen syslog-Server weitergeleitet, wird beidseitig UDP-Port 514 verwendet.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		UDP	1024:	514		Meldung eines Clients an syslog Server
>		UDP	514	514		Weiterleitung einer Meldung von syslog Server zu syslog Server

62 T.120

T.120 ist ein Protokoll für Konferenzen laut Standard der International Telecommunications Union (ITU). T.120 unterstützt Dateiübertragung, Chat, Whiteboard und gemeinsam genutzte Anwendungen. Es verwendet eine einfache TCP-Verbindung zum Port 1503. T.120 definiert keine Sicherheitsmechanismen.

Wegen der Möglichkeit der Dateiübertragung und der gemeinsamen Programmnutzung wäre der Einsatz eines intelligenten Proxy, der das T.120 Protokoll versteht sinnvoll. Derzeit dürfte es noch keine Proxies dafür geben. NAT kann bei T.120 eingesetzt werden.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	1503	NEW, ESTABLISHED	T.120 Verbindung Client -> Server
	<	TCP	1503	1024:	ESTABLISHED	Serverantwort

63 TACACS

TACACS wird, wie Radius, dazu verwendet um Benutzer, die sich zu einem Server einwählen, zu Authentifizieren und den Zugriff zu kontrollieren. Es gibt verschiedene Varianten. Während TACACS und XTACACS unverschlüsselte Benutzer- und Kennwortdaten übertragen, verwendet TACACS+ mit MD5 verschlüsselte Informationen.

TACACS verwendet UDP-Port 49. Es kann auch TCP verwendet werden, dann erfolgt das aber nicht zwingend auf TCP-Port 49. TACACS+ verwendet dagegen sicher TCP-Port 49.

TACACS-Protokolle sind einfach und können durch generische Proxies kontrolliert werden. Da die Art der Schlüssel jedoch nicht definiert wird, könnte es Implementierungen geben, die die IP-Adresse verwenden. Das würde den Einsatz von Proxies schwierig machen. Dasselbe gilt für NAT. Generell sollten die älteren Versionen nicht verwendet werden, sondern lediglich TACACS+.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		UDP	1024:	49	NEW, ESTABLISHED	Clientanfrage an TACACS / XTACACS Server
	<	UDP	49	1024:	ESTABLISHED	Antwort von TACACS / XTACACS Server
>		TCP	1024:	49[43]	NEW, ESTABLISHED	Client -> TACACS / TACACS+ Server
	<	TCP	49	1024:	ESTABLISHED	TACACS / TACACS+ Server -> Client

64 talk

talk ist ein textorientiertes Konferenzsystem, das aber im Gegensatz zu IRC oder ICQ nur 2 Partner miteinander verbindet. Es gibt 2 Versionen des talk-Protokolls. Die ältere Version (*talk*) setzte gleiche CPU-Architektur voraus (wegen der Byteanordnung). Die aktuelle Version (*ntalk*) hat dieses Problem nicht, ist aber nicht kompatibel zur alten.

[43] Bei TACACS könnte es auch ein anderes Port sein.

Der Verbindungsaufbau erfolgt über UDP. Damit wird die endgültige Verbindung über TCP ausverhandelt. Die Verbindungen während des Aufbaues sind sehr komplex, da ein Client nicht nur mit einem Server kommuniziert, sondern mit 'anrufendem Server', 'antwortendem Server' und 'antwortendem Client'. Das alte Protokoll verwendet UDP-Port 517, das neue Protokoll verwendet UDP-Port 518. Die Clients verwendet UDP-Ports oberhalb von 1023. Für die TCP-Verbindungen werden beidseitig beliebige Ports oberhalb 1023 benützt.

Generische Proxies sind nicht verwendbar. Spezielle Proxies existieren nicht und wird es vermutlich auch nicht mehr geben, da dieses Protokoll durch IRC und ICQ immer mehr verdrängt wird. Auch NAT ist kaum einsetzbar, weil das NAT-System das talk-Protokoll verstehen muss. Das talk-Protokoll sollte man nicht über eine Firewall zulassen.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		UDP	1024:	518[44]	NEW, ESTABLISHED	Verbindungsaufbau von Client -> Server
	<	UDP	518	1024:	ESTABLISHED	Serverantwort
>		TCP	1024:	1024:	NEW, ESTABLISHED	Kommunikation Client -> Server
	<	TCP	1024:	1024:	ESTABLISHED	Kommunikation Server -> Client

65 TDS – Tabular Data Stream

TDS ist ein Protokoll für Datenbankzugriffe über Netzwerk. Es wird u.a. von Sybase (Lizenzeigentümer von TDS) und Microsoft SQL-Server verwendet. TDS definiert kein bestimmtes Port, jede Implementierung verwendet ihr eigenes Port. Der Client verwendet eine einzige, einfache TCP-Verbindung zum Datenbankserver. TDS wird sowohl mit als auch ohne Verschlüsselung eingesetzt. TDS kann durch einfache generische Proxies kontrolliert werden.

66 Telnet

telnet ist das am weitesten verbreitete Programm, um aus der Ferne einen Rechner zu bedienen. Clients dafür existieren für die meisten Betriebssysteme. Benutzerkennungen, Passwörter und Daten werden völlig unverschlüsselt übertragen. Ab Microsoft Windows 2000 existiert eine Version, welche sicherer ist. Voraussetzung ist, dass beidseitig Windows 2000 verwendet wird und NTLM-Authentifizierung ermöglicht wird. Es gestattet eine verschlüsselte Authentifizierung. Die Daten werden dennoch unverschlüsselt übertragen. Man sollte generell *telnet* durch sichere Protokolle wie *ssh* ersetzen.

Telnet kann einfach durch Proxies geführt werden. Auch NAT macht keine Probleme.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	23	NEW, ESTABLISHED	telnet Verbindungsaufbau
	<	TCP	23	1024:	ESTABLISHED	Serverantwort

[44] 517 bei der älteren Protokollversion

67 TFTP – Trivial File Transfer Protocol

TFTP ist ein sehr einfaches Protokoll für die Übertragung von Dateien. Es wird u.a. verwendet, um über Netz zu booten oder Firmware-Updates durchzuführen, da es in einem Chip unterzubringen ist und kein aufwändiges Betriebssystem benötigt. Allerdings besitzt es keinerlei Sicherheitseigenschaften wie Authentifizierung oder Verschlüsselung.

TFTP verwendet das UDP-Port 69. Der Einsatz von Proxies ist prinzipiell möglich, kann aber die Sicherheit auch nicht verbessern. Bei NAT ist mit Problemen zu rechnen, da der TFTP-Server die Antworten auf einem anderen Port sendet als Pakete empfangen werden. Zudem werden sowohl vom Client als auch vom Server die Quellports überwacht. Ändert sich der Quellport, wird die Verbindung unterbrochen. Unter Umständen müsste die Firewall für alle UDP-Ports >1023 geöffnet werden, damit TFTP läuft.

Man sollte TFTP niemals über eine Firewall führen!

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		UDP	1024:	69	NEW, ESTABLISHED	TFTP Verbindungsaufbau
	<	UDP	1024:	1024:	[45]	Antwort auf Verbindungsanfrage
>		UDP	1024:	1024:		weitere Pakete vom Client

68 TLS – Transport Layer Security SSL – Secure Socket Layer

SSL und *TLS* ermöglichen gute Verschlüsselung. Bei SSL sollte Version 3 verwendet werden. Version 2 enthält bekannte Sicherheitsprobleme. SSL und TLS werden i.A. von höheren Protokollen verwendet.[46]

TLS und SSL verwenden keine bestimmten Ports. Da sie aber im Zusammenhang mit anderen Protokollen verwendet werden, kommt eine gezielte Portzuweisung häufig vor (Port 443 z.Bsp. für https).

Bei TLS und SSL-Verbindungen macht nur die Verwendung von generischen Proxies Sinn, da die Pakete selbst ohnehin verschlüsselt sind und vom Proxy nicht interpretiert werden können.

NAT ist möglich sofern höhere Protokolle keine eingebetteten IP-Adressen oder Portnummern verwenden.

69 Tooltalk

Tooltalk ist Bestandteil des *CDE* (*Common Desktop Environment*) und ermöglicht z.Bsp. die Übernahme von Objekten zwischen Anwendungen durch Mausziehen. Tooltalk setzt auf Sun-RPC auf und sollte deshalb ebenso nicht über Firewalls geführt werden. Keinesfalls sollte Tooltalk auf Bastion-Hosts eingesetzt werden![47]

70 VNC –Virtual Network Computing

VNC ist ein 'Remote Display System'. Man kann damit auf fernen Systemen arbeiten, als ob man direkt davor sitzen würde, ohne auf eine graphische Oberfläche verzichten zu müssen. Es ist frei verfügbar und

[45] Siehe Text oberhalb der Tabelle!
[46] ssh verwendet in der Version 2 z.Bsp. TLS; https entspricht http + SSL; ESMTP verwendet TLS bei der STARTTLS-Erweiterung;
[47] Auf Bastion-Hosts sollten graphische Oberflächen überhaupt vermieden werden!

steht unter der GNU Public Lizenz, ist plattformunabhängig und sehr kompakt. Es ermöglicht auch den Zugriff über einen JAVA-fähigen Browser.

Der Verbindungsaufbau zum VNC-Server ist passwortgeschützt. Es wird ein zufälliges 'Challenge Response System' verwendet, sodass das Passwort nicht mitgelesern werden kann. Die Daten selbst werden anschließend jedoch nicht verschlüsselt. Ist höhere Sicherheit erforderlich, wird die Verwendung eines SSH-Tunnels empfohlen (eine Anleitung befindet sich in der mitgelieferten Dokumentation).

Erfolgt die Verbindung über das Programm 'vncviewer', wird das Standard TCP-Port 59*xx* verwendet, wobei *xx* die Nummer des Displays ist. Im Normalfall ist das '1', sodass die Portnummer 5901 lautet. Man kann sich auch über einen normalen JAVA-fähigen Webbrowser verbinden. In diesem Fall wird das TCP-Port 58*xx* (also 5801 für das 1. Display) verwendet.

NAT kann problemlos durchgeführt werden. Da die Source frei verfügbar ist, kann sie für die Verwendung mit Proxies kompiliert werden. Es existiert auch ein VncProxy (siehe Dokumentation).

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	5900+*n*[48]	NEW, ESTABLISHED	Verbindungsaufbau von vncviewer zum VNC-Server
	<	TCP	5900+*n*	1024:	ESTABLISHED	Serverantwort
>		TCP	1024:	5800+*n*[49]	NEW, ESTABLISHED	Verbindungsaufbau von Webbrowser per vnc-http zum Server
	<	TCP	5800+*n*	1024:	ESTABLISHED	Serverantwort

Ist man gezwungen, eine graphische Oberfläche über eine Firewall zu führen, möchte aber X-Window aufgrund seiner Sicherheitsrisiken nicht einsetzen, bietet VNC eine bessere Alternative.

71 WAIS

WAIS ist nur mehr selten zu finden. Es wurde verwendet, um große Textdatenbestände zu indizieren, damit darin rasch Textstellen gefunden werden können. Suchmaschinen werden heute auf Webservern mit CGI-Programmen realisiert. Proxies und NAT sollten keine Probleme darstellen.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	210	NEW, ESTABLISHED	Client Anfrage
	<	TCP	210	1024:	ESTABLISHED	Server Antwort

[48] 5901 entspricht Display 1
[49] 5801 ist das 1. VNC-Display

72 whois

whois-Server liefern Daten über Internet-Domänen und deren Verwalter. whois verwendet TCP-Port 43.

Richtung		Proto-	Port		Status	Anmerkung
Client	Server	koll	Quelle	Ziel		
>		TCP	1024:	43	NEW, ESTABLISHED	Client Anfrage
	<	TCP	43	1024:	ESTABLISHED	Server Antwort

73 Windows Fernbetreuungsprogramme

Der Einsatz von Fernbetreuungsprogrammen bestimmt meist den Sicherheitslevel eines Systems. Die Schwachstelle wird nämlich so gut wie immer dort zu finden sein! Es existieren eine Reihe von Produkten für diesen Zweck. Je umfangreicher die Möglichkeiten sind, desto größer ist die Wahrscheinlichkeit, Sicherheitslücken zu öffnen. Wenn nicht darauf verzichtet werden kann, sollte man sich überlegen, ob man sich besser für eine Minimallösung entscheidet. Beispielsweise bietet das frei erhältliche Programm VNC (siehe Kapitel 70 auf Seite 47) keine Dateiübertragung, sondern nur die Arbeit wie an der Graphikkonsole. Es ist darauf zu achten, dass das jeweilige Programm vor allem die Authentifizierung verschlüsselt durchführt und möglichst auch die Daten verschlüsselt. Häufig wird die Domänen-Authentifizierung verwendet. Werden die Anmeldedaten nicht auf sicherem Weg übertragen, so kennt der Angreifer die lokalen Daten!

74 X11 Window-System

Das X11 Window-System ist ein weit verbreitetes System für graphische Oberflächen auf UNIX-Systemen. Für Firewalls ist X11 schwierig zu behandeln und sollte vermieden werden. Wenn es wirklich zum Einsatz kommen muss, sollte man es in Verbindung mit der Secure Shell (ssh) einsetzen.

Das Hauptproblem ist, dass der Verbindungsaufbau entgegengesetzt zu den üblichen Protokollen erfolgt. Der X11-Server befindet sich auf jenem Rechner, auf dem der Benutzer arbeitet (ein X-Terminal oder ein PC mit optionaler X-Server Software). Programme, die am fernen 'Applikations-Server' laufen, stellen eine Verbindung zum X11-Server her, um die Daten darzustellen und Benutzereingaben entgegenzunehmen. Hat man sich dazu entschlossen, keinerlei Verbindungen von außen ins lokale Netz zuzulassen, ist der Einsatz von X11 so nicht möglich! Durch Verwendung von ssh kann X11 getunnelt werden und man umgeht so das Problem vom eingehenden Verbindungsaufbau.

XDMCP (X Display Manager Control Protocol) ermöglicht es 'dummen' X-Terminals, Rechner zu finden, um sich dort anzumelden. X-Terminals versuchen nach dem Start, per Broadcast oder auch per Unicast einen XDMCP-Server zu finden. Unter UNIX ist XDMCP meist im Programm *xdm* (X Display Manager) implementiert.

Der X-Font Server stellt Zeichensätze für X-Terminals zur Verfügung, da diese oft viel Platz in Anspruch nehmen. Ein X-Server, der eine bestimmte Schrift zur Darstellung benötigt, kann diese bei einem X-Font Server anfordern. Dieser wiederum kann die Anfrage auf einen anderen Server mit beliebigem Port weiterleiten.

Wenn XDMCP benötigt wird, sollte es eingeschränkt auf einem Bastionhost laufen. Ein X-Font Server sollte immer im lokalen Netz installiert werden und nie über eine Firewall betrieben werden.

X-Server verwenden TCP-Ports von 6000 an aufwärts. Jedes Display verwendet ein Port. Da die Menge der übertragenen Daten meist sehr groß ist, wirkt sich der Einsatz von Proxies oft durch schlechte Performance aus. Am sinnvollsten wird X11 mit einem SSH-Tunnel verwendet. Für diesen Einsatz gibt es X11-Proxy-Server (z.Bsp. TIS FWTK). NAT ist im Prinzip möglich, da keine eingebetteten IP-Adressen verwendet

werden. Durch die unübliche Richtung des Verbindungsaufbaues, kann es aber dazu kommen, dass kein Weg vom Anwendungsserver zum X-Server gefunden werden kann. Das NAT-System muss speziell für diesen Fall angepasst werden.

Richtung		Proto-	Port		Status	Anmerkung
Client[50]	Server[51]	koll	Quelle	Ziel		
	<	TCP	1024:	6000+*n*[52]	NEW, ESTABLISHED	Verbindungsaufbau von Anwendungsserver zum X-Server
>		TCP	6000+*n*	1024:	ESTABLISHED	Antwort (Benutzereingaben) von X-Server zum Anwendungsserver
>		UDP	1024:	177	NEW, ESTABLISHED	eingehende XDMCP Anforderung von einem X-Terminal
	<	UDP	177	1024:	ESTABLISHED	XDMCP Antwort
>		UDP	1024:	7100	NEW, ESTABLISHED	Anfrage von X-Server > X-Fontserver
	<	UDP	7100	1024:	ESTABLISHED	Antwort von X-Fontserver

[52] Bei älteren Protokollen von SUN wurde Port 2000+*n* verwendet.